2022 American Control Conference (ACC)
Atlanta, USA, June 8-10, 2022

A Multi-Agent Deep Reinforcement Learning Coordination Framework for Connected and Automated Vehicles at Merging Roadways

Sai Krishna Sumanth Nakka, *IEEE Student Member*, Behdad Chalaki, *IEEE Student Member*,
Andreas A. Malikopoulos, *IEEE Senior Member*

***Abstract*— The steady increase in the number of vehicles operating on the highways continues to exacerbate congestion, accidents, energy consumption, and greenhouse gas emissions. Emerging mobility systems, e.g., connected and automated vehicles (CAVs), have the potential to directly address these issues and improve transportation network efficiency and safety. In this paper, we consider a highway merging scenario and propose a framework for coordinating CAVs such that stop-and-go driving is eliminated. We use a decentralized form of the actor-critic approach to deep reinforcement learning—multi-agent deep deterministic policy gradient. We demonstrate the coordination of CAVs through numerical simulations and show that a smooth traffic flow is achieved by eliminating stop-and-go driving.**

I. INTRODUCTION

THE disproportionate growth in traffic volume compared to the road capacity causes an increase in congestion on highways with significant implications on the road safety and environmental footprint of road vehicles [1]. Traffic bottlenecks usually cause congestion; for instance, recurrent bottlenecks such as merging on highways are responsible for around 40-80% of congestion in the US [2]. In terms of safety, 1.7% of the total number of fatal crashes are caused by lane change/merging maneuvers [3]. Data collected by NHTSA [4] shows that a major portion of the accidents is caused due to human error. By combining communication technologies with the capabilities of automated vehicles, we can improve both safety and efficiency [5]. Connected and automated vehicles (CAVs) are able to interact with each other to gather information and make decisions to reduce the potential conflicts by $90-94\%$ [6]. In addition to increasing road safety, CAVs can enable significant improvements in traffic efficiency, energy consumption, and ultimately reduce the carbon footprint of the automotive industry [7], [8].

The problem of safely coordinating vehicles through a merging roadway was originally addressed by the influential work of Levine and Athans [9] in which they formulated the problem as a linear quadratic regulator to minimize the speed errors that determine the distance between the merging vehicles. Following that, there have been numerous studies that tackled the problem of coordinating CAVs in different traffic scenarios, including highway merging, using techniques of classical control [10]–[14]. The efficacy of some of these techniques has been demonstrated through practical implementations [15]–[17]. A thorough review of the state-of-the-art methods and challenges in coordination of CAVs is provided in [18], [19].

Since deriving analytical solutions for complex transportation applications is not practical, some studies used reinforcement learning techniques (RL) like Q-learning [20] for different traffic scenarios [21]–[27]. However, in large problems with many state-action pairs, to avoid Bellman's "curse of dimensionality," deep reinforcement learning methods (DRL), such as Deep Q-network (DQN) [28], are used where the Q-function is replaced with a deep neural network. There have been a few studies in the literature that have applied DRL techniques to the problem of highway merging. Wang and Chan [29] presented a DRL formulation for on-ramp merging of an autonomous vehicle using DQN. The same problem of freeway merging was addressed by Nishi *et al.* [30] using a combination of multi-policy decision making for choosing the possible spots to merge into and passive actor-critic method to learn the state value for choosing the policy to merge into the best spot. Nassef *et al.* [31] used a centralized trajectory recommendation framework for coordinating CAVs in a lane merging scenario. Ren *et al.* [32] addressed the CAV merging problem in the context of a lane drop caused by a highway work zone using a soft actor-critic algorithm where only the vehicles in the merging lane were considered as RL agents, while the vehicles on the main lane were controlled by a modified VISSIM driver model. A comprehensive review of DRL methods applied to transportation research can be found in [33]

The contribution of this paper is the development of a decentralized, multi-agent framework for safely coordinating CAVs in a highway on-ramp merging scenario while ensuring safety and smooth traffic flow. The solution presented in this paper has the following benefits compared to other studies addressing coordination of CAVs in a merging scenario. First, this study utilizes the added benefits of connectivity by controlling a network of CAVs rather than navigating an autonomous ego-vehicle through an environment of other vehicles [29], [30] to safely merge into one lane. Second, we consider the CAVs as multiple cooperative learning agents as opposed to a single agent as shown in [34]. In contrast to the application of DRL for merging of CAVs in [32], this paper explicitly considers rewarding safe, high-speed travel to incentivize smoother traffic flow. Moreover, this paper considers all CAVs in the network to be RL agents and hence are required to learn to safely cooperate with other CAVs.

