Proceedings of the 2nd International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/73/20240414

Car-following algorithm based on TD3 and CACC

Heng Lu

Metropolitan College, Boston University, Boston, MA

hengl@bu.edu

Abstract. For autonomous cars to drive safely and efficiently, they need to have a decent car following functionality. Many algorithms have been proposed to improve the car following functionality, but most of them focus at keeping similar speed instead of maintaining relative distances between vehicles, which makes these algorithms hard to deal with emergencies. Therefore, the paper proposed a hybrid algorithm based on twin delayed deep deterministic policy gradient (TD3) and cooperative adaptive cruise control (CACC) for car following task in autonomous driving. The algorithm is tested in an environment which simulates typical scenario in real world highway driving, and the performance is also measured when vehicle to vehicle (V2V) latency is also considered. The experiment shows that the algorithm keeps a decent dynamic relative distance between vehicles, which is crucial for collision avoidance in case of emergencies. It has a good performance and a great potential for real world applications.

Keywords: TD3, Deep Reinforcement Learning, Car Following, CACC

1. Introduction

Car following functionality is one of the most important aspects in autonomous driving. Car following algorithms with decent performance will reduce the levels of traffic congestion and the chance of traffic accident, thus saving consumer's time and being more environmentally friendly by reducing energy consumption [1]. Therefore, great effort has been made into designing a good car-following algorithm and researchers already have made some achievements, such as Adaptive Cruise Control and Cooperative Adaptive Cruise Control (CACC) [2]. However, despite their great power, their actual performance in real world is not satisfying due to the complexity of real traffic environment.

Recently, deep reinforcement learning has been widely adopted while developing autonomous vehicle algorithms [3]. Wei Xia et al. proposed a new control strategy using imitation learning based on deep Q learing [4]. Double deep Q network (DDQN) with an enhanced safety module is proposed for safe highway driving [5]. Yuan Lin et al. used deep deterministic policy gradient to address the degradation due to acceleration delay [6,7]. Then a hybrid structure based on CACC and Deep Deterministic Policy Gradient (DDPG) is proposed by Ruidong Yan [8].

However, most of these works only consider relative velocity and acceleration. It might be dangerous since vehicles might be too close and thus if one car stops suddenly due to collision, following cars have little space to avoid collision. Moreover, in real traffic situations, Vehicle to vehicle (V2V) communication is bound to have some latency which will introduce interesting dynamics. This paper introduces a hybrid car-following algorithm based on CACC and twin delayed DDPG (TD3) which takes the dynamic safe distance and V2V communication latency into consideration [9].

© 2024 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/73/20240414

This paper is structured as follows: background knowledge about TD3 and CACC is described in Section 2, the hybrid algorithm and environment setting are presented in Section 3, experiment details and analysis are described in Section 4 and conclusion is given in Section 5.

2. Background

2.1. Reinforcement Learning

Reinforcement learning agents learn optimal decisions by interacting with the environment in order to maximize the overall reward. At time t, RL agent is at state $s_t \in S$, and interacts with the environment with $a_t \in A$ according to its policy $\pi(a|s): S \to A$, receiving reward r_t and switching to new state s_{t+1}. Then the return, which is defined as the discounted sum of rewards, is given by $R_t = \sum_{i=t}^{T} \gamma^{i-t} r_t$, where $\gamma \in [0,1]$ is the discount factor weighting the importance of short-term rewards.

2.2. Deep Reinforcement Learning

Deep reinforcement learning algorithms refer to RL algorithms using deep neural network to estimate value function $V(s;\theta)$ and policy $\pi(a|s;\theta)$. The DRL algorithm used in the paper is called twin delayed deep deterministic policy gradient (TD3). It uses two critic networks Q_{θ_1}, Q_{θ_2}, an actor network π_ϕ and corresponding target networks $Q_{\theta_1'}$, $Q_{\theta_2'}$, $\pi_{\phi'}$. It updates critic networks using clipped double Q-learning and the targets are computed by

$$y' = r + \gamma min_{i=1,2} Q_{\theta_i'}(s', \pi(s')) \tag{1}$$

The actor network is updated by deterministic policy gradient:

$$\nabla_\phi J(\phi) = \frac{1}{m} \sum \nabla_a Q_{\theta_1}(s,a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s) \tag{2}$$

