Multitimescale Control and Communications With Deep Reinforcement Learning—Part I: Communication-Aware Vehicle Control

Tong Liu, *Graduate Student Member, IEEE*, Lei Lei, *Senior Member, IEEE*, Kan Zheng, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—An intelligent decision-making system enabled by vehicle-to-everything (V2X) communications is essential to achieve safe and efficient autonomous driving (AD), where two types of decisions have to be made at different timescales, i.e., vehicle control and radio resource allocation (RRA) decisions. The interplay between RRA and vehicle control necessitates their collaborative design. In this two-part paper (Part I and Part II), taking platoon control (PC) as an example use case, we propose a joint optimization framework of multitimescale control and communications (MTCCs) based on deep reinforcement learning (DRL). In this article (Part I), we first decompose the problem into a communication-aware DRL-based PC subproblem and a control-aware DRL-based RRA subproblem. Then, we focus on the PC subproblem assuming an RRA policy is given, and propose the MTCC-PC algorithm to learn an efficient PC policy. To improve the PC performance under random observation delay, the PC state space is augmented with the observation delay and PC action history. Moreover, the reward function with respect to the augmented state is defined to construct an augmented state Markov decision process (MDP). It is proved that the optimal policy for the augmented state MDP is optimal for the original PC problem with observation delay. Different from most existing works on communication-aware control, the MTCC-PC algorithm is trained in a delayed environment generated by the fine-grained embedded simulation of cellular vehicle-to-everything communications rather than by a simple stochastic delay model. Finally, experiments are performed to compare the performance of MTCC-PC with those of the baseline DRL algorithms.**

***Index Terms*—Deep reinforcement learning (DRL), multitimescale decision making, platoon control (PC).**

I. Introduction

The CELLULAR vehicle-to-everything (C-V2X) system provides message delivery services for vehicular applications using fourth-generation (4G) and fifth-generation (5G) cellular connectivity [1]. Due to its ability to provide ubiquitous coverage, high reliability, and low latency, the C-V2X system is crucial for autonomous vehicles (AVs) [2]. Meanwhile, AVs are seen as a major driving use case for enhancing C-V2X communications in six-generation (6G) wireless system [3]. Designing vehicle control-oriented C-V2X system falls into the category of networked control systems (NCSs) research, where closed-loop control relies on data transmission in communication networks [4].

In contrast to the conventional network design, the performance of NCS is measured in terms of the efficiency in accomplishing a control task rather than the network performance metrics, such as throughput and delay [5], [6]. Compared with pure NCS, the ecosystem of C-V2X is more complex due to the co-existence of safety-critical vehicle control applications as well as nonsafety applications such as infotainment. Since the latter type of applications usually requires high throughput, an effective radio resource allocation (RRA) mechanism is indispensable for assigning the limited network resources to various applications, guaranteeing the safety and efficiency of vehicle control tasks while maximizing the throughput of nonsafety applications. For this purpose, RRA in C-V2X systems should be control aware, taking into account the control performance degradation due to the delay or packet loss in control-related information delivery.

Meanwhile, the amount of control performance degradation heavily depends on the robustness of vehicle controllers to nonideal communications. Conventional controllers of AVs are usually designed based on control theory under the assumption of zero-delay and zero-loss communications [7], [8]. In order to reduce the effects of communication impairments on control performance, vehicle controllers should be communication-aware, considering the statistical properties of random delay and packet loss in C-V2X communications.

A. Collaborative Design of Communications and Control

The interplay between RRA and vehicle control necessitates the collaborative design of communications and control functions. Existing works mainly tackle the problem

Manuscript received 23 September 2023; revised 16 November 2023; accepted 21 December 2023. Date of publication 29 December 2023; date of current version 25 April 2024. This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Discovery Grant. *(Corresponding author: Lei Lei.)*

Tong Liu is with the Intelligent Computing and Communication Lab, Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China.

Lei Lei is with the School of Engineering, University of Guelph, ON N1G 2W1, Canada (e-mail: leil@uoguelph.ca).

Kan Zheng is with the College of Electrical Engineering and Computer Sciences, Ningbo University, Ningbo 315211, China.

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2023.3348590

in two directions, i.e., control-aware communications and communication-aware control.

1) Control-Aware Communications: Control-aware or task-oriented communications aim at scheduling network resources to achieve satisfactory control performance. In order to characterize the significance of transmitted information in achieving the control target, two cross-layer performance metrics are often adopted to guide the optimization. The most widely used metric is Age of Information (AoI) [9], which captures the importance of information by measuring its timeliness attribute. Meanwhile, another metric, i.e., Value of Information (VoI), measures how much the recipient of the information can reduce the uncertainty of the stochastic processes related to decision making [10]. Since the co-design problem in this research direction is tackled from communications perspective, the considered controllers are normally quite simple and are designed based on conventional control theory and ideal communications assumption.

2) Communication-Aware Control: On the other hand, communication-aware or delay-aware control aims at designing controllers that are robust to communication imperfections. Examples are networked control that analyzes the tolerance of controllers to delay and packet loss using mathematical models; and event-triggered control that determines whether or not to sample and transmit system signals based on event or time. Since the co-design problem in this research direction is tackled from control perspective, nonideal communications are usually modeled as either constant delay or stochastic delay under coarse-grained surrogate communication models that are control agnostic [11], [12], [13], [14], [15], [16], [17].

B. Motivations

While most existing works of NCS study the co-design problem from either the communications or control perspective, it is our hypothesis that great benefits will arise from joint optimization in a unified perspective, where both components are designed using advanced technologies and are aware of the necessary details of the other components. Vehicle controllers are conventionally designed based on classical control theory, such as linear controller, $\mathcal{H}_\infty$ controller, sliding mode controller (SMC), etc. [8], [18], [19]; while RRA in C-V2X systems is traditionally studied using optimization theory. One of the main limitations of such approaches is that rigorous mathematical models are required, which are either inaccurate or unavailable for real-world problems; or it is computationally expensive to solve the models. Meanwhile, both vehicle control and RRA are Sequential stochastic decision problem (SSDP), where a sequence of decisions have to be made over a specific time horizon for a dynamic system whose states evolve in the face of uncertainty. As a promising approach to solve SSDP, deep reinforcement learning (DRL) has attracted considerable attention in recent years and has been adopted for vehicle control and RRA as an emerging trend. DRL inherits the model-free learning capability from reinforcement learning (RL), which can learn an optimal control policy directly from experience data by trial and error without knowledge of the underlying SSDP model. Moreover, it deals with the curse-of-dimensionality problem of RL by approximating the value functions and/or policy functions using deep neural networks (DNNs) [20], [21]. We believe that tackling both vehicle control and RRA problems under a unified DRL framework will better reveal the interdependency between the two components and thus facilitate the joint optimization task.

Since the frequency of vehicle control and sampling [normally between 0.01 and 0.1 second (s)] is often lower than that of RRA (e.g., 1 millisecond (ms) in C-V2X), the joint optimization of RRA and vehicle control is generally a multitimescale decision problem. The most straightforward approach is solving an integrated full-space model containing detailed vehicle control and RRA submodels. However, simultaneous derivation of vehicle control and RRA decisions at multitimescales yields a large-scale optimization problem, which is computationally infeasible even for modern machine learning techniques. Our main goal in this two-part paper (Part I and Part II) is to propose an efficient DRL-based approach for multitimescale control and communications (MTCCs) in the C-V2X system. To the best of our knowledge, this is the first paper that jointly optimizes multitimescale vehicle control and RRA decisions under a unified DRL framework.

As there are a variety of control tasks for AVs, we will focus on platoon control (PC) as an example use case. Meanwhile, the modular nature of the proposed approach enables its extension to other AV tasks. As a basic function of AVs, PC aims to determine the control inputs for following vehicles so that all vehicles move at the same speed while maintaining the desired distance between each pair of preceding and following vehicles [8]. Although PC can be performed without information exchange between vehicles based on the adaptive cruise control (ACC) functionality, the more advanced cooperative adaptive cruise control (CACC) extends ACC with vehicle-to-everything (V2X) communications and is capable of improving the PC performance by reducing the intervehicle distance while guaranteeing string stability [22].

C. Contributions

The main contributions of this two-part paper (Part I and Part II) are explained in the following.

1) *Unified DRL Framework for Multitimescale Control and Communications:* The time horizon is divided into control intervals, where each control interval consists of multiple communication intervals. Instead of employing the full-space approach with formidable computation complexity, we decompose the problem into two subproblems, i.e., a) communication-aware DRL-based PC and b) control-aware DRL-based RRA. We propose the MTCC-PC algorithm to learn the PC policy assuming an RRA policy is given, and the MTCC-RRA algorithm to learn the RRA policy assuming a PC policy is given. Finally, a sample- and computational-efficient approach is proposed to jointly learn the PC and RRA policies by training MTCC-PC and MTCC-RRA algorithms in an iterative process.

2) *Integrated DRL Model Capturing the Interplay Between RRA and PC:* We conceive a communication-aware DRL model for PC and a control-aware DRL model for RRA, both of which are integrated parts of the multitimescale decision framework. Specifically, we augment the PC state space with the observation delay, which serves as a bridge between the PC and RRA models. Moreover, we incorporate the advantage function of the PC model in the RRA reward function, which quantifies the amount of PC performance degradation caused by observation delay. Finally, we augment the state space of RRA with control input history for a more well-informed RRA policy. Since both PC and RRA are formulated into DRL models, it is much easier to fully consider the interplay between them. Specifically, the MTCC-PC algorithm is trained in a delayed environment generated by the fine-grained embedded simulation of C-V2X communications rather than by a simple stochastic delay model. Moreover, the RRA decisions in the MTCC-RRA algorithm are made based on the "VoI per control interval," which provides a finer-grained VoI compared with the existing VoI calculation methods.
3) *Efficient DRL Solution Addressing Random Observation Delay, Multiagent, and Sparse Reward Problems:* To improve the performance of PC in the face of random observation delay, we augment the PC state space with PC action history and prove the Markov property of the augmented state. Moreover, we define the reward function for the augmented state to construct an augmented state Markov decision process (MDP), and prove that the optimal policy for this MDP is also optimal for the original PC problem with observation delay. To deal with the multiagent problem in the MTCC-RRA algorithm, we apply the reward shaping technique to design an individual reward for each agent, so that they can deduce their own contributions in the global reward. Moreover, to tackle the sparse reward problem in RRA, we use a reward backpropagation prioritized experience replay (RBPER) technique [23] to improve training efficiency.

D. Organization of the Two-Part Paper

The organization of this two-part paper is shown in Fig. 1. First, the multitimescale decision system model is introduced in Part I. Then, in order to jointly optimize the multitimescale PC and RRA decisions using DRL, we assume that an RRA policy is given and study the communication-aware DRL-based PC in Part I. Next, we assume that a PC policy is available and focus on the control-aware DRL-based RRA in Part II. Finally, the joint learning approach that iteratively learns the PC and RRA policies is provided in Part II.

The remainder of this article (Part I) is organized as follows. Section II introduces the related work. The MTCC system model is presented in Section III. Subsequently, Sections IV and V introduce the communication-aware DRL model for PC and the corresponding DRL solution, i.e., MTCC-PC algorithm, respectively. Then, Section VI conducts experiments to demonstrate the effectiveness of the proposed algorithm. Finally, Section VII concludes this article.

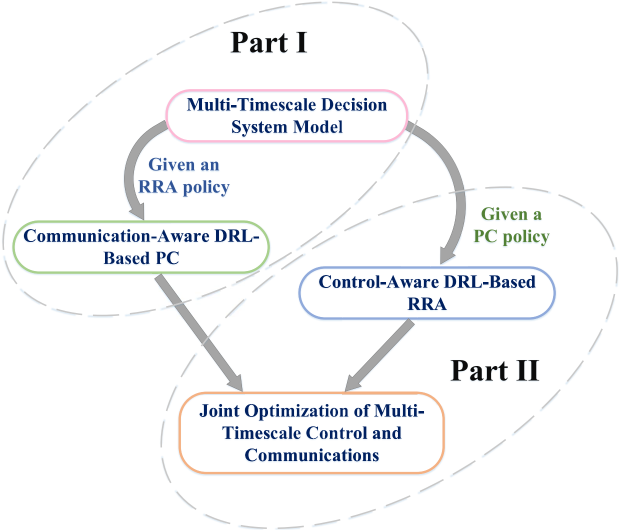

Part I
Multi-Timescale Decision System Model
Given an RRA policy
Given a PC policy
Communication-Aware DRL-Based PC
Control-Aware DRL-Based RRA
Part II
Joint Optimization of Multi-Timescale Control and Communications

Fig. 1. Organization of the two-part paper.

