
AI translation • View original & related papers at
chinarxiv.org/items/chinaxiv-202204.00083

Multi-Objective Optimization for Vehicle Following Based on Joint Lateral and Longitudinal Control: Postprint

Authors: Li Mengfan, Qin Wenhui, Cloud China

Date: 2022-04-07T15:01:56+00:00

Abstract

To address issues such as poor car-following stability, ineffective tracking, or unsafe conditions caused by significant speed fluctuations of vehicles in congested environments, a multi-objective optimization car-following scheme based on vehicle models and deep reinforcement learning is proposed. First, a vehicle car-following model is established based on the vehicle's lateral and longitudinal dynamics. Then, according to inter-vehicle spacing error, velocity error, lateral deviation, relative yaw angle, and other variables, the Deep Deterministic Policy Gradient algorithm is utilized to obtain the acceleration and steering angle of the following vehicle, thereby enabling smoother and safer control of the following vehicle. Tested and validated on the NGSIM public driving dataset, this scheme can effectively enhance the stability, comfort, and safety of the following vehicle, which is of significant importance for ensuring traffic safety and improving road capacity.

Full Text

Preamble

Vol. 39 No. 8

Application Research of Computers

Accepted Paper

Multi-objective Optimal Car-Following Model with Lateral and Longitudinal Control

Li Mengfan, Qin Wenhui†, Yun Zhonghua

(School of Instrument Science & Engineering, Southeast University, Nanjing 210096, China)

Abstract: To address issues of poor following stability, ineffective tracking, and safety concerns caused by large speed fluctuations in congested traffic, this paper proposes a multi-objective optimal car-following scheme based on vehicle dynamics and deep reinforcement learning. First, a car-following model is established based on vehicle lateral and longitudinal dynamics. Then, using the Deep Deterministic Policy Gradient (DDPG) algorithm, the following vehicle's acceleration and steering angle are determined according to inter-vehicle distance error, speed error, lateral deviation, and relative yaw angle to control the following vehicle more smoothly and safely. Tested and validated on the NGSIM public driving dataset, this scheme effectively improves the stability, comfort, and safety of car-following, which is significant for ensuring traffic safety and enhancing road capacity.

Keywords: car following; lateral and longitudinal control; deep deterministic policy gradient; NGSIM

0 Introduction

Vehicle car-following is an important autonomous driving assistance technology that can reduce driver burden, improve driving comfort, and decrease traffic accidents. However, in congested traffic conditions, frequent acceleration and deceleration lead to poor following performance, making low-speed car-following research a focus of attention [1].

Traditional autonomous car-following models are primarily theory-driven, expressing various states during the following process through mathematical and physical models based on vehicle following behavior to establish models consistent with driving experience. The first car-following model was proposed by Pipes [2], which assumed that the following vehicle's speed is proportional to the inter-vehicle distance and determined the following vehicle's speed based on headway. Subsequently, based on traffic flow heterogeneity, human factors, and road conditions, various car-following models have been proposed, including safety distance-based, psycho-physiological, stimulus-response, and cellular automata models. However, theory-driven car-following models struggle to comprehensively consider these influencing factors, resulting in poor prediction accuracy and insufficient accuracy in describing complex car-following behaviors.

Benefiting from the development of intelligent transportation, large-scale high-precision vehicle trajectory data provides a research foundation for data-driven car-following. By statistically analyzing vehicle trajectory data and mining driving behavior patterns, corresponding fitting relationships can be established to achieve effective vehicle following [3]. Current data-driven car-following models include those based on fuzzy logic, support vector regression, artificial neural networks, and deep reinforcement learning. Among them, deep reinforcement learning car-following models [4] have become a research hotspot in recent years, with methods such as convolutional neural networks, recurrent neural networks, and

