Reinforcement learning-based collision avoidance of a connected and automated vehicle at merging roads

Minseok Seo
Artificial Intelligence Graduate School
Gwangju Institute of Science and Technology
Gwangju, Republic of Korea
seominseok@gm.gist.ac.kr

Seongjae Shin
School of Mechanical Engineering
Gwangju Institute of Science and Technology
Gwangju, Republic of Korea
ssjdooly@gm.gist.ac.kr

Kyungjoong Kim
School of Integrated Technology
Gwangju Institute of Science and Technology
Gwangju, Republic of Korea
kjkim@gist.ac.kr

Kyunghwan Choi
School of Mechanical Engineering
Gwangju Institute of Science and Technology
Gwangju, Republic of Korea
khchoi@gist.ac.kr

2023 International Workshop on Intelligent Systems (IWIS) | 979-8-3503-0504-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/IWIS58789.2023.10284616

***Abstract*—Although autonomous driving technology has made significant advancements, autonomous vehicles tend to be defensive in complex environments such as merging roads, intersections, and roundabouts. Connected autonomous vehicles (CAVs) are expected to behave more naturally even in such complex environments leveraging information obtained through vehicle connectivity, such as neighboring vehicles' position and velocity. This study presents a reinforcement learning-based framework for collision avoidance of a CAV at merging roads. The key idea is to redesign the reward function of previous relevant work so that collision avoidance is realized more in a human-like and predictive manner.**

***Index Terms*—Connected automated vehicle (CAV), collision avoidance, reinforcement learning, merging roads**

I. Introduction

In recent years, significant advancements in autonomous driving technology have enabled the pilot operations of vehicles from various autonomous driving companies such as Tesla, Waymo, and General Motors in some areas of the United States [1]. Although autonomous driving technology has achieved significant accomplishments, numerous challenges are still being researched. One of them is interaction with vehicles on the road, especially at a transition period like now, the interaction between human-driven vehicles (HDV) and autonomous vehicles is very important[2]. Furthermore, Autonomous vehicles tend to be quite defensive when encountering complex interaction scenarios (intersections, roundabouts, etc.), which can lead to traffic jams or rear-end collisions.

Stability can be ensured with model predictive control (MPC), which uses system models to predict the system's future behavior based on the current state and environmental formation [3]. However, MPC is computationally complex and time-consuming, making it challenging to use in real-time applications. It also uses mathematical models of the vehicle and the traffic environment. Still, the problem is that in real life, the models do not match exactly, and it can lead to prediction errors.

Reinforcement learning is a learning method in which an agent interacts with an environment, learning optimal actions while receiving rewards and penalties. It uses a value function (or action-value function) to determine which action to take in a given state. It can learn from experience by interacting with the environment, reducing the problem of prediction error due to differences between the model and the actual situation. Also, by learning in various situations, it can learn the optimal action[4].

However, since many variables exist in the actual driving environment, a connected automated vehicle (CAV) is used to solve this problem. CAV uses communication technology to exchange information with surrounding vehicles and infrastructures. Therefore, CAV is emerging as an essential solution for autonomous driving technology as it can increase safety[5].

The merging section on a highway, one of the complex interaction scenarios described earlier, has been the subject of extensive research due to its susceptibility to congestion caused by its bottleneck structure. In a previous study [6], multi-agents were applied to CAVs in a merging section of a highway with one lane as follows, and they were made to drive cooperatively while maintaining a certain distance from each other. However, the reward function was designed to be adjusted only within the merging interval, as shown in Fig. 1, resulting in insufficient temporal flexibility for vehicle coordination.

This study presents a reinforcement learning-based control strategy for a CAV to pass through merging roads while avoiding collision with other HDVs and ensuring smooth traffic flow. The key idea is to redesign the reward function from the projected distance, used in [6], to the Euclidean distance so that collision avoidance is realized more efficiently and in a predictive manner. The deep Q-network (DQN) algorithm is adopted. The remainder of this paper is organized as follows. Section II explains DQN and reward function. Section III provides detailed information about the simulation

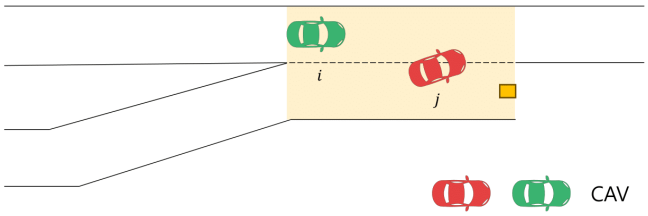

Fig. 1. Coordination scenario in [6]