This research was supported by ARPAE's NEXTCAR program under the award number DE-AR0000796.

The authors are with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716 USA (emails: {nakkash;bchalaki;andreas}@udel.edu).

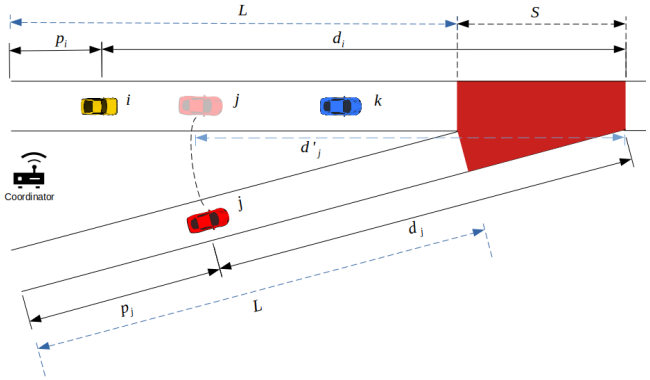


Fig. 1: Merging scenario with a coordinator communicating with CAVs inside the control zone.

The layout of this paper is as follows. In Section II, we provide our problem formulation, including the details of the adopted DRL method. In Section III, we present the simulation parameters, the type of neural networks used and how they are trained, and in Section IV, we provide simulation results. In Section V, we draw concluding remarks and discuss potential directions for future research.

II. PROBLEM FORMULATION

In this paper, we consider the problem of coordinating a network of CAVs in a scenario of highway on-ramp merging (Fig. 1). The problem setup consists of one group of CAVs traveling from the arterial road merging into another group of CAVs that are traveling on the main road of the highway via an on-ramp. To facilitate connectivity, in our problem, we consider the presence of a *coordinator* in the network which shares information among all the CAVs without participating in any decision-making process, and also stores information specific to the environment, e.g., the physical parameters of the scenario being considered. Each CAV can share information with other CAVs and the coordinator as long as they are in a predefined area of length $L \in \mathbb{R}_+$ (Fig. 1) called the *control zone*. The area near the end of the control zone, where the CAVs merge, is considered as the *merging zone* and its length is denoted by $S \in \mathbb{R}_+$ (Fig. 1). For the sake of simplicity, we assume that the highway and on-ramp are single-lane roads.

We formulate the problem as a multi-agent Markov decision process (MMDP) consisting of $N \in \mathbb{N}$ CAVs with $\mathcal{N} = \{1, 2, \cdots, N\}$ representing the set of all CAVs in the network. The problem is defined by a combination of a set of states $\mathcal{S}$: $\mathcal{S}_1 \times \cdots \times \mathcal{S}_N$ of each CAV i, $i \in \mathcal{N}$, in the environment; a combination of set of actions $\mathcal{U}$: $\mathcal{U}_1 \times \cdots \times \mathcal{U}_N$, where $\mathcal{U}_i$ is the set of feasible actions to CAV i; and deterministic policies $\mu_i : \mathcal{S}_i \to \mathcal{U}_i$. The transition between states is governed by the state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$. The reward function of the entire network is denoted by $\mathcal{R} : \mathcal{S} \times \mathcal{U} \times \mathcal{S} \to \mathbb{R}$, while $\mathcal{R}_i : \mathcal{S} \times \mathcal{U}_i \times \mathcal{S} \to \mathbb{R}$ is the reward function of CAV i.

A. Modeling Framework

For simulation purposes, each CAV $i \in \mathcal{N}$ is assumed to be governed by double-integrator dynamics.

