This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TMC.2026.3650774, IEEE Transactions on Mobile Computing

V2X-Assisted Distributed Computing and Control Framework for Connected and Automated CAVs under Ramp Merging Scenario

Jiahou Chu, Qiong Wu, *Senior Member, IEEE*, Pingyi Fan, *Senior Member, IEEE*,
Wen Chen, *Senior Member, IEEE*, Kezhi Wang, *Senior Member, IEEE*,
Nan Cheng, *Senior Member, IEEE*, and Khaled B. Letaief, *Fellow, IEEE*

***Abstract*—This paper presents a mobile computing-based framework for distributed computing and cooperative control of connected and automated vehicles (CAVs) in ramp merging scenarios under intelligent transportation systems (ITS). A centralized trajectory planning problem is first formulated to optimize merging efficiency and safety. To eliminate reliance on a central controller, a distributed solution is developed using ADMM algorithm based on V2X communication, enabling CAVs to collaboratively compute trajectories in parallel by leveraging their onboard computing power. Building on this, a multi-vehicle model predictive control (MPC) problem is proposed to enhance system stability under strict constraints. To solve it efficiently, a Distributed Cooperative Iterative MPC (DCIMPC) method is introduced, which decomposes and reformulates the problem for real-time distributed execution across CAVs. Together, these methods form a mobile edge computing-driven control framework. Simulations and experiments demonstrate significant improvements in computational efficiency and system performance, highlighting the potential of mobile computing in cooperative CAV control.**

***Index Terms*—Ramp Merging, Distributed Control, Mobile Computing, Distributed Computing, V2X, ADMM, MPC**

This work was supported in part by Jiangxi Province Science and Technology Development Programme under Grant No. 20242BCC32016, in part by the National Natural Science Foundation of China under Grant No. 61701197, 62531015 and U25A20399, in part by the Basic Research Program of Jiangsu under Grant BK20252084, in part by the National Key Research and Development Program of China under Grant No. 2021YFA1000500(4), in part by the Shanghai Kewei under Grant 24DP1500500, in part by the Research Grants Council under the Areas of Excellence Scheme under Grant AoE/E-601/22-R and in part by the 111 Project under Grant No. B23008.

Jiahou Chu and Qiong Wu are with the School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China, and also with the School of Information Engineering, Jiangxi Provincial Key Laboratory of Advanced Signal Processing and Intelligent Communications, Nanchang University, Nanchang 330031, China (e-mail: 6211924035@stu.jiangnan.edu.cn, qiongwu@jiangnan.edu.cn) (Corresponding author: Qiong Wu)

Pingyi Fan is with the Department of Electronic Engineering, State Keylaboratory of Space Network and Communications, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China (email: fpy@tsinghua.edu.cn).

Wen Chen is with Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wenchen@sjtu.edu.cn).

Kezhi Wang is with the Department of Computer Science, Brunel University London, Middlesex UB8 3PH, UK (e-mail: Kezhi.Wang@brunel.ac.uk).

Nan Cheng is with the State Key Lab. of ISN and School of Telecommunications Engineering, Xidian University, Xi'an 71007l, China (e-mail dr.nan.cheng@ieee.org).

Khaled B. Letaief is with the Department of Electrical and Computer Engineering, the Hong Kong University of Science and Technology (HKUST), Hong Kong (email: eekhaled@ust.hk).

I. Introduction

A. Background

With the number of vehicles in urban areas continuing to rise, traffic congestion is worsening, and safety concerns are becoming increasingly critical [1]. The rapid development of autonomous vehicles (AVs) offers a promising approach to addressing these challenges. However, current AV systems primarily rely on single-vehicle intelligence, where each AV conducts environment perception and decision-making solely based on data from its onboard sensors and computing units, with limited mechanisms for inter-vehicle communication or cooperation [2]. As a result, CAV-to-Everything (V2X) technology, designed specifically for vehicular communications, has emerged as a key research focus within the communications community [3]. NR-V2X (New Radio V2X), a communication protocol proposed by the 3rd Generation Partnership Project (3GPP), enables AVs to achieve low-latency direct communication, facilitating the transition from AVs to connected and autonomous vehicles (CAVs) [4]–[6]. Intelligent transportation systems (ITS), composed of CAVs and other edge devices, are rapidly developing toward connected and cooperative paradigms [7].

Enabled by V2X technology, it becomes feasible to co-design cooperative control algorithms and distributed computing frameworks for scenarios prone to traffic congestion and accidents, such as on-ramp merging [8]. Ramp merging is a common traffic scenario in which CAVs from both the main road and the ramp must frequently brake and accelerate to adjust relative position and velocity to merge [9]. Due to the varying driving styles of human drivers and the lack of effective communication, traffic rules with suboptimal performance must often be implemented to ensure safety. For example, CAVs on the ramp are typically required to slow down and wait for a safe gap in the main road traffic. However, in ITS scenarios, CAVs and Road Side Units (RSU) can leverage V2X to perform trajectory planning and cooperative control before reaching ramps, achieving smoother merging and improving traffic performance [10]. Currently, some works have explored distributed and centralized CAV control algorithms for on-ramp merging areas.

Works implementing distributed control fully utilize the advantages of NR-V2X, i.e., decentralization, and CAV computing resources, but their performance is slightly inferior

[11], [12]. In contrast, centralized control algorithms typically use V2X to collect real-time CAV information at RSUs, establish and solve the global optimization problems. Thus, compared with distributed control algorithms with suboptimal performance, centralized ones offer better control effects and lower communication performance requirements. However, their reliance on central computing facilities increases deployment costs and reduces versatility. Therefore, the research and design of a distributed computing and control framework to solve global optimization problem for on-ramp merging is of substantial importance for advancing the development of ITS [13], [14].

B. Related Work and Motivation

Currently, many works have proposed centralized computing framework for CAV planning and control schemes based on V2X for ramp merging areas. In [15], Jing *et al.* introduced a ramp merging scheme based on multiplayer cooperative game theory, considering passengers' comfort, energy consumption, and travel time as optimization objectives. They decomposed a multiplayer cooperative game problem into multiple two-player game problems and solved them. Similarly employing cooperative game theory to address multi-vehicle cooperation in ramp scenarios, Yang *et al.* considered multi-lane situations compared to [15] in [16]. Rios *et al.* formulated the multi-vehicle control problem with safety constraints as an optimal control problem with minimized energy consumption and solved it using Hamiltonian analysis in [17]. Kherroubi *et al.* incorporated machine learning into the ramp merging control problem in [18]. They proposed a method using artificial neural networks to predict the trajectories of CAVs driven by human drivers, inputting the predicted trajectories into controllers trained by reinforcement learning, and then sending the controller's output to CAV. In [19], Milanes *et al.* proposed a two-layer control architecture for ramp merging scenarios, transforming the cooperative control problem of three CAVs into an optimal control problem with multiple safety constraints. They utilized the Pontryagin's Minimum Principle and Control Barrier Functions (CBFs) to solve the problem. In [20], Beheshtitabar *et al.* studied a rule-based control algorithm, each CAV entering the ramp must choose appropriate actions based on the traffic conditions of the main road to complete the merge.

All the above works are centralized control schemes, requiring the deployment of a base station or RSU in the scenario. Some works have also studied distributed CAV control schemes for ramp merging areas. In these works, some were rule-based, where ramp CAVs, based on surrounding information, choose merging positions on the main road according to certain rules. For example, in [21] and [22], Zhou *et al.* and Liu *et al.* proposed two similar CAV control methods, in which CAVs from the ramp are connected to the main road according to certain rules and send merging requests. In [23], Hwang *et al.* first established a CAV state control framework based on a Finite-State Machine (FSM), where CAVs switch states within the FSM based on different surrounding environments. When transitioning to a lane-changing state, CAVs are controlled using a controller trained through reinforcement learning. However, in this approach, lane-changing decisions for CAVs are controlled by the FSM, resulting in lower flexibility. Differing from the works in [21]–[23], Xue *et al.* proposed a distributed framework based on optimization theory in [24]. In this framework, the upper layer is used to select a suitable merging position, and the lower layer is used to adjust the specific relative speeds and positions between CAVs. Furthermore, some works have discussed distributed control scheme for ramp merging areas based on machine learning. In [25], Chen *et al.* considered the coexistence of autonomous CAVs and human-driven CAVs in ramp merging scenarios. In the study, the authors trained a CAV controller based on Multi-Agent Reinforcement Learning (MARL), with the controller outputting discrete actions of acceleration, deceleration, and lane-changing.

In summary, centralized ramp merging frameworks are primarily based on establishing a centralized problem and solving it through a central controller, resulting in better performance compared to distributed solutions. However, distributed frameworks can operate independently of a central controller, offering greater adaptability. How to combine the advantages of centralized and distributed frameworks and avoid their drawbacks is a great challenge in the complex scenarios of ramp merging.

C. Contributions

To the authors' best knowledge, V2X assisted frameworks of distributed computing to solve the centralized trajectory planning and control problem in ramp merging scenarios has not been studied thus far. In this paper, we try to find a new way to this complex problem and propose a framework where centralized optimization problems are solved by distributed computing. The main contributions of this are are summarized as follows:

1) A trajectory planning problem and a model predictive control (MPC) problem on all CAVs in ramp merging scenario have been established. We first formulate a trajectory planning problem, in which the optimization establishes a reference trajectory that ensures CAV safety while maintaining stable and efficient traffic flow. Based on the solution to it, we formulate a multi-vehicle MPC problem containing safety constraints. It is used to perform safety control at a micro level.
2) An algorithm capable of solving the trajectory planning task in a distributed manner is proposed. In this algorithm, CAVs exchange information through V2X and can iteratively solve the problem in a short period of time. We first use the Lagrange dual theory to derive its dual problem. Then, a distributed solving method based on the dual ADMM algorithm is proposed, which can synchronously recover the solution to the original problem while solving the dual problem.
3) Distributed Cooperative Iterative MPC (DCIMPC), a method for parallel, distributed, and rapid solving of multi-vehicle MPC problem, is proposed. At the lower level of the framework, safety constraints turn the problem into a high-dimensional non-convex optimization

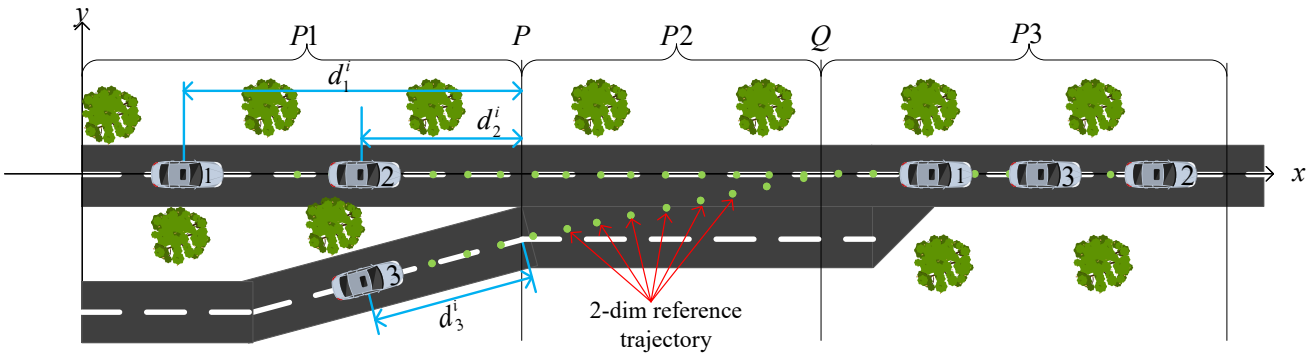


Fig. 1. System Model in Ramp Merging Area

problem. We first propose a decomposition method with the assistance of V2X. Then, we relax and convexify the safety constraints and transforms the problem into low-dimensional convex quadratic programming problems, enabling each CAV to solve the problem in a parallel manner by using its own computing resource while ensuring driving safety.

4) Extensive simulation experiments are conducted to verify the effectiveness of the proposed framework. Moreover, additional simulations demonstrate the effectiveness of DCIMPC in various scenarios. Simulation results show that our distributed trajectory planning algorithm outperforms other distributed methods in optimization quality, and that our convex reformulation approach significantly reduces computation time [1].

The rest of this paper is organized as follows. System models, the establishment of trajectory planning problems and control problems is introduced in Sec. II. The distributed solving method for trajectory planning problems is presented in Sec. III. Subsequently, the DCIMPC framework for solving multi-vehicle control problems is discussed in Sec. IV. In Sec. V, simulation results are provided and discussed, and some discussions on the feasibility of our algorithms in reality when considering communication are also included in this section. Finally, the paper is concluded in Sec. VI.

Notations: Throughout this paper, for any vector X and diagonal matrix Q, $\|X\|^2$ and $\|X\|_Q^2$ denoted as X^TX and X^TQX, respectively. $\langle X, Y\rangle$ denotes the inner product of vector X and Y. R^a and $R^{a\times b}$ denotes the set of a real-valued vectors and $a\times b$ real-valued matrix, respectively. $\frac{\partial F}{\partial X}$ denotes the the partial derivative of X to F. E^k and 0^k denotes the identity matrix and zero matrix of k, and $1^{a\times b}$ denotes the matrix of size $a\times b$ with all elements equal to 1. $\otimes$ denotes Kronecker product. For any matrix M, $M[a:]$, $M[:b]$, and $M[a:b]$ represent the submatrices of M corresponding to the elements from the a row to the last row, from the first row to the b row, and from the a row to the b row, respectively. For any vector x, $diag\{x\}$ denotes the a diagonal matrix with x as its diagonal elements.

[1]The code of this paper can be found at https://github.com/qiongwu86/V2X-Assisted-Distributed-Computing-and-Control-Framework-for-Connected-and-Automated-Vehicles.git

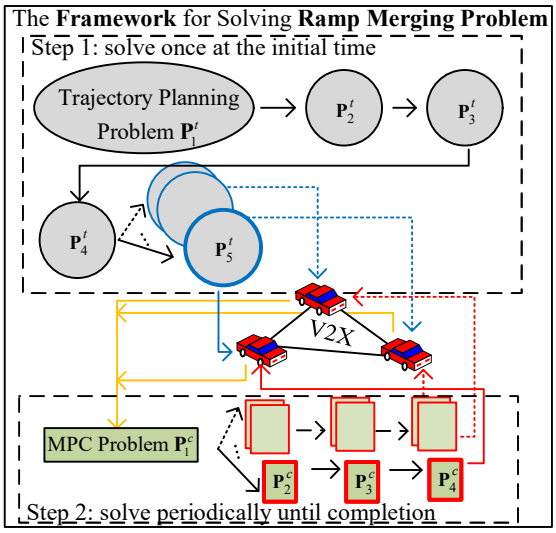


Fig. 2. The framework proposed in this paper.

