Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Fully Distributed Optimization-Based CAV Platooning Control Under Linear Vehicle Dynamics

Jinglai Shen, Eswar Kumar H. Kammara, Lili Du

To cite this article:
Jinglai Shen, Eswar Kumar H. Kammara, Lili Du (2022) Fully Distributed Optimization-Based CAV Platooning Control Under Linear Vehicle Dynamics. Transportation Science 56(2):381-403. https://doi.org/10.1287/trsc.2021.1100

http://pubsonline.informs.org/journal/trsc

TRANSPORTATION SCIENCE
Vol. 56, No. 2, March–April 2022, pp. 381–403
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Fully Distributed Optimization-Based CAV Platooning Control Under Linear Vehicle Dynamics

Jinglai Shen,[a] Eswar Kumar H. Kammara,[a] Lili Du[b]

[a] Department of Mathematics and Statistics, University of Maryland, Baltimore, Maryland 21250; [b] Department of Civil and Coastal Engineering, University of Florida, Gainesville, Florida 32608

Contact: shenj@umbc.edu, https://orcid.org/0000-0003-2172-4182 (JS); eswar1@umbc.edu (EKHK); lilidu@ufl.edu, https://orcid.org/0000-0003-1740-1209 (LD)

Received: November 23, 2020
Revised: May 26, 2021
Accepted: September 14, 2021
Published Online in Articles in Advance: January 21, 2022

https://doi.org/10.1287/trsc.2021.1100

Abstract. This paper develops distributed optimization-based, platoon-centered connected and autonomous vehicle (CAV) car-following schemes, motivated by the recent interest in CAV platooning technologies. Various distributed optimization or control schemes have been developed for CAV platooning. However, most existing distributed schemes for platoon centered CAV control require either centralized data processing or centralized computation in at least one step of their schemes, referred to as partially distributed schemes. In this paper, we develop fully distributed optimization based, platoon centered CAV platooning control under the linear vehicle dynamics via the model predictive control approach with a general prediction horizon. These fully distributed schemes do not require centralized data processing or centralized computation through the entire schemes. To develop these schemes, we propose a new formulation of an objective function and a decomposition method that decomposes a densely coupled central objective function into the sum of multiple locally coupled functions whose coupling satisfies the network topology constraint. We then exploit locally coupled optimization and operator splitting methods to develop fully distributed schemes. Control design and stability analysis is carried out to achieve desired traffic transient performance and asymptotic stability. Numerical tests demonstrate the effectiveness of the proposed fully distributed schemes and CAV platooning control.

Funding: This work was supported by the NSF Division of Civil, Mechanical and Manufacturing Innovation [Grants 1818526, 1901994, and 1902006].

Keywords: connected and autonomous vehicle • car-following control • distributed algorithm • constrained optimization • control and stability

1. Introduction

The recent advancement of connected and autonomous vehicle (CAV) technologies provides a unique opportunity to mitigate urban traffic congestion through innovative traffic flow control and operations. Supported by advanced sensing, vehicle communication, and portable computing technologies, CAVs can sense, share, and process real-time mobility data and conduct cooperative or coordinated driving. This has led to a surging interest in self-driving technologies. Among a number of emerging self-driving technologies, vehicle platooning technology receives substantial attention. Specifically, the vehicle platooning technology links a group of CAVs through cooperative acceleration or speed control. It allows adjacent group members to travel safely at a higher speed with smaller spacing between them and thus has a great potential to increase lane capacity, improve traffic flow efficiency, and reduce congestion, emission, and fuel consumption (Kavathekar and Chen 2011, Bergenhem et al. 2012).

There is extensive literature on CAV platooning control. The widely studied approaches include adaptive cruise control (ACC) (Marsden, McDonald, and Brackstone 2001, Vander Werf et al. 2002, Kesting et al. 2008, Li et al. 2011, Zhou et al. 2017), cooperative adaptive cruise control (CACC) (Van Arem, Van Driel, and Visser 2006; Shladover et al. 2012, 2015; Zhao and Zhang 2020), and platoon-centered vehicle platooning control (Wang et al. 2014, 2019; Gong, Shen, and Du 2016; Gong and Du 2018). The first two approaches intend to improve an individual vehicle's safety, mobility, and string stability rather than systematical performance of the entire platoon, although enhanced system performance is validated by simulations or field experiments. In contrast, the platoon centered approach aims to improve the performance of the entire platoon and seeks a control input that optimizes the platoon's transient traffic dynamics for a smooth traffic flow while achieving stability and other desired long-time dynamical behaviors.

The platoon centered CAV platooning control often gives rise to sophisticated, large-scale optimal control or optimization problems, and requires extensive computation. To successfully implement these control

schemes, efficient real-time computation is needed (Wang et al. 2019). However, because of high computation load and the absence of roadside computing facilities, centralized computation is either inefficient or infeasible (Wang et al. 2016). By leveraging portable computing capability of each vehicle, distributed computing is a favorable option because it has potentials to be more adaptive to different platoon network topologies, be more robust to network malfunctions, and accommodate for communication delays effectively (Mesbahi and Egerstedt 2010, Wang et al. 2016). Despite these advantages, the development of efficient distributed algorithms to solve platoon centered optimization or optimal control problems in real time is nontrivial, especially under complicated traffic conditions and constraints. It is worth pointing out that a platoon centered car following control is a centralized control approach even though its computation is distributed; that is, each vehicle computes its own control input from the central control scheme in a distributed manner. Hence, the platoon centered approach is different from decentralized control widely studied in control engineering (Barooah, Mehta, and Hespanha 2009, Zheng et al. 2017, Zhou, Wang, and Ahn 2019). Besides, the platoon-centered approach focuses on closed loop stability of the entire platoon instead of stability of individual vehicles and their interactions, for example, string stability (Zhou, Wang, and Ahn 2019).

A number of effective distributed control or optimization schemes have been proposed for CAV platooning (Wang et al. 2016, 2019; Zhou, Wang, and Ahn 2019; Zhao and Zhang 2020). A recent paper (Gong, Shen, and Du 2016) develops model predictive control (MPC)-based car-following control schemes for CAV platooning by exploiting transportation, control, and optimization methodologies. These control schemes take vehicle constraints, transient dynamics, and asymptotic dynamics of the entire platoon into account and can be computed in a distributed manner. Gong and Du (2018) extend these distributed schemes to a mixed traffic flow including both CAVs and human-driven vehicles. However, to the best of our knowledge, the proposed schemes in Gong and Du (2018) and Gong, Shen, and Du (2016), as well as many other existing distributed or decentralized schemes (Koshal, Nedic, and Shanbhag 2011), either require all vehicles to exchange information with a central component for centralized data processing or perform centralized computation in at least one step of these schemes. We refer to these schemes as *partially distributed* schemes. In contrast, the distributed schemes developed in this paper do not require centralized data processing or carry out centralized computation through the entire schemes and thus are called *fully distributed*. Distinct advantages of fully distributed schemes include but are not limited to the following: (i) no data synchronization is needed such that no central computing equipment is required; and (ii) each vehicle only interacts with its nearby vehicles through a vehicle communication network. Hence, these schemes impose less restriction on vehicle communication networks and can be easily implemented on a wide range of vehicle networks. They are also suitable for a large CAV platoon in remote areas where communication network is unreliable or roadside equipment is scarce. Furthermore, they are more robust to network malfunction or cyberattacks.

In this paper, we develop a fully distributed optimization-based and platoon-centered CAV car-following control scheme over a general vehicle communication network. We propose a general p-horizon MPC model subject to the linear vehicle dynamics and various physical or safety constraints. Typically, a fully distributed optimization scheme requires the objective function and constraints of the underlying optimization problem to be decoupled (Hu, Xiao, and Liu 2018). However, the proposed MPC is a centralized control approach, and its underlying optimization problem does not satisfy this requirement because its objective function is densely coupled, and its constraints are locally coupled (see Remark 3.2 for details). Therefore, this paper develops new techniques to overcome this difficulty.

The main contributions of this paper are summarized as follows:

1. We propose a new form of the objective function in the MPC model with new sets of weight matrices. This new formulation facilitates the development of fully distributed schemes and closed loop stability analysis whereas it can achieve desired traffic transient performance of the whole platoon. Based on the new formulation, a decomposition method is developed for the strongly convex quadratic objective function. This method decomposes the central objective function into the sum of locally coupled (strongly) convex quadratic functions, where local coupling satisfies the network topology constraint under a mild assumption on network topology. Along with locally coupled constraints in the MPC model, the underlying optimization model is formulated as a locally coupled convex quadratically constrained quadratic program (QCQP).

2. Fully distributed schemes are developed for solving the previously mentioned convex QCQP arising from the MPC model using the techniques of locally coupled optimization and operator splitting methods. Specifically, by introducing copies of local coupling variables of each vehicle, an augmented optimization model is formulated with an additional consensus constraint. A generalized Douglas-Rachford splitting method based distributed scheme is developed, where

only local information exchange is needed, leading to a fully distributed scheme. Other operator splitting method based distributed scheme are also discussed.

3. The new formulation of the weight matrices and objective function leads to different closed loop dynamics in comparison with that in Gong, Shen, and Du (2016). Besides, because a general p-horizon MPC is considered, it calls for new stability analysis of the closed loop dynamics. We perform detailed stability analysis and choose suitable weight matrices for desired traffic transient performance for a general horizon length p. In particular, we prove that up to a horizon of $p = 3$, the closed loop dynamic matrix is Schur stable. Extensive numerical tests are carried out to test the proposed distributed schemes under different MPC horizon p's and to evaluate the closed loop stability and performance.

The rest of the paper is organized as follows. Section 2 introduces the linear vehicle dynamics, state and control constraints, and vehicle communication networks. The model predictive control model with a general prediction horizon p is proposed and formulated as a constrained optimization problem in Section 3; fundamental properties of this optimization problem are established. Section 4 develops fully distributed schemes by exploiting a decomposition method for the central quadratic objective function, locally coupled optimization, and operator splitting methods. Control design and stability analysis for the closed loop dynamics is presented in Section 5 with numerical results given in Section 6. Finally, conclusions are given in Section 7.

2. Vehicle Dynamics, Constraints, and Communication Networks

We consider a platoon of multiple vehicles on a straight roadway, where the (uncontrolled) leading vehicle is labeled by the index 0 and its n following CAVs are labeled by the indices $i = 1, \dots, n$, respectively. Let x_i, v_i denote the longitudinal position and speed of the ith vehicle, respectively. Let $\tau > 0$ be the sampling time, and each time interval is given by $[k\tau, (k+1)\tau)$ for $k \in \mathbb{Z}_+ := \{0, 1, 2, \dots\}$. We consider the following kinematic model for linear vehicle dynamics widely adopted in system-level studies with the acceleration $u_i(k)$ as the control input for vehicle i:

$$\begin{aligned} x_i(k+1) &= x_i(k) + \tau v_i(k) + \frac{\tau^2}{2} u_i(k), \\ v_i(k+1) &= v_i(k) + \tau u_i(k). \end{aligned} \tag{1}$$

2.1. State and Control Constraints

Each vehicle in a platoon is subject to several important state and control constraints. For each $i = 1, \dots, n$,

(1) Control constraints: $a_{\min} \le u_i \le a_{\max}$, where $a_{\min} < 0$ and $a_{\max} > 0$ are prespecified acceleration and deceleration bounds for a vehicle.

(2) Speed constraints: $v_{\min} \le v_i \le v_{\max}$, where $0 \le v_{\min} < v_{\max}$ are prespecified bounds on longitudinal speed for a vehicle;

(3) Safety distance constraints: these constraints guarantee sufficient spacing between neighboring vehicles to avoid collision even if the leading vehicle comes to a sudden stop. This gives rise to the safety distance constraint of the following form:

$$x_{i-1} - x_i \ \ge \ L + r \cdot v_i - \frac{(v_i - v_{\min})^2}{2a_{\min}}, \tag{2}$$

where $L > 0$ is a constant depending on vehicle length, and r is the reaction time.

Besides, we assume that the leading vehicle satisfies the same acceleration and speed constraints, that is, $a_{\min} \le u_0(k) \le a_{\max}$, and $v_{\min} \le v_0(k) \le v_{\max}$ for all $k \in \mathbb{Z}_+$. Constraints (1) and (2) are decoupled across vehicles, whereas the safety distance Constraint (3) is state-control coupled because such a constraint involves control inputs of two vehicles. This yields challenges to distribution computation. Furthermore, the identical acceleration or deceleration bounds are considered in this paper, although the proposed approach can handle a general case with different acceleration or deceleration bounds.

2.2. Communication Network Topology

We consider a general communication network whose topology is modeled by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \dots, n\}$ is the set of nodes with the ith node corresponding to the ith CAV, and $\mathcal{E}$ is the set of edges connecting two nodes in $\mathcal{V}$. Let $\mathcal{N}_i$ denote the set of neighbors of node i, that is, $\mathcal{N}_i = \{j \mid (i,j) \in \mathcal{E}\}$. The following assumption on the communication network topology is made throughout the paper:

Assumption 1. *The graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is undirected and connected. Further, two neighboring CAVs form a bidirectional edge of the graph, i.e., $(1,2), (2,3), \dots, (n-1, n) \in \mathcal{E}$, and the first CAV can receive $x_0(k)$, $v_0(k)$, and $u_0(k)$ from the leading vehicle at each $k \in \mathbb{Z}_+$.*

Because the graph is undirected, for any $i, j \in \mathcal{V}$ with $i \ne j$, $(i,j) \in \mathcal{E}$ means that there exists an edge between node i and node j. In other words, vehicle i can receive information from vehicle j and send information to vehicle j, and so does vehicle j. The setting given by Assumption 1 includes many widely used communication networks of CAV platoons, for example, predecessor-following, predecessor-leader following, immediate-preceding, multiple-preceding, and preceding-and-following networks (Zheng et al. 2017).

3. MPC for CAV Platooning Control

We exploit the MPC approach for car-following control of a platoon of CAVs. Let Δ be the desired constant spacing between two adjacent vehicles, and (x_0, v_0, u_0) be the position, speed, and control input of the leading vehicle, respectively. Define the vectors: (i) $z(k) := (x_0 - x_1 - \Delta, \ldots, x_{n-1} - x_n - \Delta)(k) \in \mathbb{R}^n$, representing the relative spacing error; (ii) $z'(k) := (v_0 - v_1, \ldots, v_{n-1} - v_n)(k) \in \mathbb{R}^n$, representing the relative speed between adjacent vehicles; and (iii) $u(k) := (u_1, \ldots, u_n)(k) \in \mathbb{R}^n$, representing the control input. Furthermore, let $w_i(k) := u_{i-1}(k) - u_i(k)$ for each $i = 1, \ldots, n$, and $w(k) := (w_1, \ldots, w_n)(k) \in \mathbb{R}^n$, which stands for the difference of control input between adjacent vehicles. Hence, for any $k \in \mathbb{Z}_+$, $u(k) = -S_n w(k) + u_0(k) \cdot \mathbf{1}$, where $\mathbf{1} := (1, \ldots, 1)^T$ is the vector of ones, and

$$S_n := \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & 1 & \ldots & 1 & 0 \\ 1 & 1 & \ldots & 1 & 1 \end{bmatrix} \in \mathbb{R}^{n \times n},$$

$$S_n^{-1} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}. \tag{3}$$

Given a prediction horizon $p \in \mathbb{N}$, the p-horizon MPC control is determined by solving the following constrained optimization problem at each $k \in \mathbb{Z}_+$, involving all vehicles' control inputs for given feasible state $(x_i(k), v_i(k))_{i=1}^n$ and $(x_0(k), v_0(k), u_0(k))$ at time k subject to the vehicle dynamics model (1):

$$\begin{aligned} &\text{minimize } J(u(k), \ldots, u(k+p-1)) := \\ &\frac{1}{2}\sum_{s=1}^{p} \Big(\underbrace{\tau^2 u^T(k+s-1) S_n^{-T} Q_{w,s} S_n^{-1} u(k+s-1)}_{\text{ride comfort}} \\ &\quad + \underbrace{z^T(k+s) Q_{z,s} z(k+s) + (z'(k+s))^T Q_{z',s} z'(k+s)}_{\text{traffic stability and smoothness}} \Big), \end{aligned} \tag{4}$$

subject to the following: for each $i = 1, \ldots, n$ and each $s = 1, \ldots, p$, $z_i(k+s) = x_{i-1}(k+s) - x_i(k+s)$, and $z_i'(k+s) = v_{i-1}(k+s) - v_i(k+s)$, where $x_i(k+s)$ and $v_i(k+s)$ are given in terms of $u_i(k), \ldots, u_i(k+p-1)$ as shown in the vehicle dynamics model (1), and

$$a_{\min} \le u_i(k+s-1) \le a_{\max}, \qquad v_{\min} \le v_i(k+s) \le v_{\max}, \tag{5}$$

$$x_{i-1}(k+s) - x_i(k+s) \ge L + r \cdot v_i(k+s) - \frac{(v_i(k+s) - v_{\min})^2}{2a_{\min}}, \tag{6}$$

where $Q_{z,s}$, $Q_{z',s}$ and $Q_{w,s}$ are $n \times n$ symmetric positive semidefinite weight matrices to be discussed soon. When $p > 1$, $(x_0(k+s+1), v_0(k+s+1), u_0(k+s))$ are unknown at time k for $s = 1, \ldots, p-1$. In this case, we assume that $u_0(k+s) = u_0(k)$ for all $s = 1, \ldots, p-1$ and use these $u_0(k+s)$'s and the vehicle dynamics model (1) to predict $(x_0(k+s+1), v_0(k+s+1))$ for $s = 1, \ldots, p-1$.

