UMBC

DISSERTATION APPROVAL SHEET

Title of Dissertation: Fully Distributed Algorithms for Densely Coupled Optimization Problems in Sparse Optimization and Transportation Applications

Name of Candidate: Eswar Kumar Hathibelagal Kammara

Doctor of Philosophy, 2021

Graduate Program: Applied Mathematics

Dissertation and Abstract Approved:

Jinglai Shen

Jinglai Shen

Professor

Department of Math and Statistics

10/18/2021 | 11:34:15 AM EDT

NOTE: *The Approval Sheet with the original signature must accompany the thesis or dissertation. No terminal punctuation is to be used.

ABSTRACT

Title of dissertation: Fully Distributed Algorithms for Densely Coupled Optimization Problems in Sparse Optimization and Transportation Applications

Eswar Kumar Hathibelagal Kammara, Doctor of Philosophy, 2021

Dissertation directed by: Professor Jinglai Shen
Department of Mathematics and Statistics

Distributed algorithms are gaining increasing attention with broad applications in different areas such as multi-agent network systems, big data, machine learning, and distributed control systems, among others. Most of the distributed optimization algorithms developed assume a separable structure for the underlying optimization problems, and certain coupled optimization problems are often solved via partially distributed schemes. In this thesis, we develop fully distributed algorithms for densely coupled optimization problems in two topics, namely, column partition based sparse optimization problems and transportation applications. Firstly, we develop two-stage, fully distributed algorithms for coupled sparse optimization problems including LASSO, BPDN and their extensions. The proposed algorithms are column partition based and rely on the solution properties, exact regularization, and dual formulation of the problems. The overall convergence of two-stage schemes is shown. Numerical tests demonstrate the effectiveness of the proposed schemes. Secondly, we develop fully distributed algorithms for model predictive control (MPC) based connected and autonomous vehicle (CAV) platooning control under linear and nonlinear vehicle dynamics. In the context of linear vehicle dynamics, the underlying optimization problem of the MPC is a densely coupled, convex quadratically constrained quadratic program (QCQP). A decomposition technique is developed to formulate the densely coupled QCQP as a locally coupled convex optimization problem. We then develop operator splitting method based schemes to solve this problem in a fully distributed manner. Particularly, to meet challenging real-time implementation requirements, a generalized Douglas-Rachford splitting method based distributed algorithm is proposed, along with initial state warm up techniques. Under nonlinear vehicle dynamics, the underlying problem is a densely coupled, nonconvex optimization problem. We develop a sequential convex programming based fully distributed optimization algorithms. Control and closed loop stability analysis are carried out for both linear and nonlinear vehicle dynamics. Numerical tests performed for possibly heterogeneous CAV platoons demonstrate the effectiveness of the proposed schemes.

Fully Distributed Algorithms for Densely Coupled Optimization Problems in Sparse Optimization and Transportation Applications

by

Eswar Kumar Hathibelagal Kammara

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, Baltimore County in partial fulfillment
of the requirements for the degree of Doctorate
of Philosophy in Applied Mathematics
2021

Advisory Committee:
Professor Jinglai Shen, Chair/Advisor
Associate Professor Lili Du
Professor Muddappa Seetharama Gowda
Professor Osman Guler
Professor Muruhan Rathinam

DEDICATION

To all the advisers like mine.

ACKNOWLEDGEMENTS

I would first, and most importantly, like to thank my adviser, Dr. Jinglai Shen. He has been an outstanding mentor, as well as a very kind and patient person. This thesis wouldn't have been possible if not for the patience he showed towards me. Secondly, I would like to thank my committee members, Dr. Lili Du, Dr. Muddappa Seetharama Gowda, Dr. Osman Guler, and Dr. Muruhan Rathinam, for their time spent reviewing my thesis, their helpful comments, and an enjoyable defense. Also, I would like to thank Ben Hyatt of University of Maryland Baltimore County for his contribution to the proof of the closed loop stability when $p = 3$ under linear vehicle dynamics. I would like to thank Arun Polala, and Abhishek Balakrishna for their invaluable help, and my jeediginjalu group (in no particular order) including Lahir, Venkat, Chandana, Gokul, Varma, Sneha, Ketan, Vikram, Nikhil and others. These people made my time here at UMBC, a great time.

Finally, I would like to thank my parents and my brother Naveen, who has been my support system throughout my life.

The research in this thesis was partially supported by the NSF grant CMMI-1902006.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGMENTS iii

LIST OF TABLES vii

LIST OF FIGURES viii

I. Introduction 1

1.1 Overview of Distributed Optimization Algorithms 2
1.2 Review of Distributed Algorithms for Coupled Optimization Problems . . . 5
1.2.1 Distributed Algorithms for Locally Coupled Convex Optimization . 6
1.2.2 Operator Splitting Methods based Distributed Schemes 7
1.3 Two Topics of Fully Distributed Algorithms for Densely Coupled Optimization Problems 10
1.3.1 Column Partition based Fully Distributed Algorithms for Coupled Convex Sparse Optimization 10
1.3.2 Fully Distributed Optimization based CAV Platooning Control in Transportation Applications 13
1.4 Summary of Research Contributions 15
1.5 Organization 17
1.6 Notation 18

II. Column Partition based Fully Distributed Algorithms for Coupled Convex Sparse Optimization Problems 19

2.1 Motivation and Introduction 19
2.2 Problem Formulation and Solution Properties 20
2.3 Overview of the Development of Column Partition based Distributed Algorithms 25

2.3.1 Illustration of Main Ideas via the Standard LASSO 25
2.3.2 Overview of Key Steps for General Problems 27
2.4 Exact Regularization 29
2.4.1 Exact Regularization of Convex Piecewise Affine Function based Optimization 30
2.4.2 Exact Regularization of Grouped BP Problem Arising From Group LASSO 38
2.5 Dual Problems: Formulations and Properties 41
2.5.1 Dual Problems: General Formulations 41
2.5.2 Applications to the ℓ_1-norm based Problems 52
2.5.3 Applications to Problems Associated with the Norm from Group LASSO 56
2.6 Development of Column Partition based Distributed Algorithms 58
2.6.1 Structure of Column Partition based Distributed Schemes 59
2.6.2 Column Partition based Distributed Schemes for the Standard LASSO-like Problem 62
2.6.3 Column Partition based Distributed Schemes for the Standard BDPN-like Problem 68
2.6.4 Column Partition based Distributed Schemes for the Fused LASSO-like and Fused BDPN-like Problems 69
2.6.5 LASSO-like, BPDN-like, and Regularized BP-like Problems with the Norm from the Group LASSO 72
2.7 Overall Convergence of the Two-stage Distributed Algorithms 74
2.8 Numerical Results 86
2.8.1 Numerical Results for the LASSO-like Problems 88
2.8.2 Numerical Results for the BPDN-like Problems 92
2.8.3 Discussions and Comparison 94
2.9 Summary 96

III. Fully Distributed Optimization based CAV Platooning Control under Linear Vehicle Dynamics 97

3.1 Introduction 97
3.2 Vehicle Dynamics, Constraints, and Communication Networks 100
3.3 Model Predictive Control for CAV Platooning Control 102
3.3.1 Constrained MPC Optimization Model 106
3.4 Operator Splitting Method based Fully Distributed Algorithms for Constrained Optimization in MPC 111
3.4.1 Decomposition of a Strongly Convex Quadratic Objective Function . 111
3.4.2 Operator Splitting Method based Fully Distributed Algorithms . . . 121
3.5 Control Design and Stability Analysis of the Closed Loop Dynamics 126
3.6 Numerical Results 135
3.6.1 Numerical Experiments and Weight Matrices Design 135
3.6.2 Performance of Fully Distributed Schemes and CAV Platooning Control 138
3.7 Summary 148

IV. Nonconvex, Fully Distributed Optimization based CAV Platooning Control under Nonlinear Vehicle Dynamics . . . 150

4.1 Introduction . . . 150
4.2 Vehicle Dynamics, Constraints, and Communication Topology . . . 153
4.3 Sequential Feasibility and Properties of Constraint Sets . . . 154
4.3.1 Sequential Feasibility . . . 154
4.3.2 Nonempty Interior of the Constraint Sets . . . 158
4.4 Model Predictive Control for CAV Platooning . . . 161
4.4.1 Constrained Optimization Model under the Nonlinear Vehicle Dynamics . . . 162
4.5 Fully Distributed Algorithms for Coupled Nonconvex MPC Optimization Problem . . . 167
4.5.1 Formulation of MPC Optimization Problem as Locally Coupled Optimization . . . 167
4.5.2 Sequential Convex Programming and Operator Splitting Method based Fully Distributed Algorithms for the MPC Optimization Problem . . . 170
4.5.3 Approximation of the Objective Function and Constraint Functions 180
4.6 Control Design and Stability Analysis of Closed Loop Dynamics . . . 185
4.6.1 Reformulation of the Closed Loop Dynamics as a Tracking System . 186
4.6.2 Local Input-to-state Stability of the Closed Loop System . . . 197
4.7 Numerical Results . . . 199
4.7.1 Numerical Experiment Setup and Weight Matrix Design . . . 199
4.7.2 Performance of the Proposed Fully Distributed Scheme . . . 201
4.7.3 Performance of CAV Platooning Control . . . 206
4.8 Summary . . . 210

V. Conclusions . . . 220

5.1 Column Partition based Distributed Algorithm for Coupled Convex Sparse Optimization Problems . . . 220
5.2 Fully Distributed Optimization based CAV Platooning Control . . . 221

LIST OF TABLES

3.1 Parameters in Algorithm 9 for different MPC horizon p's 138
3.2 Scenario 1: computation time and numerical accuracy 140
3.3 Scenario 2: computation time and numerical accuracy 140
3.4 Scenario 3: computation time and numerical accuracy 140
3.5 Scenario 3: computation time and numerical accuracy with initial guess warm-up 141

4.1 Physical parameters for homogeneous small-size and large-size CAV platoons 200
4.2 Physical parameters for a heterogeneous medium-size CAV platoon with $\Delta = 60m$ 200
4.3 Error tolerances for outer and inner loops at different MPC horizon p's . . . 203
4.4 Scenario 1: computation time per CAV (sec) 204
4.5 Scenario 2: computation time per CAV (sec) 205
4.6 Scenario 3: computation time per CAV (sec) 205
4.7 Relative numerical error for $p = 1$ 206
4.8 Maximum steady state error of spacing (m) 206

LIST OF FIGURES

2.1 The topology of the random graph 87
2.2 Convergence behaviors in stage one of standard LASSO. 89
2.3 Convergence behaviors in stage two of standard LASSO. 89

3.1 Scenario 1: the proposed CAV platooning control with $p = 1$ (left column) and $p = 5$ (right column). 143
3.2 Scenario 2: the proposed CAV platooning control with $p = 1$ (left column) and $p = 5$ (right column). 144
3.3 Scenario 3: the proposed CAV platooning control with $p = 1$ (left column) and $p = 5$ (right column). 145
3.4 Scenario 3 under noises: the proposed CAV platooning control with $p = 1$ (left column) and $p = 5$ (right column). 146

4.1 Scenario 1 for the homogeneous small-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column). 211
4.2 Scenario 1 for the heterogeneous medium-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column). 212
4.3 Scenario 1 for the homogeneous large-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column). 213
4.4 Scenario 2 for the homogeneous small-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column). 214
4.5 Scenario 2 for the heterogeneous medium-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column). 215
4.6 Scenario 2 for the homogeneous large-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column). 216
4.7 Scenario 3 for the homogeneous small-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column). 217
4.8 Scenario 3 for the heterogeneous medium-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column). 218
4.9 Scenario 3 for the homogeneous large-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column). 219

CHAPTER I

Introduction

The origin of distributed optimization algorithms dates back to the 80's or earlier [5], [79], [80]. They have garnered increasing attention in the last two decades due to their widespread applications in multi-agent network systems [39], [95], large scale optimization, big data [18], [89], machine learning [11], [56], [88], and distributed control systems [21], [22], among others. A distributed algorithm is carried out by a multi-agent system interconnected via a communication network. In most distributed algorithms, each agent (e.g., a computing device) uses only partial information of a central problem; it performs its own computation on a relatively small sub-problem in the parallel manner and exchanges its numerical results with neighboring agents at each step, in order to achieve a solution of the central problem. Distributed algorithms are advantageous over centralized schemes in many important applications. In particular, since agents require access to only partial information, it facilitates the use of computing devices with lower memory and storage. With the ability to perform parallel operations, these algorithms outperform both in regards to the speed and the maximum problem size that can be addressed over the centralized techniques [77]. Further, distributed algorithms respect data privacy of agents and accommodate communication delays more effectively.

1.1 Overview of Distributed Optimization Algorithms

Various distributed algorithms have been developed to solve a wide range of optimization problems in different settings. In what follows, we provide a brief overview of these algorithms in terms of problem structure, numerical methodologies, parallel features, and network topologies.

Separable vs. Coupled Problems: Most of the distributed algorithms deal with problems with separable structure. A representative problem of separable structure is the consensus optimization problem [15], [16], [54]. Specifically, consider a group of N agents, each with a local objective function f_i, for $i = 1, \ldots, N$, with the objective of minimizing their sum, i.e., $\min_{x \in \mathbb{R}^N} \sum_{i=1}^{N} f_i(x)$. To solve such problems, there is a class of distributed optimization techniques that are developed based on consensus optimization where a local variable $x_i \in \mathbb{R}^N$ is introduced at each of the agents, and an additional consensus constraint is imposed such that the problem is formulated as follows:

$$\min_{x_1, x_2, \ldots, x_N \in \mathbb{R}^N} f_1(x_1) + f_2(x_2) + \ldots + f_N(x_N) \quad \text{subject to } x_1 = x_2 = \cdots = x_N.$$

Define the consensus space $\mathcal{A} := \{\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^{N^2} \quad | \quad x_1 = x_2 = \cdots = x_N\}$. The above problem thus simplifies to finding a solution to $\min_{\mathbf{x}} f(\mathbf{x}) + \mathcal{I}_{\mathcal{A}}(\mathbf{x})$, where $f(\mathbf{x}) = \sum_{i=1}^{N} f_i(x_i)$, and $\mathcal{I}_{\mathcal{A}}(\cdot)$ denotes the indicator function on the set $\mathcal{A}$. Operator splitting method based schemes including forward-backward-forward splitting (FBFS) [78], forward-backward splitting (FBS) [58], Douglas-Rachford splitting (DRS) [43] can be used to solve such problems. We will discuss more on fully distributed implementation of DRS in a latter section.

Further, the inclusion of a constraint set $\mathcal{C}$ brings more challenges to the algorithmic development as the above problem now becomes: $\min_{\mathbf{x}} \ f(\mathbf{x}) + \mathcal{I}_{\mathcal{C}}(\mathbf{x}) + \mathcal{I}_{\mathcal{A}}(\mathbf{x})$, where $\mathcal{C}$ is a constraint set. A three operator splitting scheme developed in [15] can be used to solve this problem. Moreover, to implement this in a distributed manner, it is necessary that the underlying constraint set $\mathcal{C}$ also has a separable structure, i.e., $\mathcal{I}_{\mathcal{C}}(\mathbf{x}) = \sum_{i=1}^{N} \mathcal{I}_{\mathcal{C}_i}(x_i)$. This decomposition into separable structure becomes increasingly challenging when $\mathcal{C}$ is non-polyhedral or is defined by densely coupled nonlinear functions.

There is a different category of problems where the underlying objective function or constraints are coupled but yet need distributed algorithms to handle them effectively and efficiently. Examples include [8], [10], [57], where either the constraints or the objective functions of the underlying problems are coupled. More precisely, the paper [57] proposes a relaxation and successive distributed decomposition algorithm to solve constraint coupled distributed optimization problems via a relaxation of the primal problem and duality theory. In [10] the authors propose an alternating direction method of multipliers (ADMM) based distributed scheme to solve the unconstrained distributed LASSO problem. The paper [8] extends it to a polyhedral constrained distributed LASSO problem via proximal dual based ADMM.

First Order vs. Second Order Techniques: The techniques to solve the above mentioned optimization problems can be either first-order methods, including distributed gradient decent scheme [92], distributed proximal gradient schemes [42], [70], subgradient based methods [53], [54] and dual decomposition methods [76], or second-order methods like Newton type methods [28], [49], [87]. In order to apply Newton type methods, it is required that the underlying function is smooth or semismooth, whereas subgradient based first-order schemes can be used if the function is not differentiable. Moreover, many distributed algorithms developed in this area are first-order schemes because in the second-order methods, one needs to compute a large size Hessian matrix which can be computationally expensive. Despite this drawback, second-order methods are more effective in terms of the computation time and typically have faster convergence rates than that of the first-order schemes. This is because the Newton type methods have higher per iteration costs, but have a significantly smaller number of iterations when compared to the first-order schemes which results in faster computation. To the best of our knowledge, Newton type distributed schemes are not developed to solve constrained optimization problems yet, exceptions include consensus constrained problems, whereas there is rich literature of first-order distributed schemes to solve constrained optimization problems.

Partially Distributed vs. Fully Distributed Algorithms: Regarding parallel features, distributed algorithms can be of two types, namely, *partially* distributed and *fully* distributed. Partially distributed schemes are referred to as those schemes that either require all agents to exchange information with a central component for centralized data processing or perform centralized computation in at least one step [37], whereas fully distributed schemes do not require centralized data processing or carry out centralized computation through the entire schemes [8], [10]. The former type includes [21], [22], [59]. In particular distributed LASSO can be solved by a proximal based partial distributed scheme proposed in [59]. The algorithms developed in [10] and [8] to solve standard LASSO and polyhedral constrained LASSO respectively are fully distributed. In [37], the authors develop gradient-based distributed algorithms for an approximation of the multiuser problem where the underlying objective function and constraints are coupled. The proposed algorithm is regularized primal-dual based and requires the computation of the gradient of the Lagrangian in one of its steps. This gradient cannot be computed in a fully distributed manner and hence a centralized computation is performed which makes the proposed schemes partially distributed. Based on this development, a model predictive control (MPC) based connected autonomous vehicles (CAV) platooning is developed in [22] and implemented by partially distributed schemes. It is worth pointing out that for a coupled optimization problem, the development of a fully distributed algorithm is much more challenging than that of a partially distributed algorithm.

Communication Networks: Communication network topology is another important aspect in the implementation of distributed algorithms. The network topology is generally characterised by a graph, where nodes of the graph represent the N agents and edges represent the communication between them. This graph could be bidirectional [63], [64], [66], or unidirectional [41], [55], i.e., there is a one way communication between the neighboring agents in case of a unidirectional graph and a two way communication with that of a bidirectional graph. Other variants include time varying [55], [82] and static [22], [50] graph which is characterized based on the dynamic or static nature of the edges with respect to time.

1.2 Review of Distributed Algorithms for Coupled Optimization Problems

Since this thesis is focused on fully distributed algorithms for densely coupled optimization problems, we review certain existing techniques for solving coupled optimization problems using distributed algorithms, namely, distributed algorithms for locally coupled convex optimization problems, and the operator splitting techniques often used for developing first-order distributed schemes.

1.2.1 Distributed Algorithms for Locally Coupled Convex Optimization

Distributed algorithms are proposed for solving locally coupled convex optimization in [26]. Specifically, consider a multi-agent network of n agents whose communication is characterized by a connected and undirect graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ 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 $\{\mathcal{I}_1, \ldots, \mathcal{I}_n\}$ be a disjoint union of the index set $\{1, \ldots, N\}$. 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 $\widehat{x}_i := \big(x_{\mathcal{I}_i}, (x_{\mathcal{I}_j})_{j \in \mathcal{N}_i}\big) \in \mathbb{R}^{n_i}$. 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 since 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:

$$(P'): \qquad \min_{x \in \mathbb{R}^N} \sum_{i=1}^n \widehat{J}_i(\widehat{x}_i), \quad \text{subject to} \quad \widehat{x}_i \in \mathcal{C}_i, \quad \forall\, i = 1, \ldots, n, \tag{1.1}$$

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, the paper [26] formulates 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; details can be found in [26].

1.2.2 Operator Splitting Methods based Distributed Schemes

In this subsection, we review some well known and useful operator splitting schemes from [15], [26]; also see [3], [12] for the original reference on this topic. A mapping $S : \mathbb{R}^n \to \mathbb{R}^n$ is a nonexpansive operator if $\|Sx - Sy\| \le \|x - y\|$, $\forall x, y \in \mathbb{R}^n$, and a mapping $T : \mathbb{R}^n \to \mathbb{R}^n$ is α-averaged if $T = (1-\alpha)I + \alpha S$ for some nonexpansive operator S, and $\alpha \in (0,1)$. Provided that T has a fixed point, the iteration $x^{k+1} = Tx^k$ converges to a point of T. This property is exploited to design an algorithm which finds an optimal solution to the following problem:

$$\min_x f(x) + g(x), \tag{1.2}$$

where $f, g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are closed convex proper (CCP) functions. The first order optimality condition for the above problem (1.2) reduces to finding an x in the $\text{zer}(A+B) := \{y \,|\, 0 \in Ay + By\}$, with $Ax = \partial f(x)$ and $Bx = \partial g(x)$, where ∂f and ∂g are the subdifferentials of f and g respectively. Further, since f and g are CCP functions, their subdifferentials ∂f and ∂g are maximal monotone operators, and the resolvent of any maximal monotone operator ∂g is given by the proximal operator

$\mathrm{prox}_{\rho g}(x) := \arg\min_z \left(g(z) + \frac{1}{2\rho}\|z-x\|^2\right) \forall x \in \mathbb{R}^n$. Let $J_A := (I+\rho A)^{-1}$ denote the resolvent of A for some $\rho > 0$, it follows from the fact $2J_A - I$ is nonexpansive that J_A is (1/2)-averaged operator. Hence a solution $x \in \mathrm{zer}(A+B)$ can be obtained as $x = J_B(z)$, where z is a fixed point of the nonexpansive map $(2J_A - I)(2J_B - I)$. Such a fixed point z can be found by iterating using the α-averaged map $(1-\alpha)I + \alpha(2J_A - I)(2J_B - I)$ for $\alpha \in (0,1)$, which results in the generalized Douglas-Rachford (DR) algorithm [16]:

$$\begin{aligned} x^{k+1} &= J_B(z^k), \\ z^{k+1} &= z^k + 2\alpha \cdot \left[J_A(2x^{k+1} - z^k) - x^{k+1} \right]. \end{aligned}$$

Starting from any z^0, the sequence x^k generated by the above algorithm will converge to $x^* \in \mathrm{zer}(A+B)$. The technique developed above can be classified as a two operator splitting scheme.

The paper [15] extends the above results to three operator splitting schemes, where the objective is to find a solution to the following problem:

$$\min_x f(x) + g(x) + h(x), \tag{1.3}$$

where $f, g, h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are CCP functions, and h is Lipschitz differentiable. Similar to the earlier problem, the first order optimality condition for (1.3) reduces to finding an x in the $\mathrm{zer}(A+B+C)$, with $Ax = \partial f(x)$, $Bx = \partial g(x)$, and $Cx = \nabla h(x)$. Note that C is β - cocoercive [15], because h is Lipschitz differentiable. Hence problem (1.3) can be solved by iterating

$$z^{k+1} = (1-\lambda_k)z^k + \lambda_k T z^k, \tag{1.4}$$

where T is given by $T = I - J_{\gamma B} + J_{\gamma A} \circ \left(2J_{\gamma B} - I - \gamma C \circ J_{\gamma B}\right)$, z^0 is an arbitrary point, $\gamma \in (0, 2\beta)$, and $\lambda_k \in (0, (4\beta - \gamma)/2\beta)$, the iteration can be implemented as follows [15, Algorithm 1]:

$$
\begin{aligned}
x_B^k &= J_{\gamma B}(z^k), \\
x_A^k &= J_{\gamma A}(2x_B^k - z^k - \gamma C x_B^k), \\
z^{k+1} &= z^k + \lambda_k (x_A^k - x_B^k).
\end{aligned}
$$

The convergence of the above algorithm is guaranteed by [15, Theorem 1.1]. The convergence rates are summarized in [15, Section 1.2]. Precisely, when f is strongly convex, the sequence x_B^k converges to x^* with a rate $o(1/\sqrt[4]{k+1})$.

Additionally, when g or h is strongly convex, by choosing appropriate scalars γ_k, and λ_k, a change of variable $z^k = x_A^{k-1} + \gamma_{k-1} u_B^{k-1}$, the authors propose the following accelerated variant of the above algorithm [15, Algorithm 2], where the convergence rate can be improved to $O(1/(k+1))$:

$$
\begin{aligned}
x_B^k &= J_{\gamma_k B}(x_A^{k-1} + \gamma_{k-1} u_B^{k-1}), \\
u_B^k &= \frac{1}{\gamma_{k-1}}\big(x_A^{k-1} + \gamma_{k-1} u_B^{k-1} - x_B^k\big), \\
x_A^k &= J_{\gamma_k A}\big(x_B^k - \gamma_k u_B^k - \gamma_k C x_B^k\big).
\end{aligned}
$$

It is worth mentioning that both the three operator splitting method based scheme and its accelerated variant critically rely on certain parameters. A conservative choice of these parameters often lead to slow convergence. Especially, even though the accelerated variant has the fast theoretical convergence rate of $O(1/(k+1))$, it may yield slow convergence in practice due to the restriction on these parameters.

1.3 Two Topics of Fully Distributed Algorithms for Densely Coupled Optimization Problems

In this section, we give an introduction to the two topics in fully distributed algorithms treated in this thesis, namely, column partition based densely coupled sparse optimization problems, and model predictive control arising from CAV platooning control in transportation applications.

1.3.1 Column Partition based Fully Distributed Algorithms for Coupled Convex Sparse Optimization

We first consider the development and analysis of fully distributed algorithms for densely coupled convex sparse optimization problems.

1.3.1.1 Background and Motivation

Sparse modeling and approximation finds broad applications in numerous fields of contemporary interest, including signal and image processing, compressed sensing, machine learning, and high dimensional statistics and data analytics. Various efficient schemes have been proposed for convex or nonconvex sparse signal recovery [19], [67]. To motivate the work, consider the well-studied LASSO problem: $\min_{x\in\mathbb{R}^N} \ \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1$, where $A \in \mathbb{R}^{m\times N}$ is the measurement (or sensing) matrix, $b \in \mathbb{R}^m$ is the measurement (or sensing) vector, $\lambda > 0$ is the penalty parameter, and $x \in \mathbb{R}^N$ is the decision variable. In the setting of sparse recovery, N is much larger than m. Besides, the measurement matrix A usually satisfies certain uniform recovery conditions for recovery efficiency, e.g., the restricted isometry property [19]. As such, A is often a dense matrix, namely, (almost) all of its elements are nonzero. We aim to develop distributed algorithms to solve the LASSO and related problems, where each agent knows the vector b and a small subset of the columns of A. Specifically, let $\{\mathcal{I}_1, \ldots, \mathcal{I}_p\}$ be a disjoint union of $\{1, \ldots, N\}$ such that $\{A_{\bullet \mathcal{I}_i}\}_{i=1}^p$ forms a column partition of A. For each i, the ith agent has the knowledge of b and $A_{\bullet \mathcal{I}_i}$ but does not know $A_{\bullet \mathcal{I}_j}$ with $j \neq i$. By running the proposed distributed scheme, it is expected that each agent i attains the subvector of an optimal solution of the LASSO corresponding to the index set $\mathcal{I}_i$, i.e., $x^*_{\mathcal{I}_i}$, at the end of the scheme, where x^* denotes an optimal solution of the LASSO.

The distributed optimization task described above is inspired by the following two scenarios arising from big data and network systems, respectively. In the context of big data, a practitioner may deal with a *ultra-large* data set, e.g., N is extremely large, so that it would be impossible to store a vector $x \in \mathbb{R}^N$ in a single computing device, let alone the measurement matrix A. When m is relatively small compared with N, the proposed distributed schemes can be used where each device only needs to store the vector b and a small number of the columns of A. The second scenario arises from multi-agent network systems, where each agent is operated by a low cost computing device which has limited memory and computing capacities. Hence, even when N is moderately large, it would be impractical for the entire matrix A to be stored or computed on one of these devices. Therefore, the proposed distributed schemes can be exploited in this scenario. Besides, the proposed algorithms can be extended to other sub-matrix partitions (in both row and column) of A, even if m is large in the above scenarios.

1.3.1.2 Literature Review

Distributed or decentralized algorithms for the LASSO and related problems, e.g., fused LASSO, basis pursuit (BP), and basis pursuit denoising (BPDN), have been extensively studied, including ADMM schemes, (sub-)gradient methods, and operator splitting schemes, e.g., [26], [44], [50], [59], [70], [92]. Particularly, the paper [50] develops row and column partition based distributed ADMM (D-ADMM) schemes for the BP that are convergent over a bipartite graph. The row partitioned LASSO and column partitioned BPDN are formulated as separable convex optimization and solved via D-ADMM [51]. Consensus based distributed schemes are developed for the row partitioned LASSO-like problems [47], e.g., the consensus ADMM (C-ADMM). An inexact C-ADMM (IC-ADMM) is established for distributed computation of the row partitioned LASSO and column partitioned logistic regression [10]. A proximal dual consensus ADMM (PDC-ADMM) scheme is used for solving column partition based LASSO under separable polyhedral constraints [8]. A decentralized gradient decent scheme is proposed for the regularized BP using column partition [92]. Besides, distributed proximal gradient schemes, e.g., PG-EXTRA, are exploited to solve the row partitioned LASSO [42], [70]. Other relevant distributed schemes include [9], [24], [29], [59], just to name a few.

Most distributed schemes in the literature deal with row partition based LASSO and BPDN. Further, column partition based distributed LASSO schemes often require the knowledge of different column blocks of A, e.g., [59], and thus cannot be implemented in a fully parallel manner. Exceptions include distributed BP [50], [92] and distributed BPDN [51] using the dual approach, and dual consensus ADMM (DC-ADMM) [10] and PDC-ADMM [8] for the LASSO with possible polyhedral constraints. However, exact regularization of the BPDN used in [51] generally fails as shown in Section 2.4, and the DC-ADMM does not guarantee the convergence of the primal variables [10, Theorem 2]. In addition, it is hard for the PDC-ADMM to handle the fused LASSO with more coupling and the BPDN-like problems with non-polyhedral constraints.

1.3.2 Fully Distributed Optimization based CAV Platooning Control in Transportation Applications

We now consider the second topic of this work, i.e., fully distributed optimization based CAV platooning control under linear and nonlinear vehicle dynamics.

Inspired by the next generation smart transportation systems, connected and autonomous vehicle (CAV) technologies emerge and offer tremendous opportunities to reduce traffic congestion and improve road safety and traffic efficiency in all aspects, through innovative traffic flow control and operations. Among a variety of CAV technologies, vehicle platooning technology links a group of CAVs through cooperative acceleration or speed control. Different from many other CAV technologies that mainly focus on neighborhood traffic efficiency and individual vehicle's safety, the vehicle platooning technology focuses on system efficiency and safety. Specifically, by using the vehicle platooning technology, adjacent group members of a CAV platoon can travel safely at a higher speed with smaller spacing. This will increase lane capacity, improve traffic flow efficiency, and reduce congestion, emission, and fuel consumption [4], [33].

There is extensive literature on CAV platooning control. The widely studied approaches include adaptive cruise control (ACC) [34], [40], [46], [83], [98], cooperative adaptive cruise control (CACC) [71], [72], [81], [96], and platoon centered vehicle platooning control [21], [22], [84], [86]. The first two approaches intend to improve an individual vehicle's safety, mobility, and string stability rather than systematical performance of the entire platoon, even though enhanced system performance is validated by simulations or field experiments.

On the other hand, the recently developed platoon centered approach seeks to optimize the platoon's transient traffic dynamics for a smooth traffic flow and to achieve stability and other desired long-time dynamical behaviors. This approach can significantly improve system performance and efficiency of the entire platoon [22], [84]. Despite this advantage, the platoon centered CAV platooning approach often encounters large-scale optimization or optimal control problems that require efficient numerical solvers for real-time computation [84]. More precisely, in order to facilitate real-time implementation, it is expected that the distributed schemes have a computation time which is less than *one second*. It is important to note that the restriction of *one second* on computation time is critical because otherwise the algorithms are not suitable for real time implementation. Distributed optimization techniques provide a favorable solution for the high demanding platoon centered approach. Supported by portable computing capability of each vehicle and vehicle-to-vehicle (V2V) communication [85], distributed computation can handle high computation load efficiently, is more flexible to communication network topologies, and is more robust to communication delays or network malfunctions [48], [85].

In spite of 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, i.e., 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 [2], [97], [99]. Besides, the platoon centered approach focuses on closed loop stability of the entire platoon instead of stability of individual vehicles and their interactions, e.g., string stability [99].

1.4 Summary of Research Contributions

We summarize the major results and contributions made in the areas of distributed algorithms for sparse optimization and transportation problems presented in this thesis as follows:

1. We develop column partition based distributed algorithms for coupled convex sparse optimization including LASSO, BPDN and their variants such as fused LASSO, fused BPDN, group LASSO and generalized LASSO, and BPDN problems. These algorithms are two-stage and are based on the favorable properties of the dual problems and exact regularization techniques. Precisely, in the first stage we compute the dual solution of the original problem, and using this dual solution we formulate an equivalent regularized basis pursuit problem whose solution is indeed a solution to the original problem guaranteed by exact regularization. The second stage consists of solving a dual problem and recovering a primal solution.

2. The overall convergence of the two-stage schemes are established via sensitivity analysis of the regularized basis pursuit (BP)-like problem. In particular we prove that for a general norm in the LASSO like and BPDN like problem, the optimal solution x^* is continuous in the vector b, where b depends on the approximate solution obtained in the first stage. Further, for the ℓ_1 norm it is shown that that solution x^* is Lipschitz continuous in b. This leads to the overall convergence and convergence rates of the two-stage distributed schemes.

3. In order to formulate the underlying coupled optimization problem as a locally coupled convex optimization problem in CAV platooning control, a decomposition method is developed for the strongly convex quadratic objective function. This method decomposes the central quadratic 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. Further, it is necessary to develop a distributed algorithm where the computation time is within one second for real-time implementation of the proposed MPC scheme under the linear vehicle dynamics. We develop Douglas Rachford splitting based distributed schemes which adhere to this restriction on the computation time. We also propose initial state warm up techniques which can be used to further reduce the computation time. It should be noted that the restriction on computation time is very crucial for real-time implementation of the proposed schemes.

4. When the nonlinear vehicle dynamics is considered in CAV platooning control, we propose a sequential convex programming (SCP) [45] based distributed scheme to solve the locally coupled nonconvex optimization problem. This SCP based scheme solves a sequence of convex, quadratically constrained quadratic programs (QCQPs) that approximate the original nonconvex program at each iteration; such a convex QCQP can be efficiently solved using (generalized) Douglas-Rachford method or other operator splitting methods [15] in the fully distributed manner. The SCP based distributed scheme converges to a stationary point, which often coincides or is close to an optimal solution, under mild assumptions. The proposed fully distributed schemes have a computation time which is less than one second which is crucial for

real-time implementation. We develop an initial warm up technique to further reduce the computation time.

5. Extensive numerical results are conducted for both the sparse optimization problems and transportation applications. In the context of sparse optimization problems, we test the proposed schemes on two types of graphs namely the cyclic graph and a random graph; see Section 2.8 for details. We perform tests on standard LASSO and BPDN, fused LASSO and BPDN, and non negative constrained LASSO and BPDN and group LASSO problems. In CAV platooning control problems, we perform tests on three scenarios under both the linear and nonlinear vehicle dynamics; see Section 3.6 for details of the scenarios. Further, in the nonlinear case, we test the proposed distributed algorithms on possibly heterogeneous CAV platoons, e.g. a homogeneous small-size platoon, a heterogeneous medium-size platoon, and a homogeneous large size platoon; see Section 4.7 for details.

1.5 Organization

This thesis is organized as follows. In Chapter II, we study the fully distributed algorithms corresponding to sparse optimization including LASSO-like and BPDN-like problems. In Chapters III and IV, we consider the fully distributed algorithms corresponding to the transportation problems. Specifically, in Chapter III we consider the linear vehicle dynamics which yields a densely coupled convex optimization problem. We develop a decomposition method for the corresponding objective function and the constraints. Further, we perform the stability analysis and present extensive numerical results. Chapter IV is an extension of Chapter III, where we consider nonlinear vehicle dynamics due to which the corresponding optimization problem becomes non-convex. Sequential convex programming techniques coupled with matrix decomposition techniques from Chapter III are used to develop fully distributed algorithms. Extensive numerical results are presented for small and large size homogeneous platoons and a medium size inhomogeneous platoon. Finally, conclusions and future research directions are discussed in Chapter V.

1.6 Notation

Let $A \in \mathbb{R}^{m \times N}$, and $R(A)$ denote the range of A. For any index set $\mathcal{S} \subseteq \{1, \ldots, N\}$, let $A_{\bullet \mathcal{S}}$ be the matrix formed by the columns of A indexed by elements of $\mathcal{S}$. Similarly, $A_{\alpha \bullet}$ is defined for an index set $\alpha \subseteq \{1, \ldots, m\}$. Let $\{\mathcal{I}_i\}_{i=1}^p$ form a disjoint union of $\{1, \ldots, N\}$, and $\{x_{\mathcal{I}_i}\}_{i=1}^p$ form a partition of $x \in \mathbb{R}^N$. For $a \in \mathbb{R}^n$, let $a_+ := \max(a, 0) \geq 0$ and $a_- := \max(-a, 0) \geq 0$. For a closed convex set $\mathcal{C}$ in $\mathbb{R}^n$, $\Pi_{\mathcal{C}}$ denotes the Euclidean projection operator onto $\mathcal{C}$. For $u, v \in \mathbb{R}^n$, $u \perp v$ stands for the orthogonality of u and v, i.e., $u^T v = 0$. Further, $\| \cdot \|_1$, $\| \cdot \|_2$ and $\| \cdot \|_\infty$ denote the ℓ_1-norm (or 1-norm), 2-norm, and ∞-norm, respectively. Let $\text{prox}_f(\cdot)$ denote the proximal operator for a proper, lower semicontinuous convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. For a given matrix W, $\lambda_{\min}(W)$, $\lambda_{\max}(W)$ respectively denote the minimum and maximum eigenvalues of W. $\mathbf{I}_C$ denotes the indicator function on a given set C, where $\mathbf{I}_C(x) = 0$ if $x \in C$, and $\mathbf{I}_C(x) = +\infty$ if $x \notin C$. For a given extended value function $F : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$, δF denotes the sub differential. $\mathcal{N}_C(x)$ denotes the normal cone of a closed convex set C at $x \in C$. An operator C defined on $\mathbb{R}^n$ is β-cocoercive, $\beta > 0$ if $\langle Cx - Cy, x - y \rangle \geq \beta \|Cx - Cy\|^2, \ \forall \ x, y, \in \mathbb{R}^n$.

CHAPTER II

Column Partition based Fully Distributed Algorithms for Coupled Convex Sparse Optimization Problems

2.1 Motivation and Introduction

This chapter develops column partition based distributed algorithms for a wide range of LASSO/BPDN-like problems by exploiting convex optimization techniques, e.g., dual problems, exact regularization, and distributed optimization. First, motivated by the dual approach for distributed BP and BPDN [50], [51], [92], we consider the Lagrangian dual problems of the LASSO/BPDN-like problems, which are separable or locally coupled and can be solved via column partition based distributed schemes. By using the solution properties of the LASSO/BPDN-like problems, we show that a primal solution is a solution to a BP-like problem depending on a dual solution. Under exact regularization conditions, a primal solution can be obtained from the dual of a regularized BP-like problem which can be solved by another column partition based distributed scheme. This leads to two-stage, column partition based distributed schemes, and many existing distributed schemes can be used at each stage. The overall convergence of the two-stage schemes is established via sensitivity analysis of the regularized BP-like problem. The proposed schemes are applicable to a broad class of generalized BP, LASSO and BPDN under mild assumptions on a network; we only assume that a network is static, connected and bidirectional. The materials of this chapter are reported in our recent journal publication [66].

The chapter is organized as follows. Section 2.2 presents problem formulations and solution properties with an overview of key ideas given in Section 2.3. Exact regularization is addressed in Section 2.4. Section 2.5 formulates dual problems and establishes properties in connection with the primal problems. Column partition based distributed schemes are developed in Section 2.6, whose overall convergence is shown in Section 2.7. Numerical results are given in Section 2.8 with summary in Section 2.9.

2.2 Problem Formulation and Solution Properties

We consider a class of convex sparse minimization problems and their generalizations or extensions whose formulations are given as follows.

• **Basis Pursuit (BP) and Extensions**. This problem intends to recover a sparse vector from noiseless measurement b given by the following linear equality constrained optimization problem

$$\text{BP}: \quad \min_{x \in \mathbb{R}^N} \|x\|_1 \quad \text{subject to} \quad Ax = b, \tag{2.1}$$

where we assume $b \in R(A)$. Geometrically, this problem seeks to minimize the 1-norm distance from the origin to the affine set defined by $Ax = b$. A generalization of the BP (2.1) is $\min_{x \in \mathbb{R}^N} \|Ex\|_1$ subject to $Ax = b$, where $E \in \mathbb{R}^{r \times N}$ is a matrix.

• **Least Absolute Shrinkage and Selection Operator (LASSO) and Extensions.** The standard LASSO intends to minimize the loss function $\|Ax - b\|_2^2$ along with the ℓ_1-norm penalty on x treated as a convex relaxation of the sparsity of x:

$$\text{LASSO}: \quad \min_{x \in \mathbb{R}^N} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1, \tag{2.2}$$

where $\lambda > 0$ is the penalty parameter. A generalized LASSO is given by $\min\limits_{x \in \mathbb{R}^N} \frac{1}{2}\|Ax-b\|_2^2 + \|Ex\|_1$, where $E \in \mathbb{R}^{r \times N}$ is a given matrix. It includes several extensions and variations of the standard LASSO:

(i) **Fused LASSO**: $\min_{x \in \mathbb{R}^N} \frac{1}{2}\|Ax-b\|_2^2 + \lambda_1 \|x\|_1 + \lambda_2 \|D_1 x\|_1$, where $D_1 \in \mathbb{R}^{(N-1) \times N}$ denotes the first order difference matrix. Letting $E := \begin{bmatrix} \lambda_1 I_N \\ \lambda_2 D_1 \end{bmatrix}$, the fused LASSO can be converted to the generalized LASSO.

(ii) **Group LASSO** which is widely used in statistics for model selection [93]: given $\lambda_i > 0$ for $i = 1, \ldots, p$,

$$\min_{x \in \mathbb{R}^N} \frac{1}{2}\|Ax - b\|_2^2 + \sum_{i=1}^{p} \lambda_i \|x_{\mathcal{I}_i}\|_2. \tag{2.3}$$

Other generalizations include generalized total variation denoising and ℓ_1-trend filtering [36]. A generalized total variation denoising is a generalized LASSO with $E = \lambda D_1$ for $\lambda > 0$, whereas the generalized ℓ_1-trend filtering has $E = \lambda D_2$, where D_2 is the second order difference matrix.

• **Basis Pursuit Denoising (BPDN) and Extensions**. Consider the following constrained optimization problem which incorporates noisy signals:

$$\text{BPDN}: \quad \min_{x\in\mathbb{R}^N} \|x\|_1 \quad \text{subject to} \quad \|Ax-b\|_2 \le \sigma, \tag{2.4}$$

where $\sigma > 0$ characterizes the bound of noise or errors. Note that when $\sigma = 0$, it reduces to the basis pursuit. We assume that $\|b\|_2 > \sigma$ since otherwise, $x_* = 0$ is the trivial solution. Similar to the LASSO, the BPDN has several generalizations and extensions. For example, it can be extended to $\min_{x\in\mathbb{R}^N} \|Ex\|_1$ subject to $\|Ax-b\|_2 \le \sigma$, where $E \in \mathbb{R}^{r\times N}$ is a matrix.

We summarize some fundamental solution properties of the aforementioned problems to be used in the subsequent development. For the convenience of generalizations and extensions, we treat the aforementioned problems in a more general setting. Let the constant $q > 1$, $E \in \mathbb{R}^{r\times N}$ be a matrix, $\|\cdot\|_\star$ be a norm on the Euclidean space, and $\mathcal{C}$ be a polyhedral set. Consider the following problems:

$$(P_1): \quad \min_{x\in\mathbb{R}^N} \|Ex\|_\star \quad \text{subject to} \quad Ax = b, \quad \text{and } x \in \mathcal{C} \tag{2.5}$$

$$(P_2): \quad \min_{x\in\mathbb{R}^N} \tfrac{1}{2}\|Ax-b\|_q^q + \|Ex\|_\star \quad \text{subject to} \quad x \in \mathcal{C} \tag{2.6}$$

$$(P_3): \quad \min_{x\in\mathbb{R}^N} \|Ex\|_\star \quad \text{subject to} \quad \|Ax-b\|_q \le \sigma \quad \text{and} \quad x \in \mathcal{C}. \tag{2.7}$$

We call (P_1), (P_2) and (P_3) the BP-like, LASSO-like, and BPND-like problems, respectively. All the BP, LASSO, and BPDN models introduced before can be formulated within the above framework. For example, in the group LASSO, $\|x\|_\star := \sum_{i=1}^{p} \lambda_i \|x_{\mathcal{I}_i}\|_2$ is a norm. Letting $E := I_N$ and $q = 2$, the group LASSO is a special case of (P_2).

Proposition 2.2.1. *Fix $q > 1$, and assume that the problems (P_1)-(P_3) are feasible. The following hold:*

(i) Each of the problems $(P_1) - (P_3)$ attains a minimizer;

(ii) Let $\mathcal{H}_2$ be the solution set of (P_2). Then $Ax = Ax'$ and $\|Ex\|_\star = \|Ex'\|_\star$ for all $x, x' \in \mathcal{H}_2$;

(iii) Suppose in (P_3) that $\|b\|_q > \sigma$, $0 \in \mathcal{C}$, and the optimal value of (P_3) is positive. Then each minimizer x_* of (P_3) satisfies $\|Ax_* - b\|_q = \sigma$ and Ax is constant on the solution set.

Proof. Statements (i) and (ii) follow from a similar proof of [52, Theorem 4.1] and [94, Lemma 4.1], respectively. For Statement (iii), by a similar argument of [94, Lemma 4.2(3)], we have that any minimizer x_* of (P_3) satisfies $\|Ax_* - b\|_q = \sigma$. Since $\|\cdot\|_q^q$ is strictly convex for $q > 1$ [68, Appendix] and $\|Ax - b\|_q^q$ is constant (whose value is σ^q) on the solution set, we deduce that Ax is constant on the solution set. □

A sufficient condition for the optimal value of (P_3) to be positive, along with the conditions that $\|b\|_q > \sigma$ and $0 \in \mathcal{C}$, is that E has full column rank. In fact, when $\|b\|_q > \sigma$ and $0 \in \mathcal{C}$, any minimizer x_* must be nonzero. If E has full column rank, then $Ex_* \neq 0$ so that $\|Ex_*\|_\star > 0$.

To compute a solution of the LASSO in (2.12) using its dual solution, we need the following result similar to [94, Theorem 2.1] or [52, Proposition 3.2]. To be self-contained, we present its proof below.

Proposition 2.2.2. *The following hold:*

(i) Let x_* be a minimizer of (P_2) given by (2.6). Then z_* is a minimizer of (P_2) if and only if z_* is a minimizer of the BP-like problem given by (2.5), i.e., (P_1) :

$\min_{z\in\mathbb{R}^N}\ \|Ez\|_\star$ *subject to* $Az = Ax_*$ *and* $z \in \mathcal{C}$. *Furthermore, the optimal value of* (P_1) *equals* $\|Ex_*\|_\star$.

(ii) Let x_* *be a minimizer of* (P_3) *given by (2.7) which satisfies:* $\|b\|_q > \sigma$, $0 \in \mathcal{C}$, *and the optimal value of* (P_3) *is positive. Then* z_* *is a minimizer of* (P_3) *if and only if* z_* *is a minimizer of the BP-like problem (2.5) with* $b := Ax_*$, *and the optimal value of this* (P_1) *equals* $\|Ex_*\|_\star$.

Proof. (i) Let $\mathcal{H}_2$ be the solution set of (P_2) given by (2.6). By Proposition 2.2.1, $Ax = Ax_*$ and $\|Ex\|_\star = \|Ex_*\|_\star$ for any $x \in \mathcal{H}_2$. Let $J(x) := \frac{1}{2}\|Ax - y\|_q^q + \|Ex\|_\star$ be the objective function. For the "if" part, let z_* be a minimizer of (P_1). Then $z_* \in \mathcal{C}$, $Az_* = Ax_*$ and $\|Ez_*\|_\star \le \|Ex_*\|_\star$. Hence, $J(z_*) \le J(x_*)$. On the other hand, $J(x_*) \le J(z_*)$ because x_* is a minimizer of (P_2). Therefore, $J(x_*) = J(z_*)$ so that z_* is a minimizer of (P_2). It also implies that $\|Ez_*\|_\star = \|Ex_*\|_\star$ or equivalently the optimal value of (P_1) equals $\|Ex_*\|_\star$. To show the "only if" part, let z_* be a minimizer of (P_2). Suppose z_* is not a minimizer of (P_1). Then there exists $u \in \mathbb{R}^N$ such that $u \in \mathcal{C}$, $Au = Ax_*$ and $\|Eu\|_\star < \|Ez_*\|_\star$. Since z_* is a minimizer of (P_2), we have $Az_* = Ax_*$ and $\|Ez_*\|_\star = \|Ex_*\|_\star$. Hence, $J(u) < J(z_*)$, yielding a contradiction.

(ii) Suppose (P_3) satisfies the specified conditions, and let $\mathcal{H}_3$ denote its solution set. By statement (iii) Proposition 2.2.1, we have $\mathcal{H}_3 = \{x \in \mathcal{C} \mid Ax = Ax_*,\ \|Ex\|_\star = \|Ex_*\|_\star\}$ for a minimizer x_* of (P_3). "If": suppose z_* be a minimizer of (P_1) with $b := Ax_*$. Then $z_* \in \mathcal{C}$, $Az_* = Ax_*$ and $\|Ez_*\|_\star \le \|Ex_*\|_\star$. This shows that z_* is a feasible point of (P_3) and hence a minimizer in view of $\|Ez_*\|_\star = \|Ex_*\|_\star$. "Only if": since any feasible point of (P_1) with $b := Ax_*$ is a feasible point of (P_3) and since x_* is a feasible point of this (P_1), we see that the optimal value of this (P_1) equals $\|Ex_*\|_\star$. Suppose z_* is a minimizer of

(P_3). Then $z_* \in \mathcal{H}_3$ such that $z_* \in \mathcal{C}$, $Az_* = Ax_*$, and $\|Ez_*\|_\star = \|Ex_*\|_\star$. Hence z_* is a feasible point of this (P_1) and thus a minimizer of this (P_1). $\square$

2.3 Overview of the Development of Column Partition based Distributed Algorithms

One of the major contributions of this chapter is to develop two-stage, column partition based distributed algorithms for the LASSO-like and BPDN-like problems (2.6)-(2.7). To facilitate the presentation of this development, we outline its main ideas and provide a road map of key steps in this section.

2.3.1 Illustration of Main Ideas via the Standard LASSO

For the simplicity of illustration, consider the standard LASSO (2.2) first. Although the primal problem (2.2) of the LASSO is densely coupled, its dual problem

$$(D): \quad \min_{y} \frac{\|y\|_2^2}{2} + b^T y, \quad \text{subject to} \quad \|A^T y\|_\infty \le \lambda$$

attains favorable properties for column partition based distributed computation since A^T is used in the constraint and the norm $\|\cdot\|_\infty$ is separable. Hence, (D) can be formulated as a separable consensus optimization problem, for which column partition based distributed schemes can be developed. Let y_* be the unique dual solution. A critical question is how to recover a primal solution from y_*. One possible way is to consider the regularized LASSO:

$$\text{r-LASSO}: \quad \min_{x \in \mathbb{R}^N} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1 + \frac{\alpha}{2}\|x\|_2^2,$$

where $\frac{\alpha}{2}\|x\|_2^2$ is the regularization term with the regularization parameter $\alpha > 0$. The dual of r-LASSO enjoys similar favorable properties for column partition based distributed computation, and there is a one-to-one correspondence between a primal solution to r-LASSO and its dual solution. However, a primal solution to r-LASSO is generally *not* a desired solution to the original LASSO (2.2) even when $\alpha > 0$ is sufficiently small (cf. Example 2.4.1). In other words, *exact regularization* of the LASSO fails in general.

Despite this negative result, it follows from duality theory (cf. Lemma 2.5.2) that any solution x_* to the LASSO (2.2) satisfies $Ax_* = b + y_*$, where y_* is the unique dual solution to (D) indicated above. Moreover, in view of Statement (i) of Proposition 2.2.2, each solution to the following BP:

$$\text{BP}: \quad \min_{z \in \mathbb{R}^N} \ \|z\|_1 \ \text{ subject to } \ Az = b + y_*$$

is a solution to the LASSO. Since the above BP is exactly regularized [50], one can solve the following regularized BP

$$\text{r-BP}: \quad \min_{x \in \mathbb{R}^N} \ \|x\|_1 + \frac{\alpha}{2}\|x\|_2^2 \ \text{ subject to } \ Ax = b + y_*$$

for a small $\alpha > 0$ from its dual problem given below via a column partition based distributed scheme:

$$(D_{\text{r-BP}}): \quad \min_{y} \ \Big((b + y_*)^T y + \frac{1}{2\alpha} \sum_{i=1}^{p} \|S\big(-(A^T y)_{\mathcal{I}_i}\big)\|_2^2 \Big),$$

where $S(\cdot)$ is the soft thresholding operator (cf. Section 2.5.2). Letting $\widehat{y}_*$ be a dual solution to $(D_{\text{r-BP}})$, the primal solution to r-BP is recovered from $\widehat{y}_*$ as $x^*_{\mathcal{I}_i} = -\frac{1}{\alpha} S\big((A^T \widehat{y}_*)_{\mathcal{I}_i}\big)$,

$\forall\, i = 1, \dots, p$, which is a solution to the LASSO. This yields a two-stage column partitioned based distributed scheme for the LASSO: its dual solution y_* is solved in the first stage, and a primal solution is obtained from the dual of a regularized BP in the second stage using y_* (cf. Algorithm 1 in Section 2.6.1).

2.3.2 Overview of Key Steps for General Problems

To apply the above ideas to a broader class of problems, e.g., the standard BPDN, the fused LASSO, and other problems possibly subject to polyhedral constraints, we address the following theoretical and numerical tasks which pave the way to developing column partition based distributed schemes:

1) *Exact Regularization.* We show that when the ℓ_1-norm is used, the BP-like problem (2.5) subject to a polyhedral constraint is exactly regularized, whereas the LASSO and BPDN are not in general (cf. Section 2.4). These results lay a ground for using regularized BP-like problems to recover a desired primal solution in the second stage and justify why regularized LASSO-like and BPDN-like problems are not considered.

2) *Dual Formulations.* We derive dual problems of the above mentioned primal problems, e.g., the regularized BP-like problem, LASSO-like problem, and BPDN-like problem. These dual formulations are used in both stages of the LASSO-like and BPDN-like problems: in the first stage, we use it to obtain a dual solution to the LASSO-like (resp. BPDN-like) problem; in the second stage, we use the dual of a regularized BP-like problem to recover a primal solution to the LASSO-like (resp. BPDN-like) problem. Further, we study the relation between a primal solution and a dual solution via duality theory (cf. Lemmas 2.5.2 and 2.5.3). Along with Proposition 2.2.2, this relation yields a regularized BP-like problem in the second stage of the LASSO-like (resp. BPDN-like)

problem. Besides, we develop various reduced dual problems which facilitate developing distributed schemes.

3) *Distributed Scheme Development.* We show that the obtained dual problems can be formulated as separable or locally coupled convex consensus optimization problems. For example, consider the fused LASSO and fused BPDN. Their dual problems and those of the corresponding regularized BP's are given by locally coupled consensus optimization such that a wide range of existing methods, e.g., operator splitting methods [15] and consensus ADMM [10], [47], can be used to develop columned partitioned based distributed schemes over undirected and connected networks. Numerical tests are conducted to evaluate performance of these schemes.

4) *Overall Convergence.* Many distributed algorithms can be used in each stage and are convergent under suitable conditions. However, the first-stage scheme generates an approximate solution to a true dual solution, and this raises the question of whether using an approximate dual solution leads to significant discrepancy when solving the regularized BP-like problem in the second stage. Using sensitivity analysis tools for the regularized BP-like problem, we establish continuous dependence of its solution on certain parameters and prove the overall convergence of the two-stage distributed algorithms.

The ℓ_1-norm will be considered in the above-mentioned key steps for many representative convex sparse optimization problems. Nevertheless, the dual formulations and related duality results can be obtained for an arbitrary norm $\|\cdot\|_\star$ in (P_1), (P_2) and (P_3). This allows us to handle the group LASSO (2.3) and its extensions defined by the norm $\|x\|_\star := \sum_{i=1}^p \lambda_i \|x_{\mathcal{I}_i}\|_2$. We will treat this general framework in Section 2.5.

2.4 Exact Regularization

A key step in the development of column partition based distributed algorithms is using dual problems. To establish a relation between solutions of a primal problem and its dual, we consider regularization of the primal problem, which is expected to give rise to a solution of the original primal problem. This pertains to the exact regularization of the original primal problem [20].

We briefly review the exact regularization of general convex programs given in [20]. Consider the convex minimization problem (P) and its regularized problem (P_ε) for some $\varepsilon \geq 0$:

$$(P): \quad \min_{x \in \mathcal{P}} f(x); \qquad\qquad (P_\varepsilon): \quad \min_{x \in \mathcal{P}} f(x) + \varepsilon h(x),$$

where $f, h : \mathbb{R}^N \to \mathbb{R}$ are real-valued convex functions, and $\mathcal{P}$ is a closed convex set. It is assumed that (P) has a solution, and h is coercive such that (P_ε) has a solution for each $\varepsilon > 0$. A weaker assumption can be made for h; see [20, Section 1.2] for details. We call the problem (P) *exactly regularized* if there exists $\overline{\varepsilon} > 0$ such that for any $\varepsilon \in (0, \overline{\varepsilon}]$, any solution of (P_ε) is a solution of (P). To establish the exact regularization, consider the following convex program: letting f_* be the optimal value of (P),

$$(P_h): \quad \min_{x \in \mathcal{P},\, f(x) \leq f_*} h(x).$$

Clearly, the constraint set of (P_h) is equivalent to $\{x \,|\, x \in \mathcal{P}, f(x) = f_*\}$, which is the solution set of (P). It is shown in [20, Theorem 2.1] or [20, Corollary 2.2] that (P) is exactly regularized by the regularization function h if and only if (P_h) has a Lagrange multiplier $\mu_* \geq 0$, i.e., there exists a constant $\mu_* \geq 0$ such that $\min_{x \in \mathcal{P},\, f(x) \leq f_*} h(x) = \min_{x \in \mathcal{P}} h(x) + \mu_* \big(f(x) - f_* \big)$.

Corollary 2.4.1. *The problem (P_h) has a Lagrange multiplier $\mu_* \geq 0$ if and only if there exists a constant $\mu \geq 0$ such that a minimizer x_* of (P_h) is a minimizer of $\min_{x \in \mathcal{P}} h(x) + \mu(f(x) - f_*)$.*

Proof. "If": suppose a constant $\mu \geq 0$ exists such that a minimizer x_* of (P_h) is that of $\min_{x \in \mathcal{P}} h(x) + \mu(f(x) - f_*)$. Since x_* is a feasible point of (P_h), we have $x_* \in \mathcal{P}$ and $f(x_*) \leq f_*$ or equivalently $f(x_*) = f_*$. Hence, the optimal value of $\min_{x \in \mathcal{P}} h(x) + \mu(f(x) - f_*)$ is given by $h(x_*)$, which equals $\min_{x \in \mathcal{P},\, f(x) \leq f_*} h(x)$. Hence, $\mu_* := \mu \geq 0$ is a Lagrange multiplier of (P_h).

"Only If": Let $\mu_* \geq 0$ be a Lagrange multiplier of (P_h), and x_* be a minimizer of (P_h). Again, we have $x_* \in \mathcal{P}$ and $f(x_*) = f_*$. This shows that $h(x_*) + \mu_*(f(x_*) - f_*) = h(x_*)$. Hence,

$$h(x_*) \; = \; \min_{x \in \mathcal{P},\, f(x) \leq f_*} h(x) \; = \; \min_{x \in \mathcal{P}} h(x) + \mu_* \big(f(x) - f_* \big) \leq h(x_*).$$

We thus deduce that x_* is a minimizer of $\min_{x \in \mathcal{P}} h(x) + \mu(f(x) - f_*)$ with $\mu := \mu_*$. $\square$

2.4.1 Exact Regularization of Convex Piecewise Affine Function based Optimization

We consider the exact regularization of convex piecewise affine functions based convex minimization problems with its applications to ℓ_1-minimization given by the BP, LASSO, and BPDN. A real-valued continuous function $f : \mathbb{R}^N \to \mathbb{R}$ is piecewise affine (PA) if there exists a finite family of real-valued affine functions $\{f_i\}_{i=1}^{\ell}$ such that $h(x) \in$

$\{f_i(x)\}_{i=1}^{\ell}$ for each $x \in \mathbb{R}^N$. A convex PA function $f : \mathbb{R}^N \to \mathbb{R}$ has the max-formulation [60, Section 19], i.e., there exists a finite family of $(p_i, \gamma_i) \in \mathbb{R}^N \times \mathbb{R}, i = 1, \ldots, \ell$ such that $f(x) = \max_{i=1,\ldots,\ell} \left(p_i^T x + \gamma_i \right)$. Convex PA functions represent an important class of nonsmooth convex functions in many applications, e.g., the ℓ_1-norm $\|\cdot\|_1$, $f(x) := \|Ex\|_1$ for a matrix E, a polyhedral gauge, and the ℓ_∞-norm; see [52] for more discussions. We first present a technical lemma whose proof is omitted.

Lemma 2.4.1. *Let $f : \mathbb{R}^N \to \mathbb{R}$ and $h : \mathbb{R}^N \to \mathbb{R}$ be (not necessarily convex) functions and $\mathcal{P}$ be a set such that $\min_{x\in\mathcal{P}} f(x)$ attains a minimizer and its optimal value is denoted by f_*. Let the set $\mathcal{W} := \{(x,t) \,|\, x \in \mathcal{P}, f(x) \le t\}$. Consider the following problems:*

$$(P_\varepsilon): \quad \min_{x\in\mathcal{P}} f(x) + \varepsilon h(x); \qquad (P'_\varepsilon): \quad \min_{(x,t)\in\mathcal{W}} t + \varepsilon h(x), \qquad \varepsilon \ge 0;$$

$$(P_h): \quad \min_{x\in\mathcal{P},\, f(x)\le f_*} h(x); \qquad (P'_h): \quad \min_{(x,t)\in\mathcal{W},\, t\le f_*} h(x).$$

Then the following hold:

(i) Fix an arbitrary $\varepsilon \ge 0$. Then (a) if x_* is an optimal solution of (P_ε), then $(x_*, f(x_*))$ is an optimal solution of (P'_ε); (b) if (x_*, t_*) is an optimal solution of (P'_ε), then $t_* = f(x_*)$ and x_* is an optimal solution of (P_ε).

(ii) (a) If x_* is an optimal solution of (P_h), then (x_*, f_*) is an optimal solution of (P'_h); (b) if (x_*, t_*) is an optimal solution of (P'_h), then $t_* = f_*$ and x_* is an optimal solution of (P_h).

The following proposition shows exact regularization for convex PA objective functions on a polyhedral set. This result has been mentioned in [91] without a formal proof; we present a proof for completeness.

Proposition 2.4.1. *Let $\mathcal{P}$ be a polyhedral set, and $f : \mathbb{R}^N \to \mathbb{R}$ be a convex PA function such that the problem $(P) : \min_{x\in\mathcal{P}} f(x)$ has the nonempty solution set, and let $h : \mathbb{R}^N \to \mathbb{R}$ be a convex regularization function which is coercive. Then there exists $\overline{\varepsilon} > 0$ such that for any $\varepsilon \in (0, \overline{\varepsilon}]$, any optimal solution of the regularized problem (P_ε) is an optimal solution of (P).*

Proof. Let f_* be the optimal value of the problem (P). In view of Lemma 2.4.1, (P) is equivalent to (P'_0) and (P_ε) is equivalent to (P'_ε) for any $\varepsilon > 0$ in the sense given by Lemma 2.4.1. Hence, to show the exact regularization of (P) via (P_ε), it suffices to show the exact regularization of (P'_0) via (P'_ε). To show the latter, it follows from [20, Theorem 2.1] or [20, Corollary 2.2] that we only need to show that (P'_h) attains a Lagrange multiplier, namely, there exists a Lagrange multiplier $\mu_* \geq 0$ such that $\min_{(x,t)\in\mathcal{W},\, t\leq f_*} h(x) = \min_{(x,t)\in\mathcal{W}} h(x) + \mu_*(t - f_*)$, where we recall that f_* is the optimal value of (P) and $\mathcal{W} := \{(x, t) \,|\, x \in \mathcal{P}, f(x) \leq t\}$. Suppose the convex PA function f is given by $f(x) = \max_{i=1,\dots,\ell} \left(p_i^T x + \gamma_i\right)$. Then $\mathcal{W} = \{(x, t) \,|\, x \in \mathcal{P},\ p_i^T x + \gamma_i \leq t,\ \forall\, i = 1, \dots, \ell\}$, and $\mathcal{W}$ is thus a polyhedral set. Since $\mathcal{W}$ is polyhedral and $t \leq f_*$ is a linear inequality constraint, it follows from [6, Proposition 5.2.1] that there exists $\mu_* \geq 0$ such that $\min_{(x,t)\in\mathcal{W},\, t\leq f_*} h(x) = \min_{(x,t)\in\mathcal{W}} h(x) + \mu_*(t - f_*)$. By [20, Corollary 2.2], (P'_ε) is the exact regularization of (P'_0) for all small $\varepsilon > 0$. □

The above proposition yields the exact regularization for the BP-like problem with the ℓ_1-norm.