$$\dot{p}_i(t) = v_i(t), \tag{1}$$
$$\dot{v}_i(t) = u_i(t), \tag{2}$$

where $p_i \in \mathcal{P}_i$, $v_i \in \mathcal{V}_i$, $u_i \in \mathcal{U}_i$ represent the position, speed, and acceleration/deceleration of CAV i at time $t \in \mathbb{R}_+$. Note that our framework does not require using a specific dynamics model, and hence enables us to utilize even high-fidelity dynamics models from traffic simulators, which is the focus of ongoing work. In the context of our problem, we consider that each CAV i is represented by its state $s_i := [\mathbf{x}_i, \mathbf{x}_k, \mathbf{x}_j]^\top$, where $s_i \in \mathcal{S}_i$; $\mathbf{x}_i := [p_i, v_i]^\top$ consists of local information of CAV i including position p_i and speed v_i; k denotes the CAV that is immediately in front of CAV i on the same road as i, while j denotes the vehicle that is ahead of i on the other road. For example, in Fig. 1, if we consider the yellow vehicle to be CAV i, then k will be the blue CAV traveling on the same road as i and j will be the red CAV traveling on the merging road. Additionally, let d_ℓ be the distance of any CAV $\ell \in \mathcal{N}$ from the end of the merging zone. For the CAVs traveling on the on-ramp, we project d_ℓ onto the main road, and use the projected distance to merging denoted by d'_ℓ instead. The action/control input $u_i \in [u_{\min}, u_{\max}]$ is bounded by the maximum, $u_{\max}$, and minimum, $u_{\min}$, acceleration limits of each CAV. In this study, for simplicity, we consider a homogeneous type of vehicles to represent the CAVs. Thus, $u_{\max}$ and $u_{\min}$ bounds do not vary among CAVs.

B. Deep Reinforcement Learning Methodology

Deep deterministic policy gradient (DDPG) [35] is an actor-critic RL method, which is used to learn a state-action value function (critic) and a continuous, deterministic policy function (actor), where each function is represented by a neural network. Using this architecture, consider the multi-agent scenario in which each CAV i is an independent learner with its own critic and takes action independent from other CAVs. Let $s_i \in \mathcal{S}_i$ and $u_i \in \mathcal{U}_i$ represent the state and action of the CAV i at the current time respectively; while $r_i \in \mathcal{R}_i$ is the reward that CAV i receives at the current time for taking an action u_i and transitioning from s_i to $s'_i \in \mathcal{S}_i$. At every time step, for each CAV i the tuple $D_i = \{s_i, u_i, r_i, s'_i\}$ is stored as experience buffer which is then used for training the neural networks. For each CAV, we define $Q_i^\phi(s_i, u_i)$ as the state-action value function and $\mu_i^\theta(s_i)$ as the policy function, where $\phi \in [0, 1]$ and $\theta \in [0, 1]$ represent the corresponding neural network parameters. The loss function, $\mathcal{L}_i(\phi)$, that needs to be minimized is given by Eq. (3) as the error between approximated state-action value function and the corresponding Bellman equation. Finally, the policy function in Eq. (5) is learned by maximizing the

estimated optimal state-action value function.

$$\mathcal{L}_i(\phi) = \mathop{\mathbb{E}}_{s,\mu,r,s'}\left[\left(Q_i^{\phi}(s_i, \mu_i^{\theta}(s_i)) - y_i\right)^2\right], \quad (3)$$

where,

$$y_i = r_i + \gamma Q_i^{\phi_t}\left(s_i', \mu_i^{\theta_t}(s_i')\right), \quad (4)$$

$$\mu_i^{\theta}(s_i) = \arg\max_{\theta} \mathop{\mathbb{E}}_{s}\left[Q_i^{\phi}(s_i, \mu_i^{\theta}(s_i))\right], \quad (5)$$

Here $\phi_t \in [0,1]$ and $\theta_t \in [0,1]$ are target critic and target actor network parameters respectively. In DRL, target networks are used to improve stability in training. Both the target state-action value network and the target policy network are similar to their respective original networks but are updated differently which is described in Section III-B. Note that the parameters of the four neural networks are specific to each CAV and hence should also be indexed by i, but this has been omitted for the sake of cleaner notation.

There is one key drawback to considering all the CAVs as independent learners. As the CAVs learn their state-action value functions independently, their policies keep changing in the training procedure. This creates a non-stationary environment from the perspective of any CAV which violates Markovian assumptions that are necessary for convergence. This formulation also degrades the gradient estimates of the state-action value function that are required for maximizing the function. To circumvent the aforementioned issues, Lowe *et al.* [36] proposed a modification to the traditional actor-critic algorithm by considering a centralized critic and decentralized actors. The critic of each CAV is supplied with the extra information from all other CAVs to form a centralized critic. This additional information, which is only provided during the training process, includes all the actions of other CAVs and can also include the observations of other CAVs. The centralized state-action value function, using the additional information, computes the state-action value of each CAV i. In this formulation, since each CAV knows the actions of all other CAVs, the environment to estimate the state-value function is now stationary. The revised formulation is given by