II. Related Work

A. RRA in C-V2X Systems

Existing works on RRA in C-V2X systems can be categorized into traditional methods and DRL methods. While traditional methods seek solutions to RRA based on classical optimization theory [24], [25], [26], [27], [28], [29], DRL methods have gained increasing attention in RRA research [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40] due to its success in learning decision-making policies in a variety of fields recently. There are two typical working modes in C-V2X, named vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V). It is generally considered that V2I links are mainly used to carry high bandwidth content, while V2V links are mostly used to deliver safety-critical messages [35], [36], [37], [41], [42], [43]. Therefore, the objective of RRA in most existing works is to efficiently share the frequency spectrum between V2I and V2V links, striking a tradeoff between maximizing V2I throughput and minimizing V2V delay.

Due to the scalability issue of the centralized solution, many works formulate RRA as multiagent RL (MARL) problems where each V2V agent can only observe its own local state. To solve MARL, the independent learner (IL) approach that directly applies single-agent RL algorithms has been adopted by a few works [30], [31], [32], [33], [34]. While IL is simple and scalable, the concurrent exploration of multiple agents can lead to nonstationarity issues, especially for DRL algorithms with experience replay. To deal with this challenge, the fingerprint-based technique is adopted in [35] and hysteretic Q-learning and concurrent experience replay trajectory approaches are leveraged in [37]. In [38], the mean-field game theory is used to enhance scalability and reduce the complexity of MARL solutions. However, another important issue in MARL, namely, the credit assignment problem, is not considered in the above works.

While most RRA algorithms in C-V2X are agnostic to vehicle control tasks, a few algorithms are specifically designed to support PC applications. The fingerprint-based deep Q networks (DQNs) algorithm similar to [35] is used in [36] for platoon-based C-V2X system. AoI-aware RRA algorithms are proposed in [39] based on the multiagent deep deterministic policy gradient (MADDPG) algorithm. Xu et al. [40] formulated a multiobjective RRA problem, which is divided into a set of scalar optimization subproblems that are modeled as the partially observable stochastic game (P-OSG) and solved by the dual-clip proximal policy optimization (CD-PPO) algorithm. By modeling the network traffic based on message delivery characteristics of PC, the above works have designed efficient RRA algorithms that can better support PC applications. However, the optimization objectives are either delay, AoI, or transmission success ratio, which fail to capture the extent to which the PC performance will be degraded by receiving stale information.

B. Delay-Aware PC and DRL-Based PC

The existing works on delay-aware PC mainly focus on designing platoon controllers that are robust to communication delay, and deriving the upper bound of communication delay satisfying the internal and string stability for the controllers [11], [12], [13], [14], [15], [16]. These works have achieved impressive results on improving PC tolerance to communication delay. However, the delay models are abstract and simple, which cannot accurately reflect the delay distribution induced by the advanced RRA mechanisms in C-V2X communications. Moreover, the platoon controllers are designed based on classical control theory, and thus have limited capability in dealing with the uncertainty and randomness of the environment.

In order to better cope with the uncertain driving environment, nonlinear vehicle dynamics, and real-time application requirement, some recent works study DRL-based PC [44], [45], [46], [47], where deep deterministic policy gradient (DDPG) is the most widely used algorithm. However, these works do not consider communication delay when the vehicles share the driving information.

In the field of theoretical research on RL, there have been some works on how the agents make decisions when delays occur in one or more forms, including observation delay, action delay, and reward delay. The pioneering work of [48] considers constant delay scenario, and reformulates the decision process with delays into an augmented state MDP without delays, where the action history is included in the augmented state. The history horizon is from the time step when the delayed observation was generated to one-time step before the current state. However, since the assumption of constant delay is usually unrealistic, recent works in this area mostly focus on the random delay scenario.

To solve the problem of uncertain augmented state dimension when the random delay occurs, [49] and [50] assume that the MDP freezes from the perspective of the agent, i.e., the agent does not take any new actions till the most recent state becomes observable. This may not be possible in practice when the agent must take new actions to interact with the environment at each time step. To deal with random reward delay, the authors augment the state with the time step at which the last observed delayed state was first observed. Moreover, they show that action delay and observation delay are equivalent in the sense that their respective decision processes with delay are both reducible to the MDP with the augmented state. In [51], the delayed time steps after observing the last delayed state is included in the augmented state. However, the authors do not mention how to solve the problem of the uncertain augmented state dimension. Moreover, it is assumed that multiple data packets cannot arrive simultaneously at the same time step, which is not the case in C-V2X communication. Different from the above works, [52] augments the state with the action history from maximum delay to the previous time step, which solves the uncertain augmented state dimension problem. The observation delay at each time step is also included in the augmented state. In our work, the augmented state is in a similar form to that defined in [52]. Moreover, we provide rigorous proofs of the Markov property of the augmented state, and of the functional equivalence of the augmented state MDP with the original decision process with delay, which are lacking in [52].

C. Joint Optimization of RRA and PC

While the RRA and PC problems are studied separately in the above works, some recent literature tackles the co-design of them. Most works fall into the class of control-aware communications or communication-aware control, emphasizing the novel design of either communications or control while considering the requirements or constraints posed by existing control or communications mechanisms.

For control-aware communications, [53] derived sufficient conditions to meet platoon stability of a sampled-data feedback controller, which is used for parameter design of event-triggered communication mechanisms. In [29], the communication delay constraints that guarantee plant stability and string stability are first derived for a nonlinear controller, and the obtained delay constraints are used to guide RRA.

For communication-aware control, a nonlinear consensus-based platoon controller was proposed in [17] based on the probability of successful communication inferred from the carrier sense multiple access with collision avoidance (CSMA/CA) mechanism. Zeng et al. [54] first derived an approximate expression for the probability that the wireless system meets the control system's delay needs, and then optimizes the control parameters of the optimal velocity model (OVM) to maximize the probability. In [55], PC is modeled as a consensus problem, where the parameters of a linear controller are dynamically adjusted according to different information typologies (IFTs) and delays.

Finally, several research works consider the novel design of both PC and communication mechanisms. Hong et al. [56] designed a modified distributed model predictive controller (DMPC), which takes into account the set of vehicles whose

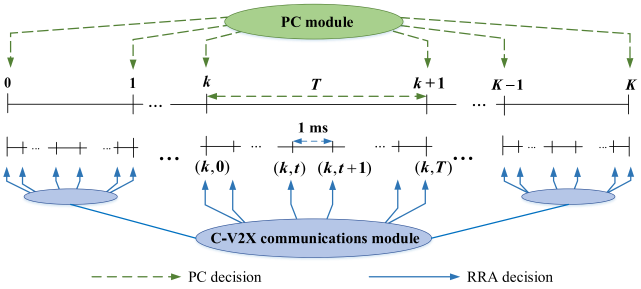


Fig. 2. Multitimescale decision-making framework.

messages are successfully received. Moreover, a communications approach to select relay vehicles to forward the information of the leading vehicle is proposed, which aims at maximizing the minimal average signal-to-noise ratio (SNR) among the vehicles in the platoon. In [28], the PC and RRA are jointly optimized in order to minimize the tracking error while guaranteeing the minimum SNR requirements of V2V communications and string stability of the platoon. Since the optimization problem is nondeterministic polynomial hard (NP-hard), it is decomposed into separate RRA and PC problems in two stages. The bipartite graph matching method is first used to approximate the subframe allocation scheme, and then the parameters of a linear controller are optimized. Both [28] and [56] consider nonideal communications in terms of reliability instead of delay. Instead of focusing on the impact of communication impairments on PC, [57] contemplated the interplay between PC and communications from a different perspective and attends to the effect of PC inputs on the reliability of V2I communications through vehicle mobility behaviors. A joint optimization scheme based on MPC was proposed for RRA and PC, with the aim of maximizing the communication reliability of V2I and minimizing the traffic oscillation of PC. Our work differs from the above research in considering that the control and communications decisions are usually made at different time scales. Moreover, the co-design is performed under a unified DRL framework, which does not suffer from model inaccuracy or high computational complexity as in conventional control theory, optimization theory, or MPC.

III. System Model

We consider a platoon with a number of $N > 2$ vehicles, i.e., $\mathcal{V} = \{0, 1, \ldots, N-1\}$. All vehicles communicate with one another using C-V2X communications. The important symbols used in this article are summarized in Table I.

A. Multitimescale Decision-Making Framework

As shown in Fig. 2, the PC problem is considered within a finite time horizon, which is discretized into K equal-length control intervals indexed by $k \in \mathcal{K} = \{0, 1, \ldots, K-1\}$. The duration of each control interval is T milliseconds (ms). The vehicle control input (commanded acceleration) $a_{i,k}^{\mathrm{CL}}$ for any vehicle $i \in \mathcal{V}$ is applied at time kT and holds constant within time period $[kT, (k+1)T)$. In the remainder of this article, we will use $x_k := x(kT)$ to represent any variable x at the control interval k.

At each control interval k, the PC module of each following vehicle (i.e., follower) $i \in \mathcal{V}\backslash\{0\}$ determines the vehicle control input $a_{i,k}^{\mathrm{CL}}$ based on the observations of the system state. The vehicle driving status is sampled at time kT (i.e., the sampling period is T ms). Specifically, the position $p_{i,k}$, velocity $v_{i,k}$, and acceleration $\mathrm{acc}_{i,k}$ of follower i are measured locally by the fusion of inertial navigation and global positioning system (GPS). Here, $p_{i,k}$ represents the 1-D position of the center of the front bumper of vehicle i at control interval k. Additionally, each follower i can obtain the driving status $p_{j,k}$, $v_{j,k}$, and $\mathrm{acc}_{j,k}$ of the other vehicles $j \in \mathcal{V}\backslash\{i\}$ via V2V communications.

We adopt the predecessors following (PF) IFT [29], [47], [58], where the collaborative adaptive message (CAM) $c_{i-1,k} = \{p_{i-1,k}, v_{i-1,k}, \mathrm{acc}_{i-1,k}\}$ of the preceding vehicle (i.e., predecessor) $i-1 \in \mathcal{V}\backslash\{N-1\}$ are transmitted to the follower i. For this purpose, each control interval k is further divided into T communication intervals indexed by $t \in \mathcal{T} = \{0, 1, \ldots, T-1\}$ on a faster timescale. Each communication interval has a length of 1 ms corresponding to the subframe duration in C-V2X communications. The vehicles transmit CAM at time $kT + t, k \in \mathcal{K}, t \in \mathcal{T}$, where the corresponding communication interval is represented as (k, t). Dynamic scheduling is considered, where the C-V2X communication module makes RRA decisions at each communication interval (k, t). In the integrated model, temporal integrity is maintained by $(k, T) = (k+1, 0)$. In the remainder of this article, we will use $x_{(k,t)} := x(kT + t)$ to represent any variable x at communication interval (k, t).

Since the PC decisions are made with a coarse time grid of every T ms, while the RRA decisions are made with a fine time grid of every 1 ms, we have a multitimescale decision-making problem.

B. Platoon Control Module

Each vehicle $i \in \mathcal{V}$ obeys the dynamics model approximated by a first-order system. The state space model in discrete time is derived on the basis of forward Euler discretization

$$p_{i,k+1} = p_{i,k} + Tv_{i,k} \tag{1}$$

$$v_{i,k+1} = v_{i,k} + Tacc_{i,k} \tag{2}$$

$$acc_{i,k+1} = \left(1 - \frac{T}{\tau_i}\right)\mathrm{acc}_{i,k} + \frac{T}{\tau_i}a_{i,k}^{\mathrm{CL}} \tag{3}$$

where τ_i is a time constant representing driveline dynamics. The first-order-system approximation in (3) is widely used in platoon controller design, which is obtained by first formulating a nonlinear model and then applying the exact feedback linearization technique to convert the nonlinear model to a linear one [7], [59], [60]. In order to ensure driving safety and comfort, the following constraints are applied:

$$\mathrm{acc}_{\min} \le \mathrm{acc}_{i,k} \le \mathrm{acc}_{\max}, \; a_{\min}^{\mathrm{CL}} \le a_{i,k}^{\mathrm{CL}} \le a_{\max}^{\mathrm{CL}} \tag{4}$$

where $\mathrm{acc}_{\min}$ and $\mathrm{acc}_{\max}$ are the acceleration limits, while $a_{\min}^{\mathrm{CL}}$ and $a_{\max}^{\mathrm{CL}}$ are the control input limits.

