DISTRIBUTED MODEL PREDICTIVE CONTROL OF CONSTRAINED LINEAR SYSTEMS

ZHEMING WANG

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

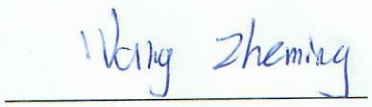
DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE

2016

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



Zheming Wang

August 2016

Acknowledgments

First and foremost I would like to express my sincere gratitude to my supervisor Prof. Ong Chong-Jin for his guidance throughout the course of this thesis. I appreciate all his contributions of time and ideas. His enthusiasm for research is very motivational for me. I benefit a lot from his insightful comments, critiques, and immense knowledge. I thank him for his continuous advice and encouragement.

I gratefully acknowledge the financial support from National University of Singapore through the Research Scholarship, which make it possible for me to complete the Ph.D. study.

Thanks also go to my friends and technicians in the Mechatronics and Control Lab. The lab has been a source of friendships as well as good advice and collaboration.

Lastly, I would like to thank my family for all their love and support. I also want to take this opportunity to thank my girlfriend, Jossie Su, for the joy she brings me.

Abstract

This thesis studies Distributed Model Predictive Control (DMPC) of a group of discrete-time linear systems with and without coupled dynamics or constraints. Several problems are studied: systems with coupled dynamics and systems with independent dynamics but coupled constraints. Under these situations, centralized MPC may be computationally inefficient and more efficient approaches are desired. For a network of dynamically-coupled linear systems, a decoupling strategy for DMPC is proposed with the use of time-varying terminal sets. The terminal sets are determined online at every time step according to some update law, which ensures recursive feasibility and stability of the approach. Time-varying terminal sets also result in a less conservative DMPC formulation compared with most DMPC approaches. For a network of linear systems with coupled constraints, this thesis proposes a DMPC approach based on the dual problem of the overall MPC optimization problem. This dual problem is then solved by the Alternating Direction Multiplier Method (ADMM) in a distributed manner. A stopping criterion is proposed that allows early termination of the ADMM process to reduce the computational burden within each sampling time. Under mild assumptions, the approach is guaranteed to converge to a small neighborhood of the optimal of the overall MPC optimization problem. The advantage of this approach is that it allows the network to be sparsely connected. To further accelerate the online computations of the DMPC problem, a distributed fast dual gradient algorithm is also proposed in this thesis with the use of finite-time consensus. The accelerated approach takes fewer iterations to terminate and thus has a faster convergence. Recursive feasibility and exponential stability of the closed-loop system are also ensured. Simulation results are provided to show the performances of the proposed DMPC approaches.

List of Publications

Journal Papers

C.J. Ong, **Z. Wang**, and M. Dehghan. Model predictive control for switching system with dwell-time restriction. IEEE Transactions on Automatic Control, accepted.

Z. Wang, C.J Ong. Distributed MPC of constrained linear systems with time-varying terminal sets. Systems & Control Letters, 88: 14-23, 2016.

C.J. Ong, **Z. Wang**. Reducing variables in Model Predictive Control of linear system with disturbances using singular value decomposition. Systems & Control Letters, 71: 62-68, 2014.

Conference Papers

Z. Wang, C.J Ong. Distributed model predictive control of linear discrete-time systems with coupled constraints. In Proceedings of 55th IEEE Conference on Decision and Control, 2016, accepted.

C.J. Ong, **Z. Wang**, and M. Dehghan. Characterization of switching sequences on system with dwell-time restriction for Model Predictive Control. In Proceedings of 54th IEEE Conference on Decision and Control, Osaka, Japan, 2015.

Z. Wang, C.J Ong. Distributed MPC of constrained linear systems with online decoupling of the terminal constraint. In Proceedings of the American Control Conference, Chicago, Illinois, 2015.

Contents

List of Figures

List of Tables

List of Algorithms

List of Notations

$\mathbb{Z}_0^+$	Set of non-negative integer
$\mathbb{Z}^+$	Set of positive integer set
$\mathbb{R}_0^+$	Set of non-negative real number
$\mathbb{R}^+$	Set of positive real number
$\mathbb{R}^n$	n-dimensional real Euclidean space
$\mathbb{R}^{n\times m}$	Set of $n \times m$ real matrix
I_n	$n \times n$ identity matrix
$\mathbf{1}_n$	n-dimensional column vector of all ones
$int(\cdot)$	Interior of a set
$Q \succ 0$	Positive definite matrix
$Q \succeq 0$	Positive semi-definite matrix
$\|\cdot\|_p$	p-norm of vector
$\|S\|$	Cardinality of a set S
$\mathbb{Z}_L^M$	$\{L, L+1, \cdots, M\}$
O_∞	The maximal constraint-admissible invariant set
$P_S[x]$	Projection of a vector x onto the set S

CHAPTER 1

Introduction

This thesis concerns the distributed control of constrained networked systems under the model predictive control (MPC) framework. It focuses on the design of Distributed Model Predictive Control(DMPC) controller for linear systems with and without coupled dynamics or constraints. This chapter provides an overview of DMPC.

1.1 Background

Control theory conventionally builds mathematical models of dynamical systems and designs controllers in a centralized fashion. Many engineering systems such as power, water distribution, traffic and manufacturing systems consist of a group of interacting subsystems that may transmit information among one another. The mathematical models of these systems are huge and complex and centralized control design may be difficult. This motivates the study of distributed control, where the controller elements are distributed throughout the system and connected using communication and monitoring networks. The advantages of distributed control are that each control element requires only local information transmission and the overall control problem is converted into several smaller problems that can be solved efficiently. The main issue is that the design of the local controllers has to guarantee the stability and performance of the overall system.

Distributed control of networked systems is a popular recent research topic. In networked systems, the internal structure of the network can impose constraints on the design of controllers arising from communication limitations and time-delays. The early literature has been devoted to the study of decentralized control strategies on large-scale systems, where each agent has only access to its

own local measurements and their controllers operate in an independent fashion, see [1–4] and the references therein. However, these strategies are reliable only for systems with weak local interaction. When strong local interaction appears, systems controlled in a independent fashion could have poor performance and instability. For this reason, it is important to take into account the information exchanges and the synchronization protocols in the control of networked systems. One well-known control strategy for networked systems is the DMPC, which is based on the standard MPC framework.

MPC is a powerful control technology for the control of constrained systems. It determines the current control by solving an online optimal control problem. In large-scale systems, the centralized MPC can lead to a huge optimal control problem in high dimension and often seen as unrealistic and inefficient. For these reasons, a DMPC framework is more desirable in dealing with large-scale or networked systems. In DMPC, the overall optimization problem is decoupled and each local system solves a small local optimization problem with communications among the systems.

Several issues should be discussed and addressed in the DMPC, including the computational efficiency of the DMPC problem and the performance of the whole system. The overall DMPC problem is a high-dimension problem that requires a huge amount of memory. A distributed implementation is therefore preferred for the DMPC problem. However, the coupled dynamics and constraints of systems can cause performance deterioration in a distributed implementation of DMPC compared to a centralized MPC, including feasibility and stability issues. The next section reviews some related past works on the control of networked systems under MPC framework.

1.2 Related Works

As discussed in the previous section, distributed control is an important strategy for large-scale or networked systems. It aims to convert a large control problem into a bunch of smaller problems where the control inputs can be computed very efficiently. This section begins with the review of decentralized control and distributed control. It is then followed by an overview of existing DMPC approaches.

1.2.1 Decentralized Control and Distributed Control

The research on the control of large-scale or networked systems has a long history. Due to the high dimensionality and complexity, it is difficult to control such systems with a centralized control scheme. To illustrate the difficulty, an example is shown in Figure 1.1, where the inputs of the two coupled subsystems are determined by a central controller. The controller requires information from both subsystems to produce the control action.

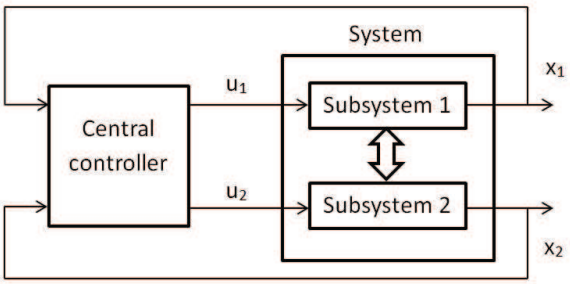


Figure 1.1: Centralized control architecture

For large-scale systems with special structure, it is possible to implement decentralized control, where the subsystems are decoupled and the controllers are designed in an independent fashion [1–4]. Figure 1.2 shows an example of decentralized control, where each subsystem is controlled by a local controller and there is no interaction between the two local controllers. Various decentral-

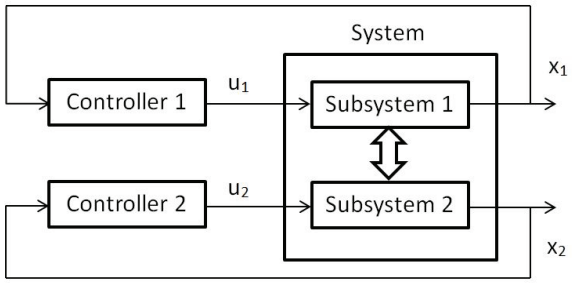


Figure 1.2: Decentralized control architecture

ized control methods have been developed to achieve the stability of the overall system with some performance assurance. Among these methods, the classic Lyapunov-based control methods are widely used [1]. Other methods include the sequential design [5], parameter optimization [6, 7], and overlapping decompositions [8–11]. Recent research on decentralized control can be found in

the survey papers [3, 12–15]. The basis of decentralized control is to break down a given central control problem into manageable subproblems which are only weakly related to each other and can be solved independently. To this end, the overall feedback gain matrix is usually restricted to be block-diagonal without considering the weak coupling. In some special structures, it is possible to make small adjustment, e.g. the overlapping and the bordered block-diagonal (BBD) control laws in [4, 16]. A more general decentralized control design can be found in [17]. There are also approaches available in [18–21] that impose the structural constraints arising from the structure of the overall system. Many of these approaches have used Linear Matrix Inequalities(LMIs) to develop sufficient conditions for stability. The decentralized control design works well for weakly coupled networked systems. However, in systems where the connections among subsystems are strong, the overall performance can be affected if the couplings are ignored. Due to the strong local connections, systems controlled in a decentralized fashion could have poor performance and even stability issues. This means that decentralized control is not always suitable for networked systems. In order to circumvent the drawbacks of decentralized control, distributed control is widely used in networked systems where local subsystems are strongly related. Unlike decentralized control, it makes use of the internal structure of the overall system with information transmission among subsystems. Figure 1.3 presents the general distributed control architecture, where the local controllers of the subsystems transmit information with each other. In the demand for good overall performance,

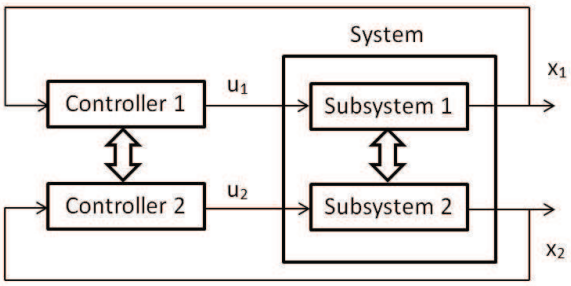


Figure 1.3: Distributed control architecture

distributed control becomes a useful strategy for networked systems. However, it is difficult to impose structural constraints in general systems directly as this will lead to computationally intractable problems [22–24]. In general, the controller and the structural constraints are designed in a linear fractional transformation (LFT) framework by solving a norm-minimizing problem subject to information constraints. Such constraints lead to nonconvex optimization problems which, in most of the

case, are intractable. In [25, 26], a sufficient condition on the structure, called quadratic invariance, is introduced to produce a tractable optimization problem. This condition is extended in [27] by the use of internal quadratic invariance, which is claimed to be less conservative. Although there are a great number of works on distributed control with structural constraints, this is still an open research problem.

1.2.2 Overview of Distributed Model Predictive Control

MPC is typically a centralized optimal control strategy, in which the current control action is obtained by solving an online finite horizon optimization control problem defined by the predicted trajectory of the plant at each sampling time. The optimization problem yields an optimal control sequence and the first control in this sequence is applied to the plant. It is widely used in constrained systems with extensive applications to various industries areas. The essential theoretical issues in MPC include the existence of solutions to optimization control problems, characterization of optimal solution, Lyapunov stability of the optimally controlled system, and algorithms for the computation of optimal feedback control and optimal control sequences, see [28–31]. To achieve recursive feasibility and stability, it is quite common to incorporate both a terminal cost and a terminal set in the optimization control problem [32–34]. A detailed review on the technical properties of MPC is provided in Section 2.2. However, in large-scale or networked systems, it is unrealistic and inefficient to solve the high-dimensional overall MPC problem online. To circumvent this problem, DMPC is proposed. Its aim is to achieve the attractive features of MPC in a distributed implementation. An overview of past DMPC approaches is provided below.

There is a wide set of DMPC approaches proposed in the literature recently [15, 35–41]. These approaches can be considered as either fully connected DMPC or partially connected DMPC depending on the topology of the communication network. Those that allow local controllers to have knowledge of the state of the whole network are called fully connected DMPC [42]. However, fully connected systems are rare in real situations. Most systems only have partial information and exchange information with their neighbours. Approaches based on this assumption are called partially connected DMPC. Fully connected DMPC usually has better performance but requires heavy communication while partially connected has some performance deterioration and requires

less communication. Of course, the former can be considered as a special case of the latter. In networked systems with weakly coupled subsystems, partially connected DMPC can be effective with low communication requirement and negligible performance deterioration.

DMPC can also be categorized according to different choices of the performance index. Some approaches are called the non-cooperative DMPC while the others are called cooperative DMPC. In the non-cooperative DMPC, each local controller only optimizes its own local performance index. On the contrary, the cooperative DMPC focuses on the global performance of the entire system rather than the local performance of each system. When the subsystems are completely uncoupled, these two strategies can have the same performance. In general, their performances are significantly different. The non-cooperative DMPC is often used in systems where subsystems are dynamically decoupled [35, 43–46]. It can also be used for dynamically coupled systems where the coupled terms are treated as disturbances to the dynamics of the local systems [47–49] or replaced by some presumed fixed trajectories [50]. The cooperative DMPC is frequently used in systems where subsystems are coupled in some forms. This cooperative strategy was first introduced by [51] and followed by [52–54]. Generally speaking, the cooperative DMPC has better performance due to the cooperation among subsystems. In the context of game theory [55], the non-cooperative DMPC allows the subsystems to move towards Nash equilibrium [40, 56, 57], while the cooperative DMPC nonetheless seeks to achieve the Pareto optimal solution [40, 58].

In terms of information exchange, there are two different types of transmission protocols, namely non-iterative DMPC and iterative DMPC. In the non-iterative DMPC, the information is transmitted only once and each subsystem solves its MPC problem once within each sampling time [37, 50, 59, 60]. Meanwhile, in the iterative DMPC, the information can be transmitted many times and each subsystem need to solve its MPC problem repeatedly within each sampling time [51, 61–64]. Usually, the iterative DMPC requires more computations and communications and its control is closer to the optimal of the centralized MPC, and hence has better performance. Additionally, some proper stopping criterion is needed in the iterative DMPC. The past works use either the maximal number of iterations [51, 61, 62] or a centralized stopping criterion [64] to ensure that the solution is within some accuracy. However, the maximal number of iterations is a loose condition and the centralized stopping criterion is not desirable in a distributed implementation.

In almost all DMPC approaches, recursive feasibility and stability of the overall system should be guaranteed. Like the centralized MPC, the terminal set of the whole networked system can be used to ensure these properties. However, due to the coupled dynamics and constraints, it is difficult to determine the terminal set. The most convenient choice of the terminal set is the origin of the individual subsystem [51, 65]. Less conservative approaches [66, 67] use an ellipsoidal set induced from a block diagonal Lyapunov matrix. While being a great improvement, restricting Lyapunov matrix to be block-diagonal is still restrictive. To the best of the author's knowledge, further research is needed for computation of the terminal set of DMPC without significant performance deterioration.

1.3 Motivation

Based on the review in the previous section, DMPC is an important control strategy for constrained networked systems and its study has not been completely explored. The following table presents a summary of the existing DMPC approaches according to different classification attributes.

Classification attributes	DMPC approaches
Topology of the communication network	fully connected
	partially connected
Performance index	non-cooperative
	cooperative
Protocol of information exchange	non-iterative
	iterative

Table 1.1: Summary of existing DMPC approaches

In the presence of coupled dynamics and constraints, it it desirable to use cooperative and iterative DMPC to avoid significant performance deterioration. This thesis considers both the fully connected and partially connected networks. The focus is placed on the partially connected network as it has lower communication cost and is more realistic for large-scale networked systems. In order to facilitate the distributed implementation, the terminal sets should be determined in a distributed manner. For iterative algorithms, some proper stopping criterion is needed to terminate the algorithm for solving the online MPC problem. A convenient choice of the stopping criterion is the maximal number of iterations. However, this stopping criterion is conservative in most of the cases. This thesis proposes some tight stopping criterions under different MPC formulations and they are done in a distributed manner. The close-loop behaviour of the DMPC will also be investigated and the

comparison will be made with the centralized MPC.

1.4 Scope and Organization of the Thesis

The thesis focuses on discrete time-invariant linear systems with coupled dynamics and linear coupled constraints. Nonlinear systems and nonconvex constraints are beyond the scope of this thesis. The rest of this thesis is organized as follows:

Chapter 2

This chapter gives a review of some theoretical results, including basic mathematical concepts, properties of the standard MPC, ADMM and some closely related existing DMPC approaches.

Chapter 3

This chapter discusses developments in the area of distributed consensus algorithms. A review of consensus coordination problems and multi-agent consensus optimization is provided. A finite-time consensus algorithm is also proposed in this chapter.

Chapter 4

This chapter proposes a decoupling strategy for DMPC for a network of dynamically-coupled linear systems with uncoupled constraints. The proposed approach uses time-varying terminal sets to ensure feasibility and stability of the overall system. Distributed implementations of this approach are proposed under two different cases: the network is fully connected (or when a central collector is used); the network is connected.

Chapter 5

This chapter proposes a DMPC approach for a family of discrete-time linear systems with local (uncoupled) and global (coupled) constraints. The proposed approach is based on the dual problem of a MPC optimization problem involving all systems, which is then distributedly solved using Alternating Direction Multiplier Method (ADMM). To improve the computational efficiency, this approach also allows early termination of the ADMM process via a finite-time consensus algorithm.

Chapter 6

This chapter proposes an accelerated DMPC approach for the same problem in Chapter 5. Similarly,

this accelerated approach is based on the dual problem of an overall MPC optimization problem involving all systems. However, this dual problem is then solved distributively based on the Nesterov-accelerated-gradient algorithm. Under reasonable assumptions, the approach is able to produce a suboptimal solution and converges faster than the approach in Chapter 5 for the same accuracy.

Chapter 7

This chapter summarizes the main contributions of this thesis and provides possible directions for future work.

CHAPTER 2

Basic Concepts and Literature Review

This chapter reviews some well-known results that are related to the proposed DMPC approaches in this thesis. It begins with some related mathematical background, followed by a review on MPC and some distributed algorithms to solve the MPC prolem. Finally, it also provides a detailed review on existing DMPC approaches.

2.1 Mathematical Background

This section provides some related mathematical background, including basic properties of convex sets and convex functions, and standard definitions in graph theory.

2.1.1 Convex Sets and Convex Functions

Definition 2.1. *A set C is convex if the line segment between any two points in C lies in C, i.e.,*

$$\theta x_1 + (1-\theta)x_2 \in C, \forall x_1, x_2 \in C, \forall \theta \in [0,1] \tag{2.1}$$

Given $x_1, x_2, \cdots, x_k$, a point is called a convex combination of $x_1, x_2, \cdots, x_k$, if it can be represented as $\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k$, where $\theta_1 + \theta_2 + \cdots + \theta_k = 1$ and $\theta_i \geq 0$ for all $i = 1, 2, \cdots, k$. For an arbitrary set S in $\mathbb{R}^n$, the convex hull can be generated from S using the convex combination.

Definition 2.2. *The convex hull of S, denoted* conv *S, is the set of all convex combinations of points*

in C, or,

$$\text{conv } S = \{\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k : x_i \in C, \theta_i \geq 0, \forall i = 1,2,\cdots,k, \theta_1 + \theta_2 + \cdots + \theta_k = 1\} \quad (2.2)$$

The following theorem can be derived from the definition of the convex hull.

Theorem 2.1. *Let S be an arbitrary set in* $\mathbb{R}^n$. *Then,* conv *S is the smallest convex set containing S*

There are some special points in a convex set that are called extreme points, whose definition is given as follows.

Definition 2.3. *Given the convex set* $S \subseteq \mathbb{R}^n$, *a point* $p \in S$ *is called an extreme point of S, if it does not lie in any open line segment joining two points of S.*

In other words, for an extreme point p of a convex set S, $p = \theta x_1 + (1-\theta)x_2$ with $x_1, x_2 \in S$ and $\theta \in (0,1)$ implies that $p = x_1 = x_2$.

One special case of convex sets is represented by polyhedral sets.

Definition 2.4. *A set* $S \subseteq \mathbb{R}^n$ *is a polyhedral set if it is the intersection of a finite number of equalities and inequalities, e.g.* $S = \{x \in \mathbb{R}^n : Ax = b, Cx \leq d\}$ *with* $A \in \mathbb{R}^{m\times n}, C \in \mathbb{R}^{p\times n}, b \in \mathbb{R}^m$ *and* $d \in \mathbb{R}^p$.

It can be easily verified that a polyhedral set is convex. A bounded polyhedral set is often called a polytope. The convex hull of a finite set of points is a bounded polyhedral set or polytope. Given a set of points $\{v_1, v_2, \cdots, v_k\}$, its convex hull can be represented as

$$\text{conv } \{v_1, v_2, \cdots, v_k\} = \{\theta_1 v_1 + \theta_2 v_2 + \cdots + \theta_k v_k : \theta_i \geq 0, i = 1,2,\cdots,k, \sum_{i=1}^{k} \theta_i = 1\} \quad (2.3)$$

Although the convex hull is a polyhedral set, it is generally difficult to convert this convex hull form into the standard expression in Definition 2.4. Both expressions can be used to represent a polyhedral set.

With the properties of convex sets, the properties of convex functions are also given.

Definition 2.5. *Let* $f : S \to \mathbb{R}^n$, *where S is a nonempty convex set in* $\mathbb{R}^n$. *The function* f *is convex (concave) on S if*

$$f(\theta x + (1-\theta)y) \leq (\geq)\ \ \theta f(x) + (1-\theta) f(y), \quad \forall x, y \in S, \forall \theta \in [0,1] \quad (2.4)$$

The function f is strictly convex (concave) on S if

$$f(\theta x+(1-\theta)y) < (>)\ \ \theta f(x)+(1-\theta)f(y), \quad \forall x,y \in S, x \neq y, \forall \theta \in (0,1) \tag{2.5}$$

From the definitions, it can be shown that a function f is concave if and only if $-f$ is convex. For a differentiable function f on *S*, it is convex if and only if

$$f(y) \geq f(x)+\nabla f(x)^T(y-x), \quad \forall x,y \in S \tag{2.6}$$

If f is twice differentiable on *S*, it is convex if and only if

$$\nabla^2 f(x) \succeq 0, \quad \forall x \in S \tag{2.7}$$

A stronger property is called strongly convex. For a differentiable function f on *S*, it is strongly convex if there exists a $\sigma > 0$ such that

$$(\nabla f(y)-\nabla f(x))^T(y-x) \geq \sigma \|y-x\|^2, \quad \forall x,y \in S \tag{2.8}$$

If f is twice differentiable on *S*, it is strongly convex if there exists a $\sigma > 0$ such that

$$\nabla^2 f(x) \succeq \sigma I, \quad \forall x \in S \tag{2.9}$$

However, if the function f is nondifferentiable, the subgradient can be defined.

Definition 2.6. *Consider a convex function $f : S \to \mathbb{R}^n$, where S is a nonempty convex set in $\mathbb{R}^n$. The vector $d \in \mathbb{R}^n$ is a subgradient of f at $x \in S$ if*

$$f(y) \geq f(x)+d^T(y-x), \quad \forall y \in S \tag{2.10}$$

The function f is subdifferentiable on the convex set *S* if it has a subgradient at any $x \in S$.

2.1.2 Graph Theory

Some standard definitions in graph theory are given next. This thesis only considers undirected graphes.

Definition 2.7. *A graph $G = (\mathcal{V}, \mathcal{E})$ is a pair of a set $\mathcal{V}$ of vertices and a set $\mathcal{E}$ of edges. Every element of $\mathcal{E}$ is a pair of vertices in $\mathcal{V}$, e.g. $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. For any two vertices, $v_1, v_2 \in \mathcal{V}$, they are adjacent if $(v_1, v_2) \in \mathcal{E}$.*

Definition 2.8. *A walk in the graph $G = (\mathcal{V}, \mathcal{E})$ is a sequence of vertices $v_1, v_2, \cdots, v_k$ such that $(v_i, v_{i+1}) \in \mathcal{E}$ for all $i \in \mathbb{Z}^{k-1}$. A path is a walk that visits each vertex at most once, e.g. $v_i \neq v_j$ for any $i \neq j$. A closed walk is a walk where $v_1 = v_k$. A circle is a closed path where $(v_k, v_1) \in \mathcal{E}$.*

Definition 2.9. *The graph $G = (\mathcal{V}, \mathcal{E})$ is connected if there exists a path for any two vertices in the set $\mathcal{V}$.*

Definition 2.10. *A tree is a connected graph with no cycles.*

Definition 2.11. *The graph $G = (\mathcal{V}, \mathcal{E})$ is fully or completely connected if every pair of vertices are adjacent, e.g. $\mathcal{E} = \mathcal{V} \times \mathcal{V}$.*

Definition 2.12. *A node is called the central or master node in a graph if it is adjacent to all the other vertices.*

2.2 Model Predictive Control

Mode predictive control (MPC) or receding horizon control (RHC) is an advanced control technology for constrained systems, with extensive applications to various industries. A large number of works [28–31] have been devoted on the theoretical properties of MPC, such as stability, closed-loop performance, and robustness. The applications of MPC can be found in [68].

Consider the discrete-time linear system

$$x(t+1) = Ax(t) + Bu(t),\ t \in \mathbb{Z}_0^+ \tag{2.11}$$

$$x(t) \in X,\ \ u(t) \in U, \tag{2.12}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and control of the system and X and U are appropriate

constraint sets on x and u respectively. The objective of the control is to steer the state to the origin with the constraint (2.12) being satisfied at every time instant.

2.2.1 The Optimal Control Problem

The general framework of model predictive control(MPC) for (2.11) is to solve an online finite horizon optimization problem (FHOP) having an objective function with constraints on the predicted state and predicted control for each stage of the horizon, together with some appropriate end stage constraint. Let the predicted control sequence be

$$\boldsymbol{u}_t = \{u_{0|t}, u_{1|t}, \cdots, u_{N-1|t}\} \tag{2.13}$$

where N is the horizon length and $u_{i|t}$ is the i^{th} predicted control from the current time instant t. Let $x_{i|t}$ denote the i^{th} predicted state and the associated predicted state sequence is

$$\boldsymbol{x}_t = \{x_{0|t}, x_{1|t}, \cdots, x_{N|t}\} \tag{2.14}$$

The cost function of MPC is defined by

$$V_N(\boldsymbol{x}_t, \boldsymbol{u}_t) = \sum_{i=0}^{N-1} l(x_{i|t}, u_{i|t}) + l_f(x_{N|t}) \tag{2.15}$$

where $l(\cdot,\cdot)$ and $l_f(\cdot)$ are some appropriate stage cost and terminal cost. The optimal predicted control sequence is then obtained by solving the following FHOP

$$\begin{aligned} \mathbb{P}(x(t)): \quad & J_N(x(t)) = \min_{\boldsymbol{u}_t} V_N(\boldsymbol{x}_t, \boldsymbol{u}_t) \\ s.t. \quad & x_{i+1|t} = Ax_{i|t} + Bu_{i|t}, x_{0|t} = x(t), i \in \mathbb{Z}_0^{N-1}, && \text{(2.16a)} \\ & x_{i|t} \in X, u_{i|t} \in U, x_{N|t} \in X_f && \text{(2.16b)} \end{aligned}$$

where X_f is some appropriate terminal set. Let the optimal control sequence be

$$\boldsymbol{u}_t^* = \{u_{0|t}^*, u_{1|t}^*, \cdots, u_{N-1|t}^*\} \tag{2.17}$$

Then, the MPC control law is the first control

$$\kappa(x(t)) = u^*_{0|t} \tag{2.18}$$

After the control law is applied to system (2.11), the new state $x(t+1)$ can be obtained at $t+1$ and $\mathbb{P}(x(t+1))$ is solved again with $x(t+1)$. The closed-loop system becomes

$$x(t+1) = Ax(t) + B\kappa(x(t)) \tag{2.19}$$

Figure 2.1 presents a conceptual picture of the standard MPC scheme.

