Centralized Policy Learning for Consensus Control of Connected and Automated Vehicles

Sidra Ghayour Bhatti [1], Pei Yu Chang[2], Nur Uddin Javed[3], Qadeer Ahmed[2]

[1]Center for Automotive Research, The Ohio State University, Columbus, OH, 43212, USA
[2] Department of Mechanical and Aerospace Engineering, The Ohio State University, OH, 43210, USA
[3] Department of Electrical and Computer Engineering, The Ohio State University, OH, 43210, USA
bhatti.39@osu.edu, chang.2314@osu.edu, javed.31@osu.edu, ahmed.358@osu.edu

Abstract

Connected and automated vehicles (CAVs) play a key role in the intelligent transportation system of the near future. They offer a promising solution for different challenges, including increased highway accidents, high energy consumption, and growing traffic congestion. The advancements in control theory and reinforcement learning (RL) have given rise to consensus control techniques for effective coordination of multiple CAVs. Multiagent RL (MARL) algorithms are widely used in literature for the consensus problem of CAVs under different driving conditions; however, they encounter several issues, including non-stationarity and computational complexity, that hinder their applicability for real-time applications. To resolve these issues, an approach similar to centralized training and centralized execution (CTCE) utilizing single-agent deep deterministic policy gradient (DDPG) is proposed for consensus control of multiple CAVs following a leader-follower pattern. The central agent is used to generate control policies for all CAVs, mitigating the non-stationarity issues while ensuring consensus. The computational complexity is reduced by using the shared critic network for all CAVs, which helps in efficient and coordinated policy optimization. Reward shaping for the consensus problem is performed using the combination of continuous and discrete reward while ensuring collision avoidance among the CAVs. The effectiveness of the proposed DDPG-based consensus control is demonstrated by simulating various traffic scenarios, including staright line path following and merging, where the effective consensus of multiple CAVs is observed. The proposed approach offers a scalable and practical solution for coordinated control of modern autonomous vehicles.

Introduction

The modern transportation system has revolutionized daily life, facilitating the efficient transportation of passengers and goods within and beyond the national boundaries. However, some challenges are observed in the 21st century, including the increasing number of vehicles resulting in more highway accidents, increased energy consumption exacerbated by traffic congestion, and more commute hours. The issues faced by the modern transportation system can be mitigated by connected and automated vehicles (CAVs). With their intelligent technologies and advanced sensors, CAVs can introduce highly efficient vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. The advancement in control theory and autonomous systems has brought attention to consensus control over the past decade. This approach tries to achieve consensus among states of multiple CAVs in a fully distributed manner. Consensus control can be applied in numerous applications, including formation control (Oh, Park, and Ahn 2015) (Li et al. 2015), distributed freeway traffic control (Kim and Ahn 2014), coordination among multiple robots (Alonso-Mora et al. 2019), decision-making processes in social networks (Amelin et al. 2018), etc. In (Koung et al. 2020), a consensus control law for formation control, navigation, and obstacle avoidance of multiple-wheeled mobile robots is presented. Ren provides an overview of consensus problems in multi-agent cooperative control in (Ren, Beard, and Atkins 2005). The research summarizes theoretical consensus-seeking results under time-invariant and dynamically changing information exchange topologies. In recent years, autonomous driving strategies leveraging learning-based approaches, including reinforcement learning (RL) have gained much attention. These methods are appealing as they can adapt to dynamic, complex environments, and decision-making can be improved through the experience. The application of RL in different driving scenarios, including car-following, has shown a substantial increase in the performance (Zhou, Fu, and Wang 2020; Song et al. 2023). Recently, considering the situation of high-dimensional state space in RL, the integration of deep learning (DL) and RL has been proposed as deep reinforcement learning (DRL), whereby a deep neural network (DNN) represents the agent's decision-making policy. In (Ghraizi, Talj, and Francis 2023), a DRL-based adaptive cruise control (ACC) system is proposed that creates safe, flexible, and responsive car-following policies. The approach uses a discrete high-level action space and a comprehensive multi-objective reward function. In (Lin, McPhee, and Azad 2020b), the research compares DRL and model predictive control (MPC) for ACC in car-following scenarios. In urban environments, another challenge faced by CAVs is the seamless merging of on-ramp CAVs with the main-lane CAVs while ensuring efficiency and safety. The CAVs on the main lane should adjust their speed to accommodate the merging CAVs, while on-ramp CAVs should try to merge promptly while ensuring safety (Bevly et al. 2016;

Lin, McPhee, and Azad 2020a; el abidine Kherroubi, Aknine, and Bacha 2021).