$$\mathcal{L}_i(\phi) = \mathop{\mathbb{E}}_{s,\mu,r,s'}\left[\left(Q_i^{\phi}(s_1, \cdots, s_N, \mu_1^{\theta}(s_1), \cdots, \mu_N^{\theta}(s_N)) - y_i\right)^2\right], \quad (6)$$

where,

$$y_i = r_i + \gamma Q_i^{\phi_t}\left(s_1', \cdots, s_N', \mu_1^{\theta_t}(s_1'), \cdots, \mu_N^{\theta_t}(s_N')\right). \quad (7)$$

C. Reward Function

Next, we provide a detailed description of different rewards/penalties imposed on each CAV to encourage distinctive individual and cooperative behaviors. CAV $i \in \mathcal{N}$ gets a reward $r_i \in \mathcal{R}_i$ when it transitions from state $s_i \in \mathcal{S}_i$ to next state $s_i' \in \mathcal{S}_i$ by taking an action $u_i \in \mathcal{U}_i$. We compute the reward at the current time step by the weighted sum of individual rewards (penalties) that CAV i receives to encourage (discourage) each of the following behaviors.

The speed of each CAV i needs to be within a certain range $[v_{\min}, v_{\max}]$. To enforce CAV i to learn and satisfy this constraint, we penalize (negative reward) any speed limit violation using $r_i^{\text{speed}} = -10$, if $v_i \geq v_{\max}$ or $v_i \leq v_{\min}$.

We encourage increased traffic throughput by rewarding CAVs for traveling as close the maximum speed limit as possible. This is defined as

$$r_i^{\text{speed}} = \frac{v_{\max} - \sqrt{(v - v_{\max})^2}}{v_{\max}}. \quad (8)$$

One critical aspect that the CAVs need to learn is to take actions that avoid collisions. For CAVs on the same road, we enforce this by penalizing rear-end collisions using $r_i^{\text{rear}} = -\frac{1}{(p_k - p_i)}$, if $0 < p_k - p_i < d_{\text{safe}}$, where k is the CAV in front of CAV i on the same road, so $p_i < p_k$.

To guarantee lateral safety as CAVs cross the merging zone, we impose $r_i^{\text{lateral}} = -\frac{1}{(d_i - d_j)}$ for CAVs i and j traveling on different roads if $d_j < S$ and $d_i < S$ and $0 < d_i - d_j < d_{\text{safe}}$. The first two conditions verify whether CAV i is in the merging zone at the same time as the vehicle in front of it, e.g., CAV j, while the third condition verifies whether a collision has occurred. Note that in this case, CAVs i and j belong to different roads and j is ahead of i so $d_j < d_i$. As mentioned previously, if a CAV is traveling on the on-ramp, we use its projected distance to merging, so d_j or d_i can be d_j' or d_i' respectively, depending on which one is on the merging road.

The total reward that CAV i receives at the current time step is given by $r_i = w_1 \cdot r_i^{\text{speed}} + w_2 \cdot r_i^{\text{rear}} + w_3 \cdot r_i^{\text{lateral}}$, where $w_1, w_2, w_3 \in \mathbb{R}_+$ are the weighting factors for individual rewards.

III. SIMULATION SETUP

In this section, we provide the specifics about the parameters of the different modules that are essential to execute the simulation intended to solve the problem presented in this paper. The simulation environment required for training the RL agents was custom-built in Python specifically for the highway merging scenario.

A. Neural Networks

In the formulation adopted for this study, each CAV i needs four different deep neural networks representing the critic, target-critic, actor, and target-actor networks. For all these networks, we use feedforward neural networks, and initialize the original and target networks identically. The decentralized actor network takes the state information as the input features so the number of units in its input layer is equal to the dimension of the state vector $s_{\text{dim}} = 6$. Since the centralized critic needs both the state and action information of all the CAVs the number of units in its network's input layer depends both on the dimension of the action vector $u_{\text{dim}} = 1$ and the number of CAVs N: $(s_{\text{dim}} + u_{\text{dim}}) \cdot N$. The rest of the hyperparameters that are common for both

TABLE I: Neural Networks Hyperparameters.