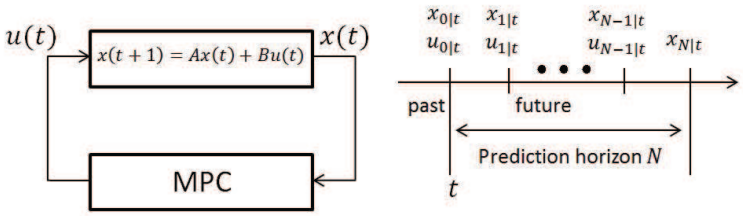


Figure 2.1: A conceptual picture of MPC

2.2.2 The Terminal Cost and Set

The terminal cost $l_f(x)$ plays an important role in the closed-loop stability of (2.19) [29]. In order to ensure the stability, the terminal cost should satisfy

$$l(x, k_f(x)) - l_f(x) + l_f(x^+) \leq 0, \forall x \in X_f \tag{2.20}$$

where $k_f(x)$ is some stabilizing local control law for system (2.11) and $x^+ = Ax + B\kappa_f(x)$. The stage cost can be defined by

$$l(x,u) = \|x\|^2_Q + \|u\|^2_R \tag{2.21}$$

for some appropriate weighting matrices $Q, R \succ 0$. The stabilizing control law and the terminal cost are typically chosen to be

$$k_f(x) = Kx, \quad l_f(x) = \|x\|_P^2 \tag{2.22}$$

where P satisfies

$$(A+BK)^T P(A+BK) - P \preceq -(Q+K^T RK) \tag{2.23}$$

for some stabilizing K of (A,B). In the standard MPC of linear systems, P and K are often obtained from the Algebraic Riccatti equation [69]

$$P = A^T PA - A^T PB(B^T PB+R)^{-1}B^T PA + Q \tag{2.24}$$

$$K = -(B^T PB+R)^{-1}B^T PA, \tag{2.25}$$

Some terminal set X_f is often used to ensure recursive feasibility of MPC. It is usually chosen to be a constraint-admissible invariant set of (2.11) with the control law $\kappa_f(x)$, see the definition in [29]. As shown in [32, 70], for linear systems, the maximal constraint-admissible invariant set exists and it can be easily determined.

2.2.3 Recursive Feasibility and Stability

Recursive feasibility and stability are two important properties of MPC. Recursive feasibility means that the feasibility of $\mathbb{P}(x(t))$ implies the feasibility of $\mathbb{P}(x(t+1))$. The basic idea is to construct a feasible solution at time $t+1$ using the solution at time t. Suppose $\mathbb{P}(x(t))$ is feasible at time t and the optimal control sequence obtained from $\mathbb{P}(x(t))$ is $\boldsymbol{u}_t^* = \{u_{0|t}^*, u_{1|t}^*, \cdots, u_{N-1|t}^*\}$ with the associated state sequence $\boldsymbol{x}_t^* = \{x_{0|t}^*, x_{1|t}^*, \cdots, x_{N|t}^*\}$. Then, by applying the control law (2.18), the new state becomes $x(t+1) = Ax(t) + B\kappa(x(t)) = x_{1|t}^*$. The control sequence $\{u_{1|t}^*, \cdots, u_{N-1|t}^*\}$ will steer $x_{1|t}^*$ to $x_{N|t}^* \in X_f$. Consider that X_f is a constraint-admissible invariant set [29], a feasible control

sequence to $\mathbb{P}(x(t+1))$ can be obtained by

$$\tilde{\boldsymbol{u}}_{t+1} := \{\tilde{u}_{0|t+1}, \tilde{u}_{1|t+1}, \cdots, \tilde{u}_{N-1|t+1}\} := \{u^*_{1|t}, \cdots, u^*_{N-1|t}, u^*_{N|t}\} \tag{2.26}$$

where $u^*_{N|t} = \kappa_f(x^*_{N|t})$. The associated state sequence is

$$\tilde{\boldsymbol{x}}_{t+1} = \{\tilde{x}_{0|t+1}, \tilde{u}_{1|t+1}, \cdots, \tilde{x}_{N|t+1}\} := \{x^*_{1|t}, \cdots, x^*_{N|t}, x^*_{N+1|t}\} \tag{2.27}$$

where $x^*_{N+1|t} = Ax^*_{N|t} + B\kappa_f(x^*_{N|t})$. Hence, $\mathbb{P}(x(t+1))$ is feasible. The stability result can be obtained by considering $J_N(x)$ as a Lyapunov function of the closed-loop system (2.19). It can be found in [29] that $J_N(x)$ is decreasing along any trajectory of (2.19) until the state reaches the origin.

Although MPC is an effective control strategy and enjoys good performance, the online computational burden increases significantly when the size of the problem $\mathbb{P}(x)$ increases. Hence, for large-scale systems, it is necessary to implement distributed computation for the online MPC problem. The rest of this chapter will review some well-known distributed algorithms that are suitable for solving the online MPC problem, including ADMM and gradient methods.

2.3 ADMM

The alternating direction method of multipliers (ADMM) was first proposed in [71–73] and it is suitable for solving structured large-scale optimization problems. Due to its strong convergence properties [74], it is widely used in the areas of image and signal processing [75], machine learning [76] and resource allocation [77]. The wide application of ADMM then motivates extensive theoretical studies [74, 78–83]. This section discusses the standard ADMM formulation, the convergence results and the implementation of ADMM on distributed optimization problems.

2.3.1 Standard ADMM Algorithm

ADMM solves optimization problems in the form of

$$\min_{x,z} \quad f(x)+g(z) \tag{2.28a}$$

$$s.t. \quad Ax+Bz=c \tag{2.28b}$$

$$x\in X, z\in Z \tag{2.28c}$$

where $A\in\mathbb{R}^{p\times n}$, $B\in\mathbb{R}^{p\times m}$, $c\in\mathbb{R}^p$, $X\subset\mathbb{R}^n$ and $Z\subset\mathbb{R}^m$ are given. $f(x)$ and $g(z)$ are assumed to be convex and subdifferentiable. The augmented Lagrangian is

$$L_\rho(x,z,y) = f(x)+g(z)+y^T(Ax+Bz-c)+\frac{1}{2}\rho\|Ax+Bz-c\|_2^2 \tag{2.29}$$

where $y\in\mathbb{R}^p$ is the multiplier of the constraint (2.28b) and $\rho>0$ is some penalty parameter. ADMM consists of the following iterates

$$x^{k+1} = \arg\min_{x\in X} L_\rho(x,z^k,y^k) \tag{2.30a}$$

$$z^{k+1} = \arg\min_{z\in Z} L_\rho(x^{k+1},z,y^k) \tag{2.30b}$$

$$y^{k+1} = y^k+\rho(Ax^{k+1}+Bz^{k+1}-c) \tag{2.30c}$$

Note that a coordinator is needed to collect information from x and z to update the dual variable y. Under some mild assumptions, ADMM converges to an optimal solution [74, 78].

The convergence rate of the ADMM algorithm is also studied in the literature . For general convex objective functions, the best known convergence rate is $O(1/k)$ [79, 80]. Although this convergence rate is not so appealing, ADMM converges to modest accuracy very fast in practice. Note that it may be still slow for ADMM to converge to high accuracy. When the objective function is strongly convex, it is possible to obtain the exponential convergence rate [81, 83].

2.3.2 Distributed Implementation of ADMM

One important advantage of ADMM is that it can be used as an effective tool in parallel and distributed optimization. This section discusses several types of problems that allows distributed implementations of ADMM. The most typical problem is the one with a shared common variable and M separable objectives corresponding to M agents connected over a network

$$\min_{x\in X} \sum_{i=1}^{M} f_i(x) \tag{2.31}$$

where $X \subset \mathbb{R}^n$ is a nonempty closed convex set and $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a convex function representing the local objective function known by agent i. This problem has received a lot of attention in the literature [74, 84–87]. A detailed review on the consensus optimization will be given in Section 3.2. This section discusses a distributed ADMM formulation of this problem. Using the node-wise constraints, (2.31) can be rewritten as a consensus problem

$$\min_{\{x_i\in X\},z} \sum_{i=1}^{M} f_i(x_i) \tag{2.32a}$$

$$\text{s.t.} \quad x_i = z, \forall i \in \mathbb{Z}^M \tag{2.32b}$$

For some penalty $\rho > 0$, the augmented lagrange is

$$L_\rho(\{x_i\}, z, \{y_i\}) = \sum_{i=1}^{M} \left(f_i(x_i) + y_i^T (x_i - z) + \frac{\rho}{2}\|x_i - z\|^2 \right) \tag{2.33}$$

where y_i is the dual variable for the constraint $x_i - z = 0$. The ADMM iterates can be given in a distributed manner

$$x_i^{k+1} = \arg\min_{x_i\in X} f_i(x_i) + (y_i^k)^T (x_i - z^k) + \frac{\rho}{2}\|x_i - z^k\|^2 \tag{2.34a}$$

$$z^{k+1} = \frac{1}{M} \sum_{i=1}^{M} \left(x_i^{k+1} + 1/\rho y_i^k \right) \tag{2.34b}$$

$$y_i^{k+1} = y_i^k + \rho (x_i^{k+1} - z^{k+1}) \tag{2.34c}$$

It can be seen from (2.34b) that a central node is needed to update z. Hence, this algorithm is not fully distributed. Consider a more general consensus problem [88]

$$\min_{\{x_i \in X^i\}, z \in \mathbb{R}^n} \quad \sum_{i=1}^{M} f_i(x_i) \tag{2.35a}$$

$$s.t. \;\; x_i - E_i z = 0, i \in \mathbb{Z}^M \tag{2.35b}$$

where $X^i \subseteq \mathbb{R}^{n_i}$ is the local constraint set of agent i, $E_i \in \{0,1\}^{n_i \times n}$ is some appropriate selection matrix such that the local variable x_i corresponds to some component of the global variable z in (2.35b). The augmented Lagrangian is

$$L_\rho(\{x_i\}, z, \{y_i\}) = \sum_{i=1}^{M} \left(f_i(x_i) + y_i^T (x_i - E_i z) + \frac{\rho}{2} \|x_i - E_i z\|^2 \right) \tag{2.36}$$

with $\rho > 0$. The ADMM iterates becomes

$$x_i^{k+1} = \arg\min_{x_i \in X^i} f_i(x_i) + (y_i^k)^T (x_i - E_i z^k) + \frac{\rho}{2} \|x_i - E_i z^k\|^2 \tag{2.37a}$$

$$z^{k+1} = \arg\min_{z} \sum_{i=1}^{N} \left(-(y_i)^T E_i z^k + \frac{1}{2} \rho \|x_i^{k+1} - E_i z\|^2 \right) \tag{2.37b}$$

$$y_i^{k+1} = y_i^k + \rho (x_i^{k+1} - E_i z^{k+1}) \tag{2.37c}$$

As shown in [74], the update of each component of z only requires the the local variables that are coupled with this component in (2.35b). When all E_i are equal to I_n, (2.37) becomes (2.34).

Both the two problems above are coupled via variables. ADMM can also be extended to solve problems that are coupled via constraints. Consider the problem with coupled constraints and M blocks of variables [80, 89–91]

$$\min_{\{x_i \in \mathbb{R}^{n_i}\}} \quad \sum_{i=1}^{M} f_i(x_i) \tag{2.38a}$$

$$s.t. \quad \sum_{i=1}^{M} A_i x_i = c \tag{2.38b}$$

$$x_i \in X_i, i \in \mathbb{Z}^M \tag{2.38c}$$

where $X_i \subset \mathbb{R}^{n_i}$ is a nonempty closed convex set known by agent i. The direct extension of ADMM

yields the following iterates

$$x_1^{k+1} = \arg\min_{x_1 \in X_1} f_1(x_1) + (y^k)^T (A_1 x_1 + \sum_{i=2}^{M} A_i x_i^k - c) + \frac{\rho}{2} \|A_1 x_1 + \sum_{i=2}^{M} A_i x_i^k - c\|^2 \tag{2.39a}$$

$$\vdots$$

$$x_i^{k+1} = \arg\min_{x_i \in X_i} f_i(x_i) + (y^k)^T (\sum_{j=1}^{i-1} A_i x_i^{k+1} + A_i x_i + \sum_{j=i+1}^{M} A_i x_i^k - c) + \frac{\rho}{2} \| \sum_{j=1}^{i-1} A_i x_i^{k+1} + A_i x_i + \sum_{j=i+1}^{M} A_i x_i^k - c\|^2 \tag{2.39b}$$

$$\vdots$$

$$x_M^{k+1} = \arg\min_{x_M \in X_M} f_M(x_M) + (y^k)^T (\sum_{i=1}^{M-1} A_i x_i^{k+1} + A_M x_M - c) + \frac{\rho}{2} \| \sum_{i=1}^{M-1} A_i x_i^{k+1} + A_M x_M - c\|^2 \tag{2.39c}$$

$$y^{k+1} = y^k + \rho (\sum_{i=1}^{M} A_i x_i^{k+1} - c) \tag{2.39d}$$

with $\rho > 0$ and y being the dual variable of the constraint $\sum_{i=1}^{M} A_i x_i - c = 0$. The first M optimization problems are solved sequentially over the M agents and the dual variable is updated only when all the new local variables are received. Although this direction extension of ADMM is very effective in solving many practical problems [92, 93], the convergence is not guaranteed generally [94] and is only guaranteed under some sufficient conditions [80, 94]. This algorithm is also known as Gauss-Seidel ADMM [89], because blocks are updated sequentially. In order to have better parallelization, Jacobian ADMM is proposed in [91]

$$x_i^{k+1} = \arg\min_{x_1 \in X_1} f_1(x_1) + (y^k)^T (A_i x_i + \sum_{j \neq i}^{M} A_j x_j^k - c) + \frac{\rho}{2} \|A_i x_i + \sum_{j \neq i}^{M} A_j x_j^k - c\|^2 \tag{2.40a}$$

$$y^{k+1} = y^k + \rho (\sum_{i=1}^{M} A_i x_i^{k+1} - c) \tag{2.40b}$$

According to the discussion above, ADMM can be regarded as a powerful tool of parallel and distributed computation. However, further investigation is needed when it is implemented for the MPC formulation including the design of the stopping criterion. Moreover, in the presence of coupled constraints, the extended ADMM formulations (2.39) and (2.40) need information from all the agents. This will require a central node or a fully connected network and hence is conservative. To circumvent this problem, this thesis will propose a consensus-based ADMM approach that only requires a

connected network.

2.4 Nestrov's Fast Gradient Method

This section discusses the Nestrov's fast gradient method including its application to the ADMM and distributed gradient formulations.

2.4.1 Gradient and Distributed Gradient Method

Consider a differentiable convex function $f:\mathbb{R}^n \to \mathbb{R}$, the gradient method takes the following iterate to minimize $f(x)$ subject to some closed convex constraint $x \in X$

$$x^{k+1} = P_X[x^k - c\nabla f(x^k)] \tag{2.41}$$

where $P_X[\cdot]$ denote the projection onto the set X and $c > 0$ is some constant step-size. We assume that $\nabla f(x)$ is Lipschitz continuous with parameter L in X

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|, \quad \forall x,y \in X \tag{2.42}$$

Let $f^* = \min\{f(x) : x \in X\}$. Provided c is sufficiently small, the gradient method will converge to f^* and the number of iterations to reach $f(x^k) - f^* \leq \varepsilon$ is $O(1/\varepsilon)$. In addition, suppose $f(x)$ is strongly convex, the gradient method converges to f^* with an exponential rate, see Chapter 2 of [95].

Suppose $f(x)$ is the sum of M differentiable and separable objective functions in the form of (2.31). The iterate in (2.41) can be rewritten as

$$x^{k+1} = P_X[x^k - c\sum_{i=1}^{M} \nabla f_i(x^k)] \tag{2.43}$$

This formulation admits a distributed implementation [85, 86, 96], where each objective function is known by an agent and the M agents are connected by a network. Consider the graph $G = (\mathcal{V}, \mathcal{E})$

with the weight matrix $W(k) = [w_{ij}(k)]$, the update rule at agent i is

$$x_i^{k+1} = P_X[\sum_{j\in N_i\cup\{i\}} w_{ij}(k)x_j^k - \alpha_k \nabla f_i(x_i^k)] \tag{2.44}$$

where $N_i := \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\}$ and $\{\alpha_k\}$ is some proper step-size sequence. The detailed discussion can be found in Section 3.2 with other distributed optimization algorithms.

2.4.2 Centralized Fast Gradient Method

A fast gradient method was proposed in [97] to accelerate the standard gradient method. The method consists of a extrapolation step and a gradient projection step

$$y^k = x^k + \beta_k(x^k - x^{k-1}) \tag{2.45}$$

$$x^{k+1} = P_X[y^k - c\nabla f(y^k)] \tag{2.46}$$

where $x^{-1} = x^0$ and $\beta_k \in (0,1)$. β_k is considered as the extrapolation factor. As shown in Section 6.9 of [98], with proper choice of β_k, the iteration complexity to reach $f(x^k) - f^* \le \varepsilon$ is $O(1/\sqrt{\varepsilon})$ or the cost function has the convergence rate $O(1/k^2)$. The choice of β_k is

$$\beta_k = \frac{\theta_k(1-\theta_{k-1})}{\theta_{k-1}} \tag{2.47}$$

where $\{\theta_k\}$ satisfies $\theta_0 = \theta_1 \in (0,1]$, and

$$\frac{1-\theta_{k+1}}{\theta_{k+1}^2} \le \frac{1}{\theta_k^2}, \quad \theta_k \le \frac{2}{k+2}, \quad k \ge 0 \tag{2.48}$$

One choice of θ_k is that $\theta_{-1} = \theta_0 = 1$ and $\frac{1-\theta_{k+1}}{\theta_{k+1}^2} = \frac{1}{\theta_k^2}$ for all $k \ge 0$. It can be concluded by induction that $\theta_k \le \frac{2}{k+2}$ and $\theta_{k+1} < \theta_k$ for all $k \ge 0$. Another choice is that $\theta_{-1} = 1$ and $\theta_k = \frac{2}{k+2}$ for all $k \ge 0$. Other two Nesterov fast gradient methods were proposed in [99, 100] with the same convergence rate. A unified framework of fast gradient methods can be found in [101]

2.4.3 Centralized Fast ADMM

The Nesterov's acceleration technique can also be adapted to ADMM to solve the following problem. The work [102] proposes a Nesterov-type acceleration for the problem (2.28) with the convergence rate $O(1/k^2)$ in the case where, $X = \mathbb{R}^n$, $Z = \mathbb{R}^m$, and A and B are identity matrices. Another fast ADMM is proposed in [103] to solve the same unconstrained problem ($X = \mathbb{R}^n$ and $Z = \mathbb{R}^m$) for general A and B with the convergence rate $O(1/k^2 + 1/k)$. In [104], a fast linearized ADMM was proposed with the convergence rate $O(1/k^2 + 1/k)$ for the cost and $O(1/k^{\frac{3}{2}} + 1/k)$ for the equality constraint. A better convergence rate $O(1/k^2)$ can be obtained in the work [105] on the condition that $f(x)$ and $g(z)$ are strongly convex. However, for weakly convex function, a restart procedure is needed to guaranteed the convergence of this algorithm [105].

2.4.4 Distributed Fast Gradient Methods

This section reviews some distributed Nesterov-type gradient methods to solve problems in the form of (2.31). In [106], the authors directly incorporated the Nesterov accelerating technique with the distributed gradient method discussed in Section 2.4.1. Consider $X = \mathbb{R}^n$ and the graph $G = (\mathcal{V}, \mathcal{E})$ with the fixed weight matrix $W = [w_{ij}]$, the update rule at agent i is

$$x_i^{k+1} = \sum_{j \in N_i \cup \{i\}} w_{ij} y_j^k - \alpha \nabla f_i(y_i^k) \tag{2.49a}$$

$$y_i^{k+1} = x_i^{k+1} + \beta_k (x_i^{k+1} - x_i^k) \tag{2.49b}$$

where $\alpha > 0$ is some fixed step-size and the extrapolation factor is set to be $\beta_k = \frac{k-1}{k+2}$. The convergence results of this algorithm is derived on the conditions that both $f_i(x)$ and $\nabla f_i(x)$ are Lipschitz continuous and the step-size α should be sufficiently small. For the constant step-size α, this algorithm does not converge to the exact optimal solution but only to some neighborhood of the optimal. The convergence rate of the cost is $O(1/\alpha + \alpha/k^2 + 1/k^2)$. The smaller the step-size α is, the closer the solution is to the optimal. However, the convergence speed will be slow for small α. In order to approach the exact solution, a diminishing step-size should be used, e.g. $\alpha_k = \frac{c}{k+1}$ for some $c > 0$. However, the convergence speed will become slow for a diminishing step-size. As shown in [107, 108], the convergence rate for the diminishing step-size $\alpha_k = \frac{c}{k+1}$ is $O(\log k/k)$, which is

slower than the one with the constant step-size. The work [109] also proposed a distributed Nesterov-type gradient method using the constant step-size and it is able to achieve $O(1/k)$ convergence rate with additional iteration steps. In [107], the authors proposed a consensus-based distributed Nesterov gradient method that uses consensus iterations in the inner loop. The convergence rate of this method is $O(1/k^2)$, which is the the best convergence rate that Nesterov's gradient methods can achieve. However, the number of the inner-loop consensus steps grows with the outer-loop index, which results in the significant increase in the number of information exchanges. This thesis aims to overcome this problem by using a inner loop with fixed number of steps.

2.5 Distributed Model Predictive Control

This section reviews existing DMPC approaches for a group of interacting systems, which can be coupled in various forms, including dynamics, cost functions and constraints.

2.5.1 Coupled Dynamics

Dynamically coupled systems can be found in the various areas such as power systems [62, 110], process control systems [111] and supply chain systems [112]. Several DMPC approaches have been proposed in [59, 113, 114] for dynamically coupled linear time-invariant systems with no state and input constraints. Conditions for stability of the closed-loop system are discussed in these approaches. [115] presents a min-max DMPC approach for dynamically coupled nonlinear time-invariant systems with state constraints, where each local system treats the neighboring system states as disturbances. The DMPC problem for system i is formulated as

$$\min_{\boldsymbol{x}^i,\boldsymbol{u}^i} \max_{\boldsymbol{v}^i} J^i(\boldsymbol{x}^i,\boldsymbol{u}^i,\boldsymbol{v}^i) \tag{2.50}$$

where $J^i(\boldsymbol{x}^i,\boldsymbol{u}^i,\boldsymbol{v}^i)$ is the performance index, $\boldsymbol{x}^i$ and $\boldsymbol{u}^i$ are the predicted state and input sequences, and $\boldsymbol{v}^i$ is the disturbance sequence. Note that $\boldsymbol{x}^i$ and $\boldsymbol{v}^i$ are subject to some proper constraints. The stability condition is obtained by imposing the predicted state bound as constraints in the DMPC problem. Each system broadcasts the bounds to its neighbors and at the same time receives the bound for $\boldsymbol{v}^i$ from the neighbors. Using these bounds, all the systems solve the min-max problem

(2.50) to optimize performance with respect to worst-case disturbances. As the solutions obtained from the approaches above are not necessary optimal, a DMPC framework is proposed in [51, 61] for a set of coupled linear systems to seek the optimal solution. Consider M coupled linear systems with the following overall DMPC problem

$$\min_{\{\boldsymbol{u}^i\}} \sum_{i=1}^{M} w_i J^i(\boldsymbol{x}^i, \boldsymbol{u}^i, \boldsymbol{u}^{j\neq i}) \tag{2.51}$$

where $w_i \geq 0$, $\sum_{i=1}^{M} w_i = 1$, and the state and input sequence are subject to proper constraints. All the systems work iteratively and cooperatively towards achieving the overall optimal solution of (2.51). At iteration p, for all $i \in \mathbb{Z}^M$, the i^{th} system solves the following local problem with all the other systems holding constant

$$\min_{\boldsymbol{u}^i} \sum_{i=1}^{M} w_i J^i(\boldsymbol{x}^i, \boldsymbol{u}^i, \boldsymbol{u}^{j\neq i,p-1}) \tag{2.52}$$

where $\boldsymbol{u}^{j\neq i,p-1}$ denotes the solutions of the other systems at iteration $p-1$. This process is repeated until the stopping criterion is met. However, the systems in [51, 61] are coupled only through the control inputs. In [37], a decentralized approach is presented for large-scale dynamical processes subject to input constraints. In [37], the overall model of the process is decoupled into several (possibly overlapping) smaller models which are used for optimizing the local problems. Suppose the overall model (2.11) is decoupled into M submodels. For all $i \in \mathbb{Z}^M$, define $x^i \in \mathbb{R}^{n_i}$ as the vector collecting a subset of the state components and $u^i \in \mathbb{R}^{m_i}$ as the vector collecting a subset of the input components

$$x^i = W_i^T x, \quad u^i = Z_i^T u, \tag{2.53}$$

where $W_i \in \mathbb{R}^{n\times n_i}$ and $Z_i \in \mathbb{R}^{m\times m_i}$ are the appropriate selection matrices with $W_i^T W_i = I_{n_i}$ and $Z_i^T Z_i = I_{m_i}$. By the definition of x^i in (2.53), the following expression can be obtained

$$x^i(t+1) = W_i^T x(t+1) = W_i^T A x(t) + W_i^T B u(t) \tag{2.54}$$

An approximation of the overall model (2.11) is obtained by replacing $W_i^T A$ and $W_i^T B$ in (2.54) with $W_i^T A W_i W_i^T$ and $W_i^T B Z_i Z_i^T$ respectively, therefore getting the reduced submodel

$$x^i(t+1) = A^i x^i(t) + B^i u(t) \tag{2.55}$$

where $A^i = W_i^T A W_i W_i^T$ and $B^i = W_i^T B Z_i Z_i^T$. Using this submodel, the DMPC can be designed individually according to the standard MPC. The limitation of this approach is that it requires the overall system and decoupled local systems to be open-loop asymptotically stable. Another DMPC approach is proposed in [50] for dynamically coupled nonlinear systems that are subject to independent input constraints. In [50], all systems exchange the predicted trajectories with their neighbors and a consistency constraint is imposed on the MPC problem to ensure that the systems stay close to the predicted trajectories. Each system then solves a local DMPC problem similar to [115]. More recently, [48, 49] propose the use of tube-based method [116] to deal with the dynamical coupling among systems. The dynamical coupling is treated as the disturbance that lies inside some bounded set. The dynamics of the i^{th} system is given by

$$x^i(t+1) = A^{ii} x^i(t) + B^i u^i(t) + w^i(t), \tag{2.56}$$

$$w^i(t) = \sum_{j \in N_i} A^{ij} x^j(t) \tag{2.57}$$

where N_i denotes the neighbors of system i. Using this model, the robust MPC can be designed for each system by the tube-based method [116]. However, in most of the approaches above, the optimality properties have not been established and the choice of the terminal set is conservative.