TABLE I
SUMMARY OF IMPORTANT SYMBOLS USED

Category	Symbol	Definition
Control	T	The control interval
	N	The number of vehicles in a platoon
	$p_{i,k}$	The one-dimensional position of vehicle i at control interval k
	$v_{i,k}$	The velocity of vehicle i at control interval k
	$acc_{i,k}$	The acceleration f vehicle i at control interval k
	$e_{pi,k}$	The gap-keeping error of vehicle i at control interval k
	$e_{vi,k}$	The velocity error of vehicle i at control interval k
	$j_{i,k}$	The jerk of vehicle i at control interval k
	$\tau_{i,k}$	The observation delay of vehicle i at control interval k
	$S_{i,k}^{\mathrm{CL}}$	The PC state of vehicle i at control interval k
	$a_{i,k}^{\mathrm{CL}}$	The PC action of vehicle i at control interval k
	$R_{i,k}^{\mathrm{CL}}$	The PC reward of vehicle i at control interval k
	J_i^{CL}	The expected cumulative reward of PC agent i
	Q_i^{CL}	The Q-value of PC agent i
	V_i^{CL}	The value function of PC agent i
	π_i^{CL}	The policy of PC agent i
	$A_{\pi_i^{\mathrm{CL}}}$	The advantage function of policy π_i^{CL}
Communication	M	The number of V2I links
	W	The bandwidth of sub-channel
	N_c	The constant CAM size
	N_Q	The buffer capacity in the number of CAM
	$\gamma_{m,(k,t)}$	The SINR of the V2I link m over the sub-channel m at communication interval (k,t)
	$\gamma_{i,m,(k,t)}$	The SINR of the V2V link i over the sub-channel m at communication interval (k,t)
	$I_{i,m,(k,t)}$	The total interference power received by V2V link i over sub-channel m
	P_m^{I}	The transmit power of V2I link m over the sub-channel m
	$P_{i,m,(k,t)}^{\mathrm{V}}$	The transmit power of V2V link i over the sub-channel m at communication interval (k,t)
	$G_{m,(k,t)}$	The channel gain of the V2I link m
	$G_{i,B,m,(k,t)}$	The interference channel gain from V2V link i transmitter to V2I link m receiver
	$G_{i,m,(k,t)}$	The channel gain of the V2V link i over the sub-channel m
	$G_{B,i,m,(k,t)}$	The interference channel gain from V2I link m transmitter to V2V link i receiver
	$G_{j,i,m,(k,t)}$	The interference channel gain from the V2V link j transmitter to the V2V link i receiver
	$\theta_{i,m,(k,t)}$	The binary allocation indicator indicating whether V2V link i occupies sub-channel m at communication interval (k,t) or not, $\theta_{i,m,(k,t)} \in \{0,1\}$
	$r_{i,(k,t)}^{\mathrm{CAM}}$	The transmission rate of V2V link i in terms of CAM at communication interval (k,t)
	$r_{m,(k,t)}$	The instantaneous data rate of V2I link m at communication interval (k,t)
	$q_{i,(k,t)}^{\mathrm{CAM}}$	The queue length of vehicle i in the number of CAM at communication interval (k,t)
	$S_{i,(k,t)}^{\mathrm{CM}}$	The RRA state of vehicle i at communication interval (k,t)
	$a_{i,(k,t)}^{\mathrm{CM}}$	The RRA action of vehicle i at communication interval (k,t)
	$R_{(k,t)}^{\mathrm{CM}}$	The RRA reward of vehicle i at communication interval (k,t)
	$R_{\mathrm{I},(k,t)}$	The RRA reward component related to V2I throughput at communication interval (k,t)
	Q_i^{CM}	The Q-value of RRA agent i

The headway of follower i at control interval k, i.e., bumper-to-bumper distance between follower i and its predecessor $i-1$, is denoted by $d_{i,k}$ with

$$d_{i,k} = p_{i-1,k} - p_{i,k} - L_{i-1} \tag{5}$$

where L_{i-1} is the body length of vehicle $i-1$.

We adopt the constant time-headway policy (CTHP), where follower i aims to maintain the desired headway

$$d_{r,i,k} = r_i + h_i v_{i,k} \tag{6}$$

where r_i is a standstill distance for the safety of follower i and h_i is a constant time gap of follower i, which represents the time that it takes for follower i to bridge the distance in between the vehicles i and $i-1$ when continuing to drive with a constant velocity.

The tracking errors, i.e., gap-keeping error $e_{pi,k}$ and velocity error $e_{vi,k}$ of follower i are defined as follows:

$$e_{pi,k} = d_{i,k} - d_{r,i,k}, \;\; e_{vi,k} = v_{i-1,k} - v_{i,k}. \tag{7}$$

C. C-V2X Communications Module

As shown in Fig. 3, we consider a typical urban C-V2X network, where V2V links coexist with V2I links. A V2I link connects a vehicle to the base station (BS) and is used for high-throughput services. According to the PF IFT, a V2V link connects a pair of predecessor and follower for periodic transmission of CAM. The link between vehicle i and $i+1$ is denoted as V2V link i. The set of $N-1$ V2V links can thus be represented by $\mathcal{V}\backslash\{N-1\}$.

We consider that there are M V2I links (uplink considered). Without loss of generality, we assume that every V2I link

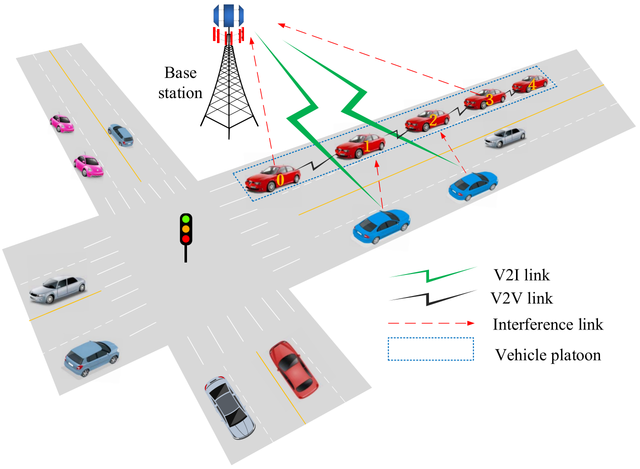


Fig. 3. Illustration of C-V2X network for the urban environment.

$m \in \mathcal{M} = \{0, \dots, M-1\}$ is preassigned subchannel m with constant transmit power $P^{\rm I}_m$ [35]. In order to enhance spectrum utilization, one or more V2V links can reuse the subchannels of the V2I links for CAM transmission. We use the binary allocation indicator $\theta_{i,m,(k,t)} \in \{0, 1\}$ to indicate whether V2V link i occupies subchannel m at communication interval (k, t) or not. Moreover, we consider that each V2V link i occupies at most one subchannel, i.e., $\sum_{m=0}^{M-1} \theta_{i,m,(k,t)} \le 1$.

At the beginning of each control interval k, each vehicle i samples its driving status to form the CAM and buffers the CAM in a queue before transmitting the data to the following vehicle $i + 1$. In each communication interval (k, t), each vehicle i transmits the data in its queue according to the local RRA decisions.

1) Channel Gain: The instantaneous channel gain of V2V link i over subchannel m (occupied by V2I link m) at communication interval (k, t) is denoted by $G_{i,m,(k,t)}$. Similarly, let $G_{m,(k,t)}$ denote the channel gain of the V2I link m; $G_{i,B,m,(k,t)}$ the interference channel gain from V2V link i transmitter to V2I link m receiver; $G_{B,i,m,(k,t)}$ the interference channel gain from V2I link m transmitter to V2V link i receiver; and $G_{j,i,m,(k,t)}$ the interference channel gain from the V2V link j transmitter to the V2V link i receiver over the subchannel m.

2) Signal-to-Interference-Plus-Noise Ratio (SINR): The SINR $\gamma_{m,(k,t)}$ of V2I link m and the SINR $\gamma_{i,m,(k,t)}$ of V2V link i on subchannel m at communication interval (k, t) are derived by

$$\gamma_{m,(k,t)} = \frac{P^{\rm I}_m G_{m,(k,t)}}{\sigma^2 + \sum_{i \in \mathcal{V}\setminus\{N-1\}} \theta_{i,m,(k,t)} P^{\rm V}_{i,m,(k,t)} G_{i,B,m,(k,t)}} \quad (8)$$

and

$$\gamma_{i,m,(k,t)} = \frac{P^{\rm V}_{i,m,(k,t)} G_{i,m,(k,t)}}{\sigma^2 + I_{i,m,(k,t)}} \quad (9)$$

respectively, where $P^{\rm V}_{i,m,(k,t)}$ is the transmit power of V2V link i over the subchannel m at communication interval (k, t). σ^2 is the power of channel noise which satisfies the independent Gaussian distribution with a zero mean value. $I_{i,m,(k,t)}$ is the total interference power received by V2V link i over subchannel m, where

$$\begin{aligned} I_{i,m,(k,t)} = {} & P^{\rm I}_m G_{B,i,m,(k,t)} \\ & + \sum_{j \in \mathcal{V}\setminus\{i,N-1\}} \theta_{j,m,(k,t)} P^{\rm V}_{j,m,(k,t)} G_{j,i,m,(k,t)}. \end{aligned}$$

3) Instantaneous Data Rate: The instantaneous data rates $r_{m,(k,t)}$ and $r_{i,(k,t)}$ of V2I link m and V2V link i at communication interval (k, t) are, respectively, derived as follows:

$$r_{m,(k,t)} = W \log_2\left(1 + \gamma_{m,(k,t)}\right) \quad (10)$$

and

$$r_{i,(k,t)} = \sum_{m=0}^{M-1} \theta_{i,m,(k,t)} W \log_2\left(1 + \gamma_{i,m,(k,t)}\right) \quad (11)$$

where W is the bandwidth of a subchannel.

Let $r^{\rm CAM}_{i,(k,t)}$ denote the transmission rate of V2V link i in terms of CAM at communication interval (k, t), which is given by

$$r^{\rm CAM}_{i,(k,t)} = \frac{r_{i,(k,t)}}{N_c} \quad (12)$$

where N_c is the constant CAM size.

4) Queuing Dynamic: Each vehicle $i \in \mathcal{V}\setminus\{N-1\}$ except for the last vehicle $N-1$ has a buffer to store its CAM, where the buffer capacity is N_Q in the number of CAM. Let $q^{\rm CAM}_{i,(k,t)}$ denote the queue length of vehicle i in the number of CAM at communication interval (k, t). If the queue length $q^{\rm CAM}_{i,(k,t)}$ reaches the buffer capacity N_Q, the subsequent arriving data will be dropped. The queue process evolves as follows:

$$q^{\rm CAM}_{i,(k,t+1)} = \begin{cases} \min\Big[N_Q, \max\Big[0, q^{\rm CAM}_{i,(k,t)} - 10^{-3} & \\ \quad \times r^{\rm CAM}_{i,(k,t)}\Big] + 1\Big], & \text{if} \quad t = 0 \\ \max\Big[0, q^{\rm CAM}_{i,(k,t)} - 10^{-3} \times r^{\rm CAM}_{i,(k,t)}\Big], & \text{otherwise.} \end{cases} \quad (13)$$

At each communication interval (k, t), the queue length $q^{\rm CAM}_{i,(k,t+1)}$ is decreased by $10^{-3} \times r^{\rm CAM}_{i,(k,t)}$, which is the number of CAM transmitted during the communication interval. Meanwhile, at every communication interval $(k, 0)$, the queue length $q^{\rm CAM}_{i,(k,t+1)}$ is increased by 1, since vehicle i samples the driving status for control interval k and buffers the generated CAM. In addition, the CAM that is not fully transmitted during control interval k will continue to be transmitted in the next control interval $k + 1$.

D. Correlation Between Platoon Control Decisions and Radio Resource Allocation Decisions

In our system model, each vehicle $i \in \mathcal{V}\setminus\{0\}$ makes PC decisions on the control input $a^{\rm CL}_{i,k}$ at every control interval $k \in \mathcal{K}$. Moreover, each vehicle $i \in \mathcal{V}\setminus\{N-1\}$ makes RRA decisions on subchannel allocation $\{\theta_{i,m,(k,t)}\}_{m\in\mathcal{M}}$ and transmit power $\{P^{\rm V}_{i,m,(k,t)}\}_{m\in\mathcal{M}}$ at every communication interval (k, t), where $k \in \mathcal{K}$ and $t \in \mathcal{T}$. It is important to note that the PC and RRA decisions are closely related to each other.