The target networks are updated by

$$\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i' \tag{3}$$

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi' \tag{4}$$

2.3. CACC

The CACC algorithm is based on differential equations and can handle highway situations and maintain a stable traffic flow, which the California PATH program has studied in depth[2]. The authors of paper [9] use a CACC and the acceleration of kth following car is given by

$$a_{k,CACC} = v_k + k_1 e_{x_{k,k-1}} + k_2 e_{v_{k,k-1}} + k_3 e_{x_{k,0}} + k_4 e_{v_{k,0}} \tag{5}$$

where $e_{x_{k,k-1}}$ and $e_{v_{k,k-1}}$ are respectively the relative distance and relative velocity between kth following vehicle and $(k-1)$th following vehicle, and $e_{x_{k,0}}$ and $e_{v_{k,0}}$ are the distance and velocity between kth following vehicle and the leading vehicle.

3. Method

3.1. Environment setting

It's crucial for autonomous vehicles to receive mutual information to decide their next actions. Predecessor-following information flow network is adopted [2], which is shown in Figure 1. In this network, vehicle sends its velocity, acceleration and travel distance during the car-following task to its successor.

DOI: 10.54254/2755-2721/73/20240414

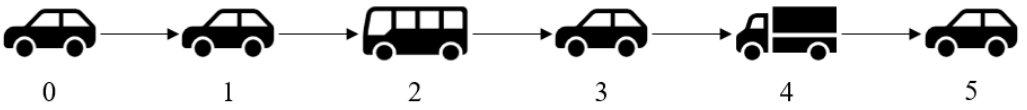

Figure 1. Vehicle to vehicle (V2V) communication topology. 0 represents the leading car, and the others represent following car.

3.2. TD3

For applying reinforcement learning algorithms, the MDP setting must be specified. First the state space is introduced. At time t, vehicle k receives velocity v'_{k-1}, acceleration a'_{k-1} and travel distance d'_{k-1} of vehicle $k-1$ sent at time $t'(t' \leq t)$ due to the V2V communication latency. The relative distance between kth and $(k-1)$th vehicles is given by

$$d_{k,k-1} = d'_{k-1} - d_k + d_0 \tag{6}$$

where d_0 is the initial distance between vehicles. The relative velocity and acceleration between kth and $(k-1)$th vehicles are given by

$$v_{k,k-1} = v'_{k-1} - v_k \tag{7}$$

$$a_{k,k-1} = a'_{k-1} - a_k \tag{8}$$

The velocity and acceleration should also be included into the state space. The state should consist of $[v_k, a_k, v_{k,k-1}, a_{k,k-1}, d_{k,k-1}]$.

The action space is very simple and consists of the acceleration a_k since all vehicles drive on the same lane and single acceleration suffices to determine state at next timestep.

The reward function is designed as follows. The performance of TD3 is largely dependent on the choice of reward functions. Speed of following vehicle should be close to its predecessor to make the vehicle platoon more stable. The distance between vehicles should be bounded near the dynamic safe distance to improve the safety and the efficiency. The vehicles should be encouraged to move forward as fast as they are allowed. Taking all of the above into consideration, the reward function is given by

$$r_k = -\frac{|v_{k,k-1}|}{v_{max}} - \frac{|d_{k,k-1} - d_{safe}|}{d_{safe}} + \frac{v_k}{v_{max}} + 1 \tag{9}$$

where v_{max} is the maximum velocity. The safe distance is defined as the minimum distance between vehicles to avoid collision when the front vehicle stops suddenly due to collision and the following vehicle brakes at its maximum acceleration a_{max} within response time τ. d_{safe} is given by

$$d_{safe} = v_k\tau - \frac{1}{2}a_{max}\tau^2 + d_{min} \tag{10}$$

where d_{min}=2m is added to avoid collision.