Parameter		Value
Units in hidden layers 1 and 2 (n_h^1, n_h^2)		64
Initial weights	Input layer	$\mathrm{U}(-w_1, w_1)$
	Hidden layer 1	$\mathrm{U}(-w_2, w_2)$
	Hidden layer 2	$\mathrm{U}(-w_3, w_3)$
Optimizer learning rate (α)		0.3

type of networks are given in Table I. We initialize the weights of input layer, first and second hidden layers using $w_1 = w_2 = 1$, $w_3 = 3 \times 10^{-3}$ respectively, and employ Adam-PyTorch as our optimizer. We consider ReLU as the activation function for the input layer and hidden layers for both actor and critic networks. Since the output of the actor networks needs to be within the acceleration/deceleration limits, we consider the activation function of its output layer to be the $\tanh$ function, while the output layer activation function of the critic network is a simple linear function.

B. Training

In order to learn the required control policies, we run the simulation for $15,000$ episodes, where the episode length is 50 s with a time step of 0.1 s, and if there is any collision in the current episode, we terminate it and start a new episode. At the start of every episode, CAVs are initialized close to the start of the control zone, on both the main and the merging roads, with speeds uniformly sampled from $[v_{\min}, v_{\max}]$. If all the CAVs in the network cross the merging zone without any collisions, we give an additional reward to each CAV in the network. During training, the neural networks of each CAV are updated periodically based on samples of experience drawn uniformly at random from the buffer of stored samples. After updating the original networks, we employ Polyak-Ruppert averaging [37] to update the target networks using a tunable polyak parameter $\tau = 0.01$, which causes a gradual update in the target networks to improve stability in training.

C. Simulation Parameters

We use the following simulation parameters: $v_{\min}$ = 5 m/s, $v_{\max}$ = 15 m/s, $u_{\min}$ = -3 m/s^2, $u_{\max}$ = 3 m/s^2, L = 90 m, S = 10 m, d_{safe} = 0.5 m. The speed limits are chosen based on the length of the control zone and the episode length such that meaningful interactions between the agents can occur before they exit the merging zone. The acceleration limits are typical for a common passenger vehicle. The weighting factors in the reward function are set as $w_1 = 1$, $w_2 = w_3 = 20$ to give a higher priority to avoid collisions when compared to the reward gained from going close to $v_{\max}$.

IV. SIMULATION RESULTS

After the training phase, we examine the learning performance by considering the 100-episode average rewards collected by all the CAVs, and this is shown in Fig. 2. The

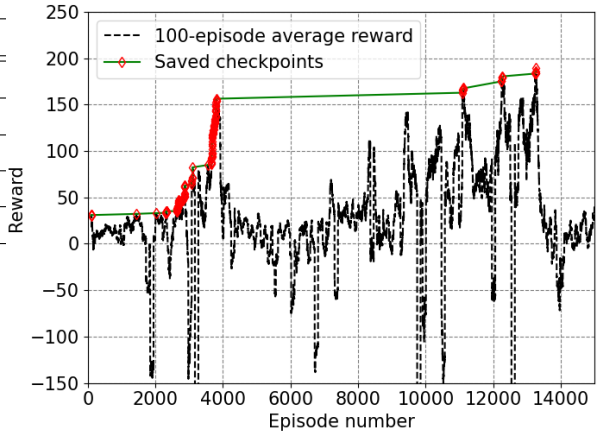
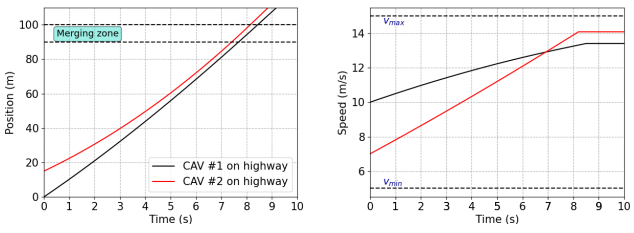


Fig. 2: 100-episode average rewards for the entire network.

rewards collected by the CAVs fluctuate during the training because there is no explicit exploitation strategy involved. This behavior is common in actor-critic methods; however, as it can be seen in Fig. 2, there is an upward trend in 100-episode average rewards showing the improvement in learning. To keep a record of the policies that resulted in accumulating higher rewards, we save the actor network parameters whenever the current 100-episode average reward is better than the previous best 100-episode average reward.

In following, we use three scenarios to demonstrate the learned behaviors. The policies learned from our framework are only responsible for controlling the CAVs while they are in the control zone and merging zone. After leaving the merging zone, the CAVs cruise at the same speed that they exited the merging zone [1].