At the beginning of each control interval k, each follower $i \in \mathcal{V}\setminus\{0\}$ determines $a^{\rm CL}_{i,k}$ based on its own driving status as well

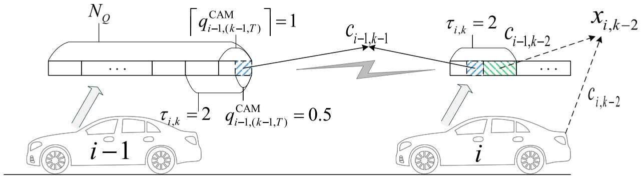

N_Q
$\left\lceil q^{\text{CAM}}_{i-1,(k-1,T)} \right\rceil = 1$
$c_{i-1,k-1}$
$\tau_{i,k} = 2$
$c_{i-1,k-2}$
$x_{i,k-2}$
$c_{i,k-2}$
$\tau_{i,k} = 2$
$q^{\text{CAM}}_{i-1,(k-1,T)} = 0.5$
$i-1$
i

Fig. 4. Schematic diagram of the relationship between observation delay $\tau_{i,k}$ and queue length $q^{\text{CAM}}_{i-1,(k-1,T)}$.

as the driving status received from its predecessor $i-1$. Let $\tau_{i,k}$ be the observation delay of follower i at control interval k. Thus, $c_{i-1,k-\tau_{i,k}} = \{p_{i-1,k-\tau_{i,k}}, v_{i-1,k-\tau_{i,k}}, \text{acc}_{i-1,k-\tau_{i,k}}\}$ is the most recent available delayed CAM at follower i, which correspond to the position, velocity, and acceleration sampled at predecessor $i-1$ in control interval $k-\tau_{i,k}$. Therefore, the observed driving status of vehicle i is defined as follows:

$$x_{i,k-\tau_{i,k}} = \{e_{pi,k-\tau_{i,k}}, e_{vi,k-\tau_{i,k}}, \text{acc}_{i,k-\tau_{i,k}}, \text{acc}_{i-1,k-\tau_{i,k}}\} \quad (14)$$

where $e_{pi,k-\tau_{i,k}} = p_{i-1,k-\tau_{i,k}} - p_{i,k-\tau_{i,k}} - L_{i-1} - d_{r,i,k}$, $e_{vi,k-\tau_{i,k}} = v_{i-1,k-\tau_{i,k}} - v_{i,k-\tau_{i,k}}$. Note that although follower i has the undelayed observation on its own $p_{i,k}$, $v_{i,k}$ and $\text{acc}_{i,k}$, the observation $x_{i,k-\tau_{i,k}}$ is defined based on $p_{i,k-\tau_{i,k}}$, $v_{i,k-\tau_{i,k}}$, and $\text{acc}_{i,k-\tau_{i,k}}$ to be aligned with the delayed information from its predecessor $i-1$.

The observation delay $\tau_{i,k}$ depends on the transmission delay of CAM over V2V link $i-1$, which can be derived from $q^{\text{CAM}}_{i-1,(k-1,T)}$ or $q^{\text{CAM}}_{i-1,(k,0)}$ as follows:

$$\tau_{i,k} = \left\lceil q^{\text{CAM}}_{i-1,(k-1,T)} \right\rceil + 1 = \left\lceil q^{\text{CAM}}_{i-1,(k,0)} \right\rceil + 1. \quad (15)$$

An example is given in Fig. 4 to illustrate the relationship between the queue length and observation delay. Assume the queue length of predecessor $i-1$ at the end of control interval $k-1$ is 0.5 CAM of $c_{i,k-1}$, i.e., $q^{\text{CAM}}_{i-1,(k-1,T)} = 0.5$. This means that the CAM $c_{i,k-1}$ generated by the predecessor $i-1$ at control interval $k-1$ is not fully received by the follower i. At control interval k, since the follower i cannot interpret an incomplete CAM of $c_{i,k-1}$, it has to make decisions based on the last fully received CAM, i.e., the CAM $c_{i,k-2}$ generated by predecessor $i-1$ at control interval $k-2$. Therefore, the observation delay is $\tau_{i,k} = \lceil 0.5 \rceil + 1 = 2$. The delayed observation for follower i at control interval k is $x_{i,k-2}$.

Please note that (15) no longer holds when the queue length $q^{\text{CAM}}_{i,(k,t)}$ reaches the buffer capacity N_Q and the subsequent arriving data are dropped. In this article, we consider the case when N_Q is large enough and the packet dropping probability is negligible. We leave the consideration of dropped packets to future work.

In this article, we consider all the generated CAMs are buffered and transmitted sequentially. Another popular buffer management strategy is to replace any old CAM that has not yet been fully delivered in the previous control interval with the newly generated CAM at the beginning of each control interval. We adopt the current strategy since the probability of successfully transmitting a partially transmitted CAM is larger than that of transmitting a completely new CAM due to the smaller amount of data left to be transmitted. However, our proposed MTCC framework can be applied with other buffer managements strategies with the change of (13) and (15).

The correlation between PC decisions and RRA decisions can be analyzed from the following two aspects.

Impact of RRA on PC: The PC decisions are made with the target of optimizing the PC performance, which is affected by the observation delay $\tau_{i,k}$. Meanwhile, $\tau_{i,k}$ is determined by $q^{\text{CAM}}_{i-1,(k-1,T)}$ according to (15), which in turn depends on the RRA decisions $\{\theta_{i,m,(k',t)}\}_{k'<k,m\in\mathcal{M}}$ and $\{P^{\text{V}}_{i,m,(k',t)}\}_{k'<k,m\in\mathcal{M}}$ according to (8)–(13). Different RRA decisions lead to diverse stationary distributions of observation delay. Therefore, the PC decisions should be optimized under the stochastic delay distributions stemmed from the de facto RRA decisions.

Impact of PC on RRA: The RRA decisions are made with the targets of: 1) maximizing the V2I throughput and 2) minimizing the PC performance degradation due to delayed observation. Unfortunately, these two targets are contradictory with each other and an optimal tradeoff should be struck. The tradeoff heavily depends on the impact of observation delay $\tau_{i,k}$ on PC performance, which in turn is affected by the PC decisions. Better PC decisions lead to higher tolerance to observation delay, which means that larger V2I throughput can be supported with negligible penalty to PC performance. Therefore, the RRA decisions should be optimized with awareness of the impact of observation delay on PC performance under the de facto PC decisions.

IV. Communication-Aware DRL-Based Platoon Control

We assume that the RRA policy π^{CM} is available and focus on learning the PC policy $\pi^{\text{CL}}_i, i \in \mathcal{V}\backslash\{0\}$. The PC problem with observation delay is essentially a Random Delay Decentralized Partially Observable MDP. Each follower $i \in \mathcal{V}\backslash\{0\}$ is a PC agent, which makes a local and delayed observation at each control interval k, and decides on its local actions to maximize its expected cumulative *individual* reward. The cumulative reward is normally referred to as the return in the RL literature.

A. PC State

The state for each PC agent i at control interval k is defined as follows:

$$S^{\text{CL}}_{i,k} = \left\{ x_{i,k-\tau_{i,k}}, \left\{ a^{\text{CL}}_{i,k'} \right\}_{k'=k-\tau_{\max}}^{k-1}, \tau_{i,k} \right\} \quad (16)$$

where $\tau_{\max}$ is the maximum observation delay that depends on the maximum queue length N_Q, i.e., $\tau_{\max} \geq \tau_{i,k}$. The delayed observation of driving status $x_{i,k-\tau_{i,k}}$ is augmented with the last $\tau_{\max}$ actions $\{a^{\text{CL}}_{i,k'}\}_{k'=k-\tau_{\max}}^{k-1}$ of PC agent i. Moreover, the observation delay $\tau_{i,k}$ is included since it provides useful information to the control agent on how old an observation is. More importantly, $\tau_{i,k}$ serves as a bridge between the control and communication modules. For PC agent i to be aware of the observation delay $\tau_{i,k}$, the predecessor $i-1$ needs to share its queue length $q^{\text{CAM}}_{i-1,(k,0)}$ at the beginning of each control interval k via control signaling. This is possible in the C-V2X system, since the queue length can be contained in sidelink control information (SCI) transmitted in the physical sidelink shared channel (PSSCH).

B. PC Action

The control input, $a^{\mathrm{CL}}_{i,k} \in [a^{\mathrm{CL}}_{\min}, a^{\mathrm{CL}}_{\max}]$ of each PC agent i is regarded as its PC action at control interval k.

C. PC Reward Function

The objective for each PC agent i is to minimize its own gap-keeping error $e_{pi,k}$ and velocity error $e_{vi,k}$ while penalizing control input $a^{\mathrm{CL}}_{i,k}$ and the jerk to reduce the fuel consumption and improve the driving comfort, respectively. Note that the jerk is the change rate in acceleration, which is given by

$$j_{i,k} = \frac{\mathrm{acc}_{i,k+1} - \mathrm{acc}_{i,k}}{T} = -\frac{1}{\tau_i}\mathrm{acc}_{i,k} + \frac{1}{\tau_i} a^{\mathrm{CL}}_{i,k} \tag{17}$$

where the second equality is due to the forward Euler discretization of (3).

The individual reward for each PC agent i is given by

$$\begin{aligned} & R^{\mathrm{CL}}_{i,k}\left(x_{i,k}, a^{\mathrm{CL}}_{i,k}\right) \\ & = -\left\{\left|\frac{e_{pi,k}}{\hat{e}_{p,\max}}\right| + \alpha_1\left|\frac{e_{vi,k}}{\hat{e}_{v,\max}}\right| + \alpha_2\left|\frac{a^{\mathrm{CL}}_{i,k}}{a^{\mathrm{CL}}_{\max}}\right| + \alpha_3\left|\frac{j_{i,k}}{2\mathrm{acc}_{\max}/T}\right|\right\} \end{aligned} \tag{18}$$

where $\hat{e}_{p,\max}$ and $\hat{e}_{v,\max}$ are the nominal maximum control errors such that it is larger than most possible control errors. α_1, α_2, and α_3 are the positive weights and can be adjusted to determine the relative importance of minimizing the gap-keeping error, the velocity error, the control input, and the jerk.

The expected return J^{CL}_i of PC agent i under policy π^{CL}_i can be expressed as follows:

$$J^{\mathrm{CL}}_i = \mathrm{E}_{\pi^{\mathrm{CM}}}\mathrm{E}_{\pi^{\mathrm{CL}}_i}\left[\sum_{k=0}^{K-1}\gamma^k R^{\mathrm{CL}}_{i,k}\right],\ 0 \le \gamma \le 1 \tag{19}$$

where γ is the PC reward discount factor and π^{CM} is the de facto communication policy.

Remark 1 (Impact of Communications Policy on Control Performance): In (19), the expectation is taken with respect to the probability distribution of the state-action trajectories when the PC agent i follows policy π^{CL}_i and the RRA policy is π^{CM}. The RRA policy π^{CM} affects the PC performance since it determines the observation delay $\tau_{i,k}$, which is a part of the augmented state $S^{\mathrm{CL}}_{i,k}$. In other words, π^{CM} has an important influence on the state transition probabilities of the RD-Dec-POMDP model.

The objective of the PC problem is for each PC agent i to find the optimal policy $\pi^{\mathrm{CL}*}_i$ under delayed observation that maximizes its individual expected return J^{CL}_i, i.e.,

$$\pi^{\mathrm{CL}*}_i = \arg\max_{\pi^{\mathrm{CL}*}_i} J^{\mathrm{CL}}_i,\ \ \forall i \in \mathcal{V}\backslash\{0\}. \tag{20}$$

V. DRL Solution

The DDPG algorithm [61] is utilized to solve the PC problem, which is the most extensively used algorithm in the existing DRL-based car-following controllers. Since DDPG is designed to solve MDP problems, it is questionable whether the algorithm is suitable for solving the RD-Dec-POMDP problem of PC. In the following, we discuss the adoption of DDPG in the multiagent setting and random delay setting, respectively.

A. Multiagent Problem in DRL-Based PC

The PC problem corresponds to a Dec-POMDP and lies in the multiagent domain. Although there are various multiagent algorithms such as MADDPG [62] for applying RL to multiagent systems, we adopt the IL approach where each agent learns independently using DDPG. The reason for choosing IL is due to its simplicity and scalability. More importantly, the nonstationary environment issue for IL is greatly alleviated in the PC problem, since it is proved in [63] that only the actions of its predecessors but not the followers will affect the environment of a PC agent. Furthermore, the credit assignment issue in multiagent problem does not exist for our PC model, as each agent optimizes its individual return instead of the global return that is the sum of individual returns over all the PC agents.

B. Random Observation Delay Problem in DRL-Based PC

The theoretical foundation of the DDPG algorithm is the deterministic policy gradient (DPG) Theorem [61], [64], which shows that DPG is the expected gradient of the action-value function for any MDP whose corresponding gradients exist. By the discussion in Section V-A, we can approximately consider that the undelayed driving status $x_{i,k}$ at PC agent i is Markov, i.e., $p(x_{i,k+1}|x_{i,k}, a_{i,k}) = p(x_{i,k+1}|\ldots, x_{i,k-1}, x_{i,k}, a_{i,k})$, ignoring the impact of the predecessors' actions on $x_{i,k+1}$. However, each PC agent i can only observe the delayed driving status $x_{i,k-\tau_{i,k}}$ instead of $x_{i,k}$, where $x_{i,k-\tau_{i,k}}$ is no longer a Markov state. It is proved in the following Theorem 1 that $S^{\mathrm{CL}}_{i,k}$ becomes a Markov state by augmenting the delayed observation of driving status with action history.