II. System Model

We will introduce the system model in this section. In the first subsection, we will describe the multi-vehicle ramp merging problem and the overview of the distributed computing and control framework proposed in this paper, which solves the problem in two steps. Then, the centralized optimization problems corresponding to these two steps will be formulated in Sec. II-B and Sec. II-C, respectively.

A. Ramp Merging Problem

Consider a ramp merging scenario that contains a number of CAVs and all of them can communicate with each other by V2X as shown in Fig. 1. The CAVs, comming from both the main road and the ramp, will communicate to each other and utilize its limited computing resource to perform distributed control to complete the merging process and pass through the ramp merging area quickly while ensuring safety. Additionally, it should be noted that this paper does not consider the impact of communication on the algorithm. That is, CAVs can synchronously receive information transmitted by others via V2X without delay or loss, and the algorithm can achieve synchronization through V2X when needed. Furthermore, in

Sec. V-C, we discuss the realistic feasibility of the algorithm under the consideration of communication impacts.

Specifically, the entire ramp merging scenario is divided into three regions: $P1$, $P2$, and $P3$. CAVs are spawned in region $P1$, then pass through region $P2$, and finally enter region $P3$ to complete the ramp merging. To enhance traffic performance for such a multi-vehicle control problem, it is essential to first maintain traffic flow stability at the macroscopic level to achieve smooth merging. Secondly, at the microscopic level, it is necessary to consider the kinematic model and ensure driving safety. Therefore, we divide the entire ramp merging problem into two steps, as illustrated in Fig. 2.

In the first step, we formulate a centralized multi-vehicle trajectories planning problem $\mathbf{P}_1^t$ at the macroscopic level in Sec. II-B, which is established and solved once at the initial moment of the merging process. This problem focuses on optimizing traffic flow stability and determines the merging order of CAVs as well as their terminal states. And its solutions will be transformed to reference trajectories used by step 2.

In the second step, we formulate a multi-vehicle MPC problem $\mathbf{P}_1^c$ at the microscopic level in Sec. II-C, based on the reference trajectory generated in the first step. The safety constraints couple the MPC problems of all CAVs, which makes the original problem $\mathbf{P}_1^c$ require centralized computation.

To enable distributed computing and control, these two problems will be transformed to $\mathbf{P}_5^t$ and $\mathbf{P}_4^c$ respectively and solved with the methods proposed later in this paper, leveraging V2X.

B. Trajectory Planning Problem

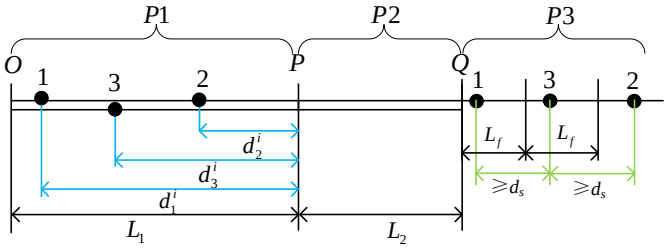


Fig. 3. 1-dim abstract model of ramp merging area.

The one-dimensional longitudinal CAV movement model used in this section is a first-order lag model [26], [27], which can be expressed as $\dot{X}^i(t) = AX^i(t) + BU^i(t)$, and

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/T_l \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1/T_l \end{bmatrix}, \tag{1}$$

where T_l represents the time constant of the lag, the superscript i denotes the CAV index, and there are a total of N CAVs. The state variables X^i and control variable U^i are defined as

$$X^i(t) = \begin{bmatrix} s^i(t) \\ v^i(t) \\ a^i(t) \end{bmatrix}, U^i(t) = \begin{bmatrix} a^i_{ref}(t) \end{bmatrix}, \tag{2}$$

where $s^i(t)$, $v^i(t)$, and $a^i(t)$ represent the longitudinal position, velocity, and acceleration of the CAV i, respectively, and $a^i_{ref}(t)$ is the input reference acceleration. Assuming a sampling time interval of T_s is adopted, the kinematic model is turn into a discrete mode using zero-order hold (ZOH) technique as

$$X^i_{k+1} = \bar{A}X^i_k + \bar{B}U^i_k, \tag{3}$$

where $\bar{A}$ and $\bar{B}$ are the discretized system matrix and input matrix, respectively. In addition, $k \in \left[0, T_P^1 - 1\right]$ is the discrete time index, and T_P^1 is the time steps for trajectory planning. The CAVs are projected onto a one-dimensional longitudinal coordinate axis, as shown in Fig. 3. Assuming the initial state of CAV i is X_0^i, and the order of entry into area $P3$ is denoted by $\mathcal{S}_i$, an integer ranging from 1 to N indicating the priority sequence. For safety and stability, all CAVs should be at safe longitudinal positions when entering area $P3$, which is constrained by the longitudinal position at the termination time T_P^1 as

$$(L_1 + L_2) + (\mathcal{S}_i - 1)L_f \leq s^i_{T_p} \leq (L_1 + L_2) + \mathcal{S}_i L_f, \tag{4}$$

where L_1 and L_2 represent the longitudinal lengths of areas $P1$ and $P2$, and L_f denotes the range of longitudinal positions for each CAV at time step T_P^1, as depicted in Fig. 3.

CAV i should always maintain a safe distance from the CAVs in front and behind it in the same lane. However, the relationships between the front and rear CAVs of CAV i change after merging. For example, $V1$, $V2$ and $V3$ represent three CAVs, which are spawned on region $P1$ and we label them with number on its icon in Fig. 1. (Here, we use three CAVs as an example to illustrate how the trajectory planing problem is formulated. In Sec. V, we will conduct simulations with a larger number of CAVs.) Before the merging of $V3$, $V1$ needs to maintain a distance from $V2$, but after merging, the sequence will become $V1 \rightarrow V3 \rightarrow V2$, which is shown in Fig. 3. To establish these time-varying constraints, the merging process of CAV i is simplified to a moment t_M^i, and the merging process is constrained within the range of $P2$ by establishing constraints on the average speed within $k \in \left[0, t_M^i\right]$ as

$$\frac{L_1 - s_0^i}{T_s} \leq \sum_{k=1}^{t_M^i} v_k^i \leq \frac{L_1 + L_2 - s_0^i}{T_s}. \tag{5}$$

In Eq. (5), the left hand side of the inequality represents the minimum average speed of CAV i at $k = t_M^i$ when reaching the right side of point P, and the right hand side represents the maximum average speed without passing point Q. Considering that the speeds of each CAV should be approximately equal after the planning to ensure the stability of the platoon, it is assumed that the time intervals t_M^i of each CAV are the same, calculated as

$$t_M^i = T_p - c_1(\mathcal{S}_i - 1) - c_0, \tag{6}$$

where c_0 is the number of time steps from the last CAV merging to passing point Q, and c_1 is a constant interval for each CAV t_M^i.

Next, safety constraints between CAVs are established based on t_M^i. When $k \leq t_M^i$, CAV i should maintain a safe distance d_S^1 from its front CAV, denoted by I_i^P, in the same lane.

However, when $k > t_M^i$, CAV i should maintain a safe distance from the front CAV in the merging order, denoted by I_i^Q. Therefore, the safety constraints can be expressed as

$$\begin{cases} d_S^1 \le s_k^{I_i^P} - s_k^i & k \le t_M^i \\ d_S^1 \le s_k^{I_i^Q} - s_k^i & k > t_M^i \end{cases}. \tag{7}$$

Finally, the trajectory planning problem $\mathbf{P}_1^t$ with safe constraint of all CAV in ramp merging is formulated as

$$\mathbf{P}_1^t: \begin{array}{c} \min\limits_{U_k^i,\, i\in\mathcal{N},\, k\in[0,T_P^1-1]} \sum_{i=1}^{N}\sum_{k=0}^{T_P^1-1}(U_k^i)^2 \\ \text{s.t.} \quad a_{\min} \le U_i^k \le a_{\max}, \\ (3),\ (4),\ (5),\ (7), \\ \forall i \in [1,N],\ k \in [0, T_P^1 - 1], \end{array} \tag{8}$$

where $a_{\min}$ and $a_{\max}$ represent the lower and upper limits of the control quantity.

The solution of problem in Eq. (8) represents a one-dimensional longitudinal trajectory. Considering the particularity of the merging area, we adopts a simple method to generate a two-dimensional reference trajectory. As shown in the green origin in Fig. 1, parametric equations with respect to the longitudinal position s_k^i are established along the lane center of the main road and the ramp from $P1$ to $P3$. They are denoted as

$$P_{main}(s) = \begin{bmatrix} x = f_x^1(s) \\ y = f_y^1(s) \end{bmatrix}, P_{merge}(s) = \begin{bmatrix} x = f_x^1(s) \\ y = f_y^1(s) \end{bmatrix}. \tag{9}$$

After solving Eq. (8) to obtain U^{i*}, CAV i first generates X_k^i for $k \in [0, T_P^1]$ based on Eq. (3), then take out their first dimension to form the sequence $\begin{bmatrix} s_0^i & s_1^i & \cdots & s_{T_P^1}^i \end{bmatrix}^T$, and input it into $P_{main}(s)$ or $P_{merge}(s)$ to obtain a two-dimensional reference trajectory with respect to the sampling points. As shown in Fig. 1, the green sampling points distributed along the road center marked by red arrows represent the two-dimensional reference trajectory. It is important to note that the two-dimensional reference trajectory at this point can only ensure the longitudinal safety of CAVs in areas $P1$ and $P3$, but not in the merging area, i.e., area $P2$.

C. Multi-Vehicle MPC Considering Safety Constraints

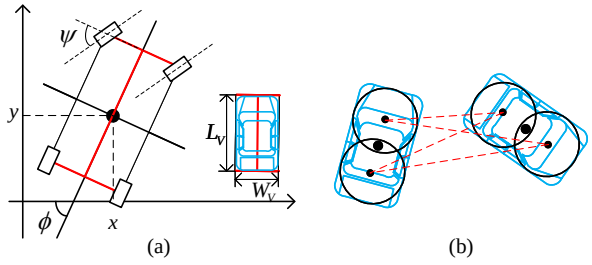


Fig. 4. Kinematic Model (a) and Circular Approximation (b)

This subsection establishes the CAV control problem based on the solution of trajectory planning problem formulated last subsection, further ensuring driving safety. The kinematic model of the CAV can be represented as $\dot{X} = f_S(X, U)$ [28]. The state variable X, control variable U, and system function f_S are defined as

$$X = \begin{bmatrix} x \\ y \\ \phi \\ v \end{bmatrix}, U = \begin{bmatrix} a \\ \psi \end{bmatrix}, f_S(X,U) = \begin{bmatrix} v\cos(\phi+\beta) \\ v\sin(\phi+\beta) \\ v\frac{\sin(\beta)}{0.5\times L_V} \\ a \end{bmatrix}, \tag{10}$$

and β is defined in [28] as $\beta = \arctan(0.5 \times \tan(\psi))$. In Eq. (10), x and y represent the lateral and longitudinal coordinates of the CAV's center point, ϕ is the heading angle, v is the speed scalar, a is the CAV's acceleration, and ψ is the front wheel steering angle as shown in Fig. 4 of the CAV kinematic model. Let k be the discrete sampling point index, then we have $X^i(k \times T_S) = X_k^i$. By applying forward differencing to kinematic model, the discretization yields

$$\dot{X}_k^i = f_S(X_k^i, U_k^i) \approx \frac{X_{k+1}^i - X_k^i}{T_S}, \tag{11}$$

and the discrete system function is given by:

$$X_{k+1}^i = f_d(X_k^i, U_k^i) = T_S f_S(X_k^i, U_k^i) + X_k^i. \tag{12}$$

In MPC, the controller solves an optimization problem at time step k to compute the next T_P^2 control inputs, but only applies the first input to the system. When reaching time step $k+1$, the prediction horizon is shifted by one sampling interval and the process is repeated. The symbols $U_{(k+l,k)}^i$, $l \in [0, T_P^2 - 1]$, represent the subsequent T_P^2 control variables to be solved at time k. Similarly, $X_{(k+l,k)}^i$, $l \in [1, T_P^2]$, represent the subsequent T_P^2 states predicted based on $U_{(k+l,k)}^i$. Furthermore, for consistency in notation, the state of CAV i at time k is denoted as $X_{(k,k)}^i$, which is a known and determined vector at time k.

Based on the characteristics of MPC, safety constraints are introduced into the predicted trajectories of CAVs. This means that for $\forall l \in [1, T_P^2]$ and $(i,j) \in \{(a,b) : a,b \in \mathcal{N} \bigcap a \neq b\}$, there should be a certain distance constraint satisfied between $X_{(k+l,k)}^i$ and $X_{(k+l,k)}^j$. Therefore, the MPC problem $\mathbf{P}_1^c$ with safety constraints at time k can be represented as,

$$\begin{aligned} \min_{\substack{U_{(k+l,k)}^i, i\in\mathcal{N} \\ l\in[0,T_P^2-1]}} & \sum_{i\in\mathcal{N}} \left[\sum_{l=1}^{T_P^2-1} \left\| M_f \left(X_{(k+l+1,k)}^i - X_{(k+l,k)}^i \right) \right\|^2 \right. \\ & \left. + \sum_{l=1}^{T_P^2} \left\| X_{(k+l,k)}^i - \tilde{X}_{(k+l,k)}^i \right\|_{Q_X}^2 + \left\| U_{(k+l-1,k)}^i \right\|_{Q_U}^2 \right] \\ \mathbf{P}_1^c: \quad & \text{s.t. } \underline{U} \le U_{(k+l,k)}^i \le \overline{U}, && \text{(13a)} \\ & X_{(k+l+1,k)}^i = f_d \left(X_{(k+l,k)}^i, U_{(k+l,k)}^i \right), && \text{(13b)} \\ & D_s \le D \left(X_{(k+l+1,k)}^i, X_{(k+l+1,k)}^j \right), && \text{(13c)} \\ & \forall i \in \mathcal{N}, l \in \left[0, T_P^2 - 1\right], j \in \mathcal{N}_i. \end{aligned}$$

In the objective in Eq. (13), the first item represents the rate of state changing. This item can stabilize the system, where the weight of it is controlled by the matrix M_f. The second item is to measure the difference between real-time

state and the reference trajectory within the subsequent T_P^2 prediction horizon, controlled by the symmetric matrix Q_X. The third item is a cost function related to control variables, controlled by the symmetric matrix Q_U. Eq. (13a) pertains to the constraints on control variables, where $\overline{U}$ and $\underline{U}$ represent the upper and lower limits of control variables, respectively. Eq. (13b) involves discretized dynamic constraints. Eq. (13c) is safety constraints, where $\mathcal{N}_i$ denotes CAVs in $\mathcal{N}$ excluding i. Safety constraints are established based on the trajectories predicted by MPC for each CAV, and $D(\cdot,\cdot)$ in Eq. (13c) is a function used to measure the distance between trajectories [28], as shown on the right side of Fig. 4. Specifically, the formula for function $D(\cdot,\cdot)$ is

$$\begin{aligned} &D(X^i_{(k+l,k)}, X^j_{(k+l,k)}) \\ &= \begin{bmatrix} \dot{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, 1, 1) \\ \dot{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, 1, -1) \\ \dot{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, -1, 1) \\ \dot{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, -1, -1) \end{bmatrix}, \end{aligned} \tag{14}$$

where $\dot{D}(\cdot,\cdot)$ is defined as the center distance between two circles in CAV [28].