2.5.2 Coupled Cost Functions

Another interesting problem is where the systems are coupled through the cost functions. Its application can be found in multi-agent coordination problems using coupling penalty functions [117, 118]. A DMPC approach for multi-vehicle formation stabilization is proposed in [43], where the systems are only coupled in the cost functions with decoupled dynamics and constraints. For each system i, the local cost function is given in the form of $J_i(x^i, x^{-i}, u^i)$ where x^{-i} denotes the states of the neighbors of system i. Let $\hat{x}^i$ denote the predicted trajectory of system i. The communication among

systems is needed so that each system can broadcast the previous optimal control trajectory to its neighbors. The stability can be obtained on the condition that the actual trajectory is within some pre-specified neighborhood of the previous optimal control trajectory. At each sampling time, each system solves

$$\min_{u^i} J_i(x^i, x^{-i}, u^i) \tag{2.58}$$

subject to the constraint on the distance to the predicted trajectory $\|x^i - \hat{x}^i\| \leq \delta$ for some pre-specified $\delta > 0$ and other standard constraints. In [35], the DMPC approach also allows the local cost functions to be dependent on the state and the inputs of the neighbors. Besides the states, the inputs are also coupled in the cost functions. The stage cost of system i can be given by

$$l^i(x^i, u^i, x^{-i}, u^{-i}) = l^{ii}(x^i, u^i) + \sum_{j \in N_i} l^{ij}(x^i, u^i, x^j, u^j) \tag{2.59}$$

where $l^{ii}(x^i, u^i)$ is the local cost and $l^{ij}(x^i, u^i, x^j, u^j)$ is the coupled term. Stability is achieved with a zero terminal constraint set. The DMPC approach in [46] uses an invariant set as the terminal set and introduces an easily verifiable constraint in each system to ensure stability of the overall system. Similar to [43], the states are coupled in the local cost functions while the inputs are decoupled. The stability is discussed under two cases. Each system is asymptotically stable in the disturbance-free case and the state of each subsystem converges to some neighborhood of the origin asymptotically in the non-zero disturbance case. More recently, the works in [66, 67, 119] study the DMPC for a set of linear systems with both coupled dynamics and cost functions. The cost functions are coupled only by neighboring states. The local cost function of system i is given by $J_i(\boldsymbol{x}^{N_i}, \boldsymbol{u}^i)$, where $\boldsymbol{x}^{N_i}$ is the predicted state sequence of all the neighbors and its own and $\boldsymbol{u}^i$ is the local predictive control sequence. Then, the overall MPC problem of all the M systems can be given by

$$\min_{\{\boldsymbol{x}^{N_i}, \boldsymbol{u}^i\}} \sum_{i=1}^{M} J_i(\boldsymbol{x}^{N_i}, \boldsymbol{u}^i) \tag{2.60}$$

subject to the state and input constraints. This problem should be solved by all the M systems in a cooperative and distributed manner. The variable $\{\boldsymbol{x}^{N_i}, \boldsymbol{u}^i\}$ is considered as the local variable of system i. Note that these local variables can overlap. These works propose to formulate this problem

into the form of (2.35) by introducing an overall consensus variable and then solve it by ADMM. This idea can be also found in a more general framework of the ADMM-based solution of the DMPC problem [88]. Recursive feasibility stability is guaranteed by introducing time-varying ellipsoidal sets induced from a block diagonal Lyapunov matrix. However, it is conservative to use ellipsoidal sets as the terminal set and restrict Lyapunov matrix to be block-diagonal for linear systems.

2.5.3 Coupled Constraints

There are also problems that the systems are linked by coupling constraints. A robust DMPC approach is proposed in [44] of a set of dynamically decoupled linear systems with coupled constraints and bounded disturbances. The DMPC problem is solved at each sampling time in a sequential process where the local problems are solved in a fixed sequence and each system is optimized with all others holding constant. Let $\tilde{\boldsymbol{Z}}^i$ denote the set of the variables of other systems that share the same coupled constraints with system i. Using the information $\tilde{\boldsymbol{Z}}^i$, the problem solved by system i is denoted by subproblem i, given by

$$\min_{\boldsymbol{u}^i} J_i(\boldsymbol{x}^i, \tilde{\boldsymbol{Z}}^i, \boldsymbol{u}^i) \tag{2.61}$$

subject to the local constraints and coupled constraints. The variables of other systems in the coupled constraints are fixed and only the local variable is free. To ensure the constraint satisfaction, communication among systems is needed and each system is required to broadcast its most recent optimal trajectory to its downstream system. The overall algorithm of the M systems is implemented at each sampling time in the following sequence: Subproblem 1 $\rightarrow$ Subproblem 2 $\rightarrow \cdots \rightarrow$ Subproblem M. To handle the bounded disturbance, the tightened method proposed in [120] is also used in this formulation. By the use of the tube-based method [116], the work [120] is then extended in [121] where only one system is optimized at each sampling time while the other systems use the previous control plan. The agents thus update in a sequence, $\{p_1, p_2, \cdots, p_k, p_{k+1}, \cdots\}$, to be chosen by the designer. Suppose $p_t = i$ at time t, system i is activated and it solves (2.61) with the information $\tilde{\boldsymbol{Z}}^i$ received from the coupled systems. In order to obtain better overall performance, [60] further extends the DMPC approach in [121] to promote inter-agent cooperation by designing a cooperative set. All the systems within the cooperating set are optimized jointly with a weighted sum of

local cost functions. For each system i, a cooperative set C_i is designed and the weighted local cost function is given by

$$J_i(\boldsymbol{u}^i) + \sum_{j \in C_i} \alpha_{ij} J_j(\hat{\boldsymbol{u}}^j) \tag{2.62}$$

where $\hat{\boldsymbol{u}}^j$ is an artificial variable and $\alpha_{ij} > 0$ is the weight. The state sequences are dropped in the cost function for notational simplicity. Let $\hat{\boldsymbol{u}}^{C_i}$ denote the set of the artificial variables of system i. When activated, system i solves the following problem

$$\min_{\boldsymbol{u}^i, \hat{\boldsymbol{u}}^{C_i}} J_i(\boldsymbol{u}^i) + \sum_{j \in C_i} \alpha_{ij} J_j(\hat{\boldsymbol{u}}^j) \tag{2.63}$$

subject to the local and coupled constraints, similar to (2.61). As a computational improvement of [60], the work in [122] introduces a parallel computation scheme by permitting the simultaneous optimizing of local problems at each time step while maintaining robust feasibility and stability. In this approach, a sequence of sets are designed, $\{\ell^{opt}(0), \ell^{opt}(1), \cdots, \ell^{opt}(k), \ell^{opt}(k+1), \cdots\}$, and all the systems in $\ell^{opt}(t)$ are activated at time t. However, in all these approaches, the optimality of the overall system still remains unclear as the optimality properties are not explicitly pursued.

CHAPTER 3

Consensus Algorithms and Finite-Time Consensus

Distributed coordination of multi-agent systems has received extensive attention in recent years and its applications can be found in various areas, including unmanned air vehicles(UAVs) [123], autonomous underwater vehicles (AUVs) [124,125], formation control [126,127], mobile robots [128], and congestion control in communication networks [129], to name a few. In many circumstances, a group of agents need to reach some common agreement. These are known as consensus problems and they can be addressed by designing consensus algorithms or protocols where the agents negotiate and exchange information with their neighbors [130, 131]. This chapter discusses the formulation of consensus coordination problems and multi-agent optimization. A finite-time consensus algorithm is also proposed in this chapter. The network of M agents can be described as a (connected) graph $G = (\mathcal{V}, \mathcal{E})$ with the vertex set $\mathcal{V} = \{1, 2, \cdots, M\}$ and edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The set of neighbors of the i^{th} agent is denoted by $N_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ and let $|N_i|$ denote the number of its neighbors. The neighbors can communicate and exchange messages with each other.

3.1 Review of Consensus Coordination Problems

The objective of the consensus problems is to drive all the agents to a common state for some given initial state. Let x_i denote the state of the i^{th} agent for $i \in \mathbb{Z}^M$. Without loss of generality, x_i is assumed to be scalar. The discussion can be easily extended to the vector case. The consensus condition is reached if $x_1 = x_2 = \cdots = x_M$. Let $x := (x_1, x_2, \cdots, x_M) \in \mathbb{R}^M$ be the overall state. The consensus can also be expressed as $x = c\mathbf{1}$ for some $c \in \mathbb{R}$.

A continuous-time distributed consensus protocol is in the form of [130, 132, 133]

$$\dot{x}_i(t) = \sum_{j \in N_i} a_{ij}(x_j(t) - x_i(t)) \tag{3.1}$$

where $\mathcal{A} = [a_{ij}]$ denotes the weighted adjacency matrix of G with $a_{ij} > 0$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ if $(i, j) \notin \mathcal{E}$. The standard adjacency matrix $\mathcal{A}$ is defined as

$$a_{ij} = \begin{cases} 1, & (i, j) \in \mathcal{E} \\ 0, & (i, j) \notin \mathcal{E} \end{cases} \tag{3.2}$$

The idea of this protocol is to drive each system to the weighted average state of its neighbors by local information exchange. Let $D := diag\{d_1, d_2, \cdots, d_M\}$ denote the degree matrix of G with $d_i = \sum_{j \in N_i} a_{ij}$ for all $i \in \mathbb{Z}^M$. The compact form of (3.1) can be expressed as

$$\dot{x}(t) = -Lx(t) \tag{3.3}$$

where $L = D - \mathcal{A}$ is known as the Laplacian matrix of G. From the choice of $d_i = \sum_{j \in N_i} a_{ij}$, it can be seen that $L\mathbf{1} = 0$. The stability of the consensus protocol is usually discussed by the use of Geršgorin disk theorem [132]. A brief summary of the stability results is given here. As shown in [132, 134], 0 is a simple eigenvalue if G is a strongly connected graph, where there is a directed path connecting any two arbitrary nodes of the graph. If G is a strongly connected graph, a consensus can be reached asymptotically [130, 133]. However, this is only a sufficient condition. The necessary and sufficient condition can be found in [132]. For an undirected connected graph G(e.g. $a_{ij} = a_{ji}$ for all $(i, j) \in \mathcal{E}$), L is symmetric and $L \succeq 0$. For such a graph, the asymptotic stability is guaranteed and the convergence rate of the consensus protocol is determined by the second smallest eigenvalue of L, which is also known as the algebraic connectivity of G [135].

The discrete-time protocol of (3.1) is given as

$$x_i(t+1) = x_i(t) + \gamma \sum_{j \in N_i} a_{ij}(x_j(t) - x_i(t)) \tag{3.4}$$

or, in the compact form as

$$x(t+1) = Wx(t) \tag{3.5}$$

where $W = I - \gamma L$ and γ is considered as the step-size. In general, $W = e^{-\gamma L}$ and (3.4) is considered as the first-order approximation of (3.1). If γ is less than the maximum degree $\Delta := \max_i\{d_i\}$, W is a row stochastic nonnegative matrix as $W\mathbf{1} = \mathbf{1}$ and $I - \gamma L$ is nonnegative [130]. The stability analysis follows similar arguments as the continuous-time case. If G is a strongly connected graph and $0 < \gamma < \Delta$, a consensus can be reached asymptotically. The details can be found in [130, 133].

3.2 Multi-Agent Consensus Optimization

Another problem arises in distributed coordination of multi-agent systems is the consensus optimization problem, where the goal is to optimize the sum of local objective functions of all the agents in the network. Consider the following unconstrained problem

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{M} f_i(x) \tag{3.6}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is a convex function that is known locally to agent i only. This problem was originally presented in [84]. Existing methods to solve this type of problems include distributed subgradient algorithms [85, 86, 96], dual averaging methods [136–138], the augmented Lagrangian (AL) methods [139–142] and the ADMM algorithms [143–146]. The dual averaging methods have better performance than the subgradient algorithms for nonsmooth problems, although theoretically they have the same convergence rate. In general, the AL methods converge faster than the standard subgradient/gradient algorithms. Among all these algorithms, the ADMM algorithms demonstrates fast convergence in many applications. This section reviews some well-known algorithms to solve problems in the form of (3.6).

3.2.1 Distributed Subgradient Algorithms

The framework of distributed subgradient algorithms is shown as follows. The distributed subgradient algorithms are designed in [85, 86, 96] based on time-varying networks and can be applied naturally to time-invariant networks. In particular, [85] studied the unconstrained optimization and [86, 96] studied the constrained optimization. We start with the unconstrained optimization. Let $x_i(t)$ denote the state of agent i. At each time t, agent i updates its state according to the following iterative rule [85].

$$x_i(t+1) = \sum_{j=1}^{M} w_{ij}(t)x_j(t) - \alpha_i(t)d_i(t) \tag{3.7}$$

where $w_{ij}(t)$ is some weight and $\alpha_i(t) > 0$ is a stepsize and the vector $d_i(t)$ is a subgradient of the local objective $f_i(x)$ at $x = x_i(t)$. Hence, the update rule is the combination of the consensus protocol and gradient descent. Let the time-varying graph be denoted by $G(t) := (\mathcal{V}, \mathcal{E}(t), W(t))$ with $W(t) := [w_{ij}(t)]$. Under some proper conditions on $G(t)$ [85], the distributed subgradient algorithm with bounded subgradients and constant step-size α in (3.7) has an error of $O(\alpha + \frac{1}{\alpha t})$ in the cost after t iterations. Hence, when $t \to \infty$, the solution obtained from the algorithm (3.7) has a constant error of $O(\alpha)$. For the optimized α, a ε-optimal solution can be obtained in $O(1/\varepsilon^2)$ steps. In order to achieve the exact solution, a diminishing step-size should be used.

This algorithm can also be extended to constrained problems. Suppose $x_i(t)$ is constrained to lie in a nonempty closed convex set $X_i \subseteq \mathbb{R}^n$ which is only known to agent i. The constrained problem can be formulated as

$$\min_{x\in\mathbb{R}^n} \sum_{i=1}^{M} f_i(x) \tag{3.8a}$$

$$\text{s.t.} \quad x \in \bigcap_{i=1}^{M} X_i \tag{3.8b}$$

The update rule of agent i becomes

$$x_i(t+1) = P_{X_i}\left[\sum_{j=1}^{M} w_{ij}(t)x_j(t) - \alpha_i(t)d_i(t)\right] \tag{3.9}$$

The convergence of the constrained distributed subgradient algorithm can also be derived. The

details of the convergence analysis can be found in [85, 86, 96].

3.2.2 Dual Averaging Methods

The problem in (3.6) can also be solved by the dual averaging method. In this section, we present the standard centralized dual averaging method, followed by its distributed implementation. This method is proposed in [147] to solve general constrained convex problems. It is then extended by [148] to regularized optimization problems. Consider the convex optimization problem

$$\min_{x\in X} f(x) \tag{3.10}$$

where $X \subseteq \mathbb{R}^n$ is some nonempty convex set and $f : X \to \mathbb{R}$ is convex(possibly nonsmooth). The dual averaging method needs a proximal function $\psi : X \to \mathbb{R}$ with $\psi(0) = 0$ and $\psi(x) \geq 0$ for all $x \in X$. ψ is also assumed to be strongly convex and satisfies:

$$\psi(y) \geq \psi(x) + \nabla\psi^T(x)(y-x) + \frac{1}{2}\|y-x\|^2, \forall x, y \in X \tag{3.11}$$

The dual averaging method generates a sequence of iterates $\{x(t), s(t)\}_{t=0}^{\infty}$ according to the following update rule

$$s(t+1) = s(t) + \lambda(t)d(t), \tag{3.12}$$

$$x(t+1) = \Pi_X^{\psi}(s(t+1), \alpha(t)) \tag{3.13}$$

where $\lambda(t) > 0$, $d(t) \in \partial f(x(t))$, $\{\alpha(t)\}_{t=0}^{\infty}$ is a non-increasing sequence of positive stepsizes, and the projection operator $\Pi_X^{\psi}(\cdot,\cdot)$ is

$$\Pi_X^{\psi}(s, \alpha) := \arg\min_{x\in X}\{s^T x + \frac{1}{\alpha}\psi(x)\} \tag{3.14}$$

There are two strategies for choosing $\lambda(t)$ [147]: 1) $\lambda(t) = 1$ (simple averages); 2) $\lambda(t) = \frac{1}{\|d(t)\|_*}$ (weighted averages), where the dual norm $\|\cdot\|_*$ is defined as $\|d\|_* := \max_x\{d^T x : \|x\| \leq 1\}$. The convergence results of the dual averaging method above can be found in [147]. The distributed dual averaging method is presented in [137, 138] to solve optimization problems in the form of (3.6) with

the constraint $x \in X$. We consider the time-invariant undirected connected graph $G = (\mathcal{V}, \mathcal{E})$ with the symmetric doubly stochastic weight matrix $W := [w_{ij}]$. Given the non-increasing sequence $\alpha(t)$, each agent i updates its state according to the following rule

$$s_i(t+1) = \sum_{j \in N_i} w_{ij} s_j(t) + d_i(t), \tag{3.15}$$

$$x_i(t+1) = \Pi_X^{\psi}(s_i(t+1), \alpha(t)) \tag{3.16}$$

where $d_i(t) \in \partial f_i(x_i(t))$. This update rule uses simple averages. Since W is doubly stochastic, the convergence rate of the distributed dual averaging method is controlled by the second largest singular value of W. The explicit convergence rate is studied in [137] under various types of graphes.

3.2.3 Augmented Lagrangian Methods

Another important approach to solve the consensus optimization problem of (3.6) is the Augmented Lagrangian (AL) method, which plays an important role in constrained optimization [149]. In order to use the AL method, for a graph $G = (\mathcal{V}, \mathcal{E})$, (3.6) is reformulated as [139–142]

$$\min_{x_1, x_2, \cdots, x_M} \sum_{i=1}^{M} f_i(x_i) \tag{3.17a}$$

$$\text{s.t.} \quad x_i = x_j, (i, j) \in \mathcal{E} \tag{3.17b}$$

where x_i is the local copy of x for agent i. The augmented Lagrangian is

$$L_\rho(x_1, x_2, \cdots, x_M, \{\lambda_{ij}\}) = \sum_{i=1}^{M} f_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \left(\lambda_{ij}^T (x_i - x_j) + \frac{\rho}{2} \|x_i - x_j\|^2 \right) \tag{3.18}$$

with λ_{ij} being the dual variable of the constraint $x_i - x_j = 0$ and the penalty parameter $\rho > 0$. The standard AL method consist of the iterates

$$(x_1^{t+1}, x_2^{t+1}, \cdots, x_M^{t+1}) = \arg \min_{x_1, x_2, \cdots, x_M} L_\rho(x_1, x_2, \cdots, x_M, \{\lambda_{ij}^t\}) \tag{3.19}$$

$$\lambda_{ij}^{t+1} = \lambda_{ij}^t + \rho(x_i^{t+1} - x_j^{t+1}), (i, j) \in \mathcal{E} \tag{3.20}$$

The inner loop problem is usually solved by Jacobi/Gauss-Seidel algorithms [84, 150] where the solution $\{x_i^{t+1}\}$ is obtained in a distributed manner. In the Jacobi algorithm, the values of x_i^t obtained in iteration t remain unchanged until all the agents complete iteration $t+1$. Agent i uses the following update rule.

$$x_i^{t+1} = \arg\min_{x_i} L_\rho(x_1^t, x_2^t, x_i, \cdots, x_M^t, \{\lambda_{ij}^t\}) \tag{3.21}$$

$$\lambda_{ij}^{t+1} = \lambda_{ij}^t + \rho(x_i^{t+1} - x_j^{t+1}), (i,j) \in \mathcal{E} \tag{3.22}$$

However, in the Gauss-Seidel algorithm, the values of x_i^t are updated as soon as agent i complete iteration $t+1$, see, e.g

$$x_1^{t+1} = \arg\min_{x_1} L_\rho(x_1, x_2^t, \cdots, x_M^t, \{\lambda_{ij}^t\}) \tag{3.23}$$

$$x_2^{t+1} = \arg\min_{x_2} L_\rho(x_1^{t+1}, x_2, \cdots, x_M^t, \{\lambda_{ij}^t\}) \tag{3.24}$$

$$\vdots$$

$$x_M^{t+1} = \arg\min_{x_M} L_\rho(x_1^{t+1}, x_2^{t+1}, \cdots, x_M, \{\lambda_{ij}^t\}) \tag{3.25}$$

$$\lambda_{ij}^{t+1} = \lambda_{ij}^t + \rho(x_i^{t+1} - x_j^{t+1}), (i,j) \in \mathcal{E} \tag{3.26}$$

From the two formulations above, the Jacobi algorithm can be implemented in parallel while the Gauss-Seidel algorithm can only be implemented sequentially. The convergence of the two algorithms is discussed in [150]. A more detailed review of the distributed AL methods can be found in [141].

3.2.4 Distributed ADMM Algorithms

The distributed ADMM algorithms have faster convergence than the methods mentioned above, see [143–146]. However, the ADMM algorithm mentioned in Section 2.3.2 is based on the node-wise formulation and is not fully distributed. This section discusses a fully distributed ADMM algorithm. Consider an undirected graph $G = (\mathcal{V}, \mathcal{E})$, a reformulation of (3.6) is needed to implement

the distributed ADMM

$$\min_{\{x_i\},\{z_{ij}\}} \sum_{i=1}^{M} f_i(x_i) \tag{3.27a}$$

$$\text{s.t.} \quad x_i = z_{ij}, x_j = z_{ij}, (i,j) \in \mathcal{E} \tag{3.27b}$$

where z_{ij} is an auxiliary edge-wise variable imposing the consensus constraint on edge $(i,j) \in \mathcal{E}$. This formulation is equivalent to (3.6) when G is connected. The formulation (3.27) can be rewritten by

$$\min_{\boldsymbol{x},\boldsymbol{z}} \sum_{i=1}^{M} f_i(x_i) \tag{3.28a}$$

$$\text{s.t.} \quad A\boldsymbol{x} + B\boldsymbol{z} = 0 \tag{3.28b}$$

where $\boldsymbol{x}$ is the vector that collects all $\{x_i\}$, $\boldsymbol{z}$ is the vector that collects $\{z_{ij}\}$, and A and B are some matrices that capture the graph G, see [146]. Hence, the standard ADMM procedure can be applied to this problem as discussed in Section 2.3.1. Let α_{ij} and β_{ij} denote the dual variables associated with $x_i = z_{ij}$ and $x_j = z_{ij}$ in (3.27) respectively. Using the manipulations of distributed ADMM [145, 146, 151, 152], the steps in the standard ADMM can be simplified by letting

$$\alpha_{ij}^k + \beta_{ij}^k = 0, \alpha_{ij}^k = \beta_{ji}^k, z_{ij}^k = \frac{x_i^k + x_j^k}{2}, \tag{3.29}$$

for all $(i,j) \in \mathcal{E}$ and k. The simplified distributed ADMM is then given by

$$x_i^{k+1} = \arg\min_{x_i} f_i(x_i) + (2\sum_{j\in N_i} \alpha_{ij}^k - \rho \sum_{j\in N_i} (x_j^k + x_j^k))^T x^i + \rho |N_i| \|x^i\|^2 \tag{3.30a}$$

$$\alpha_{ij}^{k+1} = \alpha_{ij}^k + \frac{\rho}{2}(x_i^{k+1} - x_j^{k+1}), (i,j) \in \mathcal{E} \tag{3.30b}$$

for some penalty parameter $\rho > 0$. The convergence of the distributed ADMM follows the same arguments of the standard ADMM [79, 80]. In addition, if $f_i(x_i)$ is strongly convex for all $i \in \mathbb{Z}^M$, an exponential convergence rate can be derived [146] and the stopping criterion can be determined using the convergence rate. However, for general convex functions, the best-known convergence rate is $O(1/k)$ and it is very conservative [74]. For this reason, some tight stopping criterion is needed

in the case of non-strongly convex functions.

3.3 Finite-Time Consensus

As most of the consensus algorithms above only produce asymptotic convergence, it is natural to pursue the finite-time convergence to improve the performance of the consensus algorithm. In practice, finite-time convergence is desirable as it provides higher accuracy and better robustness against uncertainties [153]. This section begins with a reviews of some finite-time consensus algorithms, followed by a proposed algorithm using the minimal polynomial of the weight matrix.

3.3.1 Review of Finite-Time Consensus Algorithms

Some finite-time consensus algorithms are reviewed below. The work in [154] studies the application of non-smooth gradient descent flows of a differentiable function for finite-time consensus. By characterizing the asymptotic convergence properties of these non-smooth gradient flows, suitable conditions are identified for finite-time convergence. [155] proposes a finite-time consensus algorithm for discrete-time systems with time-invariant topologies using the property of the minimal polynomial of weight matrix. A summary of this algorithm is given below.

We consider the consensus dynamics $x(t+1) = Wx(t)$ of (3.5). Suppose the minimal polynomial of W is $q(t) := t^T + \pi_0 t^{T-1} + \cdots + \pi_{T-1} t^0$ with T being the degree. This means, from the definition of the minimal polynomial,

$$W^T + \pi_0 W^{T-1} + \cdots + \pi_{T-1} I = 0 \tag{3.31}$$

From (3.5) and (3.31), it can be easily verified that

$$x(T) = W^T x(0) = -(\pi_0 W^{T-1} + \cdots + \pi_{T-1} I)x(0) = -(\pi_0 x(T-1) + \cdots + \pi_{T-1} x(0)) \tag{3.32}$$

Therefore, for all $t \geq 0$, $x_i(t)$ satisfies a linear difference equation of the form

$$x_i(T+t) + \pi_0 x_i(T+t-1) + \cdots + \pi_{T-1} x_i(t) = 0 \tag{3.33}$$

This also mean that $x_i(t)$ can always be computed from the set $\{x_i(0), x_i(1), \cdots, x_i(T-1)\}$ for all $t \geq 0$. Using the Z-transform of the linear difference equation (3.33), it can be obtained that

$$\lim_{t\to\infty} x_i(t) = \frac{\begin{bmatrix} x_i(T-1) & \cdots & x_i(1) & x_i(0) \end{bmatrix} S}{\begin{bmatrix} 1 & \cdots & 1 & 1 \end{bmatrix} S} \tag{3.34}$$

where

$$S = \begin{pmatrix} 1 \\ 1+\pi_0 \\ 1+\pi_0+\pi_1 \\ \vdots \\ 1+\sum_{\ell=0}^{T-1} \pi_\ell \end{pmatrix} \tag{3.35}$$

In addition, [155] also discusses the decentralized calculation of the minimal polynomial. In [156], a framework for finite-time consensus problems is presented and protocols are provided for both the bidirectional interaction case and the unidirectional interaction case. Although more works on finite-time consensus can be found in [157–161], this thesis uses the minimal polynomial to establish the finite-consensus algorithm because of its convenience and simplicity.

3.3.2 The Proposed Finite-Time Consensus Algorithm

This section derives the finite-consensus algorithm without using Z transform. Consider the case of a consensus variable $\boldsymbol{z} = (z^1, z^2, \cdots, z^M) \in \mathbb{R}^M$ over the network G satisfying (A4.3) with z^i being the scalar variable associated with the i^{th} system. Let $L(G)$ be the Laplacian matrix of the network: $L_{ii} = d_i$, the degree of node i; $L_{i,j} = -1$ if $(i,j) \in \mathcal{E}$ and 0 otherwise. A doubly stochastic matrix $W \in R^{M\times M} = I - \gamma L(G)$ for some $0 < \gamma \leq \frac{1}{max_i\{d_i\}}$ can be set up with spectral radius of 1 and the

eigenvalue of 1 is simple with $\frac{1}{\sqrt{M}}\mathbf{1}$ being both its left and right eigenvectors. Then

$$\frac{1}{M}\mathbf{1}\sum_{i=1}^{M} z^i(0) = \frac{1}{M}\mathbf{1}\mathbf{1}^T\boldsymbol{z}(0) = \lim_{\ell\to\infty}\boldsymbol{z}(\ell) \tag{3.36}$$

where $\boldsymbol{z}(\cdot)$ is the state of the consensus dynamics of

$$\boldsymbol{z}(\ell+1) = W\boldsymbol{z}(\ell) \tag{3.37}$$

The last equality of (3.36) holds because $\boldsymbol{z}(\ell) = \sum_{i=1}^{M} v_i{}^{\ell}\xi_i\zeta_i^T\boldsymbol{z}(0)$ [162] is the solution of (3.37) where v_i is the i^{th} eigenvalue of W and $\xi_i(\zeta_i)$ is the corresponding right (left) eigenvector. Since $|v_1| = 1$, $\xi_1 = \zeta_1 = \frac{1}{\sqrt{M}}\mathbf{1}$ and $|v_i| < 1$ for all $i = 2,\cdots,M$, $\lim_{\ell\to\infty}\boldsymbol{z}(\ell) = \frac{1}{M}\mathbf{1}\mathbf{1}^T\boldsymbol{z}(0)$. The expression of (3.36) can be further simplified using (3.37) as

$$\begin{aligned}\lim_{\ell\to\infty}\boldsymbol{z}(\ell) &= \lim_{\ell\to\infty}W^{\ell}\boldsymbol{z}(0) = (\lim_{\ell\to\infty}W^{\ell})\boldsymbol{z}(0)\\ &= (\sum_{\ell=0}^{T-1}\tau_\ell W^{\ell})\boldsymbol{z}(0) = \sum_{\ell=0}^{T-1}\tau_\ell\boldsymbol{z}(\ell)\end{aligned} \tag{3.38}$$

for some $T \leq M$. Here, the first equality of (3.38) follows from the closure property of W^{∞} via the minimal polynomial of W ($t^T + \pi_0 t^{T-1} + \cdots + \pi_{T-1}t^0 = 0$) and $\{\tau_0,\cdots,\tau_{T-1}\}$ can be obtained from $\{\pi_0,\cdots,\pi_{T-1}\}$ using standard results from functions of square matrices [162]. Such a representation is guaranteed to exist since, in the worst case, the characteristic polynomial becomes the minimal polynomial with $T = M$ and the closure property follows from the well-known Caley-Hamilton principle.