Theorem 1: Markov property is ensured for the augmented state $S^{\mathrm{CL}}_{i,k}$, i.e., $p(S^{\mathrm{CL}}_{i,k+1}|S^{\mathrm{CL}}_{i,k},\ a^{\mathrm{CL}}_{i,k}) = p(S^{\mathrm{CL}}_{i,k+1}|\ldots, S^{\mathrm{CL}}_{i,k-1}, S^{\mathrm{CL}}_{i,k}, a^{\mathrm{CL}}_{i,k})$.

The proof of Theorem 1 is given in Appendix A.

The reward function $R^{\mathrm{CL}}_{i,k}(x_{i,k}, a^{\mathrm{CL}}_{i,k})$ defined in (18) is a function of the undelayed observation $x_{i,k}$ instead of the augmented state $S^{\mathrm{CL}}_{i,k}$. In order to construct an MDP for the delayed observations, we define the delayed reward function $\tilde{R}^{\mathrm{CL}}_{i,k}(S^{\mathrm{CL}}_{i,k}, a^{\mathrm{CL}}_{i,k})$ for each follower $i \in \mathcal{V}\backslash\{0\}$ as the expected reward obtained by PC agent i in augmented state $S^{\mathrm{CL}}_{i,k}$, i.e.,

$$\tilde{R}^{\mathrm{CL}}_{i,k}\left(S^{\mathrm{CL}}_{i,k}, a^{\mathrm{CL}}_{i,k}\right) = \mathrm{E}_{x_{i,k}}\left[R^{\mathrm{CL}}_{i,k}\left(x_{i,k}, a^{\mathrm{CL}}_{i,k}\right)|S^{\mathrm{CL}}_{i,k}\right]. \tag{21}$$

Now we construct the augmented state MDP $\tilde{\mathcal{M}}_i = (S^{\mathrm{CL}}_{i,k}, a^{\mathrm{CL}}_{i,k}, \tilde{R}^{\mathrm{CL}}_{i,k}, p, \gamma)$. Note that under $\tilde{\mathcal{M}}_i$, the expected return

$\tilde{J}_i^{\mathrm{CL}}$ of PC agent i under policy π_i^{CL} is written as follows:

$$\tilde{J}_i^{\mathrm{CL}} = \mathrm{E}_{\pi^{\mathrm{CM}}}\mathrm{E}_{\pi_i^{\mathrm{CL}}}\left[\sum_{k=0}^{K-1}\gamma^k \tilde{R}_{i,k}^{\mathrm{CL}}\right],\ 0 \le \gamma \le 1. \tag{22}$$

Thus, the optimal policy for $\tilde{\mathcal{M}}_i$ is given as follows:

$$\tilde{\pi}_i^{\mathrm{CL}*} = \arg\max_{\pi_i^{\mathrm{CL}*}} \tilde{J}_i^{\mathrm{CL}},\ \forall i \in \mathcal{V}\backslash\{0\}. \tag{23}$$

The following Theorem 2 states that the optimal policy $\tilde{\pi}_i^{\mathrm{CL}*}$ for the augmented state MDP $\tilde{\mathcal{M}}_i$ is the same as the optimal policy $\pi_i^{\mathrm{CL}*}$ in (20) for our PC problem under delayed observation.

Theorem 2: If the initial distributions of $p(x_{i,0})$ and $p(S_{i,0}^{\mathrm{CL}})$ satisfy

$$p(x_{i,0}) = p\left(S_{i,0}^{\mathrm{CL}}\right)\mathrm{E}_{\pi^{\mathrm{CM}}}\mathrm{E}_{\pi_i^{\mathrm{CL}}}\left[\mathbf{1}(x_{i,0})|S_{i,0}^{\mathrm{CL}}\right] \tag{24}$$

we have

$$\tilde{J}_i^{\mathrm{CL}} = J_i^{\mathrm{CL}}, \tilde{\pi}_i^{\mathrm{CL}*} = \pi_i^{\mathrm{CL}*} \tag{25}$$

The proof of Theorem 2 is given in Appendix B.

Based on Theorem 2, the optimal PC policy under delayed observation $\pi_i^{\mathrm{CL}*}$ can be derived by solving $\tilde{\mathcal{M}}_i$. For this purpose, we apply the DDPG algorithm and the DPG for $\tilde{\mathcal{M}}_i$ is given in Lemma 1.

Lemma 1: The DPG for the augmented state MDP $\tilde{\mathcal{M}}_i$ is

$$\nabla_{\theta_i^\mu} J_i^{\mathrm{CL}}(\mu_i^{\mathrm{CL}}) = \mathrm{E}\Bigg[\nabla_{\theta_i^\mu}\mu_i^{\mathrm{CL}}\left(S_{i,k}^{\mathrm{CL}}|\theta_i^\mu\right) \nabla_a Q_i^{\mathrm{CL}}\left(S_{i,k}^{\mathrm{CL}}, a|\theta_i^Q\right)|_{a=\mu_i^{\mathrm{CL}}(S_{i,k}^{\mathrm{CL}}|\theta_i^\mu)}\Bigg]. \tag{26}$$

The proof of Lemma 1 is straightforward as $\tilde{\mathcal{M}}_i$ is an MDP for which the DPG Theorem can be directly applied.

In order to sample the DPG in (26), we need to evaluate the action-value function $Q_{\mu_{\theta_i}}^{\mathrm{CL}}(S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}})$ of the augmented state MDP $\tilde{\mathcal{M}}_i$. Based on the following Bellman equation:

$$Q_{\mu_{\theta_i}}^{\mathrm{CL}}\left(S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}}\right) = \mathrm{E}_{x_{i,k}}\left[R_{i,k}^{\mathrm{CL}}\left(x_{i,k}, a_{i,k}^{\mathrm{CL}}\right)|S_{i,k}^{\mathrm{CL}}\right] + \gamma \mathrm{E}_{S_{i,k+1}^{\mathrm{CL}}}\left[Q_{\mu_{\theta_i}}^{\mathrm{CL}}\left(S_{i,k+1}^{\mathrm{CL}}, \mu_{\theta_i}(S_{i,k+1}^{\mathrm{CL}})\right)|S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}}\right] \tag{27}$$

the PC agent i can sample the undelayed reward $R_{i,k}^{\mathrm{CL}}(x_{i,k}, a_{i,k}^{\mathrm{CL}})$ and next state $S_{i,k+1}^{\mathrm{CL}}$ at control interval k, and calculate the temporal-difference (TD) target as follows:

$$y_{i,k} = R_{i,k}^{\mathrm{CL}}\left(x_{i,k}, a_{i,k}^{\mathrm{CL}}\right) + \gamma Q_{\mu_{\theta_i}}^{\mathrm{CL}}\left(S_{i,k+1}^{\mathrm{CL}}, \mu_{\theta_i}(S_{i,k+1}^{\mathrm{CL}})\right). \tag{28}$$

Remark 2 (Assumption of Undelayed Reward): We assume that there is no reward delay, i.e., the reward $R_{i,k}^{\mathrm{CL}}(x_{i,k}, a_{i,k}^{\mathrm{CL}})$ based on the current driving status $x_{i,k}$ is available to the PC agent during training at each control interval k. This is possible since learning can be performed in a simulator or a laboratory in which the undelayed reward is available. After the agent learns the PC policy, the reward is no longer needed during execution when the undelayed reward is not available.

C. MTCC-PC Algorithm

Based on the above discussion, the MTCC-PC algorithm is proposed. Each PC agent i adopts the DDPG algorithm given in [61]. Specifically, DDPG develops both a pair of actor and critic networks, i.e., $\mu_i^{\mathrm{CL}}(S_{i,k}^{\mathrm{CL}}|\theta_i^\mu)$ and $Q_i^{\mathrm{CL}}(S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}}|\theta_i^Q)$, to derive the optimal policy $\mu_i^{\mathrm{CL}*}(S_{i,k}^{\mathrm{CL}}|\theta_i^\mu)$ and the corresponding action-value $Q_i^{\mathrm{CL}*}(S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}}|\theta_i^Q)$, respectively. A copy of the actor and critic networks are created as target networks, i.e., $\mu_i^{\mathrm{CL}'}(S_{i,k}^{\mathrm{CL}}|\theta_i^{\mu'})$ and $Q_i^{\mathrm{CL}'}(S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}}|\theta_i^{Q'})$, to calculate the target values. To enable stable and robust learning, DDPG uses experience replay, and the networks are updated using minibatch samples from the experience buffer. During training, the sampled DPG ascent on $Q_i^{\mathrm{CL}}(S_{i,k}, \mu_i^{\mathrm{CL}}(S_{i,k}|\theta_i^\mu)|\theta_i^Q)$ with regard to θ_i^μ is used to train the actor network, and the critic network is trained by minimizing the root mean square error (RMSE) $L_{i,k} = y_{i,k} - Q_i^{\mathrm{CL}}(S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}}|\theta_i^Q)$ using the sampled gradient descent with respect to θ_i^Q. We refer the interested readers to [61] for the details of the DDPG algorithm.

In the following Remarks 3 and 4, we highlight two important design details of the MTCC-PC algorithm.

Remark 3 (Finite-Horizon Problem in DRL-Based PC): The PC problem in Section III-A considers a finite horizon with K control intervals. However, the optimal policies are normally time dependent in a finite-horizon setting, while DDPG is designed to solve the infinite-horizon or indefinite-horizon problems, where the learned policy is the same for every time step [21]. In order to deal with this problem, we set the target values of DDPG in the last control interval $K-1$ to be derived by (28) in the way as for the other control intervals, i.e., the sum of the immediate reward and the discounted target Q value of the next state instead of only the immediate reward $R_{i,K-1}^{\mathrm{CL}}(x_{i,K-1}, a_{i,K-1}^{\mathrm{CL}})$. Thus, the PC problem is transformed from a finite horizon problem to an infinite horizon problem.

Remark 4 (Simulation of Delayed Environment When Training MTCC-PC): The proposed MTCC-PC algorithm is trained in a delayed environment generated by the simulation of C-V2X communications with de facto RRA policy rather than by a coarse-grained stochastic delay model. This is to ensure the delay distribution in the training environment is the same as that in the execution environment in practice.

VI. Experimental Results

In this section, we design experiments to demonstrate that the proposed MTCC-PC algorithm outperforms the state-of-the-art communication-aware control, where the former is trained by a delayed environment generated by the fine-grained embedded simulation of C-V2X communications while the latter is trained by a simple stochastic delay model. Specifically, the baseline algorithm is random delay-aware PC (RD-PC), where the observation delay when training DRL-based PC is assumed to follow uniform distribution within the delay set $\{1, 2, 3, 4, 5\}$. In addition, to demonstrate that MTCC-PC can improve the PC performance by augmenting the PC state with action history, we design a baseline algorithm, namely, PC without augmented state (PC_wo_AS), which is the same as MTCC-PC except that the state only includes

TABLE II
TECHNICAL CONSTRAINTS AND OPERATIONAL PARAMETERS OF THE PC AND RRA ENVIRONMENT

Description	**Value**
PC environment	
Control interval	0.05 s
Total time steps in each control episode K	120
Number of vehicles N	5
Driveline dynamics time constant τ_i	0.1 s
Time gap h_i	1 s
Standstill distance r_i	2 m
Body length of the vehicle L_i	4.5 m
Acceleration limitations $[acc_{\min}, acc_{\max}]$	$[-4.3, 2.9]$ m/s^2
Control input limitations $[u_{\min}, u_{\max}]$	$[-4.3, 2.9]$ m/s^2
Control reward coefficient $\{\alpha_1, \alpha_2, \alpha_3\}$	$\{0.2, 0.1, 0.4\}$
Nominal maximum gap-keeping error $\hat{e}_{p,\max}$	10 m
Nominal maximum velocity error $\hat{e}_{v,\max}$	10 m/s
RRA environment	
Communication interval	1 ms
Total time steps in each control interval T	50
Number of V2I links M	2
Carrier frequency f_c	2 GHz
Bandwidth of sub-channel W	180 KHz
Noise power σ^2	-114 dBm
CAM size N_c	400 bytes
V2I transmit power P_m^{I}	23 dBm
V2V transmit power $P_{i,m,(k,t)}^{\mathrm{V}}$	$\{23, 15, 5, -100\}$ dBm
BS antenna height	25 m
BS antenna gain	8 dBi
BS receiver noise figure	5 dB
Vehicle antenna height	1.5 m
Vehicle antenna gain	3 dBi
Vehicle receiver noise figure	9 dB
Communication reward coefficient $\{\kappa_1, \kappa_2\}$	$\{0.001/W, 100\}$

TABLE III
HYPER-PARAMETERS OF THE DRL ALGORITHMS FOR TRAINING

Parameter	**Value**
Actor network size	$256, 128$
Critic network size	$256, 128$
Actor activation function	relu, relu, tanh
Critic activation function	relu, relu, linear
Actor learning rate ψ	0.0001
Critic learning rate φ	0.001
Batch size N_b	64
Replay buffer size	600000
Reward discount factor γ	0.99
Soft target update of DDPG	0.001
Noise type	Ornstein-Uhlenbeck Process with $\theta = 0.15$ and $\sigma = 0.5$
Final layer weights/biases initialization	Random uniform distribution $[-3 \times 10^{-3}, 3 \times 10^{-3}]$
Other layer weights/biases initialization	Random uniform distribution$[-\frac{1}{\sqrt{f}}, \frac{1}{\sqrt{f}}]$ (f is the fan-in of the layer)

the delayed observation of driving state $x_{i,k-\tau_{i,k}}$. MTCC-PC, RD-PC, and PC_wo_AS are both trained for 10 000 episodes and tested where C-V2X communications are implemented with the random RRA policy. Therefore, the induced delay distribution is the same for RD-PC, PC_wo_AS, and MTCC-PC when evaluating their performance.