III. Distributed Trajectory Planning

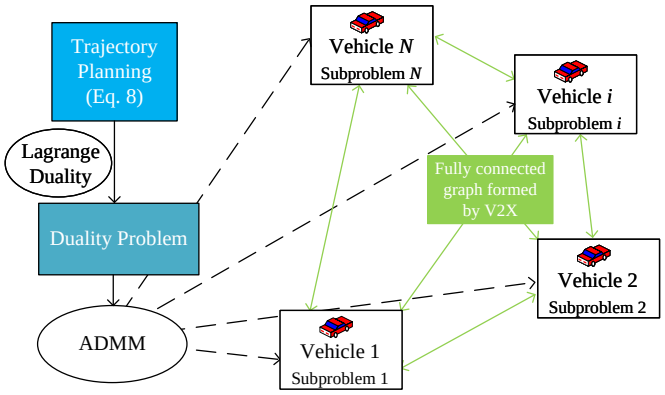


Fig. 5. The solving procedure of Eq. (8)

The processing flow for the trajectory planning problem is illustrated in Fig. 5. We first derive the dual problem of Eq. (13) based on the Lagrangian duality theory, and then uses a variant of the ADMM algorithm proposed in [29] to decompose the dual problem into sub-problems corresponding to each CAV for distributed iterative solution. Then, CAVs will communicate through V2X and use local computing resources for distributed solving.

A. The dual problem of the trajectory planning problem

For clarity, let the initial time for trajectory planning be 0, denote the initial state vector of CAV i as X_0^i. Then, concatenate the state variables for $t \in [1, T_P^1]$ and the control variables for $t \in [0, T_P^1 - 1]$ together as

$$X^i = \begin{bmatrix} X_1^i \\ X_2^i \\ \vdots \\ X_{T_P^1}^i \end{bmatrix} \in R^{3T_P^1}, U^i = \begin{bmatrix} U_0^i \\ U_1^i \\ \vdots \\ U_{T_P^1-1}^i \end{bmatrix} \in R^{T_P^1}. \tag{15}$$

Thus, the trajectory planning problem Eq. (8) can be rewritten as

$$\mathbf{P}_2^t : \begin{aligned} &\min_{U^i, X^i, i\in\mathcal{N}} \sum_{i\in\mathcal{N}} F^i(U^i, X^i) \\ &\text{s.t.} \sum_{i\in\mathcal{N}} (G^i X^i - H^i) \in \mathcal{K}, \end{aligned} \tag{16}$$

where $F_i(U_i, X_i)$ is the local objective function for CAV i, defined as

$$F^i(U^i, X^i) = \left\|U^i\right\|^2 + \mathcal{I}_{C^i}(U^i, X^i). \tag{17}$$

$\mathcal{I}_{C^i}(U^i, X^i)$ in Eq. (17) is an indicator function, which is defined as

$$\mathcal{I}_{C^i}(U^i, X^i) = \begin{cases} 0, & (U^i, X^i) \in C^i, \\ \infty, & \text{else.} \end{cases} \tag{18}$$

By this function, each CAV's own constraint conditions are incorporated into its objective function F^i. Therefore, in Eq. (17), the set C^i is the feasible domain of the control variables U^i and state variables X^i for the other constraints in Eq. (8), except for the safety constraints, i.e.,

$$C^i := \left\{ (U^i, X^i) : \begin{matrix} \forall k \in [0, T_P^1 - 1], \underline{U} \le U_k^i \le \overline{U} \\ (3) \cap (4) \cap (5) \end{matrix} \right\}. \tag{19}$$

The matrix $G_i \in R^{NT_P \times 3T_P}$, $H_i \in R^{NT_P}$, and the convex cone $\mathcal{K}$ are used to express the safety constraints constructed in Eq. (7). Specifically, some elements in G^i satisfy

$$G^i[iT_P^1 : (i+1)T_P^1] = \begin{bmatrix} -1 & 0 & 0 \end{bmatrix} \otimes E^{T_P^1}, \tag{20a}$$

$$G^i[J_i^P T_P^1 : J_i^P T_P^1 + t_{J_i^P}^M] = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \otimes E^{T_P^1}[: t_{J_i^P}^M], \tag{20b}$$

$$\begin{aligned} &G^i[J_i^Q T_P^1 + t_{J_i^Q}^M : (J_i^Q + 1)T_P^1] \\ &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \otimes E^{T_P^1}[t_{J_i^Q}^M :], \end{aligned} \tag{20c}$$

with all others being 0 elements. Eq. (20a) extracts the first dimension (longitudinal coordinate) of X_i, which is used to express the safety constraints when CAV i is the following CAV. Eq. (20b) and Eq. (20c) are used to express the safety constraints for CAV i when it is the leading CAV. Specifically, as mentioned in Sec. II-C, CAV i could either be the leading CAV before a car passes point P or the leading CAV after another car passes point Q. We use J_i^P and J_i^Q to represent the numbers of these two CAVs. Then, in equation Eq. (16), H^i satisfies

$$H^i = \begin{bmatrix} 0^{1\times(i-1)T_P^1} & d_s & \cdots & d_s & 0^{1\times(N-i)T_P^1} \end{bmatrix}^T. \tag{21}$$

The convex cone $\mathcal{K}$ is a positive quadrant cone on the space $R^{NT_P^1}$, denoted as $R^{NT_P^1+}$. Considering that Eq. (3) is a linear time-invariant system, the dynamic constraints can be

expressed in terms of X_i and U_i to simplify the expression [30]. Due to the recursive nature of the system dynamics,

$$\begin{aligned} X_1^i &= \bar{A}X_0^i + \bar{B}U_0^i, \\ X_2^i &= \bar{A}X_1^i + \bar{B}U_1^i = \bar{A}^2X_0^i + \bar{A}\bar{B}U_0^i + \bar{B}U_1^i, \\ &\vdots \\ X_{T_P}^i &= \bar{A}^{T_P^1}X_0^i + \sum_{k=0}^{T_P^1-1} \bar{A}^{(T_P^1-1-k)}\bar{B}U_k^i. \end{aligned} \tag{22}$$

Therefore, it follows that

$$X^i = \Gamma X_0^i + \Lambda U^i, \tag{23}$$

where

$$\Gamma = \begin{bmatrix} \bar{A} \\ \bar{A}^2 \\ \vdots \\ \bar{A}_P^{T_P^1} \end{bmatrix}, \Lambda = \begin{bmatrix} \bar{B} & 0 & \cdots & 0 \\ \bar{A}\bar{B} & \bar{B} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{A}^{T_P^1-1}\bar{B} & \bar{A}^{T_P^1-2}\bar{B} & \cdots & \bar{B} \end{bmatrix}. \tag{24}$$

Substituting Eq. (23) into Eq. (16), we obtain the safety constraint expressed in terms of U^i as

$$\sum_{i\in\mathcal{N}} A^iU^i - b^i \in \mathcal{K}, \tag{25}$$

where

$$A^i = G^i\Lambda, \tag{26a}$$
$$b^i = H^i - G^i\Gamma X_0^i. \tag{26b}$$

Thus, the optimization problem in Eq. (16) can be rearranged as:

$$\mathbf{P}_3^t: \begin{aligned} &\min_{U^i, i\in\mathcal{N}} \sum_{i\in\mathcal{N}} \mathcal{F}^i(U^i) + \mathcal{I}_{\mathcal{K}}(\omega) \\ &\text{s.t.} \quad \sum_{i\in\mathcal{N}} (A^iU^i - b^i) = \omega, \end{aligned} \tag{27}$$

where $\mathcal{F}^i(U^i) = F^i(U^i, X^i) = F^i(U^i, \Gamma X_0^i + \Lambda U^i)$. Given that Γ, Λ, and X_0^i are constants, the final $\mathcal{F}^i$ is a function of U^i. The optimization problem Eq. (27) is convex and satisfies the Slater condition. According to the Lagrange duality theorem, the Lagrangian function of Eq. (27) is given by:

$$\mathcal{L} = \sum_{i\in\mathcal{N}} \mathcal{F}^i(U^i) + \mathcal{I}_{\mathcal{K}}(\omega) + \left\langle y, \sum_{i\in\mathcal{N}} (A^iU^i - b^i) - \omega \right\rangle, \tag{28}$$

where $y \in R^{NT_P^1}$ represents the dual variable. Then, the Lagrangian dual function is given by

$$\begin{aligned} d_{\mathcal{L}}(y) &= \inf_{(U^i,\omega)} \mathcal{L} \\ &= \underbrace{\inf_{U^i} \left\{ \sum_{i\in\mathcal{N}} \mathcal{F}^i(U^i) + \left\langle y, A^iU^i \right\rangle \right\}}_{a} \\ &\quad + \underbrace{\inf_{\omega} \{\mathcal{I}_{\mathcal{K}}(\omega) - \langle y, \omega\rangle\}}_{b} - \sum_{i\in\mathcal{N}} \left\langle y, b^i \right\rangle, \end{aligned} \tag{29}$$

In Eq. (29), a can be expressed in the form of a convex conjugate [31]. For a function $f(x)$, its convex conjugate is defined as $f^*(y) := \sup_x \{y^Tx - f(x)\}$ Thus, we have:

$$\begin{aligned} a &= -\inf_{U^i} \left\{ -\sum_{i\in\mathcal{N}} \mathcal{F}^i(U^i) + (-y^TA^i)U^i \right\} \\ &= -\sum_{i\in\mathcal{N}} \left\{ \inf_{U^i} \left(-A^{iT}y\right)^T U^i - \mathcal{F}^i(U^i) \right\} \\ &= -\sum_{i\in\mathcal{N}} \mathcal{F}^{i*}(-A^{iT}y), \end{aligned} \tag{30}$$

As for b in Eq. (29), considering the presence of the indicator function $\mathcal{I}_{\mathcal{K}}$, if ω is not in $\mathcal{K}$, this term results in infinity. Therefore, it must hold that $\omega \in \mathcal{K}$. Hence,

$$\begin{aligned} b &= \inf_{\omega} \{\mathcal{I}_{\mathcal{K}}(\omega) - \langle y, \omega\rangle\} \\ &= \inf_{\omega\in\mathcal{K}} \{\underbrace{\mathcal{I}_{\mathcal{K}}(\omega)}_{0} - \langle y, \omega\rangle\} \\ &= -\mathcal{I}_{\mathcal{K}^\circ}(y). \end{aligned} \tag{31}$$

The last equality comes from [32], where $\mathcal{K}^\circ$ denotes the polar cone of $\mathcal{K}$. Consequently, the Lagrangian dual problem of Eq. (27) can be formulated as

$$\mathbf{P}_4^t: \begin{aligned} &\max_y d_{\mathcal{L}}(y) \\ =&\max_y \left\{ a + b - \sum_{i\in\mathcal{N}} \left\langle y, b^i \right\rangle \right\} \\ =&\max_y \left\{ \sum_{i\in\mathcal{N}} \left[- \left\langle y, b^i \right\rangle - \mathcal{F}^{i*}(-A^{iT}y) \right] - \mathcal{I}_{\mathcal{K}^\circ}(y) \right\} \\ =&\max_y \left\{ \sum_{i\in\mathcal{N}} \left[- \left\langle y, b^i \right\rangle - \mathcal{F}^{i*}(-A^{iT}y) - \mathcal{I}_{\mathcal{K}^\circ}(y) \right] \right\}. \end{aligned} \tag{32}$$

The equality in Eq. (32) holds due to $\mathcal{I}_{\mathcal{K}^\circ}(y) = C \times \mathcal{I}_{\mathcal{K}^\circ}(y)$, where C is an arbitrary constant.

B. Decomposition and solve the dual problem

The optimization problem of Eq. (32) is a decomposed consensus optimization problem with all agents making the joint decision on a common optimization variable y. Because each CAV's objective function can be decomposed into two parts, i.e., $d_{\mathcal{L}}^{i,1}$ and $d_{\mathcal{L}}^{i,2}$,

$$d_{\mathcal{L}}(y) = \sum_{i\in\mathcal{N}} \underbrace{-[\left\langle y, b^i \right\rangle + \mathcal{F}^{i*}(-A^{iT}y)]}_{d_{\mathcal{L}}^{i,1}} + \underbrace{[-\mathcal{I}_{\mathcal{K}^\circ}(y)]}_{d_{\mathcal{L}}^{i,2}}. \tag{33}$$

Therefore, we adopt Alg. 1 in [33] to solve it.

Alg. 1 needs two hyperparameters, i.e., σ and ρ, which is similar to step size in optimization algorithms. In line 1, variable k is used to mark the number of iterations and each CAV initializes local variables, i.e. $y^{i,k}$, $p^{i,k}$, and $s^{i,k}$. They are typically initialized as zero vectors. $p^{i,k}$ and $s^{i,k}$ are the local variables of CAV i. They are updated at each iteration and then used as parameters in the two optimization problems in line 6 and 7. $y^{i,k}$ is the local optimization variables of CAV i at time step k. As the iterations progress, $y^{i,k}$ of all users will gradually converge to the same vector, forming a consensus.