3.3. Hybrid structure

The hybrid structure is depicted in Figure 2. At each timestep, accelerations from CACC and TD3 are calculated independently. Acceleration from CACC is similar to paper [8] and is given by

$$a_{k,CACC} = v_k + k_1 d_{k,k-1} + k_2 v_{k,k-1} \tag{11}$$

Then hypothetical actions are taken to the environment and the environment in turn gives corresponding rewards. Finally, the action for next timestep is chosen according to equation 12 as follow:

DOI: 10.54254/2755-2721/73/20240414

$$a_k = \begin{cases} a_{k,CACC}, & r_{CACC} > r_{TD3} \\ a_{k,TD3}, & else \end{cases} \tag{12}$$

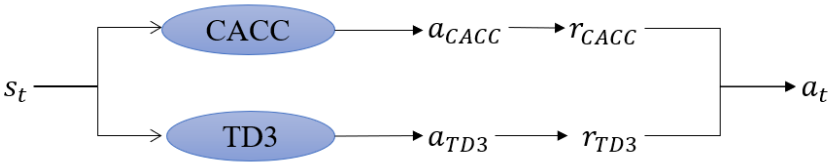

Figure 2. Algorithm Structure.

4. Experiments

The algorithm is implemented using PyTorch and runs on NVIDIA RTX 3060 laptop. In the experiment, 5 following cars learn to catch up the leading vehicle, which runs independently. The speed curve of leading vehicle is shown in Figure 3. The dynamics of leading vehicle is set to mimic common real-world vehicle dynamics, which starts at 0m/s, speeds up and keeps a velocity platform at 16.67m/s, 22.22m/s, 27.78m/s, 33.33m/s. After that, it gradually slows down to 0m/s again. The simulation is based on discrete timestep and each environment step is equivalent to 10ms.

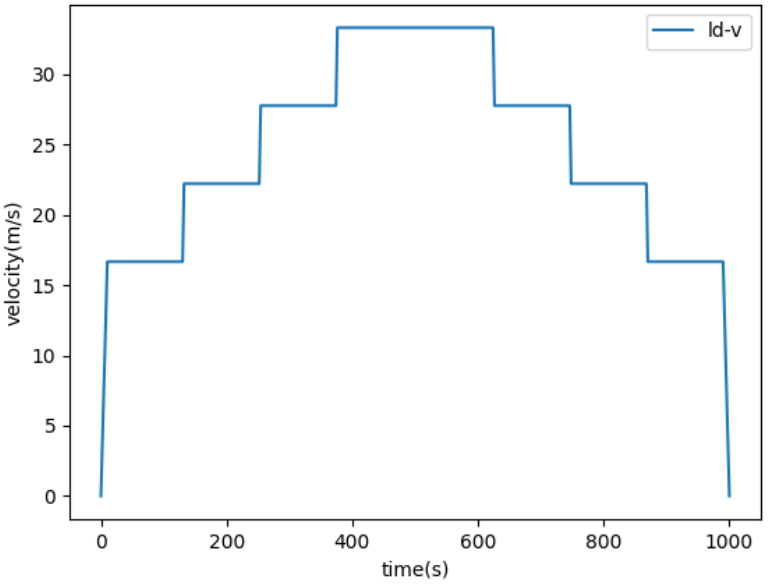

Figure 3. The speed curve of leading vehicle

The relative distance between following vehicles and the leading vehicle when the V2V communication is latency-free is shown in Figure 4. The relative distance between vehicles dynamically changes with speed and always keeps a safe distance between vehicles.