A. Experimental Setup

1) Driving Data for Leading Vehicle 0: All the DRL algorithms are trained/tested where the velocity profile of leading vehicle 0 is obtained from the open-source driving data in [65]. Specifically, the driving data from the next generation simulation (NGSIM) data set [66] was first obtained, based on which the car-following events were extracted by applying a car-following filter as described in [67]. In our experiments, the velocity of the leading vehicle 0 in each control episode follows the corresponding data of the leading vehicle in one car-following event, so that the real-world PC environment with uncertainty can be simulated. We used 900 car-following events, 800 of which are used for training, and 100 for testing.

2) Parameter Setting: The technical constraints and operational parameters of the PC and RRA environment are given in Table II. In general, the parameters of the PC environment are determined mainly using the values reported in [60] and the urban case defined in [41]. Each control episode is comprised of 120 control intervals (i.e., $K = 120$), where each control interval is set to $T = 0.05$ s [58], [68], [69]. As the number of vehicles simulated in the existing literature on PC normally ranges from 3 to 8 [44], [45], [46], we set the number of vehicles to $N = 5$. We initialize the driving status for the platoon with 2-D positions $\{p_{\mathrm{V,i,0}}\}_0^{N-1} = \{(416, 427.5), (399, 427.5), (383, 427.5), (366, 427.5), (350, 427.5)\}$, $\{v_{i,0}\}_{i=0}^{N-1} = \{10, 10, 10, 10, 10\}$ m/s, and $\{\mathrm{acc}_{i,0}\}_{i=0}^{N-1} = \{0, 0, 0, 0, 0\}$ m/s^2. Note that the 2-D positions $\{p_{\mathrm{V,i,0}}\}_0^{N-1}$ are used for RRA and the corresponding 1-D positions $\{p_i, 0\}_0^{N-1} = \{416, 399, 383, 366, 350\}$ are used for PC. For the V2I vehicles, we initialize them with 2-D positions $\{p_{\mathrm{I,i,0}}\}_0^1 = \{(391, 434.75), (358, 434.75)\}$ and constant velocity 10 m/s. The nominal maximum control errors in the reward function (18) are set to $\hat{e}_{p,\max} = 10$ m and $\hat{e}_{v,\max} = 10$ m/s so that it is larger than most possible control errors during training for all DRL algorithms. For the parameter setting of the RRA environment, we mainly follow the experimental setup in [35] for channel models of V2I and V2V links. The bandwidth of each subchannel is set to $W = 180$ kHz.

The main hyper-parameters for training are summarized in Table III. The values of all the hyper-parameters were selected by performing a grid search as in [70], using the values reported in [61] as a reference. RD-PC, PC_wo_AS, and MTCC-PC algorithms have the same network architecture for DDPG, which has two hidden layers with 256 and 128 nodes, respectively. The sizes of input layer is decided by the PF IFT. Moreover, an additional 1-D action input is fed to the second hidden layer for each critic network. The soft target update is implemented with a parameter of 0.001.

B. Performance Comparison of MTCC-PC, RD-PC, and PC_wo_AS

1) Performance for Testing Data: The individual PC performance of each follower $i \in \{1, 2, 3, 4\}$ as well as the sum PC performance of the four followers are reported in Table IV for MTCC-PC, RD-PC, and PC_wo_AS, respectively. The individual and the sum PC performance are obtained by

TABLE IV
PC PERFORMANCE AFTER TRAINING WITH NGSIM DATA SET IN ONE ITERATION. WE PRESENT THE INDIVIDUAL PERFORMANCE OF EACH FOLLOWER AS WELL AS THE SUM PC PERFORMANCE OF THE FOUR FOLLOWERS FOR MTCC-PC, RD-PC, AND PC_wo_AS, RESPECTIVELY

Algorithm	**Individual PC performance**				**Sum PC performance**
	Follower 1	**Follower 2**	**Follower 3**	**Follower 4**	
MTCC-PC	-0.8029	-0.5434	-0.3214	-0.3163	-1.9840
RD-PC	-1.4189	-1.1127	-0.8573	-0.7152	-4.1041
PC_wo_AS	-1.1750	-0.9786	-1.020	-0.5318	-3.7054

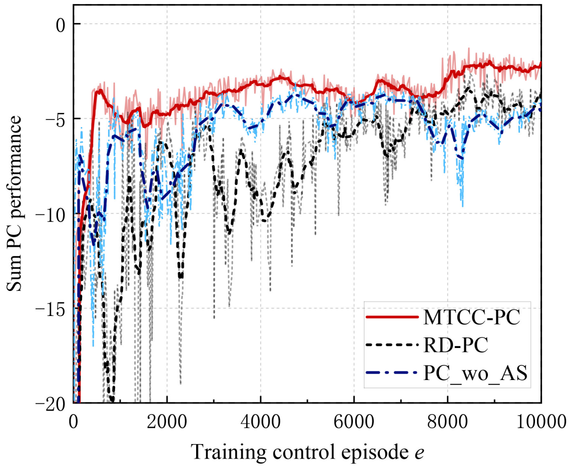


Fig. 5. Sum PC Performance during MTCC-PC, RD-PC, and PC_wo_AS training in one iteration. The vertical axis corresponds to the average returns over 10 test episodes. The dark curves correspond to smoothed curves and the light color curves correspond to the original curves.

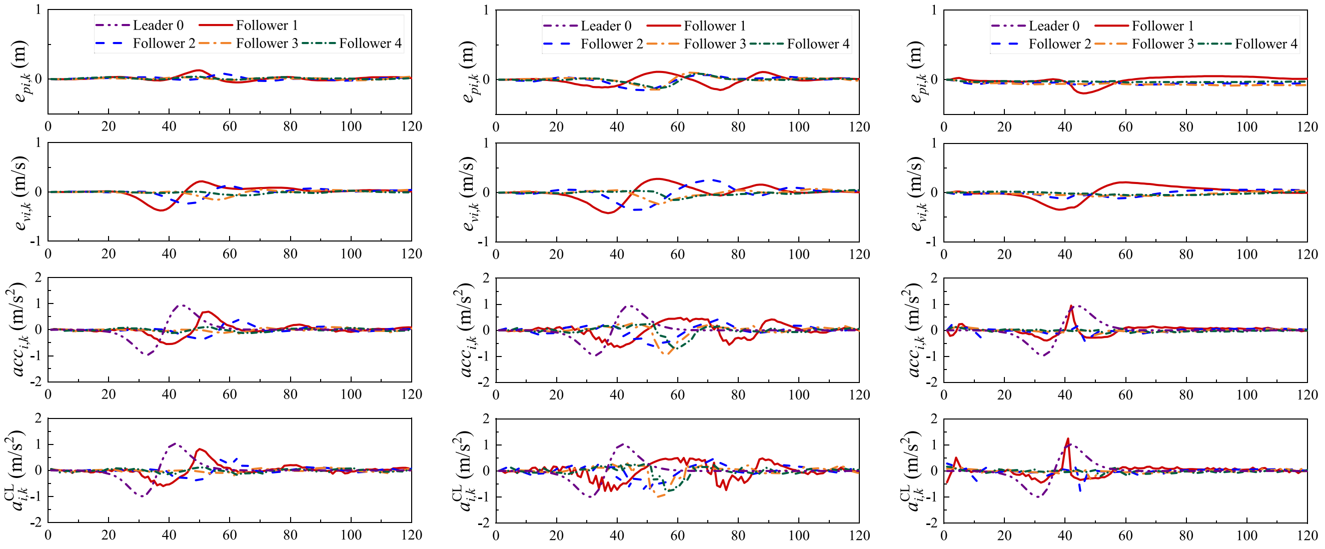
averaging the returns of the corresponding followers and the sum returns of all followers, respectively, over 100 test episodes after training is completed. Note that the return is the cumulative PC reward given in (18) of one control episode. Compared with RD-PC and PC_wo_AS, MTCC-PC consistently shows better individual PC performance for each follower $i \in \{1, 2, 3, 4\}$. Moreover, MTCC-PC outperforms RD-PC by 51.76% in terms of the sum PC performance of all followers. It demonstrates that training in a delayed environment generated by embedded simulation of C-V2X communications rather than by a simple stochastic delay model can improve PC performance. In addition, MTCC-PC outperforms PC_wo_AS by 46.46% in terms of the sum PC performance of all followers, demonstrating that ensuring the Markov property by augmenting the PC state with action history can significantly improve the PC performance.

2) Convergence Properties: The sum PC performance of MTCC-PC, RD-PC, and PC_wo_AS algorithms are evaluated periodically during training by testing in a delayed environment with fine-grained simulation of C-V2X communications under random RRA policy. Specifically, we run ten test episodes after every ten training episodes and average the sum PC performance over the ten test episodes as the performance for the latest ten training episodes. The performance as a function of the number of training episodes for MTCC-PC, RD-PC, and PC_wo_AS is plotted in Fig. 5. It can be observed from Fig. 5 that the performance of MTCC-PC is consistently better than those of RD-PC and PC_wo_AS during the whole training episode. In addition, the performance curve of RD-PC exhibits significantly larger oscillation during all the training episodes compared to that of MTCC-PC, demonstrating that the convergence of RD-PC is relatively unstable. Moreover, MTCC-PC has a significantly higher convergence rate than RD-PC and PC_wo_AS, as the performance of MTCC-PC converges at around 600 episodes, while those of RD-PC and PC_wo_AS converge at around 8000 and 3000 episodes, respectively. As explained above, MTCC-PC performs better than RD-PC since it is trained in an environment whose delay distribution is identical to that of the testing environment. Moreover, the faster and more stable convergence of MTCC-PC over RD-PC is also attributed to the fact that the observation delay in C-V2X communications is correlated between adjacent control intervals, while those generated by the uniform distribution are independent between control intervals. In addition, MTCC-PC performs better than PC_wo_AS since it ensures the Markov property of the augmented PC state $S_{i,k}^{\text{CL}}$.

3) Testing Results of One Episode: To further examine how the performance improvement of MTCC-PC over the RD-PC and PC_wo_AS algorithms in Table IV is reflected in the physical system, we focus on a specific test episode with 120 time steps and plot the tracking errors $e_{pi,k}$ and $e_{vi,k}$ of each follower $i \in \mathcal{V}\backslash\{0\}$ as well as the acceleration $\text{acc}_{i,k}$ and control input $a_{i,k}^{\text{CL}}$ of each vehicle $i \in \mathcal{V}$ for all time steps $k \in \{1, 2, \ldots, 120\}$. The results for MTCC-PC, RD-PC, and PC_wo_AS algorithms are shown in Fig. 6.

Fig. 6 shows that the performance differences among the algorithms are manifested in the speed of convergence to the steady state and the oscillations of the tracking errors, acceleration, and control input. In general, the speed of convergence to the steady state in RD-PC and PC_wo_AS for all followers is significantly slower than those in MTCC-PC. Also, the tracking errors, acceleration, and control input in RD-PC and PC_wo_AS have larger oscillations than those in MTCC-PC for all followers.

Specifically, $e_{pi,k}$ for each follower $i \in \mathcal{V}\backslash\{0\}$ in RD-PC reduces to 0 m (at around $k = 100$) later than in MTCC-PC (at around $k = 70$). It can be observed that there are positive gap-keeping errors for followers 1 in PC_wo_AS up to the end of the episode. The velocity error $e_{vi,k}$ for follower i in RD-PC has larger oscillations than those in MTCC-PC, especially from $k = 20$ to $k = 110$. $e_{vi,k}$ of follower 1 in PC_wo_AS has a slower convergence speed to 0 m than that in MTCC-PC. Regarding $\text{acc}_{i,k}$ and $a_{i,k}^{\text{CL}}$, the oscillations in RD-PC are larger than those in MTCC-PC for all followers i, especially at $k > 20$. In addition, RD-PC also has many more large jerks than MTCC-PC, which



(a) (b) (c)

Fig. 6. Results of a specific test episode. The driving status $e_{pi,k}$, $e_{vi,k}$, and $\text{acc}_{i,k}$ along with the control input $a^{\text{CL}}_{i,k}$ of each follower i are represented as different curves, respectively. (a) MTCC-PC. (b) RD-PC. (c) PC_wo_AS.

greatly reduces driving comfort. PC_wo_AS also has larger jerks than those of MTCC-PC, especially at the beginning of the episode and at around $k = 40$. Although MTCC-PC has a better performance compared to RD-PC and PC_wo_AS, there are still many small jerks for $a^{\text{CL}}_{i,k}$, especially for followers 1 and 2. This is because MTCC-PC is based on a random RRA policy for C-V2X communications.