Corollary 2.4.2. *Let $\mathcal{C}$ be a polyhedral set. Then the following problem attains the exact regularization of (P_1) for all sufficiently small $\alpha > 0$:*

$$(P_{1,\alpha}) : \qquad \min_{x\in\mathbb{R}^N} \ \|Ex\|_1 + \tfrac{\alpha}{2}\|x\|_2^2 \quad \textit{subject to} \quad Ax = b, \quad \textit{and } x \in \mathcal{C}.$$

Proof. Let $f(x) := \|Ex\|_1$ which is a convex PA function, $h(x) := \|x\|_2^2$, and $\mathcal{P} := \{x \,|\, Ax = b,\ x \in \mathcal{C}\}$. Then $\mathcal{P}$ is a polyhedral set. Applying Proposition 2.4.1, we conclude that the exact regularization holds. $\square$

2.4.1.1 Failure of Exact Regularization of the LASSO and BPDN Problems

We investigate exact regularization of the LASSO and BPDN when the ℓ_1-norm is used. For simplicity, we focus on the standard problems (i.e., $\mathcal{C} = \mathbb{R}^N$) although the results developed here can be extended. It follows from Proposition 2.2.1 that the solution sets of the standard LASSO and BPDN are polyhedral. Hence, the constraint sets of (P_h) associated with the LASSO and BPDN are polyhedral. However, unlike the BP-like problem, we show by examples that exact regularization fails in general. This motivates us to develop two-stage distributed algorithms in Section 2.6 rather than directly using the regularized LASSO and BPDN. Our first example shows that in general, the standard LASSO (2.2) is *not* exactly regularized by the regularization function $h(x) = \|x\|_2^2$.

Example 2.4.1. Let $A = [I_2 \ I_2 \ \cdots \ I_2] \in \mathbb{R}^{2\times N}$ with $N = 2r$ for some $r \in \mathbb{N}$, and $b \in \mathbb{R}^2_{++}$. Hence, we can partition a vector $x \in \mathbb{R}^N$ as $x = (x^1, \ldots, x^r)$ where each $x^i \in \mathbb{R}^2$. When $0 < \lambda < 1$, it follows from the KKT condition: $0 \in A^T(Ax_* - b) + \lambda\partial\|x_*\|_1$ and a straightforward computation that a particular optimal solution x_* is given by $x_*^i = \frac{1-\lambda}{r} b > 0$ for all $i = 1, \ldots, r$. Hence, the solution set $\mathcal{H} = \{x = (x^1, \ldots, x^r) \,|\, \sum_{i=1}^r x^i = (1-\lambda)b,\ \|x\|_1 \le (1-\lambda)\|b\|_1\}$. Consider the regularized LASSO for $\alpha > 0$: $\min_{x\in\mathbb{R}^N} \frac{1}{2}\|Ax - b\|_2^2 + \lambda\,\|x\|_1 + \frac{\alpha}{2}\|x\|_2^2$. For each $\alpha > 0$, it can be shown that its unique optimal solution $x_{*,\alpha}$ is given by $x_{*,\alpha}^i = \frac{1-\lambda}{r+\alpha} b$ for each $i = 1, \ldots, r$. Hence, $x_{*,\alpha} \notin \mathcal{H}$ for any $\alpha > 0$. $\square$

In what follows, we show the failure of exact regularization of the standard BPDN (2.4). Consider the convex minimization problem for a constant $\mu \geq 0$,

$$(P_\mu): \qquad \min_{\|Ax-b\|_2 \leq \sigma} \frac{1}{2}\|x\|_2^2 + \mu\|x\|_1,$$

where $A \in \mathbb{R}^{m\times N}$, $b \in \mathbb{R}^m$, and $\sigma > 0$ with $\|b\|_2 > \sigma$.

Further, consider the max-formulation of ℓ_1-norm, i.e., $\|x\|_1 = \max_{i=1,\ldots,2^N} p_i^T x$, where each $p_i \in \left\{(\pm 1, \pm 1, \ldots, \pm 1)^T\right\} \subset \mathbb{R}^N$; see [52, Section 4.2] for details.

Lemma 2.4.2. *A feasible point $x_* \in \mathbb{R}^N$ of (P_μ) is a minimizer of (P_μ) if and only if $\|Ax_* - b\|_2 = \sigma$ and*

$$\Big[Au = 0 \quad or \quad g^T u < 0 \Big] \;\Rightarrow\; \Big(x_*^T u + \mu \max_{i\in\mathcal{I}(x_*)} p_i^T u \Big) \geq 0, \tag{2.8}$$

where $g := A^T(Ax_* - b)$, and $\mathcal{I}(x_*) := \{i \,|\, p_i^T x_* = \|x_*\|_1\}$.

Proof. It follows from a similar argument of [94, Lemma 4.2(3)] that a minimizer x_* of (P_μ) satisfies $\|Ax_* - b\|_2 = \sigma$. The rest of the proof resembles that of [52, Theorem 3.3]; we present its proof for completeness. Since (P_μ) is a convex program, it is easy to see that x_* is a minimizer of (P_μ) if and only if $u_* = 0$ is a local minimizer of the following problem:

$$(\widetilde{P}_\mu): \quad \min_{u\in\mathbb{R}^N} \Big(x_*^T u + \mu \max_{i\in\mathcal{I}(x_*)} p_i^T u \Big), \qquad \text{subject to} \quad g^T u + \frac{1}{2}\|Au\|_2^2 \leq 0.$$

In what follows, we show that the latter holds if and only if the implication (2.8) holds. Let $J(u) := x_*^T u + \mu \max_{i\in\mathcal{I}(x_*)} p_i^T u$. "If": Let $\mathcal{U}$ be a neighborhood of $u_* = 0$. For any $u \in \mathcal{U}$ satisfying $g^T u + \frac{1}{2}\|Au\|_2^2 \leq 0$, either $Au = 0$ or $Au \neq 0$. For the latter, we have $g^T u < 0$. Hence, in both cases, we deduce from (2.8) that $J(u) \geq 0 = J(u_*)$. This shows that $u_* = 0$ is a local minimizer of $(\widetilde{P}_\mu)$. "Only If": suppose $u_* = 0$ is a local minimizer of $(\widetilde{P}_\mu)$. For any u with $Au = 0$, we have $g^T u = 0$ such that $v := \beta u$ satisfies $g^T v + \frac{1}{2}\|Av\|_2^2 = 0$ for any $\beta > 0$. Hence, βu is a locally feasible point of $(\widetilde{P}_\mu)$ for all small $\beta > 0$ such that $J(\beta u) \geq J(u_*) = 0$ for all small $\beta > 0$. Since $J(\beta u) = \beta J(u)$ for all $\beta \geq 0$, we have $J(u) \geq 0$. Next consider a vector u with $g^T u < 0$. Clearly, $g^T(\beta u) + \frac{1}{2}\|A\beta u\|_2^2 < 0$ for all small $\beta > 0$ so that βu is a locally feasible point of $(\widetilde{P}_\mu)$. By a similar argument, we have $J(u) \geq 0$. $\square$

Proposition 2.4.2. *The problem $(BPDN_h)$: $\min_{\|Ax-b\|_2 \leq \sigma,\, \|x\|_1 \leq f_*} \|x\|_2^2$ with $\|b\|_2 > \sigma$ has a Lagrange multiplier if and only if there exist a constant $\mu \geq 0$ and a minimizer x_* of $(BPDN_h)$ such that*

(i) There exist $w \in \mathbb{R}_+^{|\mathcal{I}(x_*)|}$ with $\mathbf{1}^T w = 1$ and $v \in \mathbb{R}^m$ such that $x^* + \mu \sum_{i \in \mathcal{I}(x_*)} w_i p_i + A^T v = 0$; and

(ii) There exist $w' \in \mathbb{R}_+^{|\mathcal{I}(x_*)|}$ with $\mathbf{1}^T w' = 1$ and a constant $\gamma > 0$ such that $(\mathbf{1}^T w')x^* + \mu \sum_{i \in \mathcal{I}(x_*)} w'_i p_i + \gamma g = 0$,

where $\mathbf{1}$ denotes the vector of ones, $g := A^T(Ax_* - b)$, and $\mathcal{I}(x_*) := \{i \,|\, p_i^T x_* = \|x_*\|_1\}$. Furthermore, if A has full row rank, then $(BPDN_h)$ has a Lagrange multiplier if and only if there exist a constant $\mu \geq 0$ and a minimizer x_* of $(BPDN_h)$ such that

(ii') There exist $\widehat{w} \in \mathbb{R}_+^{|\mathcal{I}(x_*)|}$ with $\mathbf{1}^T \widehat{w} = 1$ and a constant $\widehat{\gamma} > 0$ such that $x^* + \mu \sum_{i \in \mathcal{I}(x_*)} \widehat{w}_i p_i + \widehat{\gamma} g = 0$.

Proof. It follows from Corollary 2.4.1 that $(BPDN_h)$ has a Lagrange multiplier if and only if there exist a constant $\mu \geq 0$ and a minimizer x_* of $(BPDN_h)$ such that x_* is a minimizer of (P_μ). Note that any minimizer x_* of $(BPDN_h)$ satisfies $\|Ax_* - b\|_2 = \sigma$ such that $x_* \neq 0$ in light of $\|b\|_2 > \sigma$. By Lemma 2.4.2, we also deduce that x_* is a minimizer of (P_μ) if and only if $\|Ax_* - b\|_2 = \sigma$ and the implication (2.8) holds. Notice that the implication holds if and only if both the following linear inequalities have no solution:

$$(\mathrm{I}): \quad Au = 0, \quad \max_{i \in \mathcal{I}(x_*)} \big(x_* + \mu p_i\big)^T u < 0; \qquad (\mathrm{II}): \quad g^T u < 0, \quad \max_{i \in \mathcal{I}(x_*)} \big(x_* + \mu p_i\big)^T u < 0.$$

By the Theorem of Alternative, we see that the inconsistency of the inequality (I) is equivalent to the existence of $(\widetilde{w}, \widetilde{v})$ with $0 \neq \widetilde{w} \geq 0$ such that $\sum_{i \in \mathcal{I}(x_*)} \widetilde{w}_i (x_* + \mu p_i) + A^T \widetilde{v} = 0$. Letting $w := \widetilde{w}/(\mathbf{1}^T \widetilde{w})$ and $v := \widetilde{v}/(\mathbf{1}^T \widetilde{w})$, we obtain condition (i). Similarly, the inconsistency of the inequality (II) is equivalent to the existence of $(\widetilde{\gamma}, \widetilde{w}')$ with $0 \neq (\widetilde{\gamma}, \widetilde{w}') \geq 0$ such that $\sum_{i \in \mathcal{I}(x_*)} \widetilde{w}'_i (x_* + \mu p_i) + \widetilde{\gamma} g = 0$. Moreover, we deduce that $\widetilde{\gamma} > 0$, since otherwise, we must have $0 \neq \widetilde{w}' \geq 0$ such that $0 = x_*^T \sum_{i \in \mathcal{I}(x_*)} \widetilde{w}'_i (x_* + \mu p_i) = (\mathbf{1}^T \widetilde{w}')(\|x_*\|_2^2 + \mu \|x_*\|_1)$, where we use $p_i^T x_* = \|x_*\|_1$ for each $i \in \mathcal{I}(x_*)$, yielding a contradiction to $x_* \neq 0$. Hence, by suitably scaling, we conclude that the inconsistency of the inequality (II) is equivalent to condition (ii). This completes the proof of the first part of the proposition.

Suppose A has full row rank. Then condition (i) holds trivially. Furthermore, since $Ax_* - b \neq 0$ for a minimizer x_* of $(BPDN_h)$, $g := A^T(Ax_* - b)$ is a nonzero vector. Hence, w' in condition (ii) must be nonzero as $\gamma g \neq 0$. Setting $\widehat{w} := w'/(\mathbf{1}^T w')$ and $\widehat{\gamma} := \gamma/(\mathbf{1}^T w')$, we obtain condition (ii'), which is equivalent to condition (ii). □

By leveraging Proposition 2.4.2, we construct the following example which shows that in general, the standard BPDN (2.4) with the ℓ_1-norm penalty is *not* exactly regularized by $h(x) = \|x\|_2^2$.

Example 2.4.2. Let $A = [D \ D \ \cdots \ D] \in \mathbb{R}^{2 \times N}$ with $N = 2r$ for some $r \in \mathbb{N}$, where $D = \mathrm{diag}(1, \beta) \in \mathbb{R}^{2 \times 2}$ for a positive constant β. As before, we partition a vector $x \in \mathbb{R}^N$

as $x = (x^1, \ldots, x^r)$ where each $x^i \in \mathbb{R}^2$. Further, let $b = (b_1, b_2)^T \in \mathbb{R}^2$ and $\sigma = 1$. We assume that $b \geq \mathbf{1}$, which is a necessary and sufficient condition for $\|v - b\|_2 \leq \sigma \Rightarrow v \geq 0$.

We first consider the convex minimization problem: $\min_{u \in \mathbb{R}^2} \|u\|_1$ subject to $\|Du - b\|_2 \leq 1$, which has a unique minimizer u_* as D is invertible for any $\beta > 0$. Further, we must have $\|Du_* - b\|_2 = 1$ and $u_* > 0$. In light of this, the necessary and sufficient optimality conditions for u_* are: there exists $\lambda \in \mathbb{R}_+$ such that $\partial \|u_*\|_1 + \lambda D^T(Du_* - b) = 0$, and $\|Du_* - b\|_2^2 = 1$. Since $u_* > 0$, we have $\lambda > 0$ and the first equation becomes $\mathbf{1} + \lambda D^T(Du_* - b) = 0$, which further gives rise to $Du_* = b - \frac{1}{\lambda} D^{-1} \mathbf{1}$. Substituting it into the equation $\|Du_* - b\|_2 = 1$, we obtain $\lambda = \frac{\sqrt{1+\beta^2}}{\beta}$. This yields $u_* = (b_1 - \frac{1}{\lambda}, \frac{1}{\beta}(b_2 - \frac{1}{\beta\lambda}))^T$. Note that for all $\beta > 0$, $0 < \frac{1}{\lambda} < 1$ and $\frac{1}{\beta\lambda} = \frac{1}{\sqrt{1+\beta^2}}$ so that $0 < \frac{1}{\beta\lambda} < 1$. Hence, $u_* > 0$ in view of $b \geq \mathbf{1}$.

It can be shown that the solution set of the BPDN is given by

$$
\begin{aligned}
\mathcal{H} &= \{x_* = (x_*^1, \ldots, x_*^r) \,|\, \|x_*\|_1 = \|u_*\|_1, \; Ax_* = Du_*\} \\
&= \{(x_*^1, \ldots, x_*^r) \,|\, \sum_{i=1}^r \|x_*^i\|_1 = \|u_*\|_1, \quad \sum_{i=1}^r x_*^i = u_*\} \\
&= \{x_* = (x_*^1, \ldots, x_*^r) \,|\, x_*^i = \lambda_i u_*, \; \sum_{i=1}^r \lambda_i = 1, \; \lambda_i \geq 0, \forall\, i\}.
\end{aligned}
$$

Therefore, it is easy to show that the regularized BPDN with $h(x) = \|x\|_2^2$ has the unique minimizer $x_* = (x_*^i)$ with $x_i^* = \frac{u_*}{r}$ for each $i = 1, \ldots, r$. Since $u_* > 0$, we have $x_* > 0$ such that $\mathcal{I}(x_*)$ is singleton with the single vector $p = \mathbf{1}$. Since A has full row rank, it follows from Proposition 2.4.2 that $(BPDN_h)$ has a Lagrange multiplier if and only if there exist constants $\mu \geq 0$, $\gamma > 0$ such that $x_* + \mu p + \gamma g = 0$ for the unique minimizer x_*, where $p = \mathbf{1}$ and $g = A^T(Ax_* - b) = A^T(Du_* - b) = -\frac{1}{\lambda}\mathbf{1}$, where $\lambda = \sqrt{1 + \frac{1}{\beta^2}}$. Since $x_* = \frac{1}{r}(u_*, \ldots, u_*)$, constants $\mu \geq 0$ and $\gamma > 0$ exist if and only if $(u_*)_1 = (u_*)_2$ or equivalently $\beta(b_1 - \frac{1}{\lambda}) = b_2 - \frac{1}{\beta\lambda}$. The latter is further equivalent to $b_2 = \beta b_1 + \frac{1-\beta^2}{\sqrt{1+\beta^2}}$. Hence, for any $\beta > 0$, $(BPDN_h)$ has a Lagrange multiplier if and only if b satisfies $b_2 = \beta b_1 + \frac{1-\beta^2}{\sqrt{1+\beta^2}}$ and $b \geq \mathbf{1}$. The set of such b's has zero measure in $\mathbb{R}^2$. For instance, when $\beta = 1$, $(BPDN_h)$ has a Lagrange multiplier if and only if $b = \theta \cdot \mathbf{1}$ for all $\theta \geq 1$. Thus the BPDN is *not* exactly regularized by $h(x) = \|x\|_2^2$ in general. $\square$

2.4.2 Exact Regularization of Grouped BP Problem Arising From Group LASSO

Motivated by the group LASSO (2.3), we investigate exact regularization of the following BP-like problem: $\min \sum_{i=1}^p \|x_{\mathcal{I}_i}\|_2$ subject to $Ax = b$, where $\{\mathcal{I}_i\}_{i=1}^p$ forms a disjoint union of $\{1, \ldots, N\}$. We call this problem the *grouped basis pursuit* or grouped BP. Here we set λ_i's in the original group LASSO formulation (2.3) as one, without loss of generality. It is shown below that its exact regularization may fail.

Example 2.4.3. Consider the grouped BP: $\min_{x,y \in \mathbb{R}^2} \|x\|_2 + \|y\|_2$ subject to $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \gamma y_1 \\ \beta y_2 \end{pmatrix} = b$, where $b = (b_1, b_2)^T \in \mathbb{R}^2$ is nonzero. Let $\gamma = 0$ and $\beta > 1$. Hence, $x_1^* = b_1$, $y_1^* = 0$, and the grouped BP is reduced to $\min_{x_2, y_2} \sqrt{b_1^2 + x_2^2} + |y_2|$ subject to $x_2 + \beta y_2 = b_2$, which is further equivalent to

$$(R_1): \quad \min_{x_2 \in \mathbb{R}} J(x_2) := \sqrt{b_1^2 + x_2^2} + \frac{|b_2 - x_2|}{\beta}.$$

It is easy to show that if $b_2 > \frac{b_1}{\sqrt{\beta^2 - 1}} > 0$, then the above reduced problem attains the unique minimizer $x_2^* = \frac{b_1}{\sqrt{\beta^2-1}}$ which satisfies $\nabla J(x_2^*) = 0$. Hence, when $b_2 > \frac{b_1}{\sqrt{\beta^2-1}} > 0$, the unique solution of the grouped BP is given by $x^* = (b_1, \frac{b_1}{\sqrt{\beta^2-1}})^T$ and $y^* = (0, (b_2 -$

$\frac{b_1}{\sqrt{\beta^2-1}})/\beta)^T$. Now consider the regularized problem for $\alpha > 0$: $\min_{x,y\in\mathbb{R}^2} \|x\|_2 + \|y\|_2 + \frac{\alpha}{2}\big(\|x\|_2^2 + \|y\|_2^2\big)$ subject to $\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \beta y_2 \end{pmatrix} = b$. Similarly, we must have $x_1^* = b_1$ and $y_1^* = 0$ such that the reduced problem is given by

$$(R_2): \quad \min_{x_2\in\mathbb{R}} \sqrt{b_1^2 + x_2^2} + \frac{|b_2 - x_2|}{\beta} + \frac{\alpha}{2}\Big(b_1^2 + x_2^2 + \frac{1}{\beta^2}(b_2 - x_2)^2\Big).$$

We claim that if $b_2 > \frac{b_1}{\sqrt{\beta^2-1}} > 0$ with $b_2 \neq \frac{1+\beta^2}{\sqrt{\beta^2-1}} b_1$, then the exact regularization fails for any $\alpha > 0$. We show this claim by contradiction. Suppose the exact regularization holds for some positive constant α. Hence, $x_2^* = \frac{b_1}{\sqrt{\beta^2-1}}$ is the solution to the reduced problem (R_2). Since $\nabla J(x_2^*) = 0$, we have $\alpha\Big(x_2^* + \frac{1}{\beta^2}(x_2^* - b_2)\Big) = 0$. This leads to $x_2^* = \frac{b_2}{1+\beta^2}$, yielding a contradiction to $b_2 \neq \frac{1+\beta^2}{\sqrt{\beta^2-1}} b_1$. Hence, the exact regularization fails. □

In spite of the failure of exact regularization in Example 2.4.3, it can be shown that the exact regularization holds for the following cases: (i) $\max(|\gamma|, |\beta|) < 1$; (ii) $\min(|\gamma|, |\beta|) > 1$; (iii) $\gamma = 0$, $\beta > 1$, $b_1 = 0$, and $b_2 \neq 0$; and (iv) $\gamma = 0$, $\beta = 1$, and $b_1 \neq 0$. Especially, the first two cases hint that the spectra of $A_{\bullet\mathcal{I}_i}$'s may determine exact regularization. Inspired by this example, we present certain sufficient conditions for which the exact regularization holds.

Lemma 2.4.3. *Consider a nonzero $b \in \mathbb{R}^m$ and a column partition $\{A_{\bullet\mathcal{I}_i}\}_{i=1}^p$ of a matrix $A \in \mathbb{R}^{m\times N}$, where $\{\mathcal{I}_i\}_{i=1}^p$ form a disjoint union of $\{1, \dots, N\}$. Suppose $A_{\bullet\mathcal{I}_1}$ is invertible, $A_{\bullet\mathcal{I}_1}^{-1} A_{\bullet\mathcal{I}_i}$ is an orthogonal matrix for each $i = 1, \dots, s$, and $\|(A_{\bullet\mathcal{I}_i})^T (A_{\bullet\mathcal{I}_1})^{-T} A_{\bullet\mathcal{I}_1}^{-1} b\|_2 < \|A_{\bullet\mathcal{I}_1}^{-1} b\|_2$ for each $i = s+1, \dots, p$. Then the exact regularization holds.*

Proof. Since $A_{\bullet\mathcal{I}_1}^{-1}Ab = A_{\bullet\mathcal{I}_1}^{-1}b$, we may assume, without loss of generality, that $A_{\bullet\mathcal{I}_1}$ is the identity matrix. Hence, $A_{\bullet\mathcal{I}_i}$ is an orthogonal matrix for $i = 2, \ldots, s$. We claim that $x^* = (\frac{(A_{\bullet\mathcal{I}_1})^T b}{s}, \cdots, \frac{(A_{\bullet\mathcal{I}_s})^T b}{s}, 0, \ldots, 0)$ is an optimal solution to the grouped BP-like problem. Clearly, it satisfies the equality constraint. Besides, it follows from the KKT conditions that there exists a Lagrange multiplier $\lambda \in \mathbb{R}^m$ such that

$$\frac{x^*_{\mathcal{I}_i}}{\|x^*_{\mathcal{I}_i}\|_2} + (A_{\bullet\mathcal{I}_i})^T\lambda = 0, \quad \forall\, i = 1, \ldots, s; \qquad 0 \in B_2(0,1) + (A_{\bullet\mathcal{I}_j})^T\lambda, \quad \forall\, j = s+1, \ldots, p.$$

Note that (i) $\lambda = -b/\|b\|_2$; and (ii) for each $j = s+1, \ldots, p$, $\|(A_{\bullet\mathcal{I}_j})^T\lambda\|_2 < 1$ in view of $\|(A_{\bullet\mathcal{I}_j})^T b\|_2 < \|b\|_2$. Hence, x^* is indeed a minimizer. Now consider the regularized grouped BP-like problem with the parameter $\alpha > 0$. We claim that $x^* = (\frac{(A_{\bullet\mathcal{I}_1})^T b}{s}, \ldots, \frac{(A_{\bullet\mathcal{I}_s})^T b}{s}, 0, \ldots, 0)$ is an optimal solution of the regularized problem for any sufficiently small $\alpha > 0$. To see this, the KKT condition is given by

$$\frac{x^*_{\mathcal{I}_i}}{\|x^*_{\mathcal{I}_i}\|_2} + \alpha x^*_{\mathcal{I}_i} + (A_{\bullet\mathcal{I}_i})^T\widehat{\lambda} = 0,\ \forall\, i = 1, \ldots, s;\ 0 \in B_2(0,1) + (A_{\bullet\mathcal{I}_j})^T\widehat{\lambda},\ \forall\, j = s+1, \ldots, p.$$

Hence, $\widehat{\lambda} = -\big(\frac{1}{\|b\|_2} + \frac{\alpha}{s}\big)b$ such that $\|\widehat{\lambda}\|_2 = 1 + \frac{\alpha}{s}\|b\|_2$. Since $\|(A_{\bullet\mathcal{I}_j})^T b\|_2 < \|b\|_2$ for each $j = s+1, \ldots, p$, we have $\|(A_{\bullet\mathcal{I}_j})^T\widehat{\lambda}\|_2 = \big(\frac{1}{\|b\|_2} + \frac{\alpha}{s}\big)\|(A_{\bullet\mathcal{I}_j})^T b\|_2 \leq 1, \forall\, j = s+1, \ldots, p$ for all sufficiently small $\alpha > 0$. Hence, x^* is a solution of the regularized problem for all small $\alpha > 0$, and exact regularization holds. $\square$

If the exact knowledge of $b \neq 0$ is unknown, the condition that $\|A^T_{\bullet\mathcal{I}_i}A^{-T}_{\bullet\mathcal{I}_1}A^{-1}_{\mathcal{I}_1}b\|_2 < \|A^{-1}_{\bullet\mathcal{I}_1}b\|_2$ for each $i = s+1, \ldots, p$ can be replaced by the following condition: $\|A^T_{\bullet\mathcal{I}_i}A^{-T}_{\bullet\mathcal{I}_1}\|_2 < 1$ for each $i = s+1, \ldots, p$.

2.5 Dual Problems: Formulations and Properties

We develop dual problems of the regularized BP as well as those of the LASSO and BPDN in this section. These dual problems and their properties form a foundation for the development of column partition based distributed algorithms. As before, $\{\mathcal{I}_i\}_{i=1}^p$ is a disjoint union of $\{1, \ldots, N\}$.

Consider the problems (P_1)-(P_3) given by (2.5)-(2.7), where $E \in \mathbb{R}^{r \times N}$ and $\|\cdot\|_\star$ is a general norm on $\mathbb{R}^r$. Let $\|\cdot\|_\diamond$ be the dual norm of $\|\cdot\|_\star$, i.e., $\|z\|_\diamond := \sup\big\{z^T v \,|\, \|v\|_\star \leq 1\big\}$, $\forall\, z \in \mathbb{R}^r$. As an example, the dual norm of the ℓ_1-norm is the ℓ_∞-norm. When $\|x\|_\star := \sum_{i=1}^p \|x_{\mathcal{I}_i}\|_2$ arising from the group LASSO, its dual norm is $\|z\|_\diamond = \max_{i=1,\ldots,p} \|z_{\mathcal{I}_i}\|_2$. Since the dual of the dual norm is the original norm, we have $\|x\|_\star = \sup\big\{x^T v \,|\, \|v\|_\diamond \leq 1\big\}, \forall\, x \in \mathbb{R}^r$. Further, let $B_\diamond(0,1) := \{v \,|\, \|v\|_\diamond \leq 1\}$ denote the closed unit ball centered at the origin with respect to $\|\cdot\|_\diamond$. Clearly, the subdifferential of $\|\cdot\|_\star$ at $x = 0$ is $B_\diamond(0,1)$.

2.5.1 Dual Problems: General Formulations

Strong duality will be exploited for the above mentioned problems and their corresponding dual problems. For this purpose, the following minimax result is needed.

Lemma 2.5.1. *Consider the convex program* $(P) : \inf_{z \in \mathcal{P}, Az=b, Cz \leq d} J(z)$*, where* $J(z) := \|Ez\|_\star + f(z)$*,* $f : \mathbb{R}^n \to \mathbb{R}$ *is a convex function,* $\mathcal{P} \subseteq \mathbb{R}^n$ *is a polyhedral set,* A, C, E *are matrices, and* b, d *are vectors. Suppose that* (P) *is feasible and has a finite infimum. Then*

$$\begin{aligned}
&\inf_{z \in \mathcal{P}} \Big(\sup_{y,\, \mu \geq 0,\, \|v\|_\diamond \leq 1} \Big[(Ez)^T v + f(z) + y^T(Az - b) + \mu^T(Cz - d) \Big] \Big) \\
&\quad = \sup_{y,\, \mu \geq 0,\, \|v\|_\diamond \leq 1} \Big(\inf_{z \in \mathcal{P}} \Big[(Ez)^T v + f(z) + y^T(Az - b) + \mu^T(Cz - d) \Big] \Big).
\end{aligned}$$

Proof. Let $J_* > -\infty$ be the finite infimum of (P). Since $\mathcal{P}$ is polyhedral, it follows from [6, Proposition 5.2.1] that the strong duality holds, i.e., $J_* = \inf_{z\in\mathcal{P}} \big[\sup_{y,\mu\geq 0} J(z) + y^T(Az-b) + \mu^T(Cz-d) \big] = \sup_{y,\mu\geq 0} \big[\inf_{z\in\mathcal{P}} J(z) + y^T(Az-b) + \mu^T(Cz-d) \big]$, and the dual problem of (P) attains an optimal solution (y_*, μ_*) with $\mu_* \geq 0$ such that $J_* = \inf_{z\in\mathcal{P}} J(z) + y_*^T(Az-b) + \mu_*^T(Cz-d)$. Therefore,

$$
\begin{aligned}
J_* &= \inf_{z\in\mathcal{P}} \|Ez\|_\star + f(z) + y_*^T(Az-b) + \mu_*^T(Cz-d) \\
&= \inf_{z\in\mathcal{P}} \Big(\sup_{\|v\|_\diamond \leq 1} \Big[(Ez)^T v + f(z) + y_*^T(Az-b) + \mu_*^T(Cz-d) \Big] \Big) \\
&= \sup_{\|v\|_\diamond \leq 1} \Big(\inf_{z\in\mathcal{P}} \Big[(Ez)^T v + f(z) + y_*^T(Az-b) + \mu_*^T(Cz-d) \Big] \Big) \\
&\leq \sup_{y,\,\mu\geq 0,\,\|v\|_\diamond \leq 1} \Big(\inf_{z\in\mathcal{P}} \Big[(Ez)^T v + f(z) + y^T(Az-b) + \mu^T(Cz-d) \Big] \Big) \\
&\leq \inf_{z\in\mathcal{P}} \Big(\sup_{y,\,\mu\geq 0,\,\|v\|_\diamond \leq 1} \Big[(Ez)^T v + f(z) + y^T(Az-b) + \mu^T(Cz-d) \Big] \Big) \;=\; J_*,
\end{aligned}
$$

where the third equation follows from Sion's minimax theorem [73, Corollary 3.3] and the fact that $B_\diamond(0,1)$ is a convex compact set, and the second inequality is due to the weak duality. $\square$

In what follows, we consider a general polyhedral set of the following form unless otherwise stated

$$
\mathcal{C} := \{x \in \mathbb{R}^N \,|\, Cx \leq d\}, \quad C \in \mathbb{R}^{\ell\times N}, \quad d \in \mathbb{R}^\ell. \tag{2.9}
$$

As before, $\{\mathcal{I}_i\}_{i=1}^p$ is a disjoint union of $\{1,\ldots,N\}$.

• **Dual Problem of the Regularized BP-like Problem** Consider the regularized BP-like problem for a fixed regularization parameter $\alpha > 0$:

$$\min_{Ax=b,\, x\in\mathcal{C}} \|Ex\|_\star + \frac{\alpha}{2}\|x\|_2^2, \tag{2.10}$$

where $b \in R(A) \cap A\mathcal{C}$ with $A\mathcal{C} := \{Ax \,|\, x \in \mathcal{C}\}$. Let $\mu \in \mathbb{R}^\ell_+$ be the Lagrange multiplier for the polyhedral constraint $Cx \le d$. It follows from Lemma 2.5.1 with $z = x$ and $\mathcal{P} = \mathbb{R}^N$ that

$$\begin{aligned}
&\min_{Ax=b,\, x\in\mathcal{C}} \|Ex\|_\star + \frac{\alpha}{2}\|x\|_2^2 \\
=\;& \inf_x \Big(\sup_{y,\, \mu\ge 0,\, \|v\|_\diamond \le 1} \Big[(Ex)^T v + \frac{\alpha}{2}\|x\|_2^2 + y^T(Ax-b) + \mu^T(Cx-d) \Big] \Big) \\
=\;& \sup_{y,\, \mu\ge 0,\, \|v\|_\diamond \le 1} \Big(\inf_x \Big[(Ex)^T v + \frac{\alpha}{2}\|x\|_2^2 + y^T(Ax-b) + \mu^T(Cx-d) \Big] \Big) \\
=\;& \sup_{y,\, \mu\ge 0,\, \|v\|_\diamond \le 1} \Big(-b^T y - \mu^T d + \sum_{i=1}^p \inf_{x_{\mathcal{I}_i}} \Big[\frac{\alpha}{2}\|x_{\mathcal{I}_i}\|_2^2 + \big((A^T y + E^T v + C^T\mu)_{\mathcal{I}_i}\big)^T x_{\mathcal{I}_i} \Big] \Big) \\
=\;& \sup_{y,\, \mu\ge 0,\, \|v\|_\diamond \le 1} \Big(-b^T y - \mu^T d - \frac{1}{2\alpha} \sum_{i=1}^p \big\| \big(A^T y + E^T v + C^T\mu\big)_{\mathcal{I}_i} \big\|_2^2 \Big),
\end{aligned}$$

This leads to the equivalent dual problem:

$$(\mathrm{D}): \quad \min_{y,\, \mu\ge 0,\, \|v\|_\diamond \le 1} \Big(b^T y + d^T \mu + \frac{1}{2\alpha} \sum_{i=1}^p \big\| (A^T y + E^T v + C^T\mu)_{\mathcal{I}_i} \big\|_2^2 \Big). \tag{2.11}$$

Let $(y_*, \mu_*, v_*) \in \mathbb{R}^m \times \mathbb{R}^\ell_+ \times B_\diamond(0,1)$ be an optimal solution of the dual problem; its existence is shown in the proof of Lemma 2.5.1. Consider the Lagrangian $L(x, y, \mu, v) := (Ex)^T v + \frac{\alpha}{2}\|x\|_2^2 + y^T(Ax-b) + \mu^T(Cx-d)$. Then by the strong duality given in Lemma 2.5.1, we see from $\nabla_x L(x_*, y_*, \mu_*, v_*) = 0$ that the unique optimal solution

$x^* = (x^*_{\mathcal{I}_i})_{i=1}^p$ of (2.10) is given by

$$x^*_{\mathcal{I}_i} \;=\; -\frac{1}{\alpha}\Big(A^T y_* + E^T v_* + C^T \mu_*\Big)_{\mathcal{I}_i}, \qquad \forall\, i = 1, \ldots, p.$$

• **Dual Problem of the LASSO-like Problem** Consider the LASSO-like problem for $A \in \mathbb{R}^{m\times N}$, $b \in \mathbb{R}^m$, and $E \in \mathbb{R}^{r\times N}$:

$$\min_{x\in\mathcal{C}} \frac{1}{2}\|Ax - b\|_2^2 + \|Ex\|_\star. \tag{2.12}$$

It follows from Lemma 2.5.1 with $z = (x, u)$ and $\mathcal{P} = \mathbb{R}^N \times \mathbb{R}^m$ that

$$\begin{aligned}
\min_{x\in\mathcal{C}} \frac{1}{2}\|Ax - b\|_2^2 + \|Ex\|_\star \;&=\; \inf_{x\in\mathcal{C},\, u=Ax-b} \frac{\|u\|_2^2}{2} + \|Ex\|_\star \\
&= \inf_{x,u}\Big(\sup_{y,\,\mu\geq 0,\,\|v\|_\diamond\leq 1} \Big\{ \frac{\|u\|_2^2}{2} + (Ex)^T v + y^T(Ax - b - u) + \mu^T(Cx - d) \Big\}\Big) \\
&= \sup_{y,\,\mu\geq 0,\,\|v\|_\diamond\leq 1} \Big(\inf_{x,u} \frac{\|u\|_2^2}{2} + (Ex)^T v + y^T(Ax - b - u) + \mu^T(Cx - d)\Big) \\
&= \sup_{y,\,\mu\geq 0,\,\|v\|_\diamond\leq 1} \Big(- b^T y - \mu^T d + \inf_u \Big(\frac{\|u\|_2^2}{2} - y^T u\Big) \\
&\qquad\qquad + \sum_{i=1}^p \inf_{x_{\mathcal{I}_i}} \big[(A^T y + E^T v + C^T\mu)_{\mathcal{I}_i}\big]^T x_{\mathcal{I}_i}\Big), \\
&= \sup_{y,\,\mu\geq 0,\,\|v\|_\diamond\leq 1} \Big\{ - b^T y - \frac{\|y\|_2^2}{2} - \mu^T d \,:\, (A^T y + E^T v + C^T\mu)_{\mathcal{I}_i} = 0,\; i = 1, \ldots, p\Big\}.
\end{aligned}$$

This yields the equivalent dual problem

$$(\mathrm{D}): \quad \min_{y,\,\mu\geq 0,\,\|v\|_\diamond\leq 1} \Big\{\frac{\|y\|_2^2}{2} + b^T y + d^T\mu \,:\, (A^T y + E^T v + C^T\mu)_{\mathcal{I}_i} = 0,\; i = 1, \ldots, p\Big\}. \tag{2.13}$$

By Lemma 2.5.1, the dual problem attains an optimal solution $(y_*, \mu_*, v_*) \in \mathbb{R}^m \times \mathbb{R}^\ell_+ \times B_\diamond(0,1)$. Since the objective function of (2.13) is strictly convex in y and convex in (μ, v), y_* is unique (but (μ_*, v_*) may not).

The following lemma establishes a connection between a primal solution and a dual solution, which is critical to distributed algorithm development.

Lemma 2.5.2. *Let (y_*, μ_*, v_*) be an optimal solution to the dual problem (2.13). Then for any optimal solution x_* of the primal problem (2.12), $Ax_* - b = y_*$. Further, if $\mathcal{C}$ is a polyhedral cone (i.e., $d = 0$), then $\|Ex_*\|_\star = -(b + y_*)^T y_*$.*

Proof. Consider the equivalent primal problem for (2.12): $\min_{x \in \mathcal{C}, Ax-b=u} \frac{1}{2}\|u\|_2^2 + \|Ex\|_\star$, and let (x_*, u_*) be its optimal solution. Consider the Lagrangian

$$L(x, u, y, \mu, v) := \frac{\|u\|_2^2}{2} + (Ex)^T v + y^T(Ax - b - u) + \mu^T(Cx - d).$$

In view of the strong duality shown in Lemma 2.5.1, $(x_*, u_*, y_*, \mu_*, v_*)$ is a saddle point of L. Hence,

$$\begin{aligned} L(x_*, u_*, y, \mu, v) &\le L(x_*, u_*, y_*, \mu_*, v_*), \qquad && \forall\, y \in \mathbb{R}^m,\ \mu \in \mathbb{R}^\ell_+,\ v \in B_\diamond(0,1); \\ L(x_*, u_*, y_*, \mu_*, v_*) &\le L(x, u, y_*, \mu_*, v_*), && \forall\, x \in \mathbb{R}^N,\ u \in \mathbb{R}^m. \end{aligned}$$

The former inequality implies that $\nabla_y L(x_*, u_*, y_*, \mu_*, v_*) = 0$ such that $Ax_* - b - u_* = 0$; the latter inequality shows that $\nabla_u L(x_*, u_*, y_*, \mu_*, v_*) = 0$, which yields $u_* - y_* = 0$. These results lead to $Ax_* - b = y_*$. Lastly, when $d = 0$, it follows from the strong duality that $\frac{1}{2}\|Ax_* - b\|_2^2 + \|Ex_*\|_\star = -b^T y_* - \frac{1}{2}\|y_*\|_2^2$. Using $Ax_* - b = y_*$, we have $\|Ex_*\|_\star = -b^T y_* - \|y_*\|_2^2 = -(b + y_*)^T y_*$. □

• **Dual Problem of the BPDN-like Problem** Consider the BPDN-like problem with $\sigma > 0$:

$$\min_{x \in \mathcal{C},\, \|Ax-b\|_2 \le \sigma} \|Ex\|_\star = \inf_{x\in\mathcal{C},\, u=Ax-b,\, \|u\|_2\le\sigma} \|Ex\|_\star, \tag{2.14}$$

where we assume that the problem is feasible and has a positive optimal value, $\|b\|_2 > \sigma$, and the polyhedral set $\mathcal{C}$ satisfies $0 \in \mathcal{C}$. Note that $0 \in \mathcal{C}$ holds if and only if $d \ge 0$.

To establish the strong duality, we also assume that there is an $\widetilde{x}$ in the relative interior of $\mathcal{C}$ (denoted by $\mathrm{ri}(\mathcal{C})$) such that $\|A\widetilde{x} - b\|_2 < \sigma$ or equivalently, by [60, Theorem 6.6], there exits $\widetilde{u} \in A(\mathrm{ri}(\mathcal{C})) - b$ such that $\|\widetilde{u}\|_2 < \sigma$. A sufficient condition for this assumption to hold is that $b \in A(\mathrm{ri}(\mathcal{C}))$. Under this assumption, it follows from [60, Theorem 28.2] that there exist $y_* \in \mathbb{R}^m$, $\mu_* \ge 0$, and $\lambda_* \ge 0$ such that $\inf_{x\in\mathcal{C},\, u=Ax-b, \|u\|_2\le\sigma} \|Ex\|_\star = \inf_{x\in\mathcal{C},u} \|Ex\|_\star + y_*^T(Ax-b-u) + \lambda_*(\|u\|_2^2 - \sigma^2) + \mu_*^T(Cx-d)$. By the similar argument for Lemma 2.5.1, we have

$$\begin{aligned}
\min_{x\in\mathcal{C}, \|Ax-b\|_2\le\sigma} \|Ex\|_\star \;&=\; \inf_{x\in\mathcal{C},\, u=Ax-b, \|u\|_2\le\sigma} \|Ex\|_\star \\
&= \inf_{x\in\mathcal{C},u} \Big(\sup_{y,\, \mu\ge 0,\, \|v\|_\diamond \le 1,\, \lambda\ge 0} \Big\{ (Ex)^T v + \lambda(\|u\|_2^2 - \sigma^2) \\
&\qquad\qquad + y^T(Ax-b-u) + \mu^T(Cx-d) \Big\}\Big) \\
&= \sup_{y,\, \mu\ge 0,\, \|v\|_\diamond\le 1, \lambda\ge 0} \Big(\inf_{x,u} (Ex)^T v + \lambda(\|u\|_2^2 - \sigma^2) + y^T(Ax-b-u) + \mu^T(Cx-d) \Big) \\
&\triangleq \sup_{y,\, \mu\ge 0,\, \|v\|_\diamond\le 1, \lambda> 0} \Big(\inf_{x,u} (Ex)^T v + \lambda(\|u\|_2^2 - \sigma^2) + y^T(Ax-b-u) + \mu^T(Cx-d) \Big) \\
&= \sup_{y,\mu\ge 0, \|v\|_\diamond\le 1, \lambda>0} \Big(- b^T y - \mu^T d - \lambda\sigma^2 + \inf_u \big(\lambda\|u\|_2^2 - y^T u\big) + \\
&\qquad\qquad \sum_{i=1}^p \inf_{x_{\mathcal{I}_i}} \big((A^T y + E^T v + C^T\mu)_{\mathcal{I}_i}\big)^T x_{\mathcal{I}_i} \Big]\Big) \\
&\mathrel{\overset{\circ}{=}} \sup_{y,\mu\ge 0,\|v\|_\diamond\le 1,\lambda>0} \Big\{ - b^T y - \mu^T d - \lambda\sigma^2 - \frac{\|y\|_2^2}{4\lambda} : \\
&\qquad\qquad (A^T y + E^T v + C^T \mu)_{\mathcal{I}_i} = 0,\ i = 1, \ldots, p \Big\}
\end{aligned}$$

Here the reason for letting $\lambda > 0$ in the 4th equation (marked with $\triangleq$) is as follows: suppose $\lambda = 0$, then

$$
\begin{aligned}
&\sup_{y,\mu\geq 0,\|v\|_\diamond\leq 1} \Big\{ \inf_x \Big[(Ex)^T v + y^T(Ax-b) + \mu^T(Cx-d)\Big] + \inf_u y^T(-u)\Big\} \\
&= \sup_{y=0,\,\mu\geq 0,\,\|v\|_\diamond\leq 1} \Big\{ \inf_x \Big[(Ex)^T v + y^T(Ax-b) + \mu^T(Cx-d)\Big] + \inf_u y^T(-u)\Big\} \\
&= \sup_{\mu\geq 0,\,\|v\|_\diamond\leq 1} \Big(\inf_x \Big[(Ex)^T v + \mu^T(Cx-d)\Big]\Big) \\
&\leq \inf_x \Big(\sup_{\mu\geq 0,\,\|v\|_\diamond\leq 1} \Big[(Ex)^T v + \mu^T(Cx-d)\Big]\Big) \;=\; \inf_{x\in\mathcal{C}} \|Ex\|_\star \leq 0,
\end{aligned}
$$

where we use the fact that $0 \in \mathcal{C}$. This shows that the positive optimal value cannot be achieved when $\lambda = 0$, and thus the constraint $\lambda \geq 0$ in the 3rd equation can be replaced by $\lambda > 0$ without loss of generality. Besides, in the second-to-last equation (marked with $\stackrel{\circ}{=}$), the constraint y can be replaced with $y \neq 0$ because otherwise, i.e., $y = 0$, then we have, in light of $\mu \geq 0$ and $d \geq 0$,

$$
\begin{aligned}
&\sup_{y=0,\mu\geq 0,\|v\|_\diamond\leq 1,\lambda>0} \Big\{ -b^T y - \mu^T d - \lambda\sigma^2 - \frac{\|y\|_2^2}{4\lambda} \,: \\
&\qquad\qquad\qquad\qquad (A^T y + E^T v + C^T \mu)_{\mathcal{I}_i} = 0,\ i = 1,\dots,p \Big\} \\
&= \sup_{\mu\geq 0,\,\|v\|_\diamond\leq 1,\lambda>0} \Big\{ -\lambda\sigma^2 - \mu^T d \,:\, (E^T v + C^T \mu)_{\mathcal{I}_i} = 0,\ i = 1,\dots,p \Big\} \;\leq\; 0,
\end{aligned}
$$

which cannot achieve the positive optimal value. Hence, we only consider $y \neq 0$ whose corresponding optimal $\lambda_* = \frac{\|y\|_2}{2\sigma}$ in the second-to-last equation is indeed positive and thus satisfies the constraint $\lambda > 0$. This gives rise to the equivalent dual problem

$$\text{(D)}: \quad \min_{y,\,\mu\geq 0,\,\|v\|_\diamond\leq 1} \left\{ b^T y + \sigma\|y\|_2 + d^T\mu \,:\, (A^T y + E^T v + C^T\mu)_{\mathcal{I}_i} = 0,\ i = 1,\ldots,p \right\}. \tag{2.15}$$

By the similar argument for Lemma 2.5.1, the dual problem attains an optimal solution $(y_*, \mu_*, v_*) \in \mathbb{R}^m \times \mathbb{R}^\ell_+ \times B_\diamond(0,1)$ along with $\lambda_* \geq 0$. The following lemma establishes certain solution properties of the dual problem and a connection between a primal solution and a dual solution, which is crucial to distributed algorithm development. Particularly, it shows that the y-part of a dual solution is unique when $\mathcal{C}$ is a polyhedral cone.

Lemma 2.5.3. *Consider the BPDN (2.14), where $\|b\|_2 > \sigma$, $0 \in \mathcal{C}$, and its optimal value is positive. Assume that the strong duality holds. The following hold:*

(i) Let (y_*, μ_*, v_*) be a dual solution of (2.15). Then $y_* \neq 0$, and for any solution x_* of (2.14), $Ax_* - b = \frac{\sigma y_*}{\|y_*\|_2}$. Further, if $\mathcal{C}$ is a polyhedral cone (i.e., $d = 0$), then $\|Ex_*\|_\star = -b^T y_* - \sigma\|y_*\|_2$.

(ii) Suppose $d = 0$. Let (y_*, μ_*, v_*) and (y'_*, μ'_*, v'_*) be two arbitrary solutions of (2.15). Then $y_* = y'_*$.

Proof. (i) Consider the equivalent primal problem for (2.14): $\min_{x\in\mathcal{C},\, Ax=b=u,\, \|u\|_2\leq\sigma} \|Ex\|_\star$, and let (x_*, u_*) be its optimal solution. For a dual solution (y_*, μ_*, v_*), we deduce that $y_* \neq 0$ since otherwise, we have $-(b^T y_* + \sigma\|y_*\|_2 + d^T\mu_*) \leq 0$, which contradicts its positive optimal value by the strong duality.

Consider the Lagrangian

$$L(x, u, y, \mu, v, \lambda) := (Ex)^T v + y^T(Ax - b - u) + \lambda(\|u\|_2^2 - \sigma^2) + \mu^T(Cx - d).$$

By the strong duality, $(x_*, u_*, y_*, \mu_*, v_*, \lambda_*)$ is a saddle point of L such that

$$
\begin{aligned}
L(x_*, u_*, y, \mu, v, \lambda) &\leq L(x_*, u_*, y_*, \mu_*, v_*, \lambda_*), \\
&\qquad\qquad \forall\ y \in \mathbb{R}^m,\ \mu \in \mathbb{R}^\ell_+,\ v \in B_\diamond(0,1),\ \lambda \in \mathbb{R}_+; \\
L(x_*, u_*, y_*, \mu_*, v_*, \lambda_*) &\leq L(x, u, y_*, \mu_*, v_*, \lambda_*), \quad \forall\ x \in \mathbb{R}^N,\ u \in \mathbb{R}^m.
\end{aligned}
$$

The former inequality implies that $\nabla_y L(x_*, u_*, y_*, \mu_*, v_*, \lambda_*) = 0$, yielding $Ax_* - b - u_* = 0$, and the latter shows that $\nabla_u L(x_*, u_*, y_*, \mu_*, v_*, \lambda_*) = 0$, which gives rise to $2\lambda_* u_* = y_*$. Since $y_* \neq 0$, we have $\lambda_* > 0$ which implies $\|u_*\|_2 - \sigma = 0$ by the complementarity relation. It thus follows from $2\lambda_* u_* = y_*$ and $\|u_*\|_2 = \sigma$ that $\lambda_* = \frac{\|y_*\|_2}{2\sigma}$. This leads to $u_* = \frac{y_*}{2\lambda_*} = \frac{\sigma y_*}{\|y_*\|_2}$. Therefore, $Ax_* - b = u_* = \frac{\sigma y_*}{\|y_*\|_2}$. Finally, when $d = 0$, we deduce via the strong duality that $\|Ex_*\|_\star = -b^T y_* - \sigma \|y_*\|_2$.

(ii) Suppose $d = 0$. Let (y_*, μ_*, v_*) and (y'_*, μ'_*, v'_*) be two solutions of the dual problem (2.15), where $y_* \neq 0$ and $y'_* \neq 0$. Then $b^T y_* + \sigma \|y_*\|_2 = b^T y'_* + \sigma \|y'_*\|_2 = -\|Ex_*\|_\star < 0$. Therefore, $\|y_*\|_2 \big(b^T \frac{y_*}{\|y_*\|_2} + \sigma\big) = \|y'_*\|_2 \big(b^T \frac{y'_*}{\|y'_*\|_2} + \sigma\big)$, and $b^T \frac{y_*}{\|y_*\|_2} + \sigma < 0$. It follows from Proposition 2.2.1 that for any solution x_* of the primal problem (2.14), $Ax_* - b$ is constant. By the argument for Part (i), we have $Ax_* - b = u_*$ and $Ax'_* - b = u'_*$ such that $u_* = u'_*$, and $u_* = \frac{\sigma y_*}{\|y_*\|_2}$ and $u'_* = \frac{\sigma y'_*}{\|y'_*\|_2}$. Hence, $\frac{y_*}{\|y_*\|_2} = \frac{y'_*}{\|y'_*\|_2}$ such that $b^T \frac{y_*}{\|y_*\|_2} + \sigma = b^T \frac{y'_*}{\|y'_*\|_2} + \sigma < 0$. In light of $\|y_*\|_2 \big(b^T \frac{y_*}{\|y_*\|_2} + \sigma\big) = \|y'_*\|_2 \big(b^T \frac{y'_*}{\|y'_*\|_2} + \sigma\big)$, we have $\|y_*\|_2 = \|y'_*\|_2$. Using $\frac{y_*}{\|y_*\|_2} = \frac{y'_*}{\|y'_*\|_2}$ again, we obtain $y_* = y'_*$. □

Remark 2.5.1. The above dual problem formulations for a general polyhedral set $\mathcal{C}$ are useful for distributed computation when $\ell \ll N$, even if $C \in \mathbb{R}^{\ell \times N}$ is a dense matrix; see Section 2.6. When both N and ℓ are large, e.g., $\mathcal{C} = \mathbb{R}^N_+$, decoupling properties of $\mathcal{C}$ are preferred. In particular, consider the following polyhedral set of certain decoupling structure:

$$\mathcal{C} \;:=\; \{x = (x_{\mathcal{I}_i})_{i=1}^p \in \mathbb{R}^N \,|\, C_{\mathcal{I}_i}\, x_{\mathcal{I}_i} \le d_{\mathcal{I}_i}, \quad i = 1, \ldots, p\,\}, \tag{2.16}$$

where $C_{\mathcal{I}_i} \in \mathbb{R}^{\ell_i \times |\mathcal{I}_i|}$ and $d_{\mathcal{I}_i} \in \mathbb{R}^{\ell_i}$ for each $i = 1, \ldots, p$. Let $\ell := \sum_{i=1}^p \ell_i$. Also, let $\mu = (\mu_{\mathcal{I}_i})_{i=1}^p$ with $\mu_{\mathcal{I}_i} \in \mathbb{R}_+^{\ell_i}$ be the Lagrange multiplier for $\mathcal{C}$. The dual problems in (2.11), (2.13), and (2.15) can be easily extended to the above set $\mathcal{C}$ by replacing $\mu^T d$ with $\sum_{i=1}^p \mu_{\mathcal{I}_i}^T d_{\mathcal{I}_i}$ and $(A^T y + E^T v + C^T \mu)_{\mathcal{I}_i}$ with $(A^T y + E^T v)_{\mathcal{I}_i} + C_{\mathcal{I}_i}^T \mu_{\mathcal{I}_i}$, respectively. For example, the dual problem of the regularized problem (2.10) is:

$$(\mathrm{D}): \qquad \min_{y,\, \mu \ge 0,\, \|v\|_\diamond \le 1} \Big(b^T y + \sum_{i=1}^p \mu_{\mathcal{I}_i}^T d_{\mathcal{I}_i} + \frac{1}{2\alpha} \sum_{i=1}^p \big\| (A^T y + E^T v)_{\mathcal{I}_i} + C_{\mathcal{I}_i}^T \mu_{\mathcal{I}_i} \big\|_2^2 \Big).$$

Moreover, letting (y_*, μ_*, v_*) be a dual solution, the unique primal solution $x^* = (x^*_{\mathcal{I}_i})_{i=1}^p$ is given by $x^*_{\mathcal{I}_i} \;=\; -\frac{1}{\alpha}\Big[\big(A^T y_* + E^T v_*\big)_{\mathcal{I}_i} + C_{\mathcal{I}_i}^T (\mu_*)_{\mathcal{I}_i}\Big], \forall\, i = 1, \ldots, p$. Further, Lemmas 2.5.2 and 2.5.3 also hold for a primal solution x_* and a dual solution y_*.

- **Reduced Dual Problems for Box Constraints** Consider the box constraint set $\mathcal{C} := [l_1, u_1] \times \cdots \times [l_N, u_N]$, where $-\infty \le l_i < u_i \le +\infty$ for each $i = 1, \ldots, N$. We assume $0 \in \mathcal{C}$ or equivalently $l_i \le 0 \le u_i$ for each i, which often holds for sparse signal recovery. We may write $\mathcal{C} = \{x \in \mathbb{R}^N \,|\, \mathbf{l} \le x \le \mathbf{u}\}$, where $\mathbf{l} := (l_1, \ldots, l_N)^T$ and $\mathbf{u} := (u_1, \ldots, u_N)^T$. The dual problems for such $\mathcal{C}$ can be reduced by removing the dual variable μ as shown below.

For any given $l_i \le 0 \le u_i$ with $l_i < u_i$ for $i = 1, \ldots, N$, define the function $\theta_i : \mathbb{R} \to \mathbb{R}$ as

$$\theta_i(t) \;:=\; t^2 - (t - \Pi_{[l_i, u_i]}(t))^2 = t^2 - \big(\min(t - l_i,\, (u_i - t)_-)\big)^2, \qquad \forall\, t \in \mathbb{R}. \tag{2.17}$$

Hence, θ_i is C^1 and convex [17, Theorem 1.5.5, Exercise 2.9.13], and θ_i is increasing on $\mathbb{R}_+$ and decreasing on $\mathbb{R}_-$, and its minimal value on $\mathbb{R}$ is zero. When $\mathcal{C} = \mathbb{R}^N$, $\theta_i(s) = s^2, \forall\, i$; when $\mathcal{C} = \mathbb{R}^N_+$, $\theta_i(s) = (s_+)^2, \forall\, i$.

Define the index sets $\mathcal{L}_\infty := \{i \,|\, l_i = -\infty,\ u_i = +\infty\}$, $\mathcal{L}_+ := \{i \,|\, l_i \text{ is finite},\ u_i = +\infty\}$, $\mathcal{L}_- := \{i \,|\, l_i = -\infty,\ u_i \text{ is finite}\}$, and $\mathcal{L}_b := \{1, \ldots, N\} \setminus (\mathcal{L}_\infty \cup \mathcal{L}_+ \cup \mathcal{L}_-)$. Further, define the polyhedral cone

$$\mathcal{K} := \big\{(y, v) \in \mathbb{R}^m \times \mathbb{R}^r \mid (A^T y + E^T v)_{\mathcal{L}_\infty} = 0,\ (A^T y + E^T v)_{\mathcal{L}_+} \geq 0,\ (A^T y + E^T v)_{\mathcal{L}_-} \leq 0\big\},$$

and the extended real valued convex PA function

$$\begin{aligned} g(y, v) \quad := \quad & \sum_{i \in \mathcal{L}_+} (-l_i) \cdot \big[(A^T y + E^T v)_i\big]_+ + \sum_{i \in \mathcal{L}_-} u_i \cdot \big[(A^T y + E^T v)_i\big]_- \\ & + \sum_{i \in \mathcal{L}_b} \Big\{(-l_i) \cdot \big[(A^T y + E^T v)_i\big]_+ + u_i \cdot \big[(A^T y + E^T v)_i\big]_-\Big\}, \quad \forall\ (y, v) \in \mathcal{K}, \end{aligned}$$

and $g(y, v) := +\infty$ for each $(y, v) \notin \mathcal{K}$. Note that $g(y, v) \geq 0, \forall\, (y, v) \in \mathcal{K}$. When the box constraint set $\mathcal{C}$ is a cone, then $\mathcal{K} = \{(y, v) \,|\, A^T y + E^T v \in \mathcal{C}^*\}$ (where $\mathcal{C}^*$ is the dual cone of $\mathcal{C}$), and the corresponding $g(y, v) = 0$ for all $(y, v) \in \mathcal{K}$. Using these results, we obtain the following reduced dual problems:

(i) The dual of the regularized BP-like problem (2.10):

$$\min_{y, \|v\|_\diamond \leq 1} b^T y + \frac{\alpha}{2} \sum_{i=1}^N \theta_i\big(-\frac{1}{\alpha}\big(A^T y + E^T v\big)_i\big).$$

(ii) The dual of the LASSO-like problem (2.12):

$$\min_{\|v\|_\diamond \leq 1,\, (y,v) \in \mathcal{K}} \big(b^T y + \frac{\|y\|_2^2}{2} + g(y, v)\big).$$

(iii) Under similar assumptions, the dual of the BPDN-like problem (2.14):

$$\min_{\|v\|_\diamond \le 1,\,(y,v)\in\mathcal{K}} \big(b^T y + \sigma\|y\|_2 + g(y,v)\big).$$

The dual problems developed in this subsection can be further reduced or simplified for specific norms or polyhedral constraints. This will be shown in the subsequent subsections.

2.5.2 Applications to the ℓ_1-norm based Problems

Let $\|\cdot\|_\star$ be the ℓ_1-norm; its dual norm is the ℓ_∞-norm. As before, $\mathcal{C}$ is a general polyhedral set defined by $Cx \le d$ unless otherwise stated.

• **Reduced Dual Problem of the Regularized BP-like Problem** Consider two cases as follows:

Case (a): $E = I_N$. The dual variable v in (2.11) can be removed using the soft thresholding or shrinkage operator $S_\kappa : \mathbb{R} \to \mathbb{R}$ with the parameter $\kappa > 0$ given by

$$S_\kappa(s) := \arg\min_{t\in\mathbb{R}} \frac{1}{2}(t-s)^2 + \kappa|t| = \begin{cases} s-\kappa, & \text{if } s \ge \kappa \\ 0, & \text{if } s \in [-\kappa,\kappa] \\ s+\kappa, & \text{if } s \le -\kappa \end{cases}$$

When $\kappa = 1$, we write $S_\kappa(\cdot)$ as $S(\cdot)$ for notational convenience. It is known that $S^2(\cdot)$ is convex and C^1. Further, for a vector $v = (v_1,\ldots,v_k)^T \in \mathbb{R}^k$, we let $S(v) := (S(v_1),\ldots,S(v_k))^T \in \mathbb{R}^n$. In view of $\min_{|t|\le 1}(t-s)^2 = S^2(s), \forall\, s \in \mathbb{R}$ whose optimal solution is given by $t_* = \Pi_{[-1,1]}(s)$, the dual problem (2.11) reduces to

$$\text{(D)}: \quad \min_{y,\,\mu\geq 0} \left(b^T y + \mu^T d + \frac{1}{2\alpha} \sum_{i=1}^{p} \| S\big(-(A^T y + C^T \mu)_{\mathcal{I}_i} \big) \|_2^2 \right). \tag{2.18}$$

Letting (y_*, μ_*) be an optimal solution of the above reduced dual problem, it can be shown via the strong duality that for each $\mathcal{I}_i$, $(v_*)_{\mathcal{I}_i} = \psi\big(-(A^T y_* + C^T \mu_*)_{\mathcal{I}_i} \big)$, where $\psi(v) := (\Pi_{[-1,1]}(v_1), \ldots, \Pi_{[-1,1]}(v_k))$ for $v \in \mathbb{R}^k$. Thus the unique primal solution x^* is given by: $\forall\, i = 1, \ldots, p$,

$$x^*_{\mathcal{I}_i} = -\frac{1}{\alpha}\Big[(A^T y_* + C^T \mu_*)_{\mathcal{I}_i} + \psi\big(-(A^T y_* + C^T \mu_*)_{\mathcal{I}_i} \big) \Big] = -\frac{1}{\alpha} S\Big((A^T y_* + C^T \mu_*)_{\mathcal{I}_i} \Big). \tag{2.19}$$

When $\mathcal{C}$ is a box constraint set, the equivalent dual problem further reduces to

$$\text{(D)}: \quad \min_{y\in\mathbb{R}^m} \left[b^T y + \frac{\alpha}{2} \sum_{i=1}^{N} \theta_i \circ \Big(-\frac{1}{\alpha} S\Big(\big(A^T y\big)_i \Big) \Big) \right]. \tag{2.20}$$

Letting y_* be a dual solution, the unique primal solution x^* is given by

$$x^*_i \;=\; \max \left\{ l_i,\, \min\Big(-\frac{1}{\alpha} S\big((A^T y_*)_i \big),\, u_i \Big) \right\}, \qquad \forall\, i = 1, \ldots, N.$$

Case (b): $E = \begin{bmatrix} I_N \\ F \end{bmatrix}$ for some matrix $F \in \mathbb{R}^{k\times N}$. Such an E appears in the ℓ_1 penalty of the fused LASSO. Let $v = (v', \widetilde{v})$. Noting that $\|v\|_\infty \leq 1 \Leftrightarrow \|v'\|_\infty \leq 1, \|\widetilde{v}\|_\infty \leq 1$, and $E^T v = v' + F^T \widetilde{v}$, we have

$$\text{(D)}: \quad \min_{y,\,\mu\geq 0,\,\|\widetilde{v}\|_\infty \leq 1} \left(b^T y + \mu^T d + \frac{1}{2\alpha} \sum_{i=1}^{p} \big\| S\big(-(A^T y + F^T \widetilde{v} + C^T \mu)_{\mathcal{I}_i} \big) \big\|_2^2 \right). \tag{2.21}$$

Letting $(y_*, \mu_*, \widetilde{v}_*)$ be an optimal solution of the above reduced dual problem, it can be shown via the similar argument for Case (a) that the unique primal solution x^* is given by

$$x^*_{\mathcal{I}_i} \;=\; -\frac{1}{\alpha} S\Big((A^T y_* + F^T \widetilde{v}_* + C^T \mu_*)_{\mathcal{I}_i} \Big), \quad \forall\, i = 1, \ldots, p. \tag{2.22}$$

Similarly, when $\mathcal{C}$ is a box constraint set, the equivalent dual problem further reduces to

$$(\mathrm{D}): \quad \min_{y \in \mathbb{R}^m,\, \|\widetilde{v}\|_\infty \le 1} \Big[b^T y + \frac{\alpha}{2} \sum_{i=1}^{N} \theta_i \circ \Big(-\frac{1}{\alpha} S\Big(\big(A^T y + F^T \widetilde{v}\big)_i \Big) \Big) \Big],$$

and the primal solution x^* is expressed in term of a dual solution $(y_*, \widetilde{v}_*)$ as

$$x^*_i \;=\; \max \Big\{ l_i,\, \min \Big(-\frac{1}{\alpha} S\big((A^T y_* + F^T \widetilde{v}_*)_i \big),\, u_i \Big) \Big\}, \quad \forall\, i = 1, \ldots, N.$$

• **Reduced Dual Problem of the LASSO-like Problem** Consider the following cases:

Case (a): $E = \lambda I_N$ for a positive constant λ. The dual problem (2.13) reduces to

$$(\mathrm{D}): \quad \min_{y,\, \mu \ge 0} \Big\{ \frac{\|y\|_2^2}{2} + b^T y + d^T \mu \;:\; \|(A^T y + C^T \mu)_{\mathcal{I}_i}\|_\infty \le \lambda,\; i = 1, \ldots, p \Big\}. \tag{2.23}$$

Particularly, when $\mathcal{C} = \mathbb{R}^N$, it further reduces to $\min_{\|A^T y\|_\infty \le \lambda} \frac{\|y\|_2^2}{2} + b^T y$; when $\mathcal{C} = \mathbb{R}^N_+$, in light of the fact that the inequality $w + v \ge 0$ and $\|v\|_\infty \le 1$ is feasible for a given vector w if and only if $w \ge -\mathbf{1}$, we see that the dual problem (2.13) further reduces to $\min_{A^T y \ge -\lambda \mathbf{1}} \frac{\|y\|_2^2}{2} + b^T y$.