Combining (3.36) and (3.38) means that $\frac{1}{M}\mathbf{1}\sum_{i=1}^{M} z^i(0) = \sum_{\ell=0}^{T-1}\tau_\ell\boldsymbol{z}(\ell)$, or, considering each element of this vector equation,

$$\frac{1}{M}\sum_{i=1}^{M} z^i(0) = \sum_{\ell=0}^{T-1}\tau_\ell z^i(\ell) \tag{3.39}$$

This equation shows that the i^{th} system can obtain the value of $\frac{1}{M}\sum_{i=1}^{M} z^i(0)$ by computing its consensus state $z^i(\ell)$ for $\ell = 0,\cdots,T-1$ and evaluating the right hand side of (3.39). Note that this T

steps of z^i is obtained in a distributed manner using the i^{th} row of (3.37), or

$$z^i(\ell+1) = W^{ii}z^i(\ell) + \sum_{j\in N_i} W^{ij}z^j(\ell), \quad i \in \mathbb{Z}^M \tag{3.40}$$

The finite-time consensus algorithm is summarized in the following algorithm

Algorithm 3.1: Finite-time algorithm

Input: $z^i(0)$, $i \in \mathbb{Z}^M$
Output: y^i, $i \in \mathbb{Z}^M$
Initialization: set $k = 0$, $y^i = \tau_0 z^i(0)$, $i \in \mathbb{Z}^M$;
repeat
for all $i \in \mathbb{Z}^M$ (in parallel) **do**
Obtain $z^i(k+1)$ from (3.40);
$y^i = y^i + \tau_{k+1} z^i(k+1)$
end for
$k \leftarrow k+1$
until k=T-1

The above development is for the case where x^i is a scalar. In the case where $z^i \in \mathbb{R}^\eta$ is a vector with $\boldsymbol{z} \in \mathbb{R}^{M\eta}$, the development from (3.36) till (3.40) holds with W replaced by $W \otimes I_\eta$ where $\otimes$ refers the kronecker product of two matrices.

Remark 3.1. *A special case that deserves mention is that when G is a fully connected graph. The corresponding Laplacian matrix has all entries being -1 except the diagonals and all the diagonals have values $M-1$. Hence, $L(G) = MI - \mathbf{1}\mathbf{1}^T$. Suppose $\gamma = \frac{1}{M}$. Then $W = I - \gamma L(G) = I - \frac{1}{M}(MI - \mathbf{1}\mathbf{1}^T) = \frac{1}{M}\mathbf{1}\mathbf{1}^T$. This means that the characteristic polynomial of W is $t^{M-1}(t-1) = 0$. Since $L(G)$ is symmetric, all Jordan blocks in the Jordan decomposition of W is of order 1. Hence, the minimal polynomial of W is $t(t-1) = 0$, or $T = 2$ in (3.38). This implies that (3.38) can be simplified as* $\lim_{\ell\to\infty} \boldsymbol{z}(\ell) = (\lim_{\ell\to\infty} W^\ell)\boldsymbol{z}(0) = W\boldsymbol{z}(0) = \boldsymbol{z}(1)$.

CHAPTER 4

Distributed MPC of Constrained Linear Systems with Time-Varying Terminal Sets

4.1 Introduction

This chapter considers the Distributed Model Predictive Control (DMPC) of a network of M systems, each of which is of the form

$$x^i(t+1) = \sum_{j \in \mathbb{Z}^M} A^{ij}x^j(t) + B^i u^i(t), \tag{4.1}$$

$$x^i(t) \in X^i,\ u^i(t) \in U^i,\ \ i = 1, \cdots, M,\ \ t \in \mathbb{Z}^+ \tag{4.2}$$

where $x^i \in \mathbb{R}^{n_i}$, $u^i \in \mathbb{R}^{m_i}$ are the state and input of the i^{th} system, $A^{ij} \in \mathbb{R}^{n_i \times n_j}$ is the system dynamics relating x^i and its coupled states of x^j, X^i and U^i are the corresponding state and control constraints respectively.

The study of DMPC of network system has received considerable attention recently and several approaches have been proposed for its solution, see [15, 36–40, 66]. One typical approach is to treat the $A^{ij}x^j$ where $i \neq j$ as a disturbance to the i system, see [47–49, 163]. Others (Chapter 7 and 11 of [40] and [164]) propose the use of the dual decomposition approach to handle the coupled dynamics. In these approaches, appropriate terminal constraints and terminal costs are needed; the choices of which are also active research areas. Clearly, and the most conservative choice of the terminal constraint is the origin [51, 63, 65]. Less conservative approaches include the use of a static ellipsoidal terminal sets [50, 52, 53] and a time-varying ellipsoidal set induced from a block diagonal Lyapunov matrix [66, 67]. In the latter case, each diagonal block of the Lyapunov matrix

also determines the terminal cost function of the corresponding system. In cases where M is small or moderate, restricting Lyapunov matrix to be block-diagonal can be restrictive. This work focuses on such systems and proposes an approach that differs from the previous in several distinctive ways: the Lyapunov matrix is non-diagonal (when it exists), the terminal set is time varying and moves within the maximal constraint admissible invariant set of the overall system. These features are possible under appropriate assumptions and additional computations. The implementation of the proposed approach is easiest when the network is fully connected or when a central collector is used. When this is not the case, additional linear programming (LP) problems are needed. Fortunately, these computations can be speeded up significantly using preprocessing.

This work does not address the algorithmic details for the numerical determination of the non block-diagonal Lyapunov matrix or the consensus algorithm of the DMPC problem as they are standard in the literature, see for example [40] and [164].

The rest of the chapter is organized as follows. This section ends with the notations needed, followed by the next section on the review of preliminary results of and additional notations for the DMPC problem. Section 4.3 discusses the choice of the distributed stage and terminal cost functions and the decomposition of the terminal set, including the solutions of a series of LP problems. The feasibility and stability of the overall system is shown in Section 4.4. Section 4.5 describes preprocessing steps that result in significant saving in the computations of the series of LP. Several numerical examples, including one for which a diagonal Lyapunov matrix does not exist, are provided Section 4.6. The last section concludes the work. All proofs are given in the appendices.

The notations used in this chapter are as follows. Non-negative and positive integer sets are indicated by $\mathbb{Z}_0^+$ and $\mathbb{Z}^+$ respectively with $\mathbb{Z}^M := \{1,2,\cdots,M\}$ and $\mathbb{Z}_L^M := \{L, L+1, \cdots, M\}, M \geq L$, $M, L \in \mathbb{Z}_0^+$. Similarly, $\mathbb{R}_0^+$ and $\mathbb{R}^+$ refer respectively to the sets of non-negative and positive real number. I_n is an $n \times n$ identity matrix and $int(\cdot)$ refers to the interior of a set. Given $\sigma > 0$ and $X \subset \mathbb{R}^n$ with $0 \in int(X)$, $\sigma X = \{\sigma x : x \in X\}$. The p-norm of $x \in \mathbb{R}^n$ is $\|x\|_p$ while $\|x\|_Q^2 = x^T Q x$ for $Q \succ 0$. For a square matrix Q, $Q \succ (\succeq) 0$ means Q is positive definite (semi-definite). Given a set of vectors, $c^i \in \mathbb{R}^n, i \in \mathbb{Z}^M$, the collection of vectors, $(c^1, c^2, \cdots, c^M)$ also refers to the stack vector of $[(c^1)^T (c^2)^T \cdots (c^n)^T]^T \in \mathbb{R}^{Mn}$ for notational simplicity. Let $\Omega \subset Z^M$ be an index set, $|\Omega|$ is its cardinality and $c^\Omega := \{c^i : i \in \Omega\}$ is the collection of vectors (or stacked vector) of c^i with $i \in$

Ω. Several representations of the states and controls are needed: $x^i(t)$, $u^i(t)$ refer to the state and control of the i^{th} system at time t; x_k^i, u_k^i are the k^{th} predicted state and control of the i^{th} system; $x=(x^1,x^2,\cdots,x^M)$, $u=(u^1,u^2,\cdots,u^M)$ are the collections of x^i and u^i over the M systems; boldface $\boldsymbol{x}^i=(x_0^i,x_1^i,\cdots,x_N^i)$, $\boldsymbol{u}^i=(u_0^i,u_1^i,\cdots,u_{N-1}^i)$ are the collections of the N predicted states and predicted controls over the horizon (of length N) for the i^{th} system; in situation where the reference to time is needed, x_k^i, u_k^i can be written as $x_{k|t}^i$ and $u_{k|t}^i$. Hence, $x_{0|t}^i=x^i(t)$ and $u_{0|t}^i=u^i(t)$. Additional notations are introduced as required in the text.

4.2 Preliminaries

Combining all the M systems of (4.1), the overall system is

$$x(t+1)=Ax(t)+Bu(t),\ t\in\mathbb{Z}_0^+ \tag{4.3}$$

$$x(t)\in X,\ \ u(t)\in U \tag{4.4}$$

where $x=(x^1,x^2,\cdots,x^M)\in\mathbb{R}^n$, $u=(u^1,u^2,\cdots,u^M)\in\mathbb{R}^m$ are the overall states and controls of the full system with $n=\sum_{i\in\mathbb{Z}^M}n_i$ and $m=\sum_{i\in\mathbb{Z}^M}m_i$. Also, $A\in\mathbb{R}^{n\times n}$ is a block matrix with its (i,j) block being $A^{ij}\in\mathbb{R}^{n_i\times n_j}$ and $B\in\mathbb{R}^{n\times m}$ is a block diagonal matrix with blocks $\{B^1,B^2,\cdots,B^M\}$ and $B^i\in\mathbb{R}^{n_i\times m_i}$. The constraint sets of X and U are

$$X:=X^1\times X^2\times\cdots\times X^M,\ \ U:=U^1\times U^2\times\cdots\times U^M \tag{4.5}$$

The connection among the systems is static and can be represented as a network with its structure captured in a set of pairwise indices,

$$D:=\{(i,j):A^{ij}\neq 0\}, \tag{4.6}$$

indicating adjacency relationship among the M systems. The connection among the M systems is assumed to be arbitrary and, hence, A is not symmetric. However, the scheme proposed in this work requires the states of system i be communicated to all its neighbors. For this reason, define the set

of neighbors of i, including i, as

$$\Omega_i := \{j : (i,j) \in D \text{ or } (j,i) \in D\} \cup \{i\}. \tag{4.7}$$

In general, $|\Omega_i| < M$. When $|\Omega_i| = M$ for all $i \in \mathbb{Z}^M$, the network is fully connected in the sense that each system is a neighbor of every other system. Several other variables, sets and states can be defined based on Ω_i and its complement:

$$n_{\Omega_i} := \sum_{j \in \Omega_i} n_j, \quad n_{\overline{\Omega}_i} := n - n_{\Omega_i}, \quad \overline{\Omega}_i := \mathbb{Z}^M \backslash \Omega_i, \tag{4.8}$$

$$x^{\Omega_i} := \{x^j : j \in \Omega_i\} \in \mathbb{R}^{n_{\Omega_i}}, \quad x^{\overline{\Omega}_i} := \{x^j : j \in \overline{\Omega}_i\} \in \mathbb{R}^{n_{\overline{\Omega}_i}}. \tag{4.9}$$

The variables u^i, x^i, x^{Ω_i} and $x^{\overline{\Omega}_i}$ can be extracted from u and x respectively from

$$u^i := F^i u, \;\; x^i := S^i x, \;\; x^{\Omega_i} = E^i x, \;\; x^{\overline{\Omega}_i} = \overline{E}^i x \tag{4.10}$$

where $F^i \in \{0,1\}^{m_i \times m}$, $S^i \in \{0,1\}^{n_i \times n}$, $E^i \in \{0,1\}^{n_{\Omega_i} \times n}$ and $\overline{E}^i \in \{0,1\}^{n_{\overline{\Omega}_i} \times n}$ are the appropriate selection matrices. From (4.10) and the fact that $[(E^i)^T (\overline{E}^i)^T]^T$ is a permutation matrix,

$$\begin{aligned} A^{ij} = S^i A (S^j)^T, \quad \begin{bmatrix} x^{\Omega_i} \\ x^{\overline{\Omega}_i} \end{bmatrix} = \begin{bmatrix} E^i \\ \overline{E}^i \end{bmatrix} x, \\ x = \begin{bmatrix} E^i \\ \overline{E}^i \end{bmatrix}^{-1} \begin{bmatrix} x^{\Omega_i} \\ x^{\overline{\Omega}_i} \end{bmatrix} := H^i x^{\Omega_i} + \overline{H}^i x^{\overline{\Omega}_i} \end{aligned} \tag{4.11}$$

where $E^i H^i = I_{n_{\Omega_i}}, \overline{E}^i \overline{H}^i = I_{n_{\overline{\Omega}_i}}, E^i \overline{H}^i = 0, \overline{E}^i H^i = 0$.

Assumptions of the system, needed in the sequel, are given below.

A3.1. The sets X^i and $U^i, i \in \mathbb{Z}^M$ are polytopes and contain the origin in their respective interiors.

A3.2. There is no delay or loss of information during communication between system i and all its neighbors.

A3.3. Matrices A and B are known to all systems.

A3.4. The set of M systems (or nodes) with edges defined by (4.7) forms an undirected and connected graph.

Both A3.1 and A3.2 are mild assumptions and are standard requirement in DMPC. Assumption A3.3 is needed as the models of the overall system are used to estimate the size of the terminal set at time t by system i. Assumption A3.3 may be hard to be satisfied when the network consists of heterogeneous systems. But in the typical case where most systems are similar or are members of only a few distinctly different classes of system, A3.3 is not a strong assumption. Assumption A3.4 defines the scope of the systems considered in this work. Suppose A3.4 is violated and the set of M systems has 2 or more connected components, then the approach described hereafter can be applied to them individually.

As a comparison for DMPC, a centralized MPC (CMPC) problem is needed. The CMPC assumes that the system given by (4.3) is solved via a single online finite horizon optimization problem of the form

$$V_N^*(x) = \min_{\boldsymbol{u}} V_N(\boldsymbol{x},\boldsymbol{u}) := \min_{\boldsymbol{u}} \sum_{k=0}^{N-1} l(x_k,u_k) + l_f(x_N) \tag{4.12a}$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k, x_k \in X, u_k \in U, x_N \in X_f, x_0 = x,\ k \in \mathbb{Z}_0^{N-1}$$

where N is the prediction horizon, $\boldsymbol{x} := \{x_0\ x_1\ \cdots\ x_N\}$, $\boldsymbol{u} := \{u_0\ u_1\ \cdots\ u_{N-1}\}$ are the predicted states and inputs respectively, X and U are those given by (4.5) and X_f is an appropriate terminal set. In this setting, CMPC is like a standard MPC problem without any constraints introduced by Ω_i and has the stage and the terminal costs being

$$l(x_k,u_k) = \|x_k\|_Q^2 + \|u_k\|_R^2,\ \ l_f(x_N) = \|x_N\|_P^2 \tag{4.13}$$

for some appropriate matrices $Q,R,P \succ 0$ and a scalar $\delta > 0$ that satisfy

$$(A+BK)^T P(A+BK) - P \preceq -(Q+K^T RK) - \delta I_n \tag{4.14}$$

for some stabilizing K. In addition, $l_f : X_f \to \mathbb{R}_0$ is defined on

$$X_f := \{x \in \mathbb{R}^n | Gx \leq \mathbf{1}_L\}. \tag{4.15}$$

where X_f is chosen to be the maximal constraint-admissible invariant set [29, 32] in the sense that $(A+BK)x \in X_f$ and $Kx \in U$ for all $x \in X_f$.

4.3 Main Results

This section presents the choices for the stage, terminal costs and the terminal constraints for the individual system in a DMPC setting. A few constraint sets are needed for K, Q, R and P due to the network structure introduced by (4.7). These are

$$P^{ij} := S^i P (S^j)^T, \;\; K^{ij} := F^i K (S^j)^T, \; Q^{ij} := S^i Q (S^j)^T, \tag{4.16}$$

$$\mathcal{P}_D := \{P \in \mathbb{R}^{n\times n} : P = P^T, P \succ 0, P^{ij} = 0 \text{ for all } j \notin \Omega_i, \; i \in \mathbb{Z}^M\}, \tag{4.17}$$

$$\mathcal{K}_D := \{K \in \mathbb{R}^{m\times n} : K^{ij} = 0 \text{ for all } j \notin \Omega_i, \; i \in \mathbb{Z}^M\}, \tag{4.18}$$

$$\mathcal{Q}_D := \{Q \in \mathbb{R}^{n\times n} : Q = Q^T, Q \succ 0, \; Q^{ij} = 0 \text{ for all } j \notin \Omega_i, \; i \in \mathbb{Z}^M\}, \tag{4.19}$$

where F^i, S^i are the selection matrices mentioned in (4.10).

4.3.1 Computations of P and K

Definition 4.1. *The network system of (4.3) with network connection given by D of (4.6) is network feedback stabilizable if there exists a $K \in \mathcal{K}_D$ such that $A+BK$ is Schur-stable.*

In the most general case, the search for a $K \in \mathcal{K}_D$ such that (4.3) is network feedback stabilizable is a difficult problem [23]. However, some special cases are solvable, using for example, the method of Alternate Direction Method of Multipliers (ADMM) [74] or others. For the problem at hand, (4.14) can be converted into a semidefinite constraint using Schur complement and by letting $W = P^{-1}$ and

$Y = KW$ as

$$\begin{bmatrix} W & WA^T + Y^T B^T & W & Y^T \\ AW + BY & W & \mathbf{0} & \mathbf{0} \\ W & \mathbf{0} & (Q + \delta I_n)^{-1} & \mathbf{0} \\ Y & \mathbf{0} & \mathbf{0} & R^{-1} \end{bmatrix} \succeq 0 \tag{4.20}$$

Hence, the search for $K \in \mathcal{K}_D$ becomes a bilinear semidefinite optimization problem in variables P, K, W, Y as

$$\min_{P,K,W,Y} -\log\det(W) + \alpha\|K\|_2 \tag{4.21a}$$

$$\text{s.t. } P \in \mathcal{P}_D,\ \ K \in \mathcal{K}_D;\ \ W \text{ and } Y \text{ satisfy } (4.20) \tag{4.21b}$$

$$WP = I_n,\ Y = KW \tag{4.21c}$$

where $\alpha > 0$ is a tradeoff parameter between the sizes of W and $\|K\|$. The use of $\|K\|_2$ in (4.21a) is to prevent values of P and K from becoming unacceptably large. The numerical solvability of such a problem using the method of ADMM is discussed in Section 9 of [74] and will not be discussed here, except that the initial values of P, K, W, Y are obtained from the solution of another semidefinite optimization problem that is similar to (4.21) but with $\alpha = 0$ and without $P \in \mathcal{P}_D$ and $K \in \mathcal{K}_D$ constraints.

4.3.2 Distributed Costs

The objective of the DMPC is to produce a performance as close as possible to that of CMPC. For this purpose, let $Q^{\Omega_i},\ P^{\Omega_i} \in \mathbb{R}^{n_{\Omega_i} \times n_{\Omega_i}}, R^i \in \mathbb{R}^{m_i \times m_i}$ and $Q^{\Omega_i},\ P^{\Omega_i}, R^i \succ 0$ be the weighting matrices for the i^{th} system and define the stage and terminal costs as

$$\begin{aligned} l^i(x^{\Omega_i}, u^i) &= (x^{\Omega_i})^T Q^{\Omega_i} x^{\Omega_i} + (u^i)^T R^i u^i, \\ l_f^i(x^{\Omega_i}) &= (x^{\Omega_i})^T P^{\Omega_i} x^{\Omega_i}, \quad i \in \mathbb{Z}^M \end{aligned} \tag{4.22}$$

where x^{Ω_i} is given by (4.9). It is easy to verify that these choices are related to (4.13) via (4.10) since

$$\begin{aligned} l(x,u) :&= \sum_{i\in\mathbb{Z}^M} l^i(x^{\Omega_i},u^i) = \sum_{i\in\mathbb{Z}^M} [x^T(E^i)^T Q^{\Omega_i} E^i x + (u^i)^T R^i u^i] \\ &= x^T(\sum_{i\in\mathbb{Z}^M}(E^i)^T Q^{\Omega_i} E^i)x + u^T \mathrm{diag}\{R^1,R^2,\cdots,R^M\}u \end{aligned} \tag{4.23}$$

$$l_f(x) :=\sum_{i\in\mathbb{Z}^M} l_f^i(x^{\Omega_i}) = x^T(\sum_{i\in\mathbb{Z}^M}(E^i)^T P^{\Omega_i} E^i)x \tag{4.24}$$

The connection to (4.13) is complete by letting

$$Q := \sum_{i\in\mathbb{Z}^M}(E^i)^T Q^{\Omega_i} E^i,\ \ R := \mathrm{diag}\{R^1,R^2,\cdots,R^M\} \tag{4.25}$$

and by decomposing the P obtained from (4.21) using the following convex semidefinite optimization problem:

$$\min_{\{P^{\Omega_i}\}_{i\in\mathbb{Z}^M}} \ -\sum_{i\in\mathbb{Z}^M} \log\det(P^{\Omega_i}) \tag{4.26a}$$

$$\text{s.t.} \sum_{i\in\mathbb{Z}^M}(E^i)^T P^{\Omega_i} E^i = P,\ \ P^{\Omega_i}\succ 0,\ \ P^{\Omega_i} = (P^{\Omega_i})^T \tag{4.26b}$$

Finally, the connection to the cost function (4.12a) of the CMPC is made by letting $V_N^i(\boldsymbol{x}^{\Omega_i},\boldsymbol{u}^i) := \sum_{k=0}^{N-1} l^i(x_k^{\Omega_i},u_k^i) + l_f^i(x_N^{\Omega_i})$ and noting that

$$\begin{aligned} V_N(\boldsymbol{x},\boldsymbol{u}) &= \sum_{i\in\mathbb{Z}^M} V_N^i(\boldsymbol{x}^{\Omega_i},\boldsymbol{u}^i) = \sum_{i\in\mathbb{Z}^M}[\sum_{k=0}^{N-1} l^i(x_k^{\Omega_i},u_k^i) + l_f^i(x_N^{\Omega_i})] \\ &= \sum_{k=0}^{N-1} \underbrace{\sum_{i\in\mathbb{Z}^M} l^i(x_k^{\Omega_i},u_k^i)}_{l(x_k,u_k)} + \underbrace{\sum_{i\in\mathbb{Z}^M} l_f^i(x_N^{\Omega_i})}_{l_f(x_N)} \end{aligned} \tag{4.27}$$

The problem of (4.26) is convex and, hence, $\{P^{\Omega_i} : i\in Z^M\}$ exists if a feasible solution is available. This condition, as well as the existence of solution of (4.21) is now assumed.

A3.5. System (4.3) is network feedback stabilizable, and solutions to problems (4.21) and (4.26) exist.

4.3.3 The Decoupled Terminal Set

The choice of X_f of (4.15) is useful as it results in a large domain of attraction. However, its decomposition to $X_f^1 \times X_f^2 \times \cdots \times X_f^M$ is not obvious. Furthermore, the decomposition should be such that X_f^i is defined by x^{Ω_i} only. For this purpose, this work uses a time-varying $X_f^i(t)$ that changes its size and location while satisfying

$$X_f^1(t) \times X_f^2(t) \times \cdots \times X_f^M(t) \subset X_f. \tag{4.28}$$

for all t. Specifically,

$$X_f^i(t) := X_f^i(c^i(t), r^i(t)) = \{x \in \mathbb{R}^{n_i} : \|x - c^i(t)\|_\infty \leq r^i(t)\} \tag{4.29}$$

is an ∞-norm ball of size $r^i(t) \geq 0$ centered at $c^i(t) \in \mathbb{R}^{n_i}$. (Other choices of $X_f^i(t)$ based on the $1-$ or $2-$ norm are possible but is not discussed to focus on the main idea.) While $X_f^i(t) \subset \mathbb{R}^{n_i}$, the value of $c^i(t)$ is determined by $x^{\Omega_i}(t)$ and others, as shown in the sequel. Related results of X_f^i and X_f are given below.

Lemma 4.1. *Given $c = (c^1, c^2, \cdots, c^M)$, $r = (r^1, r^2, \cdots, r^M)$ and X_f of (4.15) satisfying*

$$X_f^1(c^1, r^1) \times X_f^2(c^2, r^2) \times \cdots \times X_f^M(c^M, r^M) \subset X_f \tag{4.30}$$

where $X_f^i(c^i, r^i)$ is defined by (4.29). Let g_γ be the γ^{th} row of matrix $G \in \mathbb{R}^{L \times n}$ of (4.15). The following properties hold:

(i) Suppose $c \in X_f$. The largest ∞-norm ball, centered at c, such that (4.30) holds is when $r^1 = \cdots = r^M = \rho(c)$ where $\rho : X_f \to \mathbb{R}_0^+$ is a continuous concave function given by

$$\rho(c) = \min_{\gamma \in \mathbb{Z}^L} (1 - g_\gamma c)/\|g_\gamma\|_1, \tag{4.31}$$

(ii) Define $h_\gamma(r, c) := \sum_{i \in \mathbb{Z}^M} r^i \|g_\gamma (S^i)^T\|_1 - 1 + g_\gamma c$ where S^i is given by (4.10). Condition (4.30) holds if and only if

$$h_\gamma(r, c) \leq 0, \quad \forall \gamma \in \mathbb{Z}^L \tag{4.32}$$

(iii) Suppose c, r and $z = (z^1, z^2, \cdots, z^M)$ with $z^i \in X_f^i(c^i, r^i)$ for all $i \in \mathbb{Z}^M$ are given. Let

$$c^{i+} = \sum_{j \in \Omega_i} (A^{ij} + B^i K^{ij}) z^j \quad i \in \mathbb{Z}^M \tag{4.33}$$

Then $c^+ := (c^{1+}, c^{2+}, \cdots, c^{M+}) \in X_f$.

Property (i) of the above lemma shows a procedure to determine the maximal ∞-ball for a given $c \in X_f$ such that (4.28) and (4.29) hold. The necessary and sufficient condition of property (ii) is useful for the numerical determination of c and ρ that ensures (4.28) and (4.29). The last property of (iii) is useful to ensure recursive feasibility of DMPC. These properties are needed in the overall DMPC scheme described in the next three subsections.