An important requirement for PC is to guarantee string stability. When oscillations of the preceding vehicle are attenuated by following vehicles upstream of the platoon, the platoon is considered string stable. For example, as shown in Fig. 6(a), the amplitudes of the oscillations in $e_{pi,k}$, $e_{vi,k}$, and $\text{acc}_{i,k}$ for each follower $i \in \mathcal{V}\backslash\{0\}$ are smaller than those of their respective predecessors $i-1$ in MTCC-PC. The reduction in oscillation amplitude demonstrates the string stability of the platoon. The string stability of the platoon is not satisfactory for RD-PC and PC_wo_AS in Fig. 6(b) and (c) since the amplitudes of the oscillations in $\text{acc}_{i,k}$ of RD-PC for follower 3 are larger than those for follower 2 at around $k = 58$, and the amplitudes of the oscillations in $\text{acc}_{i,k}$ of PC_wo_AS for follower 1 are larger than those for leading vehicle 0 at around $k = 42$. The reason why MTCC-PC performs better in terms of string stability than RD-PC and PC_wo_AS is due to the definition of the reward function $R^{\text{CL}}_{i,k}$ in (18). While the first, second, and fourth terms in $R^{\text{CL}}_{i,k}$ aim to minimize the absolute value of $e_{pi,k}$, $e_{vi,k}$, and $\text{acc}_{i,k}$, the third and fourth terms aim to minimize the value of the control input $a^{\text{CL}}_{i,k}$, which will result in smaller oscillations of $e_{pi,k}$, $e_{vi,k}$, and $\text{acc}_{i,k}$. Since MTCC-PC achieves better PC performance than RD-PC and PC_wo_AS in terms of the expected cumulative reward J^{CL}_{i} in (19), it has a higher probability of satisfying the string stability than RD-PC and PC_wo_AS.

VII. Conclusion

In this article, we have decomposed the MTCC problem into a communication-aware DRL-based PC subproblem and a control-aware DRL-based RRA subproblem. In order to solve the PC subproblem, we have augmented the PC state space with the observation delay and PC action history, and defined the reward function for the augmented state to conceive the augmented state MDP. We have proved that the optimal policy for the MDP is also optimal for the PC problem with observation delay. Finally, the experimental results have demonstrated that:

1) training in a delayed environment generated by embedded simulation of C-V2X communications in MTCC rather than by a simple stochastic delay model can improve PC performance, since the delay distribution during training complies with that during execution in practice;
2) the PC performance can be improved by augmenting the state with the action history. In Part II of this two-part paper, we will propose the MTCC-RRA algorithm to learn the RRA policy and design a sample- and computational-efficient training approach to jointly train MTCC-PC and MTCC-RRA algorithms in an iterative process.

Appendix A
Proof of Theorem 1

First, we have

$$p(S^{\text{CL}}_{i,k+1}|S^{\text{CL}}_{i,k}, a^{\text{CL}}_{i,k}) = p\Bigg(x_{i,k+1-\tau_{i,k+1}}, \left\{a^{\text{CL}}_{i,k'}\right\}_{k'=k+1-\tau_{\max}}^{k}, \tau_{i,k+1}|x_{i,k-\tau_{i,k}} \left\{a^{\text{CL}}_{i,k'}\right\}_{k'=k-\tau_{\max}}^{k-1}, \tau_{i,k}, a^{\text{CL}}_{i,k}\Bigg). \tag{29}$$

According to (13), we discuss the following two situations.

1) *If* $\tau_{i,k+1} = \tau_{i,k} + 1$, we have $x_{i,k+1-\tau_{i,k+1}} = x_{i,k-\tau_{i,k}}$. Therefore

$$\begin{aligned}&p\left(S_{i,k+1}^{\mathrm{CL}}|S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}}\right)\\ &\overset{(a)}{=} p\left(\tau_{i,k+1}|\tau_{i,k}, x_{i,k-\tau_{i,k}}\right)\mathbf{1}\left\{\left\{a_{i,k'}^{\mathrm{CL}}\right\}_{k'=k+1-\tau_{\max}}^{k}\right.\\ &\left.= \left\{a_{i,k'}^{\mathrm{CL}}\right\}_{k'=k+1-\tau_{\max}}^{k-1}, a_{i,k}^{\mathrm{CL}}\right\}\\ &= p\left(\tau_{i,k+1}|\tau_{i,k}, x_{i,k-\tau_{i,k}}\right) \qquad (30)\end{aligned}$$

where the indicator function $\mathbf{1}\{X\}$ is 1 when X is true and 0 otherwise. $p(\tau_{i,k+1}|\tau_{i,k}, x_{i,k-\tau_{i,k}})$ in (a) holds since the observation delay $\tau_{i,k}$ is derived from the queue length $q_{i-1,(k-1,T)}^{\mathrm{CAM}}$ according to (15), where $q_{i-1,(k-1,T)}^{\mathrm{CAM}}$ is related to transmission rate of V2V link $i-1$ according to (12), which further depends on the observed driving status $x_{i,k-\tau_{i,k}}$.

2) *If* $\tau_{i,k+1} = \tau_{i,k} - d$, $0 \leq d \leq \tau_{i,k} - 1$, we have

$$\begin{aligned}&p\left(S_{i,k+1}^{\mathrm{CL}}|S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}}\right)\\ &= p\left(\tau_{i,k+1}|\tau_{i,k}, x_{i,k-\tau_{i,k}}\right)\mathbf{1}\left\{\left\{a_{i,k'}^{\mathrm{CL}}\right\}_{k'=k+1-\tau_{\max}}^{k}\right.\\ &\left.= \left\{a_{i,k'}^{\mathrm{CL}}\right\}_{k'=k+1-\tau_{\max}}^{k-1}, a_{i,k}^{\mathrm{CL}}\right\}\\ &p\left(x_{i,k+1-\tau_{i,k+1}}|x_{i,k-\tau_{i,k}}, \left\{a_{i,k'}^{\mathrm{CL}}\right\}_{k'=k-\tau_{i,k}}^{k-(\tau_{i,k}+d+1)}\right)\\ &\overset{(a)}{=} p\left(\tau_{i,k+1}|\tau_{i,k}, x_{i,k-\tau_{i,k}}\right)\\ &\sum_{x_{i,k-\tau_{i,k}},\ldots,x_{i,k-(\tau_{i,k}-d+1)}} \left\{p\left(x_{i,k-(\tau_{i,k}+1)}|x_{i,k-\tau_{i,k}}, a_{i,k-\tau_{i,k}}^{\mathrm{CL}}\right)\right.\\ &p\left(x_{i,k-(\tau_{i,k}+2)}|x_{i,k-(\tau_{i,k}+1)}, a_{i,k-(\tau_{i,k}+1)}^{\mathrm{CL}}\right)\cdot\\ &\left.\ldots p\left(x_{i,k-\tau_{i,k+1}}|x_{i,k-(\tau_{i,k}-d+1)}, a_{i,k-(\tau_{i,k}-d+1)}^{\mathrm{CL}}\right)\right\} \qquad (31)\end{aligned}$$

where (a) holds since we approximately consider that the $x_{i,k}$ at PC agent i is Markov and therefore the probability of multistep transition $p(x_{i,k+1-\tau_{i,k+1}}|x_{i,k-\tau_{i,k}}, \{a_{i,k'}^{\mathrm{CL}}\}_{k'=k-\tau_{i,k}}^{k-(\tau_{i,k}+d+11)})$ is also independent of statue history $\{\ldots x_{i,k-\tau_{i,k}-1}\}$. In summary, the above derivation demonstrates that $S_{i,k+1}^{\mathrm{CL}}$ only depends on the current state and action pair $\{S_{i,k}^{\mathrm{CL}}, a_{i,k}^{\mathrm{CL}}\}$ but not the history $\{\ldots, S_{i,k-1}^{\mathrm{CL}}\}$. Therefore, the Markov property is proved for the augmented state $S_{i,k}^{\mathrm{CL}}$.

APPENDIX B
PROOF OF THEOREM 2

According to (22), we have

$$\begin{aligned}\tilde{J}_i^{\mathrm{CL}} &= \mathrm{E}_{\pi^{\mathrm{CM}}}\mathrm{E}_{\pi_i^{\mathrm{CL}}}\left[\sum_{k=0}^{K-1}\gamma^k \tilde{R}_{i,k}^{\mathrm{CL}}\right]\\ &= \mathrm{E}_{\pi^{\mathrm{CM}}}\mathrm{E}_{\pi_i^{\mathrm{CL}}}\left[\sum_{k=0}^{K-1}\gamma^k \mathrm{E}_{x_{i,k}}\left[R_{i,k}^{\mathrm{CL}}\left(x_{i,k}, a_{i,k}^{\mathrm{CL}}\right)|S_{i,k}^{\mathrm{CL}}\right]\right]\\ &= \mathrm{E}_{\pi^{\mathrm{CM}}}\mathrm{E}_{\pi_i^{\mathrm{CL}}}\left[\sum_{k=0}^{K-1}\gamma^k \sum_{x_{i,k}} p\left(x_{i,k}|S_{i,k}^{\mathrm{CL}}\right)\left[R_{i,k}^{\mathrm{CL}}\left(x_{i,k}, a_{i,k}^{\mathrm{CL}}\right)\right]\right].\end{aligned} \qquad (32)$$

In order to prove that $\tilde{J}_i^{\mathrm{CL}} = J_i^{\mathrm{CL}}$, and according to the definition J_i^{CL} in (19), we must prove that

$$\begin{aligned}&\mathrm{E}_{\pi^{\mathrm{CM}}}\mathrm{E}_{\pi_i^{\mathrm{CL}}}\left[\sum_{k=0}^{K-1}\gamma^k \sum_{x_{i,k}} p\left(x_{i,k}|S_{i,k}^{\mathrm{CL}}\right)\left[R_{i,k}^{\mathrm{CL}}\left(x_{i,k}, a_{i,k}^{\mathrm{CL}}\right)\right]\right]\\ &= \mathrm{E}_{\pi^{\mathrm{CM}}}\mathrm{E}_{\pi_i^{\mathrm{CL}}}\left[\sum_{k=0}^{K-1}\gamma^k R_{i,k}^{\mathrm{CL}}\left(x_{i,k}, a_{i,k}^{\mathrm{CL}}\right)\right].\end{aligned} \qquad (33)$$

Since both sides of (33) calculate the expected sum of undelayed reward $R_{i,k}^{\mathrm{CL}}(x_{i,k}, a_{i,k}^{\mathrm{CL}})$ between control intervals $[0, K-1]$, the equation holds if the probability distributions of the trajectories $(x_{i,0}, a_{i,0}^{\mathrm{CL}}, \ldots, x_{i,K-1}, a_{i,K-1}^{\mathrm{CL}})$ are the same on both sides of (33). Since both sides follow the same PC policy π_i^{CL} and RRA policy π^{CM}, we only need to make sure that the distributions of the initial state $x_{i,0}$ are the same on both sides. Note that the distribution of $x_{i,0}$ on the LHS of (33) depends on $S_{i,0}$, i.e., $x_{i,-\tau_{i,0}}$ before control interval 0. Therefore, given the distribution of $x_{i,0}$, i.e., $p(x_{i,0})$ on the RHS of (33), if the distribution of $S_{i,0}$, i.e., $p(S_{i,0}^{\mathrm{CL}})$ satisfies (24) in Theorem 2, the resultant distribution of $x_{i,0}$ on the LHS of (33) is the same as that on the RHS.

Since we have proved that $\tilde{J}_i^{\mathrm{CL}} = J_i^{\mathrm{CL}}$, it is obvious that $\tilde{\pi}_i^{\mathrm{CL}*} = \pi_i^{\mathrm{CL}*}$ according to (20) and (23).

REFERENCES

[1] S. Gyawali, S. Xu, Y. Qian, and R. Q. Hu, "Challenges and solutions for cellular based V2X communications," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 222–255, 1st Quart., 2020.

[2] S. Lee, Y. Jung, Y.-H. Park, and S.-W. Kim, "Design of V2X-based vehicular contents centric networks for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13526–13537, Aug. 2022.