At the beginning of each iteration, CAVs need to exchange local optimization variables $y^{i,k}$ with each other (line 3). Then, calculate $p^{i,k+1}$ and $s^{i,k+1}$ in line 4 and 5, respectively. After that, CAVs need to solve two problem in line 6 and 7. According to [34], the stopping criterion for Alg. 2 can be set to a fixed number of iterations while ensuring that the variance of $y^{i,k}$ remains below a specified threshold.

Algorithm 1: Consensus ADMM

Input: Hyper parameters $\sigma > 0$ and $\rho > 0$

1 initialization $y^{i,0}$, $p^{i,0}$ and $s^{i,0} = 0$;

2 **do**

3 exchange $y^{i,k}$ between all vehicles;

4 $p^{i,k+1} \leftarrow p^{i,k} + \rho \sum_{j \in \mathcal{N}_i} (y^{i,k} - y^{j,k})$;

5 $s^{i,k+1} \leftarrow s^{i,k} + \sigma(y^{i,k} - z^{i,k})$;

6 $y^{i,k+1} \leftarrow$

$$\underset{y}{argmin} \left\{ \begin{array}{l} \frac{\sigma}{2}\left\|y - z^{i,k}\right\|^2 + \left\langle y, p^{i,k+1} + s^{i,k+1}\right\rangle \\ \rho \sum\limits_{j \in \mathcal{N}_i} \left\| y - \frac{y^{i,k} + y^{j,k}}{2} \right\|^2 + d_{\mathcal{L}}^{i,1}(y) \end{array} \right\};$$

7 $z^{i,k+1} \leftarrow$

$$\underset{z}{argmin} \left\{ d_{\mathcal{L}}^{i,2}(z) - \left\langle z, s^{i,k+1}\right\rangle + \tfrac{\sigma}{2}\left\|z - y^{i,k+1}\right\|^2 \right\};$$

8 **while** *until termination criteria*;

When using Alg. 1, each CAV cannot directly obtain the original problem's optimization variable U^i, and the optimization problems in line 6 and 7 are complicated and difficult to solve. Therefore, **Proposition 2** from [32] is used. It states that when solving the problem in line 6, the original problem's optimization variable U^i for this iteration can be obtained simultaneously . Furthermore, as the iterations progress, U^i will gradually converge to the solution of the original problem. To use the conclusion, we consolidate the first three terms of the optimization objective in line 6 of Alg. 1 and eliminate the irrelevant constant terms and we get

$$\arg\min_{y} \left\{ d_{\mathcal{L}}^{i,1}(y) + \frac{\sigma + 2\rho d_i}{2} \left\| y - \frac{r^{i,k+1}}{\sigma + 2\rho d_i} \right\|^2 \right\}, \tag{34a}$$

$$\begin{aligned} r^{i,k+1} = {} & \sigma z^{i,k} + \rho \sum_{j \in \mathcal{N}_i} (y^{i,k} + y^{j,k}) \\ & - (b^i + p^{i,k+1} + s^{i,k+1}), \end{aligned} \tag{34b}$$

where d_i represents the degree of CAV i in the fully connected graph formed by the vehicular network. Therefore, for $\forall i \in \mathcal{N}$, $d_i = (N-1)$. The term b_i is derived from Eq. (26b). According to [32], the solution to problem Eq. (34a) is given by

$$y = \frac{A^i U^{i,k+1} + r^{i,k+1}}{\sigma + 2\rho d_i}, \tag{35a}$$

$$U^{i,k+1} = \arg\min_{U} \left\{ \mathcal{F}^i(U) + \frac{\left\| A^i U + r^{i,k+1} \right\|^2}{2(\sigma + 2\rho d_i)} \right\}. \tag{35b}$$

where A^i and $\mathcal{F}^i$ are from Eq. (26a) and Eq. (27), respectively.

Next, expanding the optimization problem in Eq. (35b), we get

$$\begin{aligned} \mathbf{P}_5^t : \quad & \min_{U^i} \quad U^{iT}(Q \otimes E^{T_P^1} + \frac{1}{2(\sigma + 2\rho d_i)} A^{iT} A^i) U^i \\ & \qquad + 2\frac{1}{2(\sigma + 2\rho d_i)} (A^{iT} r)^T U^i \\ & \text{s.t.} \quad U_l^i \preceq \begin{bmatrix} E^{2T_P^1} \\ M_X^i \Lambda \end{bmatrix} U^i \preceq U_u^i, \end{aligned} \tag{36}$$

where

$$U_l^i = \begin{bmatrix} \underline{U} \otimes 1^{T_P^1} \\ \begin{bmatrix} (L_1 + L_2) + (\mathcal{S}_i - 1) L_f \\ \left(L_1 - s_0^i\right) / T_s \end{bmatrix} - M_X^i \Gamma X_0^i \end{bmatrix}, \tag{37a}$$

$$U_u^i = \begin{bmatrix} \overline{U} \otimes 1^{T_P^1} \\ \begin{bmatrix} (L_1 + L_2) + \mathcal{S}_i L_f \\ \left(L_1 + L_2 - s_0^i\right) / T_s \end{bmatrix} - M_X^i \Gamma X_0^i \end{bmatrix}, \tag{37b}$$

$$M_X^i = \begin{bmatrix} 1^{1 \times (T_P^1 - 1)} \otimes 0^{1 \times 3} & \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ 1^{1 \times t_i^M} \otimes \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} & 1^{1 \times (T_P^1 - t_i^M)} \otimes 0^{1 \times 3} \end{bmatrix}. \tag{37c}$$

Because the quadratic term in the objective function of $\mathbf{P}_5^t$ is positive definite, it is a convex quadratic programming and can be solve easily.

The optimization problem in line 7 of Alg. 1 is relatively simple. Considering that $d_{\mathcal{L}}^{i,2}$ is the characteristic function on the set $\mathcal{K}^\circ$, the solution to this optimization problem must lie within the set $\mathcal{K}^\circ$. Since the optimization variable is z, by completing the square of the objective function with respect to z and removing irrelevant constant terms, we can obtain

$$\begin{aligned} z^{i,k+1} &= \arg\min_{z \in \mathcal{K}^\circ} \left\{ \frac{\sigma}{2}\left\| z - y^{i,k+1} \right\|^2 - \left\langle z, s^{i,k+1} \right\rangle \right\} \\ &= \arg\min_{z \in \mathcal{K}^\circ} \left\{ \left\| z - \left(y^{i,k+1} + \frac{s^{i,k+1}}{\sigma} \right) \right\|^2 \right\}. \end{aligned} \tag{38}$$

Since $\mathcal{K}^\circ$ is a convex cone formed by negative limits, truncating each element of the vector $y^{i,k+1} + \frac{s^{i,k+1}}{\sigma}$ on the negative real axis yields the solution

$$z^{i,k+1} = \min \left\{ y^{i,k+1} + \frac{s^{i,k+1}}{\sigma}, 0 \right\}. \tag{39}$$

In summary, the entire trajectory planning algorithm is summarized in Alg. 2. The communication between CAVs in lines 4 can be implemented using V2X. After the iterations in lines 3 to 11 are completed, CAV i obtains the control input $U^{i,*}$, from which the final reference trajectory is generated using Eq. (23).

IV. Distributed Cooperative Iterative MPC

As described in Sec. II-C, due to the safety constraints in problem Eq. (13c), the cooperative control is introduced but the problems of each CAV are coupled and non-convex, which generates a centralized computationally intensive task and makes it difficult to solve. Therefore, a method for rapidly and parallelly solving this problem is proposed in this section, i.e., distributed cooperative iterative MPC. Sec. IV-A introduces how to decouple the problem into subproblems, and Sec. IV-B explains how to convexify the non-convex subproblems and solve them quickly.

Algorithm 2: Trajectory Planning

Input: Hyper parameters L_1, L_2, c_1, c_0, $\sigma > 0$ and $\rho > 0$

N vehicles transmit their coordinate through **V2X** to confirm their merging sequence $\mathcal{S}^i$;
N vehicles initialize $y^{i,0}$, $p^{i,0}$ and $s^{i,0} = 0$;
do
  Exchange $y^{i,k}$ between all vehicles;
  $p^{i,k+1} \leftarrow p^{i,k} + \rho \sum_{j \in \mathcal{N}_i} (y^{i,k} - y^{j,k})$;
  $s^{i,k+1} \leftarrow s^{i,k} + \sigma(y^{i,k} - z^{i,k})$;
  $r^{i,k+1} \leftarrow \sigma z^{i,k} + \rho \sum_{i \in \mathcal{N}} - \left(b^i + p^{i,k+1} + s^{i,k+1}\right)$;
  Solve problem Eq. (36) to get $U^{i,k+l}$;
  Calculate $y^{i,k+l}$ by Eq. (35a);
  Calculate $z^{i,k+l}$ by Eq. (39);
while until termination criteria;
$U^{i,*} = U^{i,k+1}$;
Input $U^{i,*}$ into Eq. (23) to get reference trajectory;

A. Decomposition of the Coupled Control Problem

In the follwoing, the state variables obtained from the optimization problem for CAV i at time step k are denoted as $\bar{X}^i_{(k+l,k)}$, where $l \in [1, T_P^2]$. These variables concatenated into a single column vector to form nominal trajectory, denoted as $\bar{X}^i_k = [\bar{X}^{iT}_{(k+1,k)}, \bar{X}^{iT}_{(k+2,k)}, \cdots, \bar{X}^{iT}_{(k+T_P^2,k)}]^T$. Since the nominal trajectory contains information about the future trajectory, CAVs can exchange nominal trajectories by V2X and then establish local safety constraints and MPC problems, finally solving it and performing collision avoidance control. In this way, the centralized problem Eq. (13) can be solved in a distributed manner by solving problem $\mathbf{P}_2^c$ on each CAV to obtain a suboptimal solution to the original problem.

$$\begin{aligned}
\min_{\substack{U^i_{(k+l,k)},\\ l \in [0, T_P^2 - 1]}} \quad & \sum_{l=1}^{T_P^2 - 1} \left\| M_f \left(X^i_{(k+l+1,k)} - X^i_{(k+l,k)} \right) \right\|^2 \\
& + \sum_{l=1}^{T_P^2} \left\| X^i_{(k+l,k)} - \tilde{X}^i_{k+l,k} \right\|^2_{Q_X} + \left\| U^i_{(k+l-1,k)} \right\|^2_{Q_U} \\
\mathbf{P}_2^c: \quad \text{s.t. } & \underline{U} \le U^i_{(k+l,k)} \le \overline{U}, && \text{(40a)} \\
& X^i_{(k+l+1,k)} = f_d \left(X^i_{(k+l,k)}, U^i_{(k+l,k)} \right), && \text{(40b)} \\
& D_s \le D \left(X^i_{(k+l+1,k)}, X^j_{(k+l+1,k)} \right), && \text{(40c)} \\
& \forall l \in \left[0, T_P^2 - 1\right], j \in \mathcal{N}_i.
\end{aligned}$$

For problem Eq. (40), its objective function is a positive definite quadratic form. However, Eq. (40b) and Eq. (40c) are both nonlinear and non-convex constraint conditions, which will lead to difficulties in solving. For the nonlinear time-varying CAV kinematic model constraint in Eq. (40b), many works use Taylor series expansions for first-order approximations near the reference trajectory. However, due to the presence of safety constraints in Eq. (40c), the actual driving trajectory that CAVs can avoid collisions will differ significantly from the reference trajectory. In this case, if a first-order Taylor approximation is still applied to the f_d near the reference trajectory, the error will be significant. Therefore, at time k, we use the nominal trajectory obtained at time $k-1$ to perform a first-approximation of the constraint in Eq. (40b). After solving, $\bar{X}^i_k$ is obtained, and this trajectory is then extended by a sampling time and used as the nominal trajectory at time $k+1$ for the next iteration.

Then, for clarity of presentation, the following definitions are made:

$$\begin{aligned}
X^i_k &= \begin{bmatrix} X^{iT}_{(k+1,k)} & X^{iT}_{(k+2,k)} & \cdots & X^{iT}_{(k+T_P^2,k)} \end{bmatrix}^T \in R^{4T_P^2}, \\
U^i_k &= \begin{bmatrix} U^{iT}_{(k,k)} & U^{iT}_{(k+1,k)} & \cdots & U^{iT}_{(k+T_P^2,k)} \end{bmatrix}^T \in R^{2T_P^2}.
\end{aligned} \tag{41}$$

For the discretized nonlinear system function f_d in Eq. (12), first-order approximation is taken at the nominal trajectory $\bar{X}^i_{(k+l,k)}$, $\bar{U}^i_{(k+l,k)}$ as

$$\begin{aligned}
& f_d(X^i_{(k+l,k)}, U^i_{(k+l,k)}) \\
\approx & A^i_{(k+l,k)} X^i_{(k+l,k)} + B^i_{(k+l,k)} U^i_{(k+l,k)} + G^i_{(k+l,k)},
\end{aligned} \tag{42}$$

where

$$\begin{aligned}
A^i_{(k+l,k)} &= T_S \mathcal{J}^f_X \left(\bar{X}^i_{(k+l,k)}, \bar{U}^i_{(k+l,k)} \right) + E, \\
B^i_{(k+l,k)} &= T_S \mathcal{J}^f_U, \\
G^i_{(k+l,k)} &= T_S [f(\bar{X}^i_{(k+l,k)}, \bar{U}^i_{(k+l,k)}) \\
&\quad - \mathcal{J}^f_X \left(\bar{X}^i_{(k+l,k)}, \bar{U}^i_{(k+l,k)} \right) \bar{X}^i_{(k+l,k)} \\
&\quad - \mathcal{J}^f_U \left(\bar{X}^i_{(k+l,k)}, \bar{U}^i_{(k+l,k)} \right) \bar{U}^i_{(k+l,k)}].
\end{aligned} \tag{43}$$