long short-term memory networks gradually applied to car-following research. Zhu Meixin et al. [5] used deep reinforcement learning to obtain car-following strategies, establishing a human-imitating mapping model from following speed, relative speed, and inter-vehicle distance to following acceleration. Pan Feng et al. [6] analyzed real driving data based on inverse reinforcement learning to obtain human driver characteristics, designed a reward function, and obtained more natural car-following behavior. Zhu Bing [7] established a car-following control strategy based on the proximal policy optimization algorithm and a longitudinal control architecture based on a dual-predecessor following structure to achieve vehicle following control. Model Predictive Control (MPC) [8] is widely applied in car-following scenarios. Hu Xiaosong et al. from Chongqing University [9] developed an MPC-based controller to optimize vehicle speed and engine torque, achieving better fuel economy and fewer emissions while ensuring following safety. Mao Jin et al. [10] designed a multi-objective optimization method with online updated weight coefficients based on the MPC algorithm, achieving better following tracking performance and stability. Compared with other models, deep reinforcement learning car-following models can continuously learn and adapt to different driving environments, offering better generalization capabilities and helping develop autonomous driving algorithms and traffic flow models more similar to human driving behavior.

When drivers follow vehicles, they typically reduce speed and increase following distance when aware of deviation from the desired trajectory to decrease lateral control risk and longitudinal accident risk [11]. When vehicles travel on roads with changing curvature, insufficient lateral control in the model affects vehicle handling stability. Most car-following research focuses on longitudinal acceleration decision-making while neglecting lateral path tracking, and primarily concentrates on simulating human driving behavior rather than optimizing driving behavior.

This paper proposes a vehicle car-following model that simultaneously decides acceleration and steering angle based on joint lateral and longitudinal control, considering safety and comfort, using deep reinforcement learning algorithms to enable effective following of the preceding vehicle.

1 Vehicle Dynamics Model

In the motion control of a following vehicle, a vehicle dynamics model must first be established. This study simplifies the steering system by directly using the front wheel angle as the steering wheel angle of the following vehicle. Based on vehicle lateral motion, yaw motion, and longitudinal motion, a three-degree-of-freedom vehicle dynamics model is established as follows [12]:

The Actor and Critic networks are used, with the following vehicle that decides acceleration and steering angle as the agent, whose main objective is to maximize the reward function. The Actor network is primarily responsible for policy

generation, i.e., outputting the following vehicle's acceleration based on the speeds, relative speed, and relative distance between the following and preceding vehicles, and obtaining the steering wheel angle based on lateral deviation and relative yaw angle. The Critic network is responsible for policy improvement, outputting $Q(s_t, a_t)$ based on state-action pairs and updating the Actor's policy parameters in the direction of performance improvement.

Figure 1 shows the vehicle dynamics model, where XOY is the ground reference coordinate system and xoy is the vehicle coordinate system. v_x and v_y are the longitudinal and lateral velocities of the vehicle's center of gravity, θ is the vehicle yaw angle, $\dot{\theta}$ is the vehicle yaw rate, and δ is the front wheel steering angle. l_f is the distance from the vehicle's center of gravity to the front axle, l_r is the distance to the rear axle, and I_z is the vehicle's moment of inertia about the vertical axis. $F_{\{xf\}}$ and $F_{\{yf\}}$ are the longitudinal and lateral forces on the front wheels, while $F_{\{xr\}}$ and $F_{\{yr\}}$ are the forces on the rear wheels.

The lateral forces of the front and rear tires are approximately linearly related to their slip angles [13]. In equation (2), α_f and α_r are the front and rear tire slip angles, and C_f and C_r are the front and rear tire cornering stiffness [12].

2 DDPG-Based Vehicle Car-Following Decision Algorithm

The Deep Deterministic Policy Gradient algorithm possesses both the feature extraction capability of deep neural networks and the decision-making advantages of reinforcement learning, making it suitable for car-following decision problems with discrete inputs and continuous outputs. Therefore, this paper establishes an overall car-following strategy based on the DDPG algorithm, as shown in Figure 2.