DOI: 10.54254/2755-2721/73/20240414

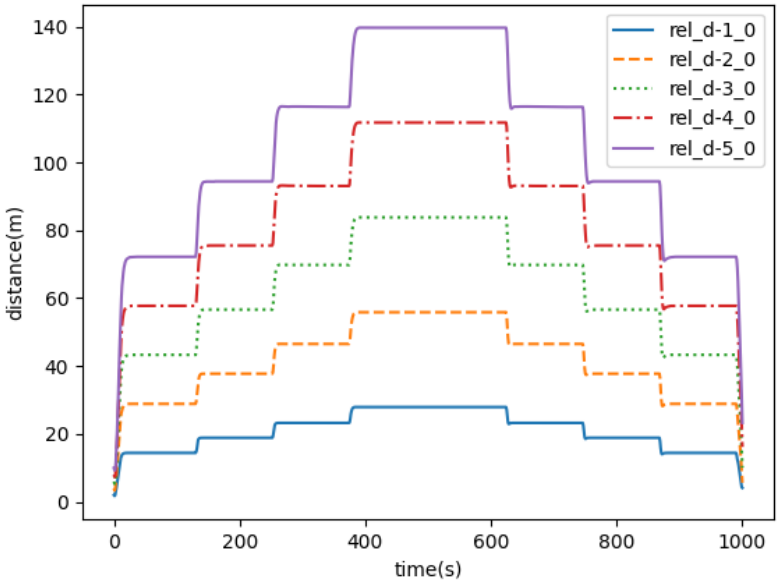

Figure 4. Relative distance between the leading vehicle and the ith following vehicle. The latency of V2V communication is zero.

The relative distance between following vehicles and the leading vehicle when the V2V communication has a fixed 10ms latency is shown in Figure 5. The overall relative distances between each following vehicle and the leading vehicle are shown in Figure 6. The relative distance between vehicles is not constant when the leading vehicle has a constant speed. The vibrations of following vehicles that are far away from the leading vehicle are more noticeable since the latency between the leading vehicle and the vehicle itself tends to accumulate. When the leading vehicle runs at 33.33m/s, there's no noticeable vibrations due to latency since the top speed of all vehicles is limited at 33.33m/s.

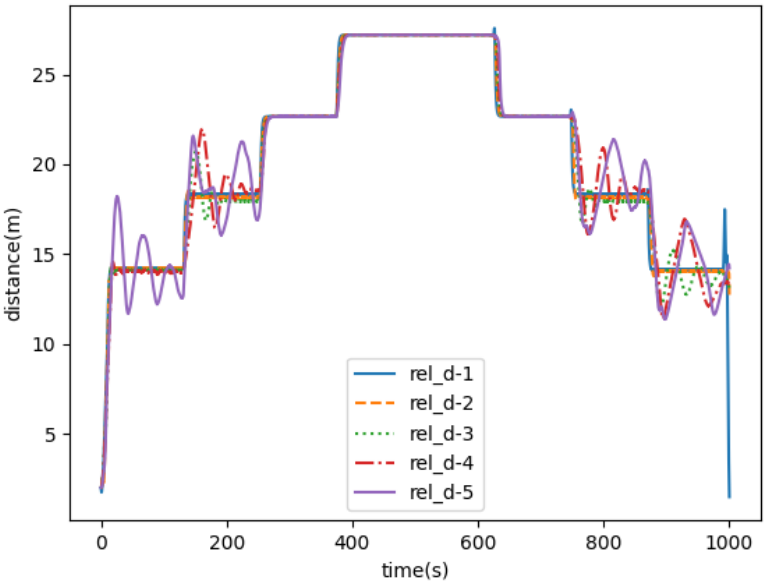

Figure 5. The curve of relative distance between ith vehicle and (i-1)th vehicle. 0th vehicle is the leading vehicle. The latency of V2V communication is 10ms.