Similar to Eq. (22), the constraints in Eq. (40b) can be organized as linear constraints on the initial state $X^i_{(k,k)}$ and X^i with U^i as

$$X^i_k = A^i_k X^i_{(k,k)} + B^i_k U^i_k + G^i_k, \tag{44}$$

where A^i_k, B^i_k and G^i_k are the coefficient matrices of linear constraints, respectively. However, it should be noted that, due to the time-varying nature of the kinematic, these three matrices will vary with different k. Finally, we can organize Eq. (40) into a form of linear kinematic constraints as

$$\begin{aligned}
\min_{X^i_k, U^i_k} \quad & \left\| X^i_k - \tilde{X}^i_k \right\|^2_{\bar{Q}_X} + \left\| U^i_k \right\|^2_{\bar{Q}_U} + \left\| \bar{M}_f X^i_k \right\|^2 \\
\mathbf{P}_3^c: \quad \text{s.t. } & E^{T_p^2} \otimes \underline{U} \le U^i_k \le E^{T_p^2} \bar{U} && \text{(45a)} \\
& X^i_k = A^i_k X^i_{(k,k)} + B^i_k U^i_k + G^i_k && \text{(45b)} \\
& E^{T_p^2} \otimes D_s \le \mathcal{D}(X^i_k, \bar{X}^j_k), j \in \mathcal{N}_i && \text{(45c)}
\end{aligned}$$

where the function $\mathcal{D}(X^i_k, \bar{X}^j_k) : R^{4T_P^2 \times 4T_P^2} \to R^{4T_P^2}$ is a function formed by concatenating the function D column-wise as

$$\mathcal{D}(X^i_k, \bar{X}^j_k) = \begin{bmatrix} D(X^i_{(k+1,k)}, \bar{X}^j_{(k+1,k)}) \\ D(X^i_{(k+2,k)}, \bar{X}^j_{(k+2,k)}) \\ \cdots \\ D(X^i_{(k+T_P^2,k)}, \bar{X}^j_{(k+T_P^2,k)}) \end{bmatrix}^T. \tag{46}$$

The definitions of $\bar{Q}_X$, $\bar{Q}_U$, and $\bar{M}_f$ are given by

$$\begin{aligned}
\bar{Q}_X &= \left[\begin{array}{c:c} E^{T_P^2-1} \otimes Q_X & \begin{matrix} 0 \\ k_X \bar{Q}_X \end{matrix} \end{array}\right], \\
\bar{Q}_U &= \left[\begin{array}{c:c} E^{T_P^2-1} \otimes Q_U & \begin{matrix} 0 \\ k_U \bar{Q}_U \end{matrix} \end{array}\right], \\
\bar{M}_f &= \begin{bmatrix} -M_f & M_f & & & \\ & -M_f & M_f & & \\ & & & \ddots & \\ & & & & -M_f \quad M_f \end{bmatrix},
\end{aligned} \tag{47}$$

where k_X and k_U are used to constrain the last point of the trajectory in model predictive control, and the elements not labeled in $\bar{M}_f$ are zeros.

B. Convex Reformulation of Subproblem

In this subsection, we will explains how to reformulate problem Eq. (45) into a convex QP problem. For simplify, we define

$$\begin{aligned}
&\tilde{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, p, q) \\
&= \min\left\{\dot{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, p, q) - D_S, 0\right\}.
\end{aligned} \tag{48}$$

Satisfying the safety constraint Eq. (45c) is equivalent to $\forall i \in \mathcal{N}$, $j \in \mathcal{N}_i$, $l \in \left\|1, T_P^2\right\|$ and $p, q \in \{-1, 1\}$,

$$\tilde{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, p, q)^2 \leq 0. \tag{49}$$

Then, we define

$$\mathcal{T}^{i,j}_{k,l} := \begin{bmatrix} \tilde{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, 1, 1) \\ \tilde{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, 1, -1) \\ \tilde{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, -1, 1) \\ \tilde{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, -1, -1) \end{bmatrix}. \tag{50}$$

Therefore, the safety constraint for CAV i with respect to neighboring CAV j at time k is equivalent to

$$\begin{aligned}
\left\|\mathcal{T}^{i,j}_k\right\|^2 &= \left\| \begin{bmatrix} \mathcal{T}^{i,j}_{k,1} \\ \mathcal{T}^{i,j}_{k,2} \\ \vdots \\ \mathcal{T}^{i,j}_{k,T_P^2} \end{bmatrix} \right\|^2 \\
&= \sum_{l=1}^{T_P^2} \sum_{p,q \in \{1,-1\}} \tilde{D}(X^i_{(k+l,k)}, X^j_{(k+l,k)}, p, q)^2 \leq 0.
\end{aligned} \tag{51}$$

Next, this term is used as a soft constraint and included in the objective function. Thus, the objective function becomes

$$\begin{aligned}
\bar{J}^i = &\left\|X^i_k - \tilde{X}^i_k\right\|^2_{\bar{Q}_X} + \left\|U^i_k\right\|^2_{\bar{Q}_U} \\
&+ \left\|\bar{M}_f X^i_k\right\|^2 + \alpha \sum_{j \in \mathcal{N}_i} \left\|\mathcal{T}^{i,j}_k\right\|^2,
\end{aligned} \tag{52}$$

where α is used to control the weight of safety constraint. However, this term in Eq. (52) is still a non-convex function of X^i_k, and it needs to be handled next. First, for the function inside the norm operation, i.e., $\mathcal{T}^{i,j}_k$, a first-order Taylor approximation is taken at the nominal trajectory and the constant term is discarded, resulting in

$$\begin{aligned}
\left\|\mathcal{T}^{i,j}_k\right\|^2 &\approx \left\| \mathcal{T}^{i,j}_k\left(\bar{X}^i_k, \bar{X}^j_k\right) + \left.\frac{\partial \mathcal{T}^{i,j}_k}{\partial X^i_k}\right|_{\bar{X}^i_k, \bar{X}^j_k} \left(X^i_k - \bar{X}^i_k\right) \right\|^2 \\
&= \Bigg\| \underbrace{\left.\frac{\partial \mathcal{T}^{i,j}_k}{\partial X^i_k}\right|_{\bar{X}^i_k, \bar{X}^j_k}}_{k^i} X^i_k \\
&\quad + \underbrace{\left[\mathcal{T}^{i,j}_k\left(\bar{X}^i_k, \bar{X}^j_k\right) - \left.\frac{\partial \mathcal{T}^{i,j}_k}{\partial X^i_k}\right|_{\bar{X}^i_k, \bar{X}^j_k} \bar{X}^i_k\right]}_{b^i} \Bigg\|^2 \\
&= \left\|k^i X^i_k + b_i\right\|^2 \\
&= X^{iT}_k (k^{iT} k^i) X^i_k + 2(k^{iT} b^i)^T X^i_k
\end{aligned} \tag{53}$$

Since Eq. (53) is a quadratic form in terms of X^i_k, and $k^{iT}k^i$ is a positive semi-definite matrix, the objective function Eq. (52) becomes a quadratic convex function, transforming the it into a convex quadratic programming problem. Finally, by incorporating the system constraints into the entire optimization problem and eliminating X^i_k while removing irrelevant constant terms, $\mathbf{P}^c_3$ can be transform to the following standard quadratic programming problem $\mathbf{P}^c_4$,

$$\begin{aligned}
\mathbf{P}^c_4: \quad &\min_{U^i_k} \ J^i = \min_{U^i_k} \ U^{iT}_k \mathcal{P} U^i_k + 2\mathcal{Q}^T U^i_k \\
&\text{s.t.} \ \ E^{T_p^2} \otimes \underline{U} \leq U^i_k \leq E^{T_p^2} \otimes \overline{U},
\end{aligned} \tag{54}$$

where

$$\begin{aligned}
\mathcal{P} &= B^{iT}_k \left(\bar{Q}_X + \alpha \sum_{j \in \mathcal{N}_i} k^{iT} k^i + \bar{M}^T_f \bar{M}_f\right) B^i_k + \bar{Q}_U, \\
\mathcal{Q} &= B^{iT}_k \Bigg[\alpha \sum_{j \in \mathcal{N}_i} k^{iT} b^i - \bar{Q}_X \tilde{X}^i_{(k,k)} \\
&\quad + \left(\bar{Q}_X + \alpha \sum_{j \in \mathcal{N}_i} k^{iT} k^i + \bar{M}^T_f \bar{M}_f\right)\left(A^i_k X^i_{(k,k)} + G^i_k\right)\Bigg].
\end{aligned} \tag{55}$$

In summary, the solution steps for the problem in Eq. (54) are summarized as Alg. 3. It should be note that, a hyper-parameter T_{iter} is introduced, which represents the number of iterations for MPC at each time step. As mentioned Sec. IV-A, the precision of Eq. (54) relys on nominal trajectories. To further improve accuracy, an iterative MPC approach is introduced. This algorithm is more suitable for time-varying kinematic models and safety constraints. At each time step k, CAV i assigns the number of iterations to a variable C (Line 3). Then, the iterative MPC is initiated. At the beginning of the iteration, CAVs exchange their nominal trajectories using V2X (Lines 5 and 6). The system constraints are linearized based on the CAV's nominal trajectory by Eq. (44), and then the Jacobian matrix k^j and vector b^j are computed based on other CAVs' nominal trajectories by Eq. (53). These values are then used to construct the optimization problem of Eq. (55) (Line 7), and the OSQP solver is used to solve it, with the nominal

trajectory as the starting point for it (Line 8) and solved (Line 9) [35]. Next, based on the obtained $U^{i*}_{(k+l,k)}, l \in [0, T_P^2-1]$, a new nominal trajectory is generated (Line 10), and the constant C is decremented by 1 (Line 11). When C reaches 0, the current iteration ends, the first control variable for the final result is executed (Line 13), and the nominal trajectory for the next time step is generated (Line 14).

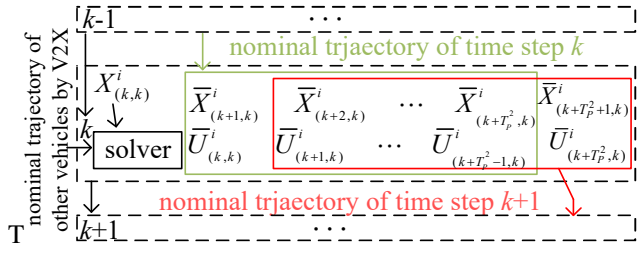


Fig. 6. The variation process of nominal trajectories in DCIMPC

The generation and utilization of nominal trajectories during the solving process are illustrated in Fig. 6. At time k, CAV i uses the nominal trajectories left from time $k-1$ as the initial nominal trajectory (marked by the green rectangle in the Fig. 6). These trajectories are updated at each iteration and extended by a sampling time to generate the nominal trajectory for time $k+1$ for the next use. The extension method is shown in the red box in Fig. 6, where the control variable at $k+T_P^2$ is set to the control variable at $k+T_P^2-1$, then $U^i_{(k+T_P^2,k)}$ and $X^i_{(k+T_P^2,k)}$ are input into the system function Eq. (10) to compute the state at $k+T_P^2+1$, which is combined with previous results to generate the initial nominal trajectory for the next iteration , as shown in Line 15 of Alg. 3. Finally, k is incremented by 1, and the process returns to Line 4 for the next control iteration.

Algorithm 3: DCIMPC of vehicle i

Input : safe factor $\alpha > 0$, iter times T_{iter}, predict length T_P^2

1 $k = 0$, reference trajectory $\tilde{X}^i$ in Alg. 2;
2 **repeat**
3 $C = T_{iter}$;
4 **repeat**
5 Transmit nominal trajectory $\bar{X}^i_k$ of time step k;
6 Receive nominal trajectories $\bar{X}^j_k$, where $j \in \mathcal{N}_i$;
7 Calculate matrixs $\mathcal{P}$ and $\mathcal{Q}$ in Eq. (55);
8 Input $\bar{X}^i_k$ to OSQP for warm start;
9 Solve QP in Eq. (54) and get $U^{i*}_{(k+l,k)}$;
10 Generate $\bar{X}^i_k$ based on $U^{i*}_{(k+l,k)}$;
11 $C \leftarrow C-1$;
12 **until** $C = 0$;
13 Execute $U^{i*}_{(k,k)}$;
14 Generate $\bar{X}^i_{k+1}$ based on $\bar{X}^i_k$;
15 $k \leftarrow k+1$
16 **until** *driving termination*;

V. Simulation Results

In Sec. V-A, we conduct simulations for the on-ramp merging scenario. Specifically, we first generate a longitudinal reference trajectory using Alg. 2, then construct a 2-dim reference trajectory based on the road geometry and feed it into Alg. 3. Subsequently, we run Alg. 3 to perform a complete simulation of multi CAVs control in the on-ramp merging scenario.

In addition, considering the original optimization problem is approximated and decomposed in Alg. 3, we conduct supplementary simulations in Sec. V-B to further evaluate the performance of DCIMPC. Finally, in Sec. V-C, we discuss the practical feasibility of the proposed algorithm under communication constraints.

The simulations in this work are based on the Python and the open-source symbolic computation library CasADi [36]. In addition, it should be noted that in our simulations, we also assume ideal V2X communication conditions, without considering the effects of communication delays, errors, or other imperfections on the algorithm's performance.

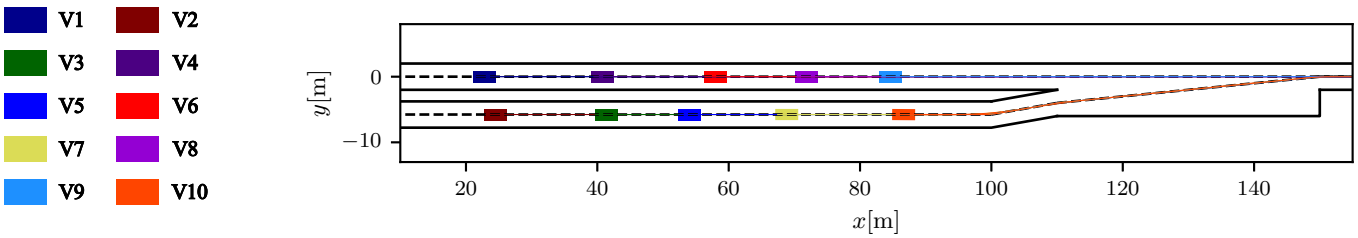
CasADi library is capable of computing algebraic differentials and integrates various optimization solvers such as OSQP, IPOPT, and others [35], [37]. The hardware platform used is the AMD Ryzen 7 6800H running at a clock speed of 3.2-4.2GHz. In the experiments, efforts are made to maintain a consistent ambient temperature to prevent CPU throttling from affecting computational performance. The parameters are listed in Tab. I. Additionally, M_f is equal to $\left[diag\{1, 0.3\}^T, 0^{2\times 2}\right] \in R^{2\times 4}$.