Case (b): $E = \begin{bmatrix} \lambda I_N \\ F \end{bmatrix}$ for some matrix $F \in \mathbb{R}^{k\times N}$ and $\lambda > 0$. Such an E appears in the ℓ_1 penalty of the fused LASSO. The dual problem (2.13) reduces to

$$(\mathrm{D}): \quad \min_{y,\, \mu\geq 0,\, \|\widetilde{v}\|_\infty \leq 1} \left\{ \frac{\|y\|_2^2}{2} + b^T y + d^T \mu \,:\, \|(A^T y + F^T \widetilde{v} + C^T \mu)_{\mathcal{I}_i}\|_\infty \leq \lambda,\ i = 1, \ldots, p \right\}. \tag{2.24}$$

Particularly, when $\mathcal{C} = \mathbb{R}^N$, it further reduces to $\min_{\|A^T y + F^T \widetilde{v}\|_\infty \leq \lambda,\, \|\widetilde{v}\|_\infty \leq 1} \frac{\|y\|_2^2}{2} + b^T y$.

• **Reduced Dual Problem of the BPDN-like Problem** Consider the following cases under the similar assumptions given below (2.14) in Section 2.5.1:

Case (a): $E = I_N$. The equivalent dual problem (2.15) becomes

$$(\mathrm{D}): \quad \min_{y,\, \mu \geq 0} \left\{ b^T y + \sigma \|y\|_2 + d^T \mu \,:\, \|(A^T y + C^T \mu)_{\mathcal{I}_i}\|_\infty \leq 1,\ i = 1, \ldots, p \right\}. \tag{2.25}$$

When $\mathcal{C} = \mathbb{R}^N$, it further reduces to $\min_{\|A^T y\|_\infty \leq 1} b^T y + \sigma \|y\|_2$; when $\mathcal{C} = \mathbb{R}^N_+$, the dual problem (2.13) further reduces to $\min_{A^T y \geq -\mathbf{1}} b^T y + \sigma \|y\|_2$.

Case (b): $E = \begin{bmatrix} I_N \\ F \end{bmatrix}$ for some $F \in \mathbb{R}^{k\times N}$. The equivalent dual problem (2.15) reduces to

$$(\mathrm{D}): \quad \min_{y,\, \mu\geq 0,\, \|\widetilde{v}\|_\infty \leq 1} \left\{ b^T y + \sigma \|y\|_2 + d^T \mu \,:\, \|(A^T y + F^T \widetilde{v} + C^T \mu)_{\mathcal{I}_i}\|_\infty \leq 1,\ i = 1, \ldots, p \right\}. \tag{2.26}$$

Particularly, when $\mathcal{C} = \mathbb{R}^N$, it further reduces to $\min_{\|A^T y + F^T \widetilde{v}\|_\infty \leq 1,\, \|\widetilde{v}\|_\infty \leq 1} b^T y + \sigma \|y\|_2$.

2.5.3 Applications to Problems Associated with the Norm from Group LASSO

Consider the norm $\|x\|_\star := \sum_{i=1}^p \|x_{\mathcal{I}_i}\|_2$ arising from the group LASSO, where its dual norm $\|x\|_\diamond = \max_{i=1,\dots,p} \|x_{\mathcal{I}_i}\|_2$.

• **Reduced Dual Problem of the Regularized BP-like Problem** We consider $E = I_N$ as follows.

Case (a): $\mathcal{C}$ is a general polyhedral set defined by $Cx \le d$. Given a vector w, we see that

$$\min_{\|v\|_\diamond \le 1} \sum_{i=1}^p \left\|(v-w)_{\mathcal{I}_i}\right\|_2^2 = \min_{(\max_{i=1,\dots,p} \|v_{\mathcal{I}_i}\|_2) \le 1} \sum_{i=1}^p \left\|(v-w)_{\mathcal{I}_i}\right\|_2^2 = \sum_{i=1}^p \min_{\|v_{\mathcal{I}_i}\|_2 \le 1} \left\|v_{\mathcal{I}_i} - w_{\mathcal{I}_i}\right\|_2^2.$$

Let $S_{\|\cdot\|_2}(z) := \left(1 - \frac{1}{\|z\|_2}\right)_+ z, \forall z \in \mathbb{R}^n$ denote the soft thresholding operator with respect to the ℓ_2-norm, and let $B_2(0,1) := \{z \,|\, \|z\|_2 \le 1\}$. It is known that given w, $z_* := \Pi_{B_2(0,1)}(w) = w - S_{\|\cdot\|_2}(w)$ and $\|z_* - w\|_2^2 = \|S_{\|\cdot\|_2}(w)\|_2^2 = [(\|w\|_2 - 1)_+]^2$. Applying these results to (2.11), we obtain the reduced dual problem

$$(\mathrm{D}): \quad \min_{y,\, \mu \ge 0} \left(b^T y + \mu^T d + \frac{1}{2\alpha} \sum_{i=1}^p \left[\left(\left\|(A^T y + C^T \mu)_{\mathcal{I}_i}\right\|_2 - 1 \right)_+ \right]^2 \right). \tag{2.27}$$

Letting (y_*, μ_*) be an optimal solution of the problem (D), the primal solution is given by

$$x^*_{\mathcal{I}_i} = -\frac{1}{\alpha} S_{\|\cdot\|_2}\Big((A^T y_* + C^T \mu_*)_{\mathcal{I}_i} \Big), \quad \forall\, i = 1, \dots, p. \tag{2.28}$$

The above results can be easily extended to the decoupled polyhedral constraint set given by (2.16).

Case (b): $\mathcal{C}$ is a box constraint with $0 \in \mathcal{C}$. In this case, the dual variable μ can be removed. In fact, it follows from the results at the end of Section 2.5.1 that the reduced dual problem is

$$\min_{y,\,(v_{\mathcal{I}_i})_{i=1}^p} \sum_{i=1}^p \Big[\tfrac{b^T y}{p} + \tfrac{\alpha}{2} \sum_{j \in \mathcal{I}_i} \theta_j \Big(-\tfrac{1}{\alpha}\big((A_{\bullet \mathcal{I}_i})^T y + v_{\mathcal{I}_i}\big)_j \Big) \Big], \quad \text{subject to } \ \|v_{\mathcal{I}_i}\|_2 \leq 1, \ \ i = 1, \ldots, p, \tag{2.29}$$

where the functions θ_j's are defined in (2.17). Given a dual solution (y_*, v_*), the primal solution $x^*_{\mathcal{I}_i} = \max\big(\mathbf{l}_{\mathcal{I}_i}, \min(-\frac{(A_{\bullet \mathcal{I}_i})^T y_* + (v_*)_{\mathcal{I}_i}}{\alpha}, \mathbf{u}_{\mathcal{I}_i})\big)$ for $i = 1, \ldots, p$. When the box constraint set $\mathcal{C}$ is a cone, the above problem can be further reduced by removing v. For example, when $\mathcal{C} = \mathbb{R}^N$, the reduced dual problem becomes $\min_{y \in \mathbb{R}^m} \big(b^T y + \frac{1}{2\alpha} \sum_{i=1}^p \big[\big(\|(A_{\bullet \mathcal{I}_i})^T y\|_2 - 1\big)_+\big]^2\big)$, and the primal solution x^* is given in term of a dual solution y_* by $x^*_{\mathcal{I}_i} = -\frac{1}{\alpha} S_{\|\cdot\|_2}((A_{\bullet \mathcal{I}_i})^T y_*)$ for $i = 1, \ldots, p$. When $\mathcal{C} = \mathbb{R}^N_+$, the reduced dual problem becomes: $\min_{y \in \mathbb{R}^m} \big(b^T y + \frac{1}{2\alpha} \sum_{i=1}^p \big[\big(\|[(A_{\bullet \mathcal{I}_i})^T y]_-\|_2 - 1\big)_+\big]^2\big)$. Given a dual solution y_*, the unique primal solution x^* is given by:

$$x^*_{\mathcal{I}_i} = \Big[-\frac{1}{\alpha}((A_{\bullet \mathcal{I}_i})^T y_*)_+ + \frac{1}{\alpha} S_{\|\cdot\|_2}\big(((A_{\bullet \mathcal{I}_i})^T y_*)_-\big) \Big]_+, \qquad \forall\, i = 1, \ldots, p.$$

• **Reduced Dual Problem of the LASSO-like Problem** Let $E = \lambda I_N$ for a positive constant λ. For a general polyhedral constraint $Cx \leq d$, the dual problem (2.13) reduces to

$$(\mathrm{D}): \quad \min_{y,\, \mu \geq 0} \Big\{ \frac{\|y\|_2^2}{2} + b^T y + d^T \mu \ : \ \|(A^T y + C^T \mu)_{\mathcal{I}_i}\|_2 \leq \lambda, \ i = 1, \ldots, p \Big\}. \tag{2.30}$$

When $\mathcal{C} = \mathbb{R}^N$, the dual problem becomes $\min_y \big(b^T y + \frac{\|y\|_2^2}{2}\big)$ subject to $\|(A_{\bullet \mathcal{I}_i})^T y\|_2 \leq \lambda, i = 1, \ldots, p$. When $\mathcal{C} = \mathbb{R}^N_+$, the dual problem is $\min_y \big(b^T y + \frac{\|y\|_2^2}{2}\big)$ subject to $\|v\|_\diamond \leq$

$1, A^T y + \lambda v \geq 0$, which is equivalent to $\|v_{\mathcal{I}_i}\|_2 \leq 1, (A^T y)_{\mathcal{I}_i} + v_{\mathcal{I}_i} \geq 0$ for all $i = 1, \ldots, p$. Note that for a given vector $w \in \mathbb{R}^k$, the inequality system $v + w \geq 0, \|v\|_2 \leq 1$ is feasible if and only if $w \in B_2(0,1) + \mathbb{R}^k_+$. Hence, when $\mathcal{C} = \mathbb{R}^N_+$, the dual problem is given by $\min_y \left(b^T y + \frac{\|y\|_2^2}{2}\right)$ subject to $(A^T y)_{\mathcal{I}_i} \in B_2(0, \lambda) + \mathbb{R}^{|\mathcal{I}_i|}_+$ for all $i = 1, \ldots, p$.

• **Reduced Dual Problem of the BPDN-like Problem** Let $E = I_N$. Suppose the similar assumptions indicated in Section 2.5.1 hold. For a general polyhedral set $\mathcal{C}$, the dual problem (2.15) reduces to

$$\text{(D)}: \quad \min_{y,\, \mu \geq 0} \left\{ b^T y + \sigma \|y\|_2 + d^T \mu \; : \; \|(A^T y + C^T \mu)_{\mathcal{I}_i}\|_2 \leq 1, \; i = 1, \ldots, p \right\}. \qquad (2.31)$$

When $\mathcal{C} = \mathbb{R}^N$, the dual problem is $\min_y \left(b^T y + \sigma \|y\|_2\right)$ subject to $\|(A_{\bullet \mathcal{I}_i})^T y\|_2 \leq 1, i = 1, \ldots, p$. When $\mathcal{C} = \mathbb{R}^N_+$, the dual problem is $\min_y \left(b^T y + \sigma \|y\|_2\right)$ subject to $(A^T y)_{\mathcal{I}_i} \in B_2(0,1) + \mathbb{R}^{|\mathcal{I}_i|}_+$ for all $i = 1, \ldots, p$.

2.6 Development of Column Partition based Distributed Algorithms

In this section, we develop column partition based distributed schemes for the LASSO-like problem (2.12) and the BPDN like problem (2.14), which include a board class of convex sparse optimization problems as special cases. As a by-product, column partition based distributed schemes are also developed for the regularized BP-like problem (2.10).

Consider a network of p agents modeled by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, p\}$ is the set of agents, and $\mathcal{E}$ denotes the set of edges, each of which connects two agents in $\mathcal{V}$. For each $i \in \mathcal{V}$, $\mathcal{N}_i$ denotes the set of neighbors of agent i. The following assumptions are made throughout this section:

A.1 The graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is undirected and connected;

A.2 The matrix $A \in \mathbb{R}^{m \times N}$ attains a column partition $\{A_{\bullet \mathcal{I}_i}\}_{i=1}^p$, where $\{\mathcal{I}_1, \ldots, \mathcal{I}_p\}$ is a disjoint union of $\{1, \ldots, N\}$. Each agent i knows $A_{\bullet \mathcal{I}_i}$, b, and possibly other information but does not know $A_{\bullet \mathcal{I}_j}$'s with $j \neq i$.

The assumption **A.2** is motivated by the fact that $m \ll N$ and each agent has limited memory in the large scale problems indicated in Section 2.1. We also consider a general polyhedral set $\mathcal{C}$ given by (2.9) satisfying $\ell \ll N$ or having decoupling structure given by (2.16), e.g., the box constraints.

2.6.1 Structure of Column Partition based Distributed Schemes

We first present a general structure of the proposed column partition based distributed schemes for the LASSO/BPDN-like problems. Recall that these problems are densely coupled but not exactly regularized in general (cf. Section 2.4.1). However, it follows from Proposition 2.2.2 that if Ax_* is known, where x_* is a minimizer of the LASSO/BPDN-like problem, then an exact primal solution can be solved via the dual of a regularized BP-like problem using column partition of A, under exact regularization. To find Ax_*, it follows from Lemmas 2.5.2 and 2.5.3 that $Ax_* = b + y_*$ (resp. $Ax_* = b + \frac{\sigma y_*}{\|y_*\|_2}$), where y_* is a dual solution to the LASSO (resp. BPDN)-like problem. Since the dual of the LASSO/BPDN-like problem can be solved distributively using column partition of A, this yields column partition based two-stage distributed schemes summarized in Algorithm 1. See Section 2.3 for more illustration.

The dual problems used in each stage of Algorithm 1 have been derived in Section 2.5. We will show that these dual problems can be formulated as separable or locally coupled convex optimization problems to which a wide range of existing distributed schemes can be applied. For the purpose of illustration, we consider operator splitting method based schemes including Douglas-Rachford (D-R) algorithm and its variations [15], [26], consensus ADMM (C-ADMM) schemes [47], and inexact C-ADMM (IC-ADMM) schemes [10]. Specific distributed schemes in each stage are given in the next subsections. It should be pointed out that it is *not* our goal to improve the performance of the existing schemes or seek the most efficient existing scheme but rather to demonstrate their applicability to the obtained dual problems. In fact, many other synchronous or asynchronous distributed schemes can be exploited under even weaker assumptions.

Algorithm 1 Two-stage Distributed Algorithm for LASSO/BPDN-like Problem: General Structure

1: Initialization
2: $\boxed{\text{Stage 1}}$ Compute a dual solution y_* to the LASSO-like problem (2.12) or BPDN-like problem (2.14) using a column partition based distributed scheme;
3: $\boxed{\text{Stage 2}}$ Solve the dual of the following regularized BP-like problem for a sufficiently small $\alpha > 0$ using y_* and a column partition based distributed scheme:

$$\text{r-BP}_{\text{LASSO}} : \min_{Ax=b+y_*,\ x\in\mathcal{C}} \|Ex\|_\star + \frac{\alpha}{2}\|x\|_2^2, \tag{2.32}$$

or

$$\text{r-BP}_{\text{BPDN}} : \min_{Ax=b+\frac{\sigma y_*}{\|y_*\|_2},\ x\in\mathcal{C}} \|Ex\|_\star + \frac{\alpha}{2}\|x\|_2^2, \tag{2.33}$$

4: Output: obtain the subvector $x^*_{\mathcal{I}_i}$ from a dual solution to (2.32) or (2.33) for each $i = 1, \ldots, p$

Remark 2.6.1. Before ending this subsection, we discuss a variation of the BP formulation in the second stage for an important special case by exploiting solution properties of (2.12) and (2.14). Consider $E = \lambda I_N$ with $\lambda > 0$ and $\mathcal{C}$ is a polyhedral cone (i.e., $d = 0$), and let y_* be the unique dual solution to (2.12) or (2.14). For (2.12), by Lemma 2.5.2 and $E = \lambda I_N$, we have $Ax_* = b + y_*$ and $\lambda\|x_*\|_1 = -y_*^T(b + y_*)$ for any minimizer x_* of (2.12), noting that $\|x_*\|_1$ is constant on the solution set by Proposition 2.2.1. Suppose $x_* \neq 0$ or equivalently $b + y_* \neq 0$. Then $\|x_*\|_1 = -\frac{1}{\lambda}y_*^T(y_* + b)$, and $\frac{Ax_*}{\|x_*\|_1} = -\frac{\lambda(y_*+b)}{y_*^T(y_*+b)}$. Consider the scaled regularized BP (or scaled r-BP) for $\alpha > 0$:

$$\min_{z\in\mathbb{R}^N} \|z\|_1 + \frac{\alpha}{2}\|z\|_2^2 \ \text{ s.t. } \ Az = -\frac{\lambda(y_* + b)}{y_*^T(y_* + b)}, \ z \in \mathcal{C}. \tag{2.34}$$

Once the unique minimizer z_* of the above r-BP is obtained (satisfying $\|z_*\|_1 = 1$), the least 2-norm minimizer x_* of the LASSO-like problem is given by $x_* = -\frac{1}{\lambda}y_*^T(y_* + b)z_*$. Similarly, for (2.14), by Lemma 2.5.3 and the assumption that the optimal value of (2.14) is positive, we have $-b^Ty_* - \sigma\|y_*\|_2 > 0$. Hence, x_* can be solved from the following scaled r-BP:

$$\min_{z\in\mathbb{R}^N} \ \|z\|_1 + \frac{\alpha}{2}\|z\|_2^2 \ \text{ s.t. } Az = -\frac{b + \frac{\sigma y_*}{\|y_*\|_2}}{b^Ty_* + \sigma\|y_*\|_2}, \ x \in \mathcal{C}. \tag{2.35}$$

Once the unique minimizer z_* is obtained (satisfying $\|z_*\|_1 = 1$), the least 2-norm minimizer x_* of the BPDN-like problem is given by $x_* = -(b^Ty_* + \sigma\|y_*\|_2)z_*$.

The advantages of using the scaled r-BP (2.34) or (2.35) are two folds. First, since $\|x_*\|_1$ may be small or near zero in some applications, a direct application of the r-BP$_{\text{LASSO}}$ or r-BP$_{\text{BPDN}}$ using y_* in Algorithm 1 may be sensitive to round-off errors. Using the scaled r-BP (2.34) or (2.35) can avoid such a problem. Second, the suitable value of α achieving exact regularization is often unknown. A simple rule for choosing such an α is [38]: $\alpha \leq \frac{1}{10\|\widehat{x}\|_\infty}$, where $\widehat{x} \neq 0$ is a sparse vector to be recovered. An estimate of the upper bound of α is $\frac{1}{10\|\widehat{x}\|_1}$ in view of $\|\widehat{x}\|_1 \geq \|\widehat{x}\|_\infty$. When the scaled r-BP (2.34) or (2.35) is used, we can simply choose $\alpha \leq \frac{1}{10}$ as $\|z_*\|_1 = 1$.

2.6.2 Column Partition based Distributed Schemes for the Standard LASSO-like Problem

Consider the standard LASSO-like problem, i.e., the LASSO-like problem (2.12) with $E = \lambda I_N$ for a constant $\lambda > 0$, $\|\cdot\|_\star = \|\cdot\|_1$, and a general polyhedral set $\mathcal{C}$ given by (2.9).

Stage One. We solve the dual problem (2.23), i.e.,

$$\min_{y,\mu\geq 0} \Big\{ \frac{\|y\|_2^2}{2} + b^T y + d^T \mu : \|(A^T y + C^T \mu)_{\mathcal{I}_i}\|_\infty \leq \lambda, \forall i \Big\}.$$

Consider $\ell \ll N$ first. Let $\mathbf{y} := (\mathbf{y}_1, \ldots, \mathbf{y}_p) \in \mathbb{R}^{mp}$ and $\boldsymbol{\mu} := (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_p) \in \mathbb{R}^{\ell p}$. Define the consensus subspace $\mathcal{A}_{\mathbf{y}}$ and the consensus cone $\mathcal{A}_{\boldsymbol{\mu}}$:

$$\mathcal{A}_{\mathbf{y}} \ := \ \{\mathbf{y} \,|\, \mathbf{y}_i = \mathbf{y}_j, \, \forall\, (i,j) \in \mathcal{E}\}, \tag{2.36}$$

$$\mathcal{A}_{\boldsymbol{\mu}} \ := \ \{\boldsymbol{\mu} \geq 0 \,|\, \boldsymbol{\mu}_i = \boldsymbol{\mu}_j, \, \forall\, (i,j) \in \mathcal{E}\}. \tag{2.37}$$

Hence, the dual problem (2.23) is equivalent to the consensus convex optimization problem:

$$\min_{(\mathbf{y},\boldsymbol{\mu})\in\mathcal{A}_{\mathbf{y}}\times\mathcal{A}_{\boldsymbol{\mu}}} \sum_{i=1}^p J_i(\mathbf{y}_i, \boldsymbol{\mu}_i), \text{ s.t.} \, (\mathbf{y}_i, \boldsymbol{\mu}_i) \in \mathcal{W}_i, \quad \forall\, i, \tag{2.38}$$

where for each $i = 1, \ldots, p$, the function

$$J_i(\mathbf{y}_i, \boldsymbol{\mu}_i) \ := \ \frac{1}{p}\Big(\frac{\|\mathbf{y}_i\|_2^2}{2} + b^T \mathbf{y}_i + d^T \boldsymbol{\mu}_i \Big), \tag{2.39}$$

and the set $\mathcal{W}_i := \{(\mathbf{y}_i, \boldsymbol{\mu}_i) \,|\, \|(A_{\bullet\mathcal{I}_i})^T \mathbf{y}_i + (C_{\bullet\mathcal{I}_i})^T \boldsymbol{\mu}_i\|_\infty \leq \lambda\}$. To present distributed schemes for (2.38), the following notation is used: let $\mathbf{w} := (\mathbf{y}, \boldsymbol{\mu}) \in \mathbb{R}^{mp} \times \mathbb{R}^{\ell p}$ and $\mathbf{w}_i :=$

$(\mathbf{y}_i, \boldsymbol{\mu}_i) \in \mathbb{R}^m \times \mathbb{R}^\ell$ for each i. Let $\mathbf{w}^k := (\mathbf{y}^k, \boldsymbol{\mu}^k) \in \mathbb{R}^{mp} \times \mathbb{R}^{\ell p}$, $\mathbf{z}^k = (\mathbf{z}^k_{\mathbf{y}}, \mathbf{z}^k_{\boldsymbol{\mu}}) \in \mathbb{R}^{mp} \times \mathbb{R}^{\ell p}$, $\mathbf{w}^k_i = (\mathbf{y}^k_i, \boldsymbol{\mu}^k_i) \in \mathbb{R}^m \times \mathbb{R}^\ell$, and $\mathbf{z}^k_i = ((\mathbf{z}^k_{\mathbf{y}})_i, (\mathbf{z}^k_{\boldsymbol{\mu}})_i) \in \mathbb{R}^m \times \mathbb{R}^\ell$ for each $i = 1, \ldots, p$.

1.1) *Distributed Averaging based Operator Splitting Scheme.* Given $\mathbf{y} = (\mathbf{y}_i)_{i=1}^p$, let $\overline{\mathbf{y}} := \mathbf{1} \otimes [\frac{1}{p} \sum_{i=1}^p \mathbf{y}_i]$ denote the averaging of $\mathbf{y}$. Similarly, $\overline{\boldsymbol{\mu}}$ denotes the averaging of $\boldsymbol{\mu} = (\boldsymbol{\mu}_i)_{i=1}^p$. Such averaging can be computed via a distributed averaging scheme, e.g., [90]. A distributed averaging based operator splitting scheme [15] is shown in Algorithm 2.

Algorithm 2 Distributed averaging based operator splitting scheme for solving (2.38)

1: Initialization with suitable constants $\eta > 0$ and $\varrho > 0$
2: **repeat**
3: Compute $\widetilde{\mathbf{w}}^{k+1} = \big(\overline{\mathbf{z}^k_{\mathbf{y}}}, (\overline{\mathbf{z}^k_{\boldsymbol{\mu}}})_+\big)$ via a distributed averaging scheme
4: $\mathbf{z}^{k+1}_i = \mathbf{z}^k_i + \varrho\Big[\Pi_{\mathcal{W}_i}\Big(2\widetilde{\mathbf{w}}^{k+1}_i - \mathbf{z}^k_i - \eta \nabla J_i\big((\widetilde{\mathbf{w}}^{k+1})_i\big)\Big) - \widetilde{\mathbf{w}}^{k+1}_i\Big]$ for each $i = 1, \ldots, p$
5: $k \leftarrow k + 1$
6: **until** Stopping criterion is met
7: Output: obtain a dual solution $y_* = (\overline{\mathbf{z}^k_{\mathbf{y}}})_i$ for each i.

1.2) *Consensus-ADMM (C-ADMM) Scheme.* A distributed consensus-ADMM scheme is given in Algorithm 3.

Algorithm 3 Distributed C-ADMM scheme for solving (2.38)

1: Initialization with a suitable constant $\eta > 0$
2: **repeat**
3: $\mathbf{p}^{k+1}_i = \mathbf{p}^k_i + \eta \sum_{j \in \mathcal{N}_i} (\mathbf{w}^k_i - \mathbf{w}^k_j)$ for $i = 1, \ldots, p$
4: $\mathbf{w}^{k+1}_i = \arg\min_{\mathbf{w}_i = (\mathbf{y}_i, \boldsymbol{\mu}_i)} J_i(\mathbf{w}_i) + \mathbf{w}_i^T \mathbf{p}^{k+1}_i + \eta \sum_{j \in \mathcal{N}_i} \big\|\mathbf{w}_i - \frac{\mathbf{w}^k_i + \mathbf{w}^k_j}{2}\big\|_2^2$ subject to $\mathbf{w}_i = (\mathbf{y}_i, \boldsymbol{\mu}_i) \in \mathcal{W}_i$, $\boldsymbol{\mu}_i \geq 0$ for each $i = 1, \ldots, p$
5: $k \leftarrow k + 1$
6: **until** Stopping criterion is met
7: Output: obtain a dual solution $y_* = \mathbf{y}^k_i$ for each i.

1.3) *Local Averaging based Douglas-Rachford (D-R) Scheme.* Recall $\mathbf{w}_i = (\mathbf{y}_i, \boldsymbol{\mu}_i)$ for each i. For each $i = 1, \ldots, p$, define $\mathbf{x}_i := (\mathbf{w}_i, (\mathbf{w}_{ij})_{j \in \mathcal{N}_i})$ with $\mathbf{w}_{ij}$ denoting local copies of $\mathbf{w}_j$'s for agent i [26], the set $\mathcal{A}^{\mathrm{L}}_i := \{\mathbf{x}_i \,|\, \boldsymbol{\mu}_i \geq 0, \, \mathbf{w}_i = \mathbf{w}_{ij}, \, \forall j \in \mathcal{N}_i\}$, and $\widehat{J}_i(\mathbf{x}_i) :=$

$J_i(\mathbf{w}_i)+\boldsymbol{\delta}_{\mathcal{A}_i^{\mathrm{L}}}(\mathbf{x}_i)+\boldsymbol{\delta}_{\mathcal{W}_i}(\mathbf{w}_i)$, where $\boldsymbol{\delta}$ denotes the indicator function. Similarly, define $\mathbf{u}_i := (\mathbf{w}'_i, (\mathbf{w}'_{ij})_{j\in\mathcal{N}_i})$ for each i. Given $\mathbf{u} = (\mathbf{u}_i)_{i=1}^p$, we also define $\overline{\mathbf{u}}_i^{\mathrm{LA}} = (\overline{\mathbf{w}'}_i^{\mathrm{LA}}, (\overline{\mathbf{w}'}_{ij}^{\mathrm{LA}})_{j\in\mathcal{N}_i})$, where $\overline{\mathbf{w}'}_i^{\mathrm{LA}} = \overline{\mathbf{w}'}_{ji}^{\mathrm{LA}} = \frac{1}{|\mathcal{N}_i|+1}\big(\mathbf{w}'_i + \sum_{s\in\mathcal{N}_i}\mathbf{w}'_{si}\big)$ for all $j \in \mathcal{N}_i$ denotes the local averaging [26]. This leads to Algorithm 4.

Algorithm 4 Local averaging based Douglas-Rachford (D-R) scheme for solving (2.38)

1: Initialization with suitable constants $\eta \in (0, 1)$ and $\rho > 0$
2: **repeat**
3: $\quad \mathbf{x}_i^{k+1} = \overline{\mathbf{u}^k}_i^{\mathrm{LA}}, \forall\, i = 1, \dots, p$
4: $\quad \mathbf{u}_i^{k+1} = \mathbf{u}_i^k + 2\eta\big(\mathrm{prox}_{\rho\widehat{J}_i}(2\mathbf{x}_i^{k+1} - \mathbf{u}_i^k) - \mathbf{x}_i^{k+1}\big), \forall\, i$
5: $\quad k \leftarrow k + 1$
6: **until** Stopping criterion is met
7: Output: obtain a dual solution $y_* = (\overline{\mathbf{u}^k})_i$ for each i.

Remark 2.6.2. The values of the parameters for convergence of the above schemes are as follows: (i) $0 < \eta < \frac{2}{L}$ and $0 < \varrho < 2 - \frac{L\eta}{2}$ for Algorithm 2, where $L > 0$ is the Lipschtiz constant of $\sum_{I=1}^p \nabla J_i$ [15]; (ii) $\eta > 0$ for the C-ADMM scheme (Algorithm 3) [47]; and (iii) $\eta \in (0, 1)$ and $\rho > 0$ for the local averaging based D-R scheme (Algorithm 4). Similar values can be found for the subsequent schemes.

We then consider the case where ℓ is large and $\mathcal{C} \in \mathbb{R}^{\ell\times N}$ is given by (2.16), i.e. $\mathcal{C} := \{x = (x_{\mathcal{I}_i})_{i=1}^p \in \mathbb{R}^N \,|\, C_{\mathcal{L}_i\mathcal{I}_i}\, x_{\mathcal{I}_i} \le d_{\mathcal{L}_i}, \forall\, i\}$. Recall that $\mu := (\mu_{\mathcal{L}_i})_{i=1}^p \in \mathbb{R}^\ell$ with $\mu_{\mathcal{L}_i} \in \mathbb{R}^{\ell_i}$. By Remark 2.5.1 and Section 2.5.2, the dual problem (2.23) is equivalent to the consensus convex optimization problem:

$$\min_{(\mathbf{y},\mu)\in\mathcal{A}_{\mathbf{y}}\times\mathbb{R}^\ell_+} \sum_{i=1}^p F_i(\mathbf{y}_i, \mu_{\mathcal{L}_i}), \text{ s.t. } (\mathbf{y}_i, \mu_{\mathcal{L}_i}) \in \mathcal{U}_i, \ \forall\, i, \tag{2.40}$$

where, fore each $i = 1, \dots, p$, the function

$$F_i(\mathbf{y}_i, \mu_{\mathcal{L}_i}) := \frac{1}{p}\Big(\frac{\|\mathbf{y}_i\|_2^2}{2} + b^T\mathbf{y}_i\Big) + d_{\mathcal{L}_i}^T\mu_{\mathcal{L}_i}, \tag{2.41}$$

and $\mathcal{U}_i := \{(\mathbf{y}_i, \mu_{\mathcal{L}_i}) \,|\, \|(A_{\bullet \mathcal{I}_i})^T \mathbf{y}_i + (C_{\mathcal{L}_i \mathcal{I}_i})^T \mu_{\mathcal{L}_i}\|_\infty \leq \lambda\}$.

1.4) *Distributed Averaging based Operator Splitting Scheme for (2.40).* Let $\mathbf{w} := (\mathbf{y}, \mu)$, $\mathbf{w}_i := (\mathbf{y}_i, \mu_{\mathcal{L}_i})$, and $\mathbf{z} = (\mathbf{z}_\mathbf{y}, \mathbf{z}_\mu)$. The following operator splitting scheme can be used for suitable constants $\eta > 0$ and $\varrho > 0$ [15]:

$$\widehat{\mathbf{w}}^{k+1} = \big(\overline{\mathbf{z}^k_\mathbf{y}}, (\mathbf{z}^k_\mu)_+\big) \tag{2.42a}$$

$$\widehat{\mathbf{u}}^{k+1}_i = \Pi_{\mathcal{U}_i}\Big(2\widehat{\mathbf{w}}^{k+1}_i - \mathbf{z}^k_i - \eta \nabla F_i\big((\widehat{\mathbf{w}}^{k+1})_i\big)\Big), \forall\, i \tag{2.42b}$$

$$\mathbf{z}^{k+1}_i = \mathbf{z}^k_i + \varrho\big(\widehat{\mathbf{u}}^{k+1}_i - \widehat{\mathbf{w}}^{k+1}_i\big), \ \forall\, i = 1, \ldots, p \tag{2.42c}$$

where in (2.42a), $\overline{\mathbf{z}^k_\mathbf{y}}$ is computed via a distributed averaging scheme, and the projection $(\mathbf{z}^k_\mu)_+$ is easily implemented distributively. The output is a dual solution $y_* = (\overline{\mathbf{z}^k_\mathbf{y}})_i, \forall\, i$.

1.5) *C-ADMM Scheme for (2.40).* To develop a distributed C-ADMM scheme, it is easy to see that for any fixed $\mathbf{y}_i$, $h_i(\mathbf{y}) := \min_{\mu_{\mathcal{L}_i} \geq 0} \sum_{i=1}^p F_i(\mathbf{y}_i, \mu_{\mathcal{L}_i})$ subject to $(\mathbf{y}_i, \mu_{\mathcal{L}_i}) \in \mathcal{U}_i$ is a real-valued convex function. Thus the sub-gradient of h_i always exists, yielding Algorithm (2.43) for some $\eta > 0$.

$$\mathbf{q}^{k+1}_i = \mathbf{q}^k_i + \eta \sum_{j \in \mathcal{N}_i} (\mathbf{y}^k_i - \mathbf{y}^k_j), \ \forall\ i \tag{2.43a}$$

$$\begin{aligned}(\mathbf{y}^{k+1}_i, \mu^{k+1}_{\mathcal{L}_i}) = \underset{(\mathbf{y}_i, \mu_{\mathcal{L}_i}) \in \mathcal{U}_i,\, \mu_{\mathcal{L}_i} \geq 0}{\arg\min} \ & F_i(\mathbf{y}_i, \mu_{\mathcal{L}_i}) + \ \mathbf{y}_i^T \mathbf{q}^{(k+1)}_i \\ & + \eta \sum_{j \in \mathcal{N}_i} \Big\| \mathbf{y}_i - \frac{\mathbf{y}^k_i + \mathbf{y}^{(k)}_j}{2} \Big\|_2^2, \ \forall\ i\end{aligned} \tag{2.43b}$$

The output is a dual solution $y_* = \mathbf{y}^k_i$ for each i.

Stage Two. The 2nd stage is defined by the regularized BP-like problem (2.10) with the regularization parameter $\alpha > 0$ and b replaced by $b + y_*$. Further, $\lambda = 1$ such that $E = I_N$. When $\|\cdot\|_\star$ is the ℓ_1-norm, Corollary 2.4.2 shows that exact regularization holds, i.e., the regularized problem attains a solution to the original BP-like problem for all small $\alpha > 0$.

Consider $\ell \ll N$ first. Using the consensus subspace $\mathcal{A}_{\mathbf{y}}$ in (2.36) and the consensus cone $\mathcal{A}_{\boldsymbol{\mu}}$ in (2.37), the reduced dual problem (2.18) becomes the consensus convex optimization:

$$\min_{(\mathbf{y},\boldsymbol{\mu})\in\mathcal{A}_{\mathbf{y}}\times\mathcal{A}_{\boldsymbol{\mu}}} \sum_{i=1}^{p} G_i(\mathbf{y}_i,\boldsymbol{\mu}_i), \tag{2.44}$$

where for each $i = 1, \ldots, p$, the function

$$G_i(\mathbf{y}_i,\boldsymbol{\mu}_i) := \frac{1}{p}\Big(b^T\mathbf{y}_i + d^T\boldsymbol{\mu}_i\Big) + \frac{1}{2\alpha}\big\|S\big(-(A_{\bullet\mathcal{I}_i})^T\mathbf{y}_i - (C_{\bullet\mathcal{I}_i})^T\boldsymbol{\mu}_i\big)\big\|_2^2. \tag{2.45}$$

The same notation introduced below (2.39) is used, e.g., $\mathbf{w}, \mathbf{w}_i, \mathbf{w}^k, \mathbf{w}_i^k, \mathbf{z}^k, \mathbf{z}_i^k$ for each $i = 1, \ldots, p$. Further, given $\mathbf{y} = (\mathbf{y}_i)_{i=1}^p$ and $\boldsymbol{\mu} = (\boldsymbol{\mu}_i)_{i=1}^p$, let $\overline{\mathbf{y}}$ and $\overline{\boldsymbol{\mu}}$ be the averaging of $\mathbf{y}$ and $\boldsymbol{\mu}$, respectively.

2.1) *Distributed Averaging based D-R Scheme.* Algorithm 5 presents a distributed averaging based D-R scheme.

Algorithm 5 Distributed averaging based D-R scheme for (2.44)

1: Initialization with suitable constants $\eta \in (0,1)$ and $\rho > 0$
2: **repeat**
3: Compute $\mathbf{w}^{k+1} = \big(\overline{\mathbf{z}_{\mathbf{y}}^k}, (\overline{\mathbf{z}_{\boldsymbol{\mu}}^k})_+\big)$ via a distributed averaging scheme
4: $\mathbf{z}_i^{k+1} = \mathbf{z}_i^k + 2\eta\Big(\operatorname{prox}_{\rho G_i}\big(2\mathbf{w}_i^{k+1} - \mathbf{z}_i^k\big) - \mathbf{w}_i^{k+1}\Big), \forall\, i$
5: $k \leftarrow k+1$
6: **until** Stopping criterion is met
7: Output: obtain $(y_*, \mu_*) = ((\overline{\mathbf{z}_{\mathbf{y}}^k})_i, (\overline{\mathbf{z}_{\boldsymbol{\mu}}^k})_+)_i)$ for each i.

2.2) *Distributed C-ADMM Scheme.* A distributed C-ADMM scheme for solving (2.44) is given below.

Algorithm 6 Distributed C-ADMM scheme for solving (2.44)

1: Initialization with a suitable constant $\eta > 0$
2: **repeat**
3: $\quad \mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \eta \sum_{j \in \mathcal{N}_i} (\mathbf{w}_i^k - \mathbf{w}_j^k)$ for $i = 1, \ldots, p$
4: $\quad \mathbf{w}_i^{k+1} = \arg\min_{\mathbf{w}_i = (\mathbf{y}_i, \boldsymbol{\mu}_i)} G_i(\mathbf{w}_i) + \mathbf{w}_i^T \mathbf{p}_i^{k+1} + \eta \sum_{j \in \mathcal{N}_i} \big\| \mathbf{w}_i - \frac{\mathbf{w}_i^k + \mathbf{w}_j^k}{2} \big\|_2^2$ s.t. $\boldsymbol{\mu}_i \geq 0$ for each i
5: $\quad k \leftarrow k + 1$
6: **until** Stopping criterion is met
7: Output: obtain$(y_*, \mu_*) = (\mathbf{w}^k)_i$ for each i.

In Algorithms 5 and 6, once a dual solution (y_*, μ_*) is found, it follows from (2.19) that the primal solution x^* is given by $x^*_{\mathcal{I}_i} = -\frac{1}{\alpha} S\big((A_{\bullet \mathcal{I}_i})^T y_* + (C_{\bullet \mathcal{I}_i})^T \mu_*\big)$ for $i = 1, \ldots, p$.

Remark 2.6.3. Since the function $G_i(\cdot)$ given by (2.45) involves the soft thresholding operator S, it may be difficult to solve the subproblem in Line 4 of the above C-ADMM scheme. In practice, we formulate this subproblem as: $\mathbf{w}_i^{k+1} = (\mathbf{y}_i^*, \boldsymbol{\mu}_i^*)$, where $\mathbf{w}_i = (\mathbf{y}_i, \boldsymbol{\mu}_i)$ and $(\mathbf{y}_i^*, \boldsymbol{\mu}_i^*, v^*_{\mathcal{I}_i}) = \arg\min_{(\mathbf{w}_i, v_{\mathcal{I}_i})} \frac{1}{p}\Big(b^T \mathbf{y}_i + d^T \boldsymbol{\mu}_i\Big) + \frac{1}{2\alpha} \big\| (A_{\bullet \mathcal{I}_i})^T \mathbf{y}_i + (C_{\bullet \mathcal{I}_i})^T \boldsymbol{\mu}_i + v_{\mathcal{I}_i} \big\|_2^2 + \mathbf{w}_i^T \mathbf{p}_i^{k+1} + \eta \sum_{j \in \mathcal{N}_i} \big\| \mathbf{w}_i - \frac{\mathbf{w}_i^k + \mathbf{w}_j^k}{2} \big\|_2^2$ subject to $\boldsymbol{\mu}_i \geq 0$ and $\|v_{\mathcal{I}_i}\|_\infty \leq 1$, for each i. This new subproblem can be efficiently solved via a quadratic program. Besides, the subproblem in Line 4 of Algorithm 5 can be solved in a similar way.

Remark 2.6.4. Another scheme for solving (2.44) is the distributed averaging based operator splitting scheme [15]:

$$\boxed{\begin{aligned} &\mathbf{w}^{k+1} = \big(\overline{\mathbf{z}^k_{\mathbf{y}}}, (\overline{\mathbf{z}^k_{\boldsymbol{\mu}}})_+\big), \\ &\mathbf{z}_i^{k+1} = \mathbf{z}_i^k + \varrho \Big[\mathbf{w}_i^{k+1} - \mathbf{z}_i^k - \eta \nabla G_i\big(\mathbf{w}_i^{k+1}\big) \Big], \ \forall\, i, \end{aligned}} \qquad \begin{aligned} &(2.46a) \\ &(2.46b) \end{aligned}$$

where (2.46a) is solved via distributed averaging, and ∇G_i is easy to compute. When $\mathcal{C}$ is a box constraint with $0 \in \mathcal{C}$, the dual problem can be reduced to (2.20) depending on y only, and a distributed scheme similar to (2.46) can be developed. A drawback of (2.46) is that the Lipschitz constant of $\sum_i \nabla G_i$ is given by $(\|A\|_F^2 + \|C\|_F^2)/\alpha$, which is large for a large N. This yields a small $\eta > 0$ and thus slow convergence. Nonetheless, the scheme (2.46) can be used for a small or moderate N.

We then consider a large ℓ with $\mathcal{C}$ given by (2.16). It follows from Remark 2.5.1 and Section 2.5.2 that the equivalent dual problem is given by: recalling that $\mu := (\mu_{\mathcal{L}_i})_{i=1}^p \in \mathbb{R}^\ell$,

$$\min_{(\mathbf{y},\mu)\in\mathcal{A}_{\mathbf{y}}\times\mathbb{R}^\ell_+} \sum_{i=1}^{p} \widetilde{F}_i(\mathbf{y}_i, \mu_{\mathcal{L}_i}),$$

where $\widetilde{F}_i(\mathbf{y}_i, \mu_{\mathcal{L}_i}) := \frac{1}{2\alpha}\big\|S(-(A_{\bullet\mathcal{I}_i})^T\mathbf{y}_i - (C_{\mathcal{L}_i\mathcal{I}_i})^T\mu_{\mathcal{L}_i})\big\|_2^2 + \frac{1}{p}(b^T\mathbf{y}_i) + d^T_{\mathcal{L}_i}\mu_{\mathcal{L}_i}$ for $i = 1, \ldots, p$. Let $\mathbf{w} := (\mathbf{y}, \mu)$ and $\mathbf{w}_i := (\mathbf{y}_i, \mu_{\mathcal{L}_i})$, and $\mathbf{z} = (\mathbf{z}_{\mathbf{y}}, \mathbf{z}_\mu)$. Distributed schemes similar to Algorithms 5-6 can be developed by replacing G_i with $\widetilde{F}_i$.

2.6.3 Column Partition based Distributed Schemes for the Standard BDPN-like Problem

Consider the standard BPDN-like problem, i.e., the BPDN-like problem (2.14) with $E = I_N$, $\|\cdot\|_\star = \|\cdot\|_1$, and a polyhedral set $\mathcal{C}$ given by (2.9). Suppose the assumptions given below (2.14) in Section 2.5.1 hold. Consider the dual problem (2.15). As shown in Lemma 2.5.3, a dual solution $y_* \neq 0$. Hence, the function $\|y\|_2$ is differentiable near y_*.

Stage One. Consider $\ell \ll N$ first. In light of (2.25), it is easy to verify that the distributed Algorithms 2-4 can be applied by replacing the functions J_i in (2.39) with

$$\widehat{J}_i(\mathbf{y}_i, \boldsymbol{\mu}_i) := \frac{1}{p}\Big(\sigma\|\mathbf{y}_i\|_2 + b^T\mathbf{y}_i + d^T\boldsymbol{\mu}_i\Big), \quad \forall\ i = 1, \ldots, p,$$

and setting $\lambda = 1$ in $\mathcal{W}_i$. Moreover, an inexact C-ADMM scheme [10] can be applied; its details are omitted. When ℓ is large and $\mathcal{C}$ is given by (2.16), letting $\mu := (\mu_{\mathcal{L}_i})_{i=1}^p \in \mathbb{R}^\ell$, the schemes in (2.42)-(2.43) can be used by replacing F_i's in (2.41) by $\widehat{F}_i(\mathbf{y}_i, \mu_{\mathcal{L}_i}) := \frac{1}{p}\Big(\sigma\|\mathbf{y}_i\|_2 + b^T\mathbf{y}_i\Big) + d_{\mathcal{L}_i}^T \mu_{\mathcal{L}_i}$ and setting $\lambda = 1$ in $\mathcal{U}_i$. When $\mathcal{C} = \mathbb{R}^N$ or $\mathcal{C} = \mathbb{R}^N_+$, μ or $\boldsymbol{\mu}$ can be removed; see the discussions below (2.25).

Stage Two. The 2nd stage is defined by the regularized BP-like problem (2.10) with the regularization parameter $\alpha > 0$, b replaced by $b + \frac{\sigma y_*}{\|y_*\|_2}$, and $E = I_N$. Hence, all the results for the 2nd stage of the standard LASSO-like problem given in Section 2.6.2 apply.

2.6.4 Column Partition based Distributed Schemes for the Fused LASSO-like and Fused BDPN-like Problems

Through this section, let $\|\cdot\|_\star$ be the ℓ_1-norm, $D_1 \in \mathbb{R}^{(N-1)\times N}$ be the first order difference matrix, and we assume in addition that the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ satisfies $(i, i+1) \in \mathcal{E}, \forall\, i = 1, \ldots, p-1$. Consider the fused LASSO-like problem first, i.e., the LASSO-like problem (2.12) with $E = \begin{bmatrix} \lambda I_N \\ \gamma D_1 \end{bmatrix}$ for positive constants λ and γ and a general polyhedral set $\mathcal{C}$ as before.

Stage One. Consider $\ell \ll N$ first. To solve the dual problem (2.24) with $F = \gamma D_1$ and $\widetilde{v} = (\widetilde{v}_1, \ldots, \widetilde{v}_{N-1}) \in \mathbb{R}^{N-1}$, define $n_s := \sum_{i=1}^s |\mathcal{I}_i|$ for $s = 1, \ldots, p$. Without loss of generality, let $\mathcal{I}_1 = \{1, \ldots, n_1\}$, and $\mathcal{I}_{i+1} = \{n_i + 1, \ldots, n_i + |\mathcal{I}_{i+1}|\}$ for each $i = 1, \ldots, p-1$. Define the index sets $\mathcal{S}_1 := \mathcal{I}_1$, $\mathcal{S}_i := \{n_{i-1}\} \cup \mathcal{I}_i$ for $i = 2, \ldots, p-1$, and $\mathcal{S}_p := \{n_{p-1}, \ldots, N-1\}$. Define $r_i := |\mathcal{S}_i|$ and $\mathbf{v}_i := \widetilde{v}_{\mathcal{S}_i}$ for each $i = 1, \ldots, p$. Thus for $i = 1, \ldots, p-1$, $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ overlap on one variable $\widetilde{v}_{n_i}$. Let $\mathbf{v} := (\mathbf{v}_i)_{i=1}^p \in \mathbb{R}^{N+p-2}$, and the local coupling constraint $\mathcal{A}_{LC} := \{\mathbf{v} \in \mathbb{R}^{N+p-2} \,|\, (\mathbf{v}_i)_{r_i} = (\mathbf{v}_{i+1})_1, \ \forall\, i = 1, \ldots, p-1\}$.

For each $i = 1, \ldots, p$, define the function

$$H_i(\mathbf{y}_i, \boldsymbol{\mu}_i, \mathbf{v}_i) \;:=\; \frac{1}{p}\Big(\frac{\|\mathbf{y}_i\|_2^2}{2} + b^T \mathbf{y}_i + d^T \boldsymbol{\mu}_i \Big), \tag{2.47}$$

and the set $\mathcal{V}_i := \{ (\mathbf{y}_i, \boldsymbol{\mu}_i, \mathbf{v}_i) \,|\, \|\mathbf{v}_i\|_\infty \le 1, \ \|(A_{\bullet \mathcal{I}_i})^T \mathbf{y}_i + (C_{\bullet \mathcal{I}_i})^T \boldsymbol{\mu}_i + \gamma [(D_1)_{\mathcal{S}_i \mathcal{I}_i}]^T \mathbf{v}_i \|_\infty \le \lambda \}$. The dual problem (2.24) is formulated as the locally coupled convex program:

$$\min_{(\mathbf{y}, \boldsymbol{\mu}, \mathbf{v}) \in \mathcal{A}_{\mathbf{y}} \times \mathcal{A}_{\boldsymbol{\mu}} \times \mathcal{A}_{LC}} \sum_{i=1}^{p} H_i(\mathbf{y}_i, \boldsymbol{\mu}_i, \mathbf{v}_i), \tag{2.48}$$

subject to $(\mathbf{y}_i, \boldsymbol{\mu}_i, \mathbf{v}_i) \in \mathcal{V}_i, \ \forall\, i = 1, \ldots, p$.

Let $\mathbf{z}^k = (\mathbf{z}^k_{\mathbf{y}}, \mathbf{z}^k_{\boldsymbol{\mu}}, \mathbf{z}^k_{\mathbf{v}}) \in \mathbb{R}^{mp} \times \mathbb{R}^{\ell p} \times \mathbb{R}^{N+p-2}$, and η, ρ are suitable positive constants depending on the Lipschitz constant of $\sum_{i=1}^p \nabla H_i$; see [15, Thoerem 1] for details. For any $\mathbf{v} = (\mathbf{v}_i)_{i=1}^p \in \mathbb{R}^{N+p-2}$ defined above, $\widetilde{\mathbf{v}} = (\widetilde{\mathbf{v}}_i)_{i=1}^p := \Pi_{\mathcal{A}_{LC}}(\mathbf{v})$ is $(\widetilde{\mathbf{v}}_i)_{r_i} = (\widetilde{\mathbf{v}}_{i+1})_1 = \frac{1}{2}[(\mathbf{v}_i)_{r_i} + (\mathbf{v}_{i+1})_1]$ for $i = 1, \ldots, p-1$, and for each i, $(\widetilde{\mathbf{v}}_i)_j = (\mathbf{v}_i)_j$ for the other indices j. Clearly, this local averaging can be computed distributively. A distributed averaging based operator splitting scheme is given in Algorithm 7. A local averaging based operating splitting scheme can be developed in a similar way. These schemes can be extended to a large ℓ with $\mathcal{C}$ given by (2.16) and be extended to the generalized total variation denoising or ℓ_1-trend filtering with $E = \lambda D_1$ or $E = \lambda D_2$.

Algorithm 7 Distributed averaging based operator splitting scheme for solving (2.48)

1: Initialization with suitable constants $\eta > 0$ and $\varrho > 0$
2: **repeat**
3: Compute $\widehat{\mathbf{w}}^{k+1} = \big(\overline{\mathbf{z}^k_{\mathbf{y}}}, (\overline{\mathbf{z}^k_{\boldsymbol{\mu}}})_+, \widetilde{\mathbf{z}^k_{\mathbf{v}}}\big)$, where $\overline{\mathbf{z}^k_{\mathbf{y}}}$ and $\overline{\mathbf{z}^k_{\boldsymbol{\mu}}}$ are solved via a distributed averaging scheme, and $\widetilde{\mathbf{z}^k_{\mathbf{v}}} = ((\widetilde{\mathbf{z}^k_{\mathbf{v}}})_i)_{i=1}^p$ with $(\widetilde{\mathbf{z}^k_{\mathbf{v}}})_i$ computed distributively
4: $\mathbf{z}_i^{k+1} = \mathbf{z}_i^k + \varrho \Big[\Pi_{\mathcal{V}_i}\Big(2\widehat{\mathbf{w}}_i^{k+1} - \mathbf{z}_i^k - \eta \nabla H_i\big(\widehat{\mathbf{w}}_i^{k+1}\big)\Big) - \widehat{\mathbf{w}}_i^{k+1} \Big]$ for each $i = 1, \ldots, p$
5: $k \leftarrow k + 1$
6: **until** Stopping criterion is met
7: Output: obtain a dual solution $y_* = (\overline{\mathbf{z}^k_{\mathbf{y}}})_i$ for each i.

Stage Two. The 2nd stage is given by the regularized BP-like problem (2.10) with the parameter $\alpha > 0$, b replaced by $b + y_*$, and $E = \begin{bmatrix} I_N \\ \\ \gamma D_1 \end{bmatrix}$ for a constant $\gamma > 0$ after scaling. Consider a general polyhedral set $\mathcal{C}$ with $\ell \ll N$ first.

We follow the same notation used in the first stage. Hence, the reduce dual problem (2.21) can be formulated as the following locally coupled convex program:

$$\min_{(\mathbf{y}, \boldsymbol{\mu}, \mathbf{v}) \in \mathcal{A}_{\mathbf{y}} \times \mathcal{A}_{\boldsymbol{\mu}} \times \mathcal{A}_{LC}} \sum_{i=1}^{p} P_i(\mathbf{y}_i, \boldsymbol{\mu}_i, \mathbf{v}_i), \tag{2.49}$$

where, for each $i = 1, \ldots, p$, the function

$$P_i(\mathbf{y}_i, \boldsymbol{\mu}_i, \mathbf{v}_i) := \frac{1}{p}\Big(b^T \mathbf{y}_i + d^T \boldsymbol{\mu}_i\Big) + \frac{1}{2\alpha}\big\|S\big(-(A_{\bullet \mathcal{I}_i})^T \mathbf{y}_i - (C_{\bullet \mathcal{I}_i})^T \boldsymbol{\mu}_i - \gamma[(D_1)_{\mathcal{S}_i \mathcal{I}_i}]^T \mathbf{v}_i\big)\big\|_2^2.$$

Let $\mathbf{z}^k = (\mathbf{z}^k_{\mathbf{y}}, \mathbf{z}^k_{\boldsymbol{\mu}}, \mathbf{z}^k_{\mathbf{v}}) \in \mathbb{R}^{mp} \times \mathbb{R}^{\ell p} \times \mathbb{R}^{N+p-2}$, $\mathbf{w} := (\mathbf{y}, \boldsymbol{\mu}, \mathbf{v}) \in \mathbb{R}^{mp} \times \mathbb{R}^{\ell p} \times \mathbb{R}^{N+p-2}$ and $\mathbf{w}_i := (\mathbf{y}_i, \boldsymbol{\mu}_i, \mathbf{v}_i) \in \mathbb{R}^m \times \mathbb{R}^\ell \times \mathbb{R}^{r_i}$ for each i. This leads to Algorithm 8 below.

Algorithm 8 Distributed averaging based D-R scheme for (2.49)

1: Initialization with suitable constants $\eta > 0$ and $\rho > 0$
2: **repeat**
3: Compute $\widehat{\mathbf{w}}^{k+1} = \big(\overline{\mathbf{z}^k_{\mathbf{y}}}, (\overline{\mathbf{z}^k_{\boldsymbol{\mu}}})_+, \widetilde{\mathbf{z}^k_{\mathbf{v}}}\big)$, where $\overline{\mathbf{z}^k_{\mathbf{y}}}$ and $\overline{\mathbf{z}^k_{\boldsymbol{\mu}}}$ are solved via a distributed averaging scheme, and $\widetilde{\mathbf{z}^k_{\mathbf{v}}} = ((\widetilde{\mathbf{z}^k_{\mathbf{v}}})_i)_{i=1}^p$ with $(\widetilde{\mathbf{z}^k_{\mathbf{v}}})_i$ computed distributively
4: $\mathbf{z}^{k+1}_i = \mathbf{z}^k_i + 2\eta\Big(\operatorname{prox}_{\rho P_i}\big(2\widehat{\mathbf{w}}^{k+1}_i - \mathbf{z}^k_i\big) - \widehat{\mathbf{w}}^{k+1}_i\Big)$, $\forall\, i$
5: $k \leftarrow k + 1$
6: **until** Stopping criterion is met
7: Output: obtain a dual solution $(y_*, \mu_*, \widetilde{v}_*)$ from $\big(\overline{\mathbf{z}^k_{\mathbf{y}}}, (\overline{\mathbf{z}^k_{\boldsymbol{\mu}}})_+, \widetilde{\mathbf{z}^k_{\mathbf{v}}}\big)$

Once a dual solution $(y_*, \mu_*, \widetilde{v}_*)$ is obtained from Algorithm 8, the primal solution $(x^*_{\mathcal{I}_i})_{i=1}^p$ is computed using (2.22). Moreover, to solve the subproblem in Line 4 of Algorithm 8, we apply the similar technique given in Remark 2.6.3 to formulate it as a quadratic program.

Another scheme for solving (2.49) is the distributed averaging based operator splitting scheme, which is suitable for a small or moderate N:

$$\boxed{\begin{aligned} \mathbf{w}^{k+1} &= \big(\overline{\mathbf{z}^k_{\mathbf{y}}}, (\overline{\mathbf{z}^k_{\boldsymbol{\mu}}})_+, \widetilde{\mathbf{z}^k_{\mathbf{v}}}\big) && (2.50a) \\ \mathbf{u}^{k+1}_i &= \Pi_{\mathcal{R}_i}\Big(2\mathbf{w}^{k+1}_i - \mathbf{z}^k_i - \eta \nabla P_i\big(\mathbf{w}^{k+1}_i\big)\Big), \ \forall\, i && (2.50b) \\ \mathbf{z}^{k+1}_i &= \mathbf{z}^k_i + \varrho\big(\mathbf{u}^{k+1}_i - \mathbf{w}^{k+1}_i\big), \ \forall\, i = 1, \ldots, p && (2.50c) \end{aligned}}$$

where for each i, the set $\mathcal{R}_i := \mathbb{R}^m \times \mathbb{R}^\ell \times \{\mathbf{v}_i \,|\, \|\mathbf{v}_i\|_\infty \le 1\}$.

Similar distributed schemes can be developed for the decoupled constraint given by (2.16). Moreover, they can be extended to the generalized total variation denoising and ℓ_1-trend filtering where $E = \lambda D_1$ or $E = \lambda D_2$ with $\lambda > 0$.

Remark 2.6.5. Consider the fused BPDN-like problem, i.e., the BPDN-like problem (2.14) with $E = \begin{bmatrix} I_N \\ \\ \gamma D_1 \end{bmatrix}$ for a constant $\gamma > 0$. Suppose the assumptions given below (2.14) in Section 2.5.1 hold. In the first stage, to solve the dual problem (2.26) with $F = \gamma D_1$ and $\ell \ll N$, define the function $\widehat{H}_i(\mathbf{y}_i, \boldsymbol{\mu}_i, \mathbf{v}_i) := \big(\sigma\|\mathbf{y}_i\|_2 + b^T\mathbf{y}_i + d^T\boldsymbol{\mu}_i\big)/p$. Then Algorithm 7 can be applied by replacing H_i with $\widehat{H}_i$. Similar results can be made for a large ℓ with $\mathcal{C}$ given by (2.16). The second stage is almost identical to that of the fused LASSO-like problem, except that $b + y_*$ is replaced by $b + \frac{\sigma y_*}{\|y_*\|_2}$.

2.6.5 LASSO-like, BPDN-like, and Regularized BP-like Problems with the Norm from the Group LASSO

Consider $\|x\|_\star := \sum_{i=1}^p \|x_{\mathcal{I}_i}\|_2$ from the group LASSO (2.3); its dual norm $\|x\|_\diamond = \max_{i=1,\ldots,p} \|x_{\mathcal{I}_i}\|_2$. Many preceding results for the ℓ_1-norm can be extended to this case.

For illustration, consider the standard LASSO-like problem with $E = \lambda I_N$ for $\lambda > 0$ and a small ℓ. In the first stage, by virtue of the dual problem (2.30), Algorithms 2-4 can be used by replacing the set $\mathcal{W}_i$ with the set $\widehat{\mathcal{W}}_i := \{(\mathbf{y}_i, \boldsymbol{\mu}_i) \,|\, \|(A_{\bullet\mathcal{I}_i})^T\mathbf{y}_i + (C_{\bullet\mathcal{I}_i})^T\boldsymbol{\mu}_i\|_2 \leq \lambda\}$, which has nonempty interior. When ℓ is large and $\mathcal{C} \in \mathbb{R}^{\ell\times N}$ is given by (2.16), the schemes in (2.42) and (2.43) can be used after the same replacement. In the second stage, we assume that exact regularization holds (cf. Section 2.4.2). When $E = I_N$ and $\mathcal{C}$ is a general polyhedral set, the reduced dual problem (2.27) is formulated as the convex consensus optimization problem: $\min_{(\mathbf{y},\boldsymbol{\mu})\in\mathcal{A}_\mathbf{y}\times\mathcal{A}_{\boldsymbol{\mu}}} \sum_{i=1}^p J_i(\mathbf{y}_i, \boldsymbol{\mu}_i)$, where $J_i(\mathbf{y}_i, \boldsymbol{\mu}_i) := (b^T\mathbf{y}_i + d^T\boldsymbol{\mu}_i)/p + \frac{1}{2\alpha}\big[\big(\big\|(A_{\bullet\mathcal{I}_i})^T\mathbf{y}_i + (C_{\bullet\mathcal{I}_i})^T\boldsymbol{\mu}_i\big\|_2 - 1\big)_+\big]^2$ for $i = 1, \ldots, p$, and $\mathcal{A}_\mathbf{y}, \mathcal{A}_{\boldsymbol{\mu}}$ are defined in (2.36)-(2.37). Thus a distributed scheme similar to (2.46) can be applied.

In the second stage, when $\mathcal{C}$ is a box constraint set, consider the reduced dual problem (2.29). By introducing p copies of y's given by $\mathbf{y}_i$ and imposing the consensus condition on $\mathbf{y}_i$'s, this problem can be converted to a convex program of the variable $(\mathbf{y}_i, v_{\mathcal{I}_i})_{i=1}^p$ with a separable objective function and separable constraint sets with nonempty interiors. By Slater's condition, the D-R scheme or operator splitting schemes similar to the scheme (2.46) can be developed. If, in addition, $\mathcal{C}$ is a cone, the dual problems can be further reduced to unconstrained problems of the variable y only, e.g., those for $\mathcal{C} = \mathbb{R}^N$ and $\mathcal{C} = \mathbb{R}^N_+$ given in Case (b) of Section 2.5.3. These problems can be formulated as consensus convex programs and solved by column partition based distributed schemes. Finally, the primal solution $x^*_{\mathcal{I}_i}$ can be computed distributively using a dual solution y_* and the operator $S_{\|\cdot\|_2}$ (cf. Section 2.5.3).

The above results can be easily extended to the standard BPDN-like and fused LASSO/BPDN-like problems; these details are omitted.

2.7 Overall Convergence of the Two-stage Distributed Algorithms

In this section, we analyze the overall convergence of the two-stage distributed algorithms proposed in Section 2.6, assuming that a distributed algorithm in each stage is convergent. To motivate the overall convergence analysis, it is noted that an algorithm of the first-stage generates an approximate solution y^k to the solution y_* of the dual problem, and this raises the question of whether using this approximate solution in the second stage leads to significant discrepancy when solving the second-stage problem (2.32) or (2.33). Inspired by this question and its implication to the overall convergence of the two-stage algorithms, we establishes the continuity of the solution of the regularized BP-like problem (2.10) in b, which is closely related to sensitivity analysis of the problem (2.10). We first present some technical preliminaries.

Lemma 2.7.1. *Let $\|\cdot\|_\star$ be a norm on $\mathbb{R}^n$ and $\|\cdot\|_\diamond$ be its dual norm. Then for any $x \in \mathbb{R}^n$, $\|v\|_\diamond \leq 1$ for any $v \in \partial\|x\|_\star$.*

Proof. Fix $x \in \mathbb{R}^n$, and let $v \in \partial\|x\|_\star$. Hence, $\|y\|_\star \geq \|x\|_\star + \langle v, y-x\rangle$ for all $y \in \mathbb{R}^n$. Since $\|y-x\|_\star \geq \big|\|y\|_\star - \|x\|_\star\big| \geq \langle v, y-x\rangle$, we have $\langle v, \frac{y-x}{\|y-x\|_\star}\rangle \leq 1$ for any $y \neq x$. This shows that $\|v\|_\diamond \leq 1$. $\square$

Another result we will use is concerned with the Lipschitz property of the linear complementarity problem (LCP) under certain singleton property. Specifically, consider the LCP (q, M): $0 \leq u \perp Mu + q \geq 0$ for a given matrix $M \in \mathbb{R}^{n\times n}$ and a vector $q \in \mathbb{R}^n$. Let $\mathrm{SOL}(q, M)$ denote its solution set. The following theorem is an extension of a well-known fact in the LCP and variational inequality theory, e.g., [17, Propositioin 4.2.2], [23, Theorem 10], and [69].

Theorem 2.7.1. *Consider the LCP (q, M). Suppose a matrix $E \in \mathbb{R}^{p\times n}$ and a set $\mathcal{W} \subseteq \mathbb{R}^n$ are such that for any $q \in \mathcal{W}$, $SOL(q, M)$ is nonempty and $ESOL(q, M)$ is singleton. The following hold:*

(i) $ESOL(\cdot, M)$ is locally Lipschitz at each $q \in \mathcal{W}$, i.e., there exist a constant $L_q > 0$ and a neighborhood $\mathcal{N}$ of q such that $\|ESOL(q', M) - ESOL(q, M)\| \le L_q\|q' - q\|$ for any $q' \in \mathcal{N} \cap \mathcal{W}$;

(ii) If $\mathcal{W}$ is a convex set, then $ESOL(\cdot, M)$ is (globally) Lipschitz continuous on $\mathcal{W}$, i.e., there exists a constant $L > 0$ such that $\|ESOL(q, M) - ESOL(q', M)\| \le L\|q - q'\|$ for all $q', q \in \mathcal{W}$.