Remark 3.1. The three terms in the objective function J intend to minimize traffic flow oscillations via mild control. Particularly, the first term penalizes the magnitude of control for mild control and ride comfort, whereas the second and last terms penalize the variations of the relative spacing and relative speed to reduce traffic oscillations, respectively. The weight matrices $Q_{z,s}, Q_{z',s}$ and $Q_{w,s}$ in the previous J are chosen such that smooth traffic dynamics and asymptotic stability is achieved in the closed loop dynamics (see Section 5 for details). The presence of the matrix S_n in the first term is because of the coupled vehicle dynamics through the CAV platoon. To illustrate this, let $\widetilde{w}(k+s-1) := w(k+s-1) - u_0(k) \cdot \mathbf{e}_1$ for $s = 1, \ldots, p$. Thus, $\widetilde{w}(k+s-1) = -S_n^{-1} u(k+s-1)$ for each $s = 1, \ldots, p$. Therefore, the first term in J satisfies $\tau^2 u^T(k+s-1) S_n^{-T} Q_{w,s} S_n^{-1} u(k+s-1) = \tau^2 \widetilde{w}^T(k+s-1) Q_{w,s} \widetilde{w}(k+s-1)$ for each s. Last, nonconstant spacing car following polices can be considered (see Remark 4.3 for details).

The weight matrices $Q_{z,s}$, $Q_{z',s}$, and $Q_{w,s}$, $s = 1, \ldots, p$ determine transient and asymptotic dynamics, and they depend on vehicle network topologies and can be chosen by stability analysis and transient dynamics criteria of the closed loop system. To develop fully distributed schemes for a broad class of vehicle network topologies and to facilitate control design and analysis, we make the following blanket assumption on $Q_{z,s}$, $Q_{z',s}$, and $Q_{w,s}$ throughout the rest of the paper.

Assumption 2. *For each $s = 1, \ldots, p$, $Q_{z,s}$ and $Q_{z',s}$ are diagonal and positive semidefinite (PSD), and $Q_{w,s}$ is diagonal and positive definite (PD).*

The reasons for considering this class of diagonal positive semidefinite or positive definite weight matrices are three folds: (i) Diagonal matrices have a simpler interpretation in transportation engineering so that the selection of such matrices is easier to practitioners. For instance, the diagonal $Q_{z,s}$ and $Q_{z',s}$ mean that one imposes penalties on each element of $z(k+s)$ and $z'(k+s)$ without considering their coupling. Furthermore, by suitably choosing the weight matrices $Q_{w,s}$, it can be shown that the ride comfort term in Equation (4), which corresponds to acceleration of CAVs, is similar to imposing direct penalties on u_i's, which simplifies control design. (ii) This class of weight matrices facilitates the development of fully

distributed schemes for general vehicle network topologies. (iii) Closed-loop stability and performance analysis are relatively simpler (although still nontrivial) when using this class of weight matrices. The detailed discussions of choosing diagonal, positive semidefinite or positive definite weight matrices for satisfactory closed loop dynamics will be given in Section 5.

The sequential feasibility has been established in Gong and Du (2018) and Gong, Shen, and Du (2016) for the MPC model (4) when $r \geq \tau$. Define $\mathcal{P}((x_i, v_i)_{i=0}^n, u_0) := \{u \in \mathbb{R}^n \mid a_{\min} \leq u_i \leq a_{\max},\ v_{\min} \leq v_i + \tau u_i \leq v_{\max},\ h_i(u) \leq 0,\ \forall i = 1, \ldots, n\}$, where $h_i(u) := L + r(v_i + \tau u_i) - \frac{(v_i + \tau u_i - v_{\min})^2}{2a_{\min}} + (x_i - x_{i-1}) + \tau(v_i - v_{i-1}) + \frac{\tau^2}{2}[u_i - u_{i-1}]$ for each $i = 1, \ldots, n$. Specifically, the sequential feasibility implies that for any feasible $x_i(k), v_i(k), u_0(k)$ at time k, the constraint set $\mathcal{P}((x_i(k), v_i(k))_{i=0}^n, u_0(k))$ is nonempty such that the MPC model (4) has a solution $u_*(k)$ such that the constraint set $\mathcal{P}((x_i(k+1), v_i(k+1))_{i=0}^n, u_0(k+1))$ is nonempty. Using this result, we show below that under a mild assumption, the constraint sets of the MPC model have nonempty interior for any MPC horizon $p \in \mathbb{N}$. This result is important to the development of distributed algorithms.

Corollary 3.1. *Consider the linear vehicle dynamics (1) and assume $r \geq \tau$. Suppose the leading vehicle is such that $(v_0(k), u_0(k))$ is feasible and $v_0(k) > v_{\min}$ for all $k \in \mathbb{Z}_+$. Then the constraint set of the p-horizon MPC model (4) has nonempty interior at each k.*

Proof. Fix an arbitrary $k \in \mathbb{Z}_+$. Because $v_0(k) > v_{\min}$, it follows from Gong, Shen, and Du (2016, proposition 3.1) that there exists a vector denoted by $\widehat{u}(k)$ in the interior of the set $\mathcal{P}((x_i(k), v_i(k))_{i=0}^n, u_0(k))$. Let $x_i(k+1)$ and $v_i(k+1)$ be generated by $\widehat{u}(k)$ (and $(x_i(k), v_i(k))_{i=0}^n, u_0(k)$). Because $v_0(k+1) > v_{\min}$, we deduce via Gong, Shen, and Du (2016, proposition 3.1) again that there exists a vector denoted by $\widehat{u}(k+1)$ in the interior of the constraint set $\mathcal{P}((x_i(k+1), v_i(k+1))_{i=0}^n, u_0(k+1))$. Continuing this process in p-steps, we derive the existence of an interior point in the constraint set of the p-horizon MPC model (4). □

3.1. Constrained MPC Optimization Model

Consider the constrained MPC optimization model (4) at an arbitrary but fixed time $k \in \mathbb{Z}_+$ subject to the linear vehicle dynamics (1). In view of the following results: for each $s = 1, \ldots, p$,

$$v_i(k+s) = v_i(k) + \tau \sum_{j=0}^{s-1} u_i(k+j),$$

$$z'(k+s) = z'(k) + \tau \sum_{j=0}^{s-1} w(k+j),$$

$$z(k+s) = z(k) + s\tau z'(k) + \tau^2 \sum_{j=0}^{s-1} \frac{2(s-j)-1}{2} w(k+j),$$

$$w(k+s) = S_n^{-1}[-u(k+s) + u_0(k) \cdot \mathbf{1}],$$

we formulate (4) as the constrained convex minimization problem (where we omit k because it is fixed):

$$\begin{aligned} \text{minimize} \quad & J(\mathbf{u}) := \frac{1}{2}\mathbf{u}^T W \mathbf{u} + c^T \mathbf{u} + \gamma, \\ \text{subject to} \quad & \mathbf{u}_i \in \mathcal{X}_i, \quad (H_i(\mathbf{u}))_s \leq 0, \quad \forall i = 1, \ldots, n, \\ & \qquad\qquad\qquad\qquad\quad \forall s = 1, \ldots, p, \end{aligned} \tag{7}$$

where $\mathbf{u} := (\mathbf{u}_1, \ldots, \mathbf{u}_n) \in \mathbb{R}^{np}$ with $\mathbf{u}_i := (u_i(k), \ldots, u_i(k+p-1)) \in \mathbb{R}^p$, W is a PD matrix to be shown in Lemma 3.1, $c \in \mathbb{R}^{np}$, $\gamma \in \mathbb{R}$, each $\mathcal{X}_i := \{z \in \mathbb{R}^p \mid a_{\min} \cdot \mathbf{1} \leq z \leq a_{\max} \cdot \mathbf{1},\ (v_{\min} - v_i(k)) \cdot \mathbf{1} \leq \tau S_p z \leq (v_{\max} - v_i(k)) \cdot \mathbf{1}\}$ is a polyhedral set, and each $(H_i(\cdot))_s$ is a convex quadratic function characterizing the safety distance given by (12). Here S_p is the $p \times p$ matrix of the form given by (3). Furthermore, $\mathbf{u}_0 := u_0(k) \cdot \mathbf{1}_p \in \mathbb{R}^p$ for the given $u_0(k)$. An important property of the matrix W in (7) is given later.

Lemma 3.1. *Suppose that $Q_{z,s}$ and $Q_{z',s}$ are PSD and $Q_{w,s}$ are PD for all $s = 1, \ldots, p$ (but not necessarily diagonal). Then the matrix W in (7) is PD.*

Proof. Let $\mathbf{u}$ be an arbitrary nonzero vector in $\mathbb{R}^{np}$. Because $J(\cdot)$ is quadratic, we have $\frac{1}{2}\mathbf{u}^T W \mathbf{u} = \lim_{\lambda \to \infty} \frac{J(\lambda \mathbf{u})}{\lambda^2}$. In view of the equivalent formulation of $J(\cdot)$ given by (4), we deduce that for any $\lambda > 0$, $J(\lambda \mathbf{u}) = J(\lambda u(k), \ldots, \lambda u(k+p-1)) \geq \frac{\lambda^2}{2} \sum_{s=1}^p \tau^2 u^T(k+s-1) S_n^{-T} Q_{w,s} S_n^{-1} u(k+s-1) > 0$, where the first inequality follows from the fact that $Q_{z,s}$ and $Q_{z',s}$ are PSD, and the second inequality holds because $Q_{w,s}$, and thus $S_n^{-T} Q_{w,s} S_n^{-1}$, are PD. Therefore, $\frac{J(\lambda \mathbf{u})}{\lambda^2} \geq \frac{1}{2} \sum_{s=1}^p \tau^2 u^T(k+s-1) S_n^{-1} Q_{w,s} S_n^{-1} u(k+s-1) > 0$, leading to $\frac{1}{2}\mathbf{u}^T W \mathbf{u} \geq \frac{1}{2} \sum_{s=1}^p \tau^2 u^T(k+s-1) S_n^{-1} Q_{w,s} S_n^{-1} u(k+s-1) > 0$. Hence, W is PD. □

To establish the closed form expressions of the matrix W and the vector c in (7), we define the following matrices for any $i, j \in \{1, \ldots, p\}$:

$$V_{i,j} := S_n^{-T} \left[\sum_{s=\max(i,j)}^{p} \left(\frac{\tau^4}{4}[2(s-i)+1] \cdot [2(s-j)+1] Q_{z,s} + \tau^2 Q_{z',s} \right) \right] S_n^{-1} \in \mathbb{R}^{n \times n}.$$

Clearly, $V_{i,j} = V_{j,i}$ for any i, j. Moreover, let $\widetilde{Q}_{w,s} := S_n^{-T} Q_{w,s} S_n^{-1}$ for $s = 1, \ldots, p$. Hence, the symmetric

matrix W is given by $W = E^T V E$, where

$$V = \begin{bmatrix} V_{1,1}+\tau^2\widetilde{Q}_{w,1} & V_{1,2} & V_{1,3} & \cdots & \cdots & V_{1,p} \\ V_{2,1} & V_{2,2}+\tau^2\widetilde{Q}_{w,2} & V_{2,3} & \cdots & \cdots & V_{2,p} \\ \cdots & & \cdots & & \cdots & \\ \cdots & & \cdots & & \cdots & \\ V_{p,1} & V_{p,2} & V_{p,3} & \cdots & \cdots & V_{p,p}+\tau^2\widetilde{Q}_{w,p} \end{bmatrix} \in \mathbb{R}^{np\times np}, \tag{8}$$

and $E \in \mathbb{R}^{np\times np}$ is the permutation matrix satisfying

$$\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+p-1) \end{bmatrix} = E \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{bmatrix}.$$

Specifically, the (i,j)-entry of the matrix E is given by

$$E_{i,j} = \begin{cases} 1 & \text{if } \ i = n\cdot k + s,\ j = p\cdot(s-1)+k+1, \\ & \quad \text{for } \ k = 0,\dots,p-1,\ s = 1,\dots,n; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

In particular, when $p = 1$, we have $E = I_n$.

For a fixed $k \in \mathbb{Z}_+$, we also define for each $s = 1,\dots,p$,

$$d_s(k) := z(k) + s\tau z'(k) + \tau^2 \sum_{j=0}^{s-1} \frac{2(s-j)-1}{2} S_n^{-1}\cdot \mathbf{1}\cdot u_0(k),$$

$$f_s(k) := z'(k) + \tau \sum_{j=0}^{s-1} S_n^{-1}\cdot \mathbf{1}\cdot u_0(k).$$

In light of S_n^{-1} given by (3), we have $S_n^{-1}\cdot\mathbf{1} = \mathbf{e}_1$. Therefore, we obtain

$$\begin{aligned} d_s(k) &= z(k) + s\tau z'(k) + \frac{\tau^2}{2} s^2 \mathbf{e}_1 u_0(k), \\ f_s(k) &= z'(k) + \tau s \mathbf{e}_1 u_0(k). \end{aligned} \tag{10}$$

In view of

$$z(k+s) = d_s(k) - \tau^2 \sum_{j=0}^{s-1} \frac{2(s-j)-1}{2} S_n^{-1} u(k+j),$$

$$z'(k+s) = f_s(k) - \tau \sum_{j=0}^{s-1} S_n^{-1} u(k+j),$$

the linear terms in the objective function J are given by

$$-\sum_{i=1}^{p}\left(\sum_{s=i}^{p}\left[\frac{\tau^2}{2}[2(s-i)+1]d_s^T(k)Q_{z,s} + \tau f_s^T(k) Q_{z',s}\right]\right) S_n^{-1} \cdot u(k+i-1). \tag{11}$$

Using the permutation matrix E given in (9), we can write $c^T\mathbf{u}$ as $c^T\mathbf{u} = \sum_{i=1}^n c_{\mathcal{I}_i}^T \mathbf{u}_i$, where $c_{\mathcal{I}_i}$ is the subvector of c corresponding to $\mathbf{u}_i$. Because $Q_{z,s}$ and $Q_{z',s}$ are diagonal, it is easy to obtain the following lemma via $d_s(k), f_s(k)$ in (10) and the structure of S_n^{-1} given by (3).

Lemma 3.2. *Consider the vector $c = (c_{\mathcal{I}_1},\dots,c_{\mathcal{I}_n})$ given previously. Then for each $i = 1,\dots,n$, the subvector $c_{\mathcal{I}_i}$ depends only on $z_i(k), z_i'(k), z_{i+1}(k), z_{i+1}'(k)$'s for $i = 1,\dots,n-1$, and $c_{\mathcal{I}_n}$ depends only on $z_n(k), z_n'(k)$. Furthermore, only $c_{\mathcal{I}_1}$ depends on $u_0(k)$.*

The previous lemma shows that each $c_{\mathcal{I}_i}$ only depends on the information of the adjacent vehicles of vehicle i, and thus can be easily established from any vehicle network. This property is important for developing fully distributed schemes to be shown in Section 4.3.

To find the vector form of the safety constraint, we note that for $s = 1,\dots,p$,

$$x_i(k+s) = x_i(k) + s\tau v_i(k) + \tau^2 \sum_{j=0}^{s-1} \frac{2(s-j)-1}{2} u_i(k+j),$$

$$v_i(k+s) = v_i(k) + \tau \sum_{j=0}^{s-1} u_i(k+j).$$

The safety distance constraint for the ith vehicle at time k is given by: for $s = 1,\dots,p$,

$$\begin{aligned} 0 \ge & -[x_{i-1}(k) + s\tau v_{i-1}(k) - (x_i(k) + s\tau v_i(k))] \\ & - \tau^2 \sum_{j=0}^{s-1} \frac{2(s-j)-1}{2} [u_{i-1}(k+j) - u_i(k+j)] + L + r v_i(k) \\ & + r\tau \sum_{j=0}^{s-1} u_i(k+j) \\ & - \frac{1}{2a_{\min}}\left[\tau^2\left(\sum_{j=0}^{s-1} u_i(k+j)\right)^2 + 2\tau(v_i(k) - v_{\min}) \right. \\ & \left. \sum_{j=0}^{s-1} u_i(k+j) + (v_i(k) - v_{\min})^2\right] := (H_i(\mathbf{u}_{i-1},\mathbf{u}_i))_s, \end{aligned} \tag{12}$$

where $(H_i(\cdot,\cdot))_s$ is a convex quadratic function for each $s = 1,\dots,p$. Hence, the set $\mathcal{Z}_i := \{\mathbf{u} \in \mathbb{R}^{np} \mid (H_i(\mathbf{u}_{i-1},\mathbf{u}_i))_s \le 0,\ \forall\, i = 1,\dots,p\}$ is closed and convex. Problem (7) becomes $\min_{\mathbf{u}} J(\mathbf{u})$ subject to $\mathbf{u}_i \in \mathcal{X}_i$ and $\mathbf{u} \in \mathcal{Z}_i$ for all $i = 1,\dots,n$, which is a convex quadratically constrained quadratic program (QCQP) and can be solved via a second-order cone program or a semidefinite program in the centralized manner.

Remark 3.2. The previous results show that each $\mathcal{X}_i$ is decoupled from the other vehicles, whereas the constraint function H_i for vehicle i is locally coupled with its neighboring vehicles. Specifically, H_i depends not only on $\mathbf{u}_i$ but also on $\mathbf{u}_{i-1}$ of vehicle $(i-1)$, which can exchange information with vehicle i. We will explore this local coupling property to develop fully distributed schemes for solving (7) in the next section.

4. Operator Splitting Method–Based Fully Distributed Algorithms for Constrained Optimization in MPC

We develop fully distributed algorithms for solving the underlying optimization problem given by (7) at each time k using the techniques of locally coupled convex optimization and operator splitting methods.

4.1. Brief Overview of Locally Coupled Convex Optimization

One of major techniques for developing fully distributed schemes for the underlying optimization problem given by (7) is to formulate it as a locally coupled convex optimization problem (Hu, Xiao, and Liu 2018). To be self-contained, we briefly describe this formulation as follows.