In congested road sections, the following vehicle's acceleration a and steering angle δ are typically influenced by the preceding vehicle's motion state, necessitating control strategies based on the predecessor's state. After collecting the speed difference, relative distance, lateral deviation, and relative yaw angle between the vehicles, the DDPG algorithm transforms the car-following problem into a Markov decision process under a specific reward function. Through iterative interaction between the deep reinforcement learning agent and the car-following environment, the lateral and longitudinal control strategy for the following vehicle—namely its acceleration and steering angle—is obtained to adjust the following vehicle's motion state and achieve optimal control [14].

2.1 DDPG Algorithm Principle

Deep reinforcement learning consists of an agent that continuously observes and receives rewards while interacting with the environment, and an environment

that changes based on the agent's actions. Deep Q-Networks are suitable for models with few discrete outputs but may fail in continuous action spaces. This study employs the Deep Deterministic Policy Gradient (DDPG) algorithm [15], which performs well in continuous control domains, to learn the Actor and Critic network structures.

The Actor and Critic network architectures shown in Figure 3 consist of input layers, output layers, and hidden layers containing multiple neurons. DDPG first initializes the replay buffer, Actor and Critic network parameters $\hat{\pi}$ and \hat{Q} , and the target network weight parameters $\hat{\pi}'$ and \hat{Q}' for the Actor and Critic. In each training episode, the following vehicle's acceleration and steering angle are calculated based on the Actor. Next, the reward value r_t and next state s_{t+1} are observed. After obtaining the reward and state values, the Critic network evaluates the action a_t taken in the current state s_t , updates the Critic network parameters \hat{Q} according to the loss function L , and updates the Actor network parameters $\hat{\pi}$ using policy gradients. Finally, the target network weights $\hat{\pi}'$ and \hat{Q}' are updated based on the update directions of the Actor and Critic network weights. This process continuously optimizes the Actor and Critic until convergence.

During this process, the optimization objective of the DDPG algorithm [15] is to update the Critic network according to loss function (4). The Actor network is updated using policy gradient (5). After k optimizations, strategy (6) is used to update the target network parameters of the Actor and Critic.

2.2 Vehicle Car-Following Error

During the following process, the following vehicle needs to plan its control strategy based on the preceding vehicle's motion state and trajectory. To track the predecessor's speed and trajectory, characterize the positional, speed, and trajectory relationship between the vehicles, maintain a safe distance, and travel along the desired path, a lateral and longitudinal joint control car-following error model is established as shown in Figure 4.

In vehicle longitudinal motion, the following vehicle determines its acceleration based on vehicle speed and distance to the predecessor to follow safely and effectively. The microscopic driver behavior safety distance model [16] is given by equation (7). Since the speed difference between leading and following vehicles is small during car-following, the $\lambda(v_{\text{follow}} - v_{\text{lead}})^2$ term is ignored. Therefore, based on the fixed time headway algorithm [16], the safety distance model is designed as equation (8).

According to the relative motion relationship between the two vehicles, combined with relative speed and distance error, the vehicle car-following longitudinal model (9) is defined to intuitively reflect the driving states of both vehicles in following mode. Here, e_v is the difference between the following vehicle speed v_{follow} and leading vehicle speed v_{lead} , and e_d is the difference between the actual distance d_{real} and safe following distance d_{safe} .

In vehicle lateral motion, the following vehicle needs to obtain lateral velocity and yaw rate based on its relative position to the trajectory, and adjust the steering angle to ensure effective lateral tracking, reducing lateral deviation and relative yaw angle [17].

2.3 Reward Function Design

In reinforcement learning, the reward is the environment's feedback to the agent's actions and a signal for evaluating action quality, typically a scalar. In joint lateral and longitudinal control of car-following, reward functions determine both lateral trajectory tracking and longitudinal speed control.

The car-following control problem can be transformed into a multi-objective optimization problem considering tracking performance, safety, and comfort [5]. To make the following vehicle approach the target path while maintaining better speed response and stable acceleration behavior, longitudinal speed error e_v , distance error e_d , lateral deviation e_y , acceleration a , and steering angle δ are used as reward function features. Additionally, penalties m are applied for abnormal conditions such as negative relative speed, excessively low following speed, and large lateral deviation. Therefore, reward function (11) is designed.