We apply the above results to the regularized BP-like problem subject to a generic polyhedral constraint, in addition to the linear equality constraint, i.e.,

$$\min_{x \in \mathcal{C},\, Ax=b} \|Ex\|_\star + \frac{\alpha}{2}\|x\|_2^2, \tag{2.51}$$

where α is a positive constant, $E \in \mathbb{R}^{r\times N}$, $A \in \mathbb{R}^{m\times N}$, the polyhedral set $\mathcal{C} := \{x \in \mathbb{R}^N \,|\, Cx \le d\}$ for some $C \in \mathbb{R}^{\ell\times N}$ and $d \in \mathbb{R}^\ell$, and $b \in \mathbb{R}^m$ with $b \in A\mathcal{C} := \{Ax \,|\, x \in \mathcal{C}\}$. We shall show that its unique optimal solution is continuous in b, where we assume that $A \neq 0$ without loss of generality. To achieve this goal, consider the necessary and sufficient optimality condition for the unique solution x_* of (2.51), namely, there exist (possibly non-unique) multipliers $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^\ell_+$ such that

$$0 \in E^T\partial\|Ex_*\|_\star + \alpha\, x_* + A^T\lambda + C^T\mu, \quad Ax_* = b, \quad 0 \le \mu \perp Cx_* - d \le 0. \tag{2.52}$$

When we need to emphasize the dependence of x_* on b, we write it as $x_*(b)$ in the following development. For a given $b \in A\mathcal{C}$ and its corresponding unique minimizer x_* of (2.51), define the set

$$\mathcal{S}(x_*) := \Big\{(w, \lambda, \mu) \,\big|\, w \in \partial\|Ex_*\|_\star, \ E^T w + \alpha x_* + A^T\lambda + C^T\mu = 0, \ 0 \le \mu \perp Cx_* - d \le 0\Big\}.$$

This set contains all the sub-gradients w and the multipliers λ, μ satisfying the optimality condition at x_*, and it is often unbounded due to possible unboundeness of λ and μ (noting that by Lemma 2.7.1, w's are bounded). To overcome this difficulty in continuity analysis, we present the following proposition.

Proposition 2.7.1. *The following hold for the minimization problem (2.51):*

(i) Let $\mathcal{B}$ be a bounded set in $\mathbb{R}^m$. Then $\{x_*(b) \,|\, b \in A\mathcal{C} \cap \mathcal{B}\}$ is a bounded set;

(ii) Let (b^k) be a convergent sequence in $A\mathcal{C}\cap\mathcal{B}$. Then there exist a constant $\gamma > 0$ and an index subsequence (k_s) such that for each k_s, there exists $(w^{k_s}, \lambda^{k_s}, \mu^{k_s}) \in \mathcal{S}(x_*(b^{k_s}))$ satisfying $\|(\lambda^{k_s}, \mu^{k_s})\| \le \gamma$.

Proof. (i) Suppose $\{x_*(b) \,|\, b \in A\mathcal{C} \cap \mathcal{B}\}$ is unbounded. Then there exists a sequence (b^k) in $A\mathcal{C} \cap \mathcal{B}$ such that the sequence $\big(x_*(b^k)\big)$ satisfies $\|x_*(b^k)\| \to \infty$. For notational simplicity, we let $x_*^k := x_*(b^k)$ for each k. Without loss of generality, we assume that $\big(\frac{x_*^k}{\|x_*^k\|}\big)$ converges to $v_* \neq 0$. In view of $A\frac{x_*^k}{\|x_*^k\|} = \frac{b^k}{\|x_*^k\|}$, $C\frac{x_*^k}{\|x_*^k\|} \le \frac{d}{\|x_*^k\|}$, and the fact that (b^k) is bounded, we have $Av_* = 0$ and $Cv_* \le 0$. Further, for each k, there exist $\lambda^k \in \mathbb{R}^m$ and $\mu^k \in \mathbb{R}^\ell_+$ and $w^k \in \partial\|Ex_*^k\|_\star$ such that $E^T w^k + \alpha x_*^k + A^T\lambda^k + C^T\mu^k = 0$, $Ax_*^k = b^k$, and $0 \le \mu^k \perp Cx_*^k - d \le 0$ for each k. We claim that $(Cv_*)^T\mu^k = 0$ for all large k. To prove this claim, we note that, by virtue of $Cv_* \le 0$, that for each index i, either $(Cv_*)_i = 0$ or $(Cv_*)_i < 0$. For the latter, it follows from $\big(C\frac{x_*^k}{\|x_*^k\|} - \frac{d}{\|x_*^k\|}\big)_i \to (Cv_*)_i$ that

$(Cx_*^k - d)_i < 0$ for all large k. Hence, we deduce from the optimality condition (2.52) that $\mu_i^k = 0$ for all large k. This shows that $(Cv_*)_i \cdot \mu_i^k = 0, \forall\, i$ for all large k. Hence, the claim holds. In view of this claim and $Av_* = 0$, we see that left multiplying v_*^T to the equation $E^T w^k + \alpha x_*^k + A^T\lambda^k + C^T\mu^k = 0$ leads to $(Ev_*)^T w^k + \alpha(v_*)^T x_*^k = 0$, or equivalently $(Ev_*)^T \frac{w^k}{\|x_k^*\|} + \alpha(v_*)^T \frac{x_*^k}{\|x_*^k\|} = 0$, for all large k. Since (w^k) is bounded by Lemma 2.7.1, we have, by taking the limit, that $\alpha\|v_*\|_2^2 = 0$, leading to $v_* = 0$, a contradiction. Hence, $\{x_*(b) \,|\, b \in A\mathcal{C} \cap \mathcal{B}\}$ is bounded.

(ii) Given a convergent sequence (b^k) in $A\mathcal{C}$, we use $x_*^k := x_*(b^k)$ for each k again. Consider a sequence $\big((w^k, \lambda^k, \mu^k)\big)$, where $(w^k, \lambda^k, \mu^k) \in \mathcal{S}(x_*^k)$ is arbitrary for each k. In view of the boundedness of (x_*^k) proven in (i) and Lemma 2.7.1, we assume by taking a suitable subsequence that $(w^k, x_*^k) \to (\widehat{w}, \widehat{x})$. Let the index set $\widehat{\mathcal{I}}_\mu := \{i \,|\, (C^T\widehat{x} - d)_i < 0\}$. If there exists an index $i \notin \widehat{\mathcal{I}}_\mu$ such that (μ_i^k) has a zero subsequence $(\mu_i^{k'})$, then let $\widehat{\mathcal{I}}'_\mu := \widehat{\mathcal{I}}_\mu \cup \{i\}$. We then consider the subsequence $(\mu^{k'})$. If there exists an index $j \notin \widehat{\mathcal{I}}'_\mu$ such that $(\mu_j^{k'})$ has a zero subsequence $(\mu_j^{k''})$, then let $\widehat{\mathcal{I}}''_\mu := \widehat{\mathcal{I}}'_\mu \cup \{j\}$ and consider the subsequence $(\mu^{k''})$. Continuing this process in finite steps, we obtain an index subsequence (k_s) and an index set $\mathcal{I}_\mu$ such that $(C^T x_*^{k_s} - d)_{\mathcal{I}_\mu} < 0$ and $\mu^{k_s}_{\mathcal{I}^c_\mu} > 0$ for all k_s's, where $\mathcal{I}^c_\mu := \{1, \ldots, N\} \setminus \mathcal{I}_\mu$. By the complementarity condition in (2.52), we have $\mu^{k_s}_{\mathcal{I}_\mu} = 0$ and $\mu^{k_s}_{\mathcal{I}^c_\mu} > 0$ for each k_s.

Since $A \neq 0$, there exits an index subset $\mathcal{J} \subseteq \{1, \ldots, m\}$ such that the columns of $(A^T)_{\bullet\mathcal{J}}$ (or equivalently $(A_{\mathcal{J}\bullet})^T$) form a basis for $R(A^T)$. Hence, for each λ^{k_s}, there exists a unique vector $\widetilde{\lambda}^{k_s}$ such that $A^T\lambda^{k_s} = (A_{\mathcal{J}\bullet})^T\widetilde{\lambda}^{k_s}$. In view of the equations $E^T w^{k_s} + \alpha x_*^{k_s} + (A_{\mathcal{J}\bullet})^T\widetilde{\lambda}^{k_s} + C^T\mu^{k_s} = 0$ and $A_{\mathcal{J}\bullet} x_*^{k_s} = b_{\mathcal{J}}^{k_s}$, we obtain via a straightforward computation that

$$\begin{aligned}
\widetilde{\lambda}^{k_s} &= -\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}\Big[\alpha b_{\mathcal{J}}^{k_s} + A_{\mathcal{J}\bullet}(E^T w^{k_s} + C^T\mu^{k_s})\Big],\\
x_*^{k_s} &= (A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}b_{\mathcal{J}}^{k_s}\\
&\qquad + \frac{1}{\alpha}\Big[(A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}A_{\mathcal{J}\bullet} - I\Big]\big(E^T w^{k_s} + C^T\mu^{k_s}\big),
\end{aligned} \tag{2.53}$$

where $C^T\mu^{k_s} = (C_{\mathcal{I}_\mu^c\bullet})^T\mu_{\mathcal{I}^c}^{k_s}$ for each k_s in view of $\mu_{\mathcal{I}_\mu^c}^{k_s} > 0$ and $\mu_{\mathcal{I}_\mu}^{k_s} = 0$. Substituting $x_*^{k_s}$ into the complementarity condition $0 \leq \mu^{k_s} \perp d - Cx_*^{k_s} \geq 0$, we deduce that $\mu_{\mathcal{I}_\mu^c}^{k_s}$ satisfies the following conditions:

$$0 \leq \mu_{\mathcal{I}_\mu^c}^{k_s} \perp d_{\mathcal{I}_\mu^c} - C_{\mathcal{I}_\mu^c\bullet}x_*^{k_s} \geq 0, \quad C_{\mathcal{I}_\mu\bullet}x_*^{k_s} - d_{\mathcal{I}_\mu} = H\,\mu_{\mathcal{I}_\mu^c}^{k_s} + h^{k_s} \leq 0,$$

where $d_{\mathcal{I}_\mu^c} - C_{\mathcal{I}_\mu^c\bullet}x_*^{k_s} = G\,\mu_{\mathcal{I}_\mu^c}^{k_s} + g^{k_s}$, and the matrices G, H and the vectors g^{k_s}, h^{k_s} are given by

$$\begin{aligned}
G &:= \frac{1}{\alpha}C_{\mathcal{I}_\mu^c\bullet}\Big[I - (A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}A_{\mathcal{J}\bullet}\Big](C_{\mathcal{I}_\mu^c\bullet})^T,\\
g^{k_s} &:= \frac{1}{\alpha}C_{\mathcal{I}_\mu^c\bullet}\Big[I - (A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}A_{\mathcal{J}\bullet}\Big]E^T w^{k_s}\\
&\qquad\qquad - C_{\mathcal{I}_\mu^c\bullet}(A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}b_{\mathcal{J}}^{k_s} + d_{\mathcal{I}_\mu^c},\\
H &:= \frac{1}{\alpha}C_{\mathcal{I}_\mu\bullet}\Big[I - (A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}A_{\mathcal{J}\bullet}\Big](C_{\mathcal{I}_\mu^c\bullet})^T,\\
h^{k_s} &:= \frac{1}{\alpha}C_{\mathcal{I}_\mu\bullet}\Big[I - (A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}A_{\mathcal{J}\bullet}\Big]E^T w^{k_s}\\
&\qquad\qquad - C_{\mathcal{I}_\mu\bullet}(A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}b_{\mathcal{J}}^{k_s} + d_{\mathcal{I}_\mu}.
\end{aligned}$$

Since $\mu_{\mathcal{I}_\mu^c}^{k_s} > 0$, we must have $G\mu_{\mathcal{I}_\mu^c}^{k_s} + g^{k_s} = 0$. For the matrices G, H and given vectors g, h, define the polyhedral set $\mathcal{K}(G, H, g, h) := \{z \,|\, z \geq 0,\ Gz + g = 0,\ Hz + h \leq 0\} =$

$\{z \mid Dz+v \geq 0\}$, where $D := \begin{bmatrix} I \\ \\ G \\ \\ -G \\ \\ -H \end{bmatrix}$ and $v := \begin{bmatrix} 0 \\ \\ g \\ \\ -g \\ \\ -h \end{bmatrix}$. Hence, for each k_s, $\mathcal{K}(G, H, g^{k_s}, h^{k_s})$ contains the vector $\mu^{k_s}_{\mathcal{I}^c_\mu} > 0$ and thus is nonempty. We write v as v^{k_s} when $(g, h) = (g^{k_s}, h^{k_s})$. Let $\widetilde{z}^{k_s}$ be the least 2-norm point of $\mathcal{K}(G, H, g^{k_s}, h^{k_s})$, i.e., $\widetilde{z}^{k_s}$ is the unique solution to $\min \frac{1}{2}\|z\|_2^2$ subject to $Dz + v^{k_s} \geq 0$. Since its underlying optimization problem has a (feasible) polyhedral constraint, its necessary and sufficient optimality condition is: $\widetilde{z}^{k_s} - D^T\nu = 0$, $0 \leq \nu \perp D\widetilde{z}^{k_s} + v^{k_s} \geq 0$ for some (possibly non-unique) multiplier ν. Let $\mathrm{SOL}(v^{k_s}, DD^T)$ be the solution set of the LCP: $0 \leq \nu \perp v^{k_s} + DD^T\nu \geq 0$. By the uniqueness of $\widetilde{z}^{k_s}$, $\widetilde{z}^{k_s} = D^T\mathrm{SOL}(v^{k_s}, DD^T)$ such that $D^T\mathrm{SOL}(v^{k_s}, DD^T)$ is singleton.

Since g^{k_s} and h^{k_s} are affine functions of (w^{k_s}, b^{k_s}) and the sequences (w^{k_s}) and (b^{k_s}) are convergent, (v^{k_s}) is convergent and we let v^* be its limit. We show as follows that the polyhedral set $\{z \mid Dz + v^* \geq 0\}$ is nonempty. Suppose not. Then it follows from a version of Farkas' lemma [13, Theorem 2.7.8] that there exists $w \geq 0$ such that $D^T w = 0$ and $w^T v^* < 0$. Since $(v^{k_s}) \to v^*$, we see that $w^T v^{k_s} < 0$ for all large k_s. By [13, Theorem 2.7.8] again, we deduce that $Dz + v^{k_s} \geq 0$ has no solution z for all large k_s, yielding a contradiction. This shows that $\{z \mid Dz + v^* \geq 0\}$ is nonempty. Thus $\mathrm{SOL}(v^*, D^T D)$ is nonempty and $D^T\mathrm{SOL}(v^*, DD^T)$ is singleton. Define the function $R(v) := D^T\mathrm{SOL}(v, DD^T)$. By Theorem 2.7.1, $R(\cdot)$ is locally Lipschitz continuous at v^*, i.e., there exist a constant $L_* > 0$ and a neighborhood $\mathcal{V}$ of v^* such that for any $v \in \mathcal{V}$ satisfying that $\{z \mid Dz + v \geq 0\}$ is nonempty, $\|R(v) - R(v^*)\| \leq L_*\|v - v^*\|$. This, along with the convergence of (v^{k_s}) to v^*, show that $\{\widetilde{z}^{k_s} \mid \widetilde{z}^{k_s} = R(v^{k_s}), \forall\, k_s\}$ is bounded. For each k_s, let $\widehat{\mu}^{k_s} := (\widehat{\mu}^{k_s}_{\mathcal{I}_\mu}, \widehat{\mu}^{k_s}_{\mathcal{I}^c_\mu}) = (0, \widetilde{z}^{k_s})$. Hence, $(\widehat{\mu}^{k_s})$ is a bounded sequence. Further, let

$\widetilde{\lambda}^{k_s} := -\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}\big[\alpha b^{k_s}_{\mathcal{J}} + A_{\mathcal{J}\bullet}(E^T w^{k_s} + C^T \widehat{\mu}^{k_s})\big]$, and $\widehat{\lambda}^{k_s} := \big(\widehat{\lambda}^{k_s}_{\mathcal{J}}, \widehat{\lambda}^{k_s}_{\mathcal{J}^c}\big) = (\widetilde{\lambda}^{k_s}, 0)$. This implies that $(\widetilde{\lambda}^{k_s})$, and thus $(\widehat{\lambda}^{k_s})$, is bounded. Hence, $\big((\widehat{\lambda}^{k_s}, \widehat{\mu}^{k_s})\big)$ is a bounded sequence, i.e., there exists $\gamma > 0$ such that $\|(\widehat{\lambda}^{k_s}, \widehat{\mu}^{k_s})\| \le \gamma$ for all k_s.

Lastly, we show that $(w^{k_s}, \widehat{\lambda}^{k_s}, \widehat{\mu}^{k_s}) \in \mathcal{S}(x^{k_s}_*)$ for each k_s. In view of (2.53), define

$$\widehat{x}^{k_s} := (A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1} b^{k_s}_{\mathcal{J}} + \frac{1}{\alpha}\Big[(A_{\mathcal{J}\bullet})^T\big(A_{\mathcal{J}\bullet}(A_{\mathcal{J}\bullet})^T\big)^{-1}A_{\mathcal{J}\bullet} - I\Big]\big(E^T w^{k_s} + C^T\widehat{\mu}^{k_s}\big).$$

Therefore, $A_{\mathcal{J}\bullet}\widehat{x}^{k_s} = b^{k_s}_{\mathcal{J}}$ for each k_s. Since the columns of $(A_{\mathcal{J}\bullet})^T$ form a basis for $R(A^T)$ and $b^{k_s} \in A\mathcal{P}$, we have $A\widehat{x}^{k_s} = b^{k_s}$. Moreover, based on the constructions of $\widehat{\lambda}^{k_s}$ and $\widehat{\mu}^{k_s}$, it is easy to show that $E^T w^{k_s} + \alpha\widehat{x}^{k_s} + A^T\widehat{\lambda}^{k_s} + C^T\widehat{\mu}^{k_s} = 0$, $(C\widehat{x}^{k_s} - d)_{\mathcal{I}_\mu} = H\widetilde{z}^{k_s} + h^{k_s} \le 0$, and $(C\widehat{x}^{k_s} - d)_{\mathcal{I}^c_\mu} = G\widetilde{z}^{k_s} + g^{k_s} = 0$ for each k_s. In light of $\widehat{\mu}^{k_s} = (\widehat{\mu}^{k_s}_{\mathcal{I}_\mu}, \widehat{\mu}^{k_s}_{\mathcal{I}^c_\mu}) = (0, \widetilde{z}^{k_s}) \ge 0$, we have $0 \le \widehat{\mu}^{k_s} \perp C\widehat{x}^{k_s} - d \le 0$ for each k_s. This implies that $(w^{k_s}, \widehat{\lambda}^{k_s}, \widehat{\mu}^{k_s}) \in \mathcal{S}(\widehat{x}^{k_s})$ for each k_s. Since the optimization problem (2.51) has a unique solution for each b^{k_s}, we must have $\widehat{x}^{k_s} = x^{k_s}_*$. This shows that $(w^{k_s}, \widehat{\lambda}^{k_s}, \widehat{\mu}^{k_s}) \in \mathcal{S}(x^{k_s}_*)$ for each k_s. □

With the help of Proposition 2.7.1, we are ready to show the desired continuity.

Theorem 2.7.2. *Let $\alpha > 0$, $E \in \mathbb{R}^{r\times N}$ $A \in \mathbb{R}^{m\times N}$, $\mathcal{C} := \{x \in \mathbb{R}^N \,|\, Cx \le d\}$ for some $C \in \mathbb{R}^{\ell\times N}$ and $d \in \mathbb{R}^\ell$, and $b \in \mathbb{R}^m$ with $b \in A\mathcal{C} := \{Ax \,|\, x \in \mathcal{C}\}$. Then the unique solution x_* of the minimization problem (2.51) is continuous in b on $A\mathcal{C}$.*

Proof. Fix an arbitrary $b \in A\mathcal{C}$. Suppose $x_*(\cdot)$ is discontinuous at this b. Then there exist $\varepsilon_0 > 0$ and a sequence (b^k) in $A\mathcal{C}$ such that (b^k) converges to b but $\|x^k_* - x_*(b)\| \ge \varepsilon_0$ for all k, where $x^k_* := x_*(b^k)$. By Statement (i) of Proposition 2.7.1, (x^k_*) is bounded and hence attains a convergent subsequence which, without loss of generality, can be itself. Let the limit of (x^k_*) be $\widehat{x}$. Further, as shown in Statement (ii) of Proposition 2.7.1, there exists a bounded subsequence $\big((w^{k_s}, \lambda^{k_s}, \mu^{k_s})\big)$ such that $(w^{k_s}, \lambda^{k_s}, \mu^{k_s}) \in \mathcal{S}(x^{k_s}_*)$ for each

k_s. Without loss of generality, we assume that $\big((w^{k_s}, \lambda^{k_s}, \mu^{k_s})\big)$ converges to $(\widehat{w}, \widehat{\lambda}, \widehat{\mu})$. Since $(Ex_*^{k_s}) \to E\widehat{x}$ and $(w^{k_s}) \to \widehat{w}$ with $w^{k_s} \in \partial \|Ex_*^{k_s}\|_\star$ for each k_s, it follows from [6, Proposition B.24(c)] that $\widehat{w} \in \partial \|E\widehat{x}\|_\star$. By taking the limit, we deduce that $(\widehat{x}, \widehat{w}, \widehat{\lambda}, \widehat{\mu})$ satisfies $E^T\widehat{w} + \alpha\widehat{x} + A^T\widehat{\lambda} + C^T\widehat{\mu} = 0$, $A\widehat{x} = b$, and $0 \le \widehat{\mu} \perp C\widehat{x} - d \le 0$, i.e., $(\widehat{w}, \widehat{\lambda}, \widehat{\mu}) \in \mathcal{S}(\widehat{x})$. This shows that $\widehat{x}$ is a solution to (2.51) for the given b. Since this solution is unique, we must have $\widehat{x} = x_*(b)$. Hence, $(x_*^{k_s})$ converges to $x_*(b)$, a contradiction to $\|x_*^{k_s} - x_*(b)\| \ge \varepsilon_0$ for all k_s. This yields the continuity of x_* in b on $A\mathcal{C}$. $\square$

When the norm $\|\cdot\|_\star$ in the objective function of the optimization problem (2.51) is given by the ℓ_1-norm or a convex PA function in general, the continuity property shown in Theorem 2.7.2 can be enhanced. Particularly, the following result establishes the Lipschitz continuity of x_* in b, which is useful in deriving the overall convergence rate of the two-stage distributed algorithm.

Theorem 2.7.3. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex piecewise affine function, $A \in \mathbb{R}^{m\times N}$, $\mathcal{C} := \{x \in \mathbb{R}^N \,|\, Cx \le d\}$ for some $C \in \mathbb{R}^{\ell\times N}$ and $d \in \mathbb{R}^\ell$, and $b \in \mathbb{R}^m$ with $b \in A\mathcal{C} := \{Ax \,|\, x \in \mathcal{C}\}$. Then for any $\alpha > 0$, $\min_{x\in\mathcal{C}} f(x) + \frac{\alpha}{2}\|x\|_2^2$ subject to $Ax = b$ has a unique minimizer x_*. Further, x_* is Lipschitz continuous in b on $A\mathcal{C}$, i.e., there exists a constant $L > 0$ such that $\|x_*(b') - x_*(b)\| \le L\|b' - b\|$ for any $b, b' \in A\mathcal{C}$.*

Proof. We first show the solution existence and uniqueness. Consider a real-valued convex PA function $f(x) = \max_{i=1,\dots,r} \big(p_i^T x + \gamma_i\big)$ for a finite family of $(p_i, \gamma_i) \in \mathbb{R}^N \times \mathbb{R}, i = 1, \dots, r$. Note that for any given $\alpha > 0$ and any nonzero x,

$$f(x) + \frac{\alpha}{2}\|x\|_2^2 \;=\; \|x\|_2^2 \cdot \left[\frac{\alpha}{2} + \max_{i=1,\dots,r}\left(p_i^T \frac{x}{\|x\|_2^2} + \frac{\gamma_i}{\|x\|_2^2}\right)\right].$$

Hence, $f(x) + \frac{\alpha}{2}\|x\|_2^2$ is coercive. Since it is continuous and strictly convex and the constraint set is closed and convex, the underlying optimization problem attains a unique minimizer.

To prove the Lipschitz property of the unique minimizer x_* in b, we consider the following equivalent form of the underlying optimization problem:

$$\min_{t_+, t_-, x} \; t_+ - t_- + \frac{\alpha}{2}\|x\|_2^2 \quad \text{subject to} \;\; t_+ \ge 0,\; t_- \ge 0,\; Ax = b,\; Cx \le d,\; p_i^T x + \gamma_i \le t_+ - t_-,\;\; i = 1, \dots, r. \tag{2.54}$$

Define the matrix $W := \begin{bmatrix} p_1^T \\ \vdots \\ p_r^T \end{bmatrix} \in \mathbb{R}^{r \times N}$ and the vector $\Gamma := \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_r \end{bmatrix} \in \mathbb{R}^r$. Then the constraints can be written as $t_+ \ge 0,\; t_- \ge 0,\; Ax = b,\; Cx \le d$, and $Wx + \Gamma - t_+\mathbf{1} + t_-\mathbf{1} \le 0$, where $\mathbf{1}$ denotes the vector of ones. Given $b \in A\mathcal{C}$, the necessary and sufficient optimality conditions for the minimizer x_* are described by a mixed linear complementarity problem, i.e., there exist Lagrange multipliers $\lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^\ell_+, \nu \in \mathbb{R}^r_+, \theta_+ \in \mathbb{R}_+$ and $\theta_- \in \mathbb{R}_+$ such that

$$\begin{aligned} \alpha x_* + A^T\lambda + C^T\mu + W^T\nu = 0, &\quad Ax_* = b, \\ 1 - \mathbf{1}^T\nu - \theta_+ = 0, &\quad -1 + \mathbf{1}^T\nu - \theta_- = 0, \\ 0 \le \mu \perp Cx_* - d \le 0, &\quad 0 \le \nu \perp Wx_* + \Gamma - t_+\mathbf{1} + t_-\mathbf{1} \le 0, \\ 0 \le \theta_+ \perp t_+ \ge 0, &\quad 0 \le \theta_- \perp t_- \ge 0. \end{aligned}$$

Note that the first and second equations are equivalent to the first equation and $\alpha b + AA^T\lambda + AC^T\mu + AW^T\nu = 0$. Further, it is noticed that $\theta_+ = \theta_- = 0$, and $\lambda = \lambda_+ - \lambda_-$ with $0 \le \lambda_+ \perp \lambda_- \ge 0$. Hence, by adding two slack variables ϑ and φ, the above mixed linear complementarity problem is equivalent to

$$
\begin{aligned}
x_* &= -\frac{1}{\alpha}\Big(A^T\lambda_+ - A^T\lambda_- + C^T\mu + W^T\nu\Big),\\
0 \le \mu \quad &\perp \quad -\frac{C}{\alpha}\Big(A^T\lambda_+ - A^T\lambda_- + C^T\mu + W^T\nu\Big) - d \le 0,\\
0 \le \nu \quad &\perp \quad -\frac{W}{\alpha}\Big(A^T\lambda_+ - A^T\lambda_- + C^T\mu + W^T\nu\Big) + \Gamma - t_+\mathbf{1} + t_-\mathbf{1} \le 0,\\
0 \le t_+ \quad &\perp \quad 1 - \mathbf{1}^T\nu \ge 0,\\
0 \le t_- \quad &\perp \quad -1 + \mathbf{1}^T\nu \ge 0,\\
0 \le \lambda_+ \quad &\perp \quad \lambda_- \ge 0,\\
0 \le \vartheta \quad &\perp \quad \alpha b + AA^T(\lambda_+ - \lambda_-) + AC^T\mu + AW^T\nu \ge 0,\\
0 \le \varphi \quad &\perp \quad \alpha b + AA^T(\lambda_+ - \lambda_-) + AC^T\mu + AW^T\nu \le 0.
\end{aligned}
$$

The latter seven complementarity conditions in the above formulation yield the following linear complementarity problem (LCP): $0 \le u \perp Mu + q \ge 0$, where $u = (\mu, \nu, t_+, t_-, \lambda_+, \lambda_-, \vartheta, \varphi) \in \mathbb{R}^\ell_+ \times \mathbb{R}^r_+ \times \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}^m_+ \times \mathbb{R}^m_+ \times \mathbb{R}^m_+ \times \mathbb{R}^m_+$, M is a constant matrix of order $(\ell + r + 4m + 2)$ that depends on A, C, W, α only, and the vector $q = (d, -\Gamma, 1, 1, 0, 0, \alpha b, -\alpha b) \in \mathbb{R}^\ell \times \mathbb{R}^r \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^m$. Denote this LCP by LCP$(q, M)$. For any given $b \in A\mathcal{C}$, LCP(q, M) attains a solution u which pertains to the Lagrange multipliers λ, μ, ν, t_+, t_- and the slack variables ϑ and φ. This shows that for any given $b \in A\mathcal{C}$, LCP(q, M) has a nonempty solution set SOL(q, M). Further, for any $\widetilde{u} = (\widetilde{\mu}, \widetilde{\nu}, \widetilde{t}_+, \widetilde{t}_-, \widetilde{\lambda}_+, \widetilde{\lambda}_-, \widetilde{\vartheta}, \widetilde{\varphi}) \in \text{SOL}(q, M)$, if follows from the last two complementarity conditions that $\widetilde{x} := -\frac{1}{\alpha}\big(A^T\widetilde{\lambda}_+ - A^T\widetilde{\lambda}_- + C^T\widetilde{\mu} + W^T\widetilde{\nu}\big)$ satisfies $A\widetilde{x} = b$. Besides, $\widetilde{\lambda} := \widetilde{\lambda}_+ - \widetilde{\lambda}_-$, $\widetilde{\mu}, \widetilde{\nu}$, and $\widetilde{\theta}_+ = \widetilde{\theta}_- = 0$ satisfy the optimality conditions of the underlying optimization problem (2.54) at $(\widetilde{t}_+, \widetilde{t}_-, \widetilde{x})$ for the given $b \in A\mathcal{C}$. Define the matrix

$$E := -\frac{1}{\alpha}\begin{bmatrix} C^T & W^T & 0 & 0 & A^T & -A^T & 0 & 0 \end{bmatrix} \in \mathbb{R}^{N\times(\ell+r+4m+2)}.$$

It follows from the solution uniqueness of the underlying optimization problem (2.54) that for any $b \in A\mathcal{C}$, $E\mathrm{SOL}(q, M)$ is singleton. Define the function $F(q) := E\mathrm{SOL}(q, M)$. Hence, $F(\cdot)$ is singleton on the closed convex set $\mathcal{W} := \{q = (d, -\Gamma, 1, 1, 0, 0, \alpha b, -\alpha b) \,|\, b \in A\mathcal{C}\}$ and $x_*(b) = F(q)$. By Theorem 2.7.1, F is Lipscthiz on $\mathcal{S}$, i.e., there exists $L > 0$ such that $\|F(q') - F(q)\|_2 \leq L\|q' - q\|_2$ for all $q', q \in \mathcal{W}$. Since $\|q' - q\|_2 = \sqrt{2}\alpha\|b' - b\|_2$ for any $b', b \in A\mathcal{C}$, the desired (global) Lipschitz property of x_* in b holds. □

For a general polyhedral set $\mathcal{C}$, it follows from Lemmas 2.5.2 and 2.5.3 that $y_* + b \in A\mathcal{C}$ (respectively $\frac{\sigma y_*}{\|y_*\|_2} + b \in A\mathcal{C}$), where y_* is a solution to the dual problem (2.13) (respectively (2.15)). Practically, y_* is approximated by a numerical sequence (y^k) generated in the first stage. For the LASSO-like problem (2.12), one uses $y^k + b$ (with a large k) instead of $y_* + b$ in the $\mathrm{BP}_{\mathrm{LASSO}}$ (2.32) in the second stage. This raises the question of whether $y^k + b \in A\mathcal{C}$ for all large k, which pertains to the feasibility of $b \in A\mathcal{C}$ subject to perturbations. The same question also arises for the BPDN-like problem (2.14). We discuss a mild sufficient condition on A and $\mathcal{C}$ for the feasibility under perturbations for a given b. Suppose $\mathcal{C}$ has a nonempty interior and A has full row rank, which holds for almost all $A \in \mathbb{R}^{m\times N}$ with $N \geq m$. In view of $\mathrm{ri}(A\mathcal{C}) = A\mathrm{ri}(\mathcal{C}) = A\mathrm{int}(\mathcal{C})$ [60, Theorem 6.6], we see that $A\mathcal{C}$ has nonempty interior given by $A\mathrm{ri}(\mathcal{C}) = A\mathrm{int}(\mathcal{C})$. Thus if $\widehat{b} := y_* + b$ is such that $\widehat{b} = A\widehat{x}$ for some $\widehat{x} \in \mathrm{int}(\mathcal{C})$, then there exists a neighborhood $\mathcal{N}$ of $\widehat{b}$ such that $b \in A\mathcal{C}$ for any $b \in \mathcal{N}$. Additional sufficient conditions independent of b can also be established. For example, suppose $\mathcal{C}$ is unbounded, and consider its recession cone $\mathcal{K} := \{x \,|\, Cx \leq 0\}$. Let $h_i \in \mathbb{R}^N$ be generators of $\mathcal{K}$, i.e., $\mathcal{K} = \mathrm{cone}\{h_1, \ldots, h_s\}$. Define the matrix $H := [h_1, \ldots, h_s]$. A sufficient condition for $A\mathcal{C}$ to be open is $A\mathcal{K} = \mathbb{R}^m$, which is equivalent to $AH\mathbb{R}^s_+ = \mathbb{R}^m$. By the Theorem of Alternative, the latter condition is further equivalent to (i) AH has full row rank; and (ii) there exists a nonnegative matrix Q such that $AH(I+Q) = 0$. Some simplified conditions can be derived from it for special cases. For instance, when $\mathcal{C} = \mathbb{R}^N$, A need to have full row rank; when $\mathcal{C} = \mathbb{R}^N_+$, A need to have full row rank and $A(I+Q) = 0$ for a nonnegative matrix Q.

Based on the previous results, we establish the overall convergence of the two-stage algorithms.

Theorem 2.7.4. *Consider the two-stage distributed algorithms for the LASSO-like problem (2.12) (resp. the BPDN-like problem (2.14)) with the norm $\|\cdot\|_\star$. Let (y^k) be a sequence generated in the first stage such that $(y^k) \to y_*$ as $k \to \infty$ and $b + y^k \in A\mathcal{C}$ (resp. $b + \frac{\sigma y^k}{\|y^k\|_2} \in A\mathcal{C}$) for all large k, where y_* is a solution to the dual problem (2.13) (resp. (2.15)), and (x^s) be a convergent sequence in the second stage for solving (2.32) (resp. (2.33)). Then the following hold:*

(i) $(x^s) \to x_*$ as $k, s \to \infty$, where x_* is the unique solution to the regularized BP$_{LASSO}$ (2.32) (resp. BP$_{BPDN}$ (2.33)).

(ii) Let $\|\cdot\|_\star$ be the ℓ_1-norm. Suppose (y^k) has the convergence rate $O(\frac{1}{k^q})$ and (x^s) has the convergence rate $O(\frac{1}{s^r})$. Then (x^s) converges to x_* in the rate of $O(\frac{1}{k^q}) + O(\frac{1}{s^r})$.

Proof. We consider the LASSO-like problem only; the similar argument holds for the BPDN-like problem.

(i) For each k, let $\widehat{b}^k := b + y^k$, where (y^k) is a sequence generated from the first stage that converges to y_*. When $\widehat{b}^k$ is used in the BP$_{\text{LASSO}}$ (2.32) in the second stage, i.e., the constraint $Ax = b + y_*$ is replaced by $Ax = \widehat{b}^k$, we have $\|x^s(\widehat{b}^k) - x_*\| \leq$

$\|x^s(\widehat{b}^k) - x_*(\widehat{b}^k)\| + \|x_*(\widehat{b}^k) - x_*\|$, where $x_*(\widehat{b}^k)$ is the unique solution to the BP_{LASSO} (2.32) corresponding to the constraint $Ax = \widehat{b}^k$ (and $x \in \mathcal{C}$). Since $\big(x^s(\widehat{b}^k)\big)$ converges to $x_*(\widehat{b}^k)$ as $s \to \infty$ (for a fixed k), $\|x^s(\widehat{b}^k) - x_*(\widehat{b}^k)\|$ converges to zero. Further, note that $x_* = x_*(\widehat{b}_*)$ with $\widehat{b}_* := b + y_*$. Then it follows from the continuity property shown in Theorem 2.7.2 that $\|x_*(\widehat{b}^k) - x_*\| = \|x_*(\widehat{b}^k) - x_*(\widehat{b}_*)\|$ converges to zero as $k \to \infty$ in view of the convergence of (y^k) to y_*. This establishes the convergence of the two-stage algorithm.

(ii) When $\|\cdot\|_\star$ is the ℓ_1-norm, we deduce via Theorem 2.7.3 that x_* is Lipschitz continuous in b on $A\mathcal{C}$, i.e., there exists a constant $L > 0$ such that $\|x_*(b) - x_*(b')\| \leq L\|b - b'\|$ for any $b, b' \in A\mathcal{C}$. Hence, $\|x^s(\widehat{b}^k) - x_*\| \leq \|x^s(\widehat{b}^k) - x_*(\widehat{b}^k)\| + \|x_*(\widehat{b}^k) - x_*(\widehat{b}_*)\| \leq \|x^s(\widehat{b}^k) - x_*(\widehat{b}^k)\| + L\|\widehat{b}^k - \widehat{b}_*\| = \|x^s(\widehat{b}^k) - x_*(\widehat{b}^k)\| + L\|y^k - y_*\| = O(\frac{1}{s^r}) + O(\frac{1}{k^q})$. $\square$

2.8 Numerical Results

We present numerical results to demonstrate the performance of the proposed two-stage column partition based distributed algorithms for the standard LASSO/BPDN, fused LASSO/BPDN, group LASSO, and their extensions, e.g., those subject to polyhedral constraints. Distributed algorithms are implemented on MATLAB and run on a computer of the following processor: Intel(R) Core(TM) i7-8550U CPU with 4 cores @ 1.80GHz and RAM: 16.0GB. We consider a network of $p = 40$ agents with two topologies: the first is a cyclic graph, and the second is a random graph shown in Figure 2.1. The matrix $A \in \mathbb{R}^{100\times 4000}$ is random normal (i.e., $m = 100$ and $N = 4000$), and $b \in \mathbb{R}^{100}$ is a random normal vector. For the standard/fused BPDN and its extensions, $\|b\|_2 = 11.63$ and the parameter $\sigma = 0.2$, satisfying $\|b\|_2 > \sigma$. We consider even column partitioning, i.e., each agent knows 100 columns of A.

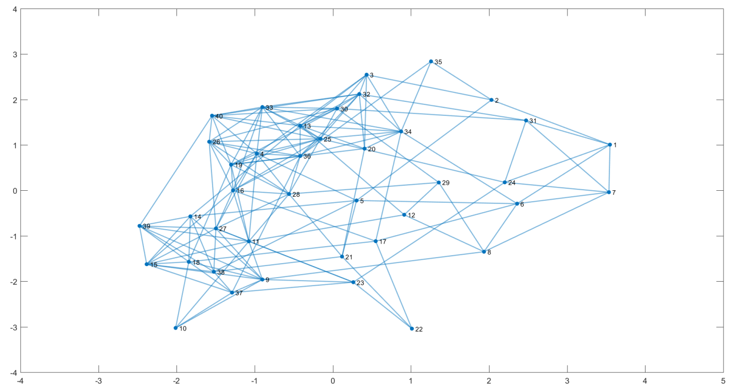


Figure 2.1: The topology of the random graph

In each scheme, the stopping criterion is measured by the absolute error of two neighboring iterates, and its termination tolerance is given below. Further, to simplify notation, we use the following abbreviations: DA for distributed averaging, LA for local averaging, DR for Douglas-Rachford, and OS for operator splitting. For instance, a DA-OS scheme represents a distributed averaging based operator splitting scheme. In each table below, *Time* stands for the computation time *per agent*.

In each DA based scheme, we use the distributed averaging scheme with optimal constant edge weight [90, Section 4.1] for consensus computation. Numerical experiments show that this scheme is highly efficient. For instance, to compute the average of $\mathbf{y} = (\mathbf{y}_i)_{i=1}^p$ with $p = 40$ and $\mathbf{y}_i \in \mathbb{R}^{100}$, it takes 0.0061 (resp. 0.001) seconds per agent to converge on the cyclic (resp. the random) graph with the relative error less than 10^{-7}. For the standard and fused LASSO/BPDN-like problems involving the ℓ_1-norm, the subproblem in each scheme, e.g., the projection step in an OS scheme, the proximal operator in DR scheme, or the subproblem in C-ADMM, is solved via a quadratic program; see Remark 2.6.3. For the group LASSO, the projection step is formulated as a second order cone program (SOCP) and solved by SeDuMi.

Standard LASSO: Stage One

Scheme	DA-OS	C-ADMM	LA-DR
Parameter	$\varrho = 1.2, \eta = 3$	$\eta = 1.5$	$\eta = 1.2, \rho = 0.6$
Time (sec)	58.8	64.9	51.2
$J_{\text{RE, 1}}$	8.1×10^{-6}	4.2×10^{-5}	4.5×10^{-5}

Standard LASSO: Stage Two

Scheme	DA-DR	C-ADMM	LA-DR
Parameter	$\rho = 0.6, \eta = 0.8$	$\eta = 5.5$	$\eta = 0.8, \rho = 0.1$
Time (sec)	41.2	26.7	66.2
$J_{\text{RE, 2}}$	6.2×10^{-6}	5.2×10^{-4}	5.6×10^{-3}
$J_{\text{RE, o}}$	3.5×10^{-7}	1.9×10^{-5}	6.2×10^{-4}

To evaluate the accuracy of the proposed schemes, let J denote the objective function of each (primal) problem, and x^*_{dist} be a numerical primal solution obtained from a proposed 2-stage distributed scheme. Let $J^*_{\text{dist}} := J(x^*_{\text{dist}})$, J^*_{true} be the true optimal value obtained from a high-precision centralized scheme, and $J_{\text{RE, o}} := \frac{|J^*_{\text{dist}} - J^*_{\text{true}}|}{|J^*_{\text{true}}|}$ be the *overall* relative error of the optimal value. We also use $J_{\text{RE, s}}$ to denote the similar relative error for stage $s = 1, 2$.

2.8.1 Numerical Results for the LASSO-like Problems

Consider the cyclic graph and $\mathcal{C} = \mathbb{R}^N$ unless otherwise stated.

• **Standard LASSO** The ℓ_1-penalty parameter $\lambda = 1.8$, and the regularization parameter in the second stage $\alpha = 0.18$. We apply three schemes for each stage: DA-OS (Algorithm 2), C-ADMM (Algorithm 3), LA-DR (Algorithm 4) for stage one, and DA-DR (Algorithm 5), C-ADMM (Algorithm 6), LA-DR (similar to Algorithm 4) for stage two. The termination tolerances for stages one and two are 10^{-4} and 10^{-5}, respectively. See the following table for the numerical results.

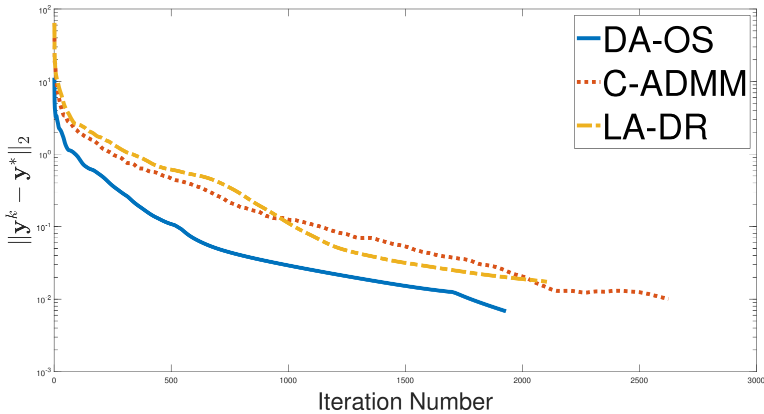


Figure 2.2: Convergence behaviors in stage one of standard LASSO.

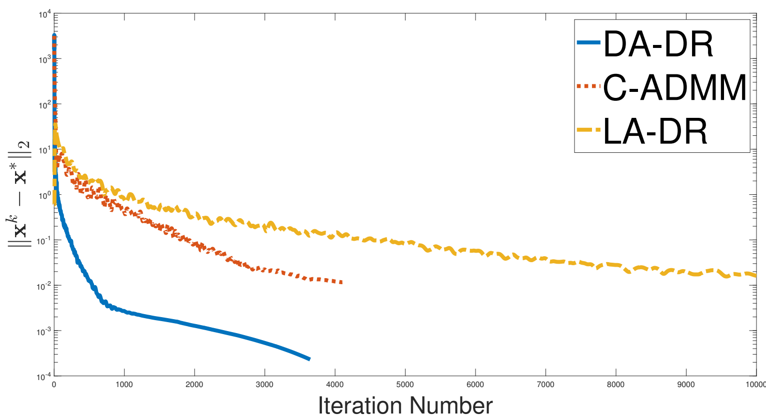


Figure 2.3: Convergence behaviors in stage two of standard LASSO.

The convergence behaviors of these schemes in the two stages are displayed in Figures 2.2-2.3. In the first stage, the errors of the dual variable is shown; in the second stage, we compute the corresponding primal variables from its numerical dual solutions and display its convergence behavior, where y^* and x^* are the true dual and primal solutions, respectively.

We also test the standard LASSO on the random graph via the DA-OS and C-ADMM for stage one, and the DA-DR for stage two, which is also used for the scaled regularized BP (cf. Remark 2.6.1) with the regularization constant $\alpha = 0.1$ in stage two.

The same termination tolerances are used. See the following table for a summary of the numerical results.

Standard LASSO: Stage One

Scheme	DA-OS	C-ADMM
Parameter	$\varrho = 1.2, \eta = 3$	$\eta = 1.5$
Time (sec)	54.7	73.3
$J_{\mathrm{RE},\ 1}$	8.3×10^{-6}	3.2×10^{-5}

Standard LASSO: Stage Two

Scheme	DA-DR	DA-DR for scaled r-BP
Parameter	$\rho = 0.6, \eta = 0.8$	$\rho = 0.6, \eta = 0.8$
Time (sec)	23.6	43.9
$J_{\mathrm{RE},\ 2}$	1.5×10^{-6}	2.9×10^{-4}
$J_{\mathrm{RE},\ \mathrm{o}}$	4.7×10^{-7}	2.1×10^{-6}

• **LASSO with $\mathcal{C} = \mathbb{R}^N_+$** This problem is known as the nonnegative garrote in the literature [93]. We apply the DA-OS and C-ADMM for stage one, and the DA-DR and C-ADMM for stage two with $\alpha = 0.18$. The termination tolerances are 10^{-4} for both the schemes in stage one, 10^{-4} for the DA-DR in stage two, and 5×10^{-5} for the C-ADMM in stage two. See the following table for the numerical results.

Constrained LASSO: Stage One

Scheme	DA-OS	C-ADMM
Parameter	$\varrho = 1.2, \eta = 3$	$\eta = 1.5$
Time (sec)	12.4	24.9
$J_{\mathrm{RE},\ 1}$	1.2×10^{-5}	7.5×10^{-5}

Constrained LASSO: Stage Two

Scheme	DA-DR	C-ADMM
Parameter	$\rho = 0.6, \eta = 0.95$	$\eta = 5.5$
Time (sec)	71.5	32.8
$J_{\mathrm{RE},\ 2}$	3.5×10^{-3}	3.5×10^{-3}
$J_{\mathrm{RE},\ \mathrm{o}}$	1.3×10^{-3}	7.8×10^{-3}

• **Fused LASSO** The matrix $E = \begin{bmatrix} \lambda I \\ \gamma D_1 \end{bmatrix}$ with $\lambda = 0.6$ and $\gamma = 0.4$, and the regularization constant $\alpha = 0.18$. We apply DA-OS (Algorithm 7) for stage one and the DA-DR (Algorithm 8) for stage two with termination tolerances 8×10^{-4} and 10^{-4}, respectively. We obtain $J_{\mathrm{RE},\ \mathrm{o}} = 1.6 \times 10^{-4}$.

Fused LASSO: Stage One

Scheme	DA-OS
Parameter	$\varrho = 0.3, \eta = 3$
Time (sec)	213.8
$J_{\mathrm{RE},\ 1}$	8.5×10^{-5}

Fused LASSO: Stage Two

Scheme	DA-DR
Parameter	$\rho = 0.4, \eta = 0.6$
Time (sec)	271.9
$J_{\mathrm{RE},\ 2}$	6.5×10^{-4}

• **Group LASSO** The penalty constant $\lambda = 1.8$, and the regularization parameter $\alpha = 0.18$. For stage one, a DA-OS scheme similar to Algorithm 2 is used by replacing the set $\mathcal{W}_i$ with $\widehat{\mathcal{W}}_i := \{\mathbf{y}_i \,|\, \|(A_{\bullet \mathcal{I}_i})^T \mathbf{y}_i\|_2 \le \lambda\}$ as in Section 2.6.5. Its projection step is formulated as a SOCP and solved by SeDuMi. For stage two, we exploit the reduced dual problem via the soft thresholding operator $S_{\|\cdot\|_2}$ and apply the DA-OS scheme similar to that in (2.46) by dropping $\boldsymbol{\mu}$ and replacing G_i with $J_i(\mathbf{y}_i) := (b^T \mathbf{y}_i)/p + \frac{1}{2\alpha}\big[\big(\big\|(A_{\bullet \mathcal{I}_i})^T \mathbf{y}_i - 1\big)_+\big]^2$ for each i. The termination tolerances for stages one and two are 10^{-5} and 8×10^{-7}, respectively. We obtain $J_{\text{RE, o}} = 3.6 \times 10^{-4}$.

Group LASSO: Stage One

Scheme	DA-OS
Parameter	$\varrho = 1.2, \eta = 2$
Time (sec)	52.8
$J_{\text{RE, 1}}$	3.6×10^{-4}

Group LASSO: Stage Two

Scheme	DA-OS
Parameter	$\rho = 0.005, \eta = 0.3$
Time (sec)	19.5
$J_{\text{RE, 2}}$	4.5×10^{-5}

2.8.2 Numerical Results for the BPDN-like Problems

Consider the cyclic graph and $\mathcal{C} = \mathbb{R}^N$ unless otherwise stated.

• **Standard BPDN** The regularization parameter in the second stage $\alpha = 0.15$. We apply DA-OS and IC-ADMM [10] (with the parameters c and β_i's) for stage one, and DA-DR, C-ADMM, LA-OS for stage two. The termination tolerances for the first and second stages are 4×10^{-4} and 10^{-5} respectively.

Standard BPDN: Stage One

Scheme	DA-OS	IC-ADMM
Parameter	$\varrho = 1.2, \eta = 3$	$c = 1.5, \beta_i = 1.7, \forall\, i$
Time (sec)	26.6	58.5
$J_{\text{RE, 1}}$	5.9×10^{-4}	1.7×10^{-5}

Standard BPDN: Stage Two

Scheme	DA-DR	C-ADMM	LA-OS
Parameter	$\rho = 0.5, \eta = 0.95$	$\eta = 5.5$	$\eta = 0.95, \rho = 0.1$
Time (sec)	13.2	24.4	33.9
$J_{\text{RE, 2}}$	1.68×10^{-5}	1.3×10^{-3}	5.1×10^{-4}
$J_{\text{RE, o}}$	1.7×10^{-5}	1.3×10^{-3}	5.1×10^{-4}

We also test the standard BPDN on the random graph with the regularization parameter $\alpha = 0.18$. We apply DA-OS for stage one, and DA-DR for stage two. The table below shows the numerical results with $J_{\text{RE, o}} = 5.6 \times 10^{-4}$.

Standard BPDN: Stage One

Scheme	DA-OS
Parameter	$\varrho = 1.2, \eta = 3$
Time (sec)	19.9
$J_{\text{RE, 1}}$	7.04×10^{-4}

Standard BPDN: Stage Two

Scheme	DA-DR
Parameter	$\rho = 0.5, \eta = 0.95$
Time (sec)	6.8
$J_{\text{RE, 2}}$	1.28×10^{-3}

• **BPDN with $\mathcal{C} = \mathbb{R}^N_+$** We apply the DA-OS for stage one, and the DA-DR for stage two with $\alpha = 0.18$. The termination tolerances for stage one and stage two are 10^{-5} and 10^{-4}, respectively. We obtain $J_{\text{RE, o}} = 4.9 \times 10^{-4}$.

Constr. BPDN: Stage One

Scheme	DA-OS
Parameter	$\varrho = 1.2, \eta = 3$
Time (sec)	37.1
$J_{\text{RE, 1}}$	1.5×10^{-4}

Constr. BPDN: Stage Two

Scheme	DA-DR
Parameter	$\rho = 0.95, \eta = 0.6$
Time (sec)	49.0
$J_{\text{RE, 2}}$	7.4×10^{-4}

• **Fused BPDN** The matrix $E = \begin{bmatrix} I_N \\ \gamma D_1 \end{bmatrix}$ with $\gamma = 2/3$, and the regularization constant $\alpha = 0.18$. We apply DA-OS and DA-DR for stage one and stage two with the termination tolerances 10^{-4} and 2×10^{-4}, respectively. We obtain $J_{\text{RE, o}} = 6.9 \times 10^{-4}$.

Fused BPDN: Stage One

Scheme	DA-OS
Parameter	$\varrho = 1.2, \eta = 3$
Time (sec)	248.4
$J_{\text{RE, 1}}$	1.5×10^{-6}

Fused BPDN: Stage Two

Scheme	DA-DR
Parameter	$\rho = 0.4, \eta = 0.8$
Time (sec)	205.3
$J_{\text{RE, 2}}$	7.7×10^{-4}

2.8.3 Discussions and Comparison

We compare the proposed two-stage schemes with two existing distributed schemes: the DC-ADMM [10], [47] for the standard LASSO with $\mathcal{C} = \mathbb{R}^N$, and the PDC-ADMM [8] for the standard LASSO with $\mathcal{C} = \mathbb{R}^N_+$. These two schemes cannot handle the non-polyhedral constraints in the BPDN-like problems and the additional coupling in the fused LASSO/BPDN. Hence, we focus on the standard LASSO and its constrained case. The results of the DC-ADMM and PDC-ADMM over the cyclic graph on the same computer with the tolerance 10^{-4} are given below.

Standard LASSO

Scheme	DC-ADMM
Parameter	$c = 0.05$
Time (sec)	307.1
$J_{\text{RE, o}}$	4.1×10^{-4}
Iteration No.	12974

Constrained LASSO

Scheme	PDC-ADMM
Parameter	$c = \tau_i = 0.01$
Time (sec)	323.5
$J_{\text{RE, o}}$	3.6×10^{-2}
Iteration No.	4632

The overall computation time of the proposed two-stage schemes for the standard (resp. constrained) LASSO is 78-121 seconds (resp. 57-94 seconds) with smaller $J_{\text{RE, o}}$. Hence, the proposed schemes outperform the DC-ADMM and PDC-ADMM in both computation time and numerical accuracy.

Since the communication cost is proportional to the number of iterations, we compare the number of iterations of the proposed two-stage schemes vs. DC-ADMM (resp. PDC-ADMM) for the standard LASSO (resp. constrained LASSO) on the cyclic graph. The following table summarizes the number of iterations for the proposed two-stage schemes.

Stage	Standard LASSO: Iteration No.			Constr. LASSO: Iteration No.	
	DA-OS/DR	C-ADMM	LA-DR	DA-OS/DR	C-ADMM
One	OS: 1930	2624	2105	OS: 1175	2864
Two	DR: 3483	4104	9999	DR: 2683	2846

For the standard LASSO, the total iteration numbers of the two-stage DA based scheme, C-ADMM, and LA-DR scheme are 5413, 6728, and 12094, respectively. For the constrained LASSO, the total iteration number is 3958 for the DA-based scheme and is 5710 for the C-ADMM. Note that the DA-based schemes need additional iterations for distributed averaging computation, leading to extensive communications. Nevertheless, the proposed two-stage C-ADMM takes fewer or a similar number of iterations and less computation time while achieving better numerical accuracy in comparison with the DC-ADMM and PDC-ADMM. Finally, the memory costs of the proposed two-stage schemes are same or similar to those of the DC-ADMM or PDC-ADMM.

2.9 Summary

In this chapter, column partition based distributed schemes are developed for a class of densely coupled convex sparse optimization problems, including BP, LASSO, BPDN and their extensions. By leveraging duality theory, exact regularization techniques, and solution properties of the aforementioned problems, we develop dual based fully distributed schemes via column partition. Sensitivity results are used to establish overall convergence of the two-stage distributed schemes for LASSO, BPDN, and their extensions.

CHAPTER III

Fully Distributed Optimization based CAV Platooning Control under Linear Vehicle Dynamics

Chapters 3 and 4 are concerned with the second topics of this thesis, namely developing fully distributed optimization algorithms for real-time implementation of CAV platooning control schemes under both linear and nonlinear vehicle dynamics.

3.1 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 [4], [33].

A number of effective distributed control or optimization schemes have been proposed for CAV platooning [84], [85], [96], [99]. The recent paper [22] 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 and asymptotic dynamics of the entire platoon into account, and can be computed in a *partially distributed manner* i.e., they require all vehicles to exchange information with a central central component for centralized data processing or perform centralized computation in at least one step of these schemes.

In this chapter, we develop a fully distributed optimization based and platoon centered CAV car following control scheme over a general vehicle communication network under linear vehicle dynamics. 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 [26]. However, the proposed MPC is a centralized control approach and its underlying optimization problem does not satisfy this requirement since its objective function is densely coupled and its constraints are locally coupled; see Remark 3.3.2 for details. Therefore, this chapter develops new techniques to overcome this difficulty.

The main contributions of this chapter 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 dis-

tributed 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 above-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. One of the major challenges in developing a fully distributed scheme for the model is the computation time. In order to facilitate real time implementation, the computation time for the scheme should be less than *one second*. A generalized Douglas-Rachford splitting method based distributed scheme is developed, where only local information exchange is needed, leading to a fully distributed scheme which can be used for real time implementation. 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 [22]. Besides, since 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 materials of this chapter are reported in our recent journal publication [64] which is in print. The rest of the chapter is organized as follows. Section 3.2 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 3.3; fundamental properties of this optimization problem are established. Section 3.4 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 3.5 with numerical results given in Section 3.6. Finally, summary is given in Section 3.7.

3.2 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 $i = 1, \ldots, n$, 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 $k \in \mathbb{Z}_+ := \{0, 1, 2, \ldots\}$. 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:

$$x_i(k+1) \;=\; x_i(k) + \tau v_i(k) + \frac{\tau^2}{2} u_i(k), \qquad v_i(k+1) \;=\; v_i(k) + \tau u_i(k). \tag{3.1}$$

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

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

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

(iii) 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:

$$x_{i-1} - x_i \;\ge\; L + r \cdot v_i - \frac{(v_i - v_{\min})^2}{2a_{\min}}, \tag{3.2}$$

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, i.e., $a_{\min} \le u_0(k) \le a_{\max}$, and $v_{\min} \le v_0(k) \le v_{\max}$ for all $k \in \mathbb{Z}_+$. Note that constraints (i) and (ii) are decoupled across vehicles, whereas the safety distance constraint (iii) is state-control coupled since such a constraint involves control inputs of two vehicles. This yields challenges to distribution computation. Further, the identical acceleration or deceleration bounds are considered, although the proposed approach can handle a general case with different acceleration or deceleration bounds.

Communication network topology. We consider a general communication network whose topology is modeled by a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes where the ith node corresponds 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, i.e., $\mathcal{N}_i = \{j \,|\, (i,j) \in \mathcal{E}\}$. The following assumption on the communication network topology is made throughout the chapter:

A.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., $(1,2), (2,3), \ldots, (n-1, n) \in \mathcal{E}$, and the first CAV can receive $x_0(k)$, $v_0(k)$, and $u_0(k)$ from the leading vehicle at each $k \in \mathbb{Z}_+$.

Since the graph is undirected, for any $i, j \in \mathcal{V}$ with $i \neq j$, $(i,j) \in \mathcal{E}$ 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 **A.1** includes many widely used communication networks of CAV platoons, e.g., predecessor-following, predecessor-leader following, immediate-preceding, multiple-preceding, and preceding-and-following networks [97].

3.3 Model Predictive Control for CAV Platooning Control

We exploit the model predictive control (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) $z(k) := \big(x_0 - x_1 - \Delta, \ldots, x_{n-1} - x_n - \Delta\big)(k) \in \mathbb{R}^n$, representing the relative spacing error; (ii) $z'(k) := \big(v_0 - v_1, \ldots, v_{n-1} - v_n\big)(k) \in \mathbb{R}^n$, representing the relative speed between adjacent vehicles; and (iii) $u(k) := \big(u_1, \ldots, u_n\big)(k) \in \mathbb{R}^n$, representing the control input. Further, let $w_i(k) := u_{i-1}(k) - u_i(k)$ for each $i = 1, \ldots, n$, and $w(k) := \big(w_1, \ldots, w_n\big)(k) \in \mathbb{R}^n$, 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

$$S_n := \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 1 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 1 & 1 & \ldots & 1 & 0 \\ 1 & 1 & \ldots & 1 & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}, \qquad S_n^{-1} = \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}. \quad (3.3)$$

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 (3.1):

$$\begin{aligned} &\text{minimize} \;\; J(u(k), \ldots, u(k+p-1)) := \qquad (3.4) \\ &\frac{1}{2}\sum_{s=1}^{p} \Big(\underbrace{\tau^2 u^T(k+s-1) S_n^{-T} Q_{w,s} S_n^{-1} u(k+s-1)}_{\text{ride comfort}} \\ &\qquad + \underbrace{z^T(k+s) Q_{z,s} z(k+s) + (z'(k+s))^T Q_{z',s} z'(k+s)}_{\text{traffic stability and smoothness}} \Big) \end{aligned}$$

subject to: 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 (3.1), and

$$a_{\min} \;\le u_i(k+s-1) \;\le a_{\max}, \qquad v_{\min} \;\le\; v_i(k+s) \;\le\; v_{\max}, \qquad (3.5)$$

$$x_{i-1}(k+s) - x_i(k+s) \quad \geq L + r \cdot v_i(k+s) - \frac{(v_i(k+s) - v_{\min})^2}{2a_{\min}}, \tag{3.6}$$

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 (3.1) to predict $(x_0(k+s+1), v_0(k+s+1))$ for $s = 1, \ldots, p-1$.

Remark 3.3.1. 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 above J are chosen such that smooth traffic dynamics and asymptotic stability is achieved in the closed loop dynamics; see Section 3.5 for details. The presence of the matrix S_n in the first term is due to 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. Lastly, non-constant spacing car following polices can be considered; see Remark 3.4.3 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 chapter:

A.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. Further, by suitably choosing the weight matrices $Q_{w,s}$, it can be shown that the ride comfort term in equation (3.4), 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 is 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 3.5.

The sequential feasibility has been established in [21], [22] for the MPC model (3.4) when $r \geq \tau$. Define $\mathcal{P}((x_i, v_i)_{i=0}^n, u_0) := \{u \in \mathbb{R}^n \,|\, a_{\min} \leq u_i \leq a_{\max},\, v_{\min} \leq v_i + \tau u_i \leq v_{\max},\, h_i(u) \leq 0,\ \forall\, i = 1, \ldots, n\}$, 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 non-empty such that the MPC model (3.4) 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 non-empty.

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 3.3.1. *Consider the linear vehicle dynamics (3.1) 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 (3.4) has nonempty interior at each k.*

Proof. Fix an arbitrary $k \in \mathbb{Z}_+$. Since $v_0(k) > v_{\min}$, it follows from [22, Proposition 3.1] that there exists a vector denoted by $\widehat{u}(k)$ in the interior of $\mathcal{P}((x_i(k), v_i(k))_{i=0}^n, u_0(k))$. Let $x_i(k+1)$ and $v_i(k+1)$ be generated by $\widehat{u}(k)$ (and $(x_i(k), v_i(k))_{i=0}^n, u_0(k))$. Since $v_0(k+1) > v_{\min}$, we deduce via [22, 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 (3.4). $\square$

3.3.1 Constrained MPC Optimization Model

Consider the constrained MPC optimization model (3.4) at an arbitrary but fixed time $k \in \mathbb{Z}_+$ subject to the linear vehicle dynamics (3.1). In view of the following results: for each $s = 1, \ldots, p$,

$$
\begin{aligned}
v_i(k+s) &= v_i(k) + \tau \sum_{j=0}^{s-1} u_i(k+j), \quad z'(k+s) = z'(k) + \tau \sum_{j=0}^{s-1} w(k+j), \\
z(k+s) &= z(k) + s\tau z'(k) + \tau^2 \sum_{j=0}^{s-1} \frac{2(s-j)-1}{2} w(k+j), \\
w(k+s) &= S_n^{-1}\big[-u(k+s) + u_0(k) \cdot \mathbf{1} \big],
\end{aligned}
$$

we formulate (3.4) as the constrained convex minimization problem (where we omit k since it is fixed):

$$\begin{aligned}
&\text{minimize} \quad J(\mathbf{u}) := \tfrac{1}{2}\mathbf{u}^T W \mathbf{u} + c^T \mathbf{u} + \gamma, \\
&\text{subject to} \quad \mathbf{u}_i \in \mathcal{X}_i, \quad (H_i(\mathbf{u}))_s \le 0, \quad \forall\, i = 1, \dots, n, \quad \forall\, s = 1, \dots, p,
\end{aligned} \tag{3.7}$$

where $\mathbf{u} := (\mathbf{u}_1, \dots, \mathbf{u}_n) \in \mathbb{R}^{np}$ with $\mathbf{u}_i := (u_i(k), \dots, u_i(k+p-1)) \in \mathbb{R}^p$, W is a PD matrix to be shown in Lemma 3.3.1 below, $c \in \mathbb{R}^{np}$, $\gamma \in \mathbb{R}$, each $\mathcal{X}_i := \{z \in \mathbb{R}^p \,|\, a_{\min} \cdot \mathbf{1} \le z \le a_{\max} \cdot \mathbf{1},\ (v_{\min} - v_i(k)) \cdot \mathbf{1} \le \tau S_p z \le (v_{\max} - v_i(k)) \cdot \mathbf{1}\}$ is a polyhedral set, and each $(H_i(\cdot))_s$ is a convex quadratic function characterizing the safety distance given by (3.12). Here S_p is the $p \times p$ matrix of the form given by (3.3). Further, $\mathbf{u}_0 := u_0(k) \cdot \mathbf{1}_p \in \mathbb{R}^p$ for the given $u_0(k)$. An important property of the matrix W in (3.7) is given below.