TABLE I
Simulation Parameters

parameters	values	parameters	values
T_S	0.1s	L_1	110m
L_2	40m	T_P^1	90
T_P^2	30	L_V	3.5m
W_V	1.7m	c_0	3.0s
c_1	0.6s	Q_X	$diag\{1, 1, 0.0\}$
k_X	10	Q_U	$diag\{1, 0.1\}$
$\overline{U}$	$[7.0, 34°]^T$	$\underline{U}$	$[-7.0, -34°]^T$
d_S^l	$10m$	d_S^2	2.5m
α	8	T_{iter}	3
T_a	0.1s	road width	4.5m
k_U	10		

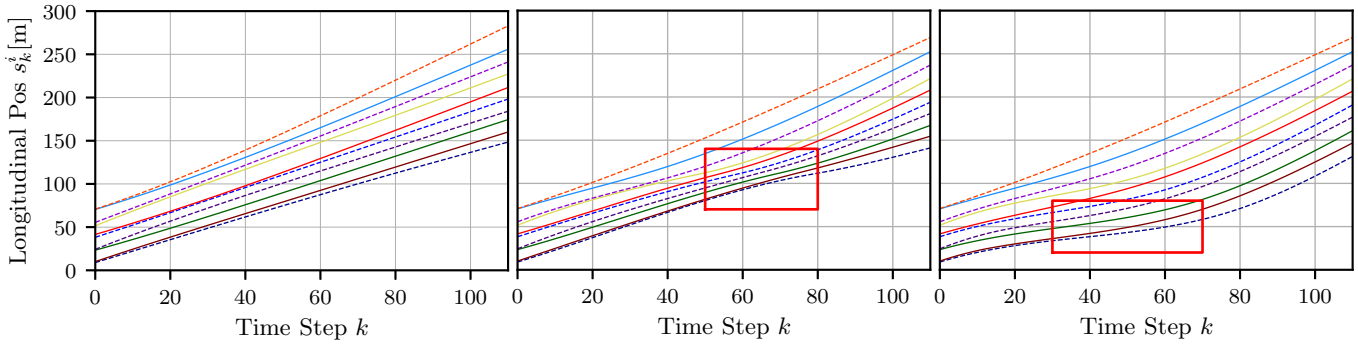
A. Simulation of Ramp Merging

In the ramp merging scenario, the initial state of CAVs is shown in Fig. 7. CAVs are represented by rectangles and differentiated by different colors. Lines with different color in the Fig. 7(b) indicate the nominal trajectory of the CAVs. Considering computational efficiency, our algorithm can accommodate at most 10 CAVs. Therefore, in this section we conduct simulations with 10 CAVs. There are 5 CAVs randomly generated within the initial range of 100 meters on both main and ramp road. The initial distance between CAVs



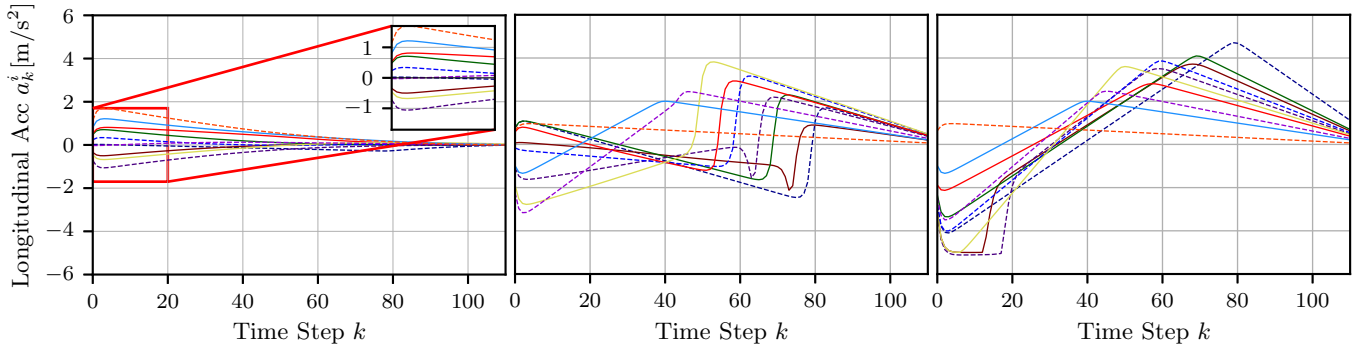
(a) Color of CAVs (b) Initial Position of CAVs

Fig. 7. The color (a) and initial position (b) of CAVs in ramp merging simulation.



(a) s_k^i generated by Alg. 2 (b) s_k^i generated by algorithm SeqA (c) s_k^i generated by algorithm SeqB

Fig. 8. Longitudinal position s_k^i from trajectories of different CAVs.



(a) a_k^i generated by Alg. 2 (b) a_k^i generated by algorithm SeqA (c) a_k^i generated by algorithm SeqB

Fig. 9. Longitudinal acceleration a_k^i from trajectories of different CAVs.

is uniformly distributed in the range of 10 meters to 20 meters, and the initial velocity is uniformly distributed in the range of 65 km/h to 75 km/h.

We compare Alg. 2 with 2 distributed baseline to demonstrate how the proposed algorithm achieves inter-vehicle cooperation by optimizing a global problem. The first baseline, denoted as SeqA, determines the merging order based on the CAVs longitudinal positions. Then, following this order, each CAV computes its reference trajectory sequentially. Specifically, a following CAV formulates its safety constraints using the preceding CAV's reference trajectory, and each CAV's trajectory is obtained by minimizing the control effort subject to safety and other operational constraints. The second baseline, SeqB, is similar to SeqA, except that the constraint about merging point, i.e., Eq. (5), is removed. This relaxation provides each CAV with greater flexibility in the control.

Fig. 8 and Fig. 9 illustrate the longitudinal positions and longitudinal accelerations of the reference trajectories generated by the three algorithms. The line colors correspond to the CAV colors in Fig. 7(a). From Fig. 8, we observe that in SeqA and SeqB, the trajectories of CAVs starting farther back become increasingly irregular. In contrast, all trajectories produced by our algorithm remain smooth. Although all three methods are distributed, only our algorithm performs distributed global optimization, meaning that all CAVs participate collectively in the trajectory planning process. In the two baselines, each CAV only considers the trajectory of the CAV ahead when computing its own, causing the control burden to propagate backward. As a result, CAVs starting farther behind accumulate more pressure and yield less smooth trajectories.

This phenomenon is even more evident in Fig. 9. By using our algorithm, except for V7, the CAVs starting far-

ther ahead—V5, V6, V8, V9, and V10—exhibit non-negative accelerations, whereas those starting farther behind show non-positive accelerations. This coordinated pattern arises because CAVs exchange the variable $y^{i,k}$ through V2X, allowing the control pressure to be shared among all CAVs and enabling true multi-vehicle cooperation. V7 is an exception: due to its randomly assigned higher initial speed, it requires some deceleration. By contrast, in the two baseline distributed schemes, the farther back a CAV is, the larger the magnitude of its acceleration tends to be—indicating that it must absorb a disproportionately large control burden.

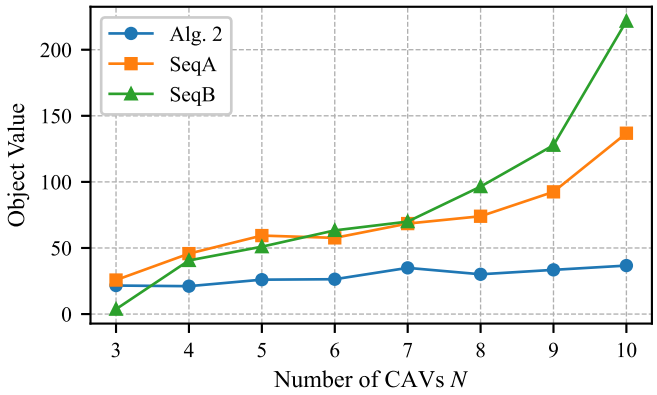


Fig. 10. Optimization objectives values with different number of CAVs.

Fig. 10 compares the optimization objectives of our algorithm with those of the two baseline methods under different numbers of CAVs. The objective is defined as the total control effort of all CAVs, $\sum_{i=1}^{N}\sum_{k=0}^{T_P^1-1}(U_k^i)^2$, where a smaller value indicates lower energy consumption and improved ride comfort. As shown in the figure, when the number of CAVs is small, our method performs similarly to the baselines. However, as the traffic volume increases, the objective value of our algorithm grows only marginally and remains significantly lower than those of both baselines, demonstrating its superior scalability and efficiency.

TABLE II
THE VALUE OF ρ AND σ.

Iteration	3	4-24	25-50
ρ	10	20	100
σ	10	20	100

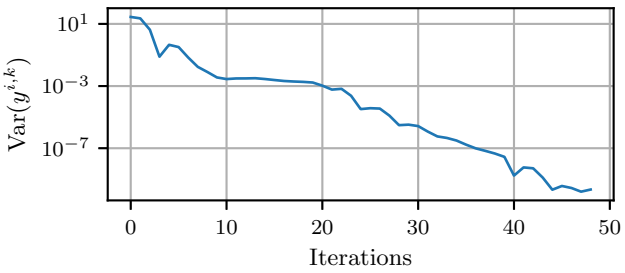


Fig. 11. Variance of $y^{i,k}$ with respect to the number of iterations.

Fig. 11 illustrates the change in the variance of the iteration $y^{i,k}$ with the iteration count, where the y-axis is in logarithmic scale. The formula for calculating the variance of $y^{i,k}$ is given by

$$Var(y^{i,k}) = \sum_{i=\mathcal{N}} \left\| y^{i,k} - \bar{y}^{i,k} \right\|^2, \tag{56}$$

where $\bar{y}^{i,k} = \sum_{i=\mathcal{N}} y^{i,k} / |\mathcal{N}|$ represents the mean vector of $y^{i,k}$. At iteration 40, the variance of $y^{i,k}$ converges to around 10^{-7}, and after iteration 50, the variance remains almost unchanged. This indicates that as the iterations progress, $y^{i,k}$ among CAVs become essentially equal, forming a "consensus". And Therefore, we choose the iteration count to be 40.

Fig. 12 shows the CAVs' moving simulation controlled by Alg. 3 using the reference trajectories generated by Alg. 2. The nominal trajectory of each CAV is plotted in the same color as the CAV itself. As illustrated, the two platoons successfully merge near the acceleration lane. In addition, when CAVs are close to each other, they deviate slightly from the lane center to satisfy the safety constraints enforced by Alg. 3. For instance, in Fig. 12(c), both V5 and V6 exhibit such deviations.

To more clearly observe this phenomenon, Fig. 13 presents the deviations of the actual trajectories from the reference trajectories in both the x and y coordinates, denoted as Δx and Δy. Since V5 and V6 show the most noticeable deviations in Fig. 12, their curves are highlighted in Fig. 13. The plot indicates that both CAVs reach their maximum Δx and Δy around $k = 60$.

TABLE III
COMPUTATION TIME WITH DIFFERENT NUMBERS OF CAVS N

N	T^p_{avg} (ms)	T^p_{max} (ms)	T^c_{avg} (ms)	T^c_{max} (ms)
3	1.908	2.290	4.074	5.584
4	2.085	2.328	5.054	10.653
5	2.125	2.511	6.219	11.528
6	2.143	2.560	7.101	13.948
7	2.258	2.930	7.937	13.974
8	2.546	3.198	8.651	14.332
9	2.813	3.326	9.444	15.941
10	3.063	3.794	10.211	18.394

Tab. III illustrates the computation time of Alg. 2 and Alg. 3 without considering the impact of communication delay. T^p_{avg} denotes the average computation time per CAV during each iteration of Alg.2 (Lines 4-10). T^p_{max} refers to maximum computation time for each CAV over the entire execution of Alg. 2. It can be observed from the table that both T^c_{avg} and T^c_{max} increase with the number of CAV. It is because that the dimension of $p^{i,k+1}$, $r^{i,k+1}$, $s^{i,k+1}$, and $z^{i,k+1}$ increases as the number of CAVs grows, leading to an increase in computation time.

Before introduce T^c_{avg} and T^c_{max} about Alg. 3 in Tab. III, we need to explain the multi-level nested loops in Alg. 3 in more detail. Specifically, as shown in Fig. 14, the gray area represents the entire simulation process, while the yellow area indicates the time consumed by lines 3-15 of Alg. 3 at a given control step k. The inner brown area corresponds

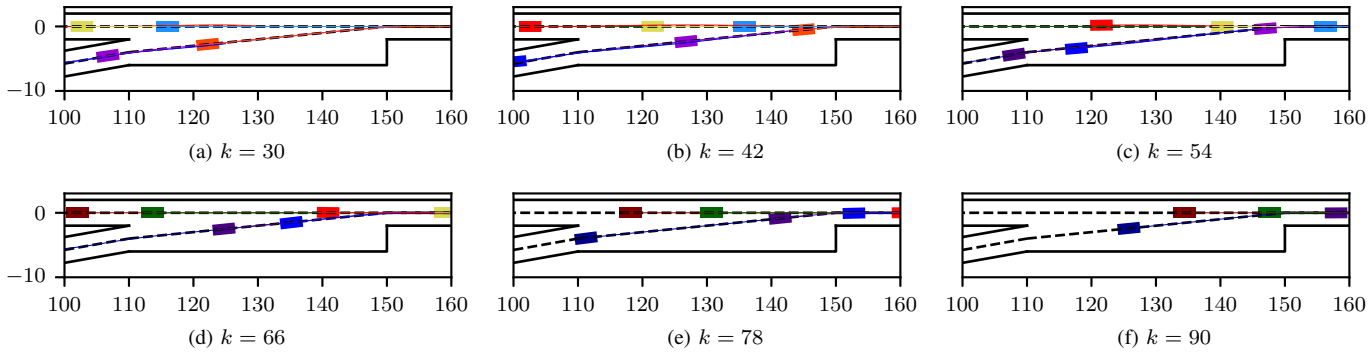


Fig. 12. The positions of CAVs at different times in the ramp merging area.

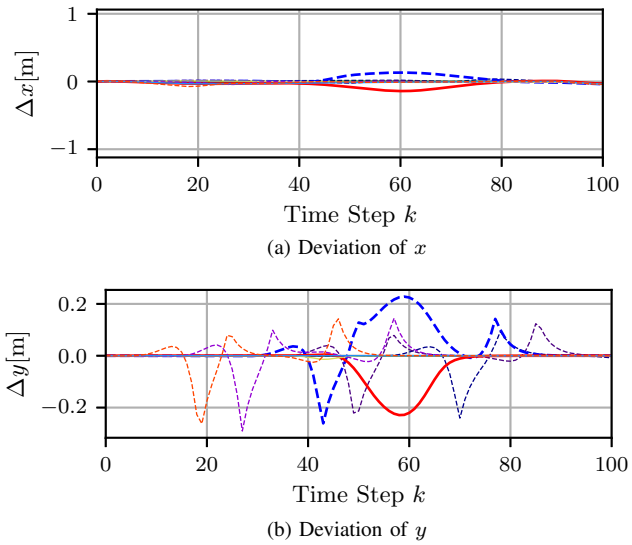


Fig. 13. Deviation between reference trajectories and real trajectories.