For effective tracking, reward function (12) is designed based on longitudinal speed error and lateral deviation to achieve both longitudinal speed tracking and lateral path following. Smaller following speed error and lateral deviation yield larger r_{follow} , with positive reward H applied when lateral deviation is less than 0.1m and following speed error is less than 1m/s for more precise path and speed tracking.

For comfort, reward function (13) is designed based on acceleration and steering angle. Smaller following acceleration and steering angle result in more stable lateral and longitudinal following and better comfort.

For safety, reward function (14) is designed based on safety distance error. Additionally, termination conditions are set as $|e_y| > 1$, $v < 0.5$, $d < 0$, with penalty m applied when these conditions are triggered to prevent excessive lateral deviation, excessively low following speed, and avoid collisions.

3 Experiments and Analysis

3.1 Model Training

The focus of DDPG-based joint lateral and longitudinal control for car-following is feature selection and fusion. Since using driver-perspective visual images as model input suffers from poor interpretability and may prevent the neural network from learning useful information, this model uses environmental feature vector X_{input} as model input.

The model outputs the following vehicle's acceleration a and steering angle δ according to the current policy and updates its position and speed. Simultaneously, the environment updates the predecessor's state and returns the reward for the current step and updates the policy.

Network training proceeds as follows: Design deep reinforcement learning elements for the car-following problem based on action space, state space, and reward function; initialize the Actor-Critic network and reset the environment; obtain observation s_0 from the environment and calculate initial action a_0 , then set it as the current action a ; apply action a to the environment to obtain next observation s' and reward r , then learn from the experience set; calculate next action a' to update the current action, and update the current observation with s' ; repeat this loop until termination conditions are met.

DDPG algorithm hyperparameters are shown in Table 1.

Table 1. Super parameters of DDPG algorithm

Parameter	Value
Actor network learning rate	0.0001
Critic network learning rate	0.001
Memory batch size	64
Experience replay storage pool	100000
Maximum episodes	5000
Maximum steps per episode	500

Figure 5 shows the reward variation trend during car-following model training. The training used data from 40 car-following events selected from the car-following dataset. The red curve represents the average reward per training episode, the blue curve shows the reward for each episode, and the yellow curve shows the Critic network's estimate of discounted long-term reward at the beginning of each episode. Higher average reward indicates better car-following performance. After 3548 episodes of training, the reward function gradually converged at approximately 1400 episodes, as clearly shown in Figure 5.

Figure 6 shows the reward values for the last 100 episodes of car-following model training, demonstrating that the algorithm is stable and effective.

3.2 Model Testing

A car-following control simulation system was built using MATLAB/Simulink to establish a complete vehicle dynamics model. Vehicle dynamics parameters are shown in Table 2.

Table 2. Vehicle dynamics parameters

Parameter	Value
Vehicle mass	1600 kg
Moment of inertia about z-axis	$2875 \text{ kg} \cdot \text{m}^2$
Distance from CG to front axle	1.2 m
Distance from CG to rear axle	1.3 m
Front tire cornering stiffness	19000 N/rad
Rear tire cornering stiffness	33000 N/rad

The car-following control strategy was tested and validated using the renowned NGSIM real driver car-following dataset. 1341 car-following events were extracted from vehicle trajectory data on the I-80 highway segment, each containing leading vehicle speed, following vehicle speed, relative speed, and inter-vehicle distance for durations over 15 seconds. Table 3 shows partial data from one car-following event.

Table 3. Car following data structure

Time (s)	Distance (m)	Following speed (m/s)	Relative speed (m/s)	Leading speed (m/s)
0	15.02	6.70	0.61	6.09
0.1	14.95	6.71	0.62	6.09

The following vehicle adjusts acceleration and steering angle to follow the predecessor along a road with changing curvature, calculating optimal car-following actions while satisfying constraints on safety distance, speed, acceleration, and steering angle. Based on vehicle dynamics physical limitations, the ranges for following vehicle acceleration a and steering angle δ are set as equation (16).