Lemma 3.3.1. *Suppose that $Q_{z,s}$ and $Q_{z',s}$ are PSD and $Q_{w,s}$ are PD for all $s = 1, \dots, p$ (but not necessarily diagonal). Then the matrix W in (3.7) is PD.*

Proof. Let $\mathbf{u}$ be an arbitrary nonzero vector in $\mathbb{R}^{np}$. Since $J(\cdot)$ 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 (3.4), we deduce that for any $\lambda > 0$, $J(\lambda \mathbf{u}) = J(\lambda u(k), \dots, \lambda u(k+p-1)) \ge \frac{\lambda^2}{2} \sum_{s=1}^{p} \tau^2 u^T(k+s-1) S_n^{-T} Q_{w,s} S_n^{-1} u(k+s-1) > 0$, 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, $\frac{J(\lambda \mathbf{u})}{\lambda^2} \ge \frac{1}{2} \sum_{s=1}^{p} \tau^2 u^T(k+s-1) S_n^{-1} Q_{w,s} S_n^{-1} u(k+s-1) > 0$, leading to $\frac{1}{2}\mathbf{u}^T W \mathbf{u} \ge \frac{1}{2} \sum_{s=1}^{p} \tau^2 u^T(k+s-1) S_n^{-1} Q_{w,s} S_n^{-1} u(k+s-1) > 0$. Hence, W is PD. $\square$

To establish the closed form expressions of the matrix W and the vector c in (3.7), 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

$$V \;=\; \begin{bmatrix} V_{1,1} + \tau^2 \widetilde{Q}_{w,1} & V_{1,2} & V_{1,3} & \cdots & \cdots & V_{1,p} \\ V_{2,1} & V_{2,2} + \tau^2 \widetilde{Q}_{w,2} & V_{2,3} & \cdots & \cdots & V_{2,p} \\ \cdots & & \cdots & & \cdots & \\ \cdots & & \cdots & & \cdots & \\ V_{p,1} & V_{p,2} & V_{p,3} & \cdots & \cdots & V_{p,p} + \tau^2 \widetilde{Q}_{w,p} \end{bmatrix} \in \mathbb{R}^{np \times np}, \quad (3.8)$$

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

$$E_{i,j} = \begin{cases} 1 & \text{if } i = n \cdot k + s, \; j = p \cdot (s-1) + k + 1, \text{ for } \; k = 0, \ldots, p-1, \; s = 1, \ldots, n; \\ 0, & \text{otherwise.} \end{cases} \quad (3.9)$$

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

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

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

In light of S_n^{-1} given by (3.3), we have $S_n^{-1} \cdot \mathbf{1} = \mathbf{e}_1$. Therefore, we obtain

$$
d_s(k) = z(k) + s\tau z'(k) + \frac{\tau^2}{2} s^2 \mathbf{e}_1 u_0(k), \qquad f_s(k) = z'(k) + \tau s \mathbf{e}_1 u_0(k). \tag{3.10}
$$

In view of

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

the linear terms in the objective function J are given by

$$
-\sum_{i=1}^{p} \left(\sum_{s=i}^{p} \left[\frac{\tau^2}{2} [2(s-i)+1] d_s^T(k) Q_{z,s} + \tau f_s^T(k) Q_{z',s} \right] \right) S_n^{-1} \cdot u(k+i-1). \tag{3.11}
$$

Using the permutation matrix E given in (3.9), 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 $\mathbf{u}_i$. Since $Q_{z,s}$ and $Q_{z',s}$ are diagonal, it is easy to obtain the following lemma via $d_s(k), f_s(k)$ in (3.10) and the structure of S_n^{-1} given by (3.3).

Lemma 3.3.2. *Consider the vector $c = (c_{\mathcal{I}_1}, \ldots, c_{\mathcal{I}_n})$ given above. Then for each $i = 1, \ldots, n$, 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 $i = 1, \ldots, n-1$, and $c_{\mathcal{I}_n}$ depends only on $z_n(k), z_n'(k)$. Further, only $c_{\mathcal{I}_1}$ depends on $u_0(k)$.*

The above 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 3.4.2.

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

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

The safety distance constraint for the i-th vehicle at time k is given by: for $s = 1, \ldots, p$,

$$\begin{aligned} 0 \geq & -\Big[x_{i-1}(k) + s\tau v_{i-1}(k) - (x_i(k) + s\tau v_i(k)) \Big] - \tau^2 \sum_{j=0}^{s-1} \frac{2(s-j)-1}{2} [u_{i-1}(k+j) \\ & - u_i(k+j)] + L + r v_i(k) + r\tau \sum_{j=0}^{s-1} u_i(k+j) - \frac{1}{2a_{\min}} \Big[\tau^2 \big(\sum_{j=0}^{s-1} u_i(k+j) \big)^2 \\ & + 2\tau(v_i(k) - v_{\min}) \sum_{j=0}^{s-1} u_i(k+j) + (v_i(k) - v_{\min})^2 \Big] := (H_i(\mathbf{u}_{i-1}, \mathbf{u}_i))_s, \end{aligned} \tag{3.12}$$

where $(H_i(\cdot,\cdot))_s$ is a convex quadratic function for each $s = 1, \ldots, p$. Hence, the set $\mathcal{Z}_i := \{\mathbf{u} \in \mathbb{R}^{np} \,|\, (H_i(\mathbf{u}_{i-1}, \mathbf{u}_i))_s \leq 0, \forall\, i = 1, \ldots, p\}$ is closed and convex. The problem (3.7) 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 $i = 1, \ldots, n$, which is a convex quadratically constrained quadratic program (QCQP) and can be solved via a second-order cone program or a semi-definite program in the centralized manner.

Remark 3.3.2. The above results show that each $\mathcal{X}_i$'s are 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 (3.7) in the next section.

3.4 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 (3.7) at each time k using the techniques of locally coupled convex optimization and operator splitting methods. One of the major techniques for developing fully distributed schemes for the underlying optimization problem given by (3.7) is to formulate it as a locally coupled convex optimization problem [26]. Section 1.2.1 shows the formulation of such problems into a locally coupled convex optimization problem. In what follows, we will develop a decomposition method for the underlying matrix W given in equation (3.7). This decomposition helps us to write the objective function as a locally coupled convex optimization problem.

3.4.1 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 (1.1) is expected to be written as the sum of mutiple locally coupled functions preserving certain desired properties, e.g., the (strong) convexity if the central objective function is so, where local coupling satisfies network topology constraints. While the constraints of the problem (3.7) have been shown to be locally coupled (cf. Remark 3.3.2), the central strongly convex quadratic objective function, particularly its quadratic term $\frac{1}{2}\mathbf{u}^T W \mathbf{u}$, 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 subsection, we address this decomposition problem under a mild assumption on network topology.

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

$$S_n^{-T} \Lambda S_n^{-1} = \begin{bmatrix} \lambda_1 + \lambda_2 & -\lambda_2 & & & \\ -\lambda_2 & \lambda_2 + \lambda_3 & -\lambda_3 & & \\ & \ddots & \ddots & \ddots & \\ & & -\lambda_{n-1} & \lambda_{n-1} + \lambda_n & -\lambda_n \\ & & & -\lambda_n & \lambda_n \end{bmatrix}. \tag{3.13}$$

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 $i, j = 1, \ldots, p$. Let $(\boldsymbol{\theta}_{i,j})_k$ denote the kth entry of the vector $\boldsymbol{\theta}_{i,j}$. For each $i = 1, \ldots, n$, define the matrix

$$U_i := \begin{bmatrix} (\boldsymbol{\theta}_{1,1})_i & (\boldsymbol{\theta}_{1,2})_i & \cdots & (\boldsymbol{\theta}_{1,p})_i \\ (\boldsymbol{\theta}_{2,1})_i & (\boldsymbol{\theta}_{2,2})_i & \cdots & (\boldsymbol{\theta}_{2,p})_i \\ \cdots & \cdots & \cdots & \\ (\boldsymbol{\theta}_{p,1})_i & (\boldsymbol{\theta}_{p,2})_i & \cdots & (\boldsymbol{\theta}_{p,p})_i \end{bmatrix} \in \mathbb{R}^{p\times p}. \tag{3.14}$$

It can be shown that $\Theta = E^T \mathrm{diag}(U_1, \ldots, U_n)E$, where E is the permutation matrix given by (3.9). Hence, Θ is PD (resp. PSD) if and only if each U_i is PD (resp. PSD).

Let

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

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 (3.9), a straightforward computation shows that $E^T V E$ is a symmetric block tridiagonal matrix given by

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

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 (3.13), 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, since $W = E^T V E = E^T \mathbf{S}^{-T} \Theta\, \mathbf{S}^{-1} E$, W is PD (resp. PSD) if and only if Θ is PD (resp. PSD), which is also equivalent to that each U_i is PD (resp. PSD); see the comment after (3.14).

In what follows, we consider PSD (resp. PD) matrix decomposition for a PSD (resp. 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)

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

(ii)

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

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

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

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

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

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

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 above decomposition to be PD.

Proposition 3.4.1. *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_i's be defined in (3.14) corresponding to $\boldsymbol{\theta}_{i,j}$'s. Note that each U_i is PSD as W is PSD. Let

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

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

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

Since 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 (3.14) is PD. Define

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

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

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. Since $\breve{W}^1$ 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,

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

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. Further, it is easy to show that the matrix $\grave{W}^3 := \breve{W}^3 + \delta_2 \cdot \begin{bmatrix} I_p & & \\ & I_p & \\ & & 0 \end{bmatrix}$ is PD such that for any $\delta_3 \in (0, \lambda_{\min}(\grave{W}^3))$, 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,

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

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. $\square$

To obtain the desired decomposition using the above proposition, we observe that the matrix V in (3.8) is given by $\mathbf{S}^{-T}\Theta\mathbf{S}^{-1}$ for some matrix Θ of the form given below (3.13) whose blocks are positive combinations of $Q_{z,s}, Q_{z',s}$ and $Q_{w,s}$. Since $Q_{z,s}$ 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 3.3.1, W is PD. Hence, there are uncountably many ways to construct positive δ_s, and thus PD $\widehat{W}^s$, as shown in the above proposition. Therefore, we obtain the following strongly convex decomposition for the objective function J in (3.7), where we set the constant $\gamma = 0$ without loss of generality:

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

where the strongly convex functions J_i are given by

$$
\begin{aligned}
J_1(\mathbf{u}_1, \mathbf{u}_2) &:= \frac{1}{2}\begin{bmatrix} \mathbf{u}_1^T & \mathbf{u}_2^T \end{bmatrix} \widehat{W}_1 \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} + c_{\mathcal{I}_1}^T \mathbf{u}_1, \\
\text{for } \ i = 2, \ldots, n-1 & \\
J_i(\mathbf{u}_{i-1}, \mathbf{u}_i, \mathbf{u}_{i+1}) &:= \frac{1}{2}\begin{bmatrix} \mathbf{u}_{i-1}^T & \mathbf{u}_i^T & \mathbf{u}_{i+1}^T \end{bmatrix} \widehat{W}_i \begin{bmatrix} \mathbf{u}_{i-1} \\ \mathbf{u}_i \\ \mathbf{u}_{i+1} \end{bmatrix} + c_{\mathcal{I}_i}^T \mathbf{u}_i, \qquad (3.15) \\
J_n(\mathbf{u}_{n-1}, \mathbf{u}_n) &:= \frac{1}{2}\begin{bmatrix} \mathbf{u}_{n-1}^T & \mathbf{u}_n^T \end{bmatrix} \widehat{W}_n \begin{bmatrix} \mathbf{u}_{n-1} \\ \mathbf{u}_n \end{bmatrix} + c_{\mathcal{I}_n}^T \mathbf{u}_n.
\end{aligned}
$$

Remark 3.4.1. The above decomposition method is applicable to any vehicle communication network satisfying the assumption **A.1** in Section 3.2, i.e., $(i, i+1) \in \mathcal{E}$ for all $i = 1, \ldots, n-1$. Besides, various alternative approaches can be developed to construct PD matrices $\widehat{W}^s$ using the similar idea given in the above proposition. Further, a similar decomposition method can be developed for other vehicle communication networks different from the 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 $i = 2, \ldots, n-1$ and $2 \in \mathcal{N}_1, n-1 \in \mathcal{N}_n$.

3.4.2 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 (cf. Remark 3.4.1). 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

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

Recall that $\mathcal{P}$ is defined by convex quadratic functions. The underlying optimization problem (3.7) 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 [26] and solve it using fully distributed algorithms. Specifically, in view of the decompositions given by (3.15), the objective function in (3.7) 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 3.3.2, 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 [26], define $\widehat{\mathbf{u}}_i := \big(\mathbf{u}_i, (\mathbf{u}_{i,j})_{j \in \mathcal{N}_i}\big)$, 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

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

Then the underlying optimization problem (3.7) can be equivalently written as

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

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

$$\min_{\widehat{\mathbf{u}}} F(\widehat{\mathbf{u}}) := \sum_{i=1}^{n} J_i(\widehat{\mathbf{u}}_i) + \sum_{i=1}^{n} \mathbf{I}_{\mathcal{P}_i}(\widehat{\mathbf{u}}_i) + \mathbf{I}_{\mathcal{A}}(\widehat{\mathbf{u}}), \tag{3.16}$$

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 3.3.1 that the Slater's condition holds under the mild assumptions given in Corollary 3.3.1, e.g., $v_0(k) > v_{\min}$ for all $k \in \mathbb{Z}_+$. Hence, by [60, Corollary 23.8.1], $\partial F(\widehat{\mathbf{u}}) = \sum_{i=1}^n \Big(\nabla J_i(\widehat{\mathbf{u}}_i) + \mathcal{N}_{\mathcal{P}_i}(\widehat{\mathbf{u}}_i)\Big) + \mathcal{N}_{\mathcal{A}}(\widehat{\mathbf{u}})$ 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 (3.16) is a locally coupled convex optimization problem; see Section 1.2.1. 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 9 Generalized Douglas-Rachford Splitting Method based Fully Distributed Algorithm

Choose constants $0 < \alpha < 1$ and $\rho > 0$
Initialize $k = 0$, and choose an initial point z^0
while the stopping criteria is not met **do**
  for $i = 1, \ldots, n$ **do**
    Compute $\overline{z}_i^k$ using equation (3.17), and let $w_i^{k+1} \leftarrow \overline{z}_i^k$
  end for
  for $i = 1, \ldots, n$ **do**
    $z_i^{k+1} \leftarrow z_i^k + 2\alpha \cdot \Big[\mathrm{Prox}_{\rho \widehat{J}_i}\big(2w_i^{k+1} - z_i^k\big) - w_i^{k+1} \Big]$
  end for
  $k \leftarrow k+1$
end while
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 $\mathrm{Prox}_f(\cdot)$ denote the proximal operator, i.e., for any given $x \in \mathbb{R}^n$,

$$\mathrm{Prox}_f(x) \ := \ \operatorname*{arg\,min}_{z \in \mathbb{R}^n} \ f(z) + \frac{1}{2}\|z - x\|_2^2.$$

Further, 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 (3.16). By grouping the first two sums (with separable variables) in the objective function of (3.16), we apply the generalized Douglas-Rachford splitting algorithm [26]. 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:

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

It is shown in [15], [26] that the sequence (w^k) converges to the unique minimizer $\widehat{\mathbf{u}}^*$ of the optimization problem (3.16). In the above 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 $\widehat{\mathbf{u}}_i := \big(\mathbf{u}_i, (\mathbf{u}_{ij})_{j\in\mathcal{N}_i}\big)$, $\overline{\mathbf{u}} := \Pi_{\mathcal{A}}(\widehat{\mathbf{u}})$ is given by [26, Section IV]:

$$\overline{\mathbf{u}}_j = \overline{\mathbf{u}}_{ij} = \frac{1}{1+|\mathcal{N}_j|}\Big(\widehat{\mathbf{u}}_j + \sum_{k\in\mathcal{N}_j}\widehat{\mathbf{u}}_{kj}\Big), \qquad \forall\,(i,j)\in\mathcal{E}. \tag{3.17}$$

Furthermore, due to the decoupled structure of $\widehat{J}_i$'s, we obtain the distributed version of the above algorithm, which is also summarized in Algorithm 9:

$$w_i^{k+1} = \overline{z}_i^k, \qquad i = 1,\ldots,n; \tag{3.18a}$$

$$z_i^{k+1} = z_i^k + 2\alpha\cdot\Big[\mathrm{Prox}_{\rho\widehat{J}_i}\big(2w_i^{k+1} - z_i^k\big) - w_i^{k+1}\Big], \quad i = 1,\ldots,n. \tag{3.18b}$$

In the distributed scheme (3.18), the first step (or Line 5 of Algorithms 9) is a consensus step, and the consensus computation is carried out in a fully distributed and synchronous manner as indicated in [26, Section IV]. The second step in (3.18) (or Line 8 of Algorithms 9) does not need inter-agent communication [26] 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 $\mathrm{Prox}_{\rho\widehat{J}_i}(\widehat{\mathbf{u}}_i)$ becomes

$$\mathrm{Prox}_{\rho\widehat{J}_i}(\widehat{\mathbf{u}}_i) \;=\; \operatorname*{arg\,min}_{z\in\mathcal{P}_i}\; J_i(z) + \frac{1}{2\rho}\|z - \widehat{\mathbf{u}}_i\|_2^2.$$

Since $\mathcal{P}_i$ is the intersection of a polyhedral set and a quadratically constrained convex set and J_i is a quadratic convex function, $\mathrm{Prox}_{\rho\widehat{J}_i}(\widehat{\mathbf{u}}_i)$ is formulated as a QCQP and can be solved via a second order cone program [1] or a semidefinite program. Efficient numerical packages, e.g., SeDuMi [75], can be used for solving the QCQP with high accuracy. Lastly, a typical (global) stopping criterion in the scheme (3.18) (or Algorithm 9) is defined by the error tolerance of two neighboring z^k's, i.e., $\|z^{k+1} - z^k\|_2 \le \varepsilon$, where $\varepsilon > 0$

is an error tolerance. For distributed computation, one can use its local version, i.e., $\|z_i^{k+1} - z_i^k\|_2 \leq \varepsilon/n$, as a stopping criterion for each vehicle.

Remark 3.4.2. Other distributed algorithms can be used to solve the underlying optimization problem (3.16). For example, the three operator splitting method based schemes developed in [15] can be applied. To describe such schemes, let $\widehat{L} := \max_{i=1,\dots,n} \|\widehat{W}^i\|_2 > 0$. Note that the Hessian $HJ(\widehat{\mathbf{u}}) = \text{diag}(\widehat{W}^i)_{i=1,\dots,n}$. Hence, ∇J is $\widehat{L}$-Lipschitz continuous and thus $1/\widehat{L}$-cocoercive. Further, 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 [15, Algorithm 1]:

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

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 above algorithm:

$$\begin{aligned} w_i^{k+1} &= \overline{z}_i^k, \qquad i = 1, \dots, n; \\ z_i^{k+1} &= z_i^k + \lambda \cdot \Big[\Pi_{\mathcal{P}_i}\big(2w_i^{k+1} - z_i^k - \gamma \cdot [\widehat{W}^i w_i^{k+1} + c_{\mathcal{I}_i}]\big) - w_i^{k+1}\Big], \quad i = 1, \dots, n. \end{aligned}$$

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 $\mu = \min_{i=1,\dots,n} \lambda_{\min}(\widehat{W}^i)$, i.e., $(x-y)^T(\nabla J(x) - \nabla J(y)) \geq \mu \|x-y\|_2^2, \forall\, x, y$. Since the subdifferential of the indicator function of a closed convex set is monotone, an accelerated scheme developed in [15, 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:

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

where $\widetilde{\mu} := \eta \cdot \mu$. It is shown in [15, 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 9 outperforms the three operator splitting method based schemes in term of real-time computation when $p \geq 3$; see Section 3.6.2 for comparison and details.

Remark 3.4.3. The proposed MPC formulation, decomposition method, and fully distributed schemes can be extended to non-constant 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 $s = 1, \ldots, p$, $(z_i(k+s), z_i'(k+s))$ depends on $u_i(k), \ldots, u_i(k+s-1)$ and $u_{i-1}(k), \ldots, u_{i-1}(k+s-1)$ only such that its MPC formulation also leads to a locally coupled QCQP, to which the proposed fully distributed schemes are applicable.

3.5 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 [22, 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

$$z(k+1) \;=\; z(k) + \tau z'(k) + \frac{\tau^2}{2} w(k), \qquad z'(k+1) \;=\; z'(k) + \tau w(k), \tag{3.19}$$

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

$$J(w(k)) \;=\; \frac{1}{2}\Big[z^T(k+1)Q_z, z(k+1) + (z'(k+1))^T Q_{z'} z'(k+1)\Big] + \frac{\tau^2}{2}\widetilde{w}^T(k) Q_w \widetilde{w}(k),$$

where we recall that $\widetilde{w}(k) = w(k) - u_0(k)\mathbf{e}_1$. It follows from the similar argument in [22, Section 5] that the the closed-loop system is given by the following linear system:

$$\begin{bmatrix} z(k+1) \\ z'(k+1) \end{bmatrix} \;=\; \begin{bmatrix} A_{\rm C} \end{bmatrix} \begin{bmatrix} z(k) \\ z'(k) \end{bmatrix} + \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n \end{bmatrix} \widehat{W} Q_w \mathbf{e}_1 \cdot u_0(k), \tag{3.20}$$

where $A_{\rm C}$ is the closed loop dynamics matrix given below, and

$$A_{\rm C} := \begin{bmatrix} I_n - \frac{\tau^2}{4}\widehat{W} Q_z & \tau I_n - \widehat{W}\Big(\frac{\tau^3}{4} Q_z + \frac{\tau}{2} Q_{z'}\Big) \\ -\frac{\tau}{2}\widehat{W} Q_z & I_n - \widehat{W}\Big(\frac{\tau^2}{2} Q_z + Q_{z'}\Big) \end{bmatrix}; \quad \widehat{W} \;:=\; \left[\frac{\tau^2 Q_z}{4} + Q_{z'} + Q_w\right]^{-1}. \tag{3.21}$$

The matrix A_C in (3.20) plays an important role in the closed loop stability and desired transient dynamical performance. Since $Q_z, Q_{z'}$ and Q_w are all diagonal and PSD (resp. 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 [22, Proposition 5.1] and is thus omitted.

Proposition 3.5.1. *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, i.e., 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 non-real, 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 $1 - (\frac{\alpha_i \tau^2}{2} + \beta_i)\frac{1}{d_i} < \mu_i < 1 - \frac{\alpha_i \tau^2}{4 d_i}$.

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 $s = 1, \ldots, p$,

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

and with slightly abusing notation, the objective function is

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

where $\widetilde{w}(k+s) := w(k+s) - u_0(k) \cdot \mathbf{e}_1$ introduced in Remark 3.3.1. It follows from the similar development in Section 3.3.1 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 3.3.1, it can be shown that $\mathbf{H} \in \mathbb{R}^{pn \times pn}$ is a symmetric PD matrix. Further, it resembles the matrix V in (3.8) (by replacing S_n^{-1} with I_n), i.e.,

$$\mathbf{H} = \begin{bmatrix} \breve{\mathbf{H}}_{1,1} + \tau^2 Q_{w,1} & \breve{\mathbf{H}}_{1,2} & \breve{\mathbf{H}}_{1,3} & \cdots & \cdots & \breve{\mathbf{H}}_{1,p} \\ \breve{\mathbf{H}}_{2,1} & \breve{\mathbf{H}}_{2,2} + \tau^2 Q_{w,2} & \breve{\mathbf{H}}_{2,3} & \cdots & \cdots & \breve{\mathbf{H}}_{2,p} \\ \cdots & & \cdots & \cdots & & \\ \cdots & & \cdots & \cdots & & \\ \breve{\mathbf{H}}_{p,1} & \breve{\mathbf{H}}_{p,2} & \breve{\mathbf{H}}_{p,3} & \cdots & \cdots & \breve{\mathbf{H}}_{p,p} + \tau^2 Q_{w,p} \end{bmatrix} \in \mathbb{R}^{np \times np}, \tag{3.22}$$

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 (3.10) and (3.11) 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 by: for each $i = 1, \ldots, p$,

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

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

$$\mathbf{K} := -\begin{bmatrix} I_n & 0 & \ldots & 0\end{bmatrix}\mathbf{H}^{-1}\mathbf{G} \in \mathbb{R}^{n\times 2n}, \qquad \mathbf{d} := \begin{bmatrix} I_n & 0 & \ldots & 0\end{bmatrix}\mathbf{H}^{-1}\mathbf{g} \in \mathbb{R}^{n}. \tag{3.23}$$

The closed loop system becomes

$$\begin{bmatrix} z(k+1) \\ z'(k+1)\end{bmatrix} = \underbrace{\left\{\begin{bmatrix} I_n & \tau I_n \\ 0 & I_n\end{bmatrix} + \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n\end{bmatrix}\mathbf{K}\right\}}_{A_{\mathrm{C}}}\begin{bmatrix} z(k) \\ z'(k)\end{bmatrix} + \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n\end{bmatrix} u_0(k)\cdot\mathbf{d}, \tag{3.24}$$

where A_{C} is the closed loop dynamics matrix, and the subscript of A_{C} represents the closed loop.

Since $Q_{z,s}, Q_{z',s}$ 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 $\boldsymbol{\alpha}^s = (\alpha^s_i)^n_{i=1}$, $\boldsymbol{\beta}^s = (\beta^s_i)^n_{i=1}$, and $\boldsymbol{\zeta}^s = (\zeta^s_i)^n_{i=1}$ 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 $A_{\mathrm{C}}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ 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 $\widetilde{A} := \widehat{E}^T A_{\mathrm{C}} \widehat{E}$ is a block diagonal matrix, i.e., $\widetilde{A} = \mathrm{diag}(\widetilde{A}_1, \widetilde{A}_2, \ldots, \widetilde{A}_n)$ whose each block

$\widetilde{A}_i \in \mathbb{R}^{2\times 2}$ is given by

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

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

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

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 $\widetilde{\mathbf{H}} = \mathrm{diag}(\widetilde{\mathbf{H}}_1, \widetilde{\mathbf{H}}_2, \dots, \widetilde{\mathbf{H}}_n) = E^T \mathbf{H} E$ for the permutation matrix $E \in \mathbb{R}^{pn\times pn}$ given by (3.9). Since $\mathbf{H}$ 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 p's. When $p = 1$, $\widetilde{\mathbf{H}}_i = \tau^2\big(\frac{\tau^2}{4}\alpha_i^1 + \beta_i^1 + \zeta_i^1\big)$ 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 $i = 1, \dots, s$. When $p = 2$, we have, for each $i = 1, \dots, n$,

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

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 3.5.1. *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 $i = 1, \dots, n$, $A_C(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is Schur stable, i.e., its spectral radius is strictly less than 1.*

Proof. By the previous argument, it suffices to show that each $\widetilde{A}_i$ is Schur stable for $i = 1, \dots, n$. 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

$$c_1 = -\frac{d_2\alpha_i^1 + \alpha_i^2(2\beta_i^2 + 3\zeta_i^2)}{2d'}, \qquad c_2 = -\frac{d_2(\frac{\tau^2}{2}\alpha_i^1 + \beta_i^1) + \frac{3\tau^2}{2}\alpha_i^2\beta_i^2 + \zeta_i^2(3\tau^2\alpha_i^2 + \beta_i^2)}{\tau d'},$$

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

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

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

$$\widetilde{A}_i = \begin{bmatrix} 1 - \frac{\alpha'\tau^2}{4d'} & \tau\Big(1 - \big(\frac{\alpha'\tau^2}{4} + \frac{\beta'}{2}\big)\frac{1}{d'}\Big) \\ -\frac{\alpha'\tau}{2d'} & 1 - \Big(\frac{\alpha'\tau^2}{2} + \beta'\Big)\frac{1}{d'} \end{bmatrix} \in \mathbb{R}^{2\times 2}.$$ It follows from [22, Proposition 5.1] that $\widetilde{A}_i$ is Schur stable, and so is $A_{\rm C}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$. $\square$

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 3.5.2. *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 $i = 1, \dots, n$,

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

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 3.5.1 that $A_{\mathrm{C}}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is Schur stable, i.e., spectral radius is strictly less than 1. Since the spectral radius of $A_{\mathrm{C}}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is continuous in α_i^s, β_i^s for all $s = 3, \dots, p$ and $i = 1, \dots, n$, a small perturbation to α_i^s, β_i^s for all $s = 3, \dots, p$ and $i = 1, \dots, n$ still leads to the Schur stable matrix A_{C}. This yields the desired result. $\square$

Based on the above 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 for $s = 2, \dots, p$, let

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

For a given MPC horizon $p \in \mathbb{N}$, suppose that the closed loop dynamic matrix $A_{\mathrm{C}}(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\zeta}, \tau)$ is Schur stable. Since the acceleration of the leading vehicle is bounded, i.e., $a_{\min} \leq u_0(k) \leq a_{\max}$ for all $k \in \mathbb{Z}_+$, the closed loop dynamics given by (3.20) or (3.24) 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 $k \to \infty$.

3.6 Numerical Results

3.6.1 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 ten (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 $\tau = 1s$, the reaction time $r = \tau = 1s$, the acceleration and deceleration limits $a_{\max} = 1.35m/s^2$ and $a_{\min} = -8m/s^2$, and the speed limits $v_{\max} = 27.78m/s$ and $v_{\min} = 10m/s$. The initial state of the platoon is $z(0) = z'(0) = 0$ and $v_i(0) = 25m/s$ for all $i = 0, 1, \ldots, n$. Further, the vehicle communication network is given by the cyclic-like graph, i.e., the bidirectional edges of the graph are $(1, 2), (2, 3), \ldots, (n-1, n) \in \mathcal{E}$.

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

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

For $p \geq 2$, 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

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

The above vectors $\boldsymbol{\alpha}^s, \boldsymbol{\beta}^s, \boldsymbol{\zeta}^s$ define the weight matrices $Q_{z,s}, Q_{z',s}, Q_{w,s}$ for $s = 1, \ldots, 5$, which further yield the closed loop dynamics matrix A_{C}; see the discussions below (3.23). 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 $p = 2, \ldots, 5$, respectively.

Discussion on the selection of MPC horizon. 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 $p = 1$. Further, 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 $s = 1, \ldots, p - 1$ 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, $p \leq 5$.

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 $k = 51s$ to $k = 54s$ with the deceleration $-2m/s^2$, and maintains a constant speed till $k = 100s$. After $k = 100s$, it restores to its original speed $25m/s$ with the acceleration $1m/s^2$.

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

• **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 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. Since the maximum of acceleration of this vehicle is close to $2m/s^2$, we choose $a_{\max} = 2m/s^2$. All the other parameters or physical limits remain the same. The experiment setup of this scenario is: $z_i(0) = 0m$, $v_i(0) = 25m/s$ for each i, and the time length is $45s$. 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 $\mathcal{N}(0.04, 0)$ is added to the first CAV, and the noise with the normal distribution $\mathcal{N}(0.02, 0)$ is added to each of the rest of the CAVs. Here a larger noise is imposed to the first CAV

Table 3.1: Parameters in Algorithm 9 for different MPC horizon p's

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}

since there are more noises and disturbances between the leading vehicle and the first CAV.

3.6.2 Performance of Fully Distributed Schemes and CAV Platooning Control

The generalized Douglas-Rachford splitting method based distributed algorithm (i.e., Algorithm 9) 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 below (3.18) for error tolerances and Table 3.1 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 3.4.2, 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 3.4.2. For the accelerated scheme, we let $\eta = 0.2$ and $\gamma_0 = 1.9/(0.8 \times \widehat{L})$.

Initial guess warm-up. For a given p, the augmented locally coupled optimization problem (3.16) 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 the assumption **A1**. 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. 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 9. This step can be efficiently computed since 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. Due to the decoupled structure of the problem (3.16), 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.

Table 3.2: Scenario 1: computation time and numerical accuracy

MPC horizon	Computation time per CAV (s)		Relative numer. error	
	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}

Table 3.3: Scenario 2: computation time and numerical accuracy

MPC horizon	Computation time per CAV (s)		Relative numer. error	
	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 3.4: Scenario 3: computation time and numerical accuracy

MPC horizon	Computation time per CAV (s)		Relative numer. error	
	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}

Table 3.5: Scenario 3: computation time and numerical accuracy with initial guess warm-up

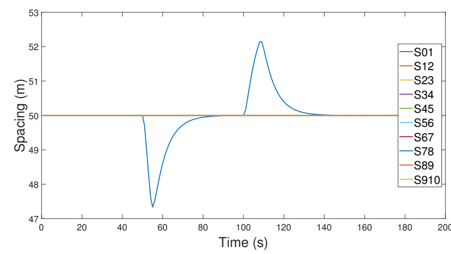
MPC horizon	Computation time per CAV (s)		Relative numer. error	
	Mean	Variance	Mean	Variance
$p = 1$	0.0243	0.0023	5.0×10^{-4}	7.0×10^{-7}
$p = 2$	0.0097	0.0017	2.6×10^{-3}	1.6×10^{-5}
$p = 3$	0.0579	0.0253	2.2×10^{-3}	1.1×10^{-5}
$p = 4$	0.1063	0.1103	3.7×10^{-3}	2.4×10^{-5}
$p = 5$	0.1258	0.1155	8.5×10^{-3}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

Performance of distributed schemes. Distributed algorithms are implemented on MATLAB and run on a computer of the following processor with 4 cores: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz and RAM: 16.0GB. We test the fully distributed Algorithm 9 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 Table 3.2- 3.4, respectively. The numerical performance for Scenario 3 under noises is similar to that without noise and is thus omitted.

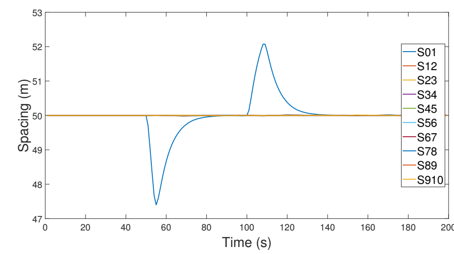
It is observed from the numerical results that when the MPC horizon p increases, more computation time is needed with mildly deteriorating numerical accuracy. This observation agrees with the discussion on the choice of p given in Section 3.6.1, which suggests a relatively small p for practical computation. Besides, we have tested the proposed initial guess warm-up technique on Scenario 3 for different p's using the same parameters and error tolerances for Algorithm 9; see Table 3.1. 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 p's. A summary of the numerical results is shown in Table 3.5. Compared with the results given in Table 3.4 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 9, 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 3.4.2. 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 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, e.g., $p \geq 3$, it takes much longer than 1 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 $p \geq 3$.

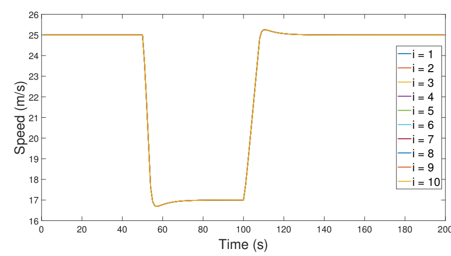
Performance of the CAV platooning control. We discuss the closed-loop performance of the proposed CAV platooning control for the three aforementioned scenarios with different MPC horizon p's. In each scenario, we evaluate the performance of the



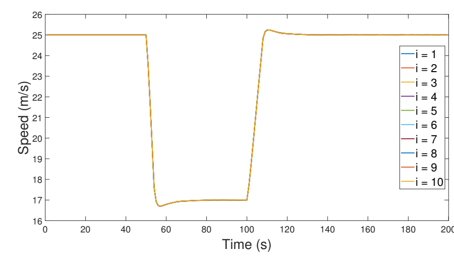
(a) Time history of spacing changes.



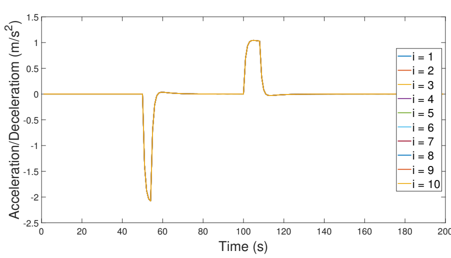
(b) Time history of spacing changes.



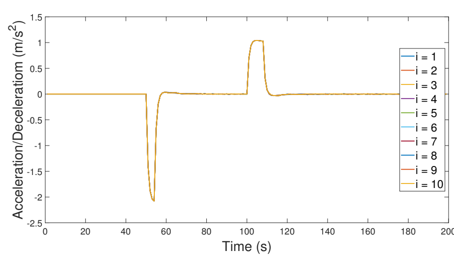
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

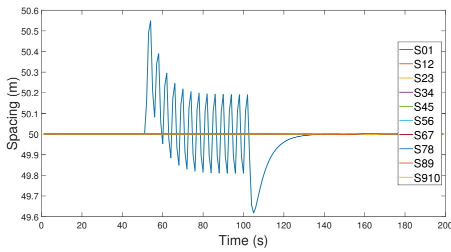


(e) Time history of control input.

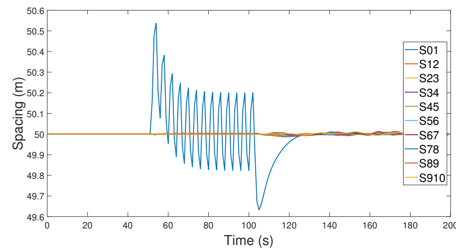


(f) Time history of control input

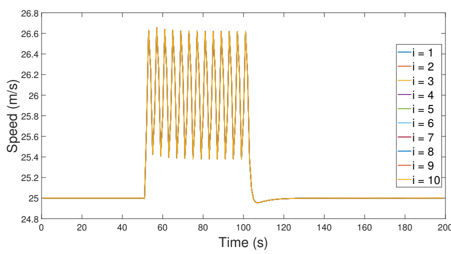
Figure 3.1: Scenario 1: the proposed CAV platooning control with $p = 1$ (left column) and $p = 5$ (right column).



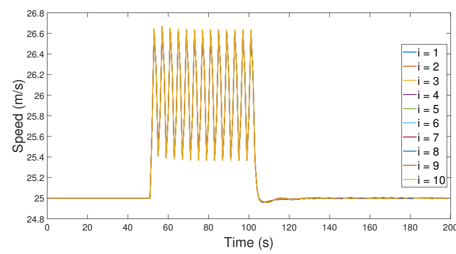
(a) Time history of spacing changes.



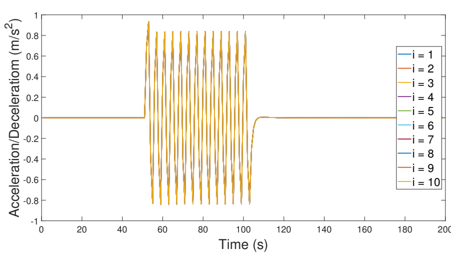
(b) Time history of spacing changes.



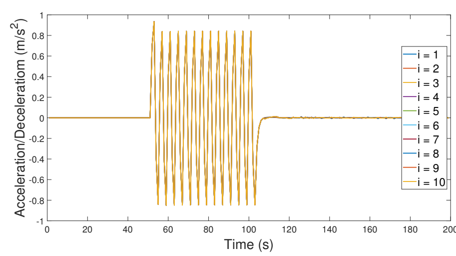
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

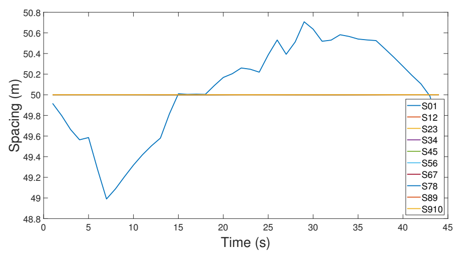


(e) Time history of control input.

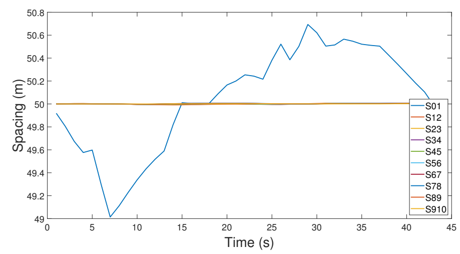


(f) Time history of control input

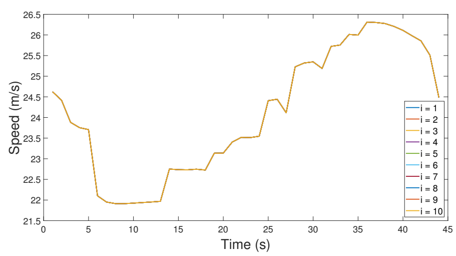
Figure 3.2: Scenario 2: the proposed CAV platooning control with $p = 1$ (left column) and $p = 5$ (right column).



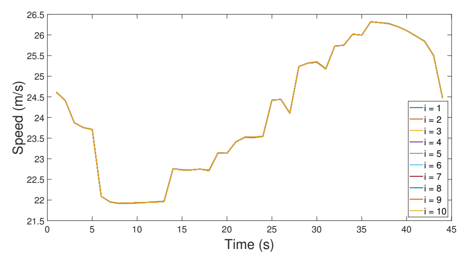
(a) Time history of spacing changes.



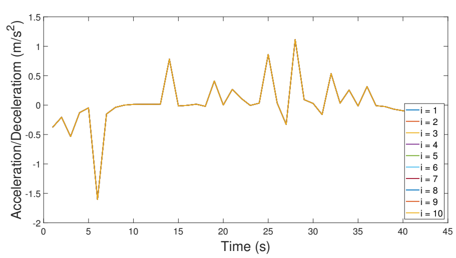
(b) Time history of spacing changes.



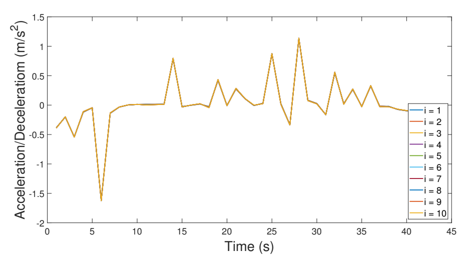
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

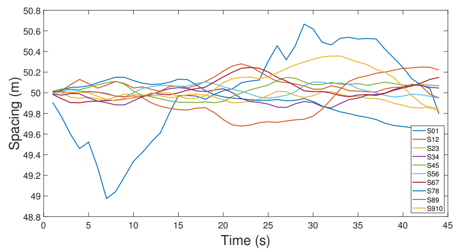


(e) Time history of control input.

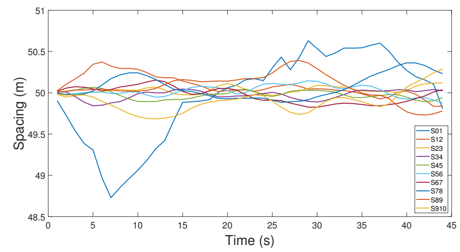


(f) Time history of control input

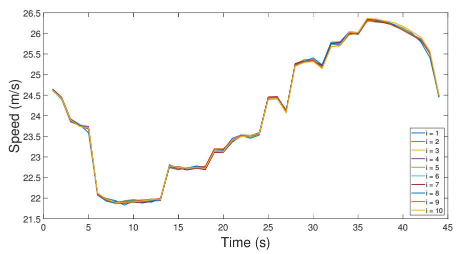
Figure 3.3: Scenario 3: the proposed CAV platooning control with $p = 1$ (left column) and $p = 5$ (right column).



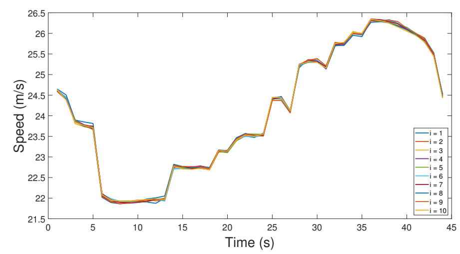
(a) Time history of spacing changes.



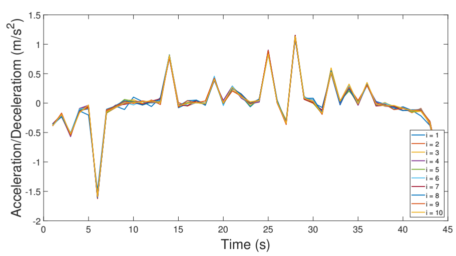
(b) Time history of spacing changes.



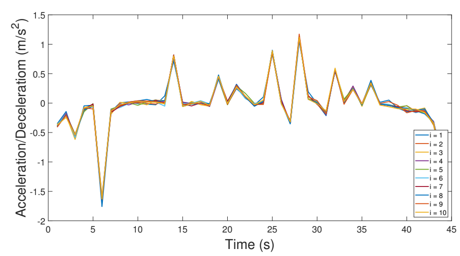
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.



(e) Time history of control input.



(f) Time history of control input

Figure 3.4: Scenario 3 under noises: the proposed CAV platooning control with $p = 1$ (left column) and $p = 5$ (right column).

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 $p = 1, 2, 3, 4, 5$. We present the closed-loop performance for $p = 1$ and $p = 5$ only for each scenario; see Figures 3.1-3.3 for (noise free) Scenarios 1-3 respectively, and Figure 3.4 for Scenario 3 with noises. In fact, it is observed from these figures (and the other tests) that there is little difference in control performance between $p = 1$ and a higher p, e.g., $p = 5$. We comment more on the closed-loop performance of each scenario as follows:

(i) Scenario 1. Figure 3.1 shows that the spacing between the uncontrolled leading vehicle and the first CAV, i.e., $S_{0,1}$, 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. Further, it takes about $35s$ for $S_{0,1}$ to converge to the steady state with the maximum spacing deviation $2.66m$. 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 exactly 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 3.2 displays that under the periodic acceleration/deceleration of the leading vehicle, the CAV platoon also demonstrates a "coordinated" motion 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 $0.22m$. Such oscillations quickly decay to zero within $30s$ 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 $45s$. In spite of such oscillations, it is seen from Figure 3.3 that only $S_{0,1}$ demonstrates small spacing variations with the maximum magnitude less than $1m$, 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 1-2.

(iv) Scenario 3 subject to noises. Figure 3.4 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 due to the noises. However, the variation of $S_{0,1}$ remains to be within $1m$, and the maximum deviation of each $S_{i-1,i}$ with $i \geq 2$ is less than $0.5m$. Further, the profiles of the CAV speed and control still demonstrate a nearly "coordinated" motion in spite of 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, e.g., [22], the proposed control scheme performs better since 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.

3.7 Summary

The present chapter 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 [63], closed loop stability analysis under non-constant spacing car following polices, and various robust issues under uncertainties and disturbances, e.g., communication network malfunction and failure, time delays in communication and control, model mismatch, and sensing errors.

CHAPTER IV

Nonconvex, Fully Distributed Optimization based CAV Platooning Control under Nonlinear Vehicle Dynamics

4.1 Introduction

Most of the existing research considers the linear vehicle dynamics [21], [22], [64], [84]. Although the linear vehicle dynamics is suitable for smaller passenger vehicles, nonlinear dynamic effects, e.g, aerodynamiec drag, friction, and rolling resistance, play a non-negligible role in trucks, heavy duty vehicles, and other types of CAVs. Motivated by the lack of research for nonlinear vehicle dynamics, this chapter aims to develop fully distributed optimization based and platoon centered CAV platooning under nonlinear vehicle dynamics over a general vehicle communication network. To achieve this goal, we propose a general p-horizon MPC model subject to the nonlinear vehicle dynamics of the CAVs and various physical or safety constraints. Several new challenges arise for the MPC horizon $p \geq 2$ when the nonlinear vehicle dynamics is considered. First, the underlying MPC optimization problem gives rise to a densely coupled, *nonconvex* optimization problem, where both the objective function and constraints are nonconvex. This is very different from the linear vehicle dynamics treated in Chapter 3, for which a convex MPC model is obtained so that various convex distributed optimization schemes can be used. Second, a local optimal solution to the MPC is characterized by a highly sophisticated nonlinear equation and does not attain a closed-form expression. Hence, the closed loop system is defined by a time-varying nonlinear dynamical system, whose right-hand side has no closed form expression, subject to non-vanishing external disturbances. These pose a difficulty in closed loop stability analysis and control design. To address these challenges, we exploit new techniques for distributed algorithm development and control analysis and design, which constitute main contributions of this chapter.

The major contributions of this chapter are summarized as follows:

(1) To develop fully distributed schemes for the nonconvex MPC optimization problem when $p \geq 2$, we first formulate the underlying densely coupled MPC optimization problem as a locally coupled, albeit nonconvex, optimization problem using a decomposition method developed for the linear CAV dynamics. Furthermore, we propose a sequential convex programming (SCP) [45] based distributed scheme to solve the locally coupled optimization problem. This SCP based scheme solves a sequence of convex, quadratically constrained quadratic programs (QCQPs) that approximate the original nonconvex program at each iteration; such a convex QCQP can be efficiently solved using (generalized) Douglas-Rachford method or other operator splitting methods [15] in the fully distributed manner. The SCP based distributed scheme converges to a stationary point, which often coincides or is close to an optimal solution, under mild assumptions.

(2) To analyze the closed loop dynamics, we first formulate the closed loop system as a tracking system defined by a time-varying, nonlinear dynamical system subject

to non-vanishing external disturbances. We apply stability theory of linear time-varying systems and Lyapunov theory for input-to-state stability to show that for all sufficiently small parameters pertaining to the nonlinear dynamics terms, the closed loop system is locally input-to-state stable provided that the corresponding linear closed loop dynamics under the linear vehicle dynamics (or equivalently when the abovementioned parameters are zero) is Schur stable.

(3) For real-time implementation, the computation time of the schemes should be less than *one second*. This is the major reason for the choice of Douglas-Rachford method based fully distributed schemes. Moreover, we develop initial warm-up techniques to further improve the computation time. Extensive numerical tests have been carried out for three types of CAV platoons in different scenarios: a homogeneous small-size CAV platoon, a heterogeneous medium-size CAV platoon, and a homogeneous large-size CAV platoon. The numerical results illustrate the effectiveness of the proposed distributed scheme and CAV platooning control under the nonlinear vehicle dynamics.

This chapter is organized as follows. Section 4.2 introduces the nonlinear vehicle dynamics, state and control constraints, and vehicle communication networks. Sequential feasibility and properties of the constraint sets are established in Section 4.3; these properties lay a ground for distributed optimization. A MPC model with a general prediction horizon p is proposed in Section 4.4 and is formulated as a nonconvex constrained optimization problem. Section 4.5 develops sequentially convex programming based fully distributed schemes for the densely coupled nonconvex MPC optimization problem. Control design and stability analysis for the closed loop dynamics is given in Section 4.6, and

Numerical tests and their results are presented in Section 4.7. Finally, summary is given in Section 4.8.

4.2 Vehicle Dynamics, Constraints, and Communication Topology

We use the similar terminology given in Section 3.2 for the linear dynamics. Here we introduce the following nonlinear vehicle dynamical model which in addition captures aerodynamic drag, friction, and rolling resistance [97]:

$$x_i(k+1) = x_i(k) + \tau v_i(k) + \frac{\tau^2}{2}\big(u_i(k) - c_{2,i} \cdot v_i^2(k) - c_{3,i} \cdot g\big), \tag{4.1a}$$

$$v_i(k+1) = v_i(k) + \tau\big(u_i(k) - c_{2,i} \cdot v_i^2(k) - c_{3,i} \cdot g\big), \tag{4.1b}$$

where $u_i(k)$ denotes the desired driving/braking acceleration treated as the control input. $c_{2,i} \cdot v_i^2(k)$ characterizes the deceleration due to aerodynamic drag with the coefficient $c_{2,i} > 0$, and $c_{3,i} \cdot g$ characterizes friction and rolling resistance with $g = 9.8m/s^2$ being the gravity constant and $c_{3,i} > 0$ being the rolling friction coefficient. For different vehicles, the coefficients $c_{2,i}, c_{3,i}$ can be different.

The coefficients $c_{2,i}$ and $c_{3,i}$ in model (4.1) are usually small for many different types of cars or road conditions. For example, $c_{2,i}$ typically ranges from $2.5 \times 10^{-4}/m$ to $4.5 \times 10^{-4}/m$, and $c_{3,i}$ typically ranges from 0.006 to 0.015 [97]. Since these coefficients are small, the nonlinear terms in (4.1) are often neglected in system-level studies. This yields the widely adopted double-integrator linear model given in linear dynamics (3.1).

The model (3.1) is suitable for small-size passenger cars, while model (4.1) can be used for medium-size or large-size vehicles, e.g., trucks and heavy-duty vehicles. These models are all well studied and widely accepted in the literature.

State and control constraints. Each vehicle in a platoon is subject to several important state and control constraints given in Section 3.2. Further it is possible that the vehicle platoon is possibly inhomogeneous, i.e., different vehicles have different control constraints. Hence, we let the acceleration/deceleration bounds $a_{i,\max}, a_{i,\min}$ as well as the vehicle length L_i and the reaction time r_i be different for different types of vehicles.

Communication network topology. The communication network topology for the nonlinear dynamics is assumed to be the same as that of linear dynamics. In particular, we assume that **A**.**1** holds true; see Section 3.2 for details.

4.3 Sequential Feasibility and Properties of Constraint Sets

The sequential feasibility has been shown for a CAV platoon under the linear vehicle dynamics [22]; see the definition of sequential feasibility on page 105. In what follows, we establish the sequential feasibility under the nonlinear vehicle dynamics (4.1).

4.3.1 Sequential Feasibility

For notational convenience, define $a_i(k, u_i(k)) := u_i(k) - c_{2,i} v_i^2(k) - c_{3,i} g$ for given $(v_i(k), u_i(k))$'s. It is noted that $u_0(k)$ is the actual acceleration of the leading vehicle at time k instead of its control. Hence, we set $c_{2,0} = c_{3,0} = 0$ for notational convenience. When k is fixed, we often write it as $a_i(u_i)$ to emphasize the dependence of a_i on u_i. Then the nonlinear vehicle dynamics given by (4.1) can be written as

$$x_i(k+1) \,=\, x_i(k) + \tau v_i(k) + \frac{\tau^2}{2} a_i(k, u_i(k)), \qquad v_i(k+1) \,=\, v_i(k) + \tau a_i(k, u_i(k)).$$

Given $(x_i, v_i)_{i=0}^n$ and u_0, we introduce the following constraint set on the control u subject to the nonlinear vehicle dynamics and the state and control constraints:

$$\mathcal{W}((x_i, v_i)_{i=0}^n, u_0) := \Big\{ u \in \mathbb{R}^n \,|\, a_{i,\min} \le u_i \le a_{i,\max}, v_{\min} \le v_i + \tau a_i(u_i) \le v_{\max}, \\ h_i(u) \le 0,\, i = 1, \ldots, n \Big\},$$

where the function h_i's are given by

$$h_i(u) := L_i + r_i(v_i + \tau a_i(u_i)) - \frac{(v_i + \tau a_i(u_i) - v_{\min})^2}{2a_{i,\min}} + (x_i - x_{i-1}) + \tau(v_i - v_{i-1}) \\ + \frac{\tau^2}{2}[a_i(u_i) - a_{i-1}(u_{i-1})], \qquad (4.2)$$

and by abusing notation, $a_i(u_i) := u_i - c_{2,i}v_i^2 - c_{3,i}g$ for each $i = 0, 1, \ldots, n$. Further, we define the following functions that describe the safety distances between two adjacent vehicles on their current position and speed:

$$p_j((x_i, v_i)_{i=0}^n) \;:=\; L_j + r_j v_j - \frac{(v_j - v_{\min})^2}{2a_{j,\min}} + (x_j - x_{j-1}), \qquad \forall\, j = 1, \ldots, n.$$

To simplify notation, we often write these functions as p_j when $(x_i, v_i)_{i=0}^n$ and u_0 are fixed in the subsequent development. The sequential feasibility holds if $\mathcal{W}((x_i, v_i)_{i=0}^n, u_0)$ is nonempty for any given feasible $(x_i, v_i)_{i=0}^n$ and u_0, i.e., $a_{0,\min} \le u_0 \le a_{0,\max}$, $v_{\min} \le v_0 \le v_{\max}$, $v_{\min} \le v_0 + \tau u_0 \le v_{\max}$, $v_{\min} \le v_i \le v_{\max}$ and $p_i((x_i, v_i)_{i=0}^n) \le 0$ for all $i = 1, \ldots, n$.

Proposition 4.3.1. *Consider the nonlinear vehicle dynamics given by (4.1). Suppose the nonnegative constants $c_{2,i}, c_{3,i}$ are such that $c_{2,i}v_{\max}^2 + c_{3,i}g \le a_{i,\max}$ and $r_i \ge \tau$ for each $i = 1, \ldots, n$. Then the system is sequentially feasible for an arbitrary feasible initial condition.*

Proof. We present some technical preliminaries first. For each given v_i satisfying $v_{\min} \le v_i \le v_{\max}$, define the continuous function $q_i : \mathbb{R} \to \mathbb{R}$ as

$$q_i(w) \ := \ v_i + \left(\frac{\tau}{2} + r_i\right) \cdot w - \frac{v_i - v_{\min}}{a_{i,\min}} w - \frac{\tau w^2}{2a_{i,\min}}, \quad \forall \ i = 1, \ldots, n. \tag{4.3}$$

Moreover, in view of the definition of h_i given by (4.2), we write h_i as $h_i(a_i, a_{i-1})$ by slightly abusing notation. We claim the following result:

Claim : Given any feasible $(x_i, v_i)_{i=0}^n$ and u_0, assume that there exist a_i's such that $v_i + \tau a_i \ge v_{\min}, \forall\, i = 0, 1, \ldots, n$. If $q_i(a_i) \le v_{\min}$, then $h_i(a_i, a_{i-1}) \le 0$, for all $i = 1, \ldots, n$.

To prove this claim, we first show that $\frac{v_{i-1} + (v_{i-1} + \tau a_{i-1})}{2} \ge v_{\min}$ for each $i = 1, \ldots, n$. Clearly, we deduce via $v_0 \ge v_{\min}$ and $v_0 + \tau a_0 \ge v_{\min}$ that $\frac{v_{i-1} + (v_{i-1} + \tau a_{i-1})}{2} \ge v_{\min}$ when $i = 1$. For each $i \ge 2$, it follows from the feasibility of $(x_i, v_i)_{i=0}^n$ that $v_{i-1} \ge v_{\min}$. This, along with the assumption that $v_{i-1} + \tau a_{i-1} \ge v_{\min}$, yields $\frac{v_{i-1} + (v_{i-1} + \tau a_{i-1})}{2} \ge v_{\min}$ for each $i \ge 2$. Further, for each $i = 1, \ldots, n$, we obtain, using $p_i \le 0$, that

$$\begin{aligned}
h_i(a_i, a_{i-1}) \ &= \ L + r_i(v_i + \tau a_i) - \frac{(v_i + \tau a_i - v_{\min})^2}{2a_{i,\min}} + (x_i - x_{i-1}) + \tau(v_i - v_{i-1}) \\
&\qquad + \frac{\tau^2}{2}(a_i - a_{i-1}) \\
&= \ \underbrace{\left[L + r_i v_i - \frac{(v_i - v_{\min})^2}{2a_{i,\min}} + (x_i - x_{i-1})\right]}_{=p_i} \\
&\quad + \tau\Big[r_i a_i + (v_i - v_{i-1}) + \frac{\tau}{2}(a_i - a_{i-1}) - \frac{\tau a_i^2}{2a_{i,\min}} - \frac{a_i}{a_{i,\min}}(v_i - v_{\min}) \Big] \\
&\le \ \tau\Big[v_i + \big(r_i + \frac{\tau}{2}\big)a_i - \frac{a_i}{a_{i,\min}}\big(v_i - v_{\min}\big) - \frac{\tau a_i^2}{2a_{i,\min}}
\end{aligned}$$

$$
\begin{aligned}
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad - \frac{v_{i-1} + (v_{i-1} + \tau a_{i-1})}{2} \Big] \\
&= \tau \Big[q_i(a_i) - \frac{v_{i-1} + (v_{i-1} + \tau a_{i-1})}{2} \Big] \\
&\leq \tau \Big[q_i(a_i) - v_{\min} \Big],
\end{aligned}
$$

where the last inequality follows from $\frac{v_{i-1}+(v_{i-1}+\tau a_{i-1})}{2} \geq v_{\min}$ for each $i = 1, \ldots, n$. Since $q_i(a_i) \leq v_{\min}$, we have $h_i(a_i, a_{i-1}) \leq 0$ for each $i = 1, \ldots, n$, completing the proof of the claim.

With the help of the above result, we prove that there exists $u = (u_i)_{i=1}^n \in \mathcal{W}((x_i, v_i)_{i=0}^n, u_0)$ for any feasible $(x_i, v_i)_{i=0}^n$ and u_0. For this purpose, consider the following choice of u_i's for a feasible v_i:

$$
u_i := \begin{cases} a_{i,\min} + c_{2,i} v_i^2 + c_{3,i} g, & \text{if } \ v_i + \tau a_{i,\min} \geq v_{\min} \\ \frac{v_{\min} - v_i}{\tau} + c_{2,i} v_i^2 + c_{3,i} g, & \text{if } \ v_i + \tau a_{i,\min} < v_{\min} \end{cases}, \quad \forall\, i = 1, \ldots, n. \tag{4.4}
$$

We show that the above $u = (u_i)_{i=1}^n \in \mathcal{W}((x_i, v_i)_{i=0}^n, u_0)$ by considering the following two cases for each fixed i:

(C.1) $v_i + \tau a_{i,\min} \geq v_{\min}$. In this case, $a_i(u_i) = a_{i,\min} < 0$.

- (a) Control constraint. In view of $c_{2,i}, c_{3,i} > 0$, we have $u_i > a_{i,\min}$. By the assumption that $c_{2,i} v_{\max}^2 + c_{3,i} g \leq a_{i,\max}$, we have $u_i = a_{i,\min} + c_{2,i} v_i^2 + c_{3,i} g \leq c_{2,i} v_{\max}^2 + c_{3,i} g \leq a_{i,\max}$. Thus $a_{i,\min} < u_i \leq a_{i,\max}$.
- (b) Speed constraint. By virtue of $a_i(u_i) = a_{i,\min}$, $\tau a_{i,\min} < 0$, and the assumption $v_i + \tau a_{i,\min} \geq v_{\min}$, we have $v_{\min} \leq v_i + \tau a_{i,\min} < v_i \leq v_{\max}$. Thus $v_{\min} \leq v_i + \tau a_i(u_i) = v_i + \tau a_{i,\min} < v_{\max}$.

(C.2) $v_i + \tau a_{i,\min} < v_{\min}$. In this case, $a_i(u_i) = \frac{v_{\min} - v_i}{\tau} \leq 0$.

(a) Control constraint. Since $c_{2,i}, c_{3,i} > 0$ and $v_{\min} > v_i + \tau a_{i,\min}$, we have $u_i > a_i(u_i) = \frac{v_{\min}-v_i}{\tau} > \frac{v_i+\tau a_{i,\min}-v_i}{\tau} = a_{i,\min}$. By the assumption that $c_{2,i}v_{\max}^2 + c_{3,i}g \le a_{i,\max}$ and using $a_i(u_i) = \frac{v_{\min}-v_i}{\tau} \le 0$, we also have $u_i \le c_{2,i}v_i^2 + c_{3,i}g \le a_{i,\max}$. Thus $a_{i,\min} < u_i \le a_{i,\max}$.

(b) Speed constraint. Since $v_i + \tau a_i(u_i) = v_{\min}$ and $a_i(u_i) \le 0$, we have $v_{\min} = v_i + \tau a_i(u_i) \le v_{\max}$.

These results show that the $a_i(u_i)$'s satisfy $v_i + \tau a_i(u_i) \ge v_{\min}, \forall\, i = 0, 1, \ldots, n$ such that all the assumptions in the claim proved above hold.

To show that the proposed u given by (4.4) satisfies the safety distance constraints, we consider cases (C.1) and (C.2) separately. For the former case, note that $q_i(a_i(u_i)) = q_i(a_{i,\min}) = r_i \cdot a_{i,\min} + v_{\min} < v_{\min}$. By the claim proved above, $h_i(a_i, a_{i-1}) \le 0$. For the latter case, in light of $v_i = v_{\min} - \tau a_i(u_i)$, we have

$$q_i(a_i(u_i)) = v_{\min} + (r_i - \tau) \cdot a_i(u_i) - \frac{v_i - v_{\min}}{a_{i,\min}} a_i(u_i) + \frac{\tau}{2} a_i(u_i) \Big[1 - \frac{a_i(u_i)}{a_{i,\min}} \Big].$$

Since $r_i \ge \tau$, $a_i(u_i) \le 0$ and $v_i - v_{\min} \ge 0$, we have $(r_i - \tau) \cdot a_i(u_i) \le 0$ and $\frac{-(v_i - v_{\min})}{a_{i,\min}} a_i(u) \le 0$. It has been shown in part (a) of (C.2) that $a_i(u) > a_{i,\min}$. This yields $\frac{\tau}{2} a_i(u_i)[1 - \frac{a_i(u_i)}{a_{i,\min}}] \le 0$. We thus conclude that $q_i(a_i(u_i)) \le v_{\min}$, leading to $h_i(a_i, a_{i-1}) \le 0$. Consequently, $u \in \mathcal{W}((x_i, v_i)_{i=0}^n, u_0)$. $\square$

4.3.2 Nonempty Interior of the Constraint Sets

Consider the non-polyhedral constraint set arising from the nonlinear vehicle dynamics subject to the control, speed, and safety distance constraints. We show that under mild assumptions, this constraint set has nonempty interior. This property is critical for the Slater's constraint qualification in optimization.

Proposition 4.3.2. *Consider the nonlinear vehicle dynamics (4.1). Suppose the non-negative constants* $c_{2,i}, c_{3,i}$ *are such that* $c_{2,i}v_{\max}^2 + c_{3,i}g < a_{i,\max}$ *and* $r_i \geq \tau$ *for each* $i = 1, \ldots, n$. *For any feasible* $(x_i, v_i)_{i=0}^n$ *and* u_0, *if* $v_0 > v_{\min}$ *and* $v_0 + \tau u_0 > v_{\min}$, *then* $\mathcal{W}((x_i, v_i)_{i=0}^n, u_0)$ *has nonempty interior.*

Proof. Given a feasible $(x_i, v_i)_{i=0}^n$ and u_0 such that $a_{0,\min} \leq u_0 \leq a_{0,\max}$, $v_{\min} < v_0 \leq v_{\max}$, $v_{\min} < v_0 + \tau a_0 \leq v_{\max}$, $v_{\min} \leq v_i \leq v_{\max}$ and $p_i \leq 0$ for all $i = 1, \ldots, n$, we show that for each $i = 1, \ldots, n$, there exists $\widehat{u}_i$ satisfying $a_{i,\min} < \widehat{u}_i < a_{i,\max}$, $v_{\min} < v_i + \tau a_i(\widehat{u}_i) < v_{\max}$, and $h_i(\widehat{u}_{i-i}, \widehat{u}_i) < 0$, where $\widehat{u} := (\widehat{u}_1, \ldots, \widehat{u}_n)^T \in \mathbb{R}^n$. We prove this result by induction on i as follows.