4.3.4 Online DMPC Problem

Suppose $\{X_f^i(t) : i \in \mathbb{Z}^M\}$ are known such that (4.28) and (4.29) are satisfied, the collective online DMPC optimization problem is

$$V_N^*(x(t)) = \min_{\{\boldsymbol{x}^{\Omega_i}, \boldsymbol{u}^i\}_{i \in \mathbb{Z}^M}, \, \bar{\boldsymbol{x}}} \sum_{i \in \mathbb{Z}^M} V_N^i(\boldsymbol{x}^{\Omega_i}, \boldsymbol{u}^i) \tag{4.34a}$$

$$s.t. \; x_{k+1}^i = \sum_{j \in \Omega_i} A^{ij} x_k^j + B^i u_k^i, \; x_k^i \in X^i, \; u_k^i \in U^i, \; x_N^i \in X_f^i(t), \tag{4.34b}$$

$$x_0^i = x^i(t), \; x_k^{\Omega_i} = E^i \bar{x}_k, \; x_N^{\Omega_i} = E^i \bar{x}_N, \; k \in \mathbb{Z}_0^{N-1}, \;\; i \in Z^M \tag{4.34c}$$

where $\bar{\boldsymbol{x}} = (\bar{x}_0, \; \bar{x}_1, \cdots, \bar{x}_N)$ with $\bar{x}_k = (\bar{x}_k^1, \cdots, \bar{x}_k^M)$ is the overall predicted state and a global consensus variable. The individual predicted state of $x_k^{\Omega_i}$ is extracted from $\bar{\boldsymbol{x}}$ via (4.34c) and $\boldsymbol{x}^{\Omega_i} = \{x_0^{\Omega_i}, x_1^{\Omega_i}, \cdots, x_N^{\Omega_i}\}$ is the collection of the N predicted states for the neighbors of the i^{th} system and itself. The numerical solution to this problem via distributed optimization has been suggested in several works, see for example sections 5 and 7 of [74] and others [165, 166]. In particular, the alternative direction method of multipliers(ADMM) is quite popular, see Chapter 7 of [40] and [164], and this work follows the same approach. Problem (4.34) is solved distributively via iterative computations using ADMM by having each agent solving its local problem in parallel. Implicit assumption here is that the ADMM algorithm has converged within the sampling period. Suppose the optimal

states and controls for system i at time t are

$$\boldsymbol{x}^i(t) := \{x^i_{0|t}\ x^i_{1|t}\ \cdots\ x^i_{N|t}\},\ \boldsymbol{u}^i(t) := \{u^i_{0|t}\ u^i_{1|t}\ \cdots\ u^i_{N-1|t}\} \tag{4.35}$$

The control law for system i, applied at time t, is

$$u^i(t) = u^i_{0|t}. \tag{4.36}$$

The next important step for the overall DMPC approach is to compute $(c^i(t+1), r^i(t+1))$ of $X^i_f(t+1)$ such that (4.28) remains true at time $t+1$.

4.3.5 Update of c^i

The computation of $c^i(t+1)$ is simple. For each system i, let $z^i(t) = x^i_{N|t}$. The value of $c^i(t+1)$ is then obtained from (4.33) as

$$c^i(t+1) = \sum_{j\in\Omega_i} (A^{ij} + B^i K^{ij}) x^j_{N|t}. \tag{4.37}$$

Consider the case where the network is fully connected or a central collector is available. System i computes $c^i(t+1)$ based on the above and sends its value to the central collector which will then distribute it to all other systems (or broadcast to all neighbors in the case of fully connected network). Hence, $c^i(t+1)$ for all i is known exactly. In the case where a central collector is not present or the network is not fully connected, $c^i(t+1)$ is still updated using (4.37). However, since system i does not have the knowledge of the full $c(t+1)$, a more conservative $r^i(t+1)$ is needed to ensure satisfaction of (4.28).

4.3.6 Update of r^i

Like the previous section, the computation of $r^i(t+1)$ is considered in two different cases: the first where there is a central collector (or fully connected network) and the second where there is not (or not fully connected). In the first case, $r^i(t+1)$ is set to be $\rho(c(t+1))$ for all i using property (i) of Lemma 4.1. Recall that $c(t+1)$ is known to every system for this case. Obviously, it follows that

(4.28) holds for $X_f^i(c^i(t+1), r^i(t+1))$.

When there is no central collector, $r^i(t+1)$ should be set at a value such that (4.28) holds for all possible (but unknown) values of $c(t+1)$. Recall that system i computes $c^i(t+1)$ from (4.37); it then sends $(c^i(t+1), c^i(t), r^i(t), z^i(t))$ to all its neighbors and receive the same from them. With these exchanges,

$$\begin{aligned}\omega^i(t) &:= \left(c^{\Omega_i}(t+1), z^{\Omega_i}(t), c^{\Omega_i}(t), r^{\Omega_i}(t)\right) \\ &= \left(c^{\Omega_i}(t+1), x^{\Omega_i}_{N|t}, c^{\Omega_i}(t), r^{\Omega_i}(t)\right)\end{aligned} \tag{4.38}$$

is now available to system i, or stated differently, $\overline{\omega}^i(t) := \{c^{\overline{\Omega}_i}(t+1), z^{\overline{\Omega}_i}(t), c^{\overline{\Omega}_i}(t), r^{\overline{\Omega}_i}(t)\}$ is NOT available to system i. Hence, the value of $r^i(t+1)$ is set to be the smallest over all admissible $\overline{\omega}^i$. This can be done by setting the value of $r^i(t+1)$ as the solution of the following optimization problem, $O(\omega^i)$, over the unknown variables $\overline{\omega}^i$:

$$\min_{\hat{c}^{\overline{\Omega}_i+}, \hat{z}^{\overline{\Omega}_i}, \hat{c}^{\overline{\Omega}_i}, \hat{r}^{\overline{\Omega}_i}} \rho(c^{\Omega_i+}, \hat{c}^{\overline{\Omega}_i+}) \tag{4.39a}$$

$$s.t. \quad c^{\Omega_i+} = E^i A_K (H^i z^{\Omega_i} + \overline{H}^i \hat{z}^{\overline{\Omega}_i}) \tag{4.39b}$$

$$\hat{c}^{\overline{\Omega}_i+} = \overline{E}^i A_K (H^i z^{\Omega_i} + \overline{H}^i \hat{z}^{\overline{\Omega}_i}) \tag{4.39c}$$

$$\|\hat{z}^j - \hat{c}^j\|_\infty \le \hat{r}^j, \quad j \in \overline{\Omega}_i \tag{4.39d}$$

$$h_\gamma((r^{\Omega_i}(t), \hat{r}^{\overline{\Omega}_i}), (c^{\Omega_i}(t), \hat{c}^{\overline{\Omega}_i})) \le 0, \ \ \forall \gamma \in \mathbb{Z}^L \tag{4.39e}$$

where $\rho(\cdot)$ of (4.39a) is the function given by (4.31) since $c^+ = (c^{\Omega_i+}, \hat{c}^{\overline{\Omega}_i+})$, constraints (4.39b) (and similarly (4.39c)) follows from $c^{\Omega_i}(t+1) = E^i c(t+1) = E^i A_K z(t) = E^i A_K (H^i z^{\Omega_i}(t) + \overline{H}^i z^{\overline{\Omega}_i}(t))$ from (4.10), $z(t) = H^i z^{\Omega_i}(t) + \overline{H}^i z^{\overline{\Omega}_i}(t)$ from (4.11), and $A_K := A + BK$. Constraint (4.39d) arises from $\hat{z}^j \in X_f^j(\hat{c}^j, \hat{r}^j)$ following the condition of property (iii) of Lemma 4.1 while (4.39e) is from property (ii) of Lemma 4.1 to ensure the satisfaction of (4.30) for $(r^{\Omega_i}(t), \hat{r}^{\overline{\Omega}_i})$ and $(c^{\Omega_i}(t), \hat{c}^{\overline{\Omega}_i})$. The objective function (4.39a) is a concave function of $\hat{c}^{\overline{\Omega}_i+}$. Hence, the computations of $O(\omega^i)$ is difficult. However, there are very effective ways to overcome this difficulty and they are discussed in section 4.5. Using them, $r^i(t+1)$ and $O(\omega^i)$ has the following properties:

Lemma 4.2. *Suppose $\{X_f^i(t) : i \in \mathbb{Z}^M\}$ are known such that (4.28) is satisfied, the optimal solution $\{(\boldsymbol{x}^i(t), \boldsymbol{u}^i(t)), i \in \mathbb{Z}^M\}$ is obtained from (4.34) and $c(t+1) = (c^1(t+1), c^2(t+1), \cdots, c^M(t+1))$*

with $c^i(t+1)$ obtained from (4.37). Let $r^i(t+1)$ be the optimal solution of $O(\omega^i)$ for all $i \in \mathbb{Z}^M$. Then $r^i(t+1) \leq \rho(c(t+1))$ for all $i \in \mathbb{Z}^M$ and

$$X_f^1(t+1) \times X_f^2(t+1) \times \cdots \times X_f^M(t+1) \subset X_f \tag{4.40}$$

The above describes the overall scheme of the DMPC except for the initial values of $X_f^i(c^i(0), r^i(0))$ for all $i \in \mathbb{Z}^M$. In the absence of additional information, the initial values can be chosen as $c^i(0) = 0$ and $r^i(0) = \rho(c(0))$ for all $i \in \mathbb{Z}^M$.

4.4 Feasibility and Stability of DMPC

The results presented in this section assume implicitly that (A3.1)-(A3.5) hold. For example, they are to ensure the existence of X_f, $P \in \mathcal{P}_D, K \in \mathcal{K}_D$ and $0 \in int(X_f)$.

Theorem 4.1. *Suppose there exists a feasible solution of the DMPC problem of (4.34) at time t. Let $c^i(t+1)$ be updated according to (4.37); $r^i(t+1)$ equals $\rho(c(t+1))$ (when central collector is used or network is fully connected) or equals the solution of $O(\omega^i)$ of (4.39) in the absence of a central collector (or network is not fully connected). System (4.3) with $u(t) = (u^1(t), u^2(t), \cdots, u^m(t))$ where $u^i(t)$ is given by (4.36) has the following results: (i) there exists a feasible solution to the DMPC problem at time $t+1$; (ii) $\lim_{t\to\infty} x(t) = 0$ and $\lim_{t\to\infty} c(t) = 0$.*

The above asymptotic stability result can be strengthened in the presence of a central collector or a fully connected network.

Theorem 4.2. *Suppose $c^i(t+1)$ is updated according to (4.37) and $r^i(t+1)$ is $\rho(c(t+1))$. Then, system (4.3) with $u(t) = (u^1(t), u^2(t), \cdots, u^m(t))$ where $u^i(t)$ given by (4.36) is exponentially stable.*

Remark 4.1. *When there is no central collector or network is not fully connected, additional assumption is needed to guarantee exponential convergence. One such assumption, similar in nature to (4.50) in 4.D, is that there exist $\varepsilon > 0$ and t_ε such that $r^i(t) \geq \varepsilon$ for all $t \geq t_\varepsilon$ and for all $i \in \mathbb{Z}^M$. While this condition may be hard to verify, numerical simulation shows that it typically holds, see Example I in Section 4.6.*

4.5 Preprocessing to Speed up the LP Computations

This section is needed only when the network is not fully connected and there is no central collector. The difficulty of computing $O(\omega^i)$ can be circumvented by noting that $\rho(c) = \min_{\gamma \in \mathbb{Z}^L}(1 - g_\gamma c)/\|g_\gamma\|_1$ from (4.31). Hence, the solution of $O(\omega^i)$ is given by the minimal objective value over L linear programming (LP) problems where each LP corresponds to one choice of $\gamma \in \mathbb{Z}^L$ and having the form

$$\min_{\hat{z}^{\overline{\Omega}_i}, \hat{c}^{\overline{\Omega}_i}, \hat{r}^{\overline{\Omega}_i}} \{\frac{1 - g_\gamma \hat{c}^+}{\|g_\gamma\|_1} : (4.39b) - (4.39e)\} \tag{4.41}$$

with $\hat{c}^+ = (c^{\Omega_i +}, \hat{c}^{\overline{\Omega}_i +})$ and $\hat{c}^{\overline{\Omega}_i +}$ is given by (4.39c). Since each of the L LP has the same set of constraints, this feature can be exploited to speed up the search for the minimal over the L LPs. The simplest scheme is to use the optimal solution (including the optimizer, the active constraints and the inverse of the matrix of the active constraints) of one LP as the starting feasible point for the next LP. This scheme, Scheme 1, avoids the computation for an initial feasible point under the Simplex or Active-set-based LP solvers. The speed up is more significant if some preprocessing is done. Scheme 2 includes the idea of Scheme 1 and uses a specific sequence to solve the L LPs. For this purpose, let $\hat{g}_\gamma := \frac{g_\gamma}{\|g_\gamma\|_2}$ and define the neighbors of the i^{th} LP as

$$N_i^g := \{j : \hat{g}_j^T \hat{g}_i \geq \beta\},\ i \in \mathbb{Z}^L$$

for some threshold value $\beta < 1$. This adjacency information together with the L LPs as nodes (hereafter LP and nodes are used interchangeably) forms an undirected graph. Depending on the choice of β used, the graph may have 1 or more connected components. The algorithms to search for the number of connected components are a well-known problem in Graph theory [167]. In addition, the path from one given starting node to every other node in the same connected component of the graph can be expressed as a hierarchical tree [168]. Suppose there are N_L components in the graph and each is represented as a tree with the top node being an arbitrary LP, see Figure 4.1. The computations of the L LPs proceed from the starting node of any one of the trees. When one LP is computed, the next LP to be solved is based on a breadth-first or depth-first searching order of the hierarchical tree. In addition, the optimal solutions of ALL solved LPs are considered to determine

the starting solution of the next LP. Specifically, suppose the LP of (4.41) has been solved for all $\gamma \in \Pi \subset \mathbb{Z}^L$ with each optimal value given by $\frac{1-g_\gamma c_\gamma}{\|g_\gamma\|_1}$. Let the next node following the hierarchical tree has index ℓ. The starting feasible solution for the ℓ LP is obtained from the solution of the LP with index given by

$$argmin\{\frac{1-g_\ell^T c_\gamma}{\|g_\ell^T\|_1} : \gamma \in \Pi\}$$

Hence, such a scheme requires the storage of the optimal solutions and related information of all solved LPs but it speeds up the overall computational time for the L LP significantly, see the next section for computational details.

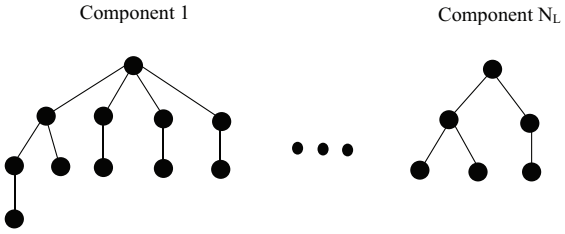


Figure 4.1: The hierarchical trees of the LPs

4.6 Numerical Results

This section serves two purposes: to demonstrate the performance of the DMPC approach and the effectiveness of the speed up scheme of section 4.5. The first two examples (Examples I and II) are four-system network with A^{ii},A^{ij}, B^i being

$$A^{ii} = \begin{bmatrix} 1.1 & 1 \\ 0 & 1.3 \end{bmatrix}, A^{ij} = \begin{bmatrix} 0 & 0.1 \\ 0.2 & 0.2 \end{bmatrix}, B^i = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, j \neq i$$

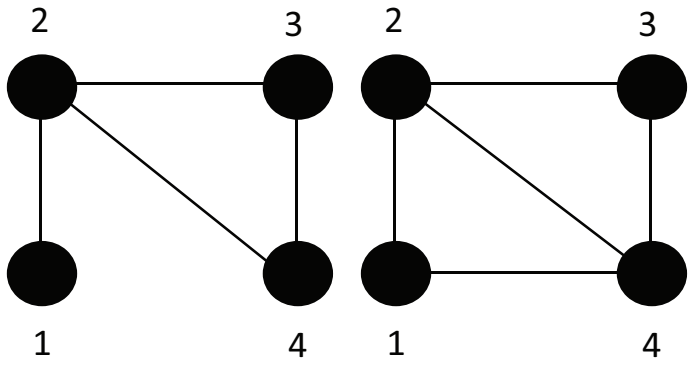


Figure 4.2: The networked system examples

and network configurations of Figures 4.2a and 4.2b. The constraints are $X^i := \{x^i \in \mathbb{R}^{n_i} |\ \|x^i\|_\infty \leq 10\}$, $U^i := \{u^i \in \mathbb{R}^{m_i} |\ \|u^i\|_\infty \leq 1\}$ with $Q^{\Omega_i} = I$ and $R^i = 0.01I$, the values of $K \in \mathcal{K}_D$ and $P \in \mathcal{P}_D$, obtained from the solution of (4.21) with $\delta = 0.01, \alpha = 2$ and $x^i(0)$ are

$$K = \begin{bmatrix} -0.6939 & -1.056 & 0.0147 & -0.1572 & 0 & 0 & 0 & 0 \\ -0.1418 & -0.1820 & -0.7452 & -1.113 & 0.0671 & -0.0888 & 0.0671 & -0.0888 \\ 0 & 0 & -0.0136 & -0.1555 & -0.7312 & -1.084 & 0.0072 & -0.1299 \\ 0 & 0 & -0.0136 & -0.1555 & 0.0072 & -0.1298 & -0.7312 & -1.084 \end{bmatrix},$$

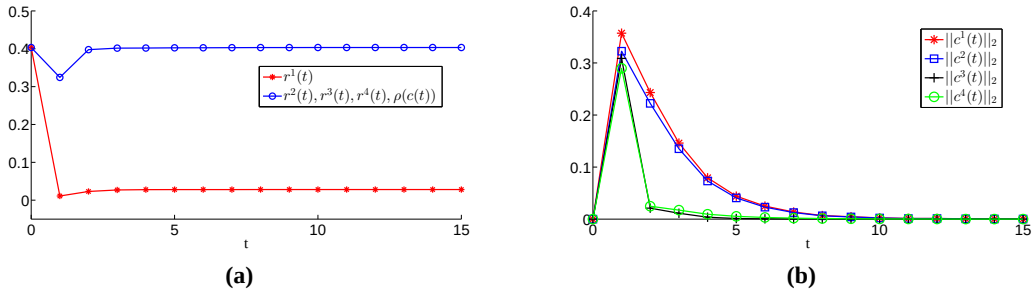
$$P = \begin{bmatrix} 4.294 & -0.3814 & -1.041 & 0.0087 & 0 & 0 & 0 & 0 \\ -0.3814 & 2.267 & -0.2486 & 0.1484 & 0 & 0 & 0 & 0 \\ -1.041 & -0.2486 & 8.062 & -0.8835 & -1.171 & -0.1419 & -1.171 & -0.1419 \\ 0.0087 & 0.1484 & -0.8835 & 4.362 & -0.0152 & 0.1890 & -0.0152 & 0.1890 \\ 0 & 0 & -1.171 & -0.0152 & 6.306 & -0.6004 & -0.8061 & 0.0229 \\ 0 & 0 & -0.1419 & 0.1890 & -0.6004 & 3.340 & 0.0229 & 0.2237 \\ 0 & 0 & -1.171 & -0.0152 & -0.8061 & 0.0229 & 6.306 & -0.6004 \\ 0 & 0 & -0.1419 & 0.1890 & 0.0229 & 0.2237 & -0.6004 & 3.340 \end{bmatrix},$$

$$x^1(0) = \begin{bmatrix} -2.155 \\ 1.893 \end{bmatrix}, x^2(0) = \begin{bmatrix} -0.4146 \\ 2.317 \end{bmatrix}, x^3(0) = \begin{bmatrix} 1.395 \\ 1.112 \end{bmatrix}, x^4(0) = \begin{bmatrix} -0.9137 \\ -1.983 \end{bmatrix}$$

with P^{Ω_i} obtained via (4.26). The rest of the parameters are: $N = 10$, $c^i(0) = 0, r^i(0) = 0.4033$ for $i = 1,2,3,4$. The values of $r^i(t)$ and $\|c^i(t)\|_2$ against t are shown in Figure 4.3a and 4.3b respectively. Figure 4.3b also includes values of $\rho(c(t))$ as a comparison. As shown, $r^2(t) = r^3(t) = r^4(t)$ are much larger than $r^1(t)$, a fact resulting from system 1 having fewer neighbors than others. To better apprecaite this fact, the approach is applied to the network of Figure 4.2b. Note that the Q,R,P,K and X_f are the same as before with $A^{41} = A^{14} = 0$ but with 1 and 4 being neighbors of each other. The values of $r^1(t)$, $r^2(t)$, $r^3(t)$, $r^4(t)$ and $\rho(c(t))$ of the second network are shown in Figure 4.4. The value of $r^1(t)$ is now equal to $\rho(c(t))$ for all t, verifying the effect of connectivity on the values of $r^i(t)$. It also shows that the proposed approach is better suited for networks that are well connected.

A comparison of the performances of DMPC and CMPC over several choices of $x(0)$ is given in Table 4.1. Obviously, both approaches use the same values of Q, R, K and P. The performance

is measured by $J^{\infty}(x(0)) = \sum_{t=0}^{\infty} x(t)^T Q x(t) + u(t)^T R u(t)$ with $x(t), u(t)$ being the true state and control of the system. This measure is approximated by $\sum_{t=0}^{T} \|x(t)\|_Q^2 + \|u(t)\|_R^2$ for some large T. The performances are denoted by $J^{\infty}_{DMPC}(x(0))$ and $J^{\infty}_{CMPC}(x(0))$ respectively and its relative difference, $\Delta_J(x) = \frac{J^{\infty}_{DMPC}(x) - J^{\infty}_{CMPC}(x)}{J^{\infty}_{CMPC}(x)} \times 100\%$. As can be seen in Table 4.1, $J^{\infty}_{DMPC}(x(0))$ is on average 5% higher than $J^{\infty}_{CMPC}(x(0))$, an expected result due to the use of a more restricted X_f^i compared to X_f.



(a) **(b)**

Figure 4.3: The individual terminal sets

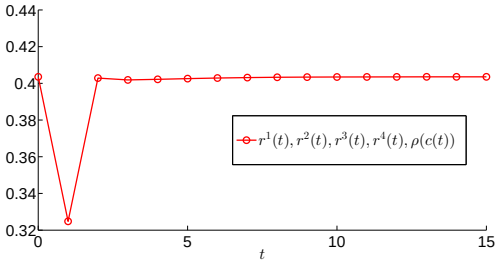


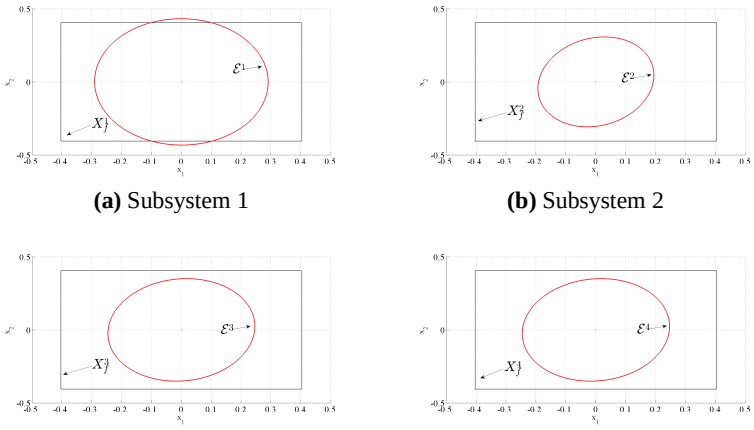
Figure 4.4: $r^1(t), r^2(t), r^3(t), r^4(t)$, and $\rho(c(t))$ of the second network

x^1	x^2	x^3	x^4	$J^{\infty}_{DMPC}(x)$	$J^{\infty}_{CMPC}(x)$	$\Delta_J(x)(\%)$
$\begin{bmatrix} -2.155 \\ 1.893 \end{bmatrix}$	$\begin{bmatrix} -0.4146 \\ 2.317 \end{bmatrix}$	$\begin{bmatrix} 1.395 \\ 1.112 \end{bmatrix}$	$\begin{bmatrix} -0.9137 \\ -1.983 \end{bmatrix}$	601.3	580.6	3.57
$\begin{bmatrix} 0.5204 \\ -2.1969 \end{bmatrix}$	$\begin{bmatrix} 1.3889 \\ 1.8441 \end{bmatrix}$	$\begin{bmatrix} 1.1516 \\ -0.6396 \end{bmatrix}$	$\begin{bmatrix} 3.1703 \\ 0.7843 \end{bmatrix}$	1059	1046	1.26
$\begin{bmatrix} 1.4281 \\ 2.3855 \end{bmatrix}$	$\begin{bmatrix} -1.6044 \\ 0.7639 \end{bmatrix}$	$\begin{bmatrix} 2.1066 \\ 2.5045 \end{bmatrix}$	$\begin{bmatrix} -0.5404 \\ -2.5454 \end{bmatrix}$	1258	1228	2.38
$\begin{bmatrix} -2.3817 \\ -3.3901 \end{bmatrix}$	$\begin{bmatrix} 2.8344 \\ 1.6336 \end{bmatrix}$	$\begin{bmatrix} 2.4512 \\ -1.7680 \end{bmatrix}$	$\begin{bmatrix} 0.2687 \\ 0.6759 \end{bmatrix}$	1401	1247	12.4
$\begin{bmatrix} -1.1407 \\ 1.3046 \end{bmatrix}$	$\begin{bmatrix} -1.9217 \\ 2.3264 \end{bmatrix}$	$\begin{bmatrix} -3.0906 \\ 0.6680 \end{bmatrix}$	$\begin{bmatrix} 3.8166 \\ -3.6878 \end{bmatrix}$	1105	1096	0.794
$\begin{bmatrix} -2.5408 \\ -1.5982 \end{bmatrix}$	$\begin{bmatrix} -1.8774 \\ -0.9442 \end{bmatrix}$	$\begin{bmatrix} 1.1081 \\ -0.4147 \end{bmatrix}$	$\begin{bmatrix} 0.5736 \\ 2.5888 \end{bmatrix}$	763.2	718.7	6.20

Table 4.1: The cost difference of DMPC and CMPC over various choices of the initial state $x(0)$.

Several works in the literature [37, 39, 66] use a block-diagonal P matrix for the convenience of a naturally decoupled terminal cost. For example, the i^{th} terminal set in the work of [66] is of the

form $\mathcal{E}^i := \{x^i : \|x^i\|^2_{P^{ii}} \leq r^i\}$ with r^i is determined by a separate semidefinite programming problem. The choice of a block-diagonal P is obviously conservative since it may not satisfy the Lyapunov equation of (4.14), see Example II. Even if it exists, the performance of DMPC may be compromised due to the restricted choices of Q and R. The use of block-diagonal P typically results in smaller terminal sets, compared to the proposed approach. A comparison of the terminal sets at $t = 0$ using the approach of [66] and the proposed approach is shown in Figure 4.5.



(a) Subsystem 1 **(b)** Subsystem 2

(c) Subsystem 3 **(d)** Subsystem 4

Figure 4.5: The comparison of terminal sets: $\mathcal{E}^i$ refers to terminal set induced by a diagonal P while X_f^i is that given by the proposed approach.