Related Works

DRL algorithms have shown state-of-the-art performance in the field of CAVs, including significant improvements in efficiency and safety (Liu et al. 2021; Kiran et al. 2021; Chen, Yuan, and Tomizuka 2019; Talpaert et al. 2019). Recently, different DRL algorithms have shown remarkable performance, including deep Q-network (DQN) (Okuyama, Gonsalves, and Upadhay 2018; Shi et al. 2020), deep deterministic policy gradient (DDPG) (Ma et al. 2024), proximal policy optimization (PPO) (Wei et al. 2019), etc. DQN is a model-free, off-policy DRL algorithm that uses Q-learning along with DNNs to handle the high-dimensional state space. In (Zhang et al. 2018), naturalistic driving data integrated with perception mechanisms is used to observe the performance of double Q-learning that outperforms traditional deep Q-learning in terms of policy quality and value accuracy. The research in (Min, Kim, and Huh 2018), uses deep Q-learning to train a supervisor for coordination of driver assistance system (DAS) functions for autonomous highway driving scenarios. In environments with continuous action spaces, DQN does not perform well without discretization and low training efficiency is also observed (Gu et al. 2016). PPO is a model-free, on-policy DRL algorithm that uses clipped surrogate objective functions for stable training. In (Zhao et al. 2024), PPO is used for autonomous driving and sensor inputs are directly mapped to control commands for vehicle. The CARLA simulator is used for validation of experimental results, demonstrating the practical application of the proposed research. The lane-changing strategy using PPO is proposed in (Ye et al. 2020) to address the challenge of safe maneuvers in dense traffic. PPO is not sample efficient and requires more data for effective training as compared to DDPG. In (Zou, Xiong, and Hou 2020), DRL framework is proposed that integrates imitation learning (IL) with DDPG to pre-train DDPG with appropriate initialization of parameters. The experimental results show better performance as compared to traditional DDPG. In (Wang, Li, and Chan 2019), lane-changing control problem is addressed using the DDPG algorithm while ensuring stability and safety and different driving scenarios are considered. In (Xu et al. 2018), the non-stationarity issue in flocking control problem for multi-vehicle systems is addressed using a DDPG framework with centralized training and distributed execution.

Problem Statement and Contributions

Multi-agent RL (MARL) approaches encounter challenges such as non-stationarity and computational complexity. The non-stationarity issue can disrupt coordination among CAVs, delay consensus, and compromise safety in consensus control problems (Xu et al. 2018; Wong et al. 2023; Gronauer and Diepold 2022). The dynamically changing policies for each CAV during training create a non-stationary environment for neighboring CAVs. To address this challenge and reduce computational complexity, a centralized training and centralized execution (CTCE) approach using single-agent DDPG is proposed. This method ensures effective policy learning in non-stationary environments for multiple CAVs operating in a leader-follower pattern. A central agent is used that generates the unified control policies for all CAVs ensuring consensus among vehicles. A shared critic network reduces complexity by eliminating the need for separate critic networks for each CAV while ensuring coordinated and efficient policy generation. This centralization minimizes redundant computations, reduces the number of parameters, and streamlines gradient updates, leading to faster convergence and lower memory usage. The main contributions of the proposed research work are:

- The CTCE approach employs single-agent DDPG to tackle the non-stationarity challenge in consensus control of multiple CAVs, ensuring their efficient coordination across various traffic scenarios.
- The reward shaping is done using the combination of continuous and discrete rewards to promote consensus among multiple CAVs while ensuring collision avoidance.

This article is organized as follows: The problem formulation followed by the architecture of the decision network is detailed in Section . Section includes the details about the reward shaping for the consensus control problem. The simulation results are demonstrated in Section . Finally, the conclusion is drawn in Section .

Problem Formulation and Decision Network

The proposed research uses consensus control based on single-agent DDPG for multiple CAVs that follow a leader-follower pattern. The RL environment contains multiple CAVs, each modeled using the bicycle kinematics model. During training, efficient control commands are generated by the DDPG-based decision network. A set of actor-critic networks makes up the decision network, which generates control policies for all CAVs at the same time. In consensus control, information sharing is essential among the multiple CAVs. Each CAV modifies its current states throughout the process using information collected from neighboring CAVs. The performance of the DDPG-based decision network for consensus control is evaluated in a straight-line path following and merging scenarios discussed in Section . Before diving into the consensus control framework based on DDPG, it is crucial to consider the dynamic models of CAVs that are involved in consensus problem. DDPG-based consensus control is intended for straight line path following and highway merging, which entails coordinating several CAVs to effectively enter the main traffic flow to reach the same consensus. These CAVs can share their states and interact via consensus control.

Environment: Dynamic Model of CAVs

The bicycle kinematics model is considered for all CAVs in this research, including the leader and follower CAVs. There are three CAVs: CAV1 acts as the leader, while CAV2 and CAV3 behave as followers. CAV2 aims to minimize its state errors relative to the leader (CAV1), and CAV3 strives to minimize its state errors relative to CAV2, thereby achieving consensus. Let O be an inertial frame of reference with the

origin represented by a point m_0. The dynamics of the i^{th} CAV can be described as:

$$\dot{x_i} = v_{r,i}(\cos\psi_i - \sin\psi_i \tan\beta_i), \tag{1a}$$

$$\dot{y_i} = v_{r,i}(\sin\psi_i + \cos\psi_i \tan\beta_i), \tag{1b}$$

$$\dot{\psi_i} = \frac{v_{r,i}}{l_r}\tan\beta_i, \tag{1c}$$

$$\dot{v}_{r,i} = a_{r,i}, \tag{1d}$$

$$\dot{\beta_i} = \omega_i, \tag{1e}$$

where the position of the center of gravity of the vehicle is denoted as x_i, y_i with respect to origin m_0. The orientation of the body-fixed frame (B_i) is denoted as ψ_i with respect to inertial frame of reference O. $v_{r,i}$ is the velocity of the rear wheel of vehicle with respect to O. The slip angle of vehicle center of gravity relative to (B_i) is denoted as β_i. It is assumed that $|B_i| < \frac{\pi}{2}$. The states of the i^{th} vehicle are denoted as $z_i = [x_i\ y_i\ \psi_i\ v_i; \beta_i]^\top$. The control input is denoted as $u_i = [\omega_i\ a_{r,i}]$, where ω_i represents the angular velocity of the angle of slip and $a_{r,i}$ is the linear acceleration of the rear wheel of the vehicle. l_r represents the distance from the center of gravity to the center of the rear wheel of the vehicle. In the proposed research, the control inputs $(\omega_i, a_{r,i})$ are obtained from DDPG-based decision network.

Agent: Architecture of Decision Network

The decision network is composed of a DDPG agent that is a model-free, off-policy DRL method comprising of two actor-critic networks and chooses actions based on the states of CAVs. The DDPG based consensus control framework for multiple CAVs following a leader-follower pattern is shown in Fig. 1. Actions are generated by the actor network, while the critic network assists the actor in refining its actions. During training, both of these networks learn together. DDPG is used to solve continuous control problems with a deterministic policy that maps the states to the actions. It integrates concepts from deep Q-network (DQN) and deterministic policy gradient (DPG). The success of DQN served as inspiration for the creation of deep deterministic policy gradient (DDPG), which aims to enhance performance for applications requiring a continuous action space. DDPG simultaneously learns a policy and a Q-function. The Q-function is learned using the Bellman equation and the policy is learned using the Q-function. Unlike DQN, which uses a probability distribution across actions, the actor is a policy network that uses the state as input and outputs the precise action (continuous). The goal of the critic's evaluation of the control actions is to determine the total future return for the control actions. The actor network consists of an input layer that receives the observations from the environment, followed by three fully connected hidden layers each having N neurons, activated by ReLU activation function. The output layer contains six neurons to match the dimensions of control actions applied to the environment (two control actions for each CAV).

Tanh activation is applied on the output layer followed by the scaling layer used to scale the control actions within the desired bounds. An action path and an observation path are the two branches of the critic network. The action path, which has a single fully connected layer, merges with the observation path at an addition layer. The observation path has three fully connected layers with L neurons and ReLU activations. The final layer, which is designed for continuous action-value estimate, receives the combined path as input and produces the Q-value. The weights of the neurons in the decision network determine the control policy. θ^μ and θ^Q represent the network weighting parameters for the actor and critic networks, respectively. The gradient descent optimization process is used to minimize the mean-squared loss between the Q values computed by critic network and $Q^{'}$ values obtained from the target actor-critic network. This is the definition of Bellman's principle of optimality given as:

$$MSE = \frac{1}{N}\sum_i \left(y_i - Q(s_i, a_i|\theta^Q)\right)^2 \tag{2a}$$

$$MSE = \frac{1}{N}\sum_i \Big(r(s_i, a_i) + \gamma Q'(s_{i+1}, \mu'(s_{i+1})|\theta^{Q'}) - Q(s_i, a_i|\theta^Q)\Big)^2 \tag{2b}$$

where $r(s_i, a_i)$ is reward function obtained by taking the action at the states at i^{th} timestep and γ is the discount factor. The output actions obtained from actor network are based on the network weight parameters (θ^μ). During training, the weights of actor network (θ^μ) are updated to maximize the Q value. The policy loss is computed by taking derivative of objective function with respect to θ^μ. The mean of the sum of gradients is computed by considering the mini-batches from experience:

$$\nabla_{\theta^\mu} J(\theta) = \frac{1}{N}\sum_i \Big[\nabla_a Q(s, a|\theta^Q)\Big|_{s=s_i, a=\mu(s_i)} \cdot \nabla_{\theta^\mu}\mu(s|\theta^\mu)\Big|_{s=s_i}\Big] \tag{3}$$

The DDPG network uses another set of actor-critic network as target network in order to stabilize the training process; the target network's weight parameters are modified as:

$$\theta^{\mu^{'}} = \tau\theta^{\mu^{'}} + (1-\tau)\theta^\mu \tag{4a}$$

$$\theta^{Q^{'}} = \tau\theta^{Q^{'}} + (1-\tau)\theta^Q \tag{4b}$$

where $\theta^{\mu^{'}}$ and $\theta^{Q^{'}}$ are the weight parameters of target-actor and target-critic networks, respectively. τ defines the weight update rate for target networks. A soft update occurs in DDPG, in which only a portion of the main network weights are transferred to the target networks. The target networks follow the main networks gradually as they are time-delayed replicas of their main networks. In DDPG, the experience replay buffer is a memory structure that holds the agent's previous experiences (state, action, reward, and next state). The actor and critic networks are updated during training by randomly sampling mini-batches of events from this buffer. In order to promote environmental exploration, DDPG adds noise to the deterministic actions. This guarantees that the agent learns more effective policies during training.