Consider a multiagent network of n agents whose communication is characterized by a connected and undirect graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{1,\dots,n\}$ is the set of agents, and $\mathcal{E}$ denotes the set of edges. For $i\in\mathcal{V}$, let $\mathcal{N}_i$ be the set of neighbors of agent i, i.e., $\mathcal{N}_i=\{j\,|\,(i,j)\in\mathcal{E}\}$. Let $\{\mathcal{I}_1,\dots,\mathcal{I}_n\}$ be a disjoint union of the index set $\{1,\dots,N\}$. Hence, for any $x\in\mathbb{R}^N$, $(x_{\mathcal{I}_i})_{i=1}^n$ forms a partition of x. We call $x_{\mathcal{I}_i}$ a local variable of each agent i. For each i, define $\widehat{x}_i:=(x_{\mathcal{I}_i},(x_{\mathcal{I}_j})_{j\in\mathcal{N}_i})\in\mathbb{R}^{n_i}$. Thus, each $\widehat{x}_i$ contains the local variable $x_{\mathcal{I}_i}$ and the variables from its neighboring agents (or locally coupled variables). Consider the convex optimization problem

$$(P):\qquad \min_{x\in\mathbb{R}^N} J(x),\qquad \text{where}\quad J(x):=\sum_{i=1}^n J_i(\widehat{x}_i),$$

where $J_i:\mathbb{R}^{n_i}\to\mathbb{R}\cup\{+\infty\}$ is an extended-valued, proper, and lower semicontinuous convex function for each i. Clearly, each J_i is locally coupled such that the problem (P) bears the name "locally coupled convex optimization." Although the problem (P) is seemingly unconstrained, it does include constrained convex optimization because J_i may contain the indicator function of a closed convex set. To impose the locally coupled convex constraint explicitly, the problem (P) can be equivalently written as

$$(P'):\qquad \min_{x\in\mathbb{R}^N}\sum_{i=1}^n \widehat{J}_i(\widehat{x}_i),\quad \text{subject to}\quad \widehat{x}_i\in\mathcal{C}_i,\quad \forall i=1,\dots,n, \tag{13}$$

where each $\widehat{J}_i:\mathbb{R}^{n_i}\to\mathbb{R}$ is a real-valued convex function, and $\mathcal{C}_i\subseteq\mathbb{R}^{n_i}$ is a closed convex set.

By introducing copies of the locally coupled variables for each agent and imposing certain consensus constraints on these copies, Hu, Xiao, and Liu (2018) formulate the problem (P') (or equivalently (P)) as a separable consensus convex optimization problems. Under suitable assumptions, Douglas-Rachford and other operator splitting–based distributed schemes are developed (Hu, Xiao, and Liu 2018).

4.2. Decomposition of a Strongly Convex Quadratic Objective Function

The framework of locally coupled optimization problems requires that both an objective function and constraints are expressed in a locally coupled manner. Especially, the central objective function in (13) is expected to be written as the sum of multiple locally coupled functions preserving certain desired properties, for example, the (strong) convexity if the central objective function is so, where local coupling satisfies network topology constraints. Although the constraints of Problem (7) have been shown to be locally coupled (Remark 3.2), the central strongly convex quadratic objective function, particularly its quadratic term $\frac{1}{2}\mathbf{u}^TW\mathbf{u}$, is densely coupled and thus need to be decomposed into the sum of locally coupled (strongly) convex quadratic functions, where the local coupling should satisfy the network topology constraint. In this section, we address this decomposition problem under a mild assumption on network topology.

We start from a slightly general setting. Let $\boldsymbol{\lambda}:=(\lambda_1,\dots,\lambda_n)\in\mathbb{R}^n$ and $\Lambda=\operatorname{diag}(\boldsymbol{\lambda})=\operatorname{diag}(\lambda_1,\dots,\lambda_n)$ be a diagonal matrix; that is, $\boldsymbol{\lambda}$ is the vector representation of the diagonal entries of Λ. Therefore, the following matrix is tridiagonal:

$$S_n^{-T}\Lambda S_n^{-1}=\begin{bmatrix}\lambda_1+\lambda_2 & -\lambda_2 & & & \\ -\lambda_2 & \lambda_2+\lambda_3 & -\lambda_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -\lambda_{n-1} & \lambda_{n-1}+\lambda_n & -\lambda_n \\ & & & -\lambda_n & \lambda_n\end{bmatrix}. \tag{14}$$

Consider a general $p\in\mathbb{N}$. Let Θ be a symmetric block diagonal matrix given by

$$\Theta=\begin{bmatrix}\Theta_{1,1} & \Theta_{1,2} & \cdots & \cdots & \Theta_{1,p}\\ \Theta_{2,1} & \Theta_{2,2} & \cdots & \cdots & \Theta_{2,p}\\ \cdots & & \cdots & & \cdots\\ \cdots & & \cdots & & \cdots\\ \Theta_{p,1} & \Theta_{p,2} & \cdots & \cdots & \Theta_{p,p}\end{bmatrix}\in\mathbb{R}^{np\times np},$$

where $\Theta_{i,j}=\operatorname{diag}(\boldsymbol{\theta}_{i,j})\in\mathbb{R}^{n\times n}$ is diagonal for some $\boldsymbol{\theta}_{i,j}\in\mathbb{R}^n$, and $\boldsymbol{\theta}_{i,j}=\boldsymbol{\theta}_{j,i}$ for all $i,j=1,\dots,p$. Let $(\boldsymbol{\theta}_{i,j})_k$ denote the kth entry of the vector $\boldsymbol{\theta}_{i,j}$. For each $i=1,\dots,n$, define the matrix

$$U_i:=\begin{bmatrix}(\boldsymbol{\theta}_{1,1})_i & (\boldsymbol{\theta}_{1,2})_i & \cdots & (\boldsymbol{\theta}_{1,p})_i\\ (\boldsymbol{\theta}_{2,1})_i & (\boldsymbol{\theta}_{2,2})_i & \cdots & (\boldsymbol{\theta}_{2,p})_i\\ \cdots & \cdots & \cdots & \\ (\boldsymbol{\theta}_{p,1})_i & (\boldsymbol{\theta}_{p,2})_i & \cdots & (\boldsymbol{\theta}_{p,p})_i\end{bmatrix}\in\mathbb{R}^{p\times p}. \tag{15}$$

It can be shown that $\Theta = E^T \mathrm{diag}(U_1, \ldots, U_n)E$, where E is the permutation matrix given by (9). Hence, Θ is PD (respectively, PSD) if and only if each U_i is PD (respectively, PSD).

Let

$$V = \underbrace{\begin{bmatrix} S_n^{-T} & & & \\ & S_n^{-T} & & \\ & & \ddots & \\ & & & S_n^{-T} \end{bmatrix}}_{:=\mathbf{S}^{-T}} \Theta \underbrace{\begin{bmatrix} S_n^{-1} & & & \\ & S_n^{-1} & & \\ & & \ddots & \\ & & & S_n^{-1} \end{bmatrix}}_{:=\mathbf{S}^{-1}} = \begin{bmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,p} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,p} \\ \cdots & & \cdots & \cdots \\ V_{p,1} & V_{p,2} & \cdots & V_{p,p} \end{bmatrix},$$

where $V_{i,j} := S_n^{-T}\Theta_{i,j}S_n^{-1}$ is symmetric and tridiagonal. Letting E be the permutation matrix given by (9), a straightforward computation shows that E^TVE is a symmetric block tridiagonal matrix given by

$$W = E^TVE = \begin{bmatrix} W_{1,1} & W_{1,2} & & & & \\ W_{2,1} & W_{2,2} & W_{2,3} & & & \\ & \ddots & \ddots & \ddots & & \\ & & W_{n-1,n-2} & W_{n-1,n-1} & W_{n-1,n} & \\ & & & W_{n,n-1} & W_{n,n} \end{bmatrix} \in \mathbb{R}^{np\times np},$$

where each $W_{i,j} \in \mathbb{R}^{p\times p}$ is symmetric and $W_{i,j} = W_{j,i}$. Furthermore, for each $i = 1, \ldots, n$ and $j \in \{i, i+1\}$,

$$W_{i,j} = \begin{bmatrix} (V_{1,1})_{i,j} & (V_{1,2})_{i,j} & \cdots & (V_{1,p})_{i,j} \\ (V_{2,1})_{i,j} & (V_{2,2})_{i,j} & \cdots & (V_{2,p})_{i,j} \\ \cdots & \cdots & \cdots & \\ (V_{p,1})_{i,j} & (V_{p,2})_{i,j} & \cdots & (V_{p,p})_{i,j} \end{bmatrix} \in \mathbb{R}^{p\times p},$$

where $(V_{r,s})_{i,j}$ denotes the (i, j)-entry of the block $V_{r,s}$. In view of $V_{i,j} = S_n^{-T}\Theta_{i,j}S_n^{-1}$ and (14), we have that $W_{i,i} = U_i + U_{i+1}$ and $W_{i,i+1} = -U_{i+1}$ for $i = 1, \ldots, n-1$, and $W_{n,n} = U_n$. Moreover, because $W = E^TVE = E^T\mathbf{S}^{-T}\Theta\,\mathbf{S}^{-1}E$, W is PD (respectively, PSD) if and only if Θ is PD (respectively, PSD), which is also equivalent to that each U_i is PD (respectively, PSD) (see the comment after (15)).

In what follows, we consider PSD (respectively, PD) matrix decomposition for a PSD (respectively, PD) W generated by $\boldsymbol{\theta}_{i,j} \in \mathbb{R}^n_+$ for $i = 1, \ldots, n$ and $j \geq i$. The goal of this decomposition is to construct PSD matrices $\widetilde{W}^s \in \mathbb{R}^{np\times np}$ for $s = 1, \ldots, n$ such that the following conditions hold:

(i)

$$\widetilde{W}^1 = \begin{bmatrix} (\widetilde{W}^1)_{1,1} & (\widetilde{W}^1)_{1,2} & & & & \\ (\widetilde{W}^1)_{2,1} & (\widetilde{W}^1)_{2,2} & & & & \\ & & 0 & \cdots & 0 \\ & & \vdots & \cdots & \vdots \\ & & 0 & \cdots & 0 \end{bmatrix},$$

$$\widetilde{W}^n = \begin{bmatrix} 0 & \cdots & 0 & & \\ \vdots & \cdots & \vdots & & \\ 0 & \cdots & 0 & & \\ & & & (\widetilde{W}^n)_{n-1,n-1} & (\widetilde{W}^n)_{n-1,n} \\ & & & (\widetilde{W}^n)_{n,n-1} & (\widetilde{W}^n)_{n,n} \end{bmatrix};$$

(ii) for each $s = 2, \ldots, n-1$,

$$\widetilde{W}^s = \begin{bmatrix} \mathbf{0}_{(s-2)p\times(s-2)p} & & & & \\ & (\widetilde{W}^s)_{s-1,s-1} & (\widetilde{W}^s)_{s-1,s} & 0 & \\ & (\widetilde{W}^s)_{s,s-1} & (\widetilde{W}^s)_{s,s} & (\widetilde{W}^s)_{s,s+1} & \\ & 0 & (\widetilde{W}^s)_{s+1,s} & (\widetilde{W}^s)_{s+1,s+1} & \\ & & & & 0 \cdots 0 \\ & & & & \vdots \cdots \vdots \\ & & & & 0 \cdots 0 \end{bmatrix}; \text{ and}$$

(iii) $W = \sum_{s=1}^n \widetilde{W}^s$.

For notational simplicity, let $\widehat{W}^s$ denote the possibly nonzero block in each $\widetilde{W}^s$. Specifically,

$$\widehat{W}^1 := \begin{bmatrix} (\widetilde{W}^1)_{1,1} & (\widetilde{W}^1)_{1,2} \\ (\widetilde{W}^1)_{2,1} & (\widetilde{W}^1)_{2,2} \end{bmatrix} \in \mathbb{R}^{2p\times 2p},$$

$$\widehat{W}^n := \begin{bmatrix} (\widetilde{W}^n)_{n-1,n-1} & (\widetilde{W}^n)_{n-1,n} \\ (\widetilde{W}^n)_{n,n-1} & (\widetilde{W}^n)_{n,n} \end{bmatrix} \in \mathbb{R}^{2p\times 2p},$$

and for each $s = 2, \ldots, n-1$,

$$\widehat{W}^s := \begin{bmatrix} (\widetilde{W}^s)_{s-1,s-1} & (\widetilde{W}^s)_{s-1,s} & 0 \\ (\widetilde{W}^s)_{s,s-1} & (\widetilde{W}^s)_{s,s} & (\widetilde{W}^s)_{s,s+1} \\ 0 & (\widetilde{W}^s)_{s+1,s} & (\widetilde{W}^s)_{s+1,s+1} \end{bmatrix} \in \mathbb{R}^{3p\times 3p}.$$

When W is PD, we also want each $\widehat{W}^s$ in the previous decomposition to be PD.

Proposition 4.1. *Let W be a PSD matrix generated by $\boldsymbol{\theta}_{i,j} \in \mathbb{R}^n_+$ for $i = 1, \ldots, n$. Then there exist PSD matrices $\widetilde{W}^s$, $s = 1, \ldots, n$ satisfying the above conditions. Moreover, suppose W is PD. Then there exist PD matrices $\widehat{W}^s$, $s = 1, \ldots, n$ such that their corresponding $\widetilde{W}^s$s satisfy the above conditions.*

Proof. Let W be generated by $\boldsymbol{\theta}_{i,j}$ s such that W is PSD, and let U_is be defined in (15) corresponding to $\boldsymbol{\theta}_{i,j}$.

Each U_i is PSD as W is PSD. Let

$$\widetilde{W}^1 = \begin{bmatrix} U_1 & 0 & & & \\ 0 & 0 & & & \\ & & 0 & \cdots & 0 \\ & & \vdots & \cdots & \vdots \\ & & 0 & \cdots & 0 \end{bmatrix}, \quad \widetilde{W}^n = \begin{bmatrix} 0 & \cdots & 0 & & \\ \vdots & \cdots & \vdots & & \\ 0 & \cdots & 0 & & \\ & & & U_n & -U_n \\ & & & -U_n & U_n \end{bmatrix},$$

and for each $s = 2, \ldots, n-1$,

$$\widetilde{W}^s = \begin{bmatrix} \mathbf{0}_{(s-2)p\times(s-2)p} & & & & & & \\ & U_s & -U_s & 0 & & & \\ & -U_s & U_s & 0 & & & \\ & 0 & 0 & 0 & & & \\ & & & & 0 & \cdots & 0 \\ & & & & \vdots & \cdots & \vdots \\ & & & & 0 & \cdots & 0 \end{bmatrix}.$$

Because each U_i is PSD, so is $\widetilde{W}^s$ for each $s = 1, \ldots, n$. Clearly, $W = \sum_{s=1}^n \widetilde{W}^s$.

Now suppose W is PD. Hence, each U_i given by (15) is PD. Define

$$\breve{W}^1 := \frac{1}{2}\begin{bmatrix} U_1 + U_2 & -U_2 \\ -U_2 & U_2 \end{bmatrix},$$

$$\breve{W}^2 := \frac{1}{2}\begin{bmatrix} U_1 + U_2 & -U_2 & 0 \\ -U_2 & U_2 + U_3 & -U_3 \\ 0 & -U_3 & U_3 \end{bmatrix},$$

$$\breve{W}^n := \frac{1}{2}\begin{bmatrix} U_n & -U_n \\ -U_n & U_n \end{bmatrix},$$

and for each $s = 3, \ldots, n-1$,

$$\breve{W}^s := \frac{1}{2}\begin{bmatrix} U_s & -U_s & 0 \\ -U_s & U_s + U_{s+1} & -U_{s+1} \\ 0 & -U_{s+1} & U_{s+1} \end{bmatrix} \in \mathbb{R}^{3p\times 3p}.$$

Note that $\breve{W}^1 = \frac{1}{2}\left\{\begin{bmatrix} U_1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} U_2 & -U_2 \\ -U_2 & U_2 \end{bmatrix}\right\}$ and the two matrices on the right-hand side are both PSD and the intersection of their null spaces is the zero subspace. Hence, $\breve{W}^1$ is PD. Similarly, $\breve{W}^2$ is PD, and the other $\breve{W}^s$ s are PSD. Because $\breve{W}^1$ is PD, we see that for an arbitrary $\delta_1 \in (0, \lambda_{\min}(\breve{W}^1))$, $\widehat{W}^1 := \breve{W}^1 - \delta_1 \cdot I_{2p}$ is PD. Hence,

$$\breve{W}^2 := \breve{W}^2 + \delta_1 \cdot \begin{bmatrix} I_p & & \\ & I_p & \\ & & 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} U_1 + U_2 + 2\delta_1 \cdot I_p & -U_2 & 0 \\ -U_2 & U_2 + U_3 + 2\delta_1 \cdot I_p & -U_3 \\ 0 & -U_3 & U_3 \end{bmatrix}$$

is also PD. Therefore, for an arbitrary $\delta_2 \in (0, \lambda_{\min}(\grave{W}^2))$, the matrix $\widehat{W}^2 := \grave{W}^2 - \delta_2 \cdot \begin{bmatrix} 0 & & \\ & I_p & \\ & & I_p \end{bmatrix}$ is PD. Furthermore, it is easy to show that the matrix $\grave{W}^3 := \grave{W}^3 + \delta_2 \cdot \begin{bmatrix} I_p & & \\ & I_p & \\ & & 0 \end{bmatrix}$ is PD such that for any $\delta_3 \in (0, \lambda_{\min}(\breve{W}^3))$, the matrix $\widehat{W}^3 := \grave{W}^3 - \delta_3 \cdot \begin{bmatrix} 0 & & \\ & I_p & \\ & & I_p \end{bmatrix}$ is PD. Continuing this process by induction, we see that $\widehat{W}^s$ is PD for all $s = 4, \ldots, n-1$ and $\widehat{W}^{n-1} := \grave{W}^{n-1} - \delta_{n-1} \cdot \begin{bmatrix} 0 & & \\ & I_p & \\ & & I_p \end{bmatrix}$ is PD for an arbitrary $\delta_{n-1} \in (0, \lambda_{\min}(\grave{W}^{n-1}))$, where $\grave{W}^{n-1}$ is PD. Finally, define $\widehat{W}^n := \breve{W}^n + \delta_{n-1} \cdot I_{2p}$, which is clearly PD. Using these PD $\widehat{W}^s$, $s = 1, \ldots, n$, we construct $\widetilde{W}^s$ by setting the possibly nonzero block in each $\widetilde{W}^s$ as $\widehat{W}^s$. Specifically,

$$\begin{bmatrix} (\widetilde{W}^1)_{1,1} & (\widetilde{W}^1)_{1,2} \\ (\widetilde{W}^1)_{2,1} & (\widetilde{W}^1)_{2,2} \end{bmatrix} = \widehat{W}^1 \in \mathbb{R}^{2p\times 2p},$$

$$\begin{bmatrix} (\widetilde{W}^n)_{n-1,n-1} & (\widetilde{W}^n)_{n-1,n} \\ (\widetilde{W}^n)_{n,n-1} & (\widetilde{W}^n)_{n,n} \end{bmatrix} = \widehat{W}^n \in \mathbb{R}^{2p\times 2p},$$

and for each $s = 2, \ldots, n-1$,

$$\begin{bmatrix} (\widetilde{W}^s)_{s-1,s-1} & (\widetilde{W}^s)_{s-1,s} & 0 \\ (\widetilde{W}^s)_{s,s-1} & (\widetilde{W}^s)_{s,s} & (\widetilde{W}^s)_{s,s+1} \\ 0 & (\widetilde{W}^s)_{s+1,s} & (\widetilde{W}^s)_{s+1,s+1} \end{bmatrix} = \widehat{W}^s \in \mathbb{R}^{3p\times 3p}.$$