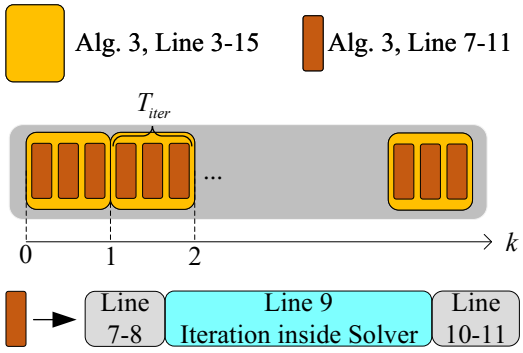


Fig. 14. Iteration of Alg. 3

to the T_{iter} iterations performed at step k (Lines 7-11, excluding communication delays). Furthermore, within Line 9 of Alg. 3, additional loops are executed inside the solvers (OSQP). Consequently, T^c_{avg} and T^c_{max} denote the average and maximum values of the time represented by the brown areas in Fig. 14, i.e., lines 7-11 of Alg. 3. They increase with the number of CAVs growing. This is because, as the number of CAVs increases, although the dimension of each CAVs' local QP, i.e., $\mathbf{P}^c_4$, remains unchanged, the additional computations required by each CAV at time step k—such as calculating $\mathcal{T}^{i,j}_k$—increase. Consequently, the computation time also grows.

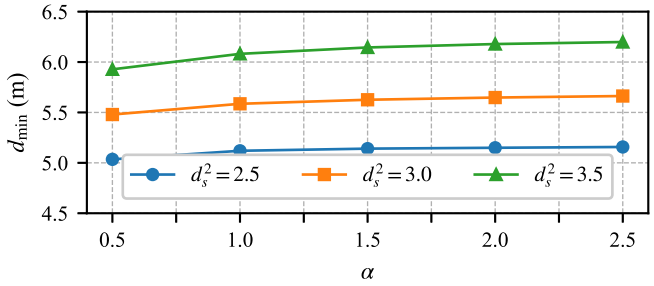


Fig. 15. Minimum distance with different α and d_s^2.

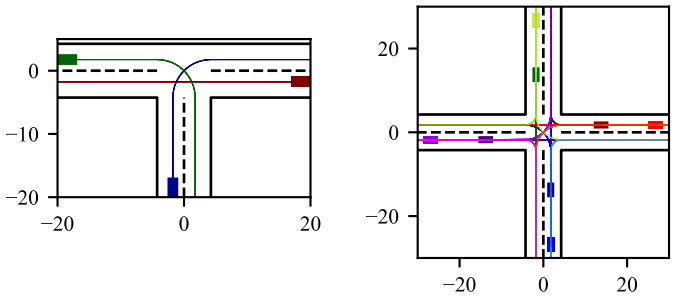
Since the strict safety constraint Eq. (13c) is relaxed into a soft constraint in $\mathbf{P}^c_4$, the safety coefficient α becomes a crucial parameter in Alg. 3. Together with d_s^2, it determines the minimum inter-vehicle distance d_{min}. To evaluate the influence of it, Fig. 15 illustrates the variation of d_{min} with α under different values of d_s^2. It can be observed that α has a smaller impact on d_{min} compared to d_s^2. This is because the reference trajectories generated by Alg. 2 already ensure relatively safe longitudinal spacing between CAVs. In other words, the 4th term in Eq. (52) is nearly 0, resulting in a small derivative with respect to α and thus a limited effect on d_{min}. To further investigate the impact of α on d_{min}, additional experiments will be conducted in Sec. V-B.

B. Simulation of DCIMPC

This section compares 3 additional algorithms, whose names correspond to the labels in the figure below:

- IPOPT: Interior Point OPTimizer (IPOPT) is a commonly used software library for large-scale nonlinear optimization of continuous systems. It can directly solves optimization problems with nonlinear constraints.
- LD-IPOPT: One of the main contributions of our work is the handling of nonlinear, non-convex safety constraints in Eq. (13c). To specifically compare this, the CAV dynamics constraints are linearized and inputted together with the original nonlinear safety constraints into IPOPT for solution, termed as LD-IPOPT.

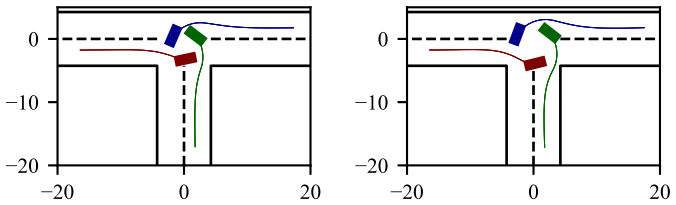
- OSQP-CS: As mentioned in Alg. 3 in Sec. IV-B, the nominal trajectory serves as the starting point for OSQP warm start. Thus, OSQP cold start is used as another comparative algorithm to evaluate the impact of this operation.



(a) T-junction (b) Intersection

Fig. 16. 2 scenarios for DCIMPC simulation.

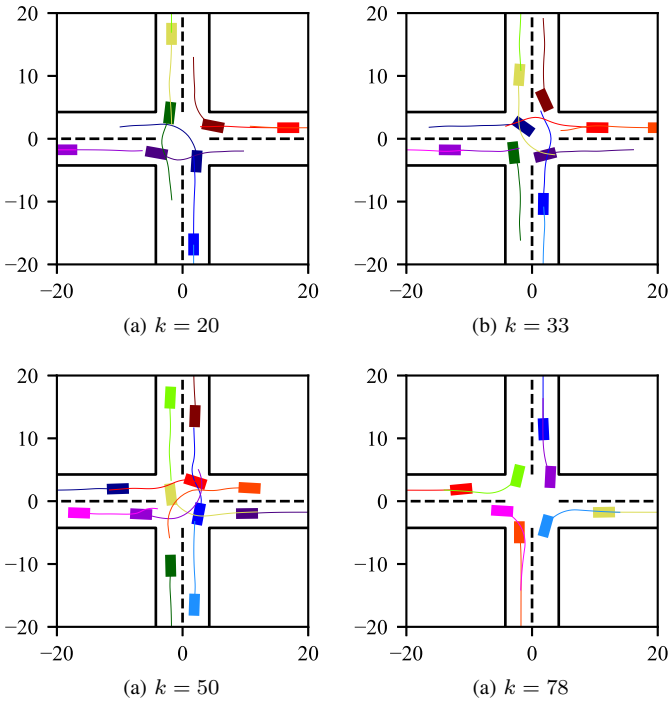
In this subsection, we consider 2 scenarios as shown in Fig. 16. The first scenario illustrates a T-junction, i.e., Fig. 16(a), where 3 CAVs from three roads intersect at the junction simultaneously. The second scenario is the intersection scene shown in Fig. 16(b). 3 CAVs are initialized in each lane, making a total of twelve, and they follow their respective reference trajectories to other lanes. The reference trajectories in this scenario are designed such that, by the end of the simulation, each lane still contains exactly 3 CAVs. In both scenarios, the reference trajectories are hand-crafted. We first specify each CAV's start and end positions, and then sample waypoints from the start to the end along the lane centerline at a fixed speed. Since this procedure produces intersecting reference trajectories at the intersection, it provides a more challenging setting to further evaluate the performance of the DCIMPC algorithm. CAVs are distinguished by rectangles of different colors. Each CAV's reference trajectory is indicated by a line in the same color as itself.



(a) $\alpha = 1.0$, $d_s^2 = 2.5m$ (a) $\alpha = 2.5$, $d_s^2 = 3.5m$

Fig. 17. Snapshots of T-junction scenario with different α and d_s^2.

Fig. 17 presents simulation snapshots of the T-junction scenario at the moments when the inter-vehicle distance reaches minimum, under different combinations of α and d_s^2. In Fig. 17 and Fig. 18, the curve with the same color as each CAV represents its nominal trajectory. As shown in the figure, these two parameters have a noticeable influence on the minimum distance between CAVs. As their values increase, the CAVs adjust their control inputs, resulting in varying degrees of deviation from their reference trajectories.



(a) $k = 20$ (b) $k = 33$

(a) $k = 50$ (a) $k = 78$

Fig. 18. Snapshots of intersection scenario at different time step k.

Fig. 18 shows simulation snapshots of the intersection scenario at different time steps. As can be seen, the nominal trajectories deviate from the reference trajectories shown in Fig. 16(b). to ensure safe driving. Considering that this scenario is considerably more complex than the T-junction case, we conducted a more comprehensive analysis of the effects of α and d_s^2 on the minimum inter-vehicle distance, as illustrated in Fig. 19.

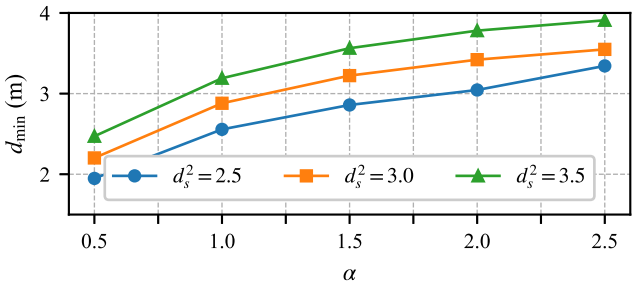


Fig. 19. Minimum distance with different α and d_s^2.

Fig. 19 illustrates the influence of the safety coefficient α on the minimum inter-vehicle distance d_{min} in the intersection scenario. Compared with Fig. 15, it can be observed that α has a more pronounced impact on d_{min} in this case. This is because the reference trajectories in the intersection scenario exhibit overlap, making the safety constraint, i.e., the 4th term in Eq. (52),contribute more dominantly to the overall objective function. As shown by the trend, d_{min} increases with both α and d_s^2, and increasing these two parameters can further enhance safety. Considering the limitation of road width, we finally set α and d_s^2 to 1.0 and 2.5m, respectively.

TABLE IV
COMPUTATION TIME OF DCIMPC AT T-JUNCTION AND INTERSECTION

	T^c_{avg} (ms)		T^c_{max} (ms)	
	T-3	I-12	T-3	I-12
DCIMPC	4.48	13.88	9.35	19.91
OSQP-CS	4.76	13.92	14.77	22.10
IPOPT	96.41	181.43	199.97	472.73
LIPOPT	108.45	153.35	211.55	382.49

Tab. IV presents the computational time consumption of Alg. 3 in both the T-junction and Intersection scenarios. T-3 and I-12 represent T-junction with 3 CAVs and intersection with 12 CAVs, respectively. As shown in the table, our algorithm achieves nearly a 20-fold improvement in efficiency compared with the IPOPT and LIOPOPT methods. This demonstrates that, although additional computations are required in Eq. (53), the overall computational cost is significantly reduced. Such improvement is particularly important for real-time applications.

C. Discussion about Practical Feasibility of Algorithms

CAVs operating in NR-V2X Mode 2 utilize sensing-based semi-persistent-scheduling (SB-SPS) for sidelink (SL) channel resource block (RB) scheduling [4]. Each RB, represented by a small rectangle in Fig. 20, occupies 1 ms in the time domain and one subchannel in the frequency domain. CAVs transmit data packets on RB with payloads at fixed time intervals, i.e., Resource Reservation Intervals (RRI). which ranges from 20 to 1000ms [38], [39]. The feasibility of Alg. 2 and Alg. 3 is discussed separately below.

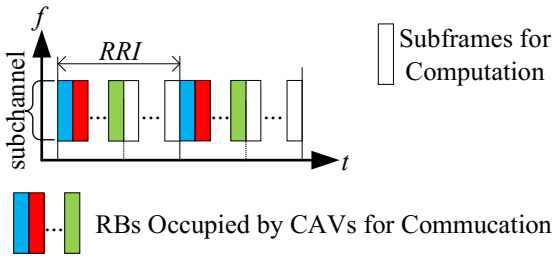


Fig. 20. RBs allocation under the NR-V2X mode 2.

In the on-ramp merging scenario, we assume a fixed RRI. Within each RRI period, the first N RBs of a single subchannel are allocated to N CAVs for one round of communication, i.e., line 2 of Alg. 2 and lines 5 to 6 of Alg. 3. (In general, multiple CAVs can transmit in different subchannels of the same subframe. However, to avoid potential issues caused by half-duplex operation, we adopt the single-subchannel assumption.) In the remaining $RRI - N$ subframes, CAVs perform local computations without occupying any RBs. After the computations are completed, they wait for the next RRI cycle to start another round of communication and computation.

Based on the above discussion, the RRI should be selected according to the number of CAVs and the computation time. Assuming $N = 10$, for Alg. 2, the simulation results in Tab. III suggest setting the RRI to 20 ms, which allows Alg. 2 to complete within approximately 0.8 s (40×20ms). For Alg. 3, since T^c_{max} is at most 18.394 ms, the RRI can be set to 30 ms. This configuration also satisfies the timing requirement of completing $T_{\text{iter}} = 3$ iterations within $T_S = 100$ ms.

Moreover, dedicated computing chips such as FPGAs and GPUs can achieve significantly higher computational efficiency compared to the general-purpose CPU used in our simulations [40], [41]. Therefore, based on the above discussion, we believe that the proposed algorithms in out work can be implemented under the existing NR-V2X communication protocol.

VI. Conclusions

This paper presented a distributed control framework designed to address the multi-vehicle control problem in ramp merging areas. The framework puts the solution to the problem into 2 steps. In the first step, the ramp merging scenario is simplified and abstracted as a multi-vehicle trajectory planning problem, designed to maintain traffic safety at a macroscopic level and enhance traffic efficiency. Then, the problem is decomposed and solved by CAVs distributedly. In the second step, a microscopic multi-vehicle control problem is formulated upon the trajectories generated from the first step. It incorporates a more realistic kinematic model and tighter safety constraints. Then, DCIMPC is proposed to efficiently solve this problem in parallel. Simulation validates the effectiveness of the proposed two step approach and the convergence of the distributed algorithm. Additionally, a comparison of the solution time under different numbers of CAVs demonstrates the feasibility of practical deployment.