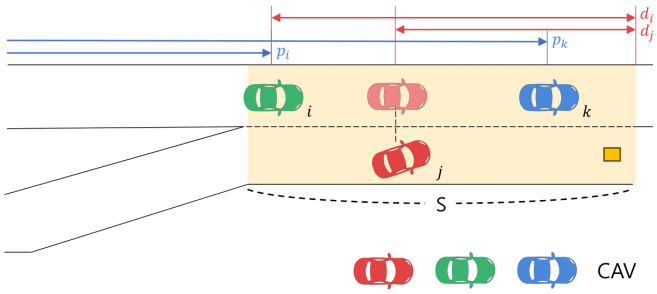

Fig. 2. Distance metric in [6] (CAV *k* is in front of CAV *i* on the same road, while CAV *i* is in front of CAV *j* on different roads)

environment, DQN network structure, hyperparameters, and training. Validation results are provided in Section IV. Finally, Section V concludes the paper with an outlook on future works.

II. Problem Formulation

A. Deep Q-Network

The state-action value function was emulated in traditional reinforcement learning using a tabular approach. However, real-world problems often have extensive state and action spaces, which impose limitations on such methods. As a solution to these problems, Deep Reinforcement Learning (DRL) emerged, combining deep learning with reinforcement learning. It gained attention starting in 2013 when DeepMind applied Deep Q-Network (DQN) [7] to the Breakout game by Atari. DeepMind used deep learning to approximate the Q function as a non-linear function with a Q table size of 34,406,400 (calculated by multiplying the pixel size and the number of possible values per pixel by the number of possible actions).

The core ideas of DQN are Experience Replay and Target Network. In deep learning, learning is performed assuming that learning data is independently extracted, but since reinforcement learning uses samples extracted while interacting with the environment, dependencies exist between successive samples. Correlation among consecutive samples increases the instability of learning. Furthermore, when learning on-policy and updating the Q function, the behavior policy changes due to the update, resulting in a change in the distribution of the generated training data. DeepMind introduced a buffer called Replay Memory to address this learning instability, which stores generated sample data and delay its usage for later, thereby improving learning stability. Furthermore, when the parameter theta of the Q network is updated, it causes both the desired update value and the reference value to change simultaneously, leading to instability in learning. To solve this problem, in DQN, the target network is fixed and updated for step C.

B. Reward Function

The reward function used in the previous research [6] is the same as (1):

$$r_t = w_1 \cdot r_{speed} + w_2 \cdot r_{rear} + w_3 \cdot r_{lateral} \tag{1}$$

Each term of the reward in (1) corresponds to the following: the speed term is the same as (5), and the terms related to rear and lateral are (2) and (3), respectively. Equations (2) and (3) impose penalties only when the conditions $0 < p_k - p_i < d_{safe}$ and $0 < d_i - d_j < d_{safe}$ and $d_i < S, d_j < S$ are satisfied, respectively.

$$r_{rear} = -\frac{1}{p_k - p_i} \tag{2}$$

$$r_{lateral} = -\frac{1}{d_i - d_j} \tag{3}$$

In this paper, the reward function was redesigned so that the CAV could adjust the distance to other vehicles in all road areas. The reward calculated at each time step follows the equation in (4).

$$r_t = w_1 \cdot r_{collision} + w_2 \cdot r_{speed} + w_3 \cdot r_{rear} + w_4 \cdot r_{lateral} \tag{4}$$

The weights for each reward are as follows: $w_1 = 50$, $w_2 = 1$, $w_3 = 20$, $w_4 = 5$. To emphasize the significance of collision avoidance, greater weights were assigned to the penalties for collisions and collisions with preceding vehicles. Each term of the reward function is defined as shown in (5), (6), and (7).

- ***Speed reward.*** In the Highway Env [8] used for simulation construction, the speed range of vehicles can be deterministically fixed, which means that not adhering to the speed limit does not result in any fines being imposed.

$$r^{speed} = \frac{v_{max} - \sqrt{(v - v_{max})^2}}{v_{max}} \tag{5}$$

- ***Collision reward.*** To make the agent aware of avoiding collisions, a boolean function was added that returns -1 if a vehicle collides and 0 otherwise.

$$r_{collision} = \begin{cases} -1, & \text{if collision} \\ 0, & \text{if no collision} \end{cases} \tag{6}$$

- ***Rear & Lateral reward.*** In the previous studies, the distance between vehicles on the merging road and the main road was calculated using the projected distance, which need to sufficiently reflect the distance difference between the main road and the merging road. Additionally, the