[3] B. Yang et al., "Edge intelligence for autonomous driving in 6G wireless system: Design challenges and solutions," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 40–47, Apr. 2021.

[4] X.-M. Zhang et al., "Networked control systems: A survey of trends and techniques," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 1, pp. 1–17, Jan. 2019.

[5] A. Redder, A. Ramaswamy, and D. E. Quevedo, "Deep reinforcement learning for scheduling in large-scale networked control systems," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 333–338, 2019.

[6] O. Ayan, P. Kutsevol, H. Y. Özkan, and W. Kellerer, "Task-oriented scheduling for networked control systems: An age of information-aware implementation on software-defined radios," 2022, *arXiv:2202.09189*.

[7] Y. Zheng, S. E. Li, J. Wang, D. Cao, and K. Li, "Stability and scalability of homogeneous vehicular platoon: Study on the influence of information flow topologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 14–26, Jan. 2015.

[8] S. E. Li et al., "Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 46–58, Jul. 2017.

[9] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE Int. Conf. Comput. Commun.(INFOCOM)*, 2012, pp. 2731–2735.

[10] A. Kosta, N. Pappas, A. Ephremides, and V. Angelakis, "Age and value of information: Non-linear age case," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 326–330.

[11] Y. Li, C. Tang, S. Peeta, and Y. Wang, "Nonlinear consensus-based connected vehicle platoon control incorporating car-following interactions and heterogeneous time delays," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2209–2219, Jun. 2018.

[12] D. Huang, S. Li, Z. Zhang, Y. Liu, and B. Mi, "Design and analysis of longitudinal controller for the platoon with time-varying delay," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 23628–23639, Dec. 2022.

[13] L. Xu, X. Jin, Y. Wang, Y. Liu, W. Zhuang, and G. Yin, "Stochastic stable control of vehicular platoon time-delay system subject to random switching topologies and disturbances," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 5755–5769, Jun. 2022.

[14] L. Xu, W. Zhuang, G. Yin, C. Bian, and H. Wu, "Modeling and robust control of heterogeneous vehicle platoons on curved roads subject to disturbances and delays," *IEEE Trans. Veh. Technol.*, vol. 68, no. 12, pp. 11551–11564, Dec. 2019.

[15] F. Ma et al., "Distributed control of cooperative vehicular platoon with nonideal communication condition," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8207–8220, Aug. 2020.

[16] J. Wang, F. Ma, Y. Yang, J. Nie, B. Aksun-Guvenc, and L. Guvenc, "Adaptive event-triggered platoon control under unreliable communication links," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1924–1935, Mar. 2020.

[17] Y. Li, W. Chen, S. Peeta, and Y. Wang, "Platoon control of connected multi-vehicle systems under V2X communications: Design and experiments," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1891–1902, May 2019.

[18] S. Öncü, J. Ploeg, N. Van De Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1527–1537, Aug. 2014.

[19] T. Yang and C. Lv, "A secure sensor fusion framework for connected and automated vehicles under sensor attacks," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22357–22365, Nov. 2022.

[20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[21] L. Lei, Y. Tan, K. Zheng, S. Liu, K. Zhang, and X. Shen, "Deep reinforcement learning for autonomous Internet of Things: Model, applications and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1722–1760, 3rd Quart., 2020.

[22] K. C. Dey et al., "A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC)," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 491–509, Feb. 2016.

[23] Y. Zhong, B. Wang, and Y. Wang, *Reward Backpropagation Prioritized Experience Replay*, Stanford Univ., Palo Alto, CA, USA, 2017.

[24] X. Li, L. Ma, R. Shankaran, Y. Xu, and M. A. Orgun, "Joint power control and resource allocation mode selection for safety-related V2X communication," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7970–7986, Aug. 2019.

[25] F. Jameel, W. U. Khan, N. Kumar, and R. Jäntti, "Efficient power-splitting and resource allocation for cellular V2X communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3547–3556, Jun. 2021.

[26] S.-Y. Lien, S.-C. Hung, D.-J. Deng, C.-L. Lai, and H.-L. Tsai, "Low latency radio access in 3GPP local area data networks for V2X: Stochastic optimization and learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4867–4879, Jun. 2018.

[27] L. F. Abanto-Leon, A. Koppelaar, C. B. Math, and S. H. De Groot, "Impact of quantized side information on subchannel scheduling for cellular V2X," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC)*, 2018, pp. 1–5.

[28] J. Mei, K. Zheng, L. Zhao, L. Lei, and X. Wang, "Joint radio resource allocation and control for vehicle platooning in LTE-V2V network," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12218–12230, Dec. 2018.

[29] Q. Han, C. Liu, H. Yang, and Z. Zuo, "Longitudinal control-oriented spectrum sharing based on C-V2X for vehicle platoons," *IEEE Syst. J.*, vol. 17, no. 1, pp. 1125–1136, Mar. 2023.

[30] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2681–2690.

[31] H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[32] H. Yang, X. Xie, and M. Kadoch, "Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoV communication networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4157–4169, May 2019.

[33] K. Zia, N. Javed, M. N. Sial, S. Ahmed, A. A. Pirzada, and F. Pervez, "A distributed multi-agent RL-based autonomous spectrum allocation scheme in D2D enabled multi-tier HetNets," *IEEE Access*, vol. 7, pp. 6733–6745, 2019.

[34] Z. Nan, Y. Jia, Z. Ren, Z. Chen, and L. Liang, "Delay-aware content delivery with deep reinforcement learning in Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8918–8929, Jul. 2021.

[35] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, Oct. 2019.

[36] H. V. Vu, Z. Liu, D. H. Nguyen, R. Morawski, and T. Le-Ngoc, "Multi-agent reinforcement learning for joint channel assignment and power allocation in platoon-based C-V2X systems," 2020, *arXiv:2011.04555*.

[37] P. Xiang, H. Shan, M. Wang, Z. Xiang, and Z. Zhu, "Multi-agent RL enables decentralized spectrum access in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10750–10762, Oct. 2021.

[38] H. Zhang et al., "Mean-field aided multiagent reinforcement learning for resource allocation in vehicular networks," *IEEE Internet Things J.*, vol. 10, no. 3, pp. 2667–2679, Feb. 2023.

[39] M. Parvini, M. R. Javan, N. Mokari, B. Abbasi, and E. A. Jorswieck, "AoI-aware resource allocation for platoon-based C-V2X networks via multi-agent multi-task reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 9880–9896, Aug. 2023.

[40] Y. Xu, K. Zhu, H. Xu, and J. Ji, "Deep reinforcement learning for multi-objective resource allocation in multi-platoon cooperative vehicular networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 9, pp. 6185–6198, Sep. 2023.

[41] "Technical specification group radio access network; study on LTE-based V2X services; (Release 14), Version 14.2.2," 3GPP, Sophia Antipolis, France, Rep. TS 36.885, 2016.

[42] "Technical specification group radio access network; study enhancement 3GPP support for 5G V2X services; (Release 15), Version 15.1.0," 3GPP, Sophia Antipolis, France, Rep. TS 22.886, 2017.

[43] R. Molina-Masegosa and J. Gozalvez, "LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications," *IEEE Veh. Technol. Mag.*, vol. 12, no. 4, pp. 30–39, Dec. 2017.

[44] G. Wang, J. Hu, Y. Huo, and Z. Zhang, "A novel vehicle platoon following controller based on deep deterministic policy gradient algorithms," in *Proc. 18th COTA Int. Con. Transp. Prof. (CICTP)*, 2018, pp. 76–86.

[45] T. Chu and U. Kalabić, "Model-based deep reinforcement learning for CACC in mixed-autonomy vehicle platoon," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, 2019, pp. 4079–4084.

[46] R. Yan, R. Jiang, B. Jia, J. Huang, and D. Yang, "Hybrid car-following strategy based on deep deterministic policy gradient and cooperative adaptive cruise control," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 4, pp. 2816–2824, Oct. 2022.

[47] T. Liu, L. Lei, K. Zheng, and K. Zhang, "Autonomous platoon control with integrated deep reinforcement learning and dynamic programming," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 5476–5489, Mar. 2023.

[48] E. Altman and P. Nain, "Closed-loop control with delayed information," *ACM Sigmetrics Perform. Eval. Rev.*, vol. 20, no. 1, pp. 193–204, 1992.

[49] K. V. Katsikopoulos and S. E. Engelbrecht, "Markov decision processes with delays and asynchronous cost collection," *IEEE Trans. Automat. control*, vol. 48, no. 4, pp. 568–574, Apr. 2003.

[50] S. Nath, M. Baranwal, and H. Khadilkar, "Revisiting state augmentation methods for reinforcement learning with stochastic delays," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 1346–1355.

[51] M. Chen, Y. Bai, H. V. Poor, and M. Wang, "Efficient RL with impaired observability: Learning to act with delayed and missing state observations," 2023, *arXiv:2306.01243*.

[52] Y. Bouteiller, S. Ramstedt, G. Beltrame, C. Pal, and J. Binas, "Reinforcement learning with random delays," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–21.

[53] S. Wen, G. Guo, B. Chen, and X. Gao, "Cooperative adaptive cruise control of vehicles using a resource-efficient communication mechanism," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 1, pp. 127–140, Mar. 2019.

[54] T. Zeng, O. Semiari, W. Saad, and M. Bennis, "Joint communication and control for wireless autonomous vehicular platoon systems," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7907–7922, Nov. 2019.

[55] R. Oliveira, C. Montez, A. Boukerche, and M. S. Wangham, "Co-design of consensus-based approach and reliable communication protocol for vehicular platoon control," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9510–9524, Sep. 2021.

[56] C. Hong et al., "A joint design of platoon communication and control based on LTE-V2V," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15893–15907, Dec. 2020.

[57] P. Zhang et al., "Joint optimization of platoon control and resource scheduling in cooperative vehicle-infrastructure system," *IEEE Trans. Intell. Veh.*, vol. 8, no. 6, pp. 3629–3646, Jun. 2023.

[58] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van De Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, 2011, pp. 260–265.

[59] S. S. Stankovic, M. J. Stanojevic, and D. D. Siljak, "Decentralized overlapping control of a platoon of vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 5, pp. 816–832, Sep. 2000.

[60] Y. Lin, J. McPhee, and N. Azad, "Comparison of deep reinforcement learning and model predictive control for adaptive cruise control," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 2, pp. 221–231, Jun. 2021.

[61] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[62] 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 *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6382–6393.

[63] L. Lei, T. Liu, K. Zheng, and L. Hanzo, "Deep reinforcement learning aided platoon control relying on V2X information," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 5811–5826, Jun. 2022.

[64] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 387–395.

[65] M. Zhu, Y. Wang, Z. Pu, J. Hu, X. Wang, and R. Ke, "Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving," *Transp. Res. Part C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102662.

[66] *(NGSIM) Next Generation Simulation*, U.S. Dept. Transp., Washington, DC, USA, 2009.

[67] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng, and F.-Y. Wang, "Capturing car-following Behaviors by deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 910–920, Mar. 2018.

[68] J. Wang, X. Xu, D. Liu, Z. Sun, and Q. Chen, "Self-learning cruise control using kernel-based least squares policy iteration," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 3, pp. 1078–1087, May 2014.

[69] M. Buechel and A. Knoll, "Deep reinforcement learning for predictive longitudinal control of automated vehicles," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, 2018, pp. 2391–2397.

[70] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

Tong Liu received the B.S. degree from Nanjing University of Posts and Telecommunications Nanjing, Nanjing, China, in 2018. He is currently pursuing the Ph.D. degree with Beijing University of Posts and Telecommunications, Beijing, China.

His current research interests include deep reinforcement learning, and modern control theory and their application in Internet of Vehicles.

Lei Lei (Senior Member, IEEE) received the B.S. and Ph.D. degrees in telecommunications engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2001 and 2006, respectively.

She is currently an Associate Professor with the College of Engineering and Physical Sciences, University of Guelph, Guelph, ON, Canada. Her research interests mainly lie in machine learning/deep reinforcement learning, Internet of Things/Internet of Vehicles, mobile edge computing, and smart grid.

Kan Zheng (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 1996, 2000, and 2005, respectively.

He is currently a Full Professor with Ningbo University, Ningbo, Zhejiang, China. He has authored over 200 journal articles and conference papers in the field of wireless communications, vehicular networks, IoT, and security. He has rich experiences in research and standardization of new emerging technologies toward 6G.

Prof. Zheng holds editorial board positions with several journals. He has also served in the organizing/TPC committees for more than ten conferences.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks.

Dr. Shen received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals, Ontario, in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and the Education Award in 2017 from the IEEE Communications Society (ComSoc), and the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the President of the IEEE ComSoc. He was the Vice President for Technical and Educational Activities, the Vice President for Publications, a Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.