References

[1] M. Sarvi and M. Kuwahara, "Microsimulation of freeway ramp merging processes under congested traffic conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 470–479, 2007.

[2] D. Gruyer, V. Magnier, K. Hamdi, L. Claussmann, O. Orfila, and A. Rakotonirainy, "Perception, information processing and modeling: Critical stages for autonomous driving applications," *Annual Reviews in Control*, vol. 44, pp. 323–341, 2017.

[3] Y. Xie, Q. Wu, P. Fan, N. Cheng, W. Chen, J. Wang, and K. B. Letaief, "Resource allocation for twin maintenance and task processing in vehicular edge computing network," *IEEE Internet of Things Journal*, vol. 12, no. 15, pp. 32 008–32 021, 2025.

[4] M. H. C. García, A. Molina-Galan, M. Boban, J. Gozálvez, B. Coll-Perales, T. Şahin, and A. Kousaridas, "A tutorial on 5g nr v2x communications," *IEEE Communications Surveys & Tutorials*, vol. 23, pp. 1972–2026, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231855545

[5] R. Sun, N. Cheng, C. Li, F. Chen, and W. Chen, "Knowledge-driven deep learning paradigms for wireless network optimization in 6g," *IEEE Network*, vol. 38, no. 2, pp. 70–78, 2024.

[6] X. Xu, Q. Wu, P. Fan, K. Wang, N. Cheng, W. Chen, and K. B. Letaief, " Enhanced Velocity-Adaptive Scheme: Joint Fair Access and Age of Information Optimization in Vehicular Networks ," *IEEE Transactions on Mobile Computing*, no. 01, pp. 1–18, Oct. 5555. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/TMC.2025.3617145

[7] X. Gu, Q. Wu, P. Fan, Q. Fan, N. Cheng, W. Chen, and K. B. Letaief, "Drl-based resource allocation for motion blur resistant federated self-supervised learning in iov," *IEEE Internet of Things Journal*, vol. 12, no. 6, pp. 7067–7085, 2025.

[8] J. Yang, Q. Ni, G. Luo, Q. Cheng, L. Oukhellou, and S. Han, "A trustworthy internet of vehicles: The dao to safe, secure and collaborative autonomous driving," *IEEE Transactions on Intelligent Vehicles*, 2023.

[9] J. Shen, N. Cheng, X. Wang, F. Lyu, W. Xu, Z. Liu, K. Aldubaikhy, and X. Shen, "Ringsfl: An adaptive split federated learning towards taming client heterogeneity," *IEEE Transactions on Mobile Computing*, vol. 23, no. 5, pp. 5462–5478, 2024.

[10] Z. Zhang, Q. Wu, P. Fan, N. Cheng, W. Chen, and K. B. Letaief, "Drl-based optimization for aoi and energy consumption in c-v2x enabled iov," *IEEE Transactions on Green Communications and Networking*, pp. 1–1, 2025.

[11] Z. Ali, S. Lagén, L. Giupponi, and R. Rouil, "3gpp nr v2x mode 2: Overview, models and system-level evaluation," *IEEE Access*, vol. 9, pp. 89 554–89 579, 2021.

[12] F. Ding, Y. Xiao, L. Xu, and Z. Fang, "Hierarchical stochastic gradient and hierarchical multi-innovation stochastic gradient identification for multivariable arx models," *International Journal of Adaptive Control and Signal Processing*, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:281695733

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

[14] T. Zeng, O. Semiari, W. Saad, and M. Bennis, "Joint communication and control system design for connected and autonomous vehicle navigation," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[15] S. Jing, F. Hui, X. Zhao, J. Rios-Torres, and A. J. Khattak, "Cooperative game approach to optimal merging sequence and on-ramp merging control of connected and automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4234–4244, 2019.

[16] L. Yang, J. Zhan, W.-L. Shang, S. Fang, G. Wu, X. Zhao, and M. Deveci, "Multi-lane coordinated control strategy of connected and automated vehicles for on-ramp merging area based on cooperative game," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[17] J. Rios-Torres and A. A. Malikopoulos, "Automated and cooperative vehicle merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 780–789, 2016.

[18] Z. el abidine Kherroubi, S. Aknine, and R. Bacha, "Novel decision-making strategy for connected and autonomous vehicles in highway on-ramp merging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12 490–12 502, 2021.

[19] V. Milanés, J. Godoy, J. Villagrá, and J. Pérez, "Automated on-ramp merging system for congested traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 500–508, 2010.

[20] E. Beheshtitabar and E. Mohammad Alipour, "A rule based control algorithm for on-ramp merge with connected and automated vehicles," in *International Conference on Transportation and Development 2020*. American Society of Civil Engineers Reston, VA, 2020, pp. 303–316.

[21] Y. Zhou, M. E. Cholette, A. Bhaskar, and E. Chung, "Optimal vehicle trajectory planning with control constraints and recursive implementation for automated on-ramp merging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3409–3420, 2018.

[22] H. Liu, W. Zhuang, G. Yin, Z. Li, and D. Cao, "Safety-critical and flexible cooperative on-ramp merging control of connected and automated vehicles in mixed traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 2920–2934, 2023.

[23] S. Hwang, K. Lee, H. Jeon, and D. Kum, "Autonomous vehicle cut-in algorithm for lane-merging scenarios via policy-based reinforcement learning nested within finite-state machine," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 594–17 606, 2022.

[24] Y. Xue, C. Ding, B. Yu, and W. Wang, "A platoon-based hierarchical merging control for on-ramp vehicles under connected environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 821–21 832, 2022.

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

[26] A. Katriniok, P. Kleibaum, and M. Joševski, "Distributed model predictive control for intersection automation using a parallelized optimization approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5940–5946, 2017.

[27] F. Ding, L. Xu, P. Liu, and X. Wang, "Two-stage parameter estimation methods for linear time-invariant continuous-time systems," *Systems & Control Letters*, vol. 204, p. 106166, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167691125001483

[28] S. Bae, D. Saxena, A. Nakhaei, C. Choi, K. Fujimura, and S. Moura, "Cooperation-aware lane change maneuver in dense traffic based on model predictive control with recurrent neural network," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 1209–1216.

[29] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.

[30] F. Ding, L. Xu, X. Zhang, H. Xu, Y. Zhou, and X. Luan, "Hierarchical generalized extended parameter identification for multivariable equation-error arma-like systems by using the filtering identification idea," *Annual Reviews in Control*, vol. 60, p. 100993, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578825000082

[31] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[32] G. Banjac, F. Rey, P. Goulart, and J. Lygeros, "Decentralized resource allocation via dual consensus admm," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 2789–2794.

[33] G. Banjac, F. Rey, P. Goulart, and J. Lygeros, "Decentralized resource allocation via dual consensus admm," in *2019 American Control Conference (ACC)*, 2019, pp. 2789–2794.

[34] Z. Huang, S. Shen, and J. Ma, "Decentralized ilqr for cooperative trajectory planning of connected autonomous vehicles via dual consensus admm," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[35] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "Osqp: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[36] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.

[37] L. T. Biegler and V. M. Zavala, "Large-scale nonlinear programming using ipopt: An integrating framework for enterprise-wide dynamic optimization," *Computers & Chemical Engineering*, vol. 33, no. 3, pp. 575–582, 2009.

[38] 3rd Generation Partnership Project (3GPP), "TS 36.213: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," 3GPP, Technical Specification v15.7.0, Release 15, Sep. 2019, available: https://www.3gpp.org/DynaReport/36213.htm.

[39] 3rd Generation Partnership Project (3GPP), "TS 36.214: Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements," 3GPP, Technical Specification v16.0.0, Release 16, Dec. 2019, available: https://www.3gpp.org/DynaReport/36214.htm.

[40] Y. Zhang, X. Niu, H. Tian, Y. Zhang, B. Yu, S. Liu, and S. Huang, "A sparsity-aware autonomous path planning accelerator with hw/sw co-design and multi-level dataflow optimization," *ACM Trans. Archit. Code Optim.*, vol. 22, no. 3, Sep. 2025. [Online]. Available: https://doi.org/10.1145/3750449

[41] F. Ding, X. Luan, L. Xu, and X. Zhang, "Hierarchical recursive gradient parameter identification for multi-input arx systems with partially-coupled information vectors," *Int. J. Adapt. Control Signal Process.*, vol. 39, no. 9, p. 1978–1995, Sep. 2025. [Online]. Available: https://doi.org/10.1002/acs.4036

Jiahou Chu received the M.S.degree and B.S.degree at School of Internet of Things Engineering, Jiangnan University, Wuxi, China. His current research interests include autonomous driving, distributed optimization, vehicular network.

Qiong Wu (Senior Member, IEEE) received the Ph.D. degree in information and communication engineering from National Mobile Communications Research Laboratory, Southeast University, Nanjing, China, in 2016. From 2018 to 2020, he was a postdoctoral researcher with the Department of Electronic Engineering, Tsinghua University, Beijing, China. He is currently an associate professor with the School of Internet of Things Engineering, Jiangnan University, Wuxi, China. His current research interest focuses on vehicular networks, autonomous driving communication technology, and machine learning.

Dr. Pingyi Fan is a professor and the director of open source data recognition innovation center, Department of Electronic Engineering, Tsinghua University. He is member (Academician) of the united states national academy of artificial intelligence (NAAI), Fellow of IET and IET Fellowship international Assessor. He received Ph.D. degree at the Department of Electronic Engineering of Tsinghua University in 1994. He has obtained many research grants, including national 973 Project, 863 Project, mobile special project and the key R&D program, national natural funds and international cooperation projects. He has published more than 600 papers (ORCID) including 171 IEEE journals and more than 10 ESI highly cited papers as well as 4 academic books. He won 2025 NAAI AI Exploration Award, and 10 best paper awards of IEEE international conferences, including IEEE ICCCS2023 and 2024, ICC2020 and Globecom 2014, and received the best paper award of IEEE TAOS Technical Committee in 2020, the excellent editor award of IEEE TWC (2009), the most popular scholar award 2023 of AEIC, the second natural Prize of CIC (2023) and several international innovation exhibition medals, i.e. Gold Medal at the Russian Invention Exhibition-2024, Silver Medal at Geneva Invention Exhibition-2023, and Silver Medal at Paris Invention Exhibition-2023 etc. and served as the editorial board member of several Journals, including IEEE and MDPI. His current research interests are in 6G wireless communication network and machine learning, semantic information theory and generalized information theory, big data processing theory, intelligent network and system detection, etc.

Wen Chen (M'03–SM'11) received BS and MS from Wuhan University, China in 1990 and 1993 respectively, and PhD from University of Electro-communications, Japan in 1999. He is now a tenured Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, China. He is a Clarivate highly cited researcher, and a fellow of Chinese Institute of Electronics. He is the Shanghai Chapter Chair of IEEE Vehicular Technology Society, and a vice president of Shanghai Institute of Electronics. He was the distinguished lecturers of IEEE Communications Society and IEEE Vehicular Technology Society, the editors of IEEE Transactions on Wireless Communications and IEEE Transactions on Communications.

Kezhi Wang received the Ph.D. degree from the University of Warwick, U.K. He is a Professor with the Department of Computer Science, Brunel University of London, U.K. His research interests include wireless communications, mobile edge computing, and machine learning. He is a Clarivate Highly Cited Researcher in 2023-2024.

Nan Cheng (Senior Member, IEEE), received the B.E. and M.S. degrees from the College of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2016. He was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, from 2017 to 2019. He is currently a Professor with the State Key Laboratory of ISN and with the School of Telecommunications Engineering, Xidian University, Xi'an, Shaanxi, China. He has published over 90 journal papers in IEEE Transactions and other top journals. His current research focuses on B5G/6G, AI-driven future networks, and space–air-ground integrated networks. Prof. Cheng serves as an Associate Editor for IEEE Transactions on Vehicle Technology, IEEE Open Journal of the Communication Society, and Peer-to-Peer Networking and Applications, and serves/served as a guest editor for several journals.

Khaled Ben Letaief (Fellow, IEEE) received the B.S. (Hons.), M.S., and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in December 1984, August 1986, and May 1990, respectively. From 1990 to 1993, he was a Faculty Member with The University of Melbourne, Melbourne, VIC, Australia. Since 1993, he has been with The Hong Kong University of Science and Technology (HKUST), Hong Kong, where he is currently the New Bright Professor of Engineering. At HKUST, he has held many administrative positions, including an Acting Provost, the Dean of Engineering, the Head of the Electronic and Computer Engineering Department, and the Director of the Hong Kong Telecom Institute of Information Technology. He is an internationally recognized leader in wireless communications and networks. His research interests include artificial intelligence, mobile cloud and edge computing, tactile Internet, and sixth-genereation (6G) systems. In these areas, he has over 720 articles with over 44,450 citations and an H-index of over 100 along with 15 patents, including 11 U.S. inventions. Dr. Letaief served as a member for the IEEE Board of Directors from 2022 to 2023. He is a member of the National Academy of Engineering, USA, and the Hong Kong Academy of Engineering Sciences; and a Fellow of the Hong Kong Institution of Engineers. He is well recognized for his dedicated service to professional societies and IEEE, where he served in many leadership positions, including the President of the IEEE Communications Society from 2018 to 2019, the world's leading organization for communications professionals with headquarter in New York City, and members in 162 countries. He is recognized by Thomson Reuters as an ISI Highly Cited Researcher and was listed among the 2020 top 30 of AI 2000 Internet of Things Most Influential Scholars. He was a recipient of many distinguished awards and honors, including the 2007 IEEE Communications Society Joseph LoCicero Publications Exemplary Award, the 2009 IEEE Marconi Prize Award in Wireless Communications, the 2010 Purdue University Outstanding Electrical and Computer Engineer Award, the 2011 IEEE Communications Society Harold Sobol Award, the 2016 IEEE Marconi Prize Paper Award in Wireless Communications, the 2016 IEEE Signal Processing Society Young Author Best Paper Award, the 2018 IEEE Signal Processing Society Young Author Best Paper Award, the 2019 IEEE Communication Society and Information Theory Society Joint Paper Award, the 2021 IEEE Communications Society Best Survey Paper Award, and the 2022 IEEE Communications Society Edwin Howard Armstrong Achievement Award. He is the Founding Editor-in-Chief of the prestigious IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS. He has been involved in organizing many flagship international conferences.