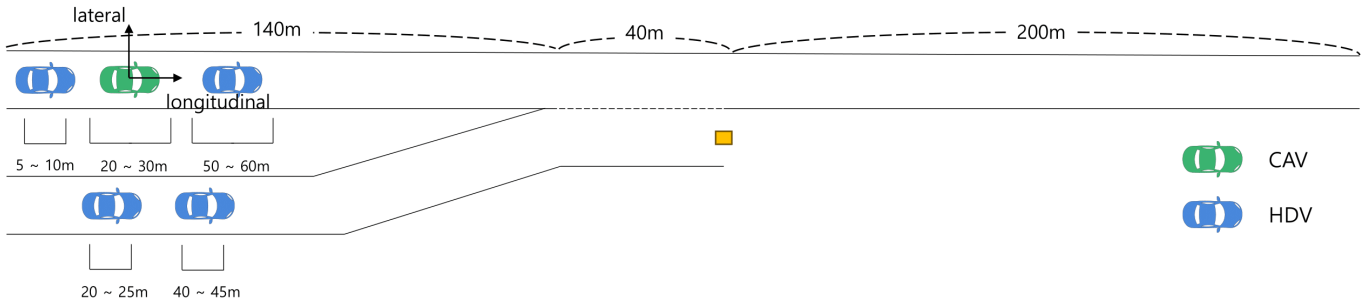

Fig. 3. The merging scenario used in this paper. The number range under the vehicle represents the range of positions where the vehicles are generated. The green vehicle represents the CAV and the blue vehicles represent HDVs.

inter-vehicle distance adjustment was designed to be applied only in the merging section. The projection distance was replaced with the Euclidean distance to address these issues, and the distance calculation was extended to all road segments. Furthermore, when calculating the distance to other vehicles on different roads, the leading vehicle changes based on the speed of the vehicles. To account for this, the closest leading vehicle is tracked and considered during the calculation. The safety distance between vehicles is set to 10 m, and if the distance between vehicles is less than or equal to 10 m, penalties are imposed for rear/lateral proximity. When the distance between vehicles reaches 5 m, a collision occurs, and the vehicles receive a collision penalty, resulting in the termination of the episode. Equation (7) imposes penalties only if the distance satisfies the condition distance $< d_{safe}$.

$$r_{rear/lateral} = -\frac{1}{\text{euclidean distance}} \tag{7}$$

III. Simulation Setup

In the previous study [6], control of CAVs was carried out using the DDPG algorithm in a multi-agent setting. However, this paper employed a single-agent approach to control a single CAV, distinguishing it from the previous research.

A. Simulation Environment

This paper considers a scenario where HDVs and a CAV coexist and drive together, as shown in Fig. 3. Each vehicle was randomly generated within the specified range beneath the vehicles, and the initial velocity was also randomly set between 5 to 15 m/s. An episode would terminate if a collision occurred or the ego vehicle passed a distance of 370 m. Road environment and vehicle models were implemented using Highway Env, a traffic simulation framework and HDV were implemented using Intelligent Driver Model (IDM) [9] and Minimizing Overall Braking Induced by Lane change (MOBIL) [10] models. Assume that each vehicle is connected by vehicle-to-vehicle (V2V) communication (communication errors were not considered). The ego vehicle controlled by reinforcement learning receives observations of up to three obstacles or vehicles in its vicinity, including their longitudinal coordinate, lateral coordinate, longitudinal velocity, and lateral velocity. It can perform five actions (Lane left, Idle, Lane right, faster, slower) through acceleration and steering control. The speed range for each vehicle is limited to 5-15 m/s.

B. DQN Network Architecture & Hyperparameter

The DQN model was implemented and trained using Stable Baselines [11], and the network architecture and hyperparameters are shown in Fig 4 and Table I.

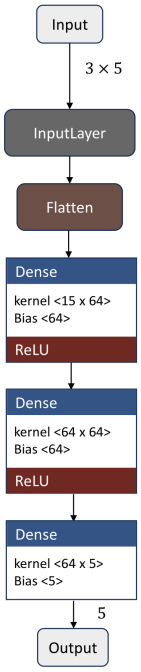

Fig. 4. DQN Network architecture

C. Training

In this paper, DQN was implemented using Stable Baselines3. Stable Baselines is a reinforcement learning library based on the OpenAI Gym [12] environment. In Gym, episodes are designed to be terminated by the environment based on a time-based criterion, providing flexibility and compatibility across various problem domains. Since Stable

TABLE I
DQN HYPERPARAMETERS

Hyperparameter	Value
learning rate	0.005
buffer size	15000
learning starts	200
batch size	32
gamma	0.8
train frequency	1
exploration fraction	0.01
target update interval	50

Baselines implements algorithms based on the Gym environment, it does not provide a direct way to specify the number of episodes during model training. Instead, it allows you to specify the total number of timesteps for training. Therefore, this model was trained over 35,000 steps.