Example II has the network connection of Figure 4.2b and the A, B matrices as given below. Note that no block-diagonal P matrix exists that satisfies (4.14) for this example when $Q^{\Omega_i} = I$, and $R^i = 0.01I$. Also, the values of K and P obtained from (4.21) with $\alpha = 2$ are also indicated below.

$$A^{11} = \begin{bmatrix} 0.8 & -0.25 \\ -0.3 & 0.8 \end{bmatrix}, A^{12} = \begin{bmatrix} -0.15 & 0.5 \\ 0.2 & -0.5 \end{bmatrix}, A^{14} = \begin{bmatrix} 0.4 & -0.5 \\ -0.05 & -0.1 \end{bmatrix}, B^1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$A^{21} = \begin{bmatrix} -0.1 & 0.3 \\ 0.5 & 0.15 \end{bmatrix}, A^{22} = \begin{bmatrix} 0.9 & -0.4 \\ 0.1 & 0.9 \end{bmatrix}, A^{23} = \begin{bmatrix} -0.35 & -0.2 \\ 0.35 & -0.01 \end{bmatrix}, A^{24} = \begin{bmatrix} -0.15 & 0.25 \\ -0.5 & -0.2 \end{bmatrix}, B^2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$A^{32} = \begin{bmatrix} -0.4 & -0.4 \\ -0.2 & -0.2 \end{bmatrix}, A^{33} = \begin{bmatrix} 0.7 & -0.1 \\ -0.4 & 0.75 \end{bmatrix}, A^{34} = \begin{bmatrix} -0.3 & -0.3 \\ 0.3 & -0.1 \end{bmatrix}, B^3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$A^{41} = \begin{bmatrix} 0.3 & 0.4 \\ -0.03 & 0.3 \end{bmatrix}, A^{42} = \begin{bmatrix} 0.1 & 0.2 \\ 0.4 & 0.5 \end{bmatrix}, A^{43} = \begin{bmatrix} -0.15 & 0.5 \\ 0.1 & 0.5 \end{bmatrix}, A^{44} = \begin{bmatrix} 1.1 & 0.4 \\ -0.2 & 1.1 \end{bmatrix}, B^4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$K = \begin{bmatrix} -1.051 & 0.3137 & 0.3854 & -0.6973 & 0 & 0 & -0.8072 & 0.8598 \\ -0.7607 & 0.2718 & -0.7058 & -0.6800 & 0.7344 & -0.07826 & -0.005887 & -0.06570 \\ 0 & 0 & 0.9479 & -0.3783 & -1.134 & -0.4131 & 0.3263 & 0.5520 \\ 0.3240 & -0.4981 & -0.0004079 & -0.4808 & -0.2830 & -0.9680 & -0.5707 & -0.6772 \end{bmatrix}$$

$$P = \begin{bmatrix} 116.5 & -49.39 & -60.83 & 87.10 & 0 & 0 & 76.97 & -74.88 \\ -49.39 & 26.89 & 25.42 & -39.25 & 0 & 0 & -31.79 & 31.26 \\ -60.83 & 25.42 & 102.8 & -93.58 & -89.61 & -13.21 & -32.56 & 77.27 \\ 87.10 & -39.25 & -93.58 & 108.6 & 64.45 & 11.02 & 48.94 & -82.06 \\ 0 & 0 & -89.61 & 64.45 & 129.5 & 14.17 & -17.82 & -49.30 \\ 0 & 0 & -13.21 & 11.02 & 14.17 & 9.716 & 1.323 & -7.280 \\ 76.97 & -31.79 & -32.56 & 48.94 & -17.82 & 1.323 & 67.75 & -48.59 \\ -74.88 & 31.26 & 77.27 & -82.06 & -49.30 & -7.280 & -48.59 & 75.28 \end{bmatrix}$$

The rest of this section illustrates the computations of $O(\omega^i)$ via two examples (Examples III and IV). Values of n, m, L, M and other parameters are given in Table 4.2. The variables $n_v^1, n_{\leq}^1, n_{=}^1$ refer to number of variables, number of inequality and equality constraints respectively for $O(\omega^1)$. Example

III uses the same A^{ij} as those in Example I while those of Example IV are:

$$A^{ii} = \begin{bmatrix} -0.2568 & 0.4331 & 0 & 0.4071 \\ -0.1495 & 0.02262 & 0.4013 & 0 \\ 0 & 0 & 0.9654 & -0.7708 \\ 0 & 0 & 0.6133 & 0.3297 \end{bmatrix}, B^i = \begin{bmatrix} 1 \\ \\ 1 \end{bmatrix}$$

$$A^{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -0.1780 & 0 & 0 & 0 \\ 0 & 0.2308 & 0 & 0 \\ 0 & 0 & 0 & 0.2311 \end{bmatrix}, j \neq i$$

Both examples have the same constraint set as Example I with $Q^{\Omega_i} = I$ and $R^i = 0.01I$. The adjacency matrices of the Examples III and IV are the same and are denoted by $\{a_{ij}\}$ with $a_{ij} = 0$ for all (i, j) except for

$$\text{If } 1 < i < M,\ a_{ij} = 1 \text{ for } j = i-1, i, i+1;$$

$$\text{If } i = 1,\ a_{ij} = 1, \text{ for } j = i, i+1;$$

$$\text{If } i = M,\ a_{ij} = 1, \text{ for } j = i-1, i.$$

For comparison purposes, the CPU times of $O(\omega^1)$ at time $t = 0$ for the various schemes are shown. Table 4.3 shows T_s, T_L, T_L^1, T_L^2 and their ratios for $O(\omega^1)$. Specifically, these variables are the CPU times of a single LP (starting from scratch), L independently LPs, L LPs using Scheme 1, and L LPs using Scheme 2 respectively. In Scheme 2, the choice of the next LP to solve is based on the breadth-first search method (see Section 4.5). The LPs are solved by Clp [169]. As shown, the timings of scheme 1 are about 21 - 42% time of the T_L while Scheme 2 is about 8 - 12% of T_L. In general, scheme 2 is about $4-20$ times the time needed for a single LP. All numerical experiments are done on a Windows 7 PC with an Intel Core i5-3570 processor and 8GB memory.

Example	n	m	M	L	β	N_L	n_v^1	$n_<^1$	$n_=^1$
III	20	10	10	44	0.2	4	40	76	2
IV	40	10	10	136	0.3	36	72	200	4

Table 4.2: Parameters of Examples III and IV

Example	T_s(s)	T_L/T_s (s)	T_L^1/T_s	T_L^2/T_s
III	9.017×10^{-4}	44	9.272	3.853
IV	2.532×10^{-3}	136	57.32	17.467

Table 4.3: CPU times of $O(\omega^1)$ using different schemes

4.7 Conclusions

This work proposes a less conservative approach to the DMPC problem using two distinct features: the terminal cost function that depends on a Lyapunov matrix that conforms to the structural constraint imposed by the network; and a terminal set that is obtained from the maximal constraint admissible invariant set of the overall system. More exactly, the approach determines a time-varying terminal set that moves within the maximal constraint admissible invariant set, changing in both size and location at each time. When the network is fully connected or a central collector is used, the terminal set can be easily computed and local exponential stability is achieved. If not, the computations of the terminal set require a series of linear programming (LP) problems; the computations of which can be speeded up, via a preprocessing step, so that the approach is suitable for online computations.

4.A Proof of Lemma 4.1:

(i) The vertices of the set $\{x \in \mathbb{R}^n : \|x-c\|_\infty \leq \beta\}$ are $\{c \pm \beta v_i : \ i = 1,2,3,\cdots,2^n\}$ where $v_i \in \{-1,1\}^{n\times 1}$. Hence, this set lies inside X_f if and only if $G(c \pm \beta v_i) \leq \mathbf{1}_L$ or $\beta G(\pm v_i) \leq \mathbf{1}_L - Gc$ for all $i = 1,2,3,\cdots,2^n$. Since $g_\gamma v_i \leq \|g_\gamma\|_1$ for all $\pm v_i$ where g_γ is the γ^{th} row of G, and there exists some v_i such that $g_\gamma v_i = \|g_\gamma\|_1$, it follows that

$$\beta \leq (1-g_\gamma c)/\|g_\gamma\|_1 \ \forall \ \gamma \in \mathbb{Z}^L \tag{4.42}$$

which implies that the maximum value of β is $\min_{\gamma\in\mathbb{Z}^L}(1-g_\gamma c)/\|g_\gamma\|_1$. Hence, setting $r^1 = r^2 = \cdots = r^M = \beta$ and the fact that $\{x \in \mathbb{R}^n : \|x-c\|_\infty \leq \beta\} = X_f^1(c^1,r^1) \times \cdots \times X_f^M(c^M,r^M)$ implies that (4.28) and (4.29) hold. That $\rho(c)$ is a concave function follows from the fact that it is the point-wise minimum of L affine functions of c.

(ii) Like (i), $X_f^i(c^i,r^i) \subset X_f$ if and only if all its vertices are inside X_f. The vertices of $X_f^i(c^i,r^i)$ are $\{c^i \pm r^i v_{j_i}, \ j_i = 1,2,3,\cdots,2^{n_i}$ where $v_{j_i} \in \{-1,1\}^{n_i\times 1}$. Since c^i and v_{j_i} are n_i-dimensional vectors, they can be expressed as n-dimensional vectors using $(S^i)^T$ where S^i is that given by (4.10). In addition,

$$(c^1 \pm r^1 v_{j_1},\cdots,c^M \pm r^M v_{j_M}) = \sum_{i\in\mathbb{Z}^M} (S^i)^T(c^i \pm r^i v_{j_i})$$

Hence,

$$G \sum_{i\in\mathbb{Z}^M} (S^i)^T(c^i \pm r^i v_{j_i}) \leq \mathbf{1}_L, \forall j_i = 1,2,\cdots,2^{n_i} \tag{4.43}$$

The γ^{th} row of the above inequality is

$$\begin{aligned} &\pm g_\gamma \sum_{i\in\mathbb{Z}^M} (S^i)^T r^i v_{j_i} \leq 1 - g_\gamma(\sum_{i\in\mathbb{Z}^M} (S^i)^T c^i) = 1-g_\gamma c, \ \ \forall j_i = 1,2,\cdots,2^{n_i} \\ \Leftrightarrow &\sum_{i\in\mathbb{Z}^M} r^i g_\gamma^i(\pm v_{j_i}) \leq 1-g_\gamma c \ \ \forall j_i = 1,2,\cdots,2^{n_i} \end{aligned} \tag{4.44}$$

where $g_\gamma^i := g_\gamma(S^i)^T$. Consider that $g_\gamma^i(\pm v_{j_i}) \leq \|g_\gamma^i\|_1$ for each i, the following inequality is a sufficient condition of (4.44)

$$\sum_{i\in\mathbb{Z}^M} r^i \|g_\gamma^i\|_1 \leq 1-g_\gamma c \tag{4.45}$$

Noting that (4.44) holds for all $j_i = 1,2,\cdots,2^{n_i}$. However, one of these 2^{n_i} inequalities is the most binding and that happens when v_{j_i} is such that $g^i_\gamma(\pm v_{j_i}) = \|g^i_\gamma\|_1$ for each i. If this inequality holds, all $j_i = 1,2,\cdots,2^{n_i}$ inequalities hold. Hence, (4.45) is also a necessary condition of (4.44). Finally, collecting over all L rows of G, the necessary and sufficient condition of (4.43) becomes

$$\sum_{i\in\mathbb{Z}^M} r^i \|g_\gamma (S^i)^T\|_1 \leq 1 - g_\gamma c, \quad \forall \gamma \in \mathbb{Z}^L \tag{4.46}$$

(iii) The property holds since X_f is a constraint admissible invariant set for system $x^+ = A_K x, u = Kx$.

4.B Proof of Lemma 4.2:

Since (4.39b)-(4.39e) contains all admissible solutions of $\overline{\omega}^i$, and the fact that (4.39a) is the minimal value of all admissible $\overline{\omega}^i$,

$$\begin{aligned} r^i(t+1) :&= \min_{\hat{c}^{\overline{\Omega}_i +}, \hat{z}^{\overline{\Omega}_i}, \hat{c}^{\overline{\Omega}_i}, \hat{r}^{\overline{\Omega}_i}} \{\rho(c^{\Omega_i}(t+1), \hat{c}^{\overline{\Omega}_i +}) : (4.39b) - (4.39e) \text{ are satisfied.}\} \\ &\leq \rho(c^{\Omega_i}(t+1), c^{\overline{\Omega}_i}(t+1)) = \rho(c(t+1)), \end{aligned}$$

where $c(t+1) = (c^{\Omega_i}(t+1), c^{\overline{\Omega}_i}(t+1))$ is the true value of c at time $t+1$. This result implies that $X^i_f(c^i(t+1), r^i(t+1)) \subset X^i_f(c^i(t+1), \rho(c(t+1)))$. This, together with property (i) of Lemma 4.1 implies that (4.40) holds. □

4.C Proof of Theorem 4.1:

The proof follows standard reasoning and is therefore short. (i) Since (4.34) is a convex quadratic programming problem, V^i_N is strictly convex and U and X satisfy Assumption A3.1, convergence to the optimal solution via distributed ADMM is known [74]. Let the optimal states and controls of (4.34) at time t be $(x^i_{0|t}, \cdots, x^i_{N|t})$ and $(u^i_{0|t}, \cdots, u^i_{N-1|t})$ respectively and choose the feasible state

and control $\{\tilde{\boldsymbol{x}}^i(t+1),\tilde{\boldsymbol{u}}^i(t+1)\}_{i\in\mathbb{Z}^M}$ as

$$\tilde{x}^i_{0|t+1} := x^i_{1|t},\ \cdots,\ \tilde{x}^i_{N-1|t+1} := x^i_{N|t},\ \tilde{x}^i_{N|t+1} := \sum_{j\in\Omega_i}(A^{ij}+B^iK^{ij})x^j_{N|t}, \tag{4.47}$$

$$\tilde{u}^i_{0|t+1} := u^i_{1|t},\ \cdots,\ \tilde{u}^i_{N-2|t+1} := u^i_{N-1|t},\ \tilde{u}^i_{N-1|t+1} := \sum_{j\in\Omega_i}K^{ij}x^j_{N|t} \tag{4.48}$$

The last assignment of (4.47) means that $c^i(t+1)=\tilde{x}^i_{N|t+1}$ following (4.37) and $z^i(t)=x^i_{N|t}$. This choice of $(c^i(t+1),r^i(t+1))$ implies that $\{X^i_f(c^i(t+1),r^i(t+1))\}_{i\in\mathbb{Z}^M}$ satisfy (4.40) following Lemma 4.2. In addition, $\tilde{u}^i_{j|t+1}\in U_i$ because $u^i_{j-1|t}\in U_i$ and $\tilde{x}^i_{j|t+1}\in X_i$ for the same reason. The last control $\tilde{u}^i_{N-1|t+1}=\sum_{j\in\Omega_i}K^{ij}x^j_{N|t}\in U_i$ because $Kx\in U$ for all $x\in X_f$ and $(x^1_{N|t},x^2_{N|t},\cdots,x^M_{N|t})\in X_f$ since $x^i_{N|t}\in X^i_f(t)$ and that $(x^i_{0|t},\ \cdots,x^i_{N|t})$ is optimal at time t. Hence, $\{\tilde{\boldsymbol{x}}^i(t+1),\tilde{\boldsymbol{u}}^i(t+1)\}_{i\in\mathbb{Z}^M}$ of (4.47) and (4.48) is feasible to (4.34) at time $t+1$.

(ii) Since $V^*_N(x(t+1))$ is the optimal solution of (4.34) while $\{\tilde{\boldsymbol{x}}^i(t+1),\tilde{\boldsymbol{u}}^i(t+1)\}_{i\in\mathbb{Z}^M}$ is a feasible solution from (i), it follows that $V^*_N(x(t+1))\le\sum_{i\in\mathbb{Z}^M}[\sum_{k=0}^{N-1}l^i(\tilde{x}^{\Omega_i}_{k|t+1},\tilde{u}^i_{k|t+1})+l^i_f(\tilde{x}^{\Omega_i}_{N|t+1})]$. Summing this expression for time t and $t+1$ yields

$$\begin{aligned}
&V^*_N(x(t+1))-V^*_N(x(t))\le\sum_{i\in\mathbb{Z}^M}[l^i(\tilde{x}^{\Omega_i}_{N-1|t+1},\tilde{u}^i_{N-1|t+1})+l^i_f(\tilde{x}^{\Omega_i}_{N|t+1})]-\sum_{i\in\mathbb{Z}^M}[l^i(x^{\Omega_i}_{0|t},u^i_{0|t})+l^i_f(x^{\Omega_i}_{N|t})]\\
&=l(\tilde{x}_{N-1|t+1},\tilde{u}_{N-1|t+1})+l_f(\tilde{x}_{N|t+1})-l(x_{0|t},u_{0|t})-l_f(x_{N|t})\\
&=-l(x_{0|t},u_{0|t})+x^T_{N|t}\big(A_K^TPA_K-P+Q+K^TRK\big)x_{N|t}\le-l(x_{0|t},u_{0|t})-\delta\|x_{N|t}\|_2^2
\end{aligned}\tag{4.49}$$

where the last inequality comes from (4.14). The above inequality implies that $\{V^*_N(x(t))\}$ is a non-increasing sequence bounded from below by 0. Hence, $\{V^*_N(x(t))\}$ converges. This and the bounded from below property imply that $\lim_{t\to\infty}l(x_{0|t},u_{0|t})=0$, $\lim_{t\to\infty}x(t)=0$ and $\lim_{t\to\infty}\|x_{N|t}\|_2^2=0$. Hence, the origin of the closed-loop DMPC system is asymptotical stability. From (4.37), $\lim_{t\to\infty}\|x_{N|t}\|_2^2=0$ implies $\lim_{t\to\infty}c(t)=0$. □

4.D Proof of Theorem 4.2:

Assumption (A3.2) implies that $0\in int(X_f)$ which implies that $\rho(0)>0$ and $0\in int(X^1_f(0,\rho(0))\times X^2_f(0,\rho(0))\times\cdots\times X^M_f(0,\rho(0)))$. This, together with $c(t)\to 0$ from (ii) of Theorem 4.1 and property

(i) of Lemma 4.1, implies that there exists a time $\bar{t}$ such that

$$\rho(c(t)) > \varepsilon \text{ and } 0 \in int(X_f^1(t) \times X_f^2(t) \times \cdots \times X_f^M(t)) \text{ for all } t \geq \bar{t} \tag{4.50}$$

for some uniform $\varepsilon > 0$. Let $\sigma := sup\{\sigma : \sigma X_f \subset X_f^1(t) \times X_f^2(t) \times \cdots \times X_f^M(t)\}$. Note that σX_f is an invariant set for system $x(t+1) = (A+BK)x(t), x(t) \in X$ and $Kx(t) \in U$.

From (ii) of Theorem 4.1, $\lim_{t\to\infty} x(t) = 0$ implies that there exists t' such that $x(t) \in \sigma X_f$ for all $t \geq t'$. Hereafter, consider all $t > max\{\bar{t}, t'\}$. When $x(t) \in \sigma X_f$,

$$\tilde{u}^i_{\ell|t} = \sum_{j\in\Omega_i} K^{ij} x^j_{\ell|t}, \ell = 0, \cdots, N-1$$

is a feasible control for system i for the online DMPC problem for all $i \in \mathbb{Z}^M$; including when $\ell = N$ since $\sigma X_f \subset X_f^1(t) \times X_f^2(t) \times \cdots \times X_f^M(t)$. Using this choice of $\{\tilde{u}^i_{\ell|t}\}_{j=0}^{N-1}$ for all $i \in \mathbb{Z}^M$ and the corresponding $\tilde{x}^i_{\ell|t}$, the overall cost function from (4.27) can be shown to be $\tilde{V}(x(t)) = x(t)^T \bar{P} x(t)$ where $\bar{P} = \sum_{\ell=0}^{N-1} (A_K^\ell)^T (Q + K^T R K) A_K^\ell + (A_K^N)^T P A_K^N$. The optimal cost from the online DMPC is $V_N^*(x(t))$ and $V_N^*(x(t)) \leq \tilde{V}(x(t)) = \|x(t)\|_{\bar{P}}$. This, together, with the fact that $V_N^*(x(t)) \geq l(x(t), u(t)) \geq \|x(t)\|_Q^2$ means that there exists a $\overline{\beta} > \underline{\beta} > 0$ such that $\underline{\beta}\|x(t)\|_2^2 \leq \|x(t)\|_Q^2 \leq V_N^*(x(t)) \leq \overline{\beta}\|x(t)\|_2^2$ From (4.49), $V_N^*(x(t+1)) - V_N^*(x(t)) \leq -l(x(t), u(t)) \leq -\|x(t)\|_Q^2 \leq -\underline{\beta}\|x(t)\|_2^2$ and the above inequality implies

$$V_N^*(x(t+1)) - V_N^*(x(t)) \leq -\underline{\beta}\|x(t)\|_2^2 \leq -aV_N^*(x(t))$$

where $a = \underline{\beta}/\overline{\beta} < 1$. Hence, $V_N^*(x(t+1)) \leq (1-a)V_N^*(x(t))$, which implies $\|x(t)\|_2^2 \leq \frac{1}{a}(1-a)^{t-\tilde{t}}\|x(\tilde{t})\|_2^2$ for all $t \geq \tilde{t}$, where $\tilde{t} := max\{\bar{t}, t'\} + 1$. This shows local exponential convergence of $x(t)$ to the origin. This result can be extended to all x in the domain of attraction following (A3.1), see for example [29] and [170]. □

CHAPTER 5

Distributed Model Predictive Control of Linear Discrete-Time Systems with Coupled Constraints

5.1 Introduction

This chapter considers the Distributed Model Predictive Control (DMPC) of M discrete-time linear dynamical systems, each of which is of the form

$$x^i(t+1) = A^i x^i(t) + B^i u^i(t), \tag{5.1}$$

$$x^i(t) \in X^i,\ u^i(t) \in U^i,\ \ i = 1, \cdots, M \tag{5.2}$$

and all of them has to satisfy a coupled/global constraint of the form

$$\sum_{i=1}^{M} \left(\Psi_x^i x^i(t) + \Psi_u^i u^i(t) \right) \leq \mathbf{1}_p, \qquad \text{for all } t \tag{5.3}$$

where x^i, u^i are the states and controls of the i^{th} system respectively and $X^i \subset \mathbb{R}^{n_i}, U^i \subset \mathbb{R}^{m_i}$ are the corresponding constraint sets; the matrices $\Psi_x^i \in \mathbb{R}^{p \times n_i}$ and $\Psi_u^i \in \mathbb{R}^{p \times m_i}$ define the coupled constraints for all M systems and $\mathbf{1}_p$ is the p-vector of all ones.

The study of DMPC is an active area of research [15, 35, 38, 40, 46] and one popular area is when the systems are dynamically coupled [48, 49, 66, 88, 163, 171]. However, these approaches are not suitable for the problem above due to the complications arising from (5.3). To the best of our knowledge, DMPC approaches for (5.1)-(5.3) are somewhat limited. The method of [44] ensures the satisfaction of (5.3) using a sequential process: one system is optimized at a time with all others stay constant; this is followed sequentially by another system so that all M systems are optimized once

in M time steps. Another approach is known as the cooperative MPC method [60, 121, 122]. While specific details vary, the basic idea is that all systems within a cooperating set (possibly a singleton) are optimized jointly (or in parallel) while systems outside the cooperating set follow their predicted states and predicted controls. These methods optimize individual or groups of systems sequentially. However, the optimality of the overall system is unclear as they are not explicitly pursued. In addition, these approaches require direct communications among systems that are coupled by (5.3) which, for a large system, can impose heavy communication requirement [172–174].

A reasonable approach for (5.1)-(5.3) [84, 149] that achieves overall optimality is to solve the dual problem involving the Lagrangian function. In this case, the Lagrangian function is the sum of M separable functions except for the dual variable associated with (5.3). This dual variable is treated as a consensus variable in a distributed consensus optimization problem (DCOP). Typically, consensus of the dual variable is ensured (Chapter 6 of [149]) using a central/master node.

This work follows the above formulation resulting in a DCOP. However, the DCOP is solved using the Distributed ADMM [144, 145, 152, 175] algorithm where each system has a local copy of the dual variable. These local copies need not reach consensus but only within some fixed bound of one another. Such an approach is used because the computational effort of the Distributed ADMM is high and allowing premature termination of the ADMM algorithm provides for computational expediency. Measures to handle such premature termination are provided, together with recursive feasibility and stability of the closed-loop system. Under reasonable assumptions, the approach is guaranteed to converge to some small neighborhood of the overall optimal solution so long as the network is connected. The approach is iterative, similar to other MPC schemes [176–178] but for multiple systems with coupled constraints.

The rest of this chapter is organized as follows. This section ends with a description of the notations used. Section 5.2 reviews some results of the standard stand-alone MPC for a single system and discusses the formulation of the overall MPC problem. Section 5.3 presents the proposed approach, including the discussion of the coupled constraint, its dual and the convergence of the distributed ADMM algorithm. The recursive feasibility and stability results are given in Section 5.5. The performance of the approach is illustrated by a numerical example in Section 5.6 with the conclusions given in Section 5.7. All proofs are given in the appendices.

The notations used in this chapter are as follows. Non-negative and positive integer sets are indicated by $\mathbb{Z}_0^+$ and $\mathbb{Z}^+$ respectively. Let $M, L \in \mathbb{Z}_0^+$ with $M \geq L$. Then $\mathbb{Z}^M := \{1, 2, \cdots, M\}$ and $\mathbb{Z}_L^M := \{L, L+1, \cdots, M\}$. Similarly, $\mathbb{R}_0^+$ and $\mathbb{R}^+$ refer respectively to the sets of non-negative and positive real number. I_n is the $n \times n$ identity matrix, $\mathbf{1}_n$ is the n-column vector of all ones (subscript omitted when the dimension is clear) and $|S|$ is the cardinality of the index set S. Given $\sigma > 0$, $X \subset \mathbb{R}^n$ with $0 \in int(X)$ where $int(\cdot)$ is the interior of a set, $\sigma X = \{\sigma x : x \in X\}$. For a square matrix Q, $Q \succ (\succeq) 0$ means Q is positive definite (semi-definite). The ℓ_p-norm of $x \in \mathbb{R}^n$ is $\|x\|_p$ while $\|x\|_Q^2 = x^T Q x$ for $Q \succ 0$. Several representations of the states and controls are needed: $x^i(t)$, $u^i(t)$ refer to the state and control of the i^{th} system at time t; x_k^i, u_k^i are the k^{th} predicted state and predicted control of the i^{th} system; $x = (x^1, x^2, \cdots, x^M)$, $u = (u^1, u^2, \cdots, u^M)$ are the collections of x^i and u^i over the M systems; boldface $\boldsymbol{x}^i = (x_1^i, x_2^i, \cdots, x_N^i)$, $\boldsymbol{u}^i = (u_0^i, u_1^i, \cdots, u_{N-1}^i)$ are respectively the collections of the N predicted states and predicted controls over the horizon (of length N) for the i^{th} system; in situation where the reference to time is needed, x_k^i, u_k^i can be written as $x_{k|t}^i$ and $u_{k|t}^i$. Hence, $x_{0|t}^i = x^i(t)$ and $u_{0|t}^i = u^i(t)$. Additional notations are introduced as required in the text.