A straightforward calculation shows that $W = \sum_{s=1}^n \widetilde{W}^s$, yielding the desired result. □

To obtain the desired decomposition using the previous proposition, we observe that the matrix V in (8) is given by $\mathbf{S}^{-T}\Theta\mathbf{S}^{-1}$ for some matrix Θ of the form given after (14) whose blocks are positive combinations of $Q_{z,s}, Q_{z',s}$ and $Q_{w,s}$. Because $Q_{z,s}$ and $Q_{z',s}$ are diagonal and PSD and $Q_{w,s}$ are diagonal and PD, each block of Θ is diagonal and PD or PSD. Moreover, by Lemma 3.1, W is PD. Hence, there are uncountably many ways to construct positive δ_s, and thus PD $\widehat{W}^s$, as shown in the previous proposition. Therefore, we obtain the following strongly convex decomposition for the objective function J in (7), where we set the

constant $\gamma = 0$ without loss of generality:

$$J(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T W\mathbf{u} + c^T\mathbf{u} = \sum_{i=1}^{n}\frac{1}{2}\mathbf{u}^T\widetilde{W}^i\mathbf{u} + \sum_{i=1}^{n} c_{\mathcal{I}_i}^T\mathbf{u}_i$$
$$= J_1(\mathbf{u}_1,\mathbf{u}_2) + \sum_{i=2}^{n-1} J_i(\mathbf{u}_{i-1},\mathbf{u}_i,\mathbf{u}_{i+1}) + J_n(\mathbf{u}_{n-1},\mathbf{u}_n),$$

where the strongly convex functions J_i are given by

$$J_1(\mathbf{u}_1,\mathbf{u}_2) := \frac{1}{2}\begin{bmatrix}\mathbf{u}_1^T & \mathbf{u}_2^T\end{bmatrix}\widehat{W}_1\begin{bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\end{bmatrix} + c_{\mathcal{I}_1}^T\mathbf{u}_1,$$
$$J_i(\mathbf{u}_{i-1},\mathbf{u}_i,\mathbf{u}_{i+1}) := \frac{1}{2}\begin{bmatrix}\mathbf{u}_{i-1}^T & \mathbf{u}_i^T & \mathbf{u}_{i+1}^T\end{bmatrix}\widehat{W}_i\begin{bmatrix}\mathbf{u}_{i-1}\\ \mathbf{u}_i\\ \mathbf{u}_{i+1}\end{bmatrix} + c_{\mathcal{I}_i}^T\mathbf{u}_i, \quad \forall i = 2,\dots,n-1,$$
$$J_n(\mathbf{u}_{n-1},\mathbf{u}_n) := \frac{1}{2}\begin{bmatrix}\mathbf{u}_{n-1}^T & \mathbf{u}_n^T\end{bmatrix}\widehat{W}_n\begin{bmatrix}\mathbf{u}_{n-1}\\ \mathbf{u}_n\end{bmatrix} + c_{\mathcal{I}_n}^T\mathbf{u}_n. \tag{16}$$

Remark 4.1. The previous decomposition method is applicable to any vehicle communication network satisfying the assumption **A**.1 in Section 2, that is, $(i, i+1) \in \mathcal{E}$ for all $i = 1,\dots,n-1$. Besides, various alternative approaches can be developed to construct PD matrices $\widehat{W}^s$ using the similar idea given in the previous proposition. Furthermore, a similar decomposition method can be developed for other vehicle communication networks different from the cyclic-like graph. For instance, if such a graph contains edges other than $(i, i+1) \in \mathcal{E}$, one can add or subtract certain small terms pertaining to these extra edges in relevant matrices, which will preserve the desired PD property.

In what follows, we write each J_i as $J_i(\mathbf{u}_i, (\mathbf{u}_j)_{j\in\mathcal{N}_i})$ for notational convenience, where $\mathcal{N}_i$ denotes the set of neighbors of vehicle i in a vehicle network such that $i-1, i+1 \in \mathcal{N}_i$ for $i = 2,\dots,n-1$ and $2 \in \mathcal{N}_1, n-1 \in \mathcal{N}_n$.

4.3. Operator Splitting Method–Based Fully Distributed Algorithms

For illustration simplicity, we consider the cyclic like network topology through this subsection, although the proposed schemes can be easily extended to other network topologies under a suitable assumption (Remark 4.1). In this case, $\mathcal{N}_1 = \{2\}$, $\mathcal{N}_n = \{n-1\}$, and $\mathcal{N}_i = \{i-1, i+1\}$ for $i = 2,\dots,n-1$.

Define the constraint set

$$\mathcal{P} := \{\mathbf{u} = (\mathbf{u}_1,\dots,\mathbf{u}_n) \in \mathbb{R}^{np} \mid \mathbf{u}_i \in \mathcal{X}_i,\ \mathbf{u} \in \mathcal{Z}_i,\ i = 1,\dots,n\}.$$

Recall that $\mathcal{P}$ is defined by convex quadratic functions. The underlying optimization problem (7) at time k becomes $\min_{\mathbf{u}} J(\mathbf{u})$ subject to $\mathbf{u} \in \mathcal{P}$.

We formulate this problem as a locally coupled convex optimization problem (Hu, Xiao, and Liu 2018) and solve it using fully distributed algorithms. Specifically, in view of the decompositions given by (16), the objective function in (7) can be written as

$$J(\mathbf{u}) = \sum_{i=1}^{n} J_i(\mathbf{u}_i, (\mathbf{u}_j)_{j\in\mathcal{N}_i}).$$

In view of Remark 3.2, the safety constraints are also locally coupled. Let $\mathbf{I}_S$ denote the indicator function of a (closed convex) set S. Define, for each $i = 1,\dots,n$,

$$\widehat{J}_i(\mathbf{u}_i, (\mathbf{u}_j)_{j\in\mathcal{N}_i}) := J_i(\mathbf{u}_i, (\mathbf{u}_j)_{j\in\mathcal{N}_i}) + \mathbf{I}_{\mathcal{X}_i}(\mathbf{u}_i) + \mathbf{I}_{\mathcal{Z}_i}(\mathbf{u}_{i-1}, \mathbf{u}_i).$$

As in Hu, Xiao, and Liu (2018), define $\widehat{\mathbf{u}}_i := (\mathbf{u}_i, (\mathbf{u}_{i,j})_{j\in\mathcal{N}_i})$, where the new variables $\mathbf{u}_{i,j}$ represent the predicted values of $\mathbf{u}_j$ of vehicle j in the neighbor of vehicle i, and let $\widehat{\mathbf{u}} := (\widehat{\mathbf{u}}_i)_{i=1,\dots,n} \in \mathbb{R}^{\ell}$. Define the consensus subspace

$$\mathcal{A} := \{\widehat{\mathbf{u}} \mid \mathbf{u}_{i,j} = \mathbf{u}_j,\ \forall (i,j) \in \mathcal{E}\}.$$

Then the underlying optimization problem (7) can be equivalently written as

$$\min_{\widehat{\mathbf{u}}} \sum_{i=1}^{n} \widehat{J}_i(\widehat{\mathbf{u}}_i), \quad \text{subject to} \quad \widehat{\mathbf{u}} \in \mathcal{A}.$$

Let $\mathcal{P}_i := \{\widehat{\mathbf{u}}_i \mid \mathbf{u}_i \in \mathcal{X}_i, (H_i(\mathbf{u}_{i,i-1}, \mathbf{u}_i))_s \le 0,\ \forall s = 1,\dots,p\}$ for notational simplicity. Then the underlying optimization problem becomes

$$\min_{\widehat{\mathbf{u}}} F(\widehat{\mathbf{u}}) := \sum_{i=1}^{n} J_i(\widehat{\mathbf{u}}_i) + \sum_{i=1}^{n} \mathbf{I}_{\mathcal{P}_i}(\widehat{\mathbf{u}}_i) + \mathbf{I}_{\mathcal{A}}(\widehat{\mathbf{u}}), \tag{17}$$

where $F : \mathbb{R}^{\ell} \to \mathbb{R} \cup \{+\infty\}$ denotes the extended-valued objective function. Thus F is the sum of two indictor functions of closed convex sets and the convex quadratic function given by $J(\widehat{\mathbf{u}}) := \sum_{i=1}^{n} J_i(\widehat{\mathbf{u}}_i)$, by slightly abusing the notation. Note that $\mathcal{A}$ is polyhedral. It is easy to show via Corollary 3.1 that the Slater's condition holds under the mild assumptions given in Corollary 3.1, for example, $v_0(k) > v_{\min}$ for all $k \in \mathbb{Z}_+$. Hence, by Rockafellar (1970, corollary 23.8.1), $\partial F(\widehat{\mathbf{u}}) = \sum_{i=1}^{n}(\nabla J_i(\widehat{\mathbf{u}}_i) + \mathcal{N}_{\mathcal{P}_i}(\widehat{\mathbf{u}}_i)) + \mathcal{N}_{\mathcal{A}}(\widehat{\mathbf{u}})$ in light of $\partial \mathbf{I}_C(x) = \mathcal{N}_C(x)$, where $\mathcal{N}_C(x)$ denotes the normal cone of a closed convex set C at $x \in C$. Finally, the formulation given by (17) is a locally coupled convex optimization problem (see Section 4.1). This formulation allows one to develop fully distributed schemes. Particularly, in fully distributed computation, each vehicle i only knows $\widehat{\mathbf{u}}_i$ and $\widehat{J}_i$ (i.e., J_i and $\mathcal{P}_i$) but does not know $\widehat{\mathbf{u}}_j$ and $\widehat{J}_j$ with $j \ne i$. Each vehicle i will exchange information with its neighboring vehicles to update $\widehat{\mathbf{u}}_i$ via a distributed scheme.

Algorithm 1 (Generalized Douglas-Rachford Splitting Method–Based Fully Distributed Algorithm)

1: Choose constants $0 < \alpha < 1$ and $\rho > 0$
2: Initialize $k = 0$, and choose an initial point z^0
3: **while** the stopping criteria is not met **do**
4: **for** $i = 1, \ldots, n$ **do**
5: Consensus step: Compute $\bar{z}_i^k$ using Equation (18), and let $w_i^{k+1} \leftarrow \bar{z}_i^k$
6: **end for**
7: **for** $i = 1, \ldots, n$ **do**
8: $z_i^{k+1} \leftarrow z_i^k + 2\alpha \cdot \left[\text{Prox}_{\rho \widehat{J}_i}(2w_i^{k+1} - z_i^k) - w_i^{k+1}\right]$
9: **end for**
10: $k \leftarrow k + 1$
11: **end while**
12: **return** $\widehat{\mathbf{u}}^* = w^k$

We introduce more notation first. For a proper, lower semicontinuous convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, let $\text{Prox}_f(\cdot)$ denote the proximal operator, that is, for any given $x \in \mathbb{R}^n$,

$$\text{Prox}_f(x) := \arg\min_{z \in \mathbb{R}^n} f(z) + \frac{1}{2}\|z - x\|_2^2 .$$

Furthermore, Π_C denotes the Euclidean projection onto a closed convex set C. Using this notation, we present a specific operator splitting method based distributed scheme for solving (17). By grouping the first two sums (with separable variables) in the objective function of (17), we apply the generalized Douglas-Rachford splitting algorithm (Hu, Xiao, and Liu 2018). Recall that $\widehat{J}_i(\widehat{\mathbf{u}}_i) := J_i(\widehat{\mathbf{u}}_i) + \mathbf{I}_{\mathcal{P}_i}(\widehat{\mathbf{u}}_i)$ for each $i = 1, \ldots, n$. For any constants α and ρ satisfying $0 < \alpha < 1$ and $\rho > 0$, this algorithm is given by

$$w^{k+1} = \Pi_{\mathcal{A}}(z^k),$$
$$z^{k+1} = z^k + 2\alpha \cdot \left[\text{Prox}_{\rho \widehat{J}_1 + \cdots + \rho \widehat{J}_n}(2w^{k+1} - z^k) - w^{k+1}\right].$$

It is shown in Davis and Yin (2017) and Hu, Xiao, and Liu (2018) that the sequence (w^k) converges to the unique minimizer $\widehat{\mathbf{u}}^*$ of the optimization problem (17). In the previous scheme, $\Pi_{\mathcal{A}}$ is the orthogonal projection onto the consensus subspace $\mathcal{A}$ such that the following holds: for any $\widehat{\mathbf{u}} := (\widehat{\mathbf{u}}_1, \ldots, \widehat{\mathbf{u}}_n)$ where $\widehat{\mathbf{u}}_i := (\mathbf{u}_i, (\mathbf{u}_{ij})_{j \in \mathcal{N}_i})$, $\overline{\mathbf{u}} := \Pi_{\mathcal{A}}(\widehat{\mathbf{u}})$ is given by (Hu, Xiao, and Liu 2018, section IV):

$$\overline{\mathbf{u}}_j = \overline{\mathbf{u}}_{ij} = \frac{1}{1 + |\mathcal{N}_j|}\left(\widehat{\mathbf{u}}_j + \sum_{k \in \mathcal{N}_j} \widehat{\mathbf{u}}_{kj}\right), \qquad \forall (i,j) \in \mathcal{E}. \tag{18}$$

Furthermore, because of the decoupled structure of $\widehat{J}_i$ s, we obtain the distributed version of the previous algorithm, which is also summarized in Algorithm 1:

$$w_i^{k+1} = \bar{z}_i^k, \qquad i = 1, \ldots, n; \tag{19a}$$

$$z_i^{k+1} = z_i^k + 2\alpha \cdot \left[\text{Prox}_{\rho \widehat{J}_i}(2w_i^{k+1} - z_i^k) - w_i^{k+1}\right], \qquad i = 1, \ldots, n. \tag{19b}$$

In the distributed scheme (19), the first step (or Line 5 of Algorithm 1) is a consensus step, and the consensus computation is carried out in a fully distributed and synchronous manner as indicated in Hu, Xiao, and Liu (2018, section IV). The second step in (19) (or Line 8 of Algorithm 1) does not need interagent communication (Hu, Xiao, and Liu 2018) and is performed using local computation only. For effective computation in the second step, recall that $\widehat{J}_i(\widehat{\mathbf{u}}_i) := J_i(\widehat{\mathbf{u}}_i) + \mathbf{I}_{\mathcal{P}_i}(\widehat{\mathbf{u}}_i)$ for each $i = 1, \ldots, n$ such that the proximal operator $\text{Prox}_{\rho \widehat{J}_i}(\widehat{\mathbf{u}}_i)$ becomes

$$\text{Prox}_{\rho \widehat{J}_i}(\widehat{\mathbf{u}}_i) = \arg\min_{z \in \mathcal{P}_i} J_i(z) + \frac{1}{2\rho}\|z - \widehat{\mathbf{u}}_i\|_2^2 .$$

Because $\mathcal{P}_i$ is the intersection of a polyhedral set and a quadratically constrained convex set and J_i is a quadratic convex function, $\text{Prox}_{\rho \widehat{J}_i}(\widehat{\mathbf{u}}_i)$ is formulated as a QCQP and can be solved via a second-order cone program (Alizadeh and Goldfarb 2003) or a semidefinite program. Efficient numerical packages, for example, SeDuMi (Sturm 1999), can be used for solving the QCQP with high accuracy. Last, a typical (global) stopping criterion in Scheme (19) (or Algorithm 1) is defined by the error tolerance of two neighboring z^ks, that is, $\|z^{k+1} - z^k\|_2 \le \varepsilon$, where $\varepsilon > 0$ is an error tolerance. For distributed computation, one can use its local version, that is, $\|z_i^{k+1} - z_i^k\|_2 \le \varepsilon/n$, as a stopping criterion for each vehicle.