DOI: 10.54254/2755-2721/73/20240414

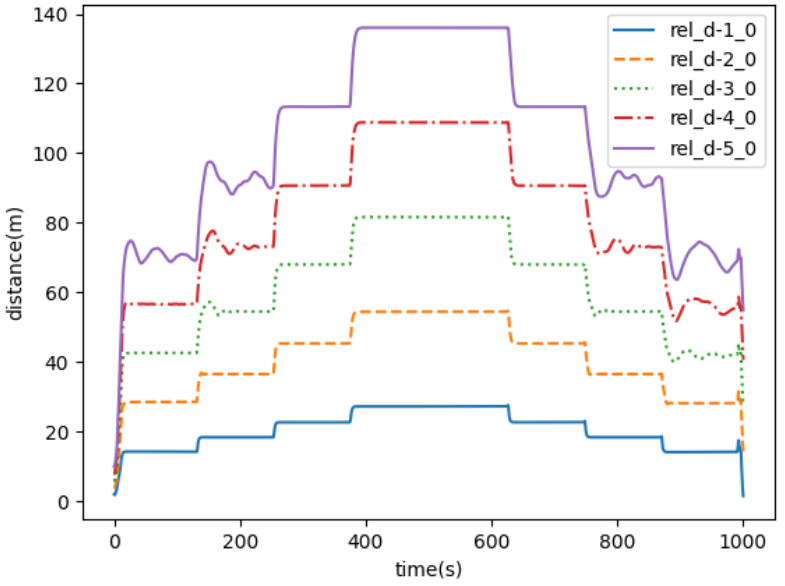

Figure 6. The curve of relative distance between the leading vehicle and the ith following vehicle. The latency of V2V communication is 10ms.

5. Conclusion

The paper uses a hybrid algorithm based on TD3 and CACC for car following task in autonomous driving. The CACC is used only at training phase to provide instructions and ensures basic car-following functionality. Meanwhile, the exploration ability of TD3 enables the algorithm to fully explore the complex state space of car-following environment and keep a dynamic distance while driving under the guidance of the reward function. The algorithm improves the reliability of car following functionality by considering the dynamic safe distance between vehicles, which allows vehicles to stop before collision.

References

[1] R. Yan, D. Yang, B. Wijaya, et al., Feedforward Compensation-Based Finite-Time Traffic Flow Controller for Intelligent Connected Vehicle Subject to Sudden Velocity Changes of Leading Vehicle, IEEE Transactions on Intelligent Transportation Systems, Aug. 2020, vol. 21, no. 8, pp. 3357-3365.

[2] S. E. Li, Y. Zheng, K. Li, et al., Dynamical Modeling and Distributed Control of Connected and Automated Vehicles: Challenges and Opportunities, IEEE Intelligent Transportation Systems Magazine, Fall 2017, vol. 9, no. 3, pp. 46-58.

[3] S. Aradi, Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles, IEEE Transactions on Intelligent Transportation Systems, Feb. 2022, vol. 23, no. 2, pp. 740-759.

[4] W. Xia, H. Li and B. Li, A Control Strategy of Autonomous Vehicles Based on Deep Reinforcement Learning, 9th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2016, pp. 198-201.

[5] A. Baheri, S. Nageshrao, H. E. Tseng, et al., Deep reinforcement learning with enhanced safety for autonomous highway driving. IEEE Intelligent Vehicles Symposium (IV). Las Vegas, USA, 2020, pp. 1550-1555.

DOI: 10.54254/2755-2721/73/20240414

[6] Y. Lin, J. McPhee and N. L. Azad, Longitudinal Dynamic versus Kinematic Models for Car-Following Control Using Deep Reinforcement Learning, IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 2019, pp. 1504-1510.

[7] T. P. Lillicrap, J. J. Hunt, A. Pritzel, et al., Continuous control with deep reinforcement learning. 4th International Conference on Learning Representations, ICLR, San Juan, Puerto Rico, May 2-4, 2016.

[8] R. Yan, R. Jiang, B. Jia, J. Huang and D. Yang, Hybrid Car-Following Strategy Based on Deep Deterministic Policy Gradient and Cooperative Adaptive Cruise Control, IEEE Transactions on Automation Science and Engineering, vol. 19, no. 4, pp. 2816-2824, Oct. 2022.

[9] S. Fujimoto, H. Hoof and D. Meger, Addressing function approximation error in actor-critic methods. International conference on machine learning, 2018, July, pp. 1587-1596.