5.2 Preliminaries and Problem Formulation

This section reviews some well-known results in standard MPC and other related concepts. Consider a stand-alone system represented by one choice of $i \in \mathbb{Z}^M$ in (5.1) with corresponding cost and constraints being

$$\min_{\boldsymbol{u}^i} J^i(x^i, \boldsymbol{u}^i) := \sum_{\ell=0}^{N-1} (\|x_\ell^i\|_{Q^i}^2 + \|u_\ell^i\|_{R^i}^2) + \|x_N^i\|_{P^i}^2 \tag{5.4a}$$

$$\text{s.t. } \boldsymbol{u}^i \in \mathcal{U}_T^i(x^i) \tag{5.4b}$$

where N is the horizon length, $\boldsymbol{u}^i := \{u_0^i, u_1^i, \cdots, u_{N-1}^i\}$, $\boldsymbol{x}^i := \{x_0^i, x_1^i, \cdots, x_N^i\}$ are the predicted controls and predicted states respectively satisfying $x_{\ell+1}^i = A^i x_\ell^i + B^i u_\ell^i$ with $x_0^i = x^i$, $J(x^i, \boldsymbol{u}^i)$ is the standard quadratic costs parameterized by $(x^i, \boldsymbol{u}^i)$ defined by (5.4a). In addition, K^i and P^i are the solution to the Algebraic Riccatti Equation (ARE) with weights $Q^i \succ 0, R^i \succ 0$. Let

$$\mathcal{U}_T^i(x^i) := \{\boldsymbol{u}^i \in \mathbb{R}^{m_i N} : x_{\ell+1}^i = A^i x_\ell^i + B^i u_\ell^i, x_0^i = x^i, x_\ell^i \in X^i, u_\ell^i \in U^i, x_N^i \in T_f^i, \ell \in \mathbb{Z}_0^{N-1}\} \tag{5.5}$$

where T_f^i is some appropriate terminal set satisfying

$$A_K^i x^i \in T_f^i, \;\; K^i x^i \in U^i \text{ for all } x^i \in T_f^i \tag{5.6}$$

with $A_K^i := A^i + B^i K^i$. The overall MPC optimization problem over the M systems incorporating (5.3) at state $x = \{x^1, \cdots, x^M\}$ is given

$$\mathbb{P}(x): \quad V(x) := \min_{\{\boldsymbol{u}^i, i\in\mathbb{Z}^M\}} \sum_{i=1}^{M} J^i(x^i, \boldsymbol{u}^i) \tag{5.7a}$$

$$s.t. \;\; \boldsymbol{u}^i \in \mathcal{U}_T^i(x^i), \;\; \forall i \in \mathbb{Z}^M, \tag{5.7b}$$

$$\sum_{i=1}^{M} \Psi_x^i x_\ell^i + \Psi_u^i u_\ell^i \leq \mathbf{1}_p, \;\; \forall \ell \in \mathbb{Z}_0^{N-1} \tag{5.7c}$$

where (5.7c) refers to the satisfaction of the coupled constraints at each predicted time step of the horizon. The conditions of (5.6) on T_f^i do not include the effect of the coupled constraint which is given by

$$\sum_{i=1}^{M} \bar{\Psi}^i x^i := \sum_{i=1}^{M} (\Psi_x^i + \Psi_u^i K^i) x^i \leq \mathbf{1}_p, \; \forall x^i \in T_f^i, \tag{5.8}$$

5.2.1 Tightening the Constraints

The formulation of (5.7) and condition of (5.8) are appropriate when the Distributed ADMM algorithm achieves convergence at every time step. However, the online verification of the convergence of a distributed algorithm is numerically expensive. Consequently, (5.7) and (5.8) need to be tightened to account for errors arising from the premature termination of the Distributed ADMM algorithm. Specifically, the tightened constraints for (5.7c) and (5.8) are

$$\sum_{i=1}^{M} \Psi_x^i x_\ell^i + \Psi_u^i u_\ell^i \leq (1 - \varepsilon M(\ell+1))\mathbf{1}_p \;\; \forall \ell \in \mathbb{Z}_0^{N-1} \tag{5.9}$$

$$\sum_{i=1}^{M} \bar{\Psi}^i x^i \leq (1 - MN\varepsilon)\mathbf{1}_p, \forall x^i \in T_f^i \tag{5.10}$$

where ε is the error arising from the inaccurate solution of the distributed ADMM algorithm. Obviously, $0 < \varepsilon < \frac{1}{MN}$ to ensure that $0 \in int(T_f^i)$ in (5.10). Note that the local constraints of (5.7b) are

not tightened as they are satisfied for all premature solutions (see next section for details). Correspondingly, the tightened DMPC formulation is

$$\min_{\{\boldsymbol{u}^i, i\in\mathbb{Z}^M\}} \{\sum_{i=1}^{M} J^i(x^i,\boldsymbol{u}^i) : (5.7b) \text{ and } (5.9)\}. \tag{5.11a}$$

for some approprate T_f^i that satisfy (5.6) and (5.10). The choice of T_f^i is now discussed and the following assumptions are made.

(A5.1): (A^i,B^i) is stabilizable and $x^i(t)$ is measurable for all $i\in\mathbb{Z}^M$;

(A5.2): X^i,U^i are polytopes containing the origins in their respective interiors for all $i\in\mathbb{Z}^M$;

The choice of T_f^i is chosen to be

$$T_f^i = \sigma_\varepsilon^i X_f^i, \ \ \forall i\in\mathbb{Z}^M$$

where X_f^i is the maximal polytope [32] that satisfy (5.6). The choice of σ_ε^i is chosen to satisfy (5.10) and can be obtained by noting that (5.10) holds if for some value of $\sigma^i, i\in\mathbb{Z}^M$,

$$\sum_{i=1}^{M}\left(max_{x^i\in\sigma^i X_f^i}\bar{\Psi}_\ell^i x^i\right) \le 1-MN\varepsilon, \forall \ell\in\mathbb{Z}^p \tag{5.12}$$

where $\bar{\Psi}_\ell^i$ is the ℓ^{th} row of matrix $\bar{\Psi}^i$. Using $h_S(v) := \max\{v^T x : x\in S\}$, the support function of set S along the direction of v, this last expression can be further simplified into

$$\sum_{i=1}^{M} h_{\sigma^i X_f^i}(\bar{\Psi}_\ell^i) = \sum_{i=1}^{M}\sigma^i h_{X_f^i}(\bar{\Psi}_\ell^i) \le 1-MN\varepsilon, \ \ \forall \ell\in\mathbb{Z}^p \tag{5.13}$$

where p is the number of rows of $\bar{\Psi}^i$ and $h_{\sigma S}(v) = \sigma h_S(v)$ for any fixed $\sigma>0$. The expression of (5.13) allows the determination of $\sigma_\varepsilon^i, i\in\mathbb{Z}^M$, such that (5.10) holds. For example, one obvious choice is $(\sigma_\varepsilon^1,\cdots,\sigma_\varepsilon^M) = \arg\min\{\sum_{i=1}^M(1-\sigma^i) : (5.13)\}$. Let

$$\mathcal{U}^i(x^i) := \mathcal{U}_T^i(x^i) \text{ when } T_f^i = \sigma_\varepsilon^i X_f^i \tag{5.14}$$

and express (5.9) in terms of $(x^i, \boldsymbol{u}^i)$ for each $i \in \mathbb{Z}^M$. The tightened DMPC formulation can be represented as

$$\mathbb{P}_\varepsilon(x): \quad V_\varepsilon(x) := \min_{\{\boldsymbol{u}^i, i\in\mathbb{Z}^M\}} \sum_{i=1}^{M} J^i(x^i, \boldsymbol{u}^i) \tag{5.15a}$$

$$s.t. \ \ \boldsymbol{u}^i \in \mathcal{U}^i(x^i) \quad i \in \mathbb{Z}^M \tag{5.15b}$$

$$\sum_{i=1}^{M} f^i(x^i, \boldsymbol{u}^i) \leq b(\varepsilon) \tag{5.15c}$$

where $b^T(\varepsilon) := [(1-M\varepsilon)\mathbf{1}_p^T, (1-2M\varepsilon)\mathbf{1}_p^T, \cdots, (1-NM\varepsilon)\mathbf{1}_p^T]^T$,

$$f^i(x^i, \boldsymbol{u}^i) = F^i \boldsymbol{u}^i + H^i x^i \tag{5.16}$$

and $F^i \in \mathbb{R}^{Np\times Nm_i}, H^i \in \mathbb{R}^{Np\times n_i}$ are appropriate matrices from (5.9) by rewriting x_ℓ^i in terms of x^i and $\boldsymbol{u}^i$. Let the feasible domain of $\mathbb{P}_\varepsilon(x)$ be

$$\mathcal{D}_\varepsilon := \{x \in \mathbb{R}^n : \text{ there exists a feasible } \{\boldsymbol{u}^i\}_{i\in\mathbb{Z}^M} \text{ to } \mathbb{P}_\varepsilon(x)\} \tag{5.17}$$

5.2.2 Network Description

The next section describes the proposed Distributed ADMM algorithm and for that, the description of network of system is needed. Using standard terminology, the M systems is represented by an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V} = \{1, 2, \cdots, M\}$ and edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. The adjacency matrix $\mathcal{A}$ of G is the $M \times M$ matrix whose (i, j) entry is 1 if $(i, j) \in \mathcal{E}$, and 0 otherwise. The set of neighbors of the i^{th} system is $N_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}, i \neq j\}$ with $d_i = |N_i|$ and $D = diag\{d_1, d_2, \cdots, d_M\}$. The connection of the graph can be arbitrary so long as

(A5.3): G is connected.

5.3 The Proposed Algorithm

5.3.1 The Dual Form

Let $\lambda \in \mathbb{R}^{Np}$ be the dual variable associated with the coupled constraint (5.15c). The Lagrangian of (5.15) is $\mathcal{L}(\{\boldsymbol{u}^i\},\lambda) = \sum_{i=1}^{M} J^i(x^i,\boldsymbol{u}^i) + \lambda^T(\sum_{i=1}^{M} f^i(x^i,\boldsymbol{u}^i) - b(\varepsilon))$ for all $\boldsymbol{u}^i \in \mathcal{U}^i(x^i), i \in \mathbb{Z}^M$ and the dual problem is

$$\max_{\lambda \geq 0} \min_{\{\boldsymbol{u}^i \in \mathcal{U}^i(x^i), i \in \mathbb{Z}^M\}} \mathcal{L}(\{\boldsymbol{u}^i\},\lambda). \tag{5.18}$$

This dual problem is also equivalent to

$$\min_{\lambda \geq 0} \max_{\{\boldsymbol{u}^i \in \mathcal{U}^i(x^i), i \in \mathbb{Z}^M\}} -\mathcal{L}(\{\boldsymbol{u}^i\},\lambda) := \min_{\lambda \geq 0} \sum_{i=1}^{M} g_i(\lambda) \tag{5.19}$$

where

$$g_i(\lambda) := \max_{\boldsymbol{u}^i \in \mathcal{U}^i(x^i)} -J^i(x^i,\boldsymbol{u}^i) - \lambda^T(f^i(x^i,\boldsymbol{u}^i) - \frac{b(\varepsilon)}{M}). \tag{5.20}$$

Let $\boldsymbol{u}^i(\lambda) = \arg\max_{\boldsymbol{u}^i \in \mathcal{U}^i(x^i)} g_i(\lambda)$. Then, the gradient of $g_i(\lambda)$ can be shown to be (see Danskin's Theorem of [149])

$$\nabla g_i(\lambda) = -(f^i(x^i,\boldsymbol{u}^i(\lambda)) - \frac{b(\varepsilon)}{M}) \tag{5.21}$$

Note that while the optimal solution of (5.15) is unique as $J^i(x^i,\boldsymbol{u}^i)$ is stictly convex in $\boldsymbol{u}^i$, the optimal solution of (5.19) may not be [177, 179]. However, as shown in the sequel, the proposed algorithm (Algorithm 5.1) will converge to unique solution of (5.15), see property (ii) of Theorem 5.1.

5.3.2 The Conversion to a Consensus Optimization Problem

The above dual problem is not fully distributed because λ appears in all M systems. From A4.3, (5.19) can be rewritten as a consensus problem with edge-wise constraints as

$$\min_{\lambda^i \geq 0, i \in \mathbb{Z}^M} \sum_{i=1}^{M} g_i(\lambda^i) \quad s.t. \quad \lambda^i = \lambda^j, (i,j) \in \mathcal{E} \tag{5.22}$$

where λ^i is the local copy of λ for the i system and the condition of $\lambda^i = \lambda^j$ enforces consensus among all the local copies. This problem can be further rewritten using a new set of variables w^{ij} in the form of

$$\min_{\boldsymbol{\lambda} \geq 0, \boldsymbol{w}} \sum_{i=1}^{M} g_i(\lambda^i) \quad s.t. \quad \lambda^i = w^{ij}, \lambda^j = w^{ij}, (i,j) \in \mathcal{E} \tag{5.23}$$

where $\boldsymbol{\lambda} := \{\lambda^1, \lambda^2, \cdots, \lambda^M\}$ and $\boldsymbol{w} := \{w^{ij} : (i,j) \in \mathcal{E}\}$, see [84, 180]. Let $\boldsymbol{\alpha} = \{\alpha_{ij}, (i,j) \in \mathcal{E}\}$ and $\boldsymbol{\beta} = \{\beta_{ij}, (i,j) \in \mathcal{E}\}$ where α_{ij} and β_{ij} are the dual variables associated with $\lambda^i - w^{ij} = 0$ and $\lambda^j - w^{ij} = 0$ respectively. Since $g_i(\lambda^i)$ is convex, this optimization problem can be solved using the standard two-block ADMM algorithm with $\boldsymbol{\lambda}$ and $\boldsymbol{w}$ being the two sets of variables. The augmented Lagrangian of (5.23) is

$$\begin{aligned} \mathcal{L}_\rho(\boldsymbol{\lambda}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{M} g_i(\lambda^i) + \sum_{i=1}^{M} \sum_{j \in N_i} \{\alpha_{ij}^T (\lambda^i - w^{ij}) \\ + \beta_{ij}^T (\lambda^j - w^{ij}) + \frac{\rho}{2} \|\lambda^i - w^{ij}\|^2 + \frac{\rho}{2} \|\lambda^j - w^{ij}\|^2 \} \end{aligned} \tag{5.24}$$

for some $\rho > 0$. The standard ADMM consists of the following iterations

$$\begin{aligned} \boldsymbol{\lambda}^{k+1} &= \arg\min_{\boldsymbol{\lambda} \geq 0} \mathcal{L}_\rho(\boldsymbol{\lambda}, \boldsymbol{w}^k, \boldsymbol{\alpha}^k, \boldsymbol{\beta}^k) \\ \boldsymbol{w}^{k+1} &= \arg\min_{\boldsymbol{w}} \mathcal{L}_\rho(\boldsymbol{\lambda}^{k+1}, \boldsymbol{w}, \boldsymbol{\alpha}^k, \boldsymbol{\beta}^k) \\ \alpha_{ij}^{k+1} &= \alpha_{ij}^k + \rho(\lambda^{i,k+1} - w^{ij,k+1}), (i,j) \in \mathcal{E} \\ \beta_{ij}^{k+1} &= \beta_{ij}^k + \rho(\lambda^{j,k+1} - w^{ij,k+1}), (i,j) \in \mathcal{E} \end{aligned}$$

5.3.3 Distributed ADMM Algorithm

Following the manipulations of Distributed ADMM [145, 146, 151, 152], the steps above can be simplified by letting

$$\alpha_{ij}^k + \beta_{ij}^k = 0, \alpha_{ij}^k = \beta_{ji}^k, w^{ij,k} = \frac{\lambda^{i,k} + \lambda^{j,k}}{2}, \tag{5.25}$$

for all $(i, j) \in \mathcal{E}$ and k. The simplified distributed ADMM is then given by

$$\lambda^{i,k+1} = \arg\min_{\lambda^i \geq 0} g_i(\lambda^i) + (v^{i,k})^T \lambda^i + \rho |N_i| \|\lambda^i\|^2 \tag{5.26}$$

$$\alpha_{ij}^{k+1} = \alpha_{ij}^k + \frac{\rho}{2}(\lambda^{i,k+1} - \lambda^{j,k+1}), (i, j) \in \mathcal{E} \tag{5.27}$$

where

$$v^{i,k} = 2 \sum_{j \in N_i} \alpha_{ij}^k - \rho \sum_{j \in N_i} (\lambda^{i,k} + \lambda^{j,k}) \tag{5.28}$$

Using the expression of $g_i(\lambda^i)$ of (5.20), the minimization problem of (5.26) can be rewritten as

$$\begin{aligned} &\min_{\lambda^i \geq 0} \max_{\boldsymbol{u}^i \in \mathcal{U}^i(x^i)} \{-J^i(x^i, \boldsymbol{u}^i) - (f^i(x^i, \boldsymbol{u}^i) - \frac{b(\varepsilon)}{M})^T \lambda^i + (v^{i,k})^T \lambda^i + \rho |N_i| \|\lambda^i\|^2\} \\ :=& \min_{\lambda^i \geq 0} \max_{\boldsymbol{u}^i \in \mathcal{U}^i(x^i)} \phi^i(\lambda^i, \boldsymbol{u}^i) \end{aligned} \tag{5.29}$$

where $\phi^i(\lambda^i, \boldsymbol{u}^i)$ is defined implicitly above. Since $\phi^i(\lambda^i, \boldsymbol{u}^i)$ is strictly convex in λ^i and strictly concave in $\boldsymbol{u}^i$, the ordering of the min and max operations can be interchanged [181], yielding

$$\max_{\boldsymbol{u}^i \in \mathcal{U}^i(x^i)} \min_{\lambda^i \geq 0} \phi^i(\lambda^i, \boldsymbol{u}^i) = \max_{\boldsymbol{u}^i \in \mathcal{U}^i(x^i)} \{\max_{s^i \geq 0} \min_{\lambda^i} \{\phi^i(\lambda^i, \boldsymbol{u}^i) - (s^i)^T \lambda^i\}\} \tag{5.30}$$

$$= \max_{\substack{\boldsymbol{u}^i \in \mathcal{U}^i(x^i) \\ s^i \geq 0}} \min_{\lambda^i} \{\phi^i(\lambda^i, \boldsymbol{u}^i) - (s^i)^T \lambda^i\} \tag{5.31}$$

where s^i is the multiplier associated with the constraint $\lambda^i \geq 0$ and the equality of (5.30) is from the strong duality [182] of the Lagrangian function of $\min_{\lambda^i \geq 0} \phi^i(\lambda^i, \boldsymbol{u}^i)$. For any $\boldsymbol{u}^i \in \mathcal{U}^i(x^i)$ and $s^i \geq 0$, the solution of the inner minimization of (5.31) is given by $\lambda^i = \frac{1}{2\rho|N_i|}(f^i(x^i, \boldsymbol{u}^i) + s^i - \frac{b(\varepsilon)}{M} - v^{i,k})$ which,

when substituted into (5.31), yields

$$(\boldsymbol{u}^{i,k+1},s^{i,k+1}) = \arg\min_{\substack{\boldsymbol{u}^i\in\mathcal{U}^i(x^i)\\ s^i\geq 0}} \{J^i(x^i,\boldsymbol{u}^i) + \frac{1}{4\rho|N_i|}\| f^i(x^i,\boldsymbol{u}^i) + s^i - \frac{b(\varepsilon)}{M} - v^{i,k}\|^2\} \tag{5.32}$$

Hence, the optimal λ^i of (5.26) can be obtained in closed-form as

$$\lambda^{i,k+1} = \frac{1}{2\rho|N_i|}(f^i(x^i,\boldsymbol{u}^{i,k+1}) + s^{i,k+1} - \frac{b(\varepsilon)}{M} - v^{i,k}). \tag{5.33}$$

For each $i \in \mathbb{Z}^M$ and iterate k, the control sequence is updated according to (5.32). Once $\boldsymbol{u}^{i,k+1}$ is obtained, $\lambda^{i,k+1}$ is updated according to (5.33). The value of v^i is then updated according to (5.28) and this process is repeated by incrementing k. The stopping condition of this ADMM process is discussed in Section 5.4 and suppose the process stops at iteration $\bar{k}$. Then, the solution from the ADMM algorithm is $\boldsymbol{u}^{i,\bar{k}} := \{u_0^{i,\bar{k}}, u_1^{i,\bar{k}}, \cdots, u_{N-1}^{i,\bar{k}}\}$. Correspondingly, the MPC control law applied on the i^{th} system is

$$\kappa^i(x) = u_0^{i,\bar{k}}, i \in \mathbb{Z}^M \tag{5.34}$$

5.3.4 Convergence of the Distributed ADMM Algorithm

To the best of authors' knowledge, (5.32) and (5.33) are not standard ADMM formulation and this section discusses their convergence. Let

$$r^k := \sum_{i=1}^{M}(f^i(x^i,\boldsymbol{u}^{i,k}) + s^{i,k}) - b(\varepsilon), \tag{5.35}$$

$$\Delta J^k := \sum_{i=1}^{M}\left(J^i(x^i,\boldsymbol{u}^{i,k}) - J^i(x^i,\boldsymbol{u}^{i*})\right) \tag{5.36}$$

where $\{\boldsymbol{u}^{i*}\}$ is the optimal solution of (5.15), r^k is the residual of the coupled constraints and ΔJ^k is the deviation of the primal objective at iteration k from its optimal. The following theorem states the convergence results.

Theorem 5.1. *Suppose (A5.1)-(A5.3) hold. For any $x \in \mathcal{D}_\varepsilon$, let $\{\boldsymbol{u}^{i,k}, s^{i,k}, \lambda^{i,k}\}_{i=1}^M$ be generated by (5.32)-(5.33) and $\boldsymbol{\alpha}^k = \{\alpha_{ij}^k, (i,j) \in \mathcal{E}\}$ be generated by (5.27) with $\lambda^{i,0} = 0$ and $\alpha_{ij}^0 = 0$ for all $j \in N_i$ and $i \in \mathbb{Z}^M$. Suppose $\{\boldsymbol{u}^{i*}\}_{i=1}^M$ is the optimal solution of (5.15), λ^* is an optimal dual variable*

associated with (5.15c) and $\boldsymbol{\alpha}^* = \{\alpha_{ij}^*, (i,j) \in \mathcal{E}\}$ *is an optimal dual variable of (5.23). Then, the following results hold.*

(i) The sequences $\boldsymbol{\lambda}^k$ *and* $\boldsymbol{\alpha}^k$ *converge.*

(ii) $\boldsymbol{u}^{i,k}$ *converges to* $\boldsymbol{u}^{i*}$ *for all* $i \in \mathbb{Z}^M$ *and* r^k *converges to* 0.

(iii) $|\Delta J^k|$ *converges to* 0 *with* $|\Delta J^k| \leq C^k$ *where*

$$C^k := \|\lambda^*\|\|r^k\| + \frac{2}{\rho}(\|\boldsymbol{\alpha}^k\| + \|\boldsymbol{\alpha}^*\|)\|\Delta\boldsymbol{\alpha}^k\| + \rho(\|\boldsymbol{\lambda}^k\| + \|\boldsymbol{\lambda}^*\|)\|\bar{\Gamma}\|\|\Delta\boldsymbol{\lambda}^k\|, \tag{5.37}$$

$\Delta\boldsymbol{\alpha}^k = \boldsymbol{\alpha}^k - \boldsymbol{\alpha}^{k-1}, \Delta\boldsymbol{\lambda}^k = \boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}$ *and* $\bar{\Gamma} = (\mathcal{A} + D) \otimes I_{Np}$.

(iv) $\boldsymbol{u}^{i,k}$ *is feasible to (5.15b) for all k,* $i \in \mathbb{Z}^M$.

In addition, the convergence rate of the distributed ADMM can also be obtained. The worst-case $O(1/k)$ convergence rate is given in the following theorem.

Theorem 5.2. *For* $k \geq 1$, *let*

$$\begin{aligned} \tilde{\boldsymbol{u}}^{i,k} &= \frac{1}{k}\sum_{\ell=1}^{k} \boldsymbol{u}^{i,\ell} = (1 - \frac{1}{k})\tilde{\boldsymbol{u}}^{i,k-1} + \frac{1}{k}\boldsymbol{u}^{i,k} \\ \tilde{s}^{i,k} &= \frac{1}{k}\sum_{\ell=1}^{k} s^{i,\ell} = (1 - \frac{1}{k})\tilde{s}^{i,k-1} + \frac{1}{k}s^{i,k} \end{aligned} \tag{5.38}$$

be the running weighted-averages of the primal iterates of (5.32) with $\tilde{\boldsymbol{u}}^{i,0} = 0$ *and* $\tilde{s}^{i,0} = 0$ *for* $i \in \mathbb{Z}^M$. *Then, the following results hold.*

(i) The sequence $\{\|\tilde{\boldsymbol{u}}^{i,k} - \boldsymbol{u}^{i*}\|^2\}_{k=1}^{\infty}$ *goes to* 0 *with the convergence rate* $O(\frac{1}{k})$ *for all* $i \in \mathbb{Z}^M$.

(ii) The coupled constraint is bounded by

$$\begin{aligned} &\sum_{i=1}^{M} f^i(\tilde{\boldsymbol{u}}^{i,k}, x^i) - b(\varepsilon) \\ &\leq \left(\frac{1}{k}\right)\left(\frac{\rho\lambda_{\max}(\Gamma)}{2}M(1 + \|\lambda^*\|)^2 + \frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2\right)\mathbf{1}_{Np} \end{aligned} \tag{5.39}$$

(iii) The primal cost error is given as

$$\begin{aligned} &|\sum_{i=1}^{M} J^i(x^i(t), \tilde{\boldsymbol{u}}^{i,k}) - \sum_{i=1}^{M} J^i(x^i(t), \boldsymbol{u}^{i*})| \\ &\leq \frac{2 + \|\lambda^*\|}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2 + \frac{\rho}{2}\lambda_{\max}(\Gamma)M(1 + \|\lambda^*\|)^2\right) \end{aligned} \tag{5.40}$$

From the theorem stated above, the proposed distributed ADMM has a convergence rate of $O(1/k)$, similar to other ADMM schemes [144, 183]. Hence, a stopping condition for (5.32)-(5.33) can be based on some choice of k. However, such a condition is well known to be very conservative [74]. This chapter proposes a stopping condition based on r^k and $|\Delta J^k|$ for early termination of (5.32)-(5.33). To do so, (5.15c) is relaxed (but not (5.15b) because of (iv) of Theorem 5.1) as given below.

Definition 5.1. *Given any $\varepsilon, \delta > 0$, the set $\{x^i, \boldsymbol{u}^i\}_{i=1}^M$ is a (ε, δ)-relaxed solution of (5.15) if*

$$\boldsymbol{u}^i \in \mathcal{U}^i(x^i), i \in \mathbb{Z}^M, \ \sum_{i=1}^M f^i(x^i, \boldsymbol{u}^i) - b(\varepsilon) \leq \varepsilon M \mathbf{1}_{pN} \text{ and } |\sum_{i=1}^M \left((J^i(x^i, \boldsymbol{u}^i) - J^i(x^i, \boldsymbol{u}^{i*}) \right)| \leq \delta \quad (5.41)$$

where $\{\boldsymbol{u}^{i*}\}_{i=1}^M$ is the optimal solution of (5.15).

The following lemma is needed to establish the stopping criterion.

Lemma 5.1. *Suppose $\{\boldsymbol{u}^{i,k}\}_{i=1}^M$ is generated from (5.32) in the distributed ADMM approach above, then there exists a $\bar{k}$ such that*

$$r^{\bar{k}} \leq \varepsilon M \mathbf{1}_{pN} \text{ and } C^{\bar{k}} \leq \delta \quad (5.42)$$

where $r^{\bar{k}}$ and $C^{\bar{k}}$ are those given by (5.35) and (5.36) respectively. In addition, $\{x^i, \boldsymbol{u}^{i,\bar{k}}\}_{i=1}^M$ is a (ε, δ)-relaxed solution of (5.15).

5.4 The Stopping Criterion for the Distributed ADMM

The stopping criterion of the proposed Distributed ADMM uses the results of Lemma 5.1. Specifically, the Distributed ADMM stops at the first value of k, $\bar{k}$, such that (5.42) hold. One possible approach is use a central node [184] to compute r^k and C^k of (5.42) by collecting all relevant information from each of the M systems. However, such a centralized approach is not desirable due to its communication requirement. This section describes a distributed approach to compute r^k and C^k of (5.42) using the finite-time average consensus algorithm in Section 3.3.2.

The use of the finite-time average consensus algorithm (3.39) for the stopping condition of the

distributed ADMM is given as follows. Recall that this stopping condition is given by (5.42). Let $z^i(0) = f^i(x^i, \boldsymbol{u}^{i,k}) + s^{i,k} - \frac{b(\varepsilon)}{M}$. It follows from (5.42) that $r^k = \sum_{i=1}^M z^i(0)$ and its value can be obtained from (3.39) using $z^i(0), \cdots, z^i(T-1)$ obtained from (3.40) for each system.

The case of C^k is similar. From (5.37), evaluation of C^k requires the values of r^k, $\|\boldsymbol{\alpha}^k\|$, $\|\Delta\boldsymbol{\alpha}^k\|$, $\|\boldsymbol{\lambda}^k\|$, $\|\Delta\boldsymbol{\lambda}^k\|$, $\|\lambda^*\|$ and $\|\boldsymbol{\alpha}^*\|$. Values of $\|\lambda^*\|$ and $\|\boldsymbol{\alpha}^*\|$ can be estimated using the offline procedure given in [177] and is not discussed here. The rest of them can be computed in the same manner as r^k. For example, $\|\boldsymbol{\lambda}^k\|^2 = \sum_{i\in\mathbb{Z}^M} \|\lambda^{i,k}\|^2 = \sum_{i\in\mathbb{Z}^M} z^i(0)$ if $z^i(0) := \|\lambda^{i,k}\|^2$ for all $i \in \mathbb{Z}^M$. Hence, $\sum_{i\in\mathbb{Z}^M} z^i(0)$ is obtained from (3.39) using $z^i(0), \cdots, z^i(T-1)$ following (3.40). The same is true for $\|\boldsymbol{\alpha}^k\|, \|\Delta\boldsymbol{\alpha}^k\|$ and $\|\boldsymbol{\lambda}^k\|$. Altogether, 5 consensus dynamics are running simultaneously for $T-1$ steps for the evaluations of r^k and C^k.