Remark 4.2. Other distributed algorithms can be used to solve the underlying optimization problem (17). For example, the three operator splitting method–based schemes developed in Davis and Yin (2017) can be applied. To describe such schemes, let $\widehat{L} := \max_{i=1,\ldots,n} \|\widehat{W}^i\|_2 > 0$. The Hessian $HJ(\widehat{\mathbf{u}}) = \text{diag}(\widehat{W}^i)_{i=1,\ldots,n}$. Hence, ∇J is $\widehat{L}$-Lipschitz continuous and thus $1/\widehat{L}$-cocoercive. Furthermore, the two indicator functions are proper, closed, and convex functions. Choose the constants γ, λ such that $0 < \gamma < 2/\widehat{L}$ and $0 < \lambda < 2 - \frac{\gamma \widehat{L}}{2}$. Then for any initial condition z^0, the iterative scheme is given by (Davis and Yin 2017, algorithm 1):

$$w^{k+1} = \Pi_{\mathcal{A}}(z^k),$$
$$z^{k+1} = z^k + \lambda \cdot [\Pi_{\mathcal{P}_1 \times \cdots \times \mathcal{P}_n}(2w^{k+1} - z^k - \gamma \nabla J(w^{k+1})) - w^{k+1}].$$

In view of the similar discussions for consensus computation and decoupled structure of the projection $\Pi_{\mathcal{P}_1 \times \cdots \times \mathcal{P}_n}$, we obtain the distributed version of the previous algorithm:

$$w_i^{k+1} = \bar{z}_i^k, \qquad i = 1, \ldots, n;$$

$$z_i^{k+1} = z_i^k + \lambda \cdot [\Pi_{\mathcal{P}_i}(2w_i^{k+1} - z_i^k - \gamma \cdot [\widehat{W}^i w_i^{k+1} + c_{\mathcal{I}_i}]) - w_i^{k+1}], \qquad i = 1, \ldots, n.$$

In this scheme, the Euclidean projection $\Pi_{\mathcal{P}_i}$ can be formulated as a QCQP or a second-order cone program and solved via SeDuMi.

When each $\widehat{W}^i$ is PD, each J_i is strongly convex. Thus ∇J is μ-strongly monotone with $\mu = \min_{i=1,\dots,n} \lambda_{\min}(\widehat{W}^i)$, that is, $(x-y)^T(\nabla J(x) - \nabla J(y)) \geq \mu \|x-y\|_2^2$, $\forall x, y$. Because the subdifferential of the indicator function of a closed convex set is monotone, an accelerated scheme developed in Davis and Yin (2017, algorithm 2) can be exploited. In particular, let η be a constant with $0 < \eta < 1$, and $\gamma_0 \in (0, 2/(\widehat{L} \cdot (1-\eta))$. Set the initial points for an arbitrary z^0, $w^0 = \Pi_{\mathcal{A}}(z^0)$ and $v^0 = (z^0 - w^0)/\gamma_0$. The distributed version of this scheme is given by

$$
\begin{aligned}
&w_i^{k+1} = \overline{z}_i^k + \gamma_k \overline{v}_i^k, \qquad i = 1, \dots, n;\\
&v_i^{k+1} = \frac{1}{\gamma_k}(z_i^k + \gamma_k v_i^k - w_i^{k+1}), \qquad i = 1, \dots, n;\\
&\gamma_{k+1} = -\widetilde{\mu}\gamma_k^2 + \sqrt{(\widetilde{\mu}\gamma_k^2)^2 + \gamma_k^2};\\
&z_i^{k+1} = \Pi_{\mathcal{P}_i}(w_i^{k+1} - \gamma_{k+1} v_i^{k+1} - \gamma_{k+1}[\widehat{W}^i w_i^{k+1} + c_{\mathcal{I}_i}]),\\
&\qquad\qquad\qquad\qquad\qquad\qquad i = 1, \dots, n,
\end{aligned}
$$

where $\widetilde{\mu} := \eta \cdot \mu$. It is shown in Davis and Yin (2017, theorem 1.2) that (w^k) converges to the unique minimizer $\widehat{\mathbf{u}}^*$ with $\|w^k - \widehat{\mathbf{u}}^*\|_2 = O(1/(k+1))$. However, our numerical results show that Algorithm 1 outperforms the three operator splitting method–based schemes in term of real-time computation when $p \geq 3$ (see Section 6.2 for comparison and details).

Remark 4.3. The proposed MPC formulation, decomposition method, and fully distributed schemes can be extended to nonconstant spacing car following policies, for example, the time-headway spacing policy. This policy is given by $\Delta_i(k) = d_0 + v_i(k) \cdot h$, $\forall k \in \mathbb{Z}_+$ for the ith vehicle, where $d_0 > 0$ is a constant, and $h > 0$ is the constant time-headway. In this case, it can be shown that $z_i(k+1) = z_i(k) + \tau z_i'(k) + \frac{\tau^2}{2} w_i(k) - \tau h \cdot u_i(k)$ and $z_i'(k+1) = z_i'(k) + \tau w_i(k)$ for each i. Therefore, for each $s = 1, \dots, p$, $(z_i(k+s), z_i'(k+s))$ depends on $u_i(k)$, $\dots, u_i(k+s-1)$ and $u_{i-1}(k), \dots, u_{i-1}(k+s-1)$ only such that its MPC formulation also leads to a locally coupled QCQP, to which the proposed fully distributed schemes are applicable.

5. Control Design and Stability Analysis of the Closed Loop Dynamics

In this section, we discuss how to choose the weight matrices $Q_{z,s}$, $Q_{z',s}$ and $Q_{w,s}$ to achieve the desired closed loop performance, including stability and traffic transient dynamics. For the similar reasons given in Gong, Shen, and Du (2016, section 5), we focus on the constraint free case.

Under the linear vehicle dynamics, the closed-loop system is also a linear system. Specifically, the linear closed-loop dynamics are given by

$$
\begin{aligned}
z(k+1) &= z(k) + \tau z'(k) + \frac{\tau^2}{2} w(k),\\
z'(k+1) &= z'(k) + \tau w(k),
\end{aligned}
\tag{20}
$$

where $w(k)$ is a unique solution to an unconstrained optimization problem arising from the MPC and is a linear function of $z(k)$ and $z'(k)$ to be determined as follows.

Case (i): $p = 1$. In this case, we write $Q_{z,1}, Q_{z',1}, Q_{w,1}$ as $Q_z, Q_{z'}, Q_w$, respectively. Then the objective function becomes

$$
\begin{aligned}
J(w(k)) = \frac{1}{2}[&z^T(k+1) Q_z, z(k+1)\\
&+ (z'(k+1))^T Q_z, z'(k+1)] + \frac{\tau^2}{2} \widetilde{w}^T(k) Q_w \widetilde{w}(k),
\end{aligned}
$$

where we recall that $\widetilde{w}(k) = w(k) - u_0(k)\mathbf{e}_1$. It follows from the similar argument in Gong, Shen, and Du (2016, section 50 that the closed-loop system is given by the following linear system:

$$
\begin{bmatrix} z(k+1) \\ z'(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} I_n - \frac{\tau^2}{4}\widehat{W}Q_z & \tau I_n - \widehat{W}\left(\frac{\tau^3}{4}Q_z + \frac{\tau}{2}Q_{z'}\right) \\ -\frac{\tau}{2}\widehat{W}Q_z & I_n - \widehat{W}\left(\frac{\tau^2}{2}Q_z + Q_{z'}\right) \end{bmatrix}}_{A_c} \begin{bmatrix} z(k) \\ z'(k) \end{bmatrix}
$$

$$
+ \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n \end{bmatrix} \widehat{W} Q_w \mathbf{e}_1 \cdot u_0(k), \tag{21}
$$

where A_c is the closed loop dynamics matrix, and

$$
\widehat{W} := \left[\frac{\tau^2 Q_z}{4} + Q_{z'} + Q_w\right]^{-1}. \tag{22}
$$

The matrix A_c in (21) plays an important role in the closed loop stability and desired transient dynamical performance. Because $Q_z, Q_{z'}$ and Q_w are all diagonal and PSD (respectively, PD), we have $Q_z = \mathrm{diag}(\boldsymbol{\alpha})$, $Q_{z'} = \mathrm{diag}(\boldsymbol{\beta})$, and $Q_w = \mathrm{diag}(\boldsymbol{\zeta})$, where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}_+^n$ and $\boldsymbol{\zeta} \in \mathbb{R}_{++}^n$ with $\boldsymbol{\alpha} = (\alpha_i)_{i=1}^n$, $\boldsymbol{\beta} = (\beta_i)_{i=1}^n$, and $\boldsymbol{\zeta} = (\zeta_i)_{i=1}^n$. Hence, we write the matrix A_c as $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ to emphasize its dependence on these parameters. The following result asserts asymptotic stability of the linear closed-loop dynamics; its proof resembles that for Gong, Shen, and Du (2016, proposition 5.1) and is thus omitted.

Proposition 5.1. *Given any $\tau \in \mathbb{R}_{++}$ and any $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta} \in \mathbb{R}_{++}^n$, the matrix $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is Schur stable, that is, each eigenvalue $\mu \in \mathbb{C}$ of $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ satisfies $|\mu| < 1$.*

Moreover, for any eigenvalue μ_i of A_c, the following hold:

(1) *If μ_i is nonreal, then $|\mu_i|^2 = \frac{\zeta_i}{d_i}$, where $d_i := \frac{\alpha_i\tau^2}{4} + \beta_i + \zeta_i$;*

(2) *If μ_i is real, then $1 - \left(\frac{\alpha_i\tau^2}{2} + \beta_i\right)\frac{1}{d_i} < \mu_i < 1 - \frac{\alpha_i\tau^2}{4d_i}$.*

Case (ii): $p > 1$. Fix k. For a general $p \in \mathbb{N}$, let $\mathbf{w} := (w(k), \ldots, w(k+p-1)) \in \mathbb{R}^{np}$. Recall that for each $s = 1, \ldots, p$,

$$z(k+s) = z(k) + s\tau z'(k) + \tau^2 \sum_{j=0}^{s-1} \frac{2(s-j)-1}{2} w(k+j),$$

$$z'(k+s) = z'(k) + \tau \sum_{j=0}^{s-1} w(k+j),$$

and with slightly abusing notation, the objective function is

$$J(\underbrace{w(k), \ldots, w(k+p-1)}_{\mathbf{w}}) = \frac{1}{2}\sum_{s=1}^{p}\Big(\tau^2 \widetilde{w}^T(k+s-1) Q_{w,s} \widetilde{w}(k+s-1) + z^T(k+s) Q_{z,s} z(k+s) + (z'(k+s))^T Q_{z',s} z'(k+s)\Big),$$

where $\widetilde{w}(k+s) := w(k+s) - u_0(k) \cdot \mathbf{e}_1$ introduced in Remark 3.1. It follows from the similar development in Section 3.1 that

$$J(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T \mathbf{H} \mathbf{w} + \mathbf{w}^T\left(\mathbf{G}\begin{bmatrix} z(k) \\ z'(k) \end{bmatrix} - u_0(k)\mathbf{g}\right) + \widetilde{\gamma},$$

where $\widetilde{\gamma}$ is a constant. By a similar argument as in Lemma 3.1, it can be shown that $\mathbf{H} \in \mathbb{R}^{pn \times pn}$ is a symmetric PD matrix. Furthermore, it resembles the matrix V in (8) (by replacing S_n^{-1} with I_n), that is,

$$\mathbf{H} = \begin{bmatrix} \breve{\mathbf{H}}_{1,1} + \tau^2 Q_{w,1} & \breve{\mathbf{H}}_{1,2} & \breve{\mathbf{H}}_{1,3} & \cdots & \cdots & \breve{\mathbf{H}}_{1,p} \\ \breve{\mathbf{H}}_{2,1} & \breve{\mathbf{H}}_{2,2} + \tau^2 Q_{w,2} & \breve{\mathbf{H}}_{2,3} & \cdots & \cdots & \breve{\mathbf{H}}_{2,p} \\ \cdots & & \cdots & & \cdots & \\ \cdots & & \cdots & & \cdots & \\ \breve{\mathbf{H}}_{p,1} & \breve{\mathbf{H}}_{p,2} & \breve{\mathbf{H}}_{p,3} & \cdots & \cdots & \breve{\mathbf{H}}_{p,p} + \tau^2 Q_{w,p} \end{bmatrix} \in \mathbb{R}^{np \times np}, \tag{23}$$

where $\breve{\mathbf{H}}_{i,j}$ s are diagonal PD matrices given by

$$\breve{\mathbf{H}}_{i,j} := \sum_{s=\max(i,j)}^{p}\left(\frac{\tau^4}{4}[2(s-i)+1]\cdot[2(s-j)+1] Q_{z,s} + \tau^2 Q_{z',s}\right) \in \mathbb{R}^{n \times n},$$

Moreover, it follows from (10) and (11) that the matrix $\mathbf{G}$ and constant vector $\mathbf{g}$ are

$$\mathbf{G} := \begin{bmatrix} \mathbf{G}_{1,1} & \mathbf{G}_{1,2} \\ \vdots & \vdots \\ \mathbf{G}_{p,1} & \mathbf{G}_{p,2} \end{bmatrix} \in \mathbb{R}^{pn \times 2n}, \qquad \mathbf{g} := \tau^2 \begin{bmatrix} Q_{w,1}\mathbf{e}_1 \\ \vdots \\ Q_{w,p}\mathbf{e}_1 \end{bmatrix} \in \mathbb{R}^{pn}.$$

where $\mathbf{G}_{i,1}, \mathbf{G}_{i,2} \in \mathbb{R}^{n \times n}$ are given as follows: for each $i = 1, \ldots, p$,

$$\mathbf{G}_{i,1} = \tau^2 \sum_{s=i}^{p} \frac{2(s-i)+1}{2} Q_{z,s},$$

$$\mathbf{G}_{i,2} = \tau^3 \sum_{s=i}^{p} s\frac{2(s-i)+1}{2} Q_{z,s} + \tau \sum_{s=i}^{p} Q_{z',s}.$$

Hence, the optimal solution is $\mathbf{w}_* = (w_*(k), w_*(k+1), \ldots, w_*(k+p-1)) = -\mathbf{H}^{-1}\left(\mathbf{G}\begin{bmatrix} z(k) \\ z'(k) \end{bmatrix} - u_0(k)\mathbf{g}\right)$, and $w_*(k) = -\begin{bmatrix} I_n & 0 & \cdots & 0 \end{bmatrix}\mathbf{H}^{-1}\left(\mathbf{G}\begin{bmatrix} z(k) \\ z'(k) \end{bmatrix} - u_0(k)\mathbf{g}\right)$. Define the matrix $\mathbf{K}$ and the vector $\mathbf{d}$ as

$$\mathbf{K} := -\begin{bmatrix} I_n & 0 & \cdots & 0 \end{bmatrix}\mathbf{H}^{-1}\mathbf{G} \in \mathbb{R}^{n \times 2n},$$
$$\mathbf{d} := \begin{bmatrix} I_n & 0 & \cdots & 0 \end{bmatrix}\mathbf{H}^{-1}\mathbf{g} \in \mathbb{R}^{n}. \tag{24}$$

The closed loop system becomes

$$\begin{bmatrix} z(k+1) \\ z'(k+1) \end{bmatrix} = \underbrace{\left\{\begin{bmatrix} I_n & \tau I_n \\ 0 & I_n \end{bmatrix} + \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n \end{bmatrix}\mathbf{K}\right\}}_{A_c}\begin{bmatrix} z(k) \\ z'(k) \end{bmatrix} + \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n \end{bmatrix} u_0(k) \cdot \mathbf{d}, \tag{25}$$

where A_c is the closed loop dynamics matrix, and the subscript of A_c represents the closed loop.

Because $Q_{z,s}, Q_{z',s}$ are diagonal PSD and $Q_{w,s}$ are diagonal PD for all $s = 1, \ldots, p$, we write them as $Q_{z,s} = \mathrm{diag}(\boldsymbol{\alpha}^s)$, $Q_{z',s} = \mathrm{diag}(\boldsymbol{\beta}^s)$, and $Q_{w,s} = \mathrm{diag}(\boldsymbol{\zeta}^s)$, where $\boldsymbol{\alpha}^s, \boldsymbol{\beta}^s \in \mathbb{R}^n_+$ and $\boldsymbol{\zeta}^s \in \mathbb{R}^n_{++}$ for all $s = 1, \ldots, p$ with $\boldsymbol{\alpha}^s = (\alpha_i^s)_{i=1}^n$, $\boldsymbol{\beta}^s = (\beta_i^s)_{i=1}^n$, and $\boldsymbol{\zeta}^s = (\zeta_i^s)_{i=1}^n$ for each s. Let $\boldsymbol{\alpha} := (\boldsymbol{\alpha}^1, \ldots, \boldsymbol{\alpha}^p)$, $\boldsymbol{\beta} := (\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^p)$, and $\boldsymbol{\zeta} := (\boldsymbol{\zeta}^1, \ldots, \boldsymbol{\zeta}^p)$. We write the matrix A_c as $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ to emphasize its dependence on these parameters.