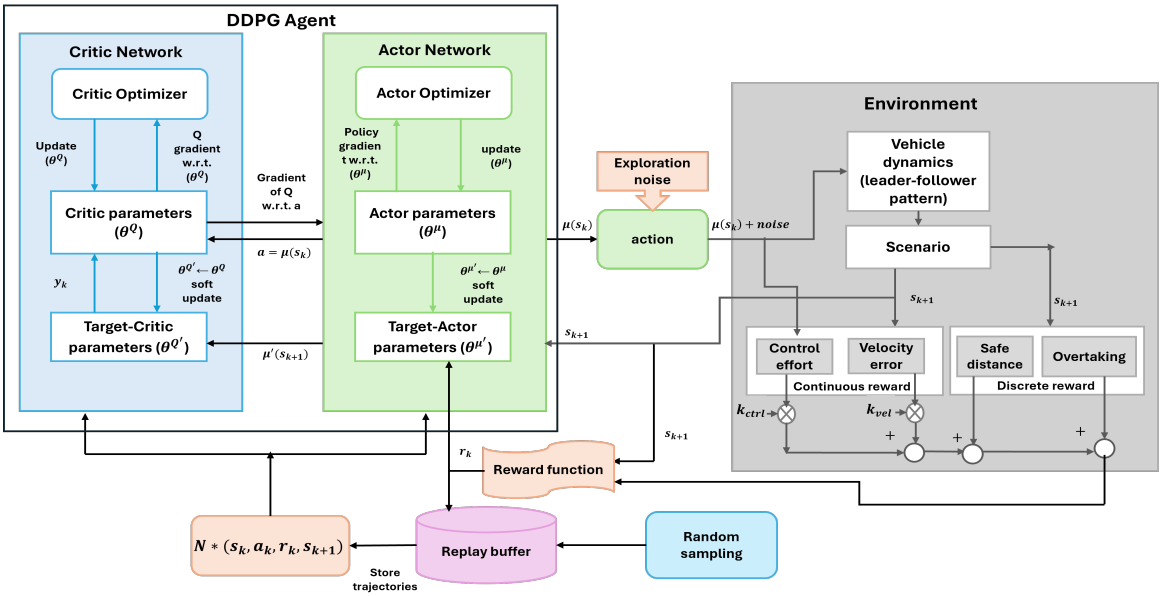


Figure 1: DDPG-based consensus control of multiple CAVs

Reward Design for Consensus Control

In order to achieve consensus among multiple CAVs, a well-designed reward function is required that synchronizes the actions of individual CAVs toward a shared goal. CAVs are motivated to cooperate rather than pursue opposing objectives when rewards are thoughtfully designed to promote cooperative behavior. However, poor reward design can lead to misaligned goals, which would impede the consensus process and decrease the performance of the system as a whole. The reward function R to achieve consensus is designed to incentivize the cooperative movement of CAVs while penalizing deviations from their desired goal. R is the combination of continuous and discrete rewards and is expressed as:

$$R = R_E + R_D + R_{OT} \tag{5}$$

The components of reward function are explained as:

1. Continuous Reward (R_E): This reward will penalize the deviations in velocities of consecutive CAVs. It will also try to minimize the control efforts for all CAVs.

$$R_E = -(k_{ref}.\Delta V_{1,ref} + k_v.(\Delta V_{1,2} + \Delta V_{2,3}) + k_{ctrl}.CA) \tag{6}$$

where,

- $\Delta V_{1,ref}$ is the squared error between leader velocity and it's reference velocity.
- $\Delta V_{1,2}$ is the squared velocity error between CAV1 and CAV2.
- $\Delta V_{2,3}$ is the squared velocity error between CAV2 and CAV3.
- CA is the sum of squared control actions for all CAVs.
- k_{ref}, k_v, and k_{ctrl} are the gains for velocity error of leader, velocity error among consecutive CAVs, and gain for penalizing the control efforts, respectively.

2. Discrete Reward (R_D): This reward will encourage the follower-CAVs to maintain safe distance from their leader-CAVs to prevent the potential collisions.

$$R_{\mathrm{D}} = \sum_{i,j} \begin{cases} \text{incentive}, & \text{if actualDistance}_{ij} \geq \text{SD} \\ \text{penalty}, & \text{if actualDistance}_{ij} < \text{SD} \end{cases} \tag{7}$$

where,

- SD is the safe distance between CAV_i and CAV_j .

3. Discrete Reward (R_{OT}): This reward will encourage the follower-CAV to follow its preceding-CAV without overtaking.

$$R_{\mathrm{OT}} = \sum_{i,j} \begin{cases} \text{incentive}, & \text{if } x_i > x_j \\ \text{penalty}, & \text{if } x_i < x_j \end{cases} \tag{8}$$

where,

- x_i is the position of leader-CAV and x_j is the position of follower-CAV.

The combination of continuous and discrete rewards will help to achieve consensus among multiple CAVs while ensuring that CAVs should not collide and overtake. Continuous rewards for control effort and velocity will guide gradual adjustments toward consensus by minimizing the errors and promoting smooth dynamics. While discrete rewards for safe distance and overtaking will enforce safe distance requirements and penalize unsafe following distances or overtaking events. When combined, these rewards make sure that CAVs reach consensus quickly and behave in a safe and coordinated manner.