To demonstrate rear-end collision avoidance behavior learned during the training process, we initialize two CAVs on the highway such that CAV #1 which is behind CAV #2 has a higher initial speed. The position and speed trajectories of two CAVs are shown in Fig. 3. As it can be seen in Fig. 3b, both CAVs are incentivized to increase their speed to $v_{\max}$. However, due to the rear-end safety constraint, CAV #1 increases its speed at a lower rate to avoid the collision.

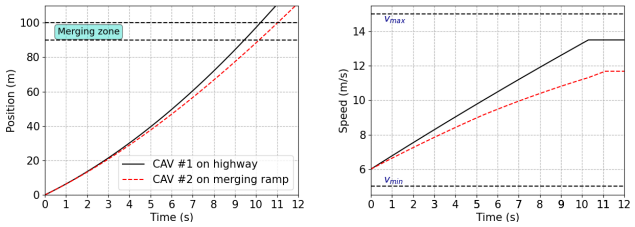


(a) Position (b) Speed

Fig. 3: Position and speed trajectories for rear-end collision avoidance scenario.

[1] Videos from our simulation analysis can be found at the supplemental site, https://sites.google.com/view/ud-ids-lab/MADRL.

For the lateral collision scenario, we initialize one CAV on the highway and one on the on-ramp with the same initial speeds. The position and speed trajectories of two CAVs are shown in Fig. 4. As it can be seen in Fig. 4b, CAV #2 on the merging road slows down in order to avoid lateral collision in the merging zone. Additionally, as CAV #1 exits the merging zone around 10 s, CAV #2 increases its speed with a higher rate.



(a) Position (b) Speed

Fig. 4: Position and speed trajectories for lateral collision avoidance scenario.

For the final scenario, to demonstrate the satisfaction of rear-end safety and lateral safety constraints when there are more CAVs present in the network, we consider coordination of 8 CAVs. We pick the initial position of these CAVs randomly, while the initial speed of CAVs on the highway and on-ramp roads are set to 13 m/s and 12 m/s, respectively. Fig. 5 illustrates the position trajectories of the CAVs on the highway (solid lines) and on the on-ramp road (dashed lines). As it can be seen from the figure, the CAVs learned to satisfy the rear-end safety constraint from the initial time until the time they exit the merging zone. Additionally, trajectories of CAVs from different paths do not intersect inside the merging zone, showing the satisfaction of the lateral safety constraint. The close-up view in Fig. 5 presents the position trajectories of the CAVs inside the merging zone, confirming that the CAVs indeed avoid lateral collisions in the merging zone by maintaining distance more than d_{safe} with their respective front vehicles.

For this scenario, we initialize the actor network parameters of each CAV with trained network parameters of one CAV from a total of three CAVs involved in the training process. This shows that we can use our framework to train the system for some finite number of vehicles, and transfer the learned policy to any number of vehicles.

We further demonstrate the effectiveness of our approach in eliminating the stop-and-go driving by performing five different simulations with random initial speeds ranging from 6 m/s to 13 m/s. The instantaneous average, maximum, and minimum speed of CAVs inside the control zone for the merging road and highway for all five simulations are shown in Fig. 6. The minimum speed of CAVs for both roads over five experiments is positive indicating smooth traffic flow.

The above scenarios clearly illustrate the efficacy of the policies learned using the reinforcement learning framework presented in this paper. The actions derived from these

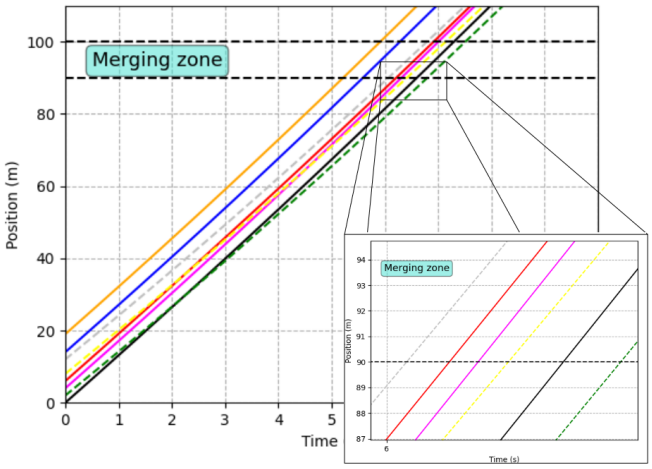


Fig. 5: Position trajectories of 8 CAVs in the network.