It can be shown that there exists a permutation matrix $\widehat{E} \in \mathbb{R}^{2n \times 2n}$ such that $\widetilde{A} := \widehat{E}^T A_c \widehat{E}$ is a block diagonal matrix, that is, $\widetilde{A} = \mathrm{diag}(\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n)$ whose each block $\widetilde{A}_i \in \mathbb{R}^{2\times 2}$ is given by

$$\widetilde{A}_i = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} \frac{\tau^2}{2} \\ \tau \end{bmatrix}\widetilde{\mathbf{K}}_i, \qquad \forall i = 1, \ldots, n.$$

Here for each $i = 1, \ldots, n$, $\widetilde{\mathbf{K}}_i := -\mathbf{e}_1^T \widetilde{\mathbf{H}}_i^{-1} \widetilde{\mathbf{G}}_i \in \mathbb{R}^{1\times 2}$, where

$$\widetilde{\mathbf{H}}_i := \begin{bmatrix} (\breve{\mathbf{H}}_{1,1} + \tau^2 Q_{w,1})_{i,i} & (\breve{\mathbf{H}}_{1,2})_{i,i} & (\breve{\mathbf{H}}_{1,3})_{i,i} & \cdots & \cdots & (\breve{\mathbf{H}}_{1,p})_{i,i} \\ (\breve{\mathbf{H}}_{2,1})_{i,i} & (\breve{\mathbf{H}}_{2,2} + \tau^2 Q_{w,2})_{i,i} & (\breve{\mathbf{H}}_{2,3})_{i,i} & \cdots & \cdots & (\breve{\mathbf{H}}_{2,p})_{i,i} \\ \cdots & & \cdots & & \cdots & \\ \cdots & & \cdots & & \cdots & \\ (\breve{\mathbf{H}}_{p,1})_{i,i} & (\breve{\mathbf{H}}_{p,2})_{i,i} & (\breve{\mathbf{H}}_{p,3})_{i,i} & \cdots & \cdots & (\breve{\mathbf{H}}_{p,p} + \tau^2 Q_{w,p})_{i,i} \end{bmatrix} \in \mathbb{R}^{p \times p},$$

and

$$\widetilde{\mathbf{G}}_i := \begin{bmatrix} (\mathbf{G}_{1,1})_{i,i} & (\mathbf{G}_{1,2})_{i,i} \\ \vdots & \vdots \\ (\mathbf{G}_{p,1})_{i,i} & (\mathbf{G}_{p,2})_{i,i} \end{bmatrix} \in \mathbb{R}^{p\times 2}.$$

Note that $\widetilde{\mathbf{H}} = \operatorname{diag}(\widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \ldots, \widetilde{\mathbf{H}}_n) = E^T \mathbf{H} E$ for the permutation matrix $E \in \mathbb{R}^{pn\times pn}$ given by (9). Because $\mathbf{H}$ is PD, so are all the $\widetilde{\mathbf{H}}_i$ s.

As examples, we give the closed-form expressions of $\widetilde{\mathbf{H}}_i$ and $\widetilde{\mathbf{G}}_i$ for some small ps. When $p = 1$, $\widetilde{\mathbf{H}}_i = \tau^2\left(\frac{\tau^2}{4}\alpha_i^1 + \beta_i^1 + \zeta_i^1\right)$ and $\widetilde{\mathbf{G}}_i = \begin{bmatrix} \frac{\tau^2}{2}\alpha_i^1 & \frac{\tau^3}{2}\alpha_i^1 + \tau\beta_i^1 \end{bmatrix}$ for each $i = 1, \ldots, s$. When $p = 2$, we have, for each $i = 1, \ldots, n$,

$$\widetilde{\mathbf{H}}_i = \tau^2 \begin{bmatrix} \frac{\tau^2}{4}\alpha_i^1 + \frac{9\tau^2}{4}\alpha_i^2 + \beta_i^1 + \beta_i^2 + \zeta_i^1 & \frac{3\tau^2}{4}\alpha_i^2 + \beta_i^2 \\ \frac{3\tau^2}{4}\alpha_i^2 + \beta_i^2 & \frac{\tau^2}{4}\alpha_i^2 + \beta_i^2 + \zeta_i^2 \end{bmatrix} \in \mathbb{R}^{2\times 2},$$

and

$$\widetilde{\mathbf{G}}_i = \begin{bmatrix} \tau^2\frac{\alpha_i^1 + 3\alpha_i^2}{2} & \frac{\tau^3}{2}\alpha_i^1 + 3\tau^3\alpha_i^2 + \tau(\beta_i^1 + \beta_i^2) \\ \frac{\tau^2}{2}\alpha_i^2 & \tau^3\alpha_i^2 + \tau\beta_i^2 \end{bmatrix} \in \mathbb{R}^{2\times 2}.$$

Lemma 5.1. *Let $p = 2$. For any $\tau > 0$, $(\alpha_i^1, \beta_i^1, \zeta_i^1) > 0$ and $0 \neq (\alpha_i^2, \beta_i^2, \zeta_i^2) \geq 0$ for each $i = 1, \ldots, n$, $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is Schur stable, that is, its spectral radius is strictly less than one.*

Proof. By the previous argument, it suffices to show that each $\widetilde{A}_i$ is Schur stable for $i = 1, \ldots, n$. Fix an arbitrary i. Letting $\widetilde{\mathbf{K}}_i = \begin{bmatrix} c_1 & c_2 \end{bmatrix}$, we have

$$\widetilde{A}_i = \begin{bmatrix} 1 + \frac{\tau^2}{2}c_1 & \tau + \frac{\tau^2}{2}c_2 \\ \tau c_1 & 1 + \tau c_2 \end{bmatrix},$$

where

$$c_1 = -\frac{d_2\alpha_i^1 + \alpha_i^2(2\beta_i^2 + 3\zeta_i^2)}{2d'},$$

$$c_2 = -\frac{d_2\left(\frac{\tau^2}{2}\alpha_i^1 + \beta_i^1\right) + \frac{3\tau^2}{2}\alpha_i^2\beta_i^2 + \zeta_i^2(3\tau^2\alpha_i^2 + \beta_i^2)}{\tau d'},$$

and $d' := \det(\widetilde{\mathbf{H}}_i)/\tau^4$, and $d_s = \frac{\tau^2}{4}\alpha_i^s + \beta_i^s + \zeta_i^s$ for $s = 1, 2$. Hence, $d' = d_1 d_2 + \tau^2\alpha_i^2(\beta_i^2 + \frac{9}{4}\zeta_i^2) + \beta_i^2\zeta_i^2$. Define

$$\alpha' := d_2\alpha_i^1 + \alpha_i^2(2\beta_i^2 + 3\zeta_i^2),$$

$$\beta' := d_2\beta_i^1 + \frac{\tau^2}{2}\alpha_i^2\beta_i^2 + \zeta_i^2\left(\frac{3}{2}\tau^2\alpha_i^2 + \beta_i^2\right), \quad \gamma' := d_2\zeta_i^1.$$

Clearly, α', β', γ' are all positive for any $\tau > 0$, $(\alpha_i^1, \beta_i^1, \zeta_i^1) > 0$ and $0 \neq (\alpha_i^2, \beta_i^2, \zeta_i^2) \geq 0$. Moreover, we deduce from a somewhat lengthy but straightforward calculation that $d' = \frac{\tau^2}{4}\alpha' + \beta' + \gamma' > 0$. Hence, $c_1 = -\frac{\alpha'}{2d'}$, $c_2 = -\frac{\tau^2\alpha'/2 + \beta'}{\tau d'}$, and $\widetilde{A}_i = \begin{bmatrix} 1 - \frac{\alpha'\tau^2}{4d'} & \tau\left(1 - \left(\frac{\alpha'\tau^2}{4} + \frac{\beta'}{2}\right)\frac{1}{d'}\right) \\ -\frac{\alpha'\tau}{2d'} & 1 - \left(\frac{\alpha'\tau^2}{2} + \beta'\right)\frac{1}{d'} \end{bmatrix} \in \mathbb{R}^{2\times 2}$. It follows from Gong, Shen, and Du (2016, proposition 5.1) that $\widetilde{A}_i$ is Schur stable and so is $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$. □

Using a similar technique but more lengthy calculations, it can be shown that when $p = 3$, the matrix $A(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is Schur stable for $\tau > 0$, $(\alpha_i^1, \beta_i^1, \zeta_i^1) > 0$, $0 \neq (\alpha_i^2, \beta_i^2, \zeta_i^2) \geq 0$ and $0 \neq (\alpha_i^3, \beta_i^3, \zeta_i^3) \geq 0$ for each $i = 1, \ldots, n$. For $p > 4$, we expect that the same result holds (supported by numerical experience) although its proof becomes much more complicated. Nevertheless, it is observed that in the p-horizon MPC, when the parameters α_i^s, β_i^s (and possibly including ζ_i^s) with $s \geq 3$ are medium or large, large control inputs are generated, which causes control or speed saturation and may lead to undesired close-loop dynamics. Motivated by this observation, we obtain the following stability result for small $(\alpha_i^s, \beta_i^s) \geq 0$ for $s = 3, \ldots, p$.

Proposition 5.2. *Let $p \geq 3$. For any $\tau > 0$, $(\alpha_i^1, \beta_i^1, \zeta_i^1) > 0$ and $0 \neq (\alpha_i^2, \beta_i^2, \zeta_i^2) \geq 0$ for each $i = 1, \ldots, n$, and $\zeta_i^s > 0$ for $s = 3, \ldots, p$ and $i = 1, \ldots, n$, there exists a positive constant $\overline{\varepsilon}$ such that for any $\alpha_i^s, \beta_i^s \in [0, \overline{\varepsilon})$ for $s = 3, \ldots, p$ and $i = 1, \ldots, n$, $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is Schur stable.*

Proof. Consider $p \geq 3$. Fix arbitrary $\tau > 0$, $(\alpha_i^1, \beta_i^1, \zeta_i^1) > 0$, and $0 \neq (\alpha_i^2, \beta_i^2, \zeta_i^2) \geq 0$ for each $i = 1, \ldots, n$ and $\zeta_i^s > 0$ for $s = 3, \ldots, p$ and $i = 1, \ldots, n$. Suppose $\alpha_i^s = \beta_i^s = 0$ for all $s = 3, \ldots, p$ and $i = 1, \ldots, n$. Then $Q_{z,s} = Q_{z',s} = 0$ for all $s \geq 3$. Hence, $\mathbf{H}_{i,j} = 0$ for all $i \geq 3$ and any j. Thus, it is easy to show that for each $i = 1, \ldots, n$,

$$\widetilde{\mathbf{H}}_i = \begin{bmatrix} \widetilde{\mathbf{H}}_i^2 & & & \\ & \tau^2\zeta_i^3 & & \\ & & \ddots & \\ & & & \tau^2\zeta_i^p \end{bmatrix} \in \mathbb{R}^{p\times p}, \qquad \widetilde{\mathbf{G}}_i = \begin{bmatrix} \widetilde{\mathbf{G}}_i^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{p\times 2},$$

where $\widetilde{\mathbf{H}}_i^2 \in \mathbb{R}^{2\times 2}$ and $\widetilde{\mathbf{G}}_i^2 \in \mathbb{R}^{2\times 2}$ correspond to $p = 2$ given before. Hence, $\widetilde{\mathbf{K}}_i := -\mathbf{e}_1^T\widetilde{\mathbf{H}}_i^{-1}\widetilde{\mathbf{G}}_i = -\mathbf{e}_1^T(\widetilde{\mathbf{H}}_i^2)^{-1}\widetilde{\mathbf{G}}_i^2$. It follows from Lemma 5.1 that $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is Schur stable; that is, its spectral radius is strictly less than one. Because the spectral radius of $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is continuous in α_i^s, β_i^s for all $s = 3, \ldots, p$ and $i = 1, \ldots, n$, a small perturbation to α_i^s, β_i^s for all $s = 3, \ldots, p$ and $i = 1, \ldots, n$ still leads to the Schur stable matrix A_c. This yields the desired result. □

Based on the previous results, one may choose $Q_{z,s}, Q_{z',s}, Q_{w,s}$ in the following way. Let $u_s, v_s \in \mathbb{R}_+^n$ and $w_s \in \mathbb{R}_{++}^n$ be positive or nonnegative vectors of the

same order. Let $\eta > 1$ (e.g., $\eta = 5$ or higher) be a constant and let $\kappa_z, \kappa_{z'}$ and κ_w be some positive constants. Then let

$$Q_{z,s} = \frac{\kappa_z}{(s-1)^\eta}\,\mathrm{diag}(u_s), \quad Q_{z',s} = \frac{\kappa_{z'}}{(s-1)^\eta}\,\mathrm{diag}(v_s),$$
$$Q_{w,s} = \frac{\kappa_w}{(s-1)^\eta}\,\mathrm{diag}(w_s), \qquad s = 2,\dots,p.$$

For a given MPC horizon $p \in \mathbb{N}$, suppose that the closed loop dynamic matrix $A_c(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is Schur stable. Because the acceleration of the leading vehicle is bounded, that is, $a_{\min} \le u_0(k) \le a_{\max}$ for all $k \in \mathbb{Z}_+$, the closed loop dynamics given by (21) or (25) is bounded-input-bounded-output stable. Particularly, if $u_0(k) \to 0$ as $k \to \infty$, then $(z(k), z'(k)) \to 0$ as $k \to \infty$.

6. Numerical Results

6.1. Numerical Experiments and Weight Matrices Design

We conduct numerical tests to evaluate the performance of the proposed fully distributed schemes and the CAV platooning control. In these tests, we consider a platoon of an uncontrolled leading vehicle labeled by the index 0 and 10 (i.e., $n = 10$) CAVs following the leading vehicle. The following physical parameters are used for the CAVs and their constraints throughout this section unless otherwise stated: the desired spacing $\Delta = 50m$, the vehicle length $L = 5m$, the sample time $\tau = 1$ seconds, the reaction time $r = \tau = 1$ seconds, the acceleration and deceleration limits $a_{\max} = 1.35\,\mathrm{m/s^2}$ and $a_{\min} = -8\,\mathrm{m/s^2}$, and the speed limits $v_{\max} = 27.78\,\mathrm{m/s}$ and $v_{\min} = 10\,\mathrm{m/s}$. The initial state of the platoon is $z(0) = z'(0) = 0$ and $v_i(0) = 25\,\mathrm{m/s}$ for all $i = 0, 1, \dots, n$. Furthermore, the vehicle communication network is given by the cyclic-like graph, that is, the bidirectional edges of the graph are $(1,2), (2,3), \dots, (n-1, n) \in \mathcal{E}$.

When $n = 10$, a particular choice of these weight matrices is given as follows: for $p = 1$,

$$\boldsymbol{\alpha}^1 = (38.85, 40.2, 41.55, 42.90, 44.25, 45.60, 46.95, 48.30, 49.65, 51.00) := \widetilde{\boldsymbol{\alpha}},$$
$$\boldsymbol{\beta}^1 = (130.61, 136.21, 141.82, 147.42, 153.03, 158.64, 164.24, 169.85, 175.46, 181.06) := \widetilde{\boldsymbol{\beta}},$$
$$\boldsymbol{\zeta}^1 = (62, 74, 90, 92, 106, 194, 298, 402, 454, 480) := \widetilde{\boldsymbol{\zeta}}.$$

For $p \ge 2$, we choose $\boldsymbol{\alpha}^1 = \widetilde{\boldsymbol{\alpha}} - \mathbf{1}$, $\boldsymbol{\beta}^1 = \widetilde{\boldsymbol{\beta}} - \mathbf{1}$, $\boldsymbol{\zeta}^1 = \widetilde{\boldsymbol{\zeta}} - \mathbf{1}$, and

$$\boldsymbol{\alpha}^s = \frac{0.0228}{(s-1)^4} \times \widetilde{\boldsymbol{\alpha}}, \quad \boldsymbol{\beta}^s = \frac{0.044}{(s-1)^4} \times \widetilde{\boldsymbol{\beta}},$$
$$\boldsymbol{\zeta}^s = \frac{0.0026}{(s-1)^4} \times \widetilde{\boldsymbol{\zeta}}, \quad s = 2,\dots,p.$$

The previous vectors $\boldsymbol{\alpha}^s, \boldsymbol{\beta}^s, \boldsymbol{\zeta}^s$ define the weight matrices $Q_{z,s}, Q_{z',s}, Q_{w,s}$ for $s = 1,\dots,5$, which further yield the closed loop dynamics matrix A_c; see the discussions after (24). It is shown that when these weights are used, the spectral radius of A_c is 0.8498 for $p = 1$ and 0.8376 for $p = 2,\dots,5$, respectively.

We discuss the choice of the MPC prediction horizon p based on numerical tests as follows. Our numerical experience shows that for $p > 1$, the weight matrices $Q_{z,1}, Q_{z',1}$ and $Q_{w,1}$ play a more important role for the closed loop dynamics. For fixed $Q_{z,1}, Q_{z',1}$ and $Q_{w,1}$ with the large penalties in $Q_{z,s}, Q_{z',s}$ and $Q_{w,s}$ for $s > 1$, the closed loop dynamics may be mildly improved but at the expense of undesired large control. Hence, we choose smaller penalties in $Q_{z,s}, Q_{z',s}$ and $Q_{w,s}$ for $s > 1$, which only lead to slightly better closed loop performance compared with the case of $p = 1$. Furthermore, when a large p is used, the underlying optimization problem has a larger size, resulting in longer computation time and slow convergence of the proposed distributed scheme. Besides, the current MPC model assumes that the future $u_0(k+s) = u_0(k)$ for all $s = 1,\dots,p-1$ at each k. This assumption is invalid when the true $u_0(k+s)$ is substantially different from $u_0(k)$, which implies that the prediction performance is poor for a large p. Hence, it is recommended that a smaller p be used, for example, $p \le 5$.

The following scenarios are used to evaluate the proposed CAV platooning control.

Scenario 1. The leading vehicle performs instantaneous deceleration/acceleration and then keeps a constant speed for a while. The goal of this scenario is to test if the platoon can maintain stable spacing and speed when the leading vehicle is subject to acceleration or deceleration disturbances. The motion profile of the leading vehicle is as follows: the leading vehicle decelerates from $k = 51$ seconds to $k = 54$ seconds with the deceleration $-2\,\mathrm{m/s^2}$, and maintains a constant speed till $k = 100$ seconds. After $k = 100$ seconds, it restores to its original speed $25\,\mathrm{m/s}$ with the acceleration $1\,\mathrm{m/s^2}$.

Scenario 2. The leading vehicle performs periodical acceleration/deceleration. The goal of this scenario is to test whether the proposed control scheme can reduce periodical spacing and speed fluctuation. The motion profile of the leading vehicle in this scenario is

Table 1. Parameters in Algorithm 1 for Different MPC Horizon ps

MPC horizon	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
α	0.95	0.95	0.95	0.8	0.8
ρ	0.3	0.3	0.3	0.1	0.1
Error tolerance	10^{-3}	2×10^{-3}	5×10^{-3}	7×10^{-3}	1.25×10^{-2}

Table 2. Scenario 1: Computation Time and Numerical Accuracy

	Computation time per CAV (s)		Relative numerical error	
MPC horizon	Mean	Variance	Mean	Variance
$p = 1$	0.0248	0.0017	3.4×10^{-4}	1.9×10^{-7}
$p = 2$	0.0603	0.0034	1.5×10^{-3}	2.6×10^{-6}
$p = 3$	0.1596	0.0764	3.2×10^{-3}	1.1×10^{-5}
$p = 4$	0.1528	0.1500	4.0×10^{-3}	1.7×10^{-5}
$p = 5$	0.2365	0.2830	6.6×10^{-3}	5.7×10^{-5}

as follows: the leading vehicle periodically changes its acceleration and deceleration from $k = 51$ seconds to $k = 100$ seconds with the period $T = 4$ seconds and acceleration/deceleration $\pm 1\,\mathrm{m/s^2}$. Then it maintains its original constant speed 25 m/s after $k = 100$ seconds.