Consider $i = 1$ first. Let u_1 be given by (4.4). By the given assumptions and the proof of Proposition 4.3.1, we obtain the following:

(C.1) If $v_1 + \tau a_{1,\min} \geq v_{\min}$, then $a_{i,\min} < u_1 \leq c_{2,1}v_{\max}^2 + c_{3,1}g < a_{i,\max}$, $v_{\min} \leq v_1 + \tau a_1(u_1) = v_1 + \tau a_{1,\min} < v_{\max}$, and $q_1(a_1(u_1)) = r_1 a_{i,\min} + v_{\min} < v_{\min}$. Since $q_1(\cdot)$ given by (4.3) is continuous, there exists a constant $\varepsilon > 0$ such that $\widehat{u}_1 := u_1 + \varepsilon$ satisfies $a_{i,\min} < \widehat{u}_1 < a_{i,\max}$, $v_{\min} < v_1 + \tau a_1(\widehat{u}_1) < v_{\max}$, and $q_1(a_1(\widehat{u}_1)) < v_{\min}$. It follows from the Claim given in the proof of Proposition 4.3.1 that $h_1(\widehat{u}_1) \leq \tau[q_1(a_1(\widehat{u}_1)) - v_{\min}] < 0$.

(C.2) If $v_1 + \tau a_{1,\min} < v_{\min}$, then $a_1(u_1) = \frac{v_{\min} - v_1}{\tau}$, $a_{i,\min} < u_1 \leq c_{2,1}v_{\max}^2 + c_{3,1}g < a_{i,\max}$, $v_{\min} = v_1 + \tau a_1(u_1) < v_{\max}$, and $q_1(a_1(u_1)) \leq v_{\min}$. As shown in the proof for the

Claim in Proposition 4.3.1,

$$h_1(u_1) \leq \tau\Big[\, q_1(a_1(u_1)) - \frac{v_0 + (v_0 + \tau u_0)}{2} \,\Big] < \tau\Big[\, q_1(a_1(u_1)) - v_{\min} \,\Big] \leq 0,$$

where we use the assumptions that $v_0 > v_{\min}$ and $v_0+\tau u_0 > v_{\min}$. Hence, $h_1(u_1) < 0$. By the continuity of $q_1(\cdot)$ and $h_1(\cdot)$, we see that there exists a small constant $\varepsilon > 0$ such that $\widehat{u}_1 := u_1 + \varepsilon$ satisfies $a_{i,\min} < \widehat{u}_1 < a_{i,\max}$, $v_{\min} < v_1 + \tau a_1(\widehat{u}_1) < v_{\max}$, and $h_1(\widehat{u}_1) < 0$.

Now assume that for each i with $1 \leq i \leq n-1$, there exists $\widehat{u}_j$ satisfying $a_{i,\min} < \widehat{u}_j < a_{i,\max}$, $v_{\min} < v_j + \tau a_j(\widehat{u}_j) < v_{\max}$, and $h_j(\widehat{u}_j, \widehat{u}_{j-1}) < 0$ for all $j = 1, \ldots, i$. Consider $i+1$ as follows. As before, let u_{i+1} be given in (4.4), and consider two cases:

(C.1') $v_{i+1} + \tau a_{i,\min} \geq v_{\min}$. By the proof of Proposition 4.3.1 and a similar argument for (C.1) given above, we deduce that there exists a constant $\varepsilon > 0$ such that $\widehat{u}_{i+1} := u_{i+1} + \varepsilon$ satisfies the desired results.

(C.2') $v_{i+1} + \tau a_{i,\min} < v_{\min}$. Similarly, $a_{i,\min} < u_{i+1} < a_{i,\max}$, $v_{\min} = v_{i+1} + \tau a_{i+1}(u_{i+1}) < v_{\max}$, and $q_{i+1}(a_{i+1}(u_{i+1})) \leq v_{\min}$, where $a_{i+1}(u_{i+1}) = \frac{v_{\min} - v_{i+1}}{\tau}$. Moreover,

$$\begin{aligned} h_{i+1}(\widehat{u}_i, u_{i+1}) \quad &\leq \quad \tau\Big[\, q_{i+1}(a_{i+1}(u_{i+1})) - \frac{v_i + (v_i + \tau a_i(\widehat{u}_i))}{2} \,\Big] \\ &< \quad \tau\Big[\, q_{i+1}(a_{i+1}(u_{i+1})) - v_{\min} \,\Big] \leq 0, \end{aligned}$$

where the strict inequality follows from the assumption that $v_i \geq v_{\min}$ and the induction hypothesis $v_i + \tau a_i(\widehat{u}_i)) > v_{\min}$. Hence, $h_{i+1}(\widehat{u}_i, u_{i+1}) < 0$. Then by the similar argument for (C.2) above, we see that there exists a constant $\varepsilon > 0$ such that $\widehat{u}_{i+1} := u_{i+1} + \varepsilon$ yields the desired results.

Consequently, by the induction principle, there exists a vector $\widehat{u}$ in the interior of $\mathcal{W}((x_i, v_i)_{i=0}^n, u_0)$. $\square$

In light of the above result, we make the following assumptions throughout the rest of the chapter unless otherwise stated:

A.3 For each $i = 1, \ldots, n$, the nonnegative constants $c_{2,i}, c_{3,i}$ satisfy $c_{2,i} v_{\max}^2 + c_{3,i} g < a_{i,\max}$ and the reaction time r_i satisfies $r_i \geq \tau$. Further, $(v_0(k), u_0(k))$ is feasible with $v_0(k) > v_{\min}$ for all $k \in \mathbb{Z}_+$.

It will be shown in Corollary 4.4.1 that under this assumption, the constraint set of a general p-horizon model predictive control model has nonempty interior.

4.4 Model Predictive Control for CAV Platooning

We consider the model predictive control (MPC) approach for CAV platooning, and we use the same formulation given in Section 3.3. Precisely, the objective function is exactly the same as (3.4) and the constraints under the nonlinear vehicle dynamics (4.1) are similar to (3.5), and (3.6) but assume that the acceleration/deceleration bounds $a_{i,\max}$, and $a_{i,\min}$, the vehicle length L_i and the reaction time r_i are different for different vehicles i, for $i = 1, \ldots, n$. In particular, we assume that **A.2** holds true; see Section 3.3 for details.

More discussions on the class of weight matrices specified in **A.2** can be found in Section 3.3. We show below that under the assumption **A.3**, the constraint set of the p-horizon MPC model has nonempty interior at each k for an arbitrary MPC horizon $p \in \mathbb{N}$.

Corollary 4.4.1. *Suppose the assumption* **A.3** *holds. Then for any $p \in \mathbb{N}$, the constraint set of the p-horizon MPC optimization problem (3.4) has nonempty interior at each k.*

Proof. The proof is similar to [64, Corollary 3.1]. Fix an arbitrary $k \in \mathbb{Z}_+$. Since $v_0(k) > v_{\min}$, by Proposition 4.3.2, there exists a vector denoted by $\widehat{u}(k)$ in the interior of the set $\mathcal{W}((x_i(k), v_i(k))_{i=0}^n, u_0(k))$. Let $x_i(k+1)$ and $v_i(k)$ be generated by $\widehat{u}(k)$ (and $(x_i(k), v_i(k))_{i=0}^n, u_0(k))$. Since $v_0(k+1) > v_{\min}$, we deduce via Proposition 4.3.2 again that there exists a vector denoted by $\widehat{u}(k+1)$ in the interior of the constraint set $\mathcal{W}((x_i(k+1), v_i(k+1))_{i=0}^n, u_0(k+1))$. Continuing this process in p-steps, we derive the existence of an interior point in the constraint set of the p-horizon MPC model (3.4). $\square$

4.4.1 Constrained Optimization Model under the Nonlinear Vehicle Dynamics

In this subsection, we discuss the constrained optimization model (3.4) arising from the MPC at each time k under the nonlinear vehicle dynamics (4.1) with the positive parameters $c_{2,i}$ and $c_{3,i}$. For notational simplicity, define the parameter vectors $\boldsymbol{\varphi}_d := (c_{2,1}, \ldots, c_{2,n}) \in \mathbb{R}^n_+$ and $\boldsymbol{\varphi}_f := (c_{3,1}, \ldots, c_{3,n}) \in \mathbb{R}^n_+$, where the subscripts d and f denote the drag and friction respectively. Further, $\boldsymbol{\varphi} := (\boldsymbol{\varphi}_d, \boldsymbol{\varphi}_f) \in \mathbb{R}^{2n}_+$. For notational convenience, we set $c_{2,0} = c_{3,0} = 0$; this is because $u_0(k)$ is the actual acceleration of the leading vehicle instead of its control.

Consider the constrained MPC optimization model (3.4) at an arbitrary but fixed time $k \in \mathbb{Z}_+$. Let $\mathbf{u}(k) := (\mathbf{u}_1(k), \ldots, \mathbf{u}_n(k)) \in \mathbb{R}^{np}$ with $\mathbf{u}_i(k) := (u_i(k), \ldots, u_i(k+p-1)) \in \mathbb{R}^p$. Recall that for each $i = 1, \ldots, n$ and $j = 0, \ldots, p-1$,

$$a_i\big(k+j, u_i(k), \ldots, u_i(k+j)\big) \;=\; u_i(k+j) - c_{2,i} v_i^2(k+j) - c_{3,i} g,$$

where we note that $v_i(k+j)$ depends on $u_i(k), \ldots, u_i(k+j-1)$ for $j \geq 1$. Specifically, for $p > 1$,

$$
\begin{aligned}
a_i(k, u_i(k)) &= u_i(k) - c_{2,i} v_i^2(k) - c_{3,i} g, \\
a_i(k+1, u_i(k), u_i(k+1)) &= u_i(k+1) - c_{2,i} \big[v_i(k) + \tau a_i(k, u_i(k)) \big]^2 - c_{3,i} g, \\
\vdots \qquad\quad & \quad \vdots \qquad\qquad \vdots \\
a_i\big(k+p-1, u_i(k), \ldots, u_i(k+p-1)\big) &= u_i(k+p-1) - c_{2,i} \Big[v_i(k) \\
&\quad + \tau \sum_{s=0}^{p-2} a_i(k+s, u_i(k), \ldots, u_i(k+s)) \Big]^2 - c_{3,i} g
\end{aligned}
$$

By slightly abusing the notation, we may denote each $a_i\big(k+j, u_i(k), \ldots, u_i(k+j)\big)$ by $a_i(k+j, \mathbf{u}_i(k))$.

Define for each $i = 1, \ldots, n$ and $j = 0, 1, \ldots, p-1$,

$$
b_i(k+j, \mathbf{u}_{i-1}(k), \mathbf{u}_i(k)) \ := \ a_{i-1}(k+j, \mathbf{u}_{i-1}(k)) - a_i(k+j, \mathbf{u}_i(k)),
$$

where $a_0(k+j, \mathbf{u}_0(k)) := u_0(k)$ for all $j = 0, 1, \ldots, p-1$ due to $\mathbf{u}_0(k) := u_0(k) \cdot \mathbf{1}$. It follows from the nonlinear vehicle dynamics (4.1) that for each $i = 1, \ldots, n$ and $j = 1, \ldots, p$,

$$
\begin{aligned}
z_i(k+j) &= z_i(k) + j\tau z_i'(k) + \tau^2 \sum_{s=0}^{j-1} \frac{2(j-s)-1}{2} b_i(k+s, \mathbf{u}_{i-1}(k), \mathbf{u}_i(k)), \quad (4.5) \\
z_i'(k+j) &= z_i'(k) + \tau \sum_{s=0}^{j-1} b_i(k+s, \mathbf{u}_{i-1}(k), \mathbf{u}_i(k)). \quad (4.6)
\end{aligned}
$$

For a fixed $k \in \mathbb{Z}_+$, define for each $i = 1, \ldots, n$,

$$
\mathbf{a}_i(\mathbf{u}_i(k)) := \Big(a_i\big(k, u_i(k)\big), \ \ldots, \ a_i\big(k+p-1, u_i(k), \ldots, u_i(k+p-1)\big) \Big).
$$

In what follow, we often omit k in $\mathbf{u}_i(k)$ when k is fixed. Further, define the function $\mathbf{a} : \mathbb{R}^{np} \to \mathbb{R}^{np}$ as $\mathbf{a}(\mathbf{u}) := \big(\mathbf{a}_1(\mathbf{u}_1), \ldots, \mathbf{a}_n(\mathbf{u}_n)\big)$. Note that if $\boldsymbol{\varphi} = (\boldsymbol{\varphi}_d, \boldsymbol{\varphi}_f) = (c_{2,i}, c_{3,i})_{i=1}^n = 0$, then $\mathbf{a}(\mathbf{u}) = \mathbf{u}$ for all $\mathbf{u} \in \mathbb{R}^{np}$. We introduce more notation. Define the following matrices:

$$\overline{Q}_w := \mathrm{diag}\Big(Q_{w,1}, \ldots Q_{w,p} \Big) \in \mathbb{R}^{np\times np}, \qquad \mathbf{S}^{-1} := \mathrm{diag}\Big(\underbrace{S_n^{-1}, \ldots, S_n^{-1}}_{p-\text{copies}} \Big) \in \mathbb{R}^{np\times np}.$$

Furthermore, let $E \in \mathbb{R}^{np\times np}$ be the permutation matrix given by (3.9). $E = I_n$ when $p = 1$, and

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

Using these matrices, it is easy to verify that the following term in the objective function J in (3.4) satisfies

$$\left(\mathbf{S}^{-1} \begin{bmatrix} u(k) \\ \vdots \\ u(k+p-1) \end{bmatrix} \right)^T \overline{Q}_w \left(\mathbf{S}^{-1} \begin{bmatrix} u(k) \\ \vdots \\ u(k+p-1) \end{bmatrix} \right) = \mathbf{u}^T \underbrace{E^T \mathbf{S}^{-T} \overline{Q}_w \mathbf{S}^{-1} E}_{:=\Psi} \mathbf{u}.$$

where $\Psi \in \mathbb{R}^{np\times np}$ is symmetric PD when **A.2** holds. Therefore, the objective function J in the MPC model (3.4) becomes

$$\begin{aligned} J(\mathbf{u}) &= J(u(k), \ldots, u(k+p-1)) \\ &= \frac{1}{2}\Big[\sum_{s=1}^{p} z^T(k+s) Q_{z,s} z(k+s) + (z'(k+s))^T Q_{z',s} z'(k+s) \Big] + \frac{\tau^2}{2} \mathbf{u}^T \Psi\, \mathbf{u} \end{aligned}$$

$$= \frac{1}{2}\Big[\sum_{s=1}^{p} z^T(k+s)Q_{z,s}z(k+s) + (z'(k+s))^T Q_{z',s}z'(k+s)\Big] + \frac{\tau^2}{2}\mathbf{a}^T(\mathbf{u})\Psi\,\mathbf{a}(\mathbf{u})$$
$$+\frac{\tau^2}{2}\Big(\mathbf{u}^T\Psi\mathbf{u} - \mathbf{a}^T(\mathbf{u})\Psi\mathbf{a}(\mathbf{u})\Big).$$

In light of the expressions for $z(k+j)$ and $z'(k+j)$ given by (4.5)-(4.6), it follows from the similar argument in [64, Section 3.1] that the objective function

$$J(\mathbf{u}) \;=\; \frac{1}{2}\mathbf{a}^T(\mathbf{u})W\mathbf{a}(\mathbf{u}) + c^T\mathbf{a}(\mathbf{u}) + \gamma + \frac{\tau^2}{2}\Big(\mathbf{u}^T\Psi\mathbf{u} - \mathbf{a}^T(\mathbf{u})\Psi\mathbf{a}(\mathbf{u})\Big),$$

where $W \in \mathbb{R}^{np\times np}$, $c \in \mathbb{R}^{np}$, and $\gamma \in \mathbb{R}$. In fact, $W = E^T\mathbf{S}^{-T}\Theta\mathbf{S}^{-1}E$ for a symmetric PSD matrix Θ whose blocks are diagonal; see [64, Section 3.1] for the closed-form expression of W. In particular, under the assumption **A.3**, W is a positive definite (PD) matrix that only depends on $Q_{z,s}, Q_{z',s}$ and $Q_{w,s}$, $s = 1,\ldots,p$ [64, Lemma 3.1]. In addition, the linear term in $J(\mathbf{u})$ can be written as $c^T\mathbf{a}(\mathbf{u}) \;=\; \sum_{i=1}^n c_{\mathcal{I}_i}^T\mathbf{a}_i(\mathbf{u}_i)$, where $c_{\mathcal{I}_i}$ is the subvector of c corresponding to $\mathbf{a}_i(\mathbf{u}_i)$. By [64, Lemma 3.2], 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 $i = 1,\ldots,n-1$, $c_{\mathcal{I}_n}$ depends only on $z_n(k), z_n'(k)$, and only $c_{\mathcal{I}_1}$ depends on $u_0(k)$. These properties are important for developing fully distributed schemes later on.

To characterize the constraints, let the matrix $S_p \in \mathbb{R}^{p\times p}$ be defined in the same way as in (3.3) with n replaced by p, and $(S_p\mathbf{u}_i)_0 := 0$. Recall that for each $i = 1,\ldots,n$ and $j = 1,\ldots,p$,

$$v_i(k+j) = v_i(k) + \tau\sum_{s=0}^{j-1} a_i(k+s, \mathbf{u}_i(k)) = v_i(k) + \tau\big(S_p\,\mathbf{a}_i(\mathbf{u}_i)\big)_j.$$

Further, $x_{i-1}(k+j) - x_i(k+j) = z_i(k+j) + \Delta$ depends only on $\mathbf{u}_i(k)$ and $\mathbf{u}_{i-1}(k)$ as shown in (4.5). Hence, we see that for each $i = 1, \ldots, n$ and each $j = 1, \ldots, p$, the safety distance constraint is given by:

$$\big(H_i(\mathbf{u}_{i-1}(k), \mathbf{u}_i(k))\big)_j := L_i + r_i \cdot v_i(k+j) - \frac{(v_i(k+j) - v_{\min})^2}{2a_{i,\min}} - [x_{i-1}(k+j) - x_i(k+j)] \le 0.$$

Note that $H_1(\cdot)$ depends only on $\mathbf{u}_1(k)$ although it is written in the above form for notational convenience. Combining the above results, the MPC model (3.4) is formulated as the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad J(\mathbf{u}) := \tfrac{1}{2}\mathbf{a}^T(\mathbf{u})\big(W - \tau^2\Psi\big)\mathbf{a}(\mathbf{u}) + c^T\mathbf{a}(\mathbf{u}) + \gamma + \tfrac{\tau^2}{2}\mathbf{u}^T\Psi\mathbf{u}, \\ &\text{subject to} \quad \mathbf{u}_i \in \mathcal{X}_i, \quad v_{\min} \le v_i(k) + \tau\big(S_p\,\mathbf{a}_i(\mathbf{u}_i)\big)_s \le v_{\max}, \\ &\qquad\qquad (H_i(\mathbf{u}_{i-1}, \mathbf{u}_i))_s \le 0, \quad \forall\, i = 1, \ldots, n, \quad \forall\, s = 1, \ldots, p, \end{aligned} \tag{4.7}$$

where $\mathcal{X}_i := \{\mathbf{u}_i \in \mathbb{R}^p \,|\, a_{i,\min}\mathbf{1} \le \mathbf{u}_i \le a_{i,\max}\mathbf{1}\}$ for each $i = 1, \ldots, n$. It can be shown via the expressions of W and Ψ given in [64, Section 3.1] that $W - \tau^2\Psi$ is PSD. When $p = 1$, (4.7) is clearly a convex optimization problem. When $p > 1$, since all but the first entry functions of $\mathbf{a}_i(\cdot)$ are nonconvex in $\mathbf{u}_i$ for each i, it is easy to verify that the objective function J is nonconvex, and the velocity and safety distance constraints are nonconvex. Hence, when $p > 1$, (4.7) yields a nonconvex optimization problem. Since J is continuous, each $\mathcal{X}_i$ is compact, and the other constraints are defined by continuous functions, the optimization problem in (4.7) has a (possibly non-unique) solution. Moreover, the objective function J is densely coupled, and the safety distance constraint function $\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j$ not only depends on $\mathbf{u}_i$ but also on $\mathbf{u}_{i-1}$ of the $(i-1)$-th vehicle, and thus is locally coupled with its neighboring vehicles. This coupling structure, together with the nonconvexity of the optimization problem (4.7), leads to many challenges in developing fully distributed schemes.

4.5 Fully Distributed Algorithms for Coupled Nonconvex MPC Optimization Problem

In this section, we develop fully distributed algorithms for solving the underlying coupled, nonconvex optimization problem (4.7) at each time $k \in \mathbb{Z}_+$. To achieve this goal, several new techniques are exploited: the formulation of locally coupled (albeit nonconvex) optimization, sequential convex programming, and operator splitting methods.

4.5.1 Formulation of MPC Optimization Problem as Locally Coupled Optimization

Since the safety distance constraint of each vehicle i is coupled with its neighboring vehicle $(i-1)$ whereas the acceleration and velocity constraints are decoupled, the constraints of the MPC optimization problem (4.7) are locally coupled [26]. Motivated by distributed computation for locally coupled *convex* optimization [26], [64], we observe that (4.7) can be formulated as a locally coupled *nonconvex* optimization problem. see Section 1.2.1 for details.

The framework of a locally coupled optimization problem requires that both its objective function and constraints are expressed in a locally coupled manner satisfying the communication network topology constraint. However, the objective function in the underlying MPC optimization problem (4.7) is densely coupled. As indicated in Subsection 3.4.1 for convex optimization, this difficulty can be overcome by using certain matrix decomposition techniques. Specifically, it is shown in Proposition 3.4.1 that under the assumption **A.2**, the PSD or PD matrix $W \in \mathbb{R}^{np \times np}$ in (4.7) can be decomposed as $W = \sum_{s=1}^{n} \widetilde{W}^s$, where all $\widetilde{W}^s \in \mathbb{R}^{np \times np}$ are PSD and satisfy the locally coupled conditions.

Since $\overline{Q}_w$ is diagonal and PD, it follows from the similar argument in [64, Lemma 4.1] that the PD matrix $\Psi \in \mathbb{R}^{np \times np}$ can be decomposed in the similarly way. Specifically, there exist matrices $\widetilde{\Psi}^s$ such that $\Psi = \sum_{s=1}^{n} \widetilde{\Psi}^s$, where $\widetilde{\Psi}^s$'s satisfy the abovementioned conditions with $\widetilde{W}^s$ (resp. $\widehat{W}^s$) replaced by $\widetilde{\Psi}^s$ (resp. $\widehat{\Psi}^s$). By setting $\gamma \equiv 0$ in (4.7) without losing generality, the objective function $J(\mathbf{u})$ in (4.7) can be decomposed as

$$J(\mathbf{u}) \;=\; J_1(\mathbf{u}_1, \mathbf{u}_2) + \sum_{i=2}^{n-1} J_i(\mathbf{u}_{i-1}, \mathbf{u}_i, \mathbf{u}_{i+1}) + J_n(\mathbf{u}_{n-1}, \mathbf{u}_n),$$

where the functions J_i's on the right hand side are given by

$$\begin{aligned}
J_1(\mathbf{u}_1, \mathbf{u}_2) \;&:=\; \frac{1}{2}\begin{bmatrix} \mathbf{a}_1^T(\mathbf{u}_1) & \mathbf{a}_2^T(\mathbf{u}_2) \end{bmatrix} \left(\widehat{W}^1 - \tau^2 \widehat{\Psi}^1\right) \begin{bmatrix} \mathbf{a}_1(\mathbf{u}_1) \\ \mathbf{a}_2(\mathbf{u}_2) \end{bmatrix} + c_{\mathcal{I}_1}^T \mathbf{a}_1(\mathbf{u}_1) \\
&\quad + \frac{\tau^2}{2} \begin{bmatrix} \mathbf{u}_1^T & \mathbf{u}_2^T \end{bmatrix} \widehat{\Psi}^1 \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}, \\
J_i(\mathbf{u}_{i-1}, \mathbf{u}_i, \mathbf{u}_{i+1}) \;&:=\; \frac{1}{2}\begin{bmatrix} \mathbf{a}_{i-1}^T(\mathbf{u}_{i-1}) & \mathbf{a}_i^T(\mathbf{u}_i) & \mathbf{a}_{i+1}^T(\mathbf{u}_{i+1}) \end{bmatrix} \left(\widehat{W}^i - \tau^2 \widehat{\Psi}^i\right) \begin{bmatrix} \mathbf{a}_{i-1}(\mathbf{u}_{i-1}) \\ \mathbf{a}_i(\mathbf{u}_i) \\ \mathbf{a}_{i+1}(\mathbf{u}_{i+1}) \end{bmatrix} \\
&\quad + c_{\mathcal{I}_i}^T \mathbf{a}_i(\mathbf{u}_i) + \frac{\tau^2}{2} \begin{bmatrix} \mathbf{u}_{i-1} \\ \mathbf{u}_i \\ \mathbf{u}_{i+1} \end{bmatrix}^T \widehat{\Psi}^i \begin{bmatrix} \mathbf{u}_{i-1} \\ \mathbf{u}_i \\ \mathbf{u}_{i+1} \end{bmatrix}, \forall\ i = 2, \ldots, n-1, \qquad (4.8) \\
J_n(\mathbf{u}_{n-1}, \mathbf{u}_n) \;&:=\; \frac{1}{2}\begin{bmatrix} \mathbf{a}_{n-1}^T(\mathbf{u}_{n-1}) & \mathbf{a}_n^T(\mathbf{u}_n) \end{bmatrix} \left(\widehat{W}^n - \tau^2 \widehat{\Psi}^n\right) \begin{bmatrix} \mathbf{a}_{n-1}(\mathbf{u}_{n-1}) \\ \mathbf{a}_n(\mathbf{u}_n) \end{bmatrix}
\end{aligned}$$

$$+c_{\mathcal{I}_n}^T \mathbf{a}_n(\mathbf{u}_n) + \frac{\tau^2}{2} \begin{bmatrix} \mathbf{u}_{n-1}^T & \mathbf{u}_n^T \end{bmatrix} \widehat{\Psi}^n \begin{bmatrix} \mathbf{u}_{n-1} \\ \\ \mathbf{u}_n \end{bmatrix}.$$

In view of the assumption $\mathbf{A.1}$, the above decomposition of J satisfies the communication network topology constraint. Note that $\widehat{W}^i - \tau^2 \widehat{\Psi}^i$ may not be PSD or PD although $W - \tau^2 \Psi$ is PSD.

Remark 4.5.1. Another decomposition of J is as follows. Note that $V := W - \tau^2 \Psi$ is PSD, and it can be written as $V = E^T \mathbf{S}^{-T} \Phi \mathbf{S}^{-1} E$ for a symmetric PSD matrix Φ whose blocks are all diagonal. Therefore, the similar decomposition can be made to V whose corresponding $\widehat{V}^i$ is PSD. In view of the objective function J in (4.7), we can decompose J in a similar way by replacing $\widehat{W}^i - \tau^2 \widehat{\Psi}^i$ in the above decomposition by $\widehat{V}^i$.

In what follows, we use the above decomposition to formulate a locally coupled optimization problem by introducing copies of local variables. We consider the cyclic like network topology through this subsection, although the proposed formulation and schemes can be easily extended to other network topologies satisfying the assumption $\mathbf{A.1}$. In this case, $\mathcal{N}_1 = \{2\}$, $\mathcal{N}_n = \{n-1\}$, and $\mathcal{N}_i = \{i-1, i+1\}$ for $i = 2, \ldots, n-1$. Hence, each J_i in the decomposition of J can be written as $J_i(\mathbf{u}_i, (\mathbf{u}_j)_{j \in \mathcal{N}_i})$.

Recall that for each $i = 1, \ldots, n$, $\mathcal{X}_i := \{\mathbf{u}_i \in \mathbb{R}^p \,|\, a_{i,\min}\mathbf{1} \le \mathbf{u}_i \le a_{i,\max}\mathbf{1}\}$. Further, define

$$\mathcal{Y}_i \quad := \quad \left\{\mathbf{u}_i \in \mathbb{R}^p \,\middle|\, v_{\min} \le v_i(k) + \tau\big(S_p\, \mathbf{a}_i(\mathbf{u}_i)\big)_s \le v_{\max}, \quad \forall\, s = 1, \ldots, p \right\}, \qquad (4.9)$$

$$\mathcal{Z}_i \quad := \quad \left\{ (\mathbf{u}_{i-1}, \mathbf{u}_i) \in \mathbb{R}^p \times \mathbb{R}^p \,\middle|\, (H_i(\mathbf{u}_{i-1}, \mathbf{u}_i))_s \le 0, \qquad \forall\, s = 1, \ldots, p \right\}. \qquad (4.10)$$

As indicated before, $\mathcal{Z}_1$ depends only on $\mathbf{u}_1$ although it is written in the above form for notational convenience. Let $\boldsymbol{\delta}_S$ denote the indicator function of a closed set S. Define, for each $i = 1, \ldots, n$,

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

For each $i = 1, \ldots, n$, define $\widehat{\mathbf{u}}_i := \big(\mathbf{u}_i, (\mathbf{u}_{i,j})_{j\in\mathcal{N}_i}\big)$, where the new variables $\mathbf{u}_{i,j}$ represent the predicted values of $\mathbf{u}_j$ of vehicle j in the neighbor $\mathcal{N}_i$ of vehicle i, and let $\widehat{\mathbf{u}} := (\widehat{\mathbf{u}}_1, \ldots, \widehat{\mathbf{u}}_n) \in \mathbb{R}^N$. Define the consensus subspace

$$\mathcal{A} \,:=\, \Big\{ \widehat{\mathbf{u}} \in \mathbb{R}^N \,\big|\, \mathbf{u}_{i,j} = \mathbf{u}_j, \ \forall\, (i,j) \in \mathcal{E} \Big\}.$$

Then the underlying optimization problem (4.7) can be equivalently written as the following locally coupled optimization problem:

$$\min_{\widehat{\mathbf{u}}} \ \sum_{i=1}^{n} \widehat{J}_i(\widehat{\mathbf{u}}_i), \qquad \text{subject to} \qquad \widehat{\mathbf{u}} \in \mathcal{A}. \tag{4.11}$$

In the above formulation, the functions $\widehat{J}_i$'s are decoupled, and the consensus constraint $\mathcal{A}$ gives rise to the only coupling in this formulation.

4.5.2 Sequential Convex Programming and Operator Splitting Method based Fully Distributed Algorithms for the MPC Optimization Problem

When the MPC horizon $p = 1$, the underlying MPC optimization problem (4.7) or (4.11) is a convex quadratically constrained quadratic program (QCQP), for which the fully distributed schemes developed in [64] can be applied. We consider $p > 1$ from now on.

In this case, the underlying MPC optimization problem (4.7) or (4.11) yields a non-convex minimization problem whose objective function and constraints are non-convex, whereas the coefficients $c_{2,i} > 0$ and $c_{3,i} > 0$ defining the nonlinearities are small. Therefore, it is expected that an optimal solution under the nonlinear vehicle dynamics is "close" to that under the linear vehicle dynamics. The latter solution, which can be obtained using fully distributed schemes [64], may be used as an initial guess for a distributed scheme for the nonlinear vehicle dynamics. We formally discuss this observation as follows.

The general theory of perturbed optimization can be found in the monograph [7]; we consider a special case here. Let $f : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}$ and $g_i : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}$ with $i = 1, \ldots, m$ be all continuous functions. Let $\Omega \subset \mathbb{R}^n$ be a compact set, and $\Theta \subseteq \mathbb{R}^q$ be a set of parameter vectors that contain the zero vector. Fix a parameter vector $\theta \in \Theta$, and define the parameter dependent constraint set

$$\mathcal{W}_\theta := \big\{ x \in \mathbb{R}^n \,\big|\, g_i(x, \theta) \le 0, \ \forall\, i = 1, \ldots, m \big\}.$$

We assume that for each parameter vector $\theta \in \Theta$, the set $\Omega \cap \mathcal{W}_\theta$ is nonempty. Since $g_i(\cdot, \theta)$ is continuous for a given θ, $\Omega \cap \mathcal{W}_\theta$ is a nonempty compact set such that for a fixed $\theta \in \Theta$, the minimization problem

$$P_\theta : \quad \min_{x \in \Omega \cap \mathcal{W}_\theta} f(x, \theta)$$

has a nonempty closed solution set denoted by $\mathcal{S}_\theta$.

For each $x \in \Omega \cap \mathcal{W}_0$, define the index set $\mathcal{I}(x) := \{ i \mid g_i(x, 0) = 0 \} \subseteq \{1, \ldots, m\}$, which corresponds to the index set of active inequality constraints. We introduce the following assumption on $\Omega \cap \mathcal{W}_0$.

A.4 For any $x^\diamond \in \Omega \cap \mathcal{W}_0$ whose corresponding $\mathcal{I}(x^\diamond)$ is nonempty, there exists a sequence (w^ℓ) in $\Omega \cap \mathcal{W}_0$ such that: (i) for each ℓ, $g_i(w^\ell, 0) < 0$ for all $i = 1, \ldots, m$; and (ii) (w^ℓ) converges to $x^\diamond$.

The following lemma presents a sufficient condition related to the Slater's condition for **A.4** to hold.

Lemma 4.5.1. *Suppose each $g_i(\cdot, 0)$ is a convex function, $\Omega \cap \mathcal{W}_0$ is a convex compact set, and there exists $z \in \Omega \cap \mathcal{W}_0$ such that $g_i(z, 0) < 0$ for all $i = 1, \ldots, m$. Then* **A.4** *holds.*

Proof. Let $z \in \Omega \cap \mathcal{W}_0$ be such that $g_i(z, 0) < 0$ for each $i = 1, \ldots, m$, and consider $x \in \Omega \cap \mathcal{W}_0$ whose index set $\mathcal{I}(x)$ is nonempty. Therefore, for any $\lambda \in (0, 1]$, $g_i(x + \lambda(z - x), 0) = g_i(\lambda z + (1 - \lambda)x, 0) \leq \lambda g_i(z, 0) + (1 - \lambda) g_i(x, 0) \leq \lambda g_i(z, 0) < 0$ for each $i = 1, \ldots, m$. Therefore, **A.4** holds. $\square$

Proposition 4.5.1. *Suppose P_0 has the unique minimizer x_*, i.e., $\mathcal{S}_0 = \{x_*\}$. Then under the abovementioned assumptions (including* **A.4***), for any $\varepsilon > 0$, there exists $\eta > 0$ such that for all $\theta \in \Theta$ with $\|\theta\| \leq \eta$, $\sup_{z \in \mathcal{S}_\theta} \|z - x_*\| < \varepsilon$.*

Proof. Suppose not. Then there exist $\varepsilon_0 > 0$ and a sequence (θ^k) in Θ with $\|\theta^k\| \to 0$ such that for each k, there exist $z^k \in \mathcal{S}_{\theta^k}$ with $\|z^k - x_*\| \geq \varepsilon_0$. Since z^k belongs to the compact set Ω, (z^k) has a convergent subsequence whose limit $z^* \in \Omega$ satisfies $\|z^* - x_*\| \geq \varepsilon$. Without loss of generality, we may assume that this subsequences is (z^k) itself. Since $g_i(z^k, \theta^k) \leq 0$ for all $i = 1, \ldots, m$, it follows from the continuity of each $g_i(\cdot, \cdot)$ that $g_i(z^*, 0) \leq 0$. Hence, $z^* \in \Omega \cap \mathcal{W}_0$. Consider a fixed but arbitrary $x \in \Omega \cap \mathcal{W}_0$. Hence, either $\mathcal{I}(x)$ is empty or $\mathcal{I}(x)$ is nonempty. For the former case, we deduce via the continuity of $g_i(x, \cdot)$ that $g_i(x, \theta^k) < 0, i = 1, \ldots, m$ for all large k. Hence, $x \in \Omega \cap \mathcal{W}_{\theta^k}$ for all large k.

This shows that $f(x, \theta^k) \geq f(z^k, \theta^k)$ for all large k, and therefore $f(x, 0) \geq f(z^*, 0)$. For the latter case, it follows from the assumption **A.4** on $\Omega \cap \mathcal{W}_0$ that there exists a sequence (w^ℓ) in $\Omega \cap \mathcal{W}_0$ which converge to x such that $g_i(w^\ell, 0) < 0$ for all ℓ and all $i = 1, \ldots, m$. By the continuity of g_i's and the fact that $(\theta^k) \to 0$, we see that for $\ell = 1$, there exists an index s_1 such that $g_i(w^1, \theta^{s_1}) < 0$ for all $i = 1, \ldots, m$. Then for $\ell = 2$, there exists an index s_2 with $s_2 > s_1$ such that $g_i(w^2, \theta^{s_2}) < 0$ for all $i = 1, \ldots, m$. Continuing this process, we obtain a strictly increasing index sequence (s_ℓ) such that for each ℓ, $g_i(w^\ell, \theta^{s_\ell}) < 0$ for all $i = 1, \ldots, m$. Hence, each $w^\ell \in \Omega \cap \mathcal{W}_{\theta^{s_\ell}}$ such that $f(w^\ell, \theta^{s_\ell}) \geq f(z^{s_\ell}, \theta^{s_\ell})$. Since (θ^{s_ℓ}) is a subsequence of (θ^k) and (z^{s_ℓ}) is a subsequence of (z^k), we have that $\theta^{s_\ell} \to 0$ and $z^{s_\ell} \to z^*$. This leads to $f(x, 0) \geq f(z^*, 0)$. Consequently, $f(x, 0) \geq f(z^*, 0)$ for all $x \in \Omega \cap \mathcal{W}_0$. This implies that z^* is a minimizer of P_0. Since x_* is the unique minimizer of P_0, we must have $z^* = x_*$, yielding a contradiction to $\|z^* - x_*\| \geq \varepsilon_0$. $\square$

We apply this proposition to the optimization problem (4.7). Recall that the parameter vector $\boldsymbol{\varphi} = (\boldsymbol{\varphi}_d, \boldsymbol{\varphi}_f) = (c_{2,i}, c_{3,i})_{i=1}^n \in \mathbb{R}^{2n}_+$. To emphasize the dependence of the objective function J on $\boldsymbol{\varphi}$, we write it as $J(\mathbf{u}, \boldsymbol{\varphi})$ by abusing the notation. Further, the constraints in (4.7) can be written as $\mathcal{X} \cap \mathcal{Y} \cap \mathcal{Z}$, where $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ is a convex and compact set, and $\mathcal{Y} \cap \mathcal{Z} = \{\mathbf{u} \,|\, g_i(\mathbf{u}, \boldsymbol{\varphi}) \leq 0,\, i = 1, \ldots, m\}$ for some real-valued functions g_i, which also depend on $\boldsymbol{\varphi}$. It is shown in [64] that when $\boldsymbol{\varphi} = 0$, $J(\mathbf{u}, 0)$ is a strongly convex quadratic function, and each $g_i(\mathbf{u}, 0)$ is an affine or a convex quadratic function. Hence, when $\boldsymbol{\varphi} = 0$, (4.7) becomes a convex optimization problem which attains a unique optimal solution $\mathbf{u}_{*,0}$. Further, when $r_i \geq \tau$ for all i and $v_0(k) > v_{\min}$, this convex optimization problem has non-empty interior [64, Corollary 3.1] such that **A.4** holds by Lemma 4.5.1. Therefore, letting $\mathcal{S}_{\boldsymbol{\varphi}}$ denote the solution set of (4.7) corresponding to the parameter vector $\boldsymbol{\varphi}$, we obtain the following corollary from Proposition 4.5.1.

Corollary 4.5.1. *Consider the optimization problem (4.7) with the parameter vector* $\boldsymbol{\varphi} \in \mathbb{R}^{2n}_{+}$ *at time* k*. Suppose* $r_i \geq \tau$ *for all* i *and* $v_0(k) > v_{\min}$*. Then for any* $\varepsilon > 0$*, there exists* $\eta > 0$ *such that for all* $\boldsymbol{\varphi} \in \mathbb{R}^{2n}_{+}$ *with* $\|\boldsymbol{\varphi}\| \leq \eta$*,* $\sup_{\mathbf{u} \in \mathcal{S}_{\boldsymbol{\varphi}}} \|\mathbf{u} - \mathbf{u}_{*,0}\| < \varepsilon$*.*

To solve the coupled non-convex optimization problem (4.7) with $\boldsymbol{\varphi} \neq 0$, we exploit the sequential convex programming (SCP) method [45]. To be self-contained, we provide a brief description of the SCP method for an important special case as follows. Consider the nonlinear program

$$(P') : \quad \min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad x \in \mathcal{P}, \quad g_i(x) - r_i(x) \leq 0, \quad \forall\, i = 1, \ldots, \ell, \tag{4.12}$$

where $\mathcal{P} \subseteq \mathbb{R}^n$ is a closed convex set, f and each g_i are C^1 (but not necessarily convex) functions, and each r_i is a convex C^1-function. We assume that ∇f and ∇g_i are Lipschitz on $\mathcal{P}$, i.e. there exist constants $L_f > 0$ and $L_{g_i} > 0$ such that $\|\nabla f(x) - \nabla f(x')\|_2 \leq L_f \|x - x'\|_2$ and $\|\nabla g_i(x) - \nabla g_i(x')\|_2 \leq L_{g_i} \|x - x'\|_2$ for all $x, x' \in \mathcal{P}$ and $i = 1, \ldots, \ell$. Let $\widehat{x}$ be a feasible point of (P'), i.e., $\widehat{x} \in \mathcal{P}'$ and $g_i(\widehat{x}) - r_i(\widehat{x}) \leq 0,\ i = 1, \ldots, \ell$. Consider an approximation of the constraint set of (P') at $\widehat{x}$:

$$\begin{aligned}
&\mathcal{C}(\widehat{x}, \{\nabla g_i(\widehat{x})\}_{i=1}^{\ell}, \{\nabla r_i(\widehat{x})\}_{i=1}^{\ell}) \\
&\quad := \Big\{ z \in \mathcal{P} \,|\, g_i(\widehat{x}) + \nabla g_i(\widehat{x})^T (z - \widehat{x}) + \frac{L_{g_i}}{2} \|z - \widehat{x}\|_2^2 - [r_i(\widehat{x}) + \nabla r_i(\widehat{x})^T (z - \widehat{x})] \leq 0, \\
&\qquad\qquad i = 1, \ldots, \ell \Big\}.
\end{aligned}$$

It is shown in [45, Lemma 3.3] that $\mathcal{C}(\widehat{x}, \{\nabla g_i(\widehat{x})\}_{i=1}^{\ell}, \{\nabla r_i(\widehat{x})\}_{i=1}^{\ell})$ is a nonempty closed convex set. The following lemma provides a simple sufficient condition for the Slater's condition to hold for the approximated constraint set; this condition is useful for convergence analysis of the SCP scheme.

Lemma 4.5.2. *Given a feasible point $\widehat{x}$ of (P'), suppose $\mathcal{C}(\widehat{x}, \{\nabla g_i(\widehat{x})\}_{i=1}^{\ell}, \{\nabla r_i(\widehat{x})\}_{i=1}^{\ell})$ is not singleton. Then the Slater's condition holds for $\mathcal{C}(\widehat{x}, \{\nabla g_i(\widehat{x})\}_{i=1}^{\ell}, \{\nabla r_i(\widehat{x})\}_{i=1}^{\ell})$, i.e., there exists $\widehat{z} \in \mathcal{P}$ such that $g_i(\widehat{x}) + \nabla g_i(\widehat{x})^T(\widehat{z} - \widehat{x}) + \frac{L_{g_i}}{2}\|\widehat{z} - \widehat{x}\|_2^2 - [r_i(\widehat{x}) + \nabla r_i(\widehat{x})^T(\widehat{z} - \widehat{x})] < 0, \forall\, i = 1, \ldots, \ell$.*

Proof. Clearly, $\widehat{x} \in \mathcal{C}(\widehat{x}, \{\nabla g_i(\widehat{x})\}_{i=1}^{\ell}, \{\nabla r_i(\widehat{x})\}_{i=1}^{\ell})$. Since $\mathcal{C}(\widehat{x}, \{\nabla g_i(\widehat{x})\}_{i=1}^{\ell}, \{\nabla r_i(\widehat{x})\}_{i=1}^{\ell})$ is not singleton, there exists $x' \in \mathcal{C}(\widehat{x}, \{\nabla g_i(\widehat{x})\}_{i=1}^{\ell}, \{\nabla r_i(\widehat{x})\}_{i=1}^{\ell})$ and $x' \neq \widehat{x}$. Let $\mathcal{I}(\widehat{x}) := \{i \,|\, g_i(\widehat{x}) - r_i(\widehat{x}) = 0\}$ denote the index set of active constraints at $\widehat{x}$ of (P). Thus $x' \in \mathcal{P}$ and $[\nabla g_i(\widehat{x}) - \nabla r_i(\widehat{x})]^T(x' - \widehat{x}) + \frac{L_{g_i}}{2}\|x' - \widehat{x}\|_2^2 \leq 0$ for all $i \in \mathcal{I}(\widehat{x})$. Since $\frac{L_{g_i}}{2}\|x' - \widehat{x}\|_2^2 > 0$ for all $i \in \mathcal{I}(\widehat{x})$, we have $[\nabla g_i(\widehat{x}) - \nabla r_i(\widehat{x})]^T(x' - \widehat{x}) < 0$ for all $i \in \mathcal{I}(\widehat{x})$. Let $d := x' - \widehat{x}$. Since $\mathcal{P}$ is a closed convex set, $d \in \mathcal{T}_{\mathcal{P}}(\widehat{x})$ [61, Lemma 3.13], where $\mathcal{T}_{\mathcal{P}}(\widehat{x})$ denotes the tangent cone of $\mathcal{P}$ at $\widehat{x}$. This shows that the (generalized) MFCQ holds. It thus follows from [45, Proposition 3.5] that the Slater's condition holds for $\mathcal{C}(\widehat{x}, \{\nabla g_i(\widehat{x})\}_{i=1}^{\ell}, \{\nabla r_i(\widehat{x})\}_{i=1}^{\ell})$. $\square$

The SCP scheme solves (P') in (4.12) as follows [45]: consider an approximation of the objective function f for a given feasible point $\widehat{x}$: $\widetilde{f}(z; \widehat{x}) := f(\widehat{x}) + [\nabla f(\widehat{x})]^T(z - \widehat{x}) + \frac{L_f}{2}\|z - \widehat{x}\|_2^2$. Clearly, $\widetilde{f}$ is a strongly convex function in z. At each step, the SCP scheme solves the convex optimization problem at x^k using the convex approximation $\widetilde{f}(\cdot; x^k)$ over the approximating convex constraint set $\mathcal{C}(x^k, \{\nabla g_i(x^k)\}_{i=1}^{\ell}, \{\nabla r_i(x^k)\}_{i=1}^{\ell})$ to generate a unique optimal solution x^{k+1}. It then updates the gradients ∇f, ∇g_i, and ∇r_i using x^{k+1}, and formulates another convex optimization problem and solves it again. It is shown in [45, Theorem 3.4] that any accumulation point of the sequence (x^k) generated by the SCP scheme is a KKT point of (P'), provided that the accumulation point x^* satisfies the Slater's condition for $\mathcal{C}(x^*, \{\nabla g_i(x^*)\}_{i=1}^{\ell}, \{\nabla r_i(x^*)\}_{i=1}^{\ell})$.

We now apply the SCP scheme to develop a fully distributed scheme for the non-convex MPC optimization problem (4.7). Consider the locally coupled formulation (4.11) of the MPC optimization problem (4.7). Recall that $\widehat{\mathbf{u}}_i := \big(\mathbf{u}_i, (\mathbf{u}_{i,j})_{j\in\mathcal{N}_i}\big)$, and $\widehat{\mathbf{u}} := (\widehat{\mathbf{u}}_1, \ldots, \widehat{\mathbf{u}}_n)$. For each $i = 1, \ldots, n$, it follows from the velocity constraint $\mathcal{Y}_i$ in (4.9) and the safety distance constraint $\mathcal{Z}_i$ in (4.10) that there are real-vauled smooth functions $g_{i,s}$ and convex quadratic functions $r_{i,s}$ for $s = 1, \ldots, 3p$ such that $\widehat{\mathbf{u}}_i \in \mathcal{Y}_i \cap \mathcal{Z}_i$ if and only if $g_{i,s}(\widehat{\mathbf{u}}_i) - r_{i,s}(\widehat{\mathbf{u}}_i) \leq 0$ for $s = 1, \ldots, 3p$; specific choices of $g_{i,s}$ and $r_{i,s}$ are given in Sections 4.5.3 and 4.7.2. In view of the real-valued objective function $J(\widehat{\mathbf{u}}) = \sum_{i=1}^n J_i(\widehat{\mathbf{u}}_i)$, the problem (4.11) becomes

$$
\begin{aligned}
&\min \sum_{i=1}^{n} J_i(\widehat{\mathbf{u}}_i),\\
&\quad \text{subject to } \widehat{\mathbf{u}} \in \mathcal{A}, \widehat{\mathbf{u}}_i \in \mathcal{X}_i, \quad g_{i,s}(\widehat{\mathbf{u}}_i) - r_{i,s}(\widehat{\mathbf{u}}_i) \leq 0, \quad \forall\, i = 1, \ldots, n, \quad s = 1, \ldots, 3p.
\end{aligned}
$$

Recall that $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ is a convex compact set. Since $\mathcal{X}$ is compact and $\mathcal{A}$ is the consensus subspace, it is easy to show that there are positive Lipschitz constants L_{J_i} and $L_{g_{i,s}}$ for the gradients of J_i and $g_{i,s}$ on $\mathcal{A} \cap \mathcal{X}$, i.e., for all $\widehat{\mathbf{u}}, \widehat{\mathbf{u}}' \in \mathcal{A} \cap \mathcal{X}$,

$$
\begin{aligned}
\|\nabla J_i(\widehat{\mathbf{u}}_i) - \nabla J_i(\widehat{\mathbf{u}}_i')\|_2 &\leq L_{J_i} \cdot \|\widehat{\mathbf{u}}_i - \widehat{\mathbf{u}}_i'\|_2, \qquad \forall\, i = 1, \ldots, n,\\
\|\nabla g_{i,s}(\widehat{\mathbf{u}}_i) - \nabla g_{i,s}(\widehat{\mathbf{u}}_i')\|_2 &\leq L_{g_{i,s}} \cdot \|\widehat{\mathbf{u}}_i - \widehat{\mathbf{u}}_i'\|_2, \qquad \forall\, i = 1, \ldots, n, \quad s = 1, \ldots, 3p.
\end{aligned}
$$

To develop a SCP based fully distributed scheme, we introduce more notation. Given any $\widehat{\mathbf{u}} = (\widehat{\mathbf{u}}_i)_{i=1}^n \in \mathcal{X}$ and any vectors d_{J_i}, $d_{g_{i,s}}$, and $d_{r_{i,s}}$ for $i = 1, \ldots, n$ and

$s = 1, \dots, 3p$, consider the following function as a convex approximation of the original nonconvex objective function J, where $y = (y_1, \dots, y_n) \in \mathbb{R}^N$ with each y_i being a suitable subvector of y:

$$f(y;\, \widehat{\mathbf{u}}, \{d_{J_i}\}_{i=1}^n) := \sum_{i=1}^{n} \Big(J_i(\widehat{\mathbf{u}}_i) + d_{J_i}^T(\widehat{\mathbf{u}}_i)(y_i - \widehat{\mathbf{u}}_i) + \frac{L_{J_i}}{2}\|y_i - \widehat{\mathbf{u}}_i\|_2^2 \Big),$$

and the following sets as convex approximations of the original nonconvex constraint sets $\mathcal{Y} \cap \mathcal{Z}$:

$$\begin{aligned} & \mathcal{C}\big(\widehat{\mathbf{u}}, \{d_{g_{i,s}}, d_{r_{i,s}},\, i = 1, \dots, n,\, s = 1, \dots, 3p\}\big) \\ := & \Big\{ y \in \mathcal{X} \mid g_{i,s}(\widehat{\mathbf{u}}_i) + d_{g_{i,s}}^T(y_i - \widehat{\mathbf{u}}_i) + \frac{L_{g_{i,s}}}{2}\|y_i - \widehat{\mathbf{u}}_i\|_2^2 \\ & \qquad - \big[r_{i,s}(\widehat{\mathbf{u}}_i) + d_{r_{i,s}}^T(y_i - \widehat{\mathbf{u}}_i)\big] \le 0, \quad i = 1, \dots, n, \quad s = 1, \dots, 3p \Big\}, \end{aligned}$$

Clearly, f is a strongly convex quadratic function in y and decoupled in y_i's, and the convex set $\mathcal{C}\big(\widehat{\mathbf{u}}, \{d_{g_{i,s}}, d_{r_{i,s}},\, i = 1, \dots, n,\, s = 1, \dots, p\}\big)$ is the Cartesian product of $\mathcal{C}_i$'s for $i = 1, \dots, n$, where each

$$\begin{aligned} \mathcal{C}_i\big(\widehat{\mathbf{u}}_i, \{d_{g_{i,s}}\}_{s=1}^{3p}, \{d_{r_{i,s}}\}_{s=1}^{3p}\big) := & \Big\{ y_i \in \mathcal{X}_i \mid g_{i,s}(\widehat{\mathbf{u}}_i) + d_{g_{i,s}}^T(y_i - \widehat{\mathbf{u}}_i) + \frac{L_{g_{i,s}}}{2}\|y_i - \widehat{\mathbf{u}}_i\|_2^2 \\ & - \big[r_{i,s}(\widehat{\mathbf{u}}_i) + d_{r_{i,s}}^T(y_i - \widehat{\mathbf{u}}_i)\big] \le 0,\;\; s = 1, \dots, 3p \Big\}. \end{aligned}$$

Using the above notation, the iterative scheme of the SCP method is: for a feasible initial guess $\widehat{\mathbf{u}}^0$,

$$\begin{aligned} \widehat{\mathbf{u}}^{k+1} = & \arg\min_y \Big\{ f(y; \widehat{\mathbf{u}}^k, \{\nabla J_i(\widehat{\mathbf{u}}_i^k)\}_{i=1}^n) \,\big|\, y \in \mathcal{A}, \quad \text{and} \\ & \qquad y \in \mathcal{C}\big(\widehat{\mathbf{u}}^k, \{\nabla g_{i,s}(\widehat{\mathbf{u}}_i^k), \nabla r_{i,s}(\widehat{\mathbf{u}}_i^k),\, i = 1, \dots, n,\, s = 1, \dots, 3p\}\big) \Big\}. \end{aligned} \tag{4.13}$$

By virtue of Corollary 4.5.1, the initial $\widehat{\mathbf{u}}^0$ can be chosen as a solution to the problem (4.7) or (4.11) whose objective function is J with $\boldsymbol{\varphi} = 0$ and the approximate constraints are polyhedral or quadratically constrained convex sets; see Section 4.5.3 for details. An efficient fully distributed scheme has been developed in [64] to compute such $\widehat{\mathbf{u}}^0$. It is shown in [45, Theorem 4.3] that if $\widehat{\mathbf{u}}^0$ is feasible, then $\widehat{\mathbf{u}}^k$ is feasible for all k and the constraint set in each step k is a nonempty closed convex set [45, Lemma 3.3].

The convex minimization problem (4.13) at each step k can be solved via operator splitting method based fully distributed schemes. Fix $\widehat{\mathbf{u}}^k = (\widehat{\mathbf{u}}_i^k)_{i=1}^n$ and the related gradients evaluated at $\widehat{\mathbf{u}}^k$. We write the objective function $f(y; \widehat{\mathbf{u}}, \{d_{J_i}\}_{i=1}^n)$ as $f(y)$ and the constraint sets $\mathcal{C}_i\big(\widehat{\mathbf{u}}_i^k, \{\nabla g_{i,s}(\widehat{\mathbf{u}}_i^k), \nabla r_{i,s}(\widehat{\mathbf{u}}_i^k),\, s = 1, \ldots, 3p\}\big)$ as $\mathcal{C}_i$'s for notational simplicity. Clearly, $\widehat{\mathbf{u}}_i^k \in \mathcal{C}_i$ for each i. If $\mathcal{C}_i$ is singleton for some i, i.e., $\mathcal{C}_i = \{\widehat{\mathbf{u}}_i^k\}$, then we have $\widehat{\mathbf{u}}_i^{k+1} = \widehat{\mathbf{u}}_i^k$ such that the optimization problem can be reduced to a simpler problem. When $\mathcal{C}_i$ is non-singleton, it follows from Lemma 4.5.2 that the Slater's condition holds for that $\mathcal{C}_i$. Let $F(y) := f(y; \widehat{\mathbf{u}}^k, \{\nabla J_i(\widehat{\mathbf{u}}_i^k)\}_{i=1}^n) + \boldsymbol{\delta}\mathcal{C}(y) + \boldsymbol{\delta}\mathcal{A}(y)$. By [60, Corollary 23.8.1], $\partial F(y) = \{\nabla f(y)\} + \mathcal{N}_{\mathcal{C}}(y) + \mathcal{N}_{\mathcal{A}}(y)$. As a result, several operator splitting method based fully distributed algorithms [15], [26] can be applied to solve the convex optimization problem (4.13).

Motivated by [64], we consider the (generalized) Douglas-Rachford splitting method based distributed scheme. Specifically, define for each $i = 1, \ldots, n$, $f_i(y_i) := J_i(\widehat{\mathbf{u}}_i^k) + d_{J_i}^T(\widehat{\mathbf{u}}_i^k)(y_i - \widehat{\mathbf{u}}_i) + \frac{L_{J_i}}{2}\|y_i - \widehat{\mathbf{u}}_i^k\|_2^2$, and $\widehat{f}_i(y) := f_i(y_i) + \boldsymbol{\delta}\mathcal{C}_i(y_i)$. Hence, the objective function $f(y) = \sum_{i=1}^n f_i(y_i)$. For any constant $0 < \alpha < 1$ and $\rho > 0$, the Douglas-Rachford splitting method based scheme is given by

$$w^{t+1} = \Pi_{\mathcal{A}}(z^t), \quad z^{t+1} = z^t + 2\alpha \cdot \Big[\mathrm{Prox}_{\rho\widehat{f}_1+\cdots+\rho\widehat{f}_n}\big(2w^{t+1} - z^t\big) - w^{t+1}\Big], \quad \forall\, t \in \mathbb{Z}_+, \tag{4.14}$$

where Prox_h denotes the proximal operator of a proper lower semicontinuous convex function h, and $\Pi_{\mathcal{A}}$ denotes the Euclidean projection onto $\mathcal{A}$. Since $\mathcal{A}$ is the consensus subspace, it is shown that [26, Section IV] that for any $\widehat{\mathbf{u}} := (\widehat{\mathbf{u}}_1, \ldots, \widehat{\mathbf{u}}_n)$ where $\widehat{\mathbf{u}}_i := \big(\mathbf{u}_i, (\mathbf{u}_{ij})_{j\in\mathcal{N}_i}\big)$, $\overline{\mathbf{u}} := \Pi_{\mathcal{A}}(\widehat{\mathbf{u}})$ is given by:

$$\overline{\mathbf{u}}_j = \overline{\mathbf{u}}_{ij} = \frac{1}{1+|\mathcal{N}_j|}\Big(\widehat{\mathbf{u}}_j + \sum_{k\in\mathcal{N}_j} \widehat{\mathbf{u}}_{kj}\Big), \qquad \forall\,(i,j) \in \mathcal{E}. \tag{4.15}$$

Furthermore, since $\widehat{f}_i$'s are decoupled, a distributed version of the above algorithm is given by:

$$w_i^{t+1} = \overline{z}_i^t, \qquad i = 1, \ldots, n; \tag{4.16a}$$

$$z_i^{t+1} = z_i^t + 2\alpha \cdot \Big[\mathrm{Prox}_{\rho\widehat{f}_i}\big(2w_i^{t+1} - z_i^t\big) - w_i^{t+1}\Big], \quad i = 1, \ldots, n. \tag{4.16b}$$

Note that the proximal operator in the second equation of (4.14) is given by $\mathrm{Prox}_{\rho\widehat{f}_i}(2w_i^{t+1}-z_i^t) = \arg\min_{y_i\in\mathcal{C}_i} f_i(y_i)+\frac{1}{2\rho}\|y_i-(2w_i^{t+1}-z_i^t)\|_2^2$, where $\mathcal{C}_i$ is the intersection of the polyhedral set $\mathcal{X}_i$ and a quadratically constrained convex set. Since f_i is a convex quadratic function, $\mathrm{Prox}_{\rho\widehat{f}_i}(2w_i^{t+1} - z_i^t)$ can be formulated as a second-order cone program or QCQP and solved by SeDuMi [75]. See Algorithm 10 for its pseudo-code.

Since $\mathcal{X}$ is a compact set, the numerical sequence $(\widehat{\mathbf{u}}^k)$ generated by Algorithm 10 always has an accumulation point denoted by $\widehat{\mathbf{u}}^*$. It follows from [45, Theorem 3.4] that

Algorithm 10 Sequential Convex Programming and Douglas-Rachford Splitting Method based Fully Distributed Algorithm for $p \geq 2$

Choose constants $0 < \alpha < 1$ and $\rho > 0$
Solve the problem (4.11) with $\boldsymbol{\varphi} = 0$ via a fully distributed scheme and obtain a solution $\widehat{\mathbf{u}}^{\text{lin}}$
Initialize $k = 0$, and set an initial point $\widehat{\mathbf{u}}^0 = \widehat{\mathbf{u}}^{\text{lin}}$
while the stopping criteria is not met **do**
    Compute $\nabla J_i(\widehat{\mathbf{u}}_i^k)$, $\nabla g_{i,s}(\widehat{\mathbf{u}}_i^k)$, $\nabla r_{i,s}(\widehat{\mathbf{u}}_i^k)$, and set $z^0 = \widehat{\mathbf{u}}^k$ and $t = 0$.
    repeat
        for $i = 1, \ldots, n$ **do**
            Compute $\overline{z}_i^t$ using equation (4.15), and let $w_i^{t+1} \leftarrow \overline{z}_i^t$
        end for
        for $i = 1, \ldots, n$ **do**
            $z_i^{t+1} \leftarrow z_i^t + 2\alpha \cdot \left[\text{Prox}_{\rho \widehat{f}_i}\left(2w_i^{t+1} - z_i^t\right) - w_i^{t+1}\right]$
        end for
        $t \leftarrow t + 1$
    until an accumulation point is achieved
    Set $\widehat{\mathbf{u}}^{k+1} = w^t$ and $k \leftarrow k + 1$
end while
return $\widehat{\mathbf{u}}^* = \widehat{\mathbf{u}}^k$

under very mild conditions, $\widehat{\mathbf{u}}^*$ is feasible and is a KKT point of the nonconvex program (4.7). Our numerical experiences show that $(\widehat{\mathbf{u}}^k)$ converges to $\widehat{\mathbf{u}}^*$ which is a local minimizer of (4.7). This coincides with the observation made in Corollary 4.5.1 when $c_{2,i}$ and $c_{3,i}$ are small.

4.5.3 Approximation of the Objective Function and Constraint Functions

When $p > 1$, the underlying MPC optimization problem (4.7) and its locally coupled formulation (4.11) give rise to non-convex optimization problems with complicated objective functions and constraints, especially the velocity and safety distance constraints for a relatively large p, due to highly sophisticated closed-form expressions for $\mathbf{a}_i(\mathbf{u}_i)$'s. To facilitate computation, particularly for real-time computation, we derive a simplified model to approximate the objective function and constraint functions below. We start with the constraints for $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ first, where we omit k since it is fixed.

The exact closed-form expressions for $\mathbf{a}_i(\mathbf{u}_i)$'s are given in the recursive manner at the beginning of Section 4.4.1. These expressions are highly sophisticated especially for large j's because of the nonlinear relation in aerodynamic drag. Since the coefficients $c_{2,i}$'s and $c_{3,i}$'s are small, we only consider the terms in $\mathbf{a}_i(\mathbf{u}_i)$ that are linearly in $c_{2,i}$'s and $c_{3,i}$'s while ignoring the terms involving higher orders of $c_{2,i}$'s and $c_{3,i}$'s. Such an approximation is accurate enough for transportation applications and facilitates numerical computation. Toward this goal, recall that the matrix $S_p \in \mathbb{R}^{p\times p}$ is defined in the same was as in (3.3) with n replaced by p, and $(S_p\mathbf{u}_i)_0 := 0$. We then have, for each $i = 1, \ldots, n$ and $s = 1, \ldots, p$,

$$\big(\mathbf{a}_i(\mathbf{u}_i)\big)_s \approx \big(\mathbf{u}_i\big)_s - c_{2,i}\Big[v_i(k) + \tau\big(S_p\mathbf{u}_i\big)_{s-1}\Big]^2 - c_{3,i}g,$$

where $\big(\mathbf{a}_i(\mathbf{u}_i)\big)_s$ denotes the s-entry of $\mathbf{a}_i(\mathbf{u}_i)$ that corresponds to $a_i(k+s-1, u_i(k), \ldots, u_i(k+s-1))$, and $\big(\mathbf{u}_i\big)_s$ denotes the s-entry of $\mathbf{u}_i$ that corresponds to $u_i(k+s-1)$. Therefore,

$$\mathbf{a}_i(\mathbf{u}_i) \approx \mathbf{u}_i - c_{2,i}\Big[v_i^2(k)\mathbf{1} + 2\tau v_i(k)\widetilde{S}_p\mathbf{u}_i + \tau^2\big(\widetilde{S}_p\mathbf{u}_i\big)\circ\big(\widetilde{S}_p\mathbf{u}_i\big)\Big] - c_{3,i}g\mathbf{1}, \tag{4.17}$$

where $\widetilde{S}_p := \begin{bmatrix} 0 & 0 \\ I_{p-1} & 0 \end{bmatrix} S_p \in \mathbb{R}^{p\times p}$, and $\circ$ denotes the Hadamard product of two vectors in $\mathbb{R}^p$. Slightly abusing notation, we let $\mathbf{a}_i(\mathbf{u}_i)$ represent its approximation given on the right-hand side of (4.17). It is easy to derive the Jacobian of $\mathbf{a}_i(\mathbf{u}_i)$ as

$$\mathbf{J}\mathbf{a}_i(\mathbf{u}_i) = I_p - 2c_{2,i}\tau v_i(k)\widetilde{S}_p - 2c_{2,i}\tau^2\mathrm{diag}(\widetilde{S}_p\mathbf{u}_i)\widetilde{S}_p, \tag{4.18}$$

where for a vector $v = (v_1, \ldots, v_p) \in \mathbb{R}^p$, $\mathrm{diag}(v)$ denotes the $p\times p$ diagonal matrix whose diagonal entries are given by $v_1, \ldots, v_p$. Here we use the fact that the Jacobian of $(Ax)\circ(Ax)$ is given by $\mathbf{J}(Ax)\circ(Ax) = 2\mathrm{diag}(Ax)A$ for a matrix A.