Simulation Results

It is important to choose appropriate hyperparmeters for DDPG agent to achieve a balanced trade-off between convergence, learning stability, and exploration versus exploitation for learning optimal policies. The DDPG agent is trained, and the simulation results are analyzed under different scenarios to evaluate its performance in achieving consensus among multiple CAVs. Two primary scenarios are considered: Straight-line path following and merging.

Parameters of DDPG Network

It is crucial to carefully choose the hyperparameters of DDPG agent to achieve the consensus by maintaining a balance between learning stability and performance. Separate optimizer settings are used for both networks: the critic uses a slightly higher learning rate (1×10^{-3}) as compared to the actor (1×10^{-4}) to guarantee smoother policy updates. The size of the experience buffer is set to store upto (1×10^{6}) samples and the mini-batch size is set to 128. The discount factor is used to prioritize the long-term rewards and its value is set to 0.99. The actor-critic target networks are updated using soft update so that target smoothing factor is set to (1×10^{-3}) that ensure stable updates for target networks. Gaussian noise is added to the actions to allow the agent to explore new policies. Together, these hyperparameters allow the agent to effectively manage the trade-off between exploration, stability, and convergence while learning an optimal policy for the consensus control problem.

Simulation Scenarios

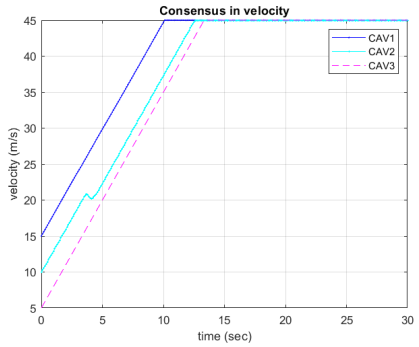
In order to evaluate the performance of the DDPG-based decision network for consensus control of multiple CAVs, different scenarios are considered. Two primary scenarios considered are straight-line path following and merging scenarios.

Scenario-1: Straight Line Path Following for 3 CAVs The highway considered in this scenario is a single-lane road having a width of about 4 meters, with the center of the lane located at y = 2 m. In this scenario, it is assumed that all three CAVs are traveling along the centerline of single-lane highway. CAV1 serves as the leader vehicle, following a straight path and heading from the negative to the positive x-axis direction trying to track the reference velocity of 45 m/s. The other two vehicles, CAV2 and CAV3, act as follower vehicles. The control inputs for all three CAVs are obtained from the trained DDPG-based decision network. The objective of this scenario is to ensure that the follower CAVs achieve consensus in velocity with the leader CAV while maintaining a safe intervehicle distance of at least or greater than 4 meters. The initial conditions (ICs) of the vehicles are given in Table 1.

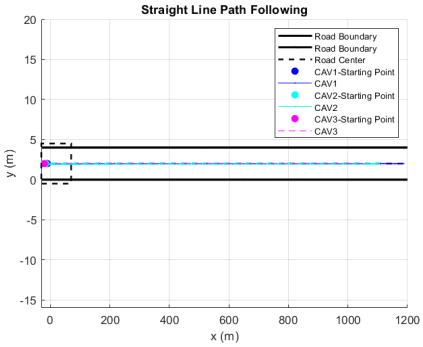
Table 1: Initial states of 3 CAVs for straight line path following

CAVs	Initial States $[x_0\ y_0\ \psi_0\ v_0; \beta_0]^\top$
Leader CAV	[-10, 2, 0, 15, -0.01]
Follower-CAV1	[-15, 2, 0, 10, -0.01]
Follower-CAV2	[-20, 2, 0, 5, 0.01]

The results illustrating the velocity consensus among all CAVs and their respective trajectories in the (x,y) coordinates are presented in Fig. 2(a) and Fig. 2(b), respectively. In Fig. 2(a), it can be observed that the leader CAV successfully tracks the reference velocity of 45 m/s, while both follower CAVs also strive to achieve a velocity consensus with the leader CAV. The (x,y) trajectories of all CAVs given in Fig. 2(b), show that all CAVs travel along the centerline of the highway while achieving a velocity consensus.



(a) Velocity Consensus



(b) Trajectories of CAVs

Figure 2: Scenario1: Straight line path following for 3 CAVs.

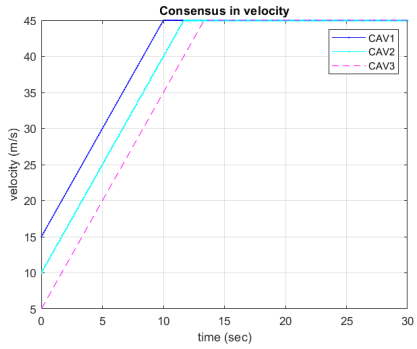
Scenario-2: Highway Merging The objective of this scenario is that all CAVs should follow the leader CAV at the same velocity while maintaining alignment with the y-axis. Both the leader-CAV and CAV1 travel on the main lane, whereas CAV3 is on the merging lane and attempts to merge seamlessly with the main-lane CAVs while achieving the consensus in velocity and maintaining alignment across the y-axis. The initial conditions (ICs) of the vehicles are given in Table 2.