Scenario 3. In this scenario, we aim to test the performance of the proposed control scheme in a real traffic environment, particularly when the leading vehicle undergoes traffic oscillations. We use real world trajectory data from an oscillating traffic flow to generate the leading vehicle's motion profile. Specifically, we consider Next Generation Simulation (NGSIM) data on eastbound I-80 in San Francisco Bay area in California. We use the data of position and speed of a real vehicle to generate its control input at each second and treat this vehicle as a leading vehicle. Because the maximum of acceleration of this vehicle is close to $2\,\mathrm{m/s^2}$, we choose $a_{\max} = 2\,\mathrm{m/s^2}$. All the other parameters or physical limits remain the same. The experiment setup of this scenario is: $z_i(0) = 0m$, $v_i(0) = 25\,\mathrm{m/s}$ for each i, and the time length is 45 seconds. To further test the proposed CAV platooning control in a more realistic traffic setting in Scenario 3, random noise is added to each CAV to simulate dynamical disturbances, model mismatch, signal noise, communication delay, and road condition perturbations. In particular, at each k, the random noise with the normal distribution $0.04 \times \mathcal{N}(0,1)$ is added to the first CAV, and the noise with the normal distribution $0.02 \times \mathcal{N}(0,1)$ is added to each of the rest of the CAVs. Here a larger noise is imposed to the first CAV since there are more noises and disturbances between the leading vehicle and the first CAV.

6.2. Performance of Fully Distributed Schemes and CAV Platooning Control

The generalized Douglas-Rachford splitting method based distributed algorithm (Algorithm 1) is tested. For each MPC horizon p, the parameters α, ρ, and the error tolerance for the stopping criteria in this algorithm are chosen to achieve desired numerical accuracy and efficiency; see the discussions after (19) for error tolerances and Table 1 for a list of these parameters and error tolerances. In particular, we choose a larger error tolerance for a larger p to meet the desired computation time requirement of one second per vehicle. For comparison, we also test the three operator splitting based distributed scheme and its accelerated version given in Remark 4.2, where we choose $\delta_i = \lambda_{\min}(\grave{W}^i)/2$, $\gamma = 1.9/\widehat{L}$ and $\lambda = 1.05$. Here $\widehat{L}$ is the Lipschitz constant defined in Remark 4.2. For the accelerated scheme, we let $\eta = 0.2$ and $\gamma_0 = 1.9/(0.8 \times \widehat{L})$.

6.2.1. Initial Guess Warm-Up. For a given p, the augmented locally coupled optimization problem (17) has nearly $3np$ scalar variables and $3np$ scalar constraints when the cyclic-like network topology is considered. These sizes can be even larger for other network topologies satisfying Assumption 1. Hence, when p is large, the underlying optimization problem is of large size, which may affect the numerical performance of the distributed schemes. Several techniques are developed to improve the efficiency of the proposed Douglas-Rachford distributed schemes for real-time computation, particularly for a large p. For illustration, we discuss the initial guess warm-up technique as follows.

Table 3. Scenario 2: Computation Time and Numerical Accuracy

	Computation time per CAV (s)		Relative numerical error	
MPC horizon	Mean	Variance	Mean	Variance
$p = 1$	0.0464	0.0039	4.0×10^{-4}	1.9×10^{-7}
$p = 2$	0.1086	0.0153	1.1×10^{-3}	1.4×10^{-6}
$p = 3$	0.3296	0.2593	3.2×10^{-3}	1.13×10^{-5}
$p = 4$	0.5049	0.6257	5.9×10^{-3}	4.6×10^{-5}
$p = 5$	0.5784	0.7981	1.13×10^{-2}	1.3×10^{-5}

Table 4. Scenario 3: Computation Time and Numerical Accuracy

	Computation time per CAV (s)		Relative numerical error	
MPC horizon	Mean	Variance	Mean	Variance
$p = 1$	0.0825	0.0023	1.30×10^{-3}	3.5×10^{-6}
$p = 2$	0.2011	0.0051	7.5×10^{-3}	1.6×10^{-4}
$p = 3$	0.5830	0.3462	1.20×10^{-2}	4.2×10^{-4}
$p = 4$	0.8904	0.4685	1.69×10^{-2}	3.3×10^{-4}
$p = 5$	0.9967	0.7467	3.25×10^{-2}	1.3×10^{-4}

When implementing the proposed scheme, we often choose a numerical solution obtained from the last step as an initial guess for the current step and run the proposed Douglas-Rachford scheme. Such the choice of an initial guess usually works well when two neighboring control solutions are relatively close. However, it is observed that the convergence of the proposed distributed scheme is sensitive to an initial guess, especially when the CAV platoon is subject to significant traffic oscillations, which results in highly different control solutions between two neighboring instants. In this case, using a neighboring control solution as an initial guess leads to rather slow convergence. To solve this problem, we propose an initial guess warm-up technique, motivated by the fact that control solutions are usually unconstrained for most of k's. Specifically, we first compute an unconstrained solution in a fully distributed manner, which can be realized by setting $\mathcal{P}_i$ as the Euclidean space in Algorithm 1. This step can be efficiently computed because the proximal operator is formulated by an unconstrained quadratic program and has a closed form solution. In fact, letting $J_i(\widehat{\mathbf{u}}_i) = \frac{1}{2}\widehat{\mathbf{u}}_i^T \widehat{W}_i \widehat{\mathbf{u}}_i + c_{\mathcal{I}_i}^T \widehat{\mathbf{u}}_i$, the closed form solution to the proximal operator is given by $\mathrm{Prox}_{\rho J_i}(\widehat{\mathbf{u}}_i') = -(\rho \widehat{W}_i + I)^{-1}(\rho c_{\mathcal{I}_i} - \widehat{\mathbf{u}}_i')$, where $\widehat{W}_i$ is PD. We then project this unconstrained solution onto the constrained set in one step. Because of the decoupled structure of Problem (17), this one-step projection can be computed in a fully distributed manner. We thus use this projected solution as an initial guess for the Douglas-Rachford scheme. Numerical experience shows that this new initial guess significantly improves computation time and solution quality when p is large.

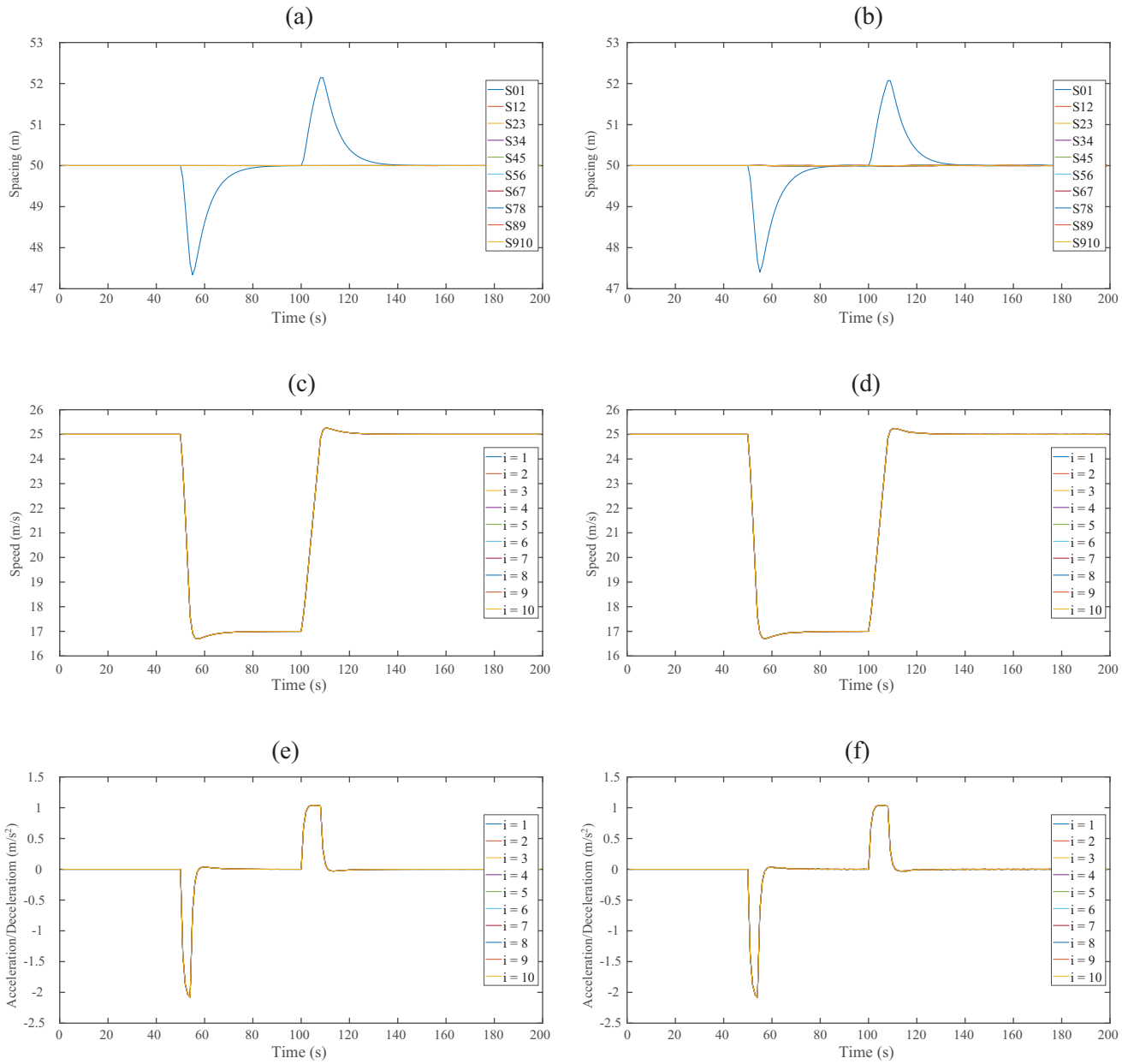
6.2.2. Performance of Distributed Schemes. Distributed algorithms are implemented on MATLAB and run on a computer of the following processor with four cores: Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz and RAM of 16.0 GB. We test the fully distributed Algorithm 1 for Scenarios 1–3. At each $k \in \mathbb{N}$, we use the optimal solution obtained from the last step as an initial guess unless otherwise stated. To evaluate the numerical accuracy of the proposed distributed scheme, we compute the relative error between the numerical solution from the distributed scheme and that from a high precision centralized scheme when the latter solution, labeled as the true solution, is nonzero. The mean and variance of computation time per vehicle and relative errors for different MPC horizon p's in noise-free Scenarios 1–3 are displayed in Tables 2–4, respectively. The numerical performance for Scenario 3 under noises is similar to that without noise and is thus omitted.

It is observed from the numerical results that when the MPC horizon p increases, more computation time is needed with mildly deteriorating numerical accuracy. This observation agrees with the discussion on the choice of p given in Section 6.1, which suggests a relatively small p for practical computation. Besides, we have tested the proposed initial guess warm-up technique on Scenario 3 for different ps using the same parameters and error tolerances for Algorithm 1 (Table 1).

Table 5. Scenario 3: Computation Time and Numerical Accuracy with Initial Guess Warm-up

	Computation time per CAV (s)		Relative numerical error	
MPC horizon	Mean	Variance	Mean	Variance
$p = 1$	0.0243	0.0023	5.0×10^{-4}	7.0×10^{-7}
$p = 2$	0.0097	0.0017	2.6×10^{-3}	1.6×10^{-5}
$p = 3$	0.0579	0.0253	2.2×10^{-3}	1.1×10^{-5}
$p = 4$	0.1063	0.1103	3.7×10^{-3}	2.4×10^{-5}
$p = 5$	0.1258	0.1155	8.5×10^{-3}	1.5×10^{-5}

Figure 1. (Color online) Scenario 1: Proposed CAV Platooning Control with $p = 1$ (Left) and $p = 5$ (Right)

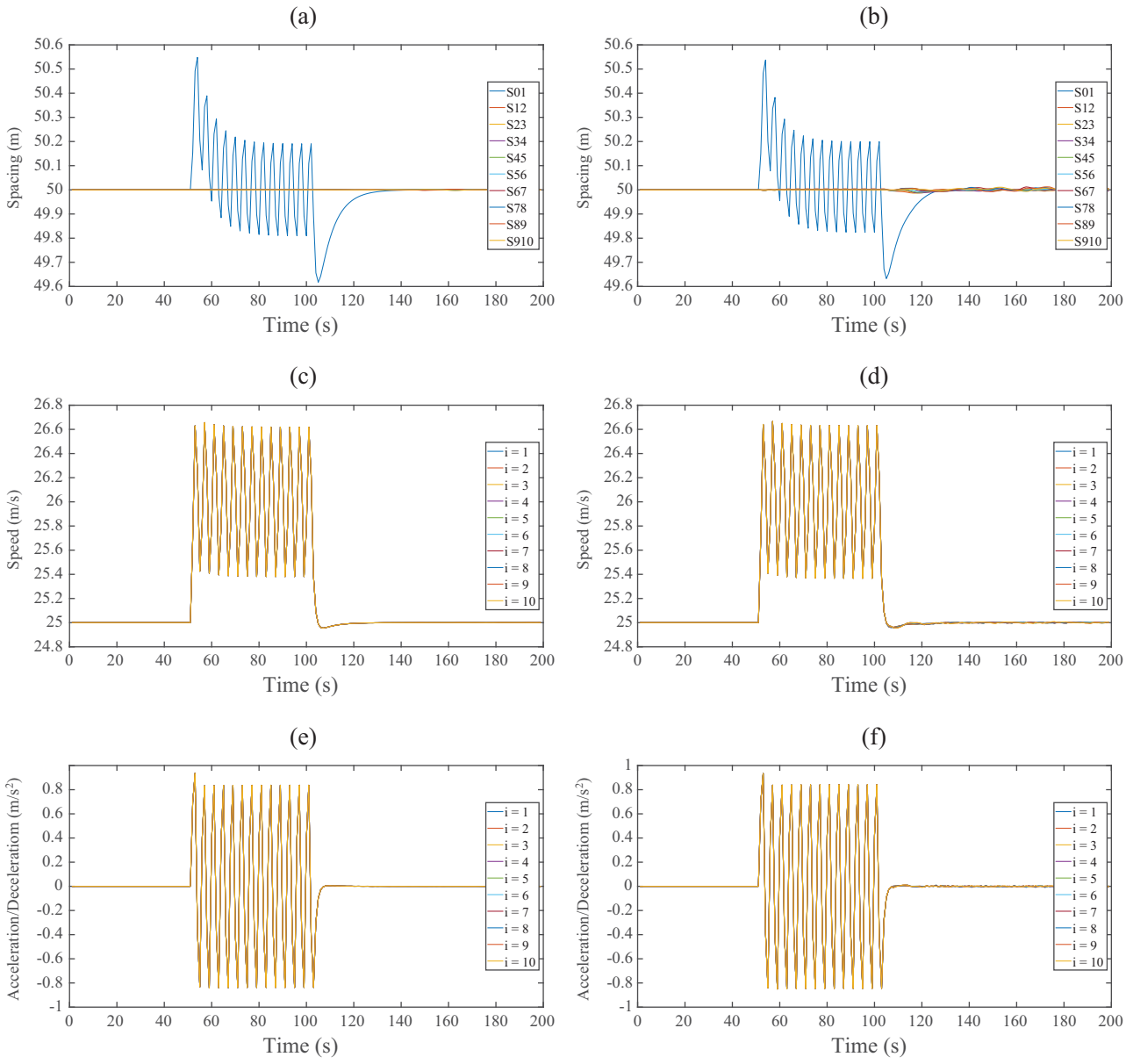


Notes. (a) Time history of spacing changes. (b) Time history of spacing changes. (c) Time history of vehicle speed. (d) Time history of vehicle speed. (e) Time history of control input. (f) Time history of control input.

To compute a warm-up initial guess using an iterative distributed scheme, we use the same α and ρ for each p with error tolerance 5×10^{-4} for $p = 1$ and 10^{-3} for the other ps. A summary of the numerical results is shown in Table 5. Compared with the results given in Table 4 without initial guess warm-up, the averaging computation time is reduced by at least 80% and the relative numerical error is reduced by at least two thirds for $p \geq 2$ when the initial guess warm-up is used. This shows that the initial guess warm-up technique considerably improves the numerical efficiency and accuracy, and it is especially suitable for real-time computation when a large p is used. Hence, we conclude that Algorithm 1, together with the initial guess warm-up technique, is suitable for real-time computation with satisfactory numerical precision.