From the 1341 car-following pairs, one set of data was randomly selected to validate the proposed car-following decision scheme and compare it with the MPC car-following scheme. Changes in vehicle spacing, speed, and acceleration are shown in Figures 7-9. The initial distance between leading and following vehicles was set to 15 m, with following vehicle longitudinal speed at 6.7 m/s and leading vehicle speed at 6.1 m/s. Figure 7 shows the inter-vehicle distance, demonstrating that the distance remains relatively stable, with DDPG algorithm maintaining a smaller following distance than human drivers and MPC, achieving efficient car-following.

Figure 8 shows that the leading vehicle first accelerates, then decelerates, and finally cruises, while the DDPG algorithm produces more stable speed variations during following.

Figure 9 shows the following vehicle's acceleration curve, demonstrating that reasonable acceleration adjustments effectively modulate speed and spacing while maintaining relatively smooth acceleration.

To evaluate the algorithm's performance, Mean Absolute Error (MAE) from equation (17) is used as the evaluation metric, where y_i is an individual observation and \bar{y} is the arithmetic mean.

Table 4. Model error

Metric	Real driver	MPC	DDPG
Speed MAE (m/s)	0.52	0.41	0.38
Acceleration MAE (m/s ²)	0.31	0.28	0.21
Distance MAE (m)	0.85	0.79	0.73

Table 4 shows that DDPG algorithm achieves the smallest acceleration MAE, realizing smoother and more comfortable car-following. DDPG also reduces speed error compared to human drivers and MPC, demonstrating effective following with strong adaptive capability and more stable speed maintenance. The DDPG algorithm's distance MAE is smaller than real drivers, showing more stable decision-making with smaller headway.

Figure 10 shows the leading vehicle's trajectory curvature. Figure 11 presents lateral control results, where the following vehicle's initial lateral deviation was set to 0.2 m and initial yaw angle to -0.1 rad. By controlling the steering angle, the yaw angle error rapidly decreases. Due to continuously changing road curvature, the following vehicle constantly fine-tunes its steering angle to maintain small lateral deviation and relative yaw angle. DDPG and MPC algorithms show similar lateral control performance.

To evaluate prediction performance, acceleration change rate (jerk) is selected to assess following comfort, and time headway (thw) is chosen to evaluate safety and effectiveness. During car-following, thw typically remains within 1-4 s—smaller thw indicates tighter tracking and higher efficiency, but thw below 1 s risks collision, while thw above 4 s generally indicates non-following behavior.

Table 5. Car following model evaluation

Metric	Real driver	MPC	DDPG
jerk (m/s ³)	0.85	0.72	0.58
thw (s)	2.8	2.5	2.2

Table 5 shows that DDPG's jerk is smaller than human drivers, ensuring following comfort and avoiding discomfort from frequent acceleration. Additionally, DDPG maintains thw within the safe 1-4 s range, with smaller thw than human drivers and MPC, indicating higher following efficiency.

To validate performance under various conditions, another car-following event was randomly selected from the NGSIM dataset. Results in Figures 12-15 and

Tables 6-7 show that DDPG again produces smaller acceleration and jerk, more stable speed, and maintains thw within 1-4 s, achieving more comfortable, stable, and safe following.

Table 6. Model error (second event)

Metric	Real driver	MPC	DDPG
Speed MAE (m/s)	0.48	0.39	0.35
Acceleration MAE (m/s ²)	0.29	0.25	0.19
Distance MAE (m)	0.81	0.76	0.71

Table 7. Car following model evaluation (second event)

Metric	Real driver	MPC	DDPG
jerk (m/s ³)	0.79	0.68	0.53
thw (s)	2.7	2.4	2.1

4 Conclusion

This paper develops a vehicle car-following control model based on a three-degree-of-freedom vehicle dynamics model with joint lateral and longitudinal control. A decision model is then established using the deep reinforcement learning DDPG algorithm to determine the following vehicle's acceleration and steering angle, ensuring safe, effective, and comfortable following. The model is trained, tested, and evaluated using the human driving dataset NGSIM and compared with MPC car-following control. Results demonstrate that the proposed method achieves smaller following distance and acceleration change rate while ensuring safety, outperforming human drivers and holding significant potential for improving traffic safety and road capacity.