Table 2: Initial states of 3 CAVs for merging

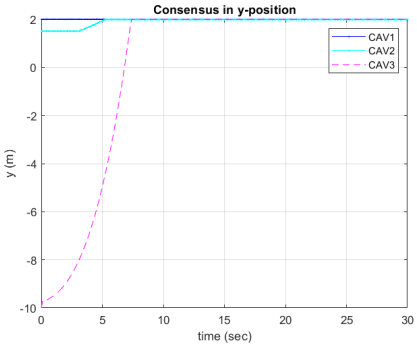
CAVs	Initial States $[x_0\ y_0\ \psi_0\ v_0; \beta_0]^\top$
Leader CAV	[-10, 2, 0, 15, -0.01]
Follower-CAV1	[-15, 1.5, 0, 10, -0.01]
Follower-CAV2	[-20, -10, 0.54, 5, 0.01]

The results illustrating the velocity consensus among all

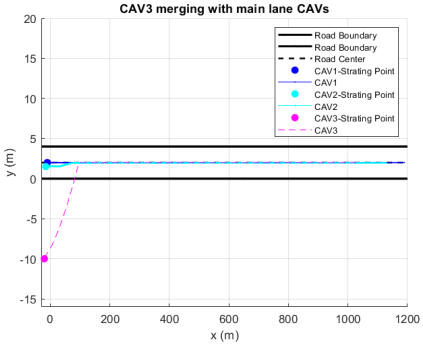
CAVs, along with their y-axis alignment and respective trajectories in the (x,y) coordinates are presented in Fig. 3(a), Fig. 3(b), and Fig. 3(c), respectively.



(a) Velocity Consensus



(b) Position Consensus



(c) Trajectories of CAVs

Figure 3: Scenario2: Merging scenario for 3 CAVs.

In Fig. 3(a), it can be observed that the leader CAV successfully tracks the reference velocity of 45 m/s, while both follower CAVs also strive to achieve a velocity consensus with the leader CAV. CAV3 also strives to perform y-axis alignment with main-lane CAVs, as shown in Fig. 3(b). The (x,y) trajectories of all CAVs, as shown in Fig. 3(c) depict that all CAVs travel along the centerline of the highway while achieving consensus in velocity and y-axis alignment.

Scenario-3: Straight Line Path Following for 4 CAVs This scenario is similar to the first scenario but here it is assumed that four CAVs are traveling along the centerline of a single-lane highway. CAV1 serves as the leader vehicle, following a straight path and heading from the negative to the positive x-axis direction, trying to track the reference velocity of 45 m/s. The other three vehicles, CAV2, CAV3, and CAV4, act as follower vehicles and try to achieve velocity consensus with the leader vehicle. The initial conditions (ICs) of the vehicles are given in Table 3.

Table 3: Initial states of 4 CAVs for straight line path following

CAVs	Initial States $[x_0\ y_0\ \psi_0\ v_0; \beta_0]^\top$
Leader CAV	[-10, 2, 0, 25, -0.01]
Follower-CAV1	[-15, 2, 0, 20, -0.01]
Follower-CAV2	[-20, 2, 0, 12, 0.01]
Follower-CAV3	[-25, 2, 0, 9, -0.01]

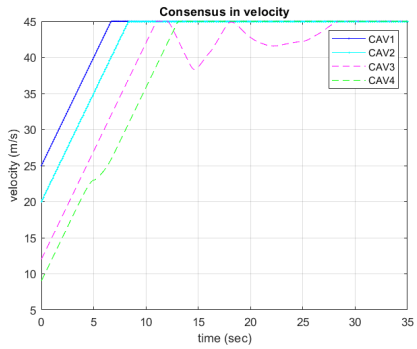
The results illustrating the velocity consensus among the four CAVs and their respective trajectories in the coordinates (x, y) are presented in Fig. 4(a) and Fig. 4(b), respectively. In Fig. 4(a), it can be observed that the leader CAV successfully tracks the reference velocity of 45 m/s, while all three follower CAVs also strive to achieve a velocity consensus with the leader CAV. The (x,y) trajectories of four CAVs given in Fig. 4(b), show that all CAVs travel along the centerline of the highway while achieving a velocity consensus.

The proposed algorithm can be extended for evaluation in real-world driving situations; future work could incorporate simulations with real-world constraints such as sensor noise, communication delays, and mixed traffic involving both autonomous and human-driven vehicles. Furthermore, deploying the algorithm on scaled autonomous vehicle platforms or in controlled test environments can provide insight into its practical feasibility, including handling environmental uncertainties, robustness to unexpected events, and adaptability to varying traffic densities.