We have also tested the three-operator splitting based distributed scheme and its accelerated version given in Remark 4.2. These schemes provide satisfactory computation time and numerical accuracy when p is small. For example, when $p = 1$, the mean of computation time per CAV is 0.0553 seconds with the

Figure 2. (Color online) Scenario 2: Proposed CAV Platooning Control with $p = 1$ (Left) and $p = 5$ (Right)

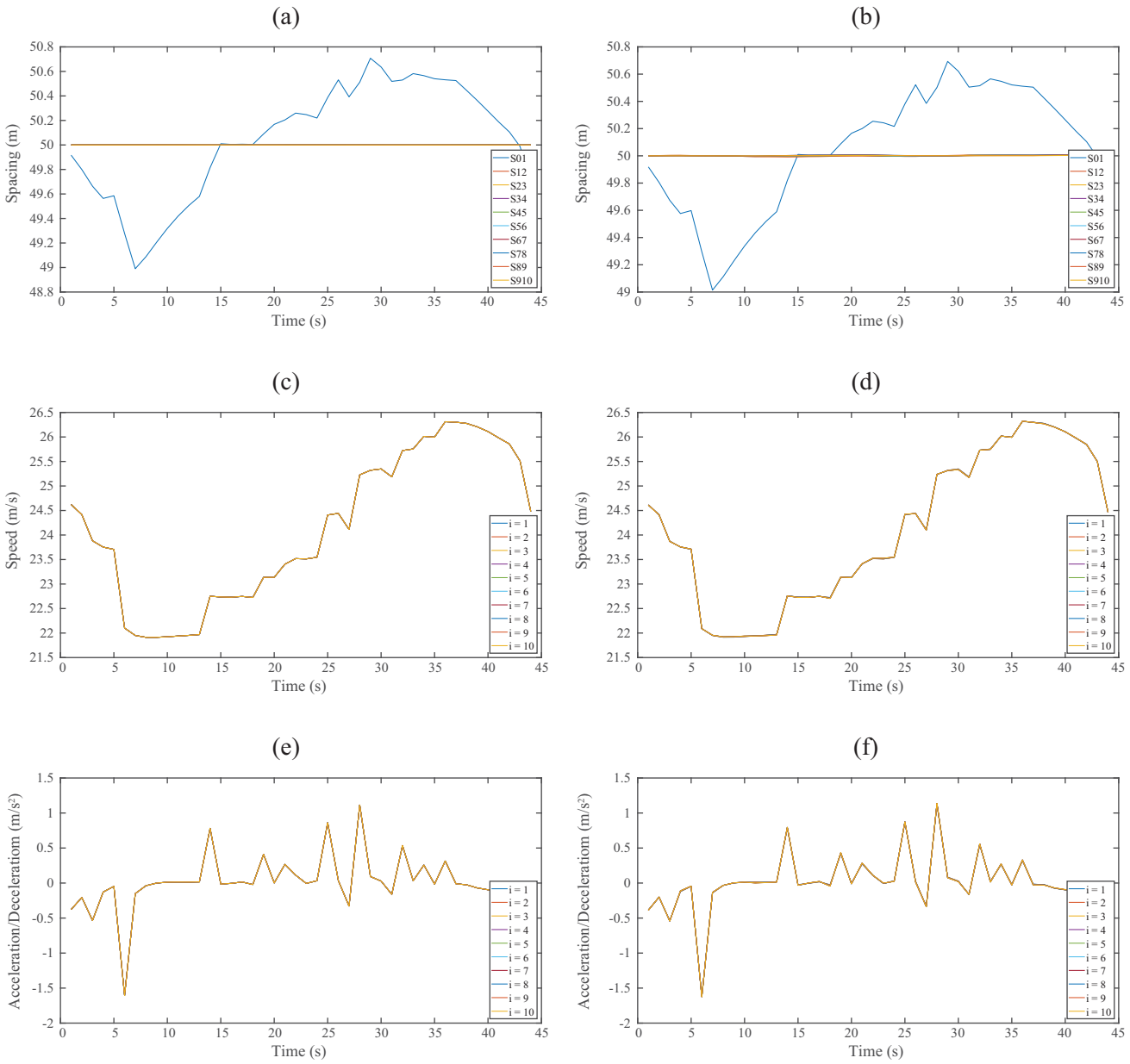


Notes. (a) Time history of spacing changes. (b) Time history of spacing changes. (c) Time history of vehicle speed. (d) Time history of vehicle speed. (e) Time history of control input. (f) Time history of control input.

variance 0.0284 for Scenario 1 and 0.219 seconds with the variance 0.138 for Scenario 2, respectively. However, for a slightly large p, for example, $p \geq 3$, it takes much longer than one second for an individual CAV to complete computation. This is because when $p \geq 3$, the Lipschitz constant $\widehat{L}$ is large, yielding a small constant γ and slow convergence. Hence, these schemes are not suitable for real-time computation when $p \geq 3$.

6.2.3. Performance of the CAV Platooning Control. We discuss the closed-loop performance of the proposed CAV platooning control for the three aforementioned scenarios with different MPC horizon ps. In each scenario, we evaluate the performance of the spacing between two neighboring vehicles (i.e., $S_{i-1,i}(k) := x_{i-1}(k) - x_i(k) = z_i(k) + \Delta$), the vehicle speed $v_i(k)$, and the control input $u_i(k)$, $i = 1, \ldots, n$ for $p = 1, 2, 3, 4, 5$. Because of the paper length limit, we present the closed-loop performance for $p = 1$ and $p = 5$ only for each scenario (see Figures 1–3 for (noise free) Scenarios 1–3 respectively, and Figure 4 for Scenario 3 with noises). In fact, it is observed from these figures (and the other tests) that

Figure 3. (Color online) Scenario 3: Proposed CAV Platooning Control with $p = 1$ (Left) and $p = 5$ (Right)



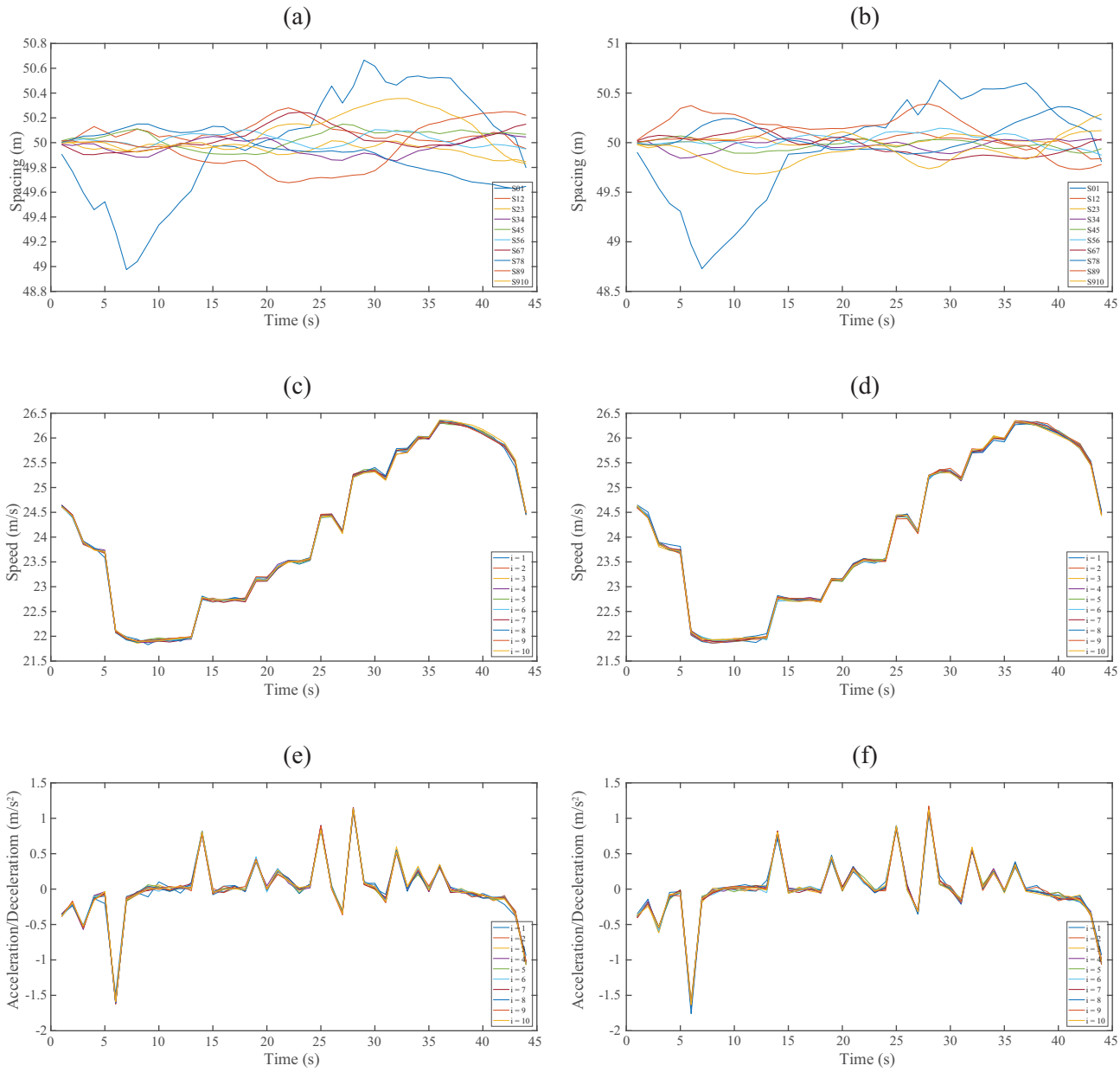
Notes. (a) Time history of spacing changes. (b) Time history of spacing changes. (c) Time history of vehicle speed. (d) Time history of vehicle speed. (e) Time history of control input. (f) Time history of control input.

there is little difference in control performance between $p = 1$ and a higher p, for example, $p = 5$. We comment more on the closed-loop performance of each scenario as follows.

(i) Scenario 1. Figure 1 shows that the spacing between the uncontrolled leading vehicle and the first CAV, that is, $S_{0,1}$, has mild deviation from the desired Δ when the leading vehicle performs instantaneous acceleration or deceleration, while the spacings between the other CAVs remain the desired constant Δ. Furthermore, it takes about 35 seconds for $S_{0,1}$ to converge to the steady state with the maximum spacing deviation 2.66 m. The similar performance can be observed for the vehicle speed and control input. In particular, it can be seen that all the CAVs show the exact same speed change and control, implying that the CAV platoon performs a "coordinated" motion with "consensus" under the proposed platooning control.

(ii) Scenario 2. Figure 2 displays that under the periodic acceleration/deceleration of the leading vehicle, the CAV platoon also demonstrates a "coordinated" motion

Figure 4. (Color online) Scenario 3 Under Noises: Proposed CAV Platooning Control with $p = 1$ (Left) and $p = 5$ (Right)



Notes. (a) Time history of spacing changes. (b) Time history of spacing changes. (c) Time history of vehicle speed. (d) Time history of vehicle speed. (e) Time history of control input. (f) Time history of control input.

with "consensus". For example, only $S_{0,1}$ demonstrates mild fluctuation, whereas the spacings between the other CAVs remain the desired constant, and all the CAVs show the exactly same speed change and control. Moreover, under the proposed platooning control, the oscillations of $S_{0,1}$ are relatively small with the maximal magnitude less than 0.22 m. Such oscillations quickly decay to zero within 30 seconds when the leading vehicle stops its periodical acceleration/deceleration.

(iii) Scenario 3. In this scenario, the leading vehicle undergoes various traffic oscillations through the time window of 45 seconds. Despite such oscillations, it is seen from Figure 3 that only $S_{0,1}$ demonstrates small spacing variations with the maximum magnitude less than 1 m, but the spacings between the other CAVs remain almost constant Δ through the entire time window. This shows that the CAV platoon also demonstrates a coordinated motion with consensus as in Scenarios 1 and 2.

(iv) Scenario 3 subject to noises. Figure 4 shows the control performance of the CAV platoon in Scenario 3 under noises. It can be seen that there are more

noticeable spacing deviations from the desired constant Δ for all CAVs because of the noises. However, the variation of $S_{0,1}$ remains to be within 1 m, and the maximum deviation of each $S_{i-1,i}$ with $i \geq 2$ is less than $0.5m$. Furthermore, the profiles of the CAV speed and control still demonstrate a nearly coordinated motion despite the noises.

In summary, the proposed platooning control effectively mitigates traffic oscillations of the spacing and vehicle speed of the platoon; it actually achieves a (nearly) consensus motion of the entire CAV platoon even under small random noises and perturbations. Compared with other platoon centered approaches (Gong, Shen, and Du 2016), the proposed control scheme performs better because it uses different weight matrices that lead to decoupled closed loop dynamics; this choice of the weight matrices also facilitates the development of fully distributed computation.

7. Conclusion

The present paper develops fully distributed optimization based MPC schemes for CAV platooning control under the linear vehicle dynamics. Such schemes do not require centralized data processing or computation and are thus applicable to a wide range of vehicle communication networks. New techniques are exploited to develop these schemes, including a new formulation of the MPC model, a decomposition method for a strongly convex quadratic objective function, formulating the underlying optimization problem as locally coupled optimization, and Douglas-Rachford method based distributed schemes. Control design and stability analysis of the closed loop dynamics is carried out for the new formulation of the MPC model. Numerical tests are conducted to illustrate the effectiveness of the proposed fully distributed schemes and CAV platooning control. Our future research will address nonlinear vehicle dynamics (Shen et al. 2022), closed loop stability analysis under nonconstant spacing car following polices, and various robust issues under uncertainties and disturbances, for example, communication network malfunction and failure, time delays in communication and control, model mismatch, and sensing errors.

Acknowledgments

The authors thank Benjamin Hyatt of University of Maryland Baltimore County for his contribution to the proof of closed loop stability when $p = 3$.

References

Alizadeh F, Goldfarb D (2003) Second-order cone programming. *Math. Programming* 95(1):3–51.

Barooah P, Mehta PG, Hespanha JP (2009) Mistuning-based control design to improve closed-loop stability margin of vehicular platoons. *IEEE Trans. Automated Control* 54(9):2100–2113.

Bergenhem C, Shladover S, Coelingh E, Englund C, Tsugawa S (2012) Overview of platooning systems. *Proc. 19th ITS World Congress* (Vienna, Austria), 1–7.

Davis D, Yin W (2017) A three-operator splitting scheme and its optimization applications. *Set-Valued Variance Anal.* 25(4):829–858.

Gong S, Du L (2018) Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles. *Transportation Res. Part B: Methodological* 116:25–61.

Gong S, Shen J, Du L (2016) Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon. *Transportation Res. Part B: Methodological* 94:314–334.

Hu J, Xiao Y, Liu J (2018) Distributed algorithms for solving locally coupled optimization problems on agent networks. *Proc. IEEE Conf. Decision Control* (IEEE, New York), 2420–2425.

Kavathekar P, Chen Y (2011) Vehicle platooning: A brief survey and categorization. *Proc. ASME Internat. Design Engrg. Tech. Conf. Comput. Inform. Engrg. Conf.* (American Society of Mechanical Engineers), 829–845.

Kesting A, Treiber M, Schönhof M, Helbing D (2008) Adaptive cruise control design for active congestion avoidance. *Transportation Res. Part C Emerging Tech.* 16(6):668–683.

Koshal J, Nedic A, Shanbhag UV (2011) Multiuser optimization: Distributed algorithms and error analysis. *SIAM J. Optim.* 21(3): 1046–1081.

Li S, Li K, Rajamani R, Wang J (2011) Model predictive multiobjective vehicular adaptive cruise control. *IEEE Trans. Control Systems Tech.* 19(3):556–566.

Marsden G, McDonald M, Brackstone M (2001) Toward an understanding of adaptive cruise control. *Transportation Res. Part C: Emerging Tech.* 9(1):33–51.

Mesbahi M, Egerstedt M (2010) *Graph Theoretic Methods in Multiagent Networks* (Princeton University Press, Princeton, NJ).

Rockafellar RT (1970) *Convex Analysis* (Princeton University Press, Princeton, NJ).

Shen J, Kammara EKH, Du L (2022) Nonconvex, fully distributed optimization based CAV platooning control under nonlinear vehicle dynamics. Preprint, submitted to IEEE Transactions on Intelligent Transportation Systems, revision under review, 2022, https://arxiv.org/abs/2104.08713.

Shladover SE, Su D, Lu X-Y (2012) Impacts of cooperative adaptive cruise control on freeway traffic flow. *Transportation Res. Record J. Transportation Res. Board* 2324:63–70.

Shladover SE, Nowakowski C, Lu X-Y, Ferlis R (2015) Cooperative adaptive cruise control: definitions and operating concepts. *Transportation Res. Record J. Transportation Res. Board* 2489:145–152.

Sturm JF (1999) Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Software* 11(1-4):625–653.

Van Arem B, Van Driel CJ, Visser R (2006) The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Trans. Intelligent Transportation Systems* 7(4):429–436.

Vander Werf J, Shladover SE, Miller MA, Kourjanskaia N (2002) Effects of adaptive cruise control systems on highway traffic flow capacity. *Transportation Res. Record* 1800(1):78–84.

Wang J, Gong S, Peeta S, Lu L (2019) A real-time deployable model predictive control-based cooperative platooning approach for connected and autonomous vehicles. *Transportation Res. Part B: Methodological* 128:271–301.

Wang M, Daamen W, Hoogendoorn SP, van Arem B (2014) Rolling horizon control framework for driver assistance systems. Part II: Cooperative sensing and cooperative control. *Transportation Res. Part C: Emerging Tech.* 40:290–311.

Wang M, Daamen W, Hoogendoorn SP, van Arem B (2016) Cooperative car-following control: Distributed algorithm and impact on moving jam features. *IEEE Trans. Intelligent Transportation Systems* 17(5):1459–1471.

Zhao S, Zhang K (2020) A distributionally robust stochastic optimization-based model predictive control with distributionally robust chance constraints for cooperative adaptive cruise control under uncertain traffic conditions. *Transportation Res. Part B: Methodological* 138:144–178.

Zheng Y, Li SE, Li K, Borrelli F, Hedrick JK (2017) Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies. *IEEE Trans. Control Systems Tech.* 25(3):899–910.

Zhou Y, Wang M, Ahn S (2019) Distributed model predictive control approach for cooperative car-following with guaranteed local and string stability. *Transportation Res. Part B: Methodological* 128:69–86.

Zhou Y, Ahn S, Chitturi M, Noyce DA (2017) Rolling horizon stochastic optimal control strategy for ACC and CACC under uncertainty. *Transportation Res. Part C: Emerging Tech.* 83: 61–76.