Approximate speed constraint. Using the approximated $\mathbf{a}_i(\cdot)$, we have, for each $i = 1, \ldots, n$ and $j = 1, \ldots, p$,

$$v_i(k+j) = v_i(k) + \tau \textstyle\sum_{s=1}^{j} \big(\mathbf{a}_i(\mathbf{u}_i)\big)_s \approx v_i(k) + \tau \left(\big(S_p \mathbf{u}_i\big)_j - j \cdot c_{3,i} g - c_{2,i} \sum_{s=0}^{j-1} \Big[v_i(k) + \tau \big(S_p \mathbf{u}_i\big)_s \Big]^2 \right). \tag{4.19}$$

It should be noted that the resulting approximation of $v_i(k+j)$ remains a nonlinear and nonconvex function in $u_i(k), \ldots, u_i(k+j-1)$ or $\mathbf{u}_i$. The approximated speed constraint for $\mathbf{u}_i$ is given by

$$\mathcal{Y}_i := \Big\{ \mathbf{u}_i \in \mathbb{R}^p \,\Big|\, v_{\min} \leq v_i(k) + \tau \Big(\big(S_p \mathbf{u}_i\big)_j - j \cdot c_{3,i} g - c_{2,i} \sum_{s=0}^{j-1} \big[v_i(k) + \tau (S_p \mathbf{u}_i)_s \big]^2 \Big) \leq v_{\max}, \quad j = 1, \ldots, p \Big\}.$$

For $j = 1, \ldots, p$, define the function

$$q_{i,j}(\mathbf{u}_i) \;:=\; v_i(k) + \tau \Big(\big(S_p \mathbf{u}_i\big)_j - j \cdot c_{3,i} g - c_{2,i} \sum_{s=0}^{j-1} \big[v_i(k) + \tau (S_p \mathbf{u}_i)_s \big]^2 \Big). \tag{4.20}$$

A straightforward calculation shows that the gradient of $q_{i,j}$ is given by

$$\nabla q_{i,j}(\mathbf{u}_i) = \tau \left((S_p)_{j,\bullet} - 2\tau c_{2,i} \sum_{s=0}^{j-1} \big[v_i(k) + \tau (S_p \mathbf{u}_i)_s \big] (S_p)_{s\bullet} \right)^T.$$

Approximate safety distance constraint. For each $i = 1, \ldots, n$ and $j = 1, \ldots, p$, the safety distance constraint is given by

$$\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j := L_i + r_i \cdot v_i(k+j) - \frac{(v_i(k+j) - v_{\min})^2}{2a_{i,\min}} - [x_{i-1}(k+j) - x_i(k+j)] \;\leq\; 0.$$

To derive an approximation of $\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j$, recall that the expression for $z_i(k+j) = x_{i-1}(k+j) - x_i(k+j)$ is given by (4.5), where for each $s = 0, 1, \ldots, j-1$, it follows from (4.19) that

$$
\begin{aligned}
& b_i(k+s, \mathbf{u}_{i-1}, \mathbf{u}_i) = \big(\mathbf{a}_{i-1}(\mathbf{u}_{i-1}) - \mathbf{a}_i(\mathbf{u}_i)\big)_{s+1} \\
& = \big[u_{i-1}(k+s) - c_{2,i-1} v_{i-1}^2(k+s) - c_{3,i-1} g\big] - \big[u_i(k+s) - c_{2,i} v_i^2(k+s) - c_{3,i} g\big] \\
& \approx \big[u_{i-1}(k+s) - u_i(k+s)\big] - \left(c_{2,i-1}\Big[v_{i-1}(k) + \tau\big(S_p \mathbf{u}_{i-1}\big)_s\Big]^2 - c_{2,i}\Big[v_i(k) + \tau\big(S_p \mathbf{u}_i\big)_s\Big]^2\right) \\
& \qquad - \big(c_{3,i-1} - c_{3,i}\big) g.
\end{aligned}
$$

Further, by using the approximation of $v_i(k+j)$ given in (4.19), we obtain

$$
\begin{aligned}
\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j \approx \quad & L_i + r_i \cdot \left[v_i(k) + \tau\Big(\big(S_p \mathbf{u}_i\big)_j - j \cdot c_{3,i} g - c_{2,i} \sum_{s=0}^{j-1} \big[v_i(k) + \tau\big(S_p \mathbf{u}_i\big)_s\big]^2\Big)\right] \\
& - \frac{1}{2a_{i,\min}} \Bigg[v_i(k) - v_{\min} + \tau\Big(\big(S_p \mathbf{u}_i\big)_j - j \cdot c_{3,i} g \\
& - c_{2,i} \sum_{s=0}^{j-1} \big[v_i(k) + \tau\big(S_p \mathbf{u}_i\big)_s\big]^2\Big)\Bigg]^2 - \Bigg\{ z_i(k) + \Delta + j\tau z_i'(k) \\
& + \tau^2 \sum_{s=0}^{j-1} \frac{2(j-s)-1}{2} \Big[u_{i-1}(k+s) - u_i(k+s) - \Big(c_{2,i-1}\big[v_{i-1}(k) \\
& + \tau\big(S_p \mathbf{u}_{i-1}\big)_s\big]^2 - c_{2,i}\big[v_i(k) + \tau\big(S_p \mathbf{u}_i\big)_s\big]^2\Big) - \big(c_{3,i-1} - c_{3,i}\big) g\Big] \Bigg\}.
\end{aligned}
\tag{4.21}
$$

By slightly abusing the notation, we let $\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j$ denote its approximation given above. Clearly, this approximating function is smooth but nonconvex. Further, when $\boldsymbol{\varphi} = (\boldsymbol{\varphi}_d, \boldsymbol{\varphi}_f) = (c_{2,i}, c_{3,i})_{i=1}^n = 0$, each $\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j$ reduces to a convex quadratic function in $(\mathbf{u}_{i-1}, \mathbf{u}_i)$. To compute the gradient of $\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j$, let the vector $\mathbf{p}$ and the matrix R_p be given by

$$\mathbf{p} := \begin{bmatrix} 1 \\ 2 \\ \vdots \\ p \end{bmatrix} \in \mathbb{R}^p, \qquad R_p := \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 3 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 2p-3 & 2p-5 & \dots & 1 & 0 \\ 2p-1 & 2p-3 & \dots & 3 & 1 \end{bmatrix} \in \mathbb{R}^{p\times p}.$$

It is noted that for each $i = 1, \dots, n$ and $j = 1, \dots, p$,

$$x_{i-1}(k+j) - x_i(k+j) = z_i(k+j) = z_i(k) + \Delta + \frac{\tau^2}{2}\Big[R_p\Big(\mathbf{a}_{i-1}(\mathbf{u}_{i-1}) - \mathbf{a}_i(\mathbf{u}_i)\Big)\Big]_j$$

Hence, using the gradient of $q_{i,j}$ and the Jacobian of $\mathbf{a}_i$ given by (4.18), the gradients of $\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j$ with respect to $\mathbf{u}_i$ and $\mathbf{u}_{i-1}$ (for $i \geq 2$) are respectively given by

$$\begin{aligned} &\nabla_{\mathbf{u}_i}\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j = r_i \cdot \nabla q_{i,j}(\mathbf{u}_i) \\ &- \frac{1}{a_{i,\min}}\left[v_i(k) - v_{\min} + \tau\Big(\big(S_p\mathbf{u}_i\big)_j - j\cdot c_{3,i}g - c_{2,i}\sum_{s=0}^{j-1}\big[v_i(k) + \tau\big(S_p\mathbf{u}_i\big)_s\big]^2\Big)\right]\cdot\nabla q_{i,j}(\mathbf{u}_i) \\ &+ \frac{\tau^2}{2}\Big[(R_p)_{j\bullet}\,\mathbf{J}\mathbf{a}_i(\mathbf{u}_i)\Big]^T, \end{aligned}$$

and for $i \geq 2$,

$$\nabla_{\mathbf{u}_{i-1}}\big(H_i(\mathbf{u}_{i-1}, \mathbf{u}_i)\big)_j = -\frac{\tau^2}{2}\Big[(R_p)_{j\bullet}\,\mathbf{J}\mathbf{a}_{i-1}(\mathbf{u}_{i-1})\Big]^T.$$

Approximate objective function. Consider the decomposition of the (central) objective function given by local objective functions J_i's in (4.8). The approximate objective function J_i can be easily obtained by substituting (4.17) into (4.8). In what follows, we compute the gradient of J_i. It follows from $J_1(\mathbf{u}_1, \mathbf{u}_2) := \frac{1}{2} \begin{bmatrix} \mathbf{a}_1^T(\mathbf{u}_1) & \mathbf{a}_2^T(\mathbf{u}_2) \end{bmatrix} \big(\widehat{W}^1 - \tau^2 \widehat{\Psi}^1\big) \begin{bmatrix} \mathbf{a}_1(\mathbf{u}_1) \\ \mathbf{a}_2(\mathbf{u}_2) \end{bmatrix} + c_{\mathcal{I}_1}^T \mathbf{a}_1(\mathbf{u}_1) + \frac{\tau^2}{2} \begin{bmatrix} \mathbf{u}_1^T & \mathbf{u}_2^T \end{bmatrix} \widehat{\Psi}^1 \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}$ that

$$\nabla_{\mathbf{u}_1} J_1(\mathbf{u}_1, \mathbf{u}_2) = \begin{bmatrix} \big(\mathbf{J}\mathbf{a}_1(\mathbf{u}_1)\big)^T & 0 \end{bmatrix} \big(\widehat{W}^1 - \tau^2 \widehat{\Psi}^1\big) \begin{bmatrix} \mathbf{a}_1(\mathbf{u}_1) \\ \mathbf{a}_2(\mathbf{u}_2) \end{bmatrix} + \big(\mathbf{J}\mathbf{a}_1(\mathbf{u}_1)\big)^T c_{\mathcal{I}_1} + \tau^2 \begin{bmatrix} I_p & 0 \end{bmatrix} \widehat{\Psi}^1 \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix},$$

$$\nabla_{\mathbf{u}_2} J_1(\mathbf{u}_1, \mathbf{u}_2) = \begin{bmatrix} 0 & \big(\mathbf{J}\mathbf{a}_2(\mathbf{u}_2)\big)^T \end{bmatrix} \big(\widehat{W}^1 - \tau^2 \widehat{\Psi}^1\big) \begin{bmatrix} \mathbf{a}_1(\mathbf{u}_1) \\ \mathbf{a}_2(\mathbf{u}_2) \end{bmatrix} + \tau^2 \begin{bmatrix} 0 & I_p \end{bmatrix} \widehat{\Psi}^1 \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}.$$

The similar result holds for $\nabla J_n(\mathbf{u}_{n-1}, \mathbf{u}_n)$ and $\nabla J_i(\mathbf{u}_{i-1}, \mathbf{u}_i, \mathbf{u}_{i+1})$ for $i = 2, \ldots, n-1$.

Under the assumption **A.2**, the convex QCQP is feasible for all small $\|\boldsymbol{\varphi}\| > 0$. This implies that the nonconvex program (4.11) with approximate constraints is feasible for all small $\|\boldsymbol{\varphi}\|$.

4.6 Control Design and Stability Analysis of Closed Loop Dynamics

In this section, we discuss the choice of 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. It should be noted that all the proofs in this section are omitted and can be found in the online version of our own manuscript [63, Section 6]. We present the statements in this thesis for completeness.

For the similar reasons given in [22, Section 5], we focus on the constraint free case. Recall that $\boldsymbol{\varphi} := (\boldsymbol{\varphi}_d, \boldsymbol{\varphi}_f) \in \mathbb{R}^{2n}_+$, where $\boldsymbol{\varphi}_d := (c_{2,1}, \ldots, c_{2,n}) \in \mathbb{R}^n_+$ and $\boldsymbol{\varphi}_f := (c_{3,1}, \ldots, c_{3,n}) \in \mathbb{R}^n_+$. Further, $c_{2,0} = c_{3,0} = 0$ as indicated before.

When $\boldsymbol{\varphi} = 0$, the nonlinear vehicle dynamics reduces to the linear vehicle dynamics given by (3.1), for which the closed loop stability of the MPC based platooning control with a general horizon p has been analyzed in [Section 3.5]. Throughout the rest of this section, we assume that for a given p, the weight matrices $Q_{z,s}, Q_{z',s}$ and $Q_{w,s}$ satisfying the assumption **A.3** are chosen such that A_{C} is Schur stable.

4.6.1 Reformulation of the Closed Loop Dynamics as a Tracking System

Consider the nonlinear vehicle dynamics (4.1). It follows from the definitions of $z(k), z'(k)$ and $w(k)$ that for $i = 1, \ldots, n$,

$$\begin{aligned} z_i(k+1) &= z_i(k) + \tau z'_i(k) \\ &\quad + \frac{\tau^2}{2}\Big(w_i(k) - [c_{2,i-1}v^2_{i-1}(k) - c_{2,i}v^2_i(k)] - [c_{3,i-1} - c_{3,i}]g\Big), \quad (4.22a) \\ z'_i(k+1) &= z'_i(k) + \tau\Big(w_i(k) - [c_{2,i-1}v^2_{i-1}(k) - c_{2,i}v^2_i(k)] - [c_{3,i-1} - c_{3,i}]g\Big). \quad (4.22b) \end{aligned}$$

For given $(v_0(k), u_0(k)), k \in \mathbb{Z}_+$, the equilibrium of the above discrete-time system is $(z_e, z'_e) = (0, 0)$ such that $v_{e,i}(k) = v_0(k)$ for all $i = 1, \ldots, n$. Hence, the corresponding

$$w_{e,i}(k) = [c_{2,i-1} - c_{2,i}]v^2_0(k) + [c_{3,i-1} - c_{3,i}]g, \qquad \forall\ i = 1, \ldots, n.$$

Let $w_e(k) := (w_{e,1}(k), \dots, w_{e,n}(k))^T$. By shifting $w(k)$ from the time-varying $w_e(k)$, we define $\widehat{w}(k) := w(k) - w_e(k)$. Hence, this yields the following equations: for $i = 1, \dots, n$,

$$z_i(k+1) \;=\; z_i(k) + \tau z_i'(k) + \frac{\tau^2}{2}\Big(\widehat{w}_i(k) + r_i(k)\Big), \quad z_i'(k+1) \;=\; z_i'(k) + \tau\Big(\widehat{w}_i(k) + r_i(k)\Big),$$

where for each $i = 1, \dots, n$,

$$\begin{aligned} r_i(k) &:= w_{e,i}(k) - [c_{2,i-1} v_{i-1}^2(k) - c_{2,i} v_i^2(k)] - [c_{3,i-1} - c_{3,i}] g \\ &= c_{2,i-1}\big(v_0^2(k) - v_{i-1}^2(k)\big) - c_{2,i}\big(v_0^2(k) - v_i^2(k)\big). \end{aligned}$$

In light of $v(k) = -S_n z'(k) + v_0(k) \cdot \mathbf{1}$, we have $v_0^2(k) - v_i^2(k) = (S_n z'(k))_i \cdot [2v_0(k) - (S_n z'(k))_i]$ for each i. We thus define the vector-valued smooth function $h : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$ as $h(z', v_0) := (h_1(z', v_0), \dots, h_n(z', v_0))^T$, where $h_1(z', v_0) := c_{2,1}(S_n z')_1\big[(S_n z')_1 - 2v_0\big]$, and for $i = 2, \dots, n$,

$$h_i(z', v_0) := c_{2,i-1}(S_n z')_{i-1}\big[2v_0 - (S_n z')_{i-1}\big] - c_{2,i}(S_n z')_i\big[2v_0 - (S_n z')_i\big].$$

Clearly, $h(0, v_0) = 0$ for any v_0. Further, we decompose h as the sum of the following two functions:

$$h(z', v_0) \;=\; v_0 \cdot \underbrace{\left(2 \begin{bmatrix} -c_{2,1} & & & \\ & c_{2,1} - c_{2,2} & & \\ & & \ddots & \\ & & & c_{2,n-1} - c_{2,n} \end{bmatrix} S_n \right)}_{:= D(\boldsymbol{\varphi}_d)} \cdot z' \;+\; \widetilde{h}(z'), \qquad (4.23)$$

where the vector-valued function $\widetilde{h} := (\widetilde{h}_1, \ldots, \widetilde{h}_n)^T : \mathbb{R}^n \to \mathbb{R}^n$ is given by:

$$\widetilde{h}_i(z') := c_{2,i}[(S_n z')_i]^2 - c_{2,i-1}[(S_n z')_{i-1}]^2, \qquad \forall\, i = 1, \ldots, n.$$

or equivalently

$$\widetilde{h}(z') := \begin{bmatrix} c_{2,1} & & & \\ -c_{2,1} & c_{2,2} & & \\ & \ddots & \ddots & \\ & & -c_{2,n-1} & c_{2,n} \end{bmatrix} \Big[\big(S_n z'\big) \circ \big(S_n z'\big) \Big] = \underbrace{S_n^{-1} \mathrm{diag}(\boldsymbol{\varphi}_d)}_{:=\widetilde{D}(\boldsymbol{\varphi}_d)} \Big[\big(S_n z'\big) \circ \big(S_n z'\big) \Big]. \tag{4.24}$$

Note that the elements of D and $\widetilde{D}$ are linear in $\boldsymbol{\varphi}_d$ such that $D(\boldsymbol{\varphi}_d) = \widetilde{D}(\boldsymbol{\varphi}_d) = 0$ when $\boldsymbol{\varphi}_d = 0$. Using this notation, the nonlinear vehicle dynamics (4.1) is described by the following discrete-time system:

$$\begin{aligned} \begin{bmatrix} z(k+1) \\ z'(k+1) \end{bmatrix} &= \begin{bmatrix} I_n & \tau I_n \\ 0 & I_n \end{bmatrix} \begin{bmatrix} z(k) \\ z'(k) \end{bmatrix} + \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n \end{bmatrix} \Big(\widehat{w}(k) + h(z'(k), v_0(k)) \Big) \\ &= \left\{ \begin{bmatrix} I_n & \tau I_n \\ 0 & I_n \end{bmatrix} + v_0(k) \cdot \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n \end{bmatrix} \begin{bmatrix} 0 & D(\boldsymbol{\varphi}_d) \end{bmatrix} \right\} \begin{bmatrix} z(k) \\ z'(k) \end{bmatrix} \\ &\qquad + \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n \end{bmatrix} \Big(\widehat{w}(k) + \widetilde{h}(z'(k)) \Big). \end{aligned}$$

By slightly abusing notation, we also write the function $\widetilde{h}$ as $\widetilde{h}_{\boldsymbol{\varphi}_d}(z')$ to emphasize its dependence on $\boldsymbol{\varphi}_d$. Noting that $\widetilde{h}$ is linear in $\boldsymbol{\varphi}_d$ for any fixed z', we see that $\widetilde{h}_0(z') \equiv 0$ for any given $z' \in \mathbb{R}^n$.

Define the following matrices:

$$A := \begin{bmatrix} I_n & \tau I_n \\ 0 & I_n \end{bmatrix}, \quad B := \begin{bmatrix} \frac{\tau^2}{2} I_n \\ \tau I_n \end{bmatrix}, \quad \Delta A(\boldsymbol{\varphi}_d) := B \begin{bmatrix} 0 & D(\boldsymbol{\varphi}_d) \end{bmatrix}, \quad \widehat{A}(k) := A + v_0(k) \cdot \Delta A(\boldsymbol{\varphi}_d). \tag{4.25}$$

As before, we often write $\widehat{A}(k)$ as $\widehat{A}(v_0(k), \boldsymbol{\varphi}_d)$ to stress its dependence on $v_0(k)$ and $\boldsymbol{\varphi}_d$. Let $\mathbf{z} := (z, z') \in \mathbb{R}^n \times \mathbb{R}^n$. We obtain the following non-autonomous nonlinear dynamical system:

$$\mathbf{z}(k+1) \;=\; \widehat{A}(k)\mathbf{z}(k) + B\Big(\widehat{w}_*(k) + \widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k))\Big), \qquad \forall\, k \in \mathbb{Z}_+, \tag{4.26}$$

where $\widehat{w}_*(k)$ is an optimal solution to the unconstrained MPC optimization problem (4.7) which implicitly depends on $\mathbf{z}(k), v_0(k)$ and $u_0(k)$. For any fixed $\boldsymbol{\varphi}_d$, the closed loop system given by (4.26) yields a non-autonomous nonlinear dynamical system, since $\widetilde{h}$ is nonlinear in z' and $v_0(k)$ is time varying. In what follows, we further discuss the non-autonomous system (4.26) for different MPC horizon p.

Case (i): $p = 1$. In this case, the closed-form expression of $\widehat{w}_*(k)$ is derived below. Letting $\widetilde{w}(k) := w(k) - u_0(k) \cdot \mathbf{e}_1 = \widehat{w}(k) + d(k)$, where $d(k) := w_e(k) - u_0(k) \cdot \mathbf{e}_1$, the unconstrained MPC becomes

$$\begin{aligned} \min J(\widehat{w}(k)) := \frac{1}{2}\Big\{ & z^T(k+1) Q_z z(k+1) + (z'(k+1))^T Q_{z'} z'(k+1) \\ & + \tau^2 [\widehat{w}(k) + d(k)]^T Q_w [\widehat{w}(k) + d(k)]^T \Big\} \end{aligned}$$

subject to $z(k+1) = z(k) + \tau z'(k) + \frac{\tau^2}{2}\big[\widehat{w}(k) + h(z'(k), v_0(k))\big]$, and $z'(k+1) = z'(k) + \tau\big[\widehat{w}(k) + h(z'(k), v_0(k))\big]$, where $Q_z := Q_{z,1}$, $Q_{z'} := Q_{z',1}$, and $Q_w := Q_{w,1}$. Hence,

$$\nabla J(\widehat{w}(k)) \;=\; \Big(\frac{\tau^4}{4} Q_z + \tau^2 Q_{z'} + \tau^2 Q_w\Big)\widehat{w}(k) + \frac{\tau^2}{2} Q_z z(k) + \Big(\frac{\tau^3}{2} Q_z + \tau Q_{z'}\Big) z'(k)$$

$$+\Big(\frac{\tau^4}{4}Q_z + \tau^2 Q_{z'}\Big)h(z'(k), v_0(k)) + \tau^2 Q_w d(k).$$

Define the matrix

$$\widehat{W} := \left[\frac{\tau^2 Q_z}{4} + Q_{z'} + Q_w\right]^{-1}. \tag{4.27}$$

Using this matrix, we obtain the closed form expression for the optimal solution $\widehat{w}_*(k)$ as

$$\widehat{w}_*(k) = -\widehat{W}\cdot\Big[\frac{Q_z}{2}z(k) + \Big(\frac{\tau Q_z}{2} + \frac{Q_{z'}}{\tau}\Big)z'(k) + \Big(\frac{\tau^2}{4}Q_z + Q_{z'}\Big)h(z'(k), v_0(k)) + Q_w d(k)\Big].$$

Substituting $\widehat{w}_*(k)$ into (4.26), using the following matrix A_{C} derived in [22, Section 5] (which agrees with the closed loop dynamics matrix in (3.21) when $p = 1$)

$$A_{\mathrm{C}} := \begin{bmatrix} I_n - \frac{\tau^2}{4}\widehat{W}Q_z & \tau I_n - \widehat{W}\Big(\frac{\tau^3}{4}Q_z + \frac{\tau}{2}Q_{z'}\Big) \\ -\frac{\tau}{2}\widehat{W}Q_z & I_n - \widehat{W}\Big(\frac{\tau^2}{2}Q_z + Q_{z'}\Big) \end{bmatrix} \tag{4.28}$$

and in view of $h(z', v_0) = v_0 D(\boldsymbol{\varphi}_d)z' + \widetilde{h}_{\boldsymbol{\varphi}_d}(z')$, the closed loop dynamics is characterized by:

$$\begin{aligned} \mathbf{z}(k+1) &= A_{\mathrm{C}}\,\mathbf{z}(k) + B\Big\{-\widehat{W}\Big(\frac{\tau^2}{4}Q_z + Q_{z'}\Big)h(z'(k), v_0(k)) \\ &\qquad -\widehat{W}Q_w d(k) + h(z'(k), v_0(k))\Big\} \\ &= \Big(A_{\mathrm{C}} + v_0(k)\cdot\Delta\bar{A}(\boldsymbol{\varphi}_d)\Big)\mathbf{z}(k) - B\widehat{W}Q_w d(k) + \breve{B}\widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k)), \end{aligned}$$

where the matrices $\Delta\bar{A}$ and $\breve{B}$ are given by

$$\breve{B} := B\Big[I_n - \widehat{W}\big(\frac{\tau^2}{4}Q_z + Q_{z'}\big)\Big], \qquad \Delta\bar{A}(\boldsymbol{\varphi}_d) := \breve{B}\begin{bmatrix}0 & D(\boldsymbol{\varphi}_d)\end{bmatrix}. \tag{4.29}$$

By using $d(k) = w_e(k) - u_0(k) \cdot \mathbf{e}_1$, the closed loop dynamics for $p = 1$ becomes:

$$\mathbf{z}(k+1) \;=\; \Big(A_{\rm C} + v_0(k) \cdot \Delta\bar{A}(\boldsymbol{\varphi}_d)\Big)\mathbf{z}(k) + B\Big(u_0(k)\mathbf{d} - \widehat{W} Q_w w_e(k)\Big) + \breve{B}\widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k)), \quad (4.30)$$

where $\mathbf{d} = \widehat{W} Q_w \mathbf{e}_1$ that agrees with what is given in (3.23) for $p = 1$, and $w_e(k)$ depends on $v_0(k)$ and $\boldsymbol{\varphi}$. Further, there exists a positive constant $\widetilde{\varkappa}$ such that $\|w_e(k)\| \le \widetilde{\varkappa} \cdot \|\boldsymbol{\varphi}\|$ for any $v_0(k) \in [v_{\min}, v_{\max}]$.

Case (ii): $p > 1$. In this case, recall that for any fixed $k \in \mathbb{Z}_+$, $u_0(k+s) = u_0(k)$ for all $s = 1, \ldots, p-1$ in the MPC model. Hence, $v_0(k+s) = v_0(k) + \tau s u_0(k)$ for all $s = 1, \ldots, p-1$. Define $\widehat{A}(k+s) := A + v_0(k+s) \cdot \Delta A(\boldsymbol{\varphi}_d)$ for all $s = 0, 1, \ldots, p-1$. Given $\widehat{A}(k+s)$ with $s = 0, \ldots, p-1$, define the state transition matrix as the following matrix product for any $s, s' \in \{0, \ldots, p\}$ with $s \le s'$,

$$\Phi_{\widehat{A}}(k+s, k+s) \;:=\; I; \qquad \Phi_{\widehat{A}}(k+s', k+s) \;:=\; \widehat{A}(k+s'-1) \times \cdots \times \widehat{A}(k+s), \quad \forall\, s' > s.$$

Based upon the above notation, we obtain, for any fixed $k \in \mathbb{Z}_+$ and $s = 1, \ldots, p$,

$$\begin{aligned}
\mathbf{z}(k+s) \;&=\; \Phi_{\widehat{A}}(k+s, k)\mathbf{z}(k) \\
&\qquad + \sum_{i=0}^{s-1} \Phi_{\widehat{A}}(k+s, k+i+1) B\Big[\widehat{w}(k+i) + \widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k+i))\Big] \quad (4.31) \\
&=\; \Phi_{\widehat{A}}(k+s, k)\mathbf{z}(k) + \sum_{i=0}^{s-1} \Phi_{\widehat{A}}(k+s, k+i+1) B\widehat{w}(k+i) \quad (4.32) \\
&\qquad + \sum_{i=0}^{s-1} \Phi_{\widehat{A}}(k+s, k+i+1) B\widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k+i)).
\end{aligned}$$

In light of (4.24) and (4.31), the following lemma can be established via an induction argument on s and straightforward calculations; its proof is omitted.

Lemma 4.6.1. *Fix an arbitrary $k \in \mathbb{Z}_+$. For each $s = 1, \ldots, p$, $\widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k+s))$ is a vector-valued function whose each entry is a multivariate polynomial in $(z'(k), u_0(k), v_0(k), \widehat{w}(k), \ldots, \widehat{w}(k+s-1))$ and $\boldsymbol{\varphi}_d$.*

Further, in view of $u_0(k+s) = u_0(k)$ for any $s \geq 0$ and a fixed $k \in \mathbb{Z}_+$, we have for each $s = 0, \ldots, p-1$,

$$\widetilde{w}(k+s) = w(k+s) - u_0(k+s)\mathbf{e}_1 = \widehat{w}(k+s) + w_e(k+s) - u_0(k)\mathbf{e}_1 = \widehat{w}(k+s) + d(k+s),$$

where $d(k+s) := w_e(k+s) - u_0(k)\mathbf{e}_1$. Here we recall that for each $s = 0, \ldots, p-1$,

$$w_{e,i}(k+s) = [c_{2,i-1} - c_{2,i}]v_0^2(k+s) + [c_{3,i-1} - c_{3,i}]g, \qquad \forall\ i = 1, \ldots, n,$$

where $v_0(k+s) = v_0(k) + s\tau u_0(k)$. Note that $w_e(k)$ depends on $\boldsymbol{\varphi}$ linearly. Consider the unconstrained MPC model. Define the following augmented matrices and vector: for $s = 1, \ldots, p$

$$\overline{Q}_{\mathbf{z},s} := \begin{bmatrix} Q_{z,s} & \\ & Q_{z',s} \end{bmatrix} \quad ; \quad \overline{Q}_w := \begin{bmatrix} Q_{w,1} & & \\ & \ddots & \\ & & Q_{w,p} \end{bmatrix} \quad ; \quad \widetilde{\mathbf{d}}(k) := \begin{bmatrix} d(k) \\ \vdots \\ d(k+p-1) \end{bmatrix}.$$

For any fixed $k \in \mathbb{Z}_+$, the objective function in the MPC model is written as

$$\begin{aligned} J(\underbrace{\widehat{w}(k), \ldots, \widehat{w}(k+p-1)}_{:=\widehat{\mathbf{w}}(k)}) &= \frac{1}{2}\Big(\sum_{s=1}^{p} \mathbf{z}(k+s)^T \overline{Q}_{\mathbf{z},s}\mathbf{z}(k+s)\Big) \\ &\quad + \frac{\tau^2}{2}[\widehat{\mathbf{w}}(k) + \widetilde{\mathbf{d}}(k)]^T \overline{Q}_w [\widehat{\mathbf{w}}(k) + \widetilde{\mathbf{d}}(k)]. \end{aligned}$$

Substituting the expression for $\mathbf{z}(k+s)$ given by (4.31) into the objective function J, we obtain the objective function written as $J(\widehat{\mathbf{w}})$ for a fixed k. It follows from the previous development and Lemma 4.6.1 that J is a polynomial function in $(\widehat{\mathbf{w}}, \mathbf{z}(k), v_0(k), u_0(k), \boldsymbol{\varphi})$. Moreover, the Hessian of the objective function J with respect to $\widehat{\mathbf{w}}$ is given by

$$HJ(\widehat{\mathbf{w}}) \;=\; \left[\frac{\partial J^2(\widehat{\mathbf{w}})}{\partial \widehat{\mathbf{w}}_i \partial \widehat{\mathbf{w}}_j} \right]_{i,j} \;:=\; \widehat{\mathbf{H}}(\widehat{\mathbf{w}}, \mathbf{z}(k), v_0(k), u_0(k), \boldsymbol{\varphi}).$$

When k is fixed, we write this Hessian as $\widehat{\mathbf{H}}(\widehat{\mathbf{w}}, \mathbf{z}, v_0, u_0, \boldsymbol{\varphi})$ to emphasize its dependence on these variables. Clearly, $\widehat{\mathbf{H}}$ is an analytic, thus smooth, function, and for any $(\widehat{\mathbf{w}}, \mathbf{z}, v_0, u_0)$, $\widehat{\mathbf{H}}(\widehat{\mathbf{w}}, \mathbf{z}, v_0, u_0, \boldsymbol{\varphi})|_{\boldsymbol{\varphi}=0} = \mathbf{H}$, where $\mathbf{H}$ is the constant PD matrix given by (3.22). Moreover, when $\boldsymbol{\varphi} = 0$, the objective function J reduces to the one for the linear vehicle dynamics whose corresponding optimal solution is given in Section 3.5 as

$$\widehat{\mathbf{w}}_*(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi})|_{\boldsymbol{\varphi}=0} \;=\; -\mathbf{H}^{-1}\big(\mathbf{G} \cdot \mathbf{z} - u_0 \cdot \mathbf{g}\big).$$

However, unlike Case (i) where $p = 1$, the closed form expression of a critical point or a local minimizer $\widehat{\mathbf{w}}_*$ is unavailable for $p > 1$, and some implicit function theorems are needed. Instead of applying the classical local implicit function theorem, we consider results on non-local (or global) implicit functions to express $\widehat{\mathbf{w}}_*$ in term of $\mathbf{z}, v_0, u_0$ and $\boldsymbol{\varphi}$, since the variables $\mathbf{z}, v_0, u_0$ can be non-local.

Proposition 4.6.1. *[63, Proposition 6.1] Let $\mathcal{U}_{\mathbf{Z}}$ be a bounded set in $\mathbb{R}^{2n}$, $\mathcal{U}_0$ be a bounded set containing $[a_{0,\min}, a_{0,\max}]$, and $\mathcal{V}_0$ be a bounded set containing $[v_{\min}, v_{\max}]$. Let $\mathcal{U}_{\widehat{\mathbf{W}}}$ be a bounded set in $\mathbb{R}^{np}$ containing all $\widehat{\mathbf{w}}_*(\mathbf{z}, v_0, u_0, 0)$ for all $\mathbf{z} \in \mathcal{U}_{\mathbf{Z}}$, $v_0 \in \mathcal{V}_0$, and $u_0 \in \mathcal{U}_0$. Then for any constant $\widetilde{\lambda}$ with $0 < \widetilde{\lambda} < \lambda_{\min}(\mathbf{H})$, there exists a positive constant $\mu_1 > 0$ such* *that* $\widehat{\mathbf{H}}(\widehat{\mathbf{w}}, \mathbf{z}, v_0, u_0, \boldsymbol{\varphi})$ *is PD and* $\lambda_{\min}(\widehat{\mathbf{H}}(\widehat{\mathbf{w}}, \mathbf{z}, v_0, u_0, \boldsymbol{\varphi})) \geq \widetilde{\lambda}$ *for all* $(\widehat{\mathbf{w}}, \mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) \in \mathcal{U}_{\widehat{\mathbf{w}}} \times \mathcal{U}_{\mathbf{z}} \times \mathcal{V}_0 \times \mathcal{U}_0 \times \mathcal{B}_\infty(0, \mu_1)$*, where* $\mathcal{B}_\infty(0, \mu_1) := \{\boldsymbol{\varphi} \,|\, \|\boldsymbol{\varphi}\|_\infty < \mu_1\}$.

To obtain an implicit form of $\widehat{\mathbf{w}}_*(k)$ in terms of $\mathbf{z}(k), v_0(k), u_0(k)$ from the MPC optimization problem for small $\|\boldsymbol{\varphi}\|$, we shall exploit the following global implicit function theorems [27], [62].

Theorem 4.6.1. *[27, Theorem 2] Consider the sets* $\mathcal{U} \subseteq \mathbb{R}^n$ *and* $\mathcal{V} \subseteq \mathbb{R}^m$*, where* $\mathcal{U}$ *is connected (*$\mathcal{U}$ *and* $\mathcal{V}$ *are not necessarily open). Let* $f : \mathcal{U}' \times \mathcal{V}' \to \mathbb{R}^m$ *be a* C^r*-function with* $r \geq 1$*, where* $\mathcal{U}' \subseteq \mathbb{R}^n$ *and* $\mathcal{V}' \subseteq \mathbb{R}^m$ *are open sets containing* $\mathcal{U}$ *and* $\mathcal{V}$ *respectively. Further, suppose the following hold:*

(i) For some $x_* \in \mathcal{U}$*, there exists exactly one* $y_* \in \mathcal{V}$ *such that* $f(x_*, y_*) = 0$*;*

(ii) For any $(x, y) \in \mathcal{G}'_f := \{(x, y) \in \mathcal{U}' \times \mathcal{V}' : f(x, y) = 0\}$*,* $D_y f(x, y)$ *is invertible;*

(iii) For any sequence $\big((x_k, y_k)\big) \in \mathcal{G}_f := \{(x, y) \in \mathcal{U} \times \mathcal{V} : f(x, y) = 0\}$ *with* $(x_k) \to x_*$*, there exists a subsequence* $(y_{k'})$ *of* (y_k) *such that* $(y_{k'})$ *converges to a point in* $\mathcal{V}$*.*

Then there exists a unique C^r*-function* $g : \mathcal{U} \to \mathcal{V}$ *such that* $f(x, g(x)) = 0, \forall\, x \in \mathcal{U}$*.*

An easily verified condition in replace of condition (iii) in the above theorem is given by the following theorem; its proof resembles that of [27, Theorem 5], which exploits the covering map argument.

Theorem 4.6.2. *[63, Theorem 6.2]] Let* $\mathcal{U} \subseteq \mathbb{R}^n$ *be a connected set, and* $\mathcal{V} \subseteq \mathbb{R}^m$ *be a closed set. Let* $f : \mathcal{U}' \times \mathcal{V}' \to \mathbb{R}^m$ *be a* C^r*-function with* $r \geq 1$*, where* $\mathcal{U}' \subseteq \mathbb{R}^n$ *and* $\mathcal{V}' \subseteq \mathbb{R}^m$ *are open sets containing* $\mathcal{U}$ *and* $\mathcal{V}$ *respectively. Suppose the following hold:*

(i) For some $x_* \in \mathcal{U}$*, there exists exactly one* $y_* \in \mathcal{V}$ *such that* $f(x_*, y_*) = 0$*;*

(ii) *For any* $(x,y) \in \mathcal{G}'_f := \{(x,y) \in \mathcal{U}' \times \mathcal{V}' : f(x,y) = 0\}$*,* $D_y f(x,y)$ *is invertible;*

(iii) *There is a positive constant* ρ *such that* $\|(D_y f(x,y))^{-1}\| \cdot \|D_x f(x,y)\| \le \rho$ *for all* $(x,y) \in \mathcal{G}'_f$*.*

Then there exists a unique C^r *function* $g : \mathcal{U} \to \mathcal{V}$ *such that* $f(x, g(x)) = 0, \forall\, x \in \mathcal{U}$*.*

Using the above theorem, we establish a result on global implication function for $\widehat{\mathbf{w}}_*$ as follows.

Proposition 4.6.2. *[63, Proposition 6.2] Let* $\mathcal{U}_{\mathbf{Z}}$ *be a bounded open convex set in* $\mathbb{R}^{2n}$*, let* $\mathcal{U}_0$ *be a bounded open convex set containing* $[a_{0,\min}, a_{0,\max}]$*, and let* $\mathcal{V}_0$ *be a bounded open convex set containing* $[v_{\min}, v_{\max}]$*. Let* $\mathcal{U}_{\widehat{\mathbf{W}}}$ *be a compact set in* $\mathbb{R}^{np}$ *containing all* $\widehat{\mathbf{w}}_*(\mathbf{z}, v_0, u_0, 0)$ *for all* $\mathbf{z} \in \mathcal{U}_{\mathbf{Z}}$*,* $v_0 \in \mathcal{V}_0$*, and* $u_0 \in \mathcal{U}_0$*. Then there exist a positive constant* $\mu_2 > 0$ *and a unique smooth function* $\mathbf{h} : \mathcal{U}_{\mathbf{Z}} \times \mathcal{V}_0 \times \mathcal{U}_0 \times \mathcal{B}_\infty(0, \mu_2) \to \mathcal{U}_{\widehat{\mathbf{W}}}$ *such that* $\widehat{\mathbf{w}}_* = \mathbf{h}(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi})$ *for all* $(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) \in \mathcal{U}_{\mathbf{Z}} \times \mathcal{V}_0 \times \mathcal{U}_0 \times \mathcal{B}_\infty(0, \mu_2)$*.*

The above proposition implies that the unconstrained nonconvex optimization problem $\min J(\widehat{\mathbf{w}})$ has a unique local minimizer $\widehat{\mathbf{w}}_*$ on $\mathcal{U}_{\widehat{\mathbf{W}}}$ for any fixed $(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) \in \mathcal{U}_{\mathbf{Z}} \times \mathcal{V}_0 \times \mathcal{U}_0 \times \mathcal{B}_\infty(0, \mu_2)$. Hence, for any $(\mathbf{z}(k), v_0(k), u_0(k), \boldsymbol{\varphi}) \in \mathcal{U}_{\mathbf{Z}} \times \mathcal{V}_0 \times \mathcal{U}_0$ at each k,

$$\widehat{\mathbf{w}}_*(k) = \mathbf{h}(\mathbf{z}(k), v_0(k), u_0(k), \boldsymbol{\varphi}), \qquad \widehat{w}_*(k) = \begin{bmatrix} I_n & 0 & \cdots & 0 \end{bmatrix} \mathbf{h}(\mathbf{z}(k), v_0(k), u_0(k), \boldsymbol{\varphi}).$$

Moreover, note that $\mathbf{h}(\mathbf{z}, v_0, u_0, 0) = -\mathbf{H}^{-1}\big(\mathbf{G} \cdot \mathbf{z} - u_0 \mathbf{g}\big)$ for any fixed $(\mathbf{z}, v_0, u_0) \in \mathcal{U}_{\mathbf{Z}} \times \mathcal{V} \times \mathcal{U}_0$. Define

$$\Delta\widehat{\mathbf{h}}(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) := \begin{bmatrix} I_n & 0 & \cdots & 0 \end{bmatrix} \Big(\mathbf{h}(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) - \mathbf{h}(\mathbf{z}, v_0, u_0, 0)\Big).$$

Since $\mathcal{U}_{\mathbf{Z}} \times \mathcal{V}_0 \times \mathcal{U}_0 \times \mathcal{B}_\infty(0,\mu_2)$ is an open convex set, it follows from the Mean-value Theorem that for any fixed $(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) \in \mathcal{U}_{\mathbf{Z}} \times \mathcal{V}_0 \times \mathcal{U}_0 \times \mathcal{B}_\infty(0,\mu_2)$,

$$\Delta\widehat{\mathbf{h}}(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) = \int_0^1 D_{\boldsymbol{\varphi}}\widehat{\mathbf{h}}(\mathbf{z}, v_0, u_0, t\boldsymbol{\varphi})dt \cdot \boldsymbol{\varphi}.$$

Therefore, there exists a positive constant $\varkappa$ such that $\|\Delta\widehat{\mathbf{h}}(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi})\| \le \varkappa\|\boldsymbol{\varphi}\|_\infty$ for all $(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) \in \mathcal{U}_{\mathbf{Z}} \times \mathcal{V}_0 \times \mathcal{U}_0 \times \mathcal{B}_\infty(0,\mu_2)$.

Substituting the above results to the closed loop dynamics (4.26), we obtain

$$\begin{aligned}
\mathbf{z}(k+1) &= \widehat{A}(k)\mathbf{z}(k) + B\Big(\widehat{w}_*(k) + \widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k))\Big) \\
&= \Big(A + v_0(k)\Delta A(\boldsymbol{\varphi}_d)\Big)\mathbf{z}(k) + B\Big[-\begin{bmatrix} I_n & 0 & \dots & 0 \end{bmatrix}\mathbf{H}^{-1}\big(\mathbf{G}\cdot\mathbf{z}(k) - u_0(k)\mathbf{g}\big)\Big] \\
&\qquad + B\Big(\Delta\widehat{\mathbf{h}}(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) + \widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k))\Big) \\
&= \Big[\underbrace{(A + B\mathbf{K})}_{A_{\mathrm{C}}} + v_0(k)\cdot\Delta A(\boldsymbol{\varphi}_d)\Big]\mathbf{z}(k) + B\Big(u_0(k)\cdot\mathbf{d} + \Delta\widehat{\mathbf{h}}(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) + \widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k))\Big),
\end{aligned}$$

where the constant matrix $\mathbf{K}$ and the constant vector $\mathbf{d}$ are given by (3.23), and A_{C} is the closed loop dynamics matrix for the linear vehicle dynamics given by (3.21). This leads to the closed loop dynamics for $p > 1$:

$$\mathbf{z}(k+1) = \Big(A_{\mathrm{C}} + v_0(k)\cdot\Delta A(\boldsymbol{\varphi}_d)\Big)\mathbf{z}(k) + B\Big[u_0(k)\mathbf{d} + \Delta\widehat{\mathbf{h}}(\mathbf{z}(k), v_0(k), u_0(k), \boldsymbol{\varphi})\Big] + B\widetilde{h}_{\boldsymbol{\varphi}_d}(z'(k)) \tag{4.33}$$

for all $(\mathbf{z}, v_0, u_0, \boldsymbol{\varphi}) \in \mathcal{U}_{\mathbf{Z}} \times \mathcal{V}_0 \times \mathcal{U}_0 \times \mathcal{B}_\infty(0,\mu_2)$, where $\mathcal{U}_{\mathbf{Z}}$ is a bounded open convex set in $\mathbb{R}^{2n}$, $\mathcal{U}_0$ is a bounded open convex set containing $[a_{0,\min}, a_{0,\max}]$, and $\mathcal{V}_0$ is a bounded open convex set containing $[v_{\min}, v_{\max}]$.

4.6.2 Local Input-to-state Stability of the Closed Loop System

We give a brief overview of (local) input-to-state stability first. Consider the discrete-time system on $\mathbb{R}^n$:

$$x(k+1) \;=\; f(x(k), u(k), k), \qquad \forall\; k \in \mathbb{Z}_+, \tag{4.34}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{Z}_+ \to \mathbb{R}^n$, and $f(\cdot, \cdot, k)$ is continuous for any fixed $k \in \mathbb{Z}_+$. Let $\mathbf{u} := (u(0), u(1), \ldots)$ be a sequence of vectors in $\mathbb{R}^m$ that represents an input function on $\mathbb{Z}_+$. We assume that $f(0, 0, k) = 0$ for all $k \in \mathbb{Z}_+$ such that $x_e = 0$ is an equilibrium of the system (4.34) under the 0-input, i.e., $\mathbf{u} = 0$. We let $\|\mathbf{u}\|_\infty := \sup\{\|u(k)\| \,:\, k \in \mathbb{Z}_+\}$. Hence, for any $\mathbf{u} \in \ell_\infty^m$, $\|\mathbf{u}\|_\infty < \infty$. For a given initial condition $\xi \in \mathbb{R}^n$ and an input function $\mathbf{u}$, let $x(k, \xi, \mathbf{u})$ denote the trajectory of the system (4.34).

We introduce the notions of $\mathcal{K}$ class of functions [35, pp. 135]. A continuous function $\alpha : \mathbb{R}_+ \to \mathbb{R}_+$ is a $\mathcal{K}$-function if it is strictly increasing on $[0, \infty)$ and $\alpha(0) = 0$. It is a $\mathcal{K}_\infty$-function if it is a $\mathcal{K}$-function and $\alpha(t) \to \infty$ as $t \to \infty$. It is a *positive definite* function if $\alpha(t) > 0$ for all $t > 0$ and $\alpha(0) = 0$. A function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ is a $\mathcal{KL}$-function if (i) for any fixed $t \geq 0$, the function $\beta(\cdot, t)$ is a $\mathcal{K}$-function; and (ii) for any fixed $s \geq 0$, the function $\beta(s, \cdot)$ is decreasing and $\beta(s, t) \to 0$ as $t \to \infty$.

Definition 4.6.1. The time-varying discrete-time system (4.34) is *locally input-to-state stable* (ISS) if there exist a $\mathcal{KL}$-function $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$, a $\mathcal{K}$-function $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$, and two positive constants θ_x, θ_u such that for all ξ with $\|\xi\| \leq \theta_x$ and $\mathbf{u} \in \ell_\infty^m$ with $\|\mathbf{u}\|_\infty \leq \theta_u$, the following holds:

$$\|x(k, \xi, \mathbf{u})\| \;\leq\; \beta(\|\xi\|, k) + \gamma(\|\mathbf{u}\|_\infty), \qquad \forall\; k \in \mathbb{Z}_+.$$

The above definition follows from [35, Definition 5.2] for continuous-time systems and [32, Definition 3.1] for global ISS of discrete-time systems. Also see [25], [65], [74] for details. In what follows, we extend the Lyapunov approach for global ISS in [32, Lemma 3.5] and [30] to local input-to-state stability (ISS) for the time-varying system (4.34); see [31, Lemma 2.3] for a similar local version of the ISS.

Theorem 4.6.3. *Consider the time-varying discrete-time system (4.34) defined by* $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{Z}_+ \to \mathbb{R}^n$*. Suppose there exists a local ISS-Lyapunov function* $V : \mathbb{R}^n \times \mathbb{Z}_+ \to \mathbb{R}_+$ *for the system (4.34), namely, there exist two sets* $\mathcal{D}_x := \{x \in \mathbb{R}^n \mid \|x\| \leq r\}$ *and* $\mathcal{D}_u := \{u \in \mathbb{R}^m \mid \|u\| \leq r_u\}$ *for some positive constants* r *and* r_u*, where* r_u *can be* $+\infty$*, such that the following hold:*

(i) There exist two $\mathcal{K}_\infty$*-functions* α_1 *and* α_2 *such that* $\alpha_1(t) \leq \alpha_2(t), \forall\, t \geq 0$ *and* $\alpha_1(\|x\|) \leq V(x,k) \leq \alpha_2(\|x\|)$ *for all* $x \in \mathcal{D}_x$ *and all* $k \in \mathbb{Z}_+$*;*

(ii) There exist a $\mathcal{K}_\infty$*-function* α_3 *and a* $\mathcal{K}$*-function* σ *such that* $V(f(x,u,k),k+1) - V(x,k) \leq -\alpha_3(\|x\|) + \sigma(\|u\|)$ *for all* $x \in \mathcal{D}_x$ *and* $u \in \mathcal{D}_u$ *and all* $k \in \mathbb{Z}_+$*.*

Then there exist positive constants $\theta_x > 0$ *and* $\theta_u > 0$ *such that the following hold:*

(i) For any ξ *with* $\|\xi\| \leq \theta_x$ *and* $\mathbf{u} = (u(k))_{k \in \mathbb{Z}_+} \in \ell_\infty^m$ *with* $\|\mathbf{u}\|_\infty \leq \theta_u$*,* $x(k,\xi,\mathbf{u}) \in \mathcal{D}_x$ *for all* $k \in \mathbb{Z}_+$*;*

(ii) The system (4.34) is locally input-to-state stable in terms of the positive constants θ_x *and* θ_u *given in Definition 4.6.1.*

Since the closed loop dynamics for $p = 1$ given by (4.30) and that for $p > 1$ given by (4.33) share the similar structure except that the latter holds on a restricted set, we provide the following result for local input-to-state stability under the assumption that the closed loop dynamic matrix A_C under the linear vehicle dynamics given in (3.21) is Schur stable.

Theorem 4.6.4. *[63, Theorem 6.4] Fix $p \in \mathbb{N}$. Suppose the weight matrices $Q_{z,s}, Q_{z',s}$ and $Q_{w,s}$ satisfying* **A.1** *are such that A_C given in (3.21) is Schur stable. Then there exist positive constants μ and ν such that for all $\boldsymbol{\varphi}$ with $\|\boldsymbol{\varphi}\|_\infty \le \mu$, any $v_0(k) \in [v_{\min}, v_{\max}]$ and any $u_0(k)$ with $|u_0(k)| \le \nu$ for all $k \in \mathbb{Z}_+$, the closed loop dynamics given by (4.30) or (4.33) is locally input-to-state stable.*

4.7 Numerical Results

4.7.1 Numerical Experiment Setup and Weight Matrix Design

Numerical tests are carried out to evaluate the performance of the proposed fully distributed schemes and the platooning control for a possibly heterogeneous CAV platoon. We consider a platoon of an uncontrolled leading vehicle labeled by the index 0 and ten CAVs, i.e., $n = 10$. The sample time $\tau = 1s$, and the speed limits $v_{\max} = 27.78m/s$ and $v_{\min} = 10m/s$. Since the physical parameters of CAV platoons as well as algorithm and control design depend heavily on vehicle types, we consider the following three types of CAV platoons: (i) a homogeneous small-size CAV platoon; (ii) a heterogeneous medium-size CAV platoon; and (iii) a homogeneous large-size CAV platoon. Identical minimal (resp. maximal) values of nonlinear dynamics coefficients $c_{2,i}$'s and $c_{3,i}$'s are chosen for the homogeneous small-size (resp. large-size) CAV platoon, whereas inhomogeneous values of $c_{2,i}$'s and $c_{3,i}$'s are chosen for the heterogeneous medius-size CAV platoon. Other parameters for the CAVs and their constraints, i.e., the vehicle length $L_i(m)$, the reaction time $r_i(s)$, the acceleration and deceleration limits $a_{i,\max}(m/s^2)$ and $a_{i,\min}(m/s^2)$, and the desired spacing $\Delta(m)$, are chosen accordingly. See Tables 4.1- 4.2 for the values of these parameters [22], [97].

Table 4.1: Physical parameters for homogeneous small-size and large-size CAV platoons

	L_i	r_i	$a_{i,\min}$	$a_{i,\max}$	$c_{2,i}(\times 10^{-4})$	$c_{3,i}(\times 10^{-2})$	$\Delta(m)$
Small-size	5	1.0	-8	1.4	2.5	0.6	50
Large-size	10	1.25	-6.8	1.4	4.5	1.5	65

The initial state of each CAV platoon is $z(0) = z'(0) = 0$ and $v_i(0) = 25m/s$ for all $i = 0, 1, \ldots, n$. The cyclic-like graph is considered for the vehicle communication network, i.e., the bidirectional edges of the graph are $(1, 2), (2, 3), \ldots, (n-1, n) \in \mathcal{E}$. Following the discussions in [64, Section 6], we choose the MPC horizon p as $1 \le p \le 5$.

We present the choices of weight matrices for each of the abovementioned three CAV platoons. Define

$$
\begin{aligned}
\widetilde{\boldsymbol{\alpha}} &:= \big(38.85, 40.2, 41.55, 42.90, 44.25, 45.60, 46.95, 48.30, 49.65, 51.00\big) \in \mathbb{R}^{10}, \\
\widetilde{\boldsymbol{\beta}} &:= \big(130.61, 136.21, 141.82, 147.42, 153.03, 158.64, 164.24, 169.85, 175.46, 181.06\big) \in \mathbb{R}^{10}, \\
\widetilde{\boldsymbol{\zeta}} &:= \big(62, 74, 90, 92, 106, 194, 298, 402, 454, 480\big) \in \mathbb{R}^{10}.
\end{aligned}
$$

Table 4.2: Physical parameters for a heterogeneous medium-size CAV platoon with $\Delta = 60m$

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$	$i=10$
$L_i(m)$	7	7	7	7	7	7	7	7	7	7
$r_i(s)$	1.21	1.155	0.99	1.045	1.21	1.155	0.99	1.045	1.155	1.045
$a_{i,\min}(m/s^2)$	-8.14	-7.77	-6.66	-7.03	-8.14	-7.77	-6.66	-7.03	-7.77	-7.03
$a_{i,\max}(m/s^2)$	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
$c_{2,i}(\times 10^{-4})$	3.85	3.675	3.15	3.325	3.85	3.675	3.15	3.325	3.675	3.325
$c_{3,i}(\times 10^{-2})$	1.155	1.103	0.945	0.998	1.155	1.103	0.945	0.998	1.103	0.998

For all the three CAV platoons, $\boldsymbol{\alpha}^1 = 6\widetilde{\boldsymbol{\alpha}}$, $\boldsymbol{\beta}^1 = \widetilde{\boldsymbol{\beta}}$, and $\boldsymbol{\zeta}^1 = 0.5\widetilde{\boldsymbol{\zeta}}$ when $p = 1$.

- Homogeneous small-size and heterogenous medium-size CAV platoons: for $p \geq 2$, $\boldsymbol{\alpha}^1 = 9(\widetilde{\boldsymbol{\alpha}} - \mathbf{1})$, $\boldsymbol{\beta}^1 = \widetilde{\boldsymbol{\beta}} - \mathbf{1}$, $\boldsymbol{\zeta}^1 = 0.5(\widetilde{\boldsymbol{\zeta}} - \mathbf{1})$, and

$$\boldsymbol{\alpha}^s = \frac{0.1368}{(s-1)^4} \times \widetilde{\boldsymbol{\alpha}}, \quad \boldsymbol{\beta}^s = \frac{0.044}{(s-1)^4} \times \widetilde{\boldsymbol{\beta}}, \quad \boldsymbol{\zeta}^s = \frac{0.0013}{(s-1)^4} \times \widetilde{\boldsymbol{\zeta}}, \quad s = 2, \ldots, \min(p, 3).$$

- Homogeneous large-size CAV platoon: for $p \geq 2$, $\boldsymbol{\alpha}^1 = 6(\widetilde{\boldsymbol{\alpha}} - \mathbf{1})$, $\boldsymbol{\beta}^1 = \widetilde{\boldsymbol{\beta}} - \mathbf{1}$, $\boldsymbol{\zeta}^1 = 0.5(\widetilde{\boldsymbol{\zeta}} - \mathbf{1})$, and

$$\boldsymbol{\alpha}^s = \frac{0.0684}{(s-1)^4} \times \widetilde{\boldsymbol{\alpha}}, \quad \boldsymbol{\beta}^s = \frac{0.044}{(s-1)^4} \times \widetilde{\boldsymbol{\beta}}, \quad \boldsymbol{\zeta}^s = \frac{0.0013}{(s-1)^4} \times \widetilde{\boldsymbol{\zeta}}, \quad s = 2, \ldots, \min(p, 3).$$

And for all the above CAV platoons: for $p = 4, 5$,

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

The above vectors $\boldsymbol{\alpha}^s, \boldsymbol{\beta}^s, \boldsymbol{\zeta}^s$ define the weight matrices $Q_{z,s}, Q_{z',s}, Q_{w,s}$ for $s = 1, \ldots, 5$, which further yield the closed loop dynamics matrix A_{C}; see the discussions below (3.23). It is shown that when these weights are used, A_{C} is Schur stable for each $p = 1, \ldots, 5$ and each CAV platoon.

We use the same three scenarios described in the linear vehicle dynamics Section 3.6.1. In scenario 3, we use $a_{i,\max} = 1.8m/s^2$ for each vehicle.

4.7.2 Performance of the Proposed Fully Distributed Scheme

As indicated in Section 4.5.2, when $p = 1$, the underlying MPC optimization problem (4.11) is a convex QCQP, for which the generalized Douglas-Rachford splitting method based fully distributed algorithm developed in [64] is used. In what follows, we focus on $p > 1$.

When $p > 1$, the underlying MPC optimization problem (4.11) is nonconvex, and the sequential convex programming and Douglas-Rachford splitting method based fully distributed scheme is applied (cf. Algorithm 10). To apply this algorithm, we discuss the choices of the smooth functions $g_{i,s}$ and the convex function $r_{i,s}$ for the (approximated) nonconvex constraint sets $\mathcal{Y}_i$ and $\mathcal{Z}_i$, where $i = 1, \ldots, n$. In view of the definition of $\mathcal{Y}_i$ given before (4.20), we see that $\mathcal{Y}_i = \{\mathbf{u}_i \,|\, v_{\min} - q_{i,j}(\mathbf{u}_i) \leq 0,\ q_{i,j}(\mathbf{u}_i) - v_{\max} \leq 0,\ j = 1, \ldots, p\}$, where $q_{i,j}(\cdot)$ is given by (4.20). Define $g_{i,s}(\mathbf{u}_i) := v_{\min} - q_{i,j}(\mathbf{u}_i)$, and $r_{i,s}(\mathbf{u}_i) :\equiv 0$ for $s = 1, \ldots, p$; $g_{i,s}(\mathbf{u}_i) :\equiv 0$, and $r_{i,s}(\mathbf{u}_i) := -q_{i,j}(\mathbf{u}_i) + v_{\max}$ for $s = p+1, \ldots, 2p$. Then $\mathcal{Y}_i = \{\mathbf{u}_i \,|\, g_{i,s}(\mathbf{u}_i) - r_{i,s}(\mathbf{u}_i) \leq 0,\ s = 1, \ldots, 2p\}$. Similarly, let $g'_{i,s}(\mathbf{u}_{i-1}, \mathbf{u}_i)$ be the right hand side of (4.21), and $r'_{i,s}(\mathbf{u}_{i-1}, \mathbf{u}_i) \equiv 0$. Then $\mathcal{Z}_i = \{\widehat{\mathbf{u}}_i \,|\, g'_{i,s}(\widehat{\mathbf{u}}_i) - r'_{i,s}(\widehat{\mathbf{u}}_i) \leq 0,\ s = 1, \ldots, p\}$. The gradient of these functions are given in Section 4.5.3. Furthermore, the Lipschitz constants L_{J_i}'s and $L_{g_{i,s}}$'s are given by $\nu_p \| H J_i(\widehat{\mathbf{u}}_i) \|_2$ and $0.9 \| H g_{i,s}(\widehat{\mathbf{u}}_i) \|_2$, where $\nu_p = 0.8$ for $p = 2, 3$ and $\nu_p = 0.9$ for $p = 4, 5$ respectively, and Hf denotes the Hessian of a real-valued smooth function f. The reasons for each Hessian scaled by these factors are twofold: (i) the 2-norm of Hessian is conservative; and (ii) the scaled Hessian leads to faster convergence.

Initial guess warm-up. To achieve real-time computation of the proposed distributed scheme (i.e., Algorithm 10), we exploit the initial guess warm-up technique for both the linear stage (cf. Line 2) and the inner loop of the SCP-DR stage (cf. Lines 6-14). For the former stage, see [64, Section 6.2] for its warm-up scheme. We discuss a warm-up scheme for the latter stage. Recall that the inner loop solves the following convex optimization problem: $\min_{y=(y_i)\in\mathcal{A}} \sum_{i=1}^{n} f_i(y_i) + \boldsymbol{\delta}\mathcal{C}_i(y_i)$, where for each $i = 1, \ldots, n$, $f_i(y_i) :=$

Table 4.3: Error tolerances for outer and inner loops at different MPC horizon p's

	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$
Outer loop	2.5×10^{-3}	6.5×10^{-3}	7.5×10^{-3}	1.0×10^{-2}	1.25×10^{-2}
Inner loop	NA	4.0×10^{-3}	5.0×10^{-3}	7.5×10^{-3}	1.0×10^{-2}

$J_i(\widehat{\mathbf{u}}_i^k) + d_{J_i}^T(\widehat{\mathbf{u}}_i^k)(y_i - \widehat{\mathbf{u}}_i) + \frac{L_{J_i}}{2}\|y_i - \widehat{\mathbf{u}}_i^k\|_2^2$, and $\mathcal{C}_i$ is the intersection of the box-constraint set $\mathcal{X}_i$ corresponding to the control constraint and a quadratically constrained convex set corresponding to the (approximated) velocity and safety distance constraints; see Section 4.5.2 for details. Since the (approximated) velocity and safety distance constraints are often inactive, we replace $\mathcal{C}_i$ by $\mathcal{X}_i$ in a warm-up scheme. Further, the generalized Douglas-Rachford scheme given by (4.16) is used to solve $\min_{y=(y_i)\in\mathcal{A}} \sum_{i=1}^n f_i(y_i) + \boldsymbol{\delta}\mathcal{X}_i(y_i)$ in a fully distributed manner by replacing $\mathcal{C}_i$ by $\mathcal{X}_i$. Since f_i and the box constraint set $\mathcal{X}_i$ are fully decoupled, solving the proximal operator based optimization problem in this scheme boils down to solving finitely many decoupled univariate optimization problems of the form: $\min_{t\in[c,d]} at^2 + bt + e$, where $t \in \mathbb{R}$, and $a, b, c, d, e \in \mathbb{R}$ are given constants with $a > 0$. Such a univariate optimization problem has a simple closed-form solution, which considerably reduces computation load of the Douglas-Rachford scheme. Numerical tests show that the proposed warm-up scheme significantly improves computation time and solution quality.

Performance of distributed schemes. We implement the proposed fully distributed algorithms via MATLAB on a computer with 4-cores processor: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz and RAM: 16.0GB. These distributed algorithm are tested for the three types CAV platoons, namely homogeneous small-size and large-size CAV platoons and a heterogeneous medium-size CAV platoon, on Scenarios 1-3 for different

MPC horizon p's. The proposed initial guess warm-up schemes are used with the error tolerance give by 10^{-7} for all the cases. Moreover, we choose $\alpha = 0.9$ and $\rho = 0.1$ for the proximal operator based Douglas-Rachford scheme in all of these algorithms. Further, the stopping criteria are characterized by the minimum of absolute and relative errors of two neighboring iterates for $p = 2, 3$, whereas for $p = 4, 5$, these criteria are characterized by absolute errors of two neighboring iterate. The list of error tolerances for the outer and inner loop (for all the three types of CAV platoons) at p's is shown in Table 4.3. Note that there is no inner loop when $p = 1$, since the underlying MPC optimization problem is a convex QCQP and solved via the fully distributed scheme given in [64]. A summary of mean and variance of computation time per CAV for different CAV platoons with different p's on the three scenarios is displayed in Tables 4.4- 4.6. Moreover, to evaluate the numerical accuracy of the proposed schemes for $p = 1$, we compute the relative error between the numerical solution from the distributed schemes and that from a high precision centralized scheme when the latter solution, treated as a true solution, is nonzero; see Tables 4.7. Note that for $p \geq 2$, a true solution is hard to compute even in a centralized manner.