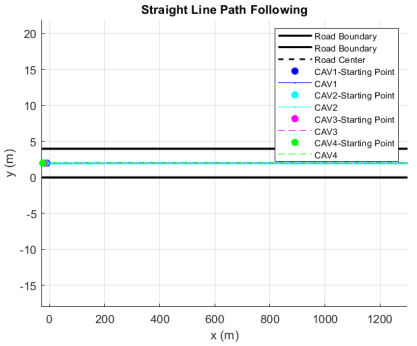
Comparison With Other Consensus Control Methods

The proposed CTCE framework and the COnsensus LeArning (COLA) algorithm in (Xu et al. 2023) represent distinct approaches to cooperative multi-agent reinforcement learning. While CTCE centralizes both training and execution to ensure globally optimized policies and mitigate non-stationarity, COLA achieves decentralized execution by inferring a consensus signal using contrastive learning, enabling agents to coordinate without communication. The decentralized nature of COLA provides scalability and robustness in partially observable environments, making it more adaptable to large-scale dynamic scenarios. In contrast, CTCE excels in tasks that require precise coordination and real-time alignment of connected and autonomous vehicles, where centralized control ensures consensus among agents. The role of immediate rewards in shaping decentralized consensus within MARL systems is discussed in (Fard and Selmic 2022). This decentralized approach enables agents to adapt

to their local environments, providing scalability and robustness in partially observable or dynamic settings. Although the proposed CTCE excels in centralized control scenarios with tightly coupled agents, the decentralized perspective in (Fard and Selmic 2022) offers insight into improving local decision-making and adaptability, highlighting a trade-off between global optimization and localized flexibility. The strengths of the proposed CTCE framework compared to the iterative neighbor and target Q-learning method in (Zhu et al. 2019) lie in its scalability, precision, and ability to manage highly coordinated multi-agent systems. In (Zhu et al. 2019), the focus is on distributed learning for consensus in leader-follower systems with fixed topology; it is highly dependent on neighbor and target networks for communication and control. In contrast, the proposed CTCE framework eliminates the dependency on explicit neighbor-based communication by centralizing policy training and execution, which mitigates non-stationarity and ensures globally optimized control policies. Additionally, CTCE's shared critic network and centralized approach make it better suited for dynamic and highly interconnected systems like connected autonomous vehicles, where precise coordination and real-time decision-making are critical. This centralized design also simplifies the implementation for large-scale systems, reducing potential issues of overestimation and instability that arise in iterative distributed algorithms.



(a) Velocity Consensus



(c) Trajectories of CAVs

Figure 4: Scenario3: Straight line path following for 4 CAVs.

Conclusion

This research presents a novel centralized training and centralized execution (CTCE) approach using DDPG with a single agent for consensus control of multiple CAVs following a leader-follower pattern. The proposed approach offers scalable and efficient coordination among multiple CAVs by addressing the key issues observed in MARL, including non-stationarity and computational complexity. The use of a shared actor and critic network for all CAVs reduces the computational overhead observed in MARL and makes the proposed approach suitable for real-time implementation. Reward shaping for the consensus problem ensures the consensus among multiple CAVs and collisions is avoided. Simulation results demonstrate the effectiveness of proposed methods in achieving consensus under different traffic scenarios, offering a scalable and efficient solution for modern autonomous vehicles. The limitation of the proposed research lies in the scalability of a centralized approach in dynamic driving scenarios. In future work, we will address societal and ethical implications by implementing robust data privacy and security measures, ensuring algorithm fairness through diverse scenario testing, and enhancing accountability with explainable AI methods.

References

Alonso-Mora, J.; Montijano, E.; Nägeli, T.; Hilliges, O.; Schwager, M.; and Rus, D. 2019. Distributed multi-robot formation control in dynamic environments. *Autonomous Robots*, 43: 1079–1100.

Amelin, K.; Amelina, N.; Ivanskiy, Y.; and Jiang, Y. 2018. Local voting protocol step-size choice for consensus achievement. *International Journal of Intelligent Engineering Informatics*, 6(1-2): 169–181.

Bevly, D.; Cao, X.; Gordon, M.; Ozbilgin, G.; Kari, D.; Nelson, B.; Woodruff, J.; Barth, M.; Murray, C.; Kurt, A.; et al. 2016. Lane change and merge maneuvers for connected and automated vehicles: A survey. *IEEE Transactions on Intelligent Vehicles*, 1(1): 105–120.

Chen, J.; Yuan, B.; and Tomizuka, M. 2019. Model-free deep reinforcement learning for urban autonomous driving. In *2019 IEEE intelligent transportation systems conference (ITSC)*, 2765–2771. IEEE.

el abidine Kherroubi, Z.; Aknine, S.; and Bacha, R. 2021. Novel decision-making strategy for connected and autonomous vehicles in highway on-ramp merging. *IEEE Transactions on Intelligent Transportation Systems*, 23(8): 12490–12502.

Fard, N. E.; and Selmic, R. 2022. Consensus of multi-agent reinforcement learning systems: The effect of immediate rewards. *Journal of Robotics and Control (JRC)*, 3(2): 115–127.

Ghraizi, D.; Talj, R.; and Francis, C. 2023. A Deep Reinforcement Learning Decision-Making Approach for Adaptive Cruise Control in Autonomous Vehicles. In *2023 21st International Conference on Advanced Robotics (ICAR)*, 71–78. IEEE.

Gronauer, S.; and Diepold, K. 2022. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2): 895–943.

Gu, S.; Lillicrap, T.; Sutskever, I.; and Levine, S. 2016. Continuous deep q-learning with model-based acceleration. In *International conference on machine learning*, 2829–2838. PMLR.

Kim, B.-Y.; and Ahn, H.-S. 2014. Distributed coordination and control for a freeway traffic network using consensus algorithms. *IEEE Systems Journal*, 10(1): 162–168.