Currently, car-following control functions relatively independently. Integrating vehicle car-following control with lane keeping assist and lane change assist systems will enable higher-level autonomous driving control.

References

- [1] Saifuzzaman M, Zheng Z. Incorporating human-factors in car-following models: a review of recent developments and research needs [J]. *Transportation research part C: emerging technologies*, 2014, 48: 379-403.
- [2] Pipes L A. An operational analysis of traffic dynamics [J]. *Journal of applied physics*, 1953, 24 (3): 274-281.

[3] Zhang Lanfang, Zhu Peixuan, Yang Minhao, et al. Modeling of car-following behavior on urban underground expressways based on data-driven Methods [J]. Journal of Tongji University: Natural Science, 2021, 49 (05): 661-669.

[4] Luo Ying, Qin Wenhui. Combination low-speed car-following model based on IDM and RBFNN [J]. Application Research of Computers, 2019, 37 (8): 1-7.

[5] Zhu Meixin, Wang Yinhai, Pu Ziyuan, et al. Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving [J]. Transportation Research Part C: Emerging Technologies, 2020, 117: 102662.

[6] Pan Feng, Bao Hong. Preceding vehicle following algorithm with human driving characteristics [J]. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2021, 235 (7): 1868-1880.

[7] Zhu Bing, Jiang YuanDe, Zhao Jian, et al. A car-following control algorithm based on deep reinforcement learning [J]. China Journal of Highway and Transport, 2019, 32 (06): 53-60.

[8] Camacho E F, Alba C B. Model predictive control [M]. Springer science & business media, 2013.

[9] Hu Xiaosong, Zhang Xiaoqian, Tang Xiaolin, et al. Model predictive control of hybrid electric vehicles for fuel economy and emission reductions in car-following scenarios [J]. Energy, 2020, 196: 117101.

[10] Mao Jin, Yang Lei, Hu Yuanbo, et al. Research on Vehicle Adaptive Cruise Control Method Based on Fuzzy Model Predictive Control [J]. Machines, 2021, 9 (8): 160.

[11] Muhrer E, Vollrath M. The effect of visual and cognitive distraction on driver's anticipation in a simulated car following scenario [J]. Transportation research part F: traffic psychology and behaviour, 2011, 14 (6): 555-566.

[12] Xu Fang, Zhang Junming, Hu Yunfeng, et al. Lateral and longitudinal coupling real-time predictive controller for intelligent vehicle path tracking [J]. Journal of Jilin University: Engineering and Technology Edition, 2021, 51 (6): 2287-2294.

[13] Wang Hong, Huang Yanjun, Khajepour A, et al. Crash mitigation in motion planning for autonomous vehicles [J]. IEEE transactions on intelligent transportation systems, 2019, 20 (9): 3313-3323.

[14] Zhu Meixin, Wang Xuesong, Wang Yinhai. Human-like autonomous car-following model with deep reinforcement learning [J]. Transportation research part C: emerging technologies, 2018, 97: 348-368.

[15] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning [J]. arXiv preprint arXiv: 1509.02971, 2015.

[16] Puan O C, Mohamed A, Idham M K, et al. Drivers behaviour on expressways: headway and speed relationships [C]// IOP Conference Series: Materials

Science and Engineering. IOP Publishing, 2019, 527 (1): 012071.

[17] Wang Yulei, Ding Haitao, Yuan Jinxin, et al. Output-feedback triple-step coordinated control for path following of autonomous ground vehicles [J]. Mechanical Systems and Signal Processing, 2019, 116: 146-159.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv –Machine translation. Verify with original.