Table 4.4: Scenario 1: computation time per CAV (sec)

MPC horizon	Small-size		Medium-size		Large-size	
	Mean	Variance	Mean	Variance	Mean	Variance
$p = 1$	0.0952	3.43×10^{-4}	0.1333	1.44×10^{-4}	0.1087	2.32×10^{-4}
$p = 2$	0.1616	1.6×10^{-3}	0.2795	4.5×10^{-3}	0.2759	1.6×10^{-3}
$p = 3$	0.1721	1.40×10^{-3}	0.2673	4.11×10^{-3}	0.2667	2.35×10^{-3}
$p = 4$	0.1665	6.33×10^{-4}	0.2535	2.02×10^{-3}	0.3038	0.1440
$p = 5$	0.2243	0.2340	0.3056	0.4440	0.3296	0.4240

Table 4.5: Scenario 2: computation time per CAV (sec)

MPC horizon	Small-size		Medium-size		Large-size	
	Mean	Variance	Mean	Variance	Mean	Variance
$p = 1$	0.1098	7.33×10^{-4}	0.1421	2.55×10^{-4}	0.1204	7.54×10^{-4}
$p = 2$	0.1771	1.2×10^{-3}	0.2857	6.7×10^{-3}	0.2814	3.3×10^{-3}
$p = 3$	0.1939	4.83×10^{-4}	0.2804	2.78×10^{-3}	0.2734	1.62×10^{-3}
$p = 4$	0.2241	9.58×10^{-3}	0.3165	5.93×10^{-3}	0.3681	0.0241
$p = 5$	0.2418	0.0113	0.3051	0.0109	0.3449	0.0150

Table 4.6: Scenario 3: computation time per CAV (sec)

MPC horizon	Small-size		Medium-size		Large-size	
	Mean	Variance	Mean	Variance	Mean	Variance
$p = 1$	0.1109	1.2×10^{-3}	0.1408	4.09×10^{-4}	0.1125	8.38×10^{-4}
$p = 2$	0.1559	2.3×10^{-3}	0.2528	6.3×10^{-3}	0.2503	4.8×10^{-3}
$p = 3$	0.2257	0.1320	0.2398	4.91×10^{-3}	0.2437	3.98×10^{-3}
$p = 4$	0.2053	7.73×10^{-3}	0.2883	9.73×10^{-3}	0.3216	0.0132
$p = 5$	0.2256	0.0136	0.2882	0.0135	0.3250	0.0249

The numerical results show that for each p and each CAV platoon type, the mean computation time is less than $0.369s$ and thus less than the reaction time r_i or sample time τ with overall fairly small variances, for all the three scenarios. Indeed, the computation time for $p = 1$ is the least and becomes larger for a higher p for most cases. Further, the computation times vary for different CAV vehicle types. In particular, the computation time for the small-size platoon is less than that of the medium-size and the large-size platoons. This is because the nonlinear effects play an increasing role in the latter types of platoons when $c_{2,i}$ and $c_{3,i}$ become larger. Additionally, the heterogeneous dynamics in the middle-size platoon also require more computation. Besides, the numeri-

Table 4.7: Relative numerical error for $p = 1$

Scenarios	Small-size		Medium-size		Large-size	
	Mean	Variance	Mean	Variance	Mean	Variance
Sc. 1	1.07×10^{-3}	1.44×10^{-7}	5.66×10^{-4}	1.24×10^{-6}	5.29×10^{-4}	5.14×10^{-8}
Sc. 2	9.11×10^{-4}	3.82×10^{-6}	1.11×10^{-3}	7.54×10^{-6}	4.38×10^{-4}	2.73×10^{-8}
Sc. 3	1.47×10^{-3}	3.51×10^{-6}	6.85×10^{-4}	8.41×10^{-7}	5.85×10^{-4}	2.45×10^{-7}

Table 4.8: Maximum steady state error of spacing (m)

MPC horizon	Scenarios 1-2 (among 10 vehicles)			Scenario 3 (among 9 vehicles)		
	Small-size	Medium-size	Large-size	Small-size	Medium-size	Large-size
$p = 1$	0.0571	0.0941	0.1138	0	0.0431	0
$p = 2$	0.1107	0.1507	0.2738	0.1003	0.1277	0.2199
$p = 3$	0.1122	0.1530	0.2793	0.1051	0.1369	0.2374
$p = 4$	0.1321	0.1687	0.2831	0.1142	0.1363	0.2106
$p = 5$	0.1438	0.1880	0.3052	0.1438	0.1538	0.2342

cal accuracy is satisfactory for $p = 1$ as shown in Table 4.7. Hence, we conclude that the proposed distributed schemes are suitable for real-time computation of a heterogenous or homogeneous CAV platoon with satisfactory numerical precision.

4.7.3 Performance of CAV Platooning Control

We evaluate the closed-loop performance of the proposed CAV platooning control with different MPC horizon p's for different CAV platoons on the three scenarios mentioned before. For each CAV platoon and scenario, we consider 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 $u_i(k)$, $i = 1, \dots, n$ for $p = 1, 2, 3, 4, 5$.

Steady state error. When $(c_{2,i}, c_{3,i}) \neq 0$ and $u_0(k) = 0$ and $v_0(k) = v_{0,\infty} > 0$ for all large k, it is observed from the numerical tests that when the closed-loop dynamics of the CAV platoon reaches its steady state $(z_{ss}, z'_{ss}) \in \mathbb{R}^n \times \mathbb{R}^n$, i.e., $(z(k), z'(k))$ becomes the constant vector (z_{ss}, z'_{ss}) for all large k, z_{ss} is nonzero. Physically, the nonzero steady state is due to nonlinear vehicle dynamics and the PD-like control structure of the MPC control scheme. To illustrate this phenomenon, consider the closed-loop dynamics in (4.30) for $p = 1$. (It is noted that only for $p = 1$, we have a closed-form expression for the closed-loop dynamics.) Since $(z_i(k), z'_i(k))$ is constant for all large k when a CAV platoon reaches its steady state, it follows from (4.22) that $w_i(k) - [c_{2,i-1}v_{i-1}^2(k) - c_{2,i}v_i^2(k)] - [c_{3,i-1} - c_{3,i}]g = 0$ for all large k. It is easy to see from (4.22) that $z'_{ss} = 0$. Let $\mathbf{z}_{ss} := (z_{ss}, z'_{ss}) = (z_{ss}, 0)$. In view of $u_0(k) = 0$ for all large k, we deduce via (4.30) that $\mathbf{z}_{ss} = \big(A_{\mathrm{C}} + v_{0,\infty}\Delta\bar{A}(\boldsymbol{\varphi}_d)\big)\mathbf{z}_{ss} - B\widehat{W}Q_w w_{e,\infty}$, where $w_{e,\infty}$ is defined in the way of $w_e(k)$ by setting $v_0(k) \equiv v_{0,\infty}$. Since $\mathbf{z}_{ss} = (z_{ss}, 0)$, we see that $(I - A_{\mathrm{C}})\mathbf{z}_{ss} = -B\widehat{W}Q_w w_{e,\infty}$. By the expression for A_{C} given in (4.28), we obtain $\frac{1}{2}B\widehat{W}Q_z z_{ss} = -B\widehat{W}Q_w w_{e,\infty}$. Since $B\widehat{W}$ has full column rank and Q_z, Q_w are diagonal PD, we further have $\frac{1}{2}Q_z z_{ss} = -Q_w w_{e,\infty}$ or equivalently $z_{ss} = -2Q_z^{-1}Q_w w_{e,\infty}$. Since $w_{e,\infty} \neq 0$, we conclude that $z_{ss} \neq 0$. Note that $w_{e,\infty}$ depends on $c_{2,i-1} - c_{2,i}$ and $c_{3,i-1} - c_{3,i}$ and $v_{0,\infty}$ with $c_{2,0} = c_{3,0} = 0$. Hence, if a CAV platoon is homogeneous, then $w_{e,\infty}$, and thus z_{ss}, is a multiple of $\mathbf{e}_1$. Moreover, in light of $Q_z^{-1}Q_w = \mathrm{diag}(\frac{\zeta_1}{\alpha_1}, \ldots, \frac{\zeta_n}{\alpha_n})$, it is observed that $|(z_{ss})_i|$ will be smaller for a large α_i and a small ζ_i. This observation agrees with numerical results. In addition, when $p \geq 2$, similar observations are made although the closed form expression of z_{ss} is hard to obtain. The maximum steady state error of spacing, i.e., $\|z_{ss}\|_\infty$, is displayed in Table 4.8 for different CAV platoons, different p's, and the three scenarios. Note that in Scenario 3, we consider the steady state errors for $S_{i-1,i}$, $i = 2, 3, \ldots, 9$ since $S_{0,1}$ does not reach its steady state in this scenario.

We present the closed-loop performance only for $p = 1$ and $p = 5$ for each type of CAV platoons in each scenario because of the length limit; see Figures 4.1-4.9. The closed-loop performance in each scenario is commented as follows:

(i) Scenario 1. Figures 4.1, 4.2, and 4.3 show the MPC control performance of the homogenous small-size CAV platoon, the heterogeneous medium-size CAV platoon, and the homogenous large-size CAV platoon in Scenario 1, respectively. It can be seen that the spacing between the leading vehicle and the first CAV, i.e., $S_{0,1}$, in all the CAV platoons has small deviations (less than $0.5m$) from the desired spacing Δ when the leading vehicle takes instantaneous acceleration or deceleration. Further, when $p = 1$, the spacings between the other CAVs in the two homogeneous CAV platoons remain the desired constant Δ, and there are small deviations from the desired spacing Δ for the other CAVs in the heterogeneous CAV platoon or the two homogeneous CAV platoons when $p = 5$. In all the cases, the convergence to the steady states is fast (within 15 secs) and the steady state errors in spacing are nonzero but are small; see Table 4.8. In fact, the maximum steady state errors increase as p becomes larger; compared with the desired spacing $\Delta = 50m$, $60m$ or $65m$, the largest relative error $\frac{\|z_{ss}\|_\infty}{\Delta} \leq 0.47\%$ for all the three types of CAV platoons. Lastly, the time history of speed and control input demonstrates satisfactory performance. In particular, it is observed that all the CAVs show the same speed change and almost identical control, implying that the CAV platoon performs a nearly coordinated motion under the proposed platooning control.

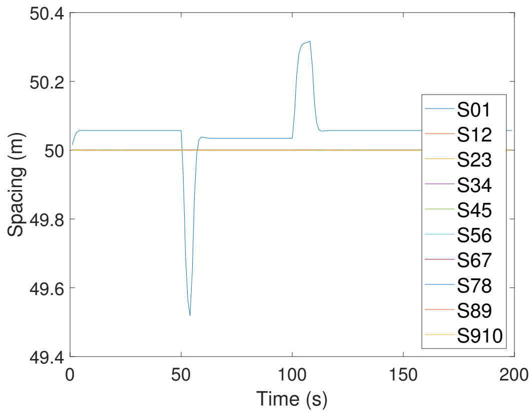
(ii) Scenario 2. Figures 4.4, 4.5, and 4.6 display the MPC control performance of the homogenous small-size, the heterogeneous medium-size, and the homogenous large-size CAV platoons in Scenario 2, respectively, where the leading vehicle undertakes periodic acceleration/deceleration. In all the cases, $S_{0,1}$ demonstrates the largest fluctuations whose maximum magnitude of deviations is $0.25m$ when $\Delta = 50m$, $0.3m$ when $\Delta = 60m$, and $0.5m$ when $\Delta = 65m$. Besides, all the CAV platoons demonstrates nearly coordinated motions. For example, when $p = 1$, the spacings $S_{i-1,i}$ for $i = 2, \ldots, 10$ remain the desired constant for the two homogeneous CAV platoons, and they have small deviations from the desired spacing for the heterogeneous CAV platoon and $p = 5$ of the two homogeneous CAV platoons. Moreover, the fluctuations of $S_{0,1}$ and other $S_{i,i+1}$'s quickly converge to their steady states within $15s$ when the leading vehicle stops its periodical acceleration. The steady state errors in spacing are as same as those in Scenario 1. The time history of speed and control input shows nearly identical behaviors for all the CAVs in each case.

(iii) Scenario 3. Figures 4.7, 4.8, and 4.9 show the control performance of the homogenous small-size, the heterogeneous medium-size, and the homogenous large-size CAV platoons in Scenario 3, respectively, where the leading vehicle undergoes various traffic oscillations through the time window of $45s$. It is observed that $S_{0,1}$ demonstrates the largest spacing variations with the maximum magnitude less than or equal to $0.25m$ when $\Delta = 50m$, $0.3m$ when $\Delta = 60m$, and $0.46m$ when $\Delta = 65m$; the other spacings $S_{i-1,i}$, $i = 2, \ldots, 10$ either are the desired constant or demonstrate nearly constant deviations with maximum magnitude less than $0.14m$, in spite of the oscillation of $S_{0,1}$. Further, the spacings $S_{i-1,i}$, $i = 2, \ldots, 10$ almost reach steady states between $5s$ and $25s$ and after $k = 35$. The maximum steady state errors of these spacings are shown in Table 4.8. It is seen that the maximum steady state error often appears at $S_{1,2}$. Compared with the desired spacing $\Delta = 50m$, $60m$ or $65m$, the largest relative error $\frac{\|z_{ss}\|_\infty}{\Delta} \leq 0.37\%$ for all the three types of CAV platoons in Scenario 3. Finally, all the CAV platoons demonstrates nearly coordinated motions.

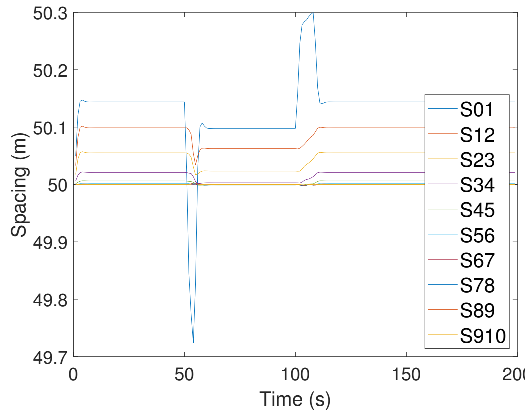
Consequently, the proposed platooning control effectively mitigates traffic oscillations of the spacing and vehicle speed of the CAV platoons of different types with small or almost negligible steady state errors. In fact, it achieves nearly consensus motions of the entire CAV platoons even under some perturbations.

4.8 Summary

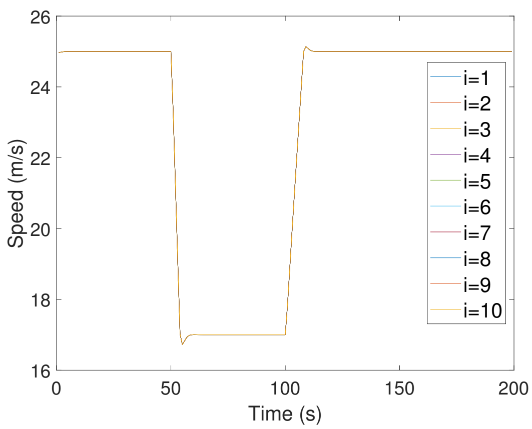
This chapter develops a nonconvex, fully distributed optimization based MPC scheme for CAV platooning control of a heterogeneous CAV platoon under the nonlinear vehicle dynamics. Different from the existing research on the linear vehicle dynamics, various new techniques are exploited to address several major challenges induced by the nonlinear vehicle dynamics, including distributed algorithm development for the coupled nonconvex MPC optimization problem, and stability analysis of time-varying nonlinear closed-loop dynamics. For the former, we apply locally coupled optimization and sequential convex programming for distributed algorithm development. For the latter, global implicit function theorems and Lyapunov theory for input-to-state stability, among many other techniques, are invoked for closed loop stability analysis. Extensive numerical tests are conducted to illustrate the effectiveness of the proposed fully distributed schemes and CAV platooning control for homogeneous and heterogeneous CAV platoons in different scenarios.



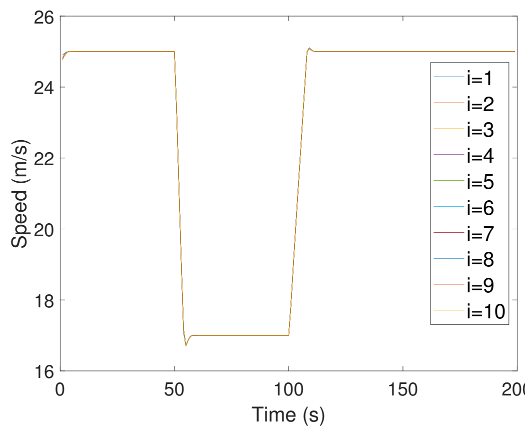
(a) Time history of spacing changes.



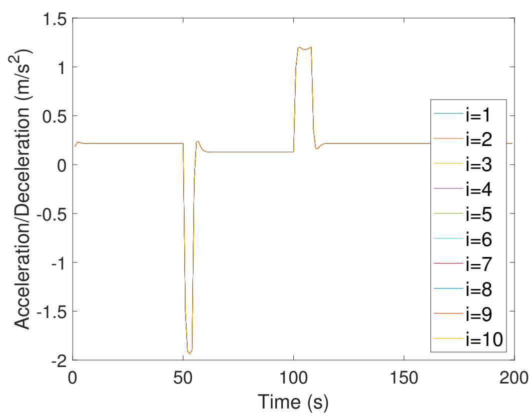
(b) Time history of spacing changes.



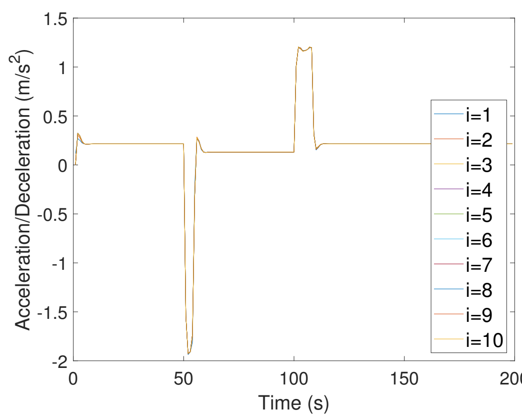
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

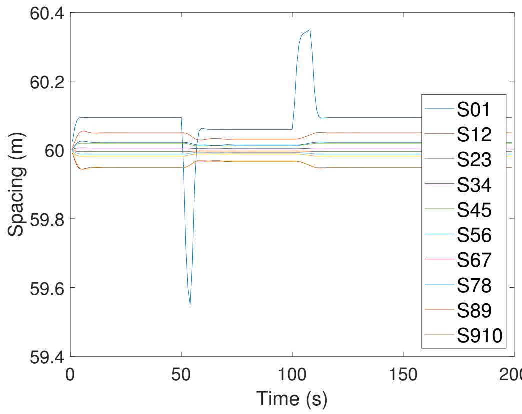


(e) Time history of control input.

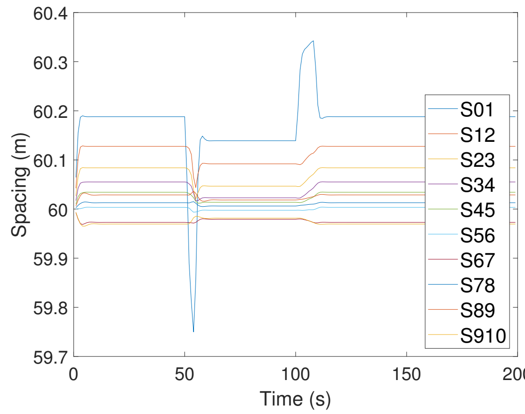


(f) Time history of control input

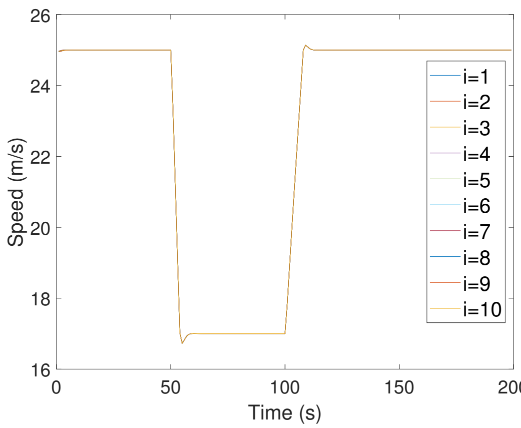
Figure 4.1: Scenario 1 for the homogeneous small-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column).



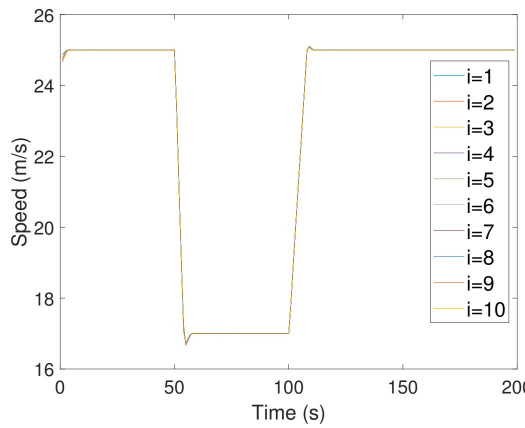
(a) Time history of spacing changes.



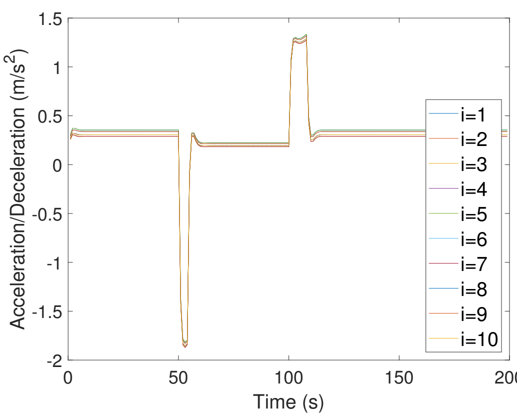
(b) Time history of spacing changes.



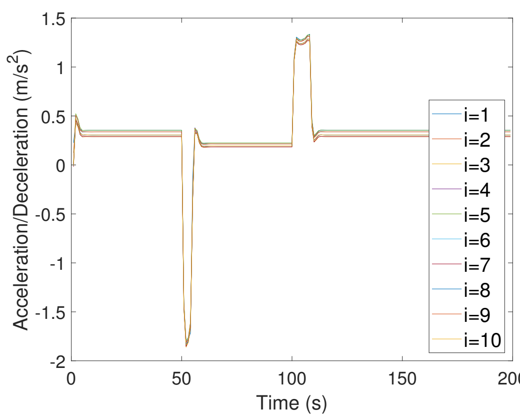
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

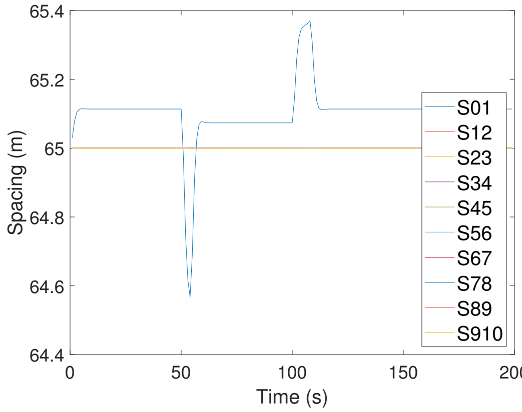


(e) Time history of control input.

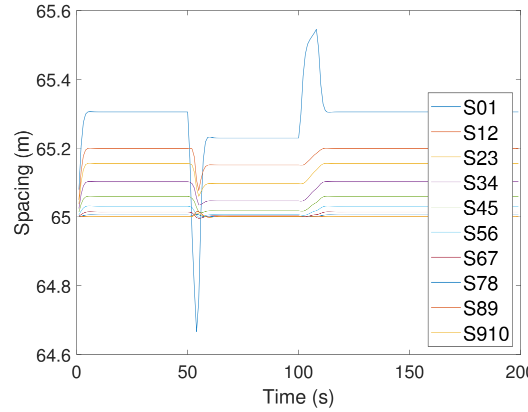


(f) Time history of control input

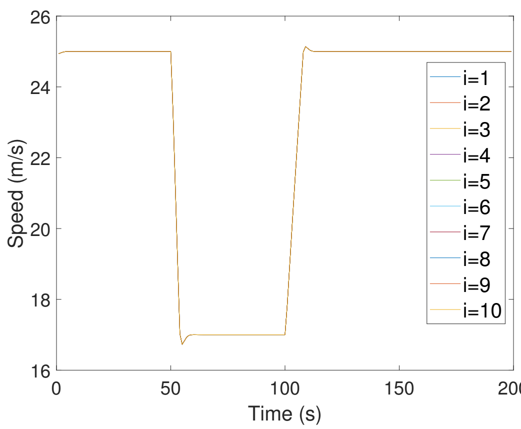
Figure 4.2: Scenario 1 for the heterogeneous medium-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column).



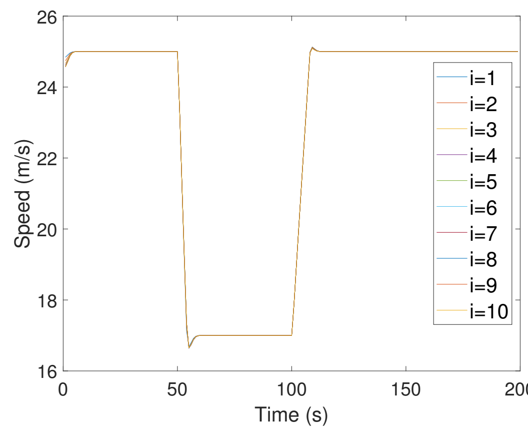
(a) Time history of spacing changes.



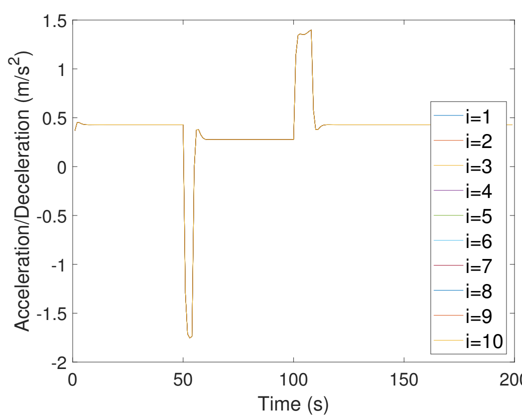
(b) Time history of spacing changes.



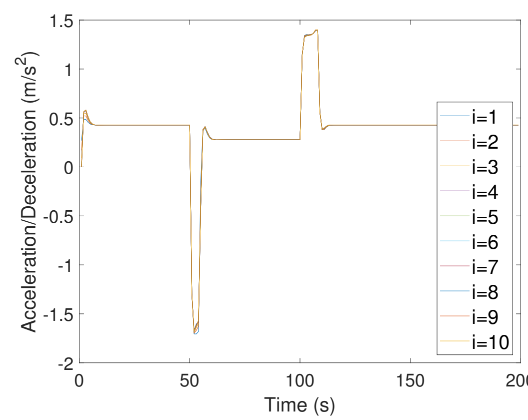
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

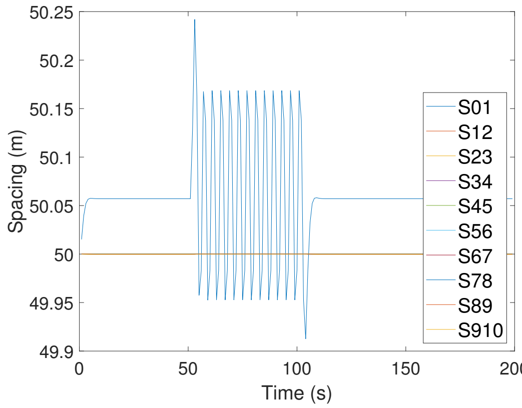


(e) Time history of control input.

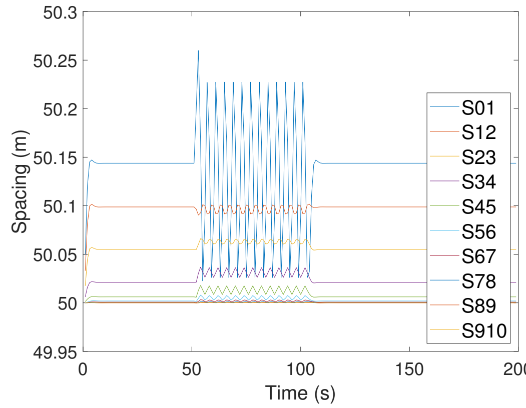


(f) Time history of control input

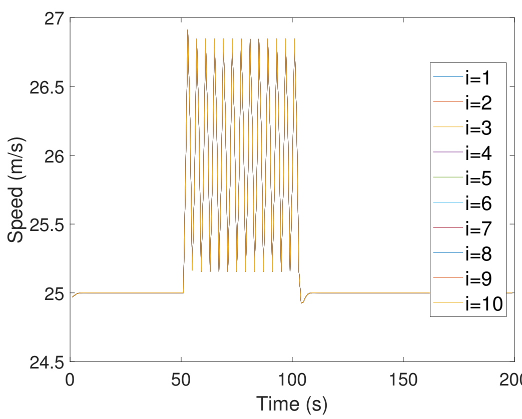
Figure 4.3: Scenario 1 for the homogeneous large-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column).



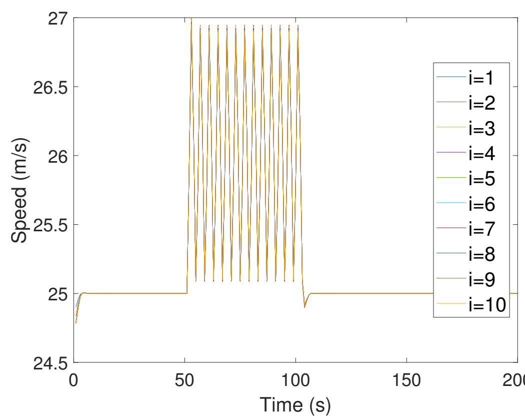
(a) Time history of spacing changes.



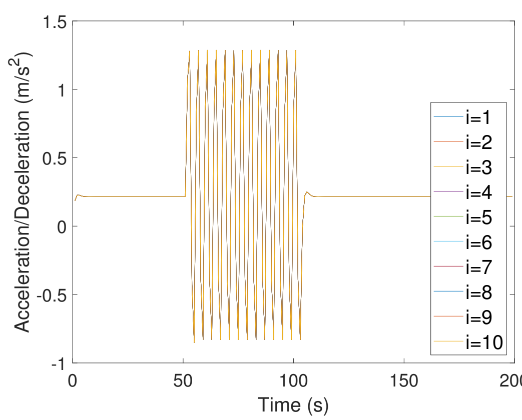
(b) Time history of spacing changes.



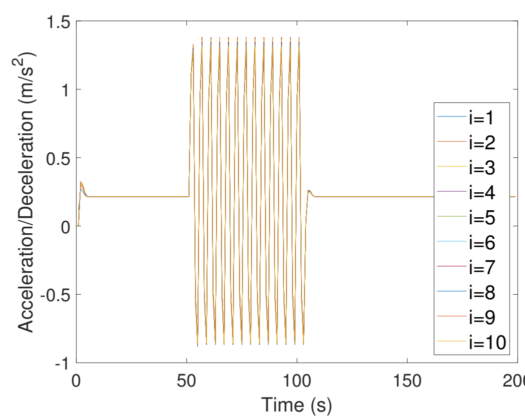
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

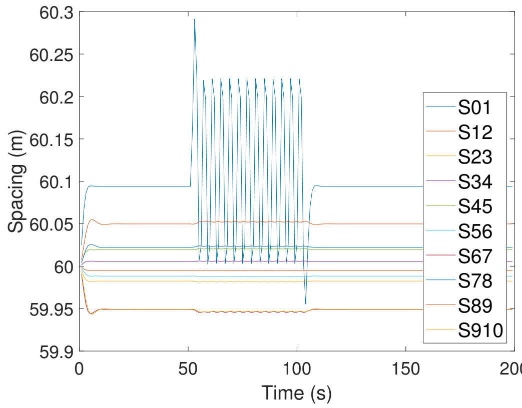


(e) Time history of control input.

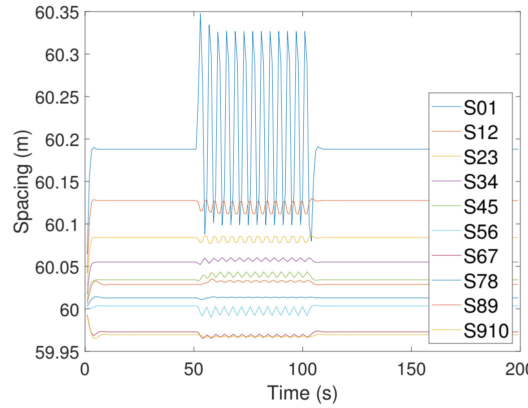


(f) Time history of control input

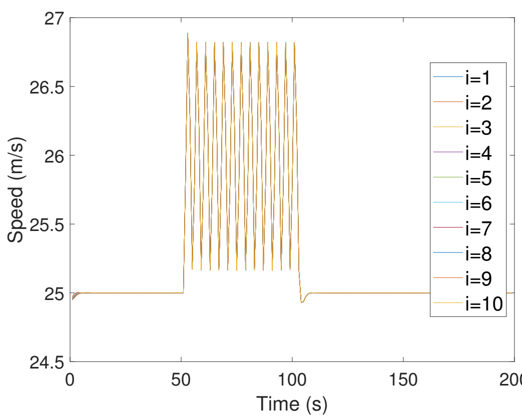
Figure 4.4: Scenario 2 for the homogeneous small-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column).



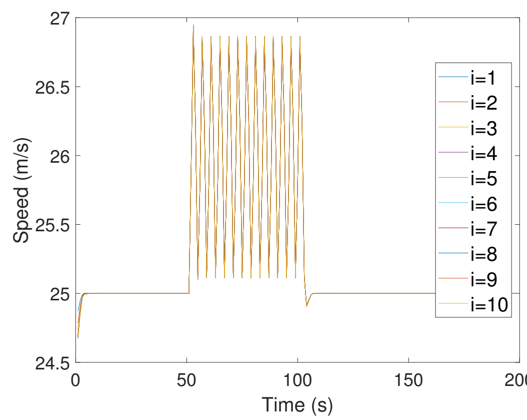
(a) Time history of spacing changes.



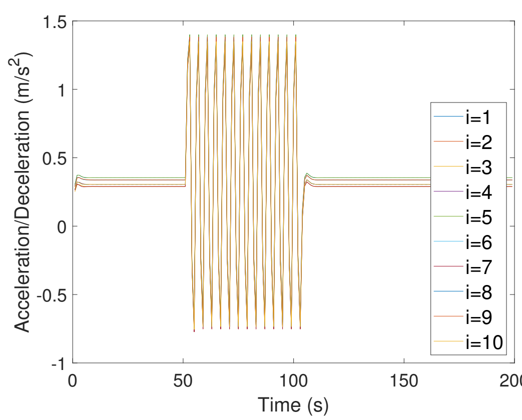
(b) Time history of spacing changes.



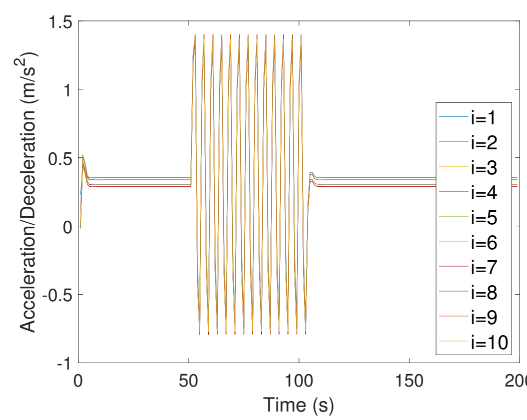
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

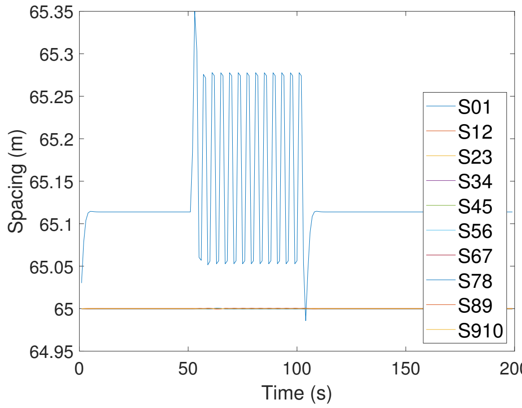


(e) Time history of control input.

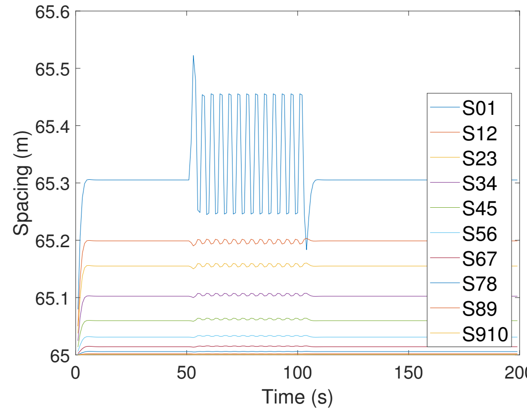


(f) Time history of control input

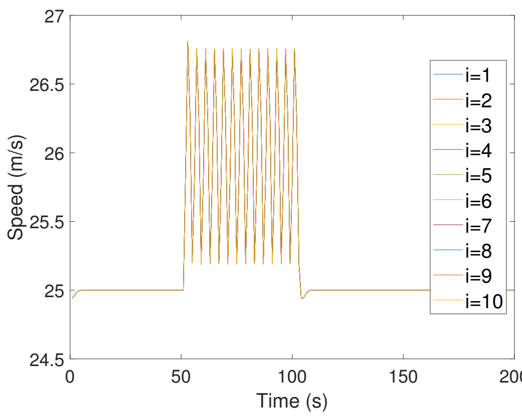
Figure 4.5: Scenario 2 for the heterogeneous medium-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column).



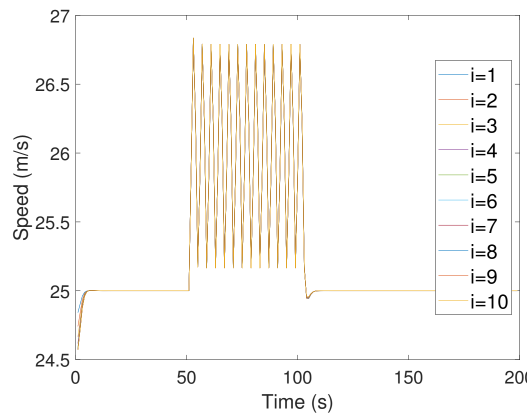
(a) Time history of spacing changes.



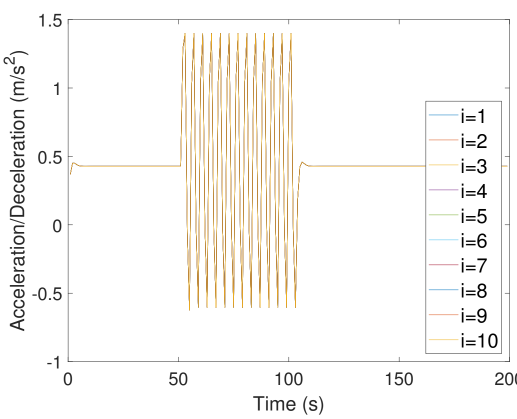
(b) Time history of spacing changes.



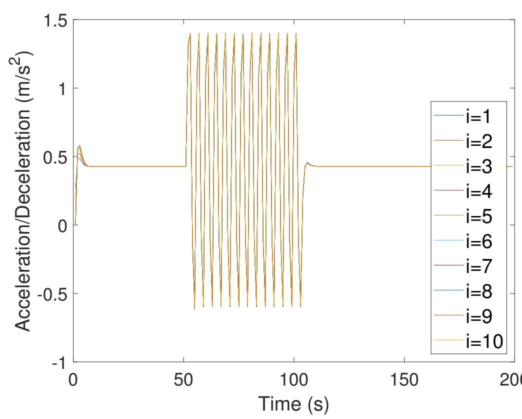
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

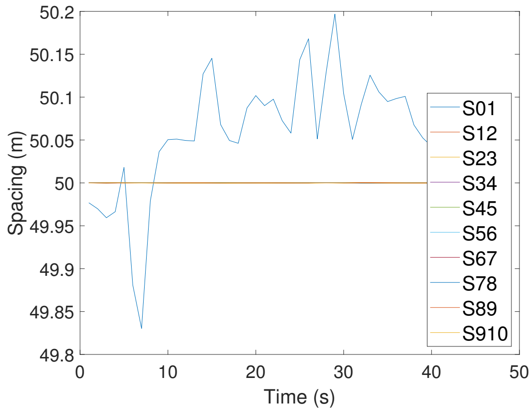


(e) Time history of control input.

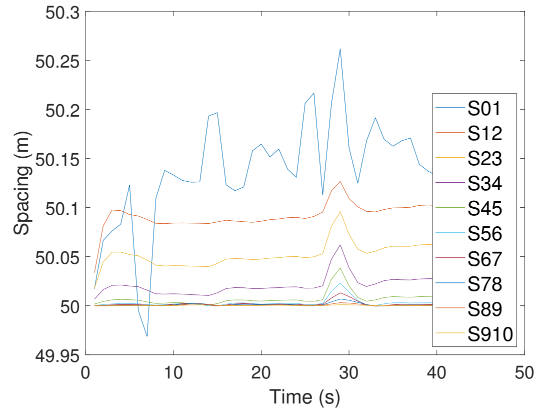


(f) Time history of control input

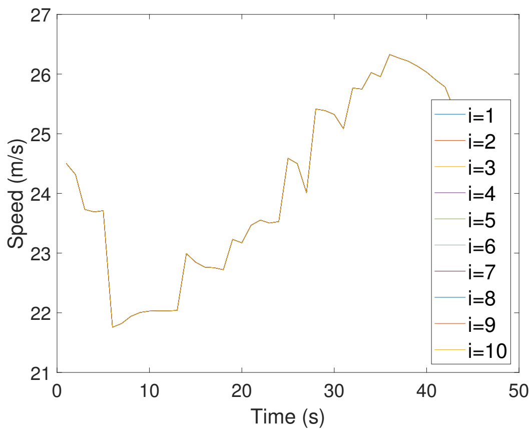
Figure 4.6: Scenario 2 for the homogeneous large-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column).



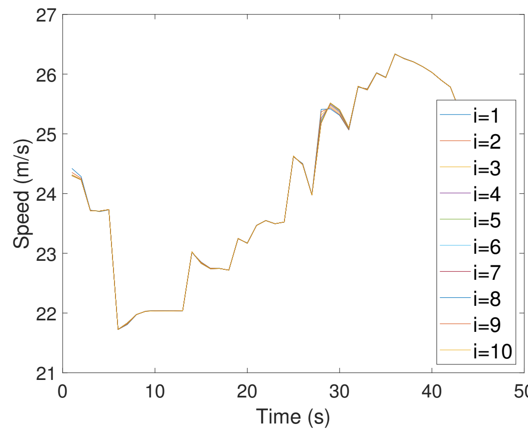
(a) Time history of spacing changes.



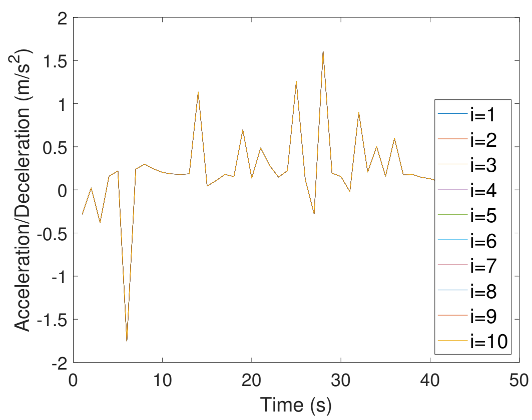
(b) Time history of spacing changes.



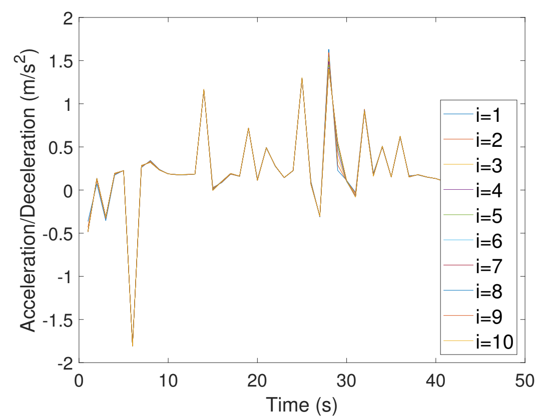
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

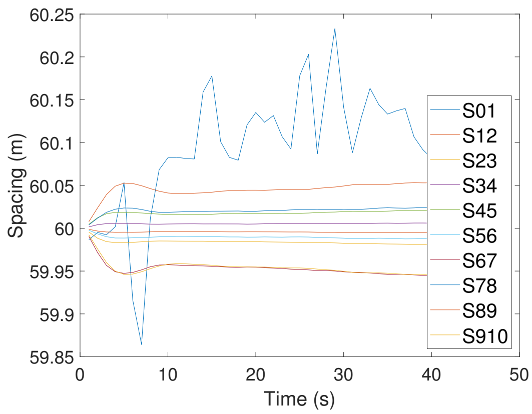


(e) Time history of control input.

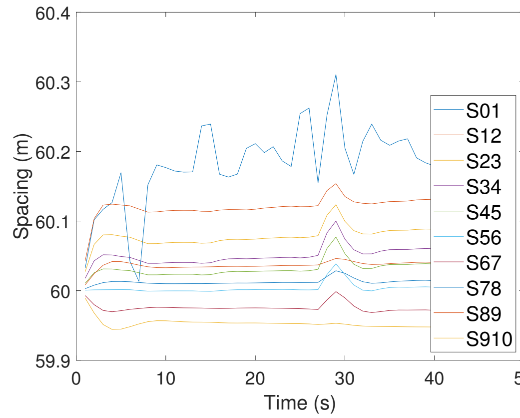


(f) Time history of control input

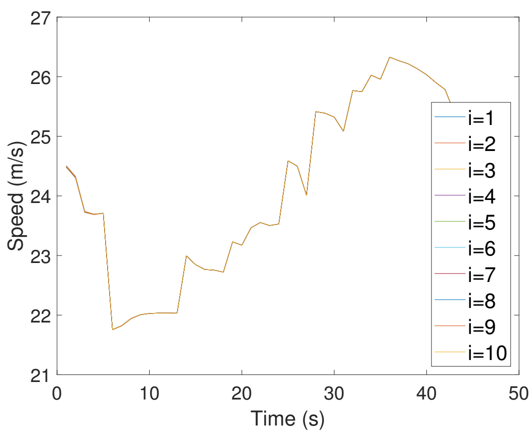
Figure 4.7: Scenario 3 for the homogeneous small-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column).



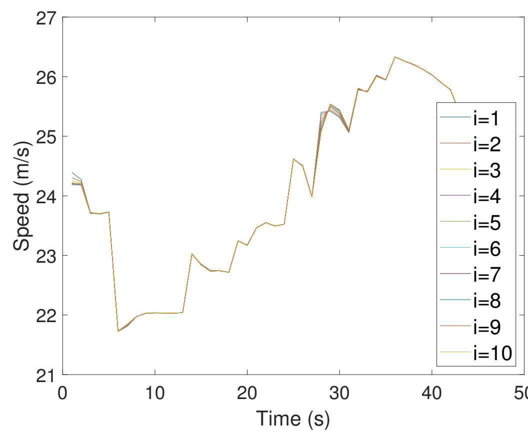
(a) Time history of spacing changes.



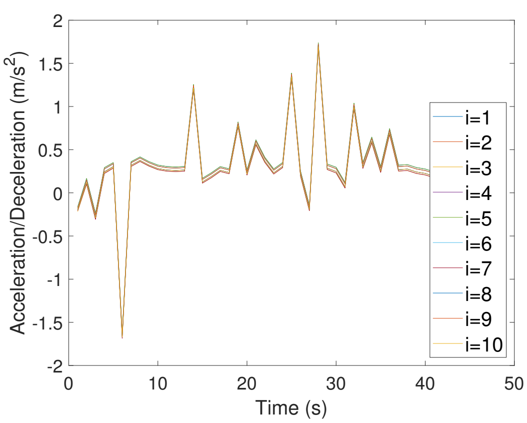
(b) Time history of spacing changes.



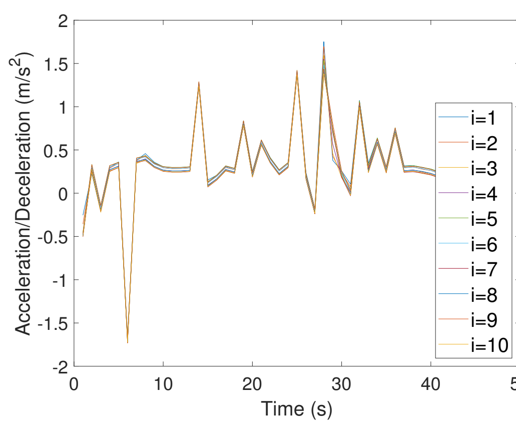
(c) Time history of vehicle speed.



(d) Time history of vehicle speed.

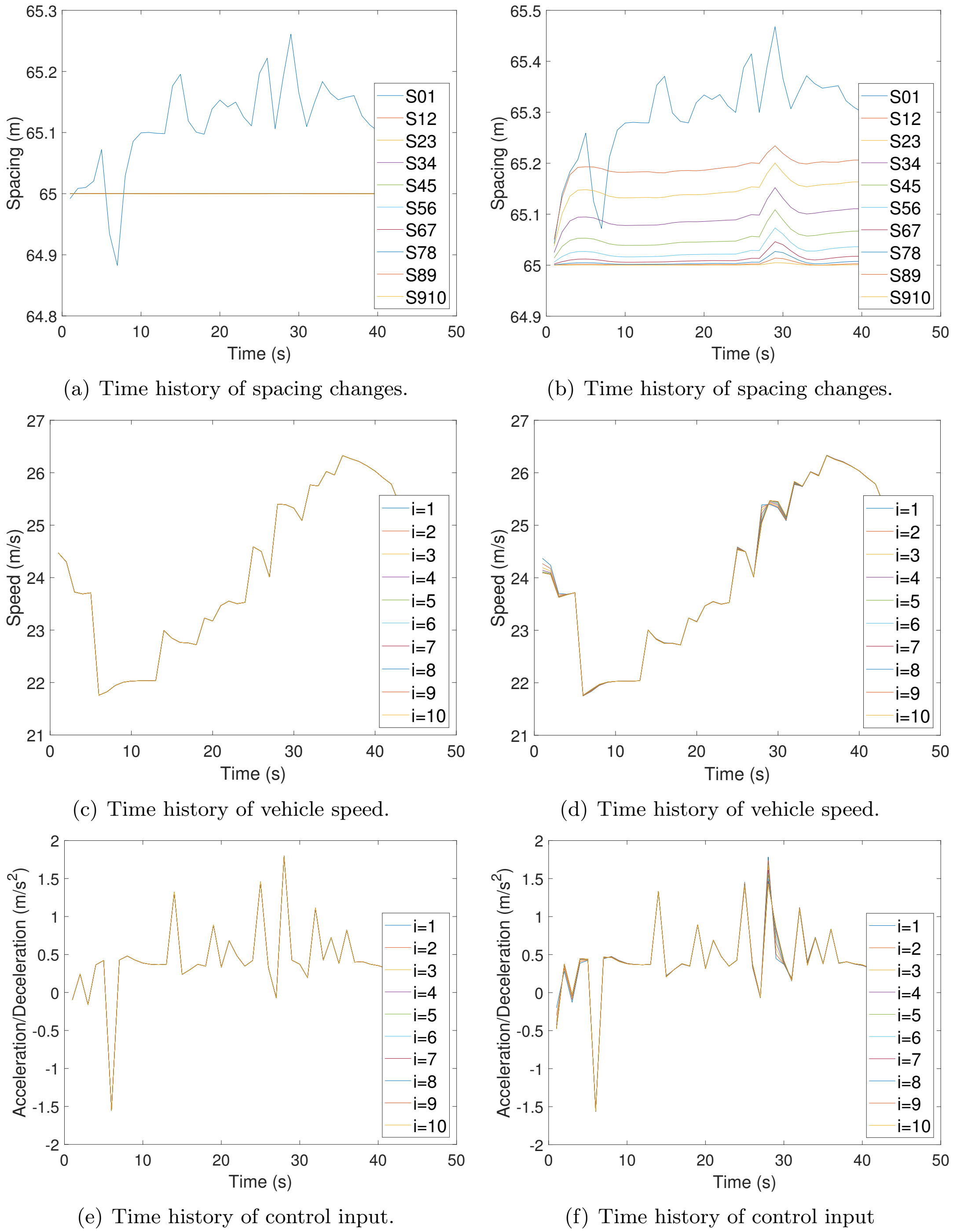


(e) Time history of control input.



(f) Time history of control input

Figure 4.8: Scenario 3 for the heterogeneous medium-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column).



(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

Figure 4.9: Scenario 3 for the homogeneous large-size CAV platoon: platooning control with $p = 1$ (left column) and $p = 5$ (right column).

CHAPTER V

Conclusions

In this thesis, we have studied two topics in fully distributed optimization algorithms i.e., (i) column partition based distributed algorithm for coupled convex sparse optimization problems, and (ii) fully distributed optimization based CAV platooning control under linear and nonlinear vehicle dynamics. In this chapter, we summarize the results that we have established in these two areas, and discuss several future research directions.

5.1 Column Partition based Distributed Algorithm for Coupled Convex Sparse Optimization Problems

In Chapter II, using duality theory, exact regularization techniques and solution properties we developed a two-stage column partition based fully distributed schemes to solve a class of densely coupled convex sparse optimization problems, including BP, LASSO, BPDN and their extensions. These schemes are dual based and are applicable when the underlying matrix is column partitioned. The overall convergence of the two-stage distributed schemes is established in Section 2.7. The numerical results given in Section 2.8 indicate that the proposed two-stage distributed schemes are effective when compared to the existing C-ADMM and PDC-ADMM schemes to solve standard lasso and polyhedral constrained lasso respectively. The proposed schemes and techniques shed light on the development of column partition based distributed schemes for a broader class of densely coupled problems, which will be future research topics. The distributed algorithm development in this thesis relies on first-order techniques, a further investigation on the usage of second-order Newton like distributed schemes will be a future research area.

5.2 Fully Distributed Optimization based CAV Platooning Control

In Chapter III, we developed 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. Major developments in this chapter include a new formulation of the MPC model, a decomposition method for a strongly convex quadratic objective function, formulation the underlying optimization problem as locally coupled optimization and Douglas-Rachford method based distributed schemes to implement them in real time. Control design and stability analysis of the closed loop dynamics is carried out for the new formulation of the MPC model. Further in Chapter IV, we extend these fully distributed CAV platooning control to nonlinear dynamics scenario where the underlying optimization problem in nonconvex. SCP based schemes are developed to solve these nonconvex problems. Extensive numerical tests are conducted to illustrate the effectiveness of the proposed fully distributed schemes and CAV platooning control for homogeneous and heterogeneous CAV platoons in different scenarios. The development of fully distributed algorithms for nonconvex minimization problem, shed light on large scale isoperimetric graph partitioning problem [14] which will be a future research topic. Further, Newton type distributed schemes do not consider constraints other than the consensus constraint. A key challenge for developing such distributed schemes for the transportation application is to effectively handle locally coupled constraints which will be a future research area.

Bibliography

[1] F. Alizadeh and D. Goldfarb, "Second-order cone programming," *Mathematical Programming*, vol. 95, no. 1, pp. 3–51, 2003.

[2] P. Barooah, P. G. Mehta, and J. P. Hespanha, "Mistuning-based control design to improve closed-loop stability margin of vehicular platoons," *IEEE Transactions on Automatic Control*, vol. 54, no. 9, pp. 2100–2113, 2009.

[3] H. H. Bauschke, P. L. Combettes, *et al.*, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011, vol. 408.

[4] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, "Overview of platooning systems," in *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)*, 2012.

[5] D. P. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 2015.

[6] D. P. Bertsekas, *Nonlinear Programming*, 2nd. Athena Scientific, Belmont, MA, 1999.

[7] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*. Springer Science & Business Media, 2000.

[8] T.-H. Chang, "A proximal dual consensus ADMM method for multi-agent constrained optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3719–3734, 2016.

[9] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang, "Asynchronous distributed ADMM for large-scale optimization-Part I: Algorithm and convergence analysis," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3118–3130, 2016.

[10] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2014.

[11] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, 2014.

[12] P. L. Combettes, "Solving monotone inclusions via compositions of nonexpansive averaged operators," *Optimization*, vol. 53, no. 5-6, pp. 475–504, 2004.

[13] R. W. Cottle, J.-S. Pang, and R. E. Stone, *The Linear Complementarity Problem*. Academic Press Inc., Cambridge, 1992.

[14] S. Danda, A. Challa, B. D. Sagar, and L. Najman, "Revisiting the isoperimetric graph partitioning problem," *IEEE Access*, vol. 7, pp. 50 636–50 649, 2019.

[15] D. Davis and W. Yin, "A three-operator splitting scheme and its optimization applications," *Set-valued and Variational Analysis*, vol. 25, no. 4, pp. 829–858, 2017.

[16] J. Eckstein and D. P. Bertsekas, "On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

[17] F. Facchinei and J.-S. Pang, *Finite Dimensional Variational Inequalities and Complementarity Problems*. Springer-Verlag, 2003.

[18] A. Forestiero and G. Papuzzo, "Distributed algorithm for big data analytics in healthcare," in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, IEEE Computer Society, 2018, pp. 776–779.

[19] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhäuser, Basel, 2013.

[20] M. P. Friedlander and P. Tseng, "Exact regularization of convex programs," *SIAM Journal on Optimization*, vol. 18, no. 4, pp. 1326–1350, 2007.

[21] S. Gong and L. Du, "Cooperative platoon control for a mixed traffic flow including human drive vehicles and connected and autonomous vehicles," *Transportation Research Part B: Methodological*, vol. 116, pp. 25–61, 2018.

[22] S. Gong, J. Shen, and L. Du, "Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon," *Transportation Research Part B: Methodological*, vol. 94, pp. 314–334, 2016.

[23] M. S. Gowda and R. Sznajder, "On the Lipschitzian properties of polyhedral multifunctions," *Mathematical Programming*, vol. 74, no. 3, pp. 267–278, 1996.

[24] M. Hong, T.-H. Chang, X. Wang, M. Razaviyayn, S. Ma, and Z.-Q. Luo, "A block successive upper-bound minimization method of multipliers for linearly constrained convex optimization," *Mathematics of Operations Research*, vol. 45, no. 3, pp. 833–861, 2020.

[25] J. Hu, J. Shen, and V. Putta, "Generalized input-to-state ℓ_2-gains of discrete-time switched linear control systems," *SIAM Journal on Control and Optimization*, vol. 54, no. 3, pp. 1475–1503, 2016.

[26] J. Hu, Y. Xiao, and J. Liu, "Distributed algorithms for solving locally coupled optimization problems on agent networks," in *2018 IEEE Conference on Decision and Control*, IEEE, 2018, pp. 2420–2425.

[27] S. Ichiraku, "A note on global implicit function theorems," *IEEE Transactions on Circuits and Systems*, vol. 32, no. 5, pp. 503–505, 1985.

[28] A. Jadbabaie, A. Ozdaglar, and M. Zargham, "A distributed Newton method for network optimization," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, IEEE, 2009, pp. 2736–2741.

[29] D. Jakovetić, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, 2014.

[30] Z. P. Jiang and Y. Wang, "A converse Lyapunov theorem for discrete-time systems with disturbances," *Systems & Control Letters*, vol. 45, no. 1, pp. 49–58, 2002.

[31] Z.-P. Jiang, Y. Lin, and Y. Wang, "Nonlinear small-gain theorems for discrete-time feedback systems and applications," *Automatica*, vol. 40, no. 12, pp. 2129–2136, 2004.

[32] Z.-P. Jiang and Y. Wang, "Input-to-state stability for discrete-time nonlinear systems," *Automatica*, vol. 37, no. 6, pp. 857–869, 2001.

[33] P. Kavathekar and Y. Chen, "Vehicle platooning: A brief survey and categorization," in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 2011, pp. 829–845.

[34] A. Kesting, M. Treiber, M. Schönhof, and D. Helbing, "Adaptive cruise control design for active congestion avoidance," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 6, pp. 668–683, 2008.

[35] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 1996.

[36] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "ℓ_1 trend filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, 2009.

[37] J. Koshal, A. Nedic, and U. V. Shanbhag, "Multiuser optimization: Distributed algorithms and error analysis," *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 1046–1081, 2011.

[38] M.-J. Lai and W. Yin, "Augmented ℓ_1 and nuclear-norm models with a globally linearly convergent algorithm," *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, pp. 1059–1091, 2013.

[39] C. Lenzen and R. Wattenhofer, "Distributed algorithms for sensor networks," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 370, no. 1958, pp. 11–26, 2012.

[40] S. Li, K. Li, R. Rajamani, and J. Wang, "Model predictive multi-objective vehicular adaptive cruise control," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 556–566, 2011.

[41] Z. Li, Z. Ding, J. Sun, and Z. Li, "Distributed adaptive convex optimization on directed graphs via continuous-time algorithms," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1434–1441, 2017.

[42] Z. Li, W. Shi, and M. Yan, "A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates," *IEEE Transactions on Signal Processing*, vol. 67, no. 17, pp. 4494–4506, 2019.

[43] P.-L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.

[44] J. Liu and S. J. Wright, "Asynchronous stochastic coordinate descent: Parallelism and convergence properties," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 351–376, 2015.

[45] Z. Lu, "Sequential convex programming methods for a class of structured nonlinear programming," Simon Fraser University, Tech. Rep., 2013.

[46] G. Marsden, M. McDonald, and M. Brackstone, "Towards an understanding of adaptive cruise control," *Transportation Research Part C: Emerging Technologies*, vol. 9, no. 1, pp. 33–51, 2001.

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

[48] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[49] A. Mokhtari, Q. Ling, and A. Ribeiro, “Network newton distributed optimization methods,” *IEEE Transactions on Signal Processing*, vol. 65, no. 1, pp. 146–161, 2016.

[50] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel, “Distributed basis pursuit,” *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 1942–1956, 2012.

[51] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, “D-ADMM: A communication-efficient distributed algorithm for separable optimization,” *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.

[52] S. Mousavi and J. Shen, “Solution uniqueness of convex piecewise affine functions based optimization with applications to constrained ℓ_1 minimization,” *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 25, p. 56, 2019.

[53] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[54] A. Nedic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[55] A. Nedić and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[56] D. Newman, A. Asuncion, P. Smyth, and M. Welling, “Distributed algorithms for topic models.,” *Journal of Machine Learning Research*, vol. 10, no. 8, 2009.

[57] I. Notarnicola and G. Notarstefano, “Constraint-coupled distributed optimization: A relaxation and duality approach,” *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 483–492, 2019.

[58] G. B. Passty, “Ergodic convergence to a zero of the sum of monotone operators in hilbert space,” *Journal of Mathematical Analysis and Applications*, vol. 72, no. 2, pp. 383–390, 1979.

[59] Z. Peng, M. Yan, and W. Yin, “Parallel and Distributed Sparse Optimization,” in *2013 Asilomar Conference on Signals, Systems and Computers*, IEEE, 2013, pp. 659–646.

[60] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[61] A. Ruszczynski, *Nonlinear Optimization*. Princeton University Press, 2006.

[62] I. Sandberg, “Global implicit function theorems,” *IEEE Transactions on Circuits and Systems*, vol. 28, no. 2, pp. 145–149, 1981.

[63] J. Shen, E. K. H. Kammara, and L. Du, “Nonconvex, fully distributed optimization based CAV platooning control under nonlinear vehicle dynamics,” *arXiv preprint arXiv:2104.08713*, 2021.

[64] J. Shen, E. K. H. Kammara., and L. Du, "Fully distributed optimization based CAV platooning control under linear vehicle dynamics," *Transportation Science*, in print, arXiv:2103.11081, 2021.

[65] J. Shen and J. Hu, "Stability of discrete-time switched homogeneous systems on cones and conewise homogeneous inclusions," *SIAM Journal on Control and Optimization*, vol. 50, no. 4, pp. 2216–2253, 2012.

[66] J. Shen, J. Hu, and E. K. H. Kammara, "Column partition based distributed algorithms for coupled convex sparse optimization: Dual and exact regularization approaches," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 7, pp. 375–391, 2021.

[67] J. Shen and S. Mousavi, "Exact support and vector recovery of constrained sparse vectors via constrained matching pursuit," *arXiv preprint arXiv:1903.07236*, 2019.

[68] J. Shen and S. Mousavi, "Least sparsity of p-norm based optimization problems with $p > 1$," *SIAM Journal on Optimization*, vol. 28, no. 3, pp. 2721–2751, 2018.

[69] J. Shen and J.-S. Pang, "Linear complementarity systems with singleton properties: Non-Zenoness," in *2007 American Control Conference*, IEEE, 2007, pp. 2769–2774.

[70] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.

[71] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative adaptive cruise control: Definitions and operating concepts," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2489, pp. 145–152, 2015.

[72] S. E. Shladover, D. Su, and X.-Y. Lu, "Impacts of cooperative adaptive cruise control on freeway traffic flow," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2324, pp. 63–70, 2012.

[73] M. Sion, "On general minimax theorems," *Pacific Journal of Mathematics*, vol. 8, no. 1, pp. 171–176, 1958.

[74] E. D. Sontag, "Input-to-state stability: Basic concepts and results," in *Nonlinear and Optimal Control Theory*, Springer, 2008, pp. 163–220.

[75] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[76] H. Terelius, U. Topcu, and R. M. Murray, "Decentralized multi-agent optimization via dual decomposition," *IFAC proceedings volumes*, vol. 44, no. 1, pp. 11 245–11 251, 2011.

[77] V. D. Thoke and V. Sangli, "Theory of distributed computing and parallel processing with its applications, advantages and disadvantages.," *International Journal of Innovation in Engineering, Research and Technology*, 2014.

[78] P. Tseng, "A modified forward-backward splitting method for maximal monotone mappings," *SIAM Journal on Control and Optimization*, vol. 38, no. 2, pp. 431–446, 2000.

[79] J. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[80] J. N. Tsitsiklis, "Problems in decentralized decision making and computation.," Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, Tech. Rep., 1984.

[81] B. Van Arem, C. J. Van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7(4), pp. 429–436, 2006.

[82] B. Van Scoy and L. Lessard, "A distributed optimization algorithm over time-varying graphs with efficient gradient evaluations," *IFAC-PapersOnLine*, vol. 52, no. 20, pp. 357–362, 2019.

[83] J. Vander Werf, S. E. Shladover, M. A. Miller, and N. Kourjanskaia, "Effects of adaptive cruise control systems on highway traffic flow capacity," *Transportation Research Record*, vol. 1800, no. 1, pp. 78–84, 2002.

[84] J. Wang, S. Gong, S. Peeta, and L. Lu, "A real-time deployable model predictive control-based cooperative platooning approach for connected and autonomous vehicles," *Transportation Research Part B: Methodological*, vol. 128, pp. 271–301, 2019.

[85] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem, "Cooperative car-following control: Distributed algorithm and impact on moving jam features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1459–1471, 2016.

[86] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem, "Rolling horizon control framework for driver assistance systems. Part II: Cooperative sensing and cooperative control," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 290–311, 2014.

[87] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed Newton method for network utility maximization–i: Algorithm," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2162–2175, 2013.

[88] S. X. Wu, H.-T. Wai, L. Li, and A. Scaglione, "A review of distributed algorithms for principal component analysis," *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1321–1340, 2018.

[89] X. Wu, J. Zhang, and F.-Y. Wang, "Stability-based generalization analysis of distributed learning algorithms for big data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 801–812, 2020.

[90] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[91] W. Yin, "Analysis and generalizations of the linearized Bregman method," *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 856–877, 2010.

[92] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[93] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[94] H. Zhang, W. Yin, and L. Cheng, "Necessary and sufficient conditions of solution uniqueness in 1-norm minimization," *Journal of Optimization Theory and Applications*, vol. 164, no. 1, pp. 109–122, 2015.

[95] M. Zhao and Y. Yang, “Optimization-based distributed algorithms for mobile data gathering in wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1464–1477, 2011.

[96] S. Zhao and K. Zhang, “A distributionally robust stochastic optimization-based model predictive control with distributionally robust chance constraints for cooperative adaptive cruise control under uncertain traffic conditions,” *Transportation Research Part B: Methodological*, vol. 138, pp. 144–178, 2020.

[97] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, “Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies,” *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2017.

[98] Y. Zhou, S. Ahn, M. Chitturi, and D. A. Noyce, “Rolling horizon stochastic optimal control strategy for ACC and CACC under uncertainty,” *Transportation Research Part C: Emerging Technologies*, vol. 83, pp. 61–76, 2017.

[99] Y. Zhou, M. Wang, and S. Ahn, “Distributed model predictive control approach for cooperative car-following with guaranteed local and string stability,” *Transportation Research part B: Methodological*, vol. 128, pp. 69–86, 2019.