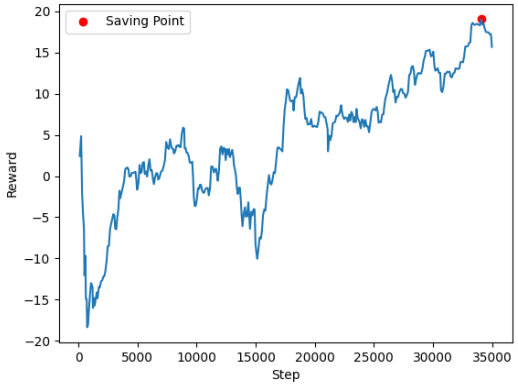

Fig. 5. Training curves for 35,000 steps

The model that achieved the highest reward during training was saved and used in the simulation results of Section IV.

IV. SIMULATION RESULTS

To evaluate the algorithm developed in this paper, three metrics were used: collision rate, average velocity, and average distance between vehicles.

- ***Collision Rate.*** The collision rate of vehicles.
- ***Average Velocity.*** The average speed of vehicles (measured only when there is no collision).
- ***Average Distance.*** The average Euclidean distance between vehicles and the nearest vehicle (measured only when there is no collision).

TABLE II
PERFORMANCE MEASUREMENT RESULTS FOR THREE METRICS

Metrics	Our method	Previous research	Projection distance
Collision rate	1.76%	28.7%	3.94%
Average velocity (*m/s*)	12.74	13.16	12.97
Average distance (*m*)	14.90	19.34	19.09

Three methods were tested over 10,000 episodes: 1) the proposed method, 2) the method presented in [6], which adopts (2) and (3) for the rear & lateral reward and includes the reward only in the yellow area shown in Fig. 1, and 3) the method same as the second one except that it includes (2) and (3) for all area. The models used for evaluation utilized the saved networks obtained when achieving the highest rewards during training.

The experimental results demonstrated that the method proposed in this paper maintains appropriate safety distance and speed while driving compared to the other two methods.

V. CONCLUSIONS

This study proposed a reinforcement learning-based framework to enable a CAV to merge onto highways while maintaining a safe distance and avoiding collisions with HDVs. By changing the method of measuring the distance between vehicles and assigning a greater weight to the reward function for collisions, it was confirmed that all vehicles on the road could pass through safely without collisions. In future research, a robust reward function that is not sensitive to these hyperparameters will be designed. Additionally, the current training and testing of the agent have focused only on highway merging scenarios. Still, in subsequent studies, efforts will be made to enable safe driving in general road environments, including intersections and roundabouts.

REFERENCES

[1] S. Singh and B. S. Saini, "Autonomous cars: Recent developments, challenges, and possible solutions," in *IOP Conference Series: Materials Science and Engineering*, vol. 1022, p. 012028, IOP Publishing, 2021.

[2] D. Chen, M. R. Hajidavalloo, Z. Li, K. Chen, Y. Wang, L. Jiang, and Y. Wang, "Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–16, 2023.

[3] V. Bhattacharyya and A. Vahidi, "Automated vehicle highway merging: Motion planning via adaptive interactive mixed-integer mpc," in *2023 American Control Conference (ACC)*, pp. 1141–1146, IEEE, 2023.

[4] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[5] A. Papadoulis, M. Quddus, and M. Imprialou, "Evaluating the safety impact of connected and autonomous vehicles on motorways," *Accident Analysis & Prevention*, vol. 124, pp. 12–22, 2019.

[6] S. K. S. Nakka, B. Chalaki, and A. A. Malikopoulos, "A multi-agent deep reinforcement learning coordination framework for connected and automated vehicles at merging roadways," in *2022 American Control Conference (ACC)*, pp. 3297–3302, IEEE, 2022.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing

atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[8] E. Leurent *et al.*, "An environment for autonomous driving decision-making," 2018.

[9] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, p. 1805, 2000.

[10] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model mobil for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.

[11] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann, "Stable baselines3," 2019.

[12] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.