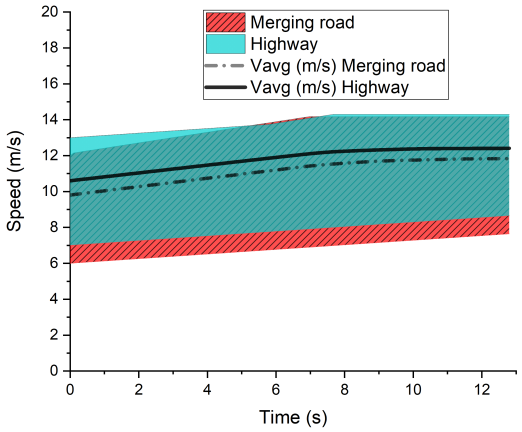


Fig. 6: Speed range for merging and highway roads over 5 experiments.

policies ensure improved traffic throughput as a consequence of high-speed travel, but simultaneously adhering to the set speed limits, along with safe coordination to execute merging maneuver without any rear-end and lateral collisions.

V. CONCLUDING REMARKS AND DISCUSSION

In this paper, we proposed a decentralized, multi-agent reinforcement learning-based framework for coordinating CAVs through a highway merging scenario. In our framework, we employed an actor-critic architecture with a centralized critic and decentralized actors to avoid the problem of a non-stationary environment [36] induced by decentralized learning CAVs. Our framework enables the capability of implementing the learned policies in any number of CAVs by transferring the policies which were learned through a few learning CAVs in the training process. In addition to ensuring rear-end and lateral safety, our choice of reward function encouraged the CAVs to learn to travel at high speeds which in turn results in smoother traffic flow. Finally, we showed the

effectiveness of the proposed approach through several simulations. As part of ongoing research, we are extending the current framework to other traffic scenarios, including urban intersections and roundabouts, while simultaneously utilizing high-fidelity dynamics models. In addition to that, we are investigating the effects of noise originating from vehicle-level control, and also of errors and delays in communication since in its current form the CAVs in our framework rely on accurate information from other CAVs, and hence are not robust to uncertainties. Our framework could also be extended to study the interaction of human-driven vehicles and CAVs in mixed-traffic scenarios.

REFERENCES

[1] B. Schrank, B. Eisele, and T. Lomax, "2019 Urban Mobility Scorecard," Texas A& M Transportation Institute, Tech. Rep., 2019.

[2] N. Spiller, K. Blizzard, and R. Margiotta, "Recurring Traffic Bottlenecks: A Primer, Focus on Low-Cost Operational Improvements, Fourth Edition," U.S. Department of Transportation. Federal Highway Administration," , 2017.

[3] National Highway Traffic Safety Administration, "Traffic Safety Facts 2017: A Compilation of Motor Vehicle Crash Data," https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812806, 2019.

[4] ——, "Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey," https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812115, 2015.

[5] L. Zhao and A. A. Malikopoulos, "Enhanced mobility with connectivity and automation: A review of shared autonomous vehicle systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 1, pp. 87–102, 2022.

[6] 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.

[7] T. Ersal, I. Kolmanovsky, N. Masoud, N. Ozay, J. Scruggs, R. Vasudevan, and G. Orosz, "Connected and automated road vehicles: state of the art and future challenges," *Vehicle system dynamics*, vol. 58, no. 5, pp. 672–704, 2020.

[8] J. Rios-Torres and A. A. Malikopoulos, "Impact of Partial Penetrations of Connected and Automated Vehicles on Fuel Consumption and Traffic Flow," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 453–462, 2018.

[9] W. Levine and M. Athans, "On the optimal error regulation of a string of moving vehicles," *IEEE Transactions on Automatic Control*, vol. 11, no. 3, pp. 355–361, 1966.

[10] G. Raravi, V. Shingde, K. Ramamritham, and J. Bharadia, "Merge algorithms for intelligent vehicles," in *Next Generation Design and Verification Methodologies for Distributed Embedded Control Systems*. Springer, 2007, pp. 51–65.

[11] D. Marinescu, J. Čurn, M. Bouroche, and V. Cahill, "On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 900–906.

[12] T. Awal, L. Kulik, and K. Ramamohanrao, "Optimal traffic merging strategy for communication- and sensor-enabled vehicles," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct 2013, pp. 1468–1474.