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A. A.; Yogamani, S.; and Pérez, P. 2021. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6): 4909–4926.

Koung, D.; Fantoni, I.; Kermorgant, O.; and Belouaer, L. 2020. Consensus-based formation control and obstacle avoidance for nonholonomic multi-robot system. In *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 92–97. IEEE.

Li, S. E.; Zheng, Y.; Li, K.; and Wang, J. 2015. An overview of vehicular platoon control under the four-component framework. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, 286–291. IEEE.

Lin, Y.; McPhee, J.; and Azad, N. L. 2020a. Anti-jerk on-ramp merging using deep reinforcement learning. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 7–14. IEEE.

Lin, Y.; McPhee, J.; and Azad, N. L. 2020b. Comparison of deep reinforcement learning and model predictive control for adaptive cruise control. *IEEE Transactions on Intelligent Vehicles*, 6(2): 221–231.

Liu, Z.; Hu, J.; Song, T.; and Huang, Z. 2021. A methodology based on deep reinforcement learning to autonomous driving with double q-learning. In *2021 7th International Conference on Computer and Communications (ICCC)*, 1266–1271. IEEE.

Ma, J.; Zhang, M.; Ma, K.; Zhang, H.; and Geng, G. 2024. A decision-making of autonomous driving method based on DDPG with pretraining. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 09544070241227303.

Min, K.; Kim, H.; and Huh, K. 2018. Deep Q learning based high level driving policy determination. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 226–231. IEEE.

Oh, K.-K.; Park, M.-C.; and Ahn, H.-S. 2015. A survey of multi-agent formation control. *Automatica*, 53: 424–440.

Okuyama, T.; Gonsalves, T.; and Upadhay, J. 2018. Autonomous driving system based on deep q learnig. In *2018 International conference on intelligent autonomous systems (ICoIAS)*, 201–205. IEEE.

Ren, W.; Beard, R. W.; and Atkins, E. M. 2005. A survey of consensus problems in multi-agent coordination. In *Proceedings of the 2005, American Control Conference, 2005.*, 1859–1864. IEEE.

Shi, J.; Zhao, L.; Wang, X.; Zhao, W.; Hawbani, A.; and Huang, M. 2020. A novel deep Q-learning-based air-assisted vehicular caching scheme for safe autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 22(7): 4348–4358.

Song, D.; Zhu, B.; Zhao, J.; Han, J.; and Chen, Z. 2023. Personalized car-following control based on a hybrid of reinforcement learning and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 24(6): 6014–6029.

Talpaert, V.; Sobh, I.; Kiran, B. R.; Mannion, P.; Yogamani, S.; El-Sallab, A.; and Perez, P. 2019. Exploring applications of deep reinforcement learning for real-world autonomous driving systems. *arXiv preprint arXiv:1901.01536*.

Wang, P.; Li, H.; and Chan, C.-Y. 2019. Continuous control for automated lane change behavior based on deep deterministic policy gradient algorithm. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 1454–1460. IEEE.

Wei, H.; Liu, X.; Mashayekhy, L.; and Decker, K. 2019. Mixed-autonomy traffic control with proximal policy optimization. In *2019 IEEE Vehicular Networking Conference (VNC)*, 1–8. IEEE.

Wong, A.; Bäck, T.; Kononova, A. V.; and Plaat, A. 2023. Deep multiagent reinforcement learning: Challenges and directions. *Artificial Intelligence Review*, 56(6): 5023–5056.

Xu, Z.; Lyu, Y.; Pan, Q.; Hu, J.; Zhao, C.; and Liu, S. 2018. Multi-vehicle flocking control with deep deterministic policy gradient method. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)*, 306–311. IEEE.

Xu, Z.; Zhang, B.; Li, D.; Zhang, Z.; Zhou, G.; Chen, H.; and Fan, G. 2023. Consensus learning for cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11726–11734.

Ye, F.; Cheng, X.; Wang, P.; Chan, C.-Y.; and Zhang, J. 2020. Automated lane change strategy using proximal policy optimization-based deep reinforcement learning. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 1746–1752. IEEE.

Zhang, Y.; Sun, P.; Yin, Y.; Lin, L.; and Wang, X. 2018. Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning. In *2018 IEEE intelligent vehicles symposium (IV)*, 1251–1256. IEEE.

Zhao, J.; Zhao, Y.; Li, W.; and Zeng, C. 2024. End-to-End Autonomous Driving Algorithm Based on PPO and Its Implementation. In *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS)*, 1852–1858. IEEE.

Zhou, Y.; Fu, R.; and Wang, C. 2020. Learning the Car-following Behavior of Drivers Using Maximum Entropy Deep Inverse Reinforcement Learning. *Journal of advanced transportation*, 2020(1): 4752651.

Zhu, X.; Yuan, X.; Wang, Y.; and Sun, C. 2019. Reinforcement Learning Consensus Control for Discrete-Time Multi-Agent Systems. In *2019 Chinese Control Conference (CCC)*, 6178–6182. IEEE.

Zou, Q.; Xiong, K.; and Hou, Y. 2020. An end-to-end learning of driving strategies based on DDPG and imitation learning. In *2020 Chinese Control And Decision Conference (CCDC)*, 3190–3195. IEEE.