[13] I. A. Ntousakis, I. K. Nikolos, and M. Papageorgiou, "Optimal vehicle trajectory planning in the context of cooperative merging on highways," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 464–488, 2016.

[14] J. Rios-Torres and A. A. Malikopoulos, "Automated and Cooperative Vehicle Merging at Highway On-Ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 780–789, 2017.

[15] A. M. I. Mahbub, V. Karri, D. Parikh, S. Jade, and A. A. Malikopoulos, "A decentralized time- and energy-optimal control framework for connected automated vehicles: From simulation to field test," in *SAE Technical Paper 2020-01-0579*. SAE International, 2020.

[16] L. E. Beaver, B. Chalaki, A. M. Mahbub, L. Zhao, R. Zayas, and A. A. Malikopoulos, "Demonstration of a Time-Efficient Mobility System Using a Scaled Smart City," *Vehicle System Dynamics*, vol. 58, no. 5, pp. 787–804, 2020.

[17] B. Chalaki, L. E. Beaver, and A. A. Malikopoulos, "Experimental validation of a real-time optimal controller for coordination of cavs in a multi-lane roundabout," in *31st IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 504–509.

[18] J. Rios-Torres and A. A. Malikopoulos, "A Survey on Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[19] J. Guanetti, Y. Kim, and F. Borrelli, "Control of connected and automated vehicles: State of the art and future challenges," *Annual Reviews in Control*, vol. 45, pp. 18–40, 2018.

[20] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.

[21] X. Ji and Z. He, "An optimal control method for expressways entering ramps metering based on q-learning," in *2009 Second International Conference on Intelligent Computation Technology and Automation*, vol. 1. IEEE, 2009, pp. 739–741.

[22] C. Jacob and B. Abdulhai, "Integrated traffic corridor control using machine learning," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4. IEEE, 2005, pp. 3460–3465.

[23] M. Davarynejad, A. Hegyi, J. Vrancken, and J. van den Berg, "Motorway ramp-metering control with queuing consideration using q-learning," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2011, pp. 1652–1658.

[24] B. Chalaki and A. A. Malikopoulos, "A hysteretic q-learning coordination framework for emerging mobility systems in smart cities," in *2021 European Control Conferences (ECC)*, 2021, pp. 17–22.

[25] E. Ivanjko, D. K. Nečoska, M. Gregurić, M. Vujić, G. Jurković, and S. Mandžuka, "Ramp metering control based on the q-learning algorithm," *Cybernetics and Information Technologies*, vol. 15, no. 5, pp. 88–97, 2015.

[26] L. Wang, F. Ye, Y. Wang, J. Guo, I. Papamichail, M. Papageorgiou, S. Hu, and L. Zhang, "A q-learning foresighted approach to ego-efficient lane changes of connected and automated vehicles on freeways," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1385–1392.

[27] D. C. K. Ngai and N. H. C. Yung, "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 509–522, 2011.

[28] 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.

[29] P. Wang and C.-Y. Chan, "Formulation of deep reinforcement learning architecture toward autonomous driving for on-ramp merge," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[30] T. Nishi, P. Doshi, and D. Prokhorov, "Merging in congested freeway traffic using multipolicy decision making and passive actor-critic learning," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 287–297, 2019.

[31] O. Nassef, L. Sequeira, E. Salam, and T. Mahmoodi, "Building a lane merge coordination for connected vehicles using deep reinforcement learning," *IEEE Internet of Things Journal*, 2020.

[32] T. Ren, Y. Xie, and L. Jiang, "Cooperative highway work zone merge control based on reinforcement learning in a connected and automated environment," *arXiv preprint arXiv:2001.08581*, 2020.

[33] N. P. Farazi, T. Ahamed, L. Barua, and B. Zou, "Deep reinforcement learning and transportation research: A comprehensive review," *arXiv preprint arXiv:2010.06187*, 2020.

[34] S. M. Seliman, A. W. Sadek, and Q. He, "Automated vehicle control at freeway lane-drops: a deep reinforcement learning approach," *Journal of Big Data Analytics in Transportation*, pp. 1–20, 2020.

[35] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[36] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in neural information processing systems*, 2017, pp. 6379–6390.

[37] B. T. Polyak, "New stochastic approximation type procedures," *Automat. i Telemekh*, vol. 7, no. 98-107, p. 2, 1990.