The overall procedure of the distributed ADMM algorithm at time t is summarized in the following algorithm.

Algorithm 5.1: Consensus ADMM algorithm

Input: x^i, $i \in \mathbb{Z}^M$

Output: $\boldsymbol{u}^{i,\bar{k}}$, $i \in \mathbb{Z}^M$

Initialization: choose $\rho > 0$, set $k = 0$, $\lambda^{i,0} = 0$ and $\alpha_{ij}^0 = 0$, for all $j \in N_i$, $i \in \mathbb{Z}^M$;

repeat

ADMM:

for all $i \in \mathbb{Z}^M$ (in parallel) **do**

Obtain $v^{i,k}$ from (5.28), $(\boldsymbol{u}^{i,k+1}, s^{i,k+1})$ from (5.32) and $\lambda^{i,k+1}$ from (5.33) respectively;

Exchange $\lambda^{i,k+1}$ with all its neighbours (those indexed by $j \in N_i$);

Update α_{ij}^{k+1} via (5.27) ;

end for

$k \leftarrow k+1$

Finite-consensus:

Set up $z^i(0)$ for r^k, $\|\boldsymbol{\alpha}^k\|$, $\|\Delta\boldsymbol{\alpha}^k\|$, $\|\boldsymbol{\lambda}^k\|$, and $\|\Delta\boldsymbol{\lambda}^k\|$

for all $i \in \mathbb{Z}^M$ (in parallel) **do**

Obtain $z^i(1), \cdots, z^i(T-1)$ using (3.40) and compute r^k and C^k from (3.39).

end for

until $r^k \leq \varepsilon M \mathbf{1}_{pN}$ and $C^k \leq \delta$

5.5 Recursive Feasibility and Stability

This section discusses the recursive feasibility and stability results of the proposed DMPC formulation. The overall MPC scheme is summarized in the following algorithm.

Algorithm 5.2: The synchronous MPC controller

1: Every system i measures it own state $x^i(t)$;
2: Every system i calls Algorithm 5.1 with $x^i(t)$ and obtain $\boldsymbol{u}^{i,\bar{k}(t)}$ as its output.
3: Every system obtains $\kappa^i(x(t))$ from $\boldsymbol{u}^{i,\bar{k}(t)}$ via (5.34) and apply $\kappa^i(x(t))$ to the i^{th} system.
4: Wait for next sampling time, let $t = t+1$ and go to step 1.

The next lemma pertains to a property of the terminal set of the overall system and is needed for stability of the closed-loop MPC system.

Lemma 5.2. *Let*

$$\bar{\sigma}^i := \min\{\sigma_\varepsilon^i, \frac{1}{M}\min_{\ell\in\mathbb{Z}^{Np}}\{b_\ell(\varepsilon)/h_{X_f^i}(\bar{F}_\ell^i)\}\} \tag{5.43}$$

where $h_{X_f^i}(\cdot)$ is the support function of X_f^i, $b_\ell(\varepsilon)$ denotes the ℓ^{th} element of $b(\varepsilon)$, $\bar{F}^i := F^iK_A^i + H^i$ from (5.16) with K_A^i defined by (5.44) below, and $\bar{F}_\ell^i$ denotes the ℓ^{th} row of $\bar{F}^i$. For any $x^i \in \bar{\sigma}^i X_f^i$, the optimal solution to Algorithm 1 for the i^{th} system is

$$\boldsymbol{u}^{i,\bar{k}} = \{u_0^{i,\bar{k}}, \cdots, u_{N-1}^{i,\bar{k}}\} = \{K^ix^i, K^iA_K^ix^i, \cdots, K^i(A_K^i)^{N-1}x^i\} := K_A^ix^i \tag{5.44}$$

with $\bar{k} = 1$.

The recursive feasible and stability results of the proposed DMPC approach are stated in the following theorem.

Theorem 5.3. *Suppose A5.1-A5.3 hold and $\mathbb{P}_\varepsilon(x(t))$ of (5.15) has a feasible solution at time t and that the MPC law of (5.34) is applied to the i^{th} system of (5.2) for all $i \in \mathbb{Z}^M$. Then, the following results hold:*

(i) $\mathbb{P}_\varepsilon(x(t+1))$ has a feasible solution at time $t+1$.

(ii) $|\sum_{i=1}^M \left(J^i(x^i(t), \boldsymbol{u}^{i,\bar{k}(t)}) - J^i(x^i(t), \boldsymbol{u}_t^{i*})\right)| \le \delta$, where $\{\boldsymbol{u}_t^{i*}\}_{i=1}^M$ is the optimal solution of (5.15).

(iii) Suppose δ is chosen such that $\{x^i : \|x^i\|_{Q^i}^2 \le \delta\} \subseteq int(\bar{\sigma}^i X_f^i)$ for all $i \in \mathbb{Z}^M$ where $\bar{\sigma}^i$ is as defined

in (5.43). Then, there exists a finite time t_f such that $x^i(t_f) \in \bar{\sigma}^i X_f^i$ for all $i \in \mathbb{Z}^M$.

(iv) Suppose the condition of (iii) holds, the closed-loop system (5.1) with the MPC law (5.34) is exponentially stable.

5.6 Numerical Results

The example chosen is a four-agent system where every agent has the same dynamics of $A^i = [1.1\ 1; 0\ 1.3], B^i = [1; 1]$ and same constraints $X^i := \{x^i \in \mathbb{R}^2 : \|x\|_\infty \leq 10\}$, $U^i := \{u^i \in \mathbb{R} : |u^i| \leq 1\}$. The coupled constraint is $|u^1| + 1.2|u^2| + 0.8|u^3| + 1.5|u^4| \leq 2.5$ and can be seen as a limit to the total amount of energy used. The values of K^i and P^i obtained from the discrete-time ARE, with $Q^i = I_2$ and $R^i = 0.1$, $i \in \mathbb{Z}^4$, are $K^i = [-0.6960\ \ -1.0664], P^i = [2.0819\ \ -0.2046; -0.2046\ 1.1944]$ for all $i \in \mathbb{Z}^4$. Consider the network connection of a ring and $W = I - 0.1L(G)$. The minimal polynomial of W is $t^3 - 2.4t^2 + 1.88t - 0.48 = 0$. Hence, value of T of (3.39) is 3 and checking of (5.42) can be done in 2 steps of (3.40). The initial conditions are, $x^i(0) = [-2.4583\ 1.1137]^T$, $i \in \mathbb{Z}^4$, and the horizon length $N = 5$. The performance of the proposed MPC approach is presented for several choices of ε and δ. The following table gives the scales $\{\sigma_\varepsilon^i\}_{i \in \mathbb{Z}^4}$ obtained from $\min\{\sum_{i=1}^M (1 - \sigma^i) : (5.13)\}$. Note that for all these choices of δ, $\{x^i : \|x^i\|_{Q^i}^2 \leq \delta\} \subseteq \bar{\sigma}^i X_f^i$ of (iii) of Theorem 5.3 holds.

ε	σ_ε^1	σ_ε^2	σ_ε^3	σ_ε^4
0.01	0.5310	0.4371	0.6248	0.2964
0.005	0.5779	0.4934	0.6623	0.3668
0.001	0.6154	0.5385	0.6923	0.4231

Table 5.1: The scales of the terminal sets for different choices of ε

The value of $\rho = 0.01$ and the values of $\bar{k}(t)$ along the trajectories starting from different initial states are given in the Table 5.2. Notice that for $t \geq 5$, $\bar{k}(t)$ are either 1 or 2 because the global constraints are not active. Obviously, $\bar{k}(t)$ increases when (ε, δ) decreases but the increase is relatively benign: a 10 times reduction of (ε, δ) results in a roughly 2.5 times increase in $\bar{k}(t)$.

While a comparison with existing approaches [44, 60, 121, 122] appears reasonable, this is not done because the problem settings are different: only connected network is needed for this approach while others need a fully connected (or a central node) network; bounds on system performance is guaranteed under the proposed approach but not others. Instead, comparison is made between

$(\varepsilon,\delta)\times 10^{-3}$	$t=0$	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$
$(10,10)$	238	243	197	269	2	2	2	1
$(5,5)$	279	287	306	377	2	2	2	1
$(1,1)$	400	874	452	433	184	2	1	1

Table 5.2: Values of $\bar{k}(t)$ along the trajectories starting from $x^i(0)=[-2.4583\ 1.1137]^T, i=1,..,4$ for different choices of ε and δ

the results of the proposed approach and that obtained by solving (5.15) with $\varepsilon=0$ using a single centralized computer, known as the centralized MPC (CMPC) solution. The terminal sets $\{\sigma_s^i X_f^i\}_{i=1}^M$ in CMPC are obtained from $\min\{\sum_{i=1}^M(1-\sigma^i):(5.13)\}$ with $\varepsilon=0$: $\sigma_s^1=0.6248, \sigma_s^2=0.5497, \sigma_s^3=0.6998$ and $\sigma_s^4=0.4372$. Table 5.3 shows the performances of the two approaches in terms of the infinite LQ cost over several initial $x(0)$: $\sum_{t=0}^{\infty}\sum_{i=1}^{M}(\|x^i(t)\|_{Q^i}^2+\|u^i(t)\|_{R^i}^2)$ where $x^i(t),u^i(t),t=0,\cdots,\infty$ are the state and control of the closed-loop system starting from $x(0)$. Entries in Table 5.3 are values of $J_{(\varepsilon,\delta)}(x)$, $J_o(x)$ and $\frac{|J_{(\varepsilon,\delta)}(x)-J_o(x)|}{J_o(x)}$, corresponding to the infinite LQ cost of the proposed approach, the optimal CMPC solution (with $\varepsilon=0$) and the relative cost in percentage respectively. As shown, the degradation in performance for $(\varepsilon,\delta)=(0.01,0.01)$ is $6.2\%\pm 2.5\%$, $(\varepsilon,\delta)=(0.005,0.005)$ is $2.8\%\pm 1.6\%$ and $(\varepsilon,\delta)=(0.001,0.001)$ is $0.8\%\pm 0.6\%$. These results suggest that 0.005 and 0.001 are sufficiently good low accuracy solution for the proposed approach.

$x^i, i \in \mathbb{Z}^M$	$(\varepsilon, \delta) \times 10^{-3}$	$J_{(\varepsilon,\delta)}(x)$	$\frac{\lvert J_{(\varepsilon,\delta)}(x) - J_o(x)\rvert}{J_o(x)}(\%)$	$J_o(x)$
$\begin{pmatrix} -3.288 \\ 0.4113 \end{pmatrix}$	$(10,10)$	125.0	4.27	119.9
	$(5,5)$	122.2	1.90	
	$(1,1)$	120.2	0.29	
$\begin{pmatrix} -1.7828 \\ -0.6326 \end{pmatrix}$	$(10,10)$	71.35	7.72	66.24
	$(5,5)$	68.79	3.84	
	$(1,1)$	67.11	1.31	
$\begin{pmatrix} -3.2847 \\ 0.971 \end{pmatrix}$	$(10,10)$	127.3	6.94	119.1
	$(5,5)$	122.5	2.87	
	$(1,1)$	119.6	0.49	
$\begin{pmatrix} 3.1377 \\ -0.1551 \end{pmatrix}$	$(10,10)$	119.9	4.53	114.7
	$(5,5)$	116.7	1.74	
	$(1,1)$	115.1	0.30	
$\begin{pmatrix} 2.3693 \\ 0.4091 \end{pmatrix}$	$(10,10)$	93.11	4.93	88.74
	$(5,5)$	90.80	2.33	
	$(1,1)$	89.09	0.40	
$\begin{pmatrix} 1.5779 \\ 0.7011 \end{pmatrix}$	$(10,10)$	64.56	8.71	59.39
	$(5,5)$	61.98	4.36	
	$(1,1)$	59.82	0.72	

Table 5.3: The cost difference of the proposed approach and the CMPC over various choices of the initial state $x(0)$.

5.7 Conclusions

A DMPC approach is proposed for a group of linear systems with local and global constraints. The proposed approach relies on the dual problem of the overall MPC problem and uses a distributed ADMM algorithm for its solution. This is made possible by introducing local copies of the dual variables in individual system and enforcing all the local copies to achieve approximate consensus value. Provision for computational expediency is made via early termination of the distributed ADMM algorithm where the inaccuracy depends on user-defined parameters. Termination conditions based on the these parameters are provided and is checked using a finite-time consensus algorithm. Under mild assumptions, this approach converges to some small neighborhood of the optimal so long as the network of systems is connected. Recursive feasibility and exponential stability of the closed-loop system are ensured.

5.A Proof of Theorem 5.1

(i) From the first order optimality condition of (5.32),

$$\begin{aligned}&\left(\nabla_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^{i,k+1})\right)^T(\boldsymbol{u}^i-\boldsymbol{u}^{i,k+1})\\&+(\lambda^{i,k+1})^T(F^i(\boldsymbol{u}^i-\boldsymbol{u}^{i,k+1})+s^i-s^{i,k+1})\geq 0\ \ \forall\,\boldsymbol{u}^i\in\mathcal{U}^i(x^i)\text{ and }\ \forall s^i\geq 0\end{aligned}\tag{5.45}$$

where $\lambda^{i,k+1}$ is that given by (5.33). Consider (5.15) and let its optimal solution be $\{\boldsymbol{u}^{i*}\}_{i\in\mathbb{Z}^M}$ with λ^* as the optimal dual variable of (5.15c). The first order optimality condition of (5.18), following standard saddle point theorem [149], is

$$\sum_{i=1}^{M}\left(\nabla_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^{i*})\right)^T(\boldsymbol{u}^i-\boldsymbol{u}^{i*})+\sum_{i=1}^{M}(\lambda^*)^T F^i(\boldsymbol{u}^i-\boldsymbol{u}^{i*})\geq 0\ \ \forall\,\boldsymbol{u}^i\in\mathcal{U}^i(x^i).\tag{5.46}$$

Let $\{\lambda^{i*}\}$ be the optimal solution of (5.23) and $\{y^{i*}\geq 0\}_{i\in\mathbb{Z}^M}$ be the optimal dual variable for the constraint $\lambda^i\geq 0$ in (5.23) satisfying $(\lambda^{i*})^T y^{i*}=0$. Then

$$(\lambda^{i*})^T(s^i-y^{i*})\geq 0\text{ for any }s^i\geq 0\tag{5.47}$$

Adding the above for $i=1,\cdots,M$ to (5.46) and noting that $\lambda^{i*}=\lambda^*$ for all $i\in\mathbb{Z}^M$ at the optimum solution of (5.23) yields

$$\sum_{i=1}^{M}\left(\nabla_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^{i*})\right)^T(\boldsymbol{u}^i-\boldsymbol{u}^{i*})+\sum_{i=1}^{M}(\lambda^*)^T\left(F^i(\boldsymbol{u}^i-\boldsymbol{u}^{i*})+s^i-y^{i*}\right)\geq 0,\tag{5.48}$$

for all $\boldsymbol{u}^i\in\mathcal{U}^i(x^i)$ and $s^i\geq 0$. Let $\boldsymbol{u}^i=\boldsymbol{u}^{i*}$ and $s^i=y^{i*}$ in (5.45) for $i=1,\cdots,M$ and adding them to (5.48) with $\boldsymbol{u}^i=\boldsymbol{u}^{i,k+1}$ and $s^i=s^{i,k+1}$ results in

$$\begin{aligned}&\sum_{i=1}^{M}\left(\nabla_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^{i,k+1})-\nabla_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^{i*})\right)^T(\boldsymbol{u}^{i,k+1}-\boldsymbol{u}^{i*})\\&+\sum_{i=1}^{M}(\lambda^{i,k+1}-\lambda^*)^T\left(F^i(\boldsymbol{u}^{i,k+1}-\boldsymbol{u}^{i*})+s^{i,k+1}-y^{i*}\right)\leq 0\end{aligned}\tag{5.49}$$

Since $J^i(x^i,\boldsymbol{u}^i)$ is strongly convex with respect to $\boldsymbol{u}^i$, there exist $\mu^i>0, i\in\mathbb{Z}^M$ such that

$$(\nabla_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^{i,k+1})-\nabla_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^{i*}))^T(\boldsymbol{u}^{i,k+1}-\boldsymbol{u}^{i*})\geq\mu^i\|(\boldsymbol{u}^{i,k+1}-\boldsymbol{u}^{i*})\|^2.$$

Using this fact in (5.49) yields

$$\sum_{i=1}^{M} \mu^i \|\boldsymbol{u}^{i,k+1} - \boldsymbol{u}^{i*}\|^2 \leq -\sum_{i=1}^{M} (\lambda^{i,k+1} - \lambda^*)^T \left(F^i(\boldsymbol{u}^{i,k+1} - \boldsymbol{u}^{i*}) + s^{i,k+1} - y^{i*} \right) \tag{5.50}$$

From (5.33), $F^i\boldsymbol{u}^{i,k+1} + s^{i,k+1}$ can be written as

$$F^i\boldsymbol{u}^{i,k+1} + s^{i,k+1} = 2\rho|N_i|\lambda^{i,k+1} - H^i x^i + \frac{b(\varepsilon)}{M} + v^{i,k} \tag{5.51}$$

Since $\{\lambda^{i*} = \lambda^*\}$ is the optimal solution of (5.23) and $\{y^{i*} \geq 0\}_{i\in\mathbb{Z}^M}$ is the optimal dual variable, they satisfy the KKT of (5.23)

$$\nabla g_i(\lambda^{i*}) + 2\sum_{j\in N_i} \alpha_{ij}^* - y^{i*} = -F^i\boldsymbol{u}^{i*} - H^i x^i + \frac{b(\varepsilon)}{M} + 2\sum_{j\in N_i} \alpha_{ij}^* - y^{i*} = 0 \tag{5.52}$$

Hence, $F^i(\boldsymbol{u}^{i,k+1} - \boldsymbol{u}^{i*}) + s^{i,k+1} - y^{i*}$ can be written in terms of $(\lambda^{i,k+1}, v^{i,k})$ and $\{\alpha_{ij}^* : j \in N_i\}$

$$\begin{aligned}
&F^i(\boldsymbol{u}^{i,k+1} - \boldsymbol{u}^{i*}) + s^{i,k+1} - y^{i*} \\
=&F^i\boldsymbol{u}^{i,k+1} + s^{i,k+1} - (F^i\boldsymbol{u}^{i*} + y^{i*}) \\
=&2\rho\|N_i\|\lambda^{i,k+1} + v^{i,k} - 2\sum_{j\in N_i} \alpha_{ij}^*
\end{aligned}$$

Consider the definition of $v^{i,k}$ of (5.28), the expression above can be rewritten as

$$\begin{aligned}
&2\rho\|N_i\|\lambda^{i,k+1} + v^{i,k} - 2\sum_{j\in N_i} \alpha_{ij}^* \\
=&2\rho\|N_i\|\lambda^{i,k+1} + 2\sum_{j\in N_i} \left(\alpha_{ij}^k - \rho(\lambda^{i,k} + \lambda^{j,k}) \right) - 2\sum_{j\in N_i} \alpha_{ij}^* \\
=&2\rho\|N_i\|\lambda^{i,k+1} + 2\sum_{j\in N_i} \left(\alpha_{ij}^{k+1} - \frac{\rho}{2}(\lambda^{i,k+1} - \lambda^{j,k+1}) \right) \\
&- \rho(|N_i|\lambda^{i,k} + \sum_{j\in N_i} \lambda^{j,k}) - 2\sum_{j\in N_i} \alpha_{ij}^* \\
=&2\sum_{j\in N_i} (\alpha_{ij}^{k+1} - \alpha_{ij}^*) + \rho\sum_{j\in N_i} (\lambda^{i,k+1} + \lambda^{j,k+1} - \lambda^{i,k} - \lambda^{j,k})
\end{aligned}$$

where the second equality is from the update of α_{ij}^k in (5.27). Using this in (5.50) yields

$$\begin{aligned}&\sum_{i=1}^{M}\mu^i\|\boldsymbol{u}^{i,k+1}-\boldsymbol{u}^{i*}\|^2\leq\\&-\sum_{i=1}^{M}(\lambda^{i,k+1}-\lambda^*)^T\left(2\sum_{j\in N_i}(\alpha_{ij}^{k+1}-\alpha_{ij}^*)+\rho\sum_{j\in N_i}(\lambda^{i,k+1}+\lambda^{j,k+1}-\lambda^{i,k}-\lambda^{j,k})\right)\end{aligned}\tag{5.53}$$

Consider the two terms on the right-hand side of (5.53). The first is

$$\begin{aligned}&2\sum_{i=1}^{M}\sum_{j\in N_i}(\lambda^{i,k+1}-\lambda^*)^T(\alpha_{ij}^{k+1}-\alpha_{ij}^*)\\=&\sum_{i=1}^{M}\sum_{j\in N_i}(\lambda^{i,k+1}-\lambda^{j,k+1})^T(\alpha_{ij}^{k+1}-\alpha_{ij}^*)\\=&\frac{2}{\rho}\sum_{i=1}^{M}\sum_{j\in N_i}(\alpha_{ij}^{k+1}-\alpha_{ij}^k)^T(\alpha_{ij}^{k+1}-\alpha_{ij}^*)\\=&\frac{2}{\rho}(\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^k)^T(\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^*)\\=&\frac{1}{\rho}(\|\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^*\|^2-\|\boldsymbol{\alpha}^k-\boldsymbol{\alpha}^*\|^2+\|\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^k\|^2)\end{aligned}\tag{5.54}$$

where the second last equality is from (5.27) and the last equality is due to the equality $(a-b)^T(a-c)=\frac{1}{2}\|a-b\|^2-\frac{1}{2}\|c-b\|^2+\frac{1}{2}\|a-c\|^2$. The second term of (5.53) is

$$\begin{aligned}&\rho\sum_{i=1}^{M}\sum_{j\in N_i}(\lambda^{i,k+1}-\lambda^*)^T(\lambda^{i,k+1}+\lambda^{j,k+1}-\lambda^{i,k}-\lambda^{j,k})\\=&\rho(\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*)^T\bar{\Gamma}(\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k)\\=&\frac{\rho}{2}(\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*\|_{\bar{\Gamma}}^2-\|\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^*\|_{\bar{\Gamma}}^2+\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|_{\bar{\Gamma}}^2)\end{aligned}\tag{5.55}$$

where $\bar{\Gamma}=(\mathcal{A}+D)\otimes I_{Np}$ and $\boldsymbol{\lambda}^*=I_M\otimes\lambda^*$. Combining (5.53), (5.54) and (5.55) yields

$$\begin{aligned}&\frac{1}{\rho}\|\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^*\|^2+\frac{\rho}{2}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*\|_{\bar{\Gamma}}^2\\\leq&\frac{1}{\rho}\|\boldsymbol{\alpha}^k-\boldsymbol{\alpha}^*\|^2+\frac{\rho}{2}\|\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^*\|_{\bar{\Gamma}}^2-\sum_{i=1}^{M}\mu^i\|\boldsymbol{u}^{i,k+1}-\boldsymbol{u}^{i*}\|^2\\&-\frac{1}{\rho}\|\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^k\|^2-\frac{\rho}{2}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|_{\bar{\Gamma}}^2\end{aligned}\tag{5.56}$$

Since $\frac{1}{\rho}\|\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^*\|^2+\frac{\rho}{2}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*\|_{\bar{\Gamma}}^2$ is lower bounded by 0, the following three conditions must

hold: $\boldsymbol{u}^{i,k+1} - \boldsymbol{u}^{i*}$ converges to 0, $\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^{k}$ converges to 0 and $\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^{k}$ converges to 0.

(ii) Convergence of $\boldsymbol{\alpha}^k$ is already shown in (i). To show convergence of r^k to 0, consider the optimality condition (5.45) of (5.32) and the fact that $J^i(x^i,\boldsymbol{u}^i) - J^i(x^i,\boldsymbol{u}^{i,k+1}) \geq \left(\nabla_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^{i,k+1})\right)^T (\boldsymbol{u}^i - \boldsymbol{u}^{i,k+1})$, it yields

$$\begin{aligned} & J^i(x^i,\boldsymbol{u}^i) - J^i(x^i,\boldsymbol{u}^{i,k+1}) \\ & + (\lambda^{i,k+1})^T (F^i(\boldsymbol{u}^i - \boldsymbol{u}^{i,k+1}) + s^i - s^{i,k+1}) \geq 0 \end{aligned} \tag{5.57}$$

for all $\boldsymbol{u}^i \in \mathcal{U}^i(x^i)$ and $s^i \geq 0$. By letting $\boldsymbol{u}^i = \boldsymbol{u}^{i*}$ and $s^i = y^{i*}$, we can obtain

$$\begin{aligned} & J^i(x^i,\boldsymbol{u}^{i*}) - J^i(x^i,\boldsymbol{u}^{i,k+1}) \\ & + (\lambda^{i,k+1})^T (F^i(\boldsymbol{u}^{i*} - \boldsymbol{u}^{i,k+1}) + y^{i*} - s^{i,k+1}) \geq 0 \end{aligned}$$

which can also be rewritten as

$$\begin{aligned} & J^i(x^i,\boldsymbol{u}^{i*}) - J^i(x^i,\boldsymbol{u}^{i,k+1}) \\ & + (\lambda^*)^T (F^i(\boldsymbol{u}^{i*} - \boldsymbol{u}^{i,k+1}) + y^{i*} - s^{i,k+1}) \\ & + (\lambda^{i,k+1} - \lambda^*)^T (F^i(\boldsymbol{u}^{i*} - \boldsymbol{u}^{i,k+1}) + y^{i*} - s^{i,k+1}) \geq 0 \end{aligned}$$

By using the same treatment on the second term as in the proof of (i), we arrive at the following inequality

$$\begin{aligned} & \Delta J^{k+1} + \sum_{i=1}^{M} (\lambda^*)^T (F^i(\boldsymbol{u}^{i,k+1} - \boldsymbol{u}^{i*}) + s^{i,k+1} - y^{i*}) \\ \leq & -\frac{2}{\rho} (\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k)^T (\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^*) - \rho (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*)^T \bar{\Gamma} (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k) \end{aligned}$$

Consider (5.52) and $\sum_{i=1}^{M}\sum_{j\in N_i}\alpha_{ij}^{*}=0$,

$$\begin{aligned}&\sum_{i=1}^{M}(\lambda^{*})^{T}(F^{i}(\boldsymbol{u}^{i,k+1}-\boldsymbol{u}^{i*})+s^{i,k+1}-y^{i*})\\=&\sum_{i=1}^{M}(\lambda^{*})^{T}(f^{i}(x^{i},\boldsymbol{u}^{i,k+1})+s^{i,k+1}-\frac{b(\varepsilon)}{M}-2\sum_{j\in N_i}\alpha_{ij}^{*})\\=&(\lambda^{*})^{T}r^{k+1}\end{aligned}$$

Consider the saddle point condition of (5.18), we know that

$$\Delta J^{k+1}+(\lambda^{*})^{T}r^{k+1}\geq\Delta J^{k+1}+(\lambda^{*})^{T}\left(\sum_{i=1}^{M}f^{i}(x^{i},\boldsymbol{u}^{i,k+1})-b(\varepsilon)\right)\geq 0$$

Therefore, we can get

$$-(\lambda^{*})^{T}r^{k+1}\leq\Delta J^{k+1}\leq-(\lambda^{*})^{T}r^{k+1}-\frac{2}{\rho}(\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^{*})^{T}\Delta\boldsymbol{\alpha}^{k+1}-\rho(\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^{*})^{T}\bar{\Gamma}(\Delta\boldsymbol{\lambda}^{k+1}) \quad (5.58)$$

From (5.33), (5.27) and the fact that $\sum_{i=1}^{M}\sum_{j\in N_i}\alpha_{ij}^{k+1}=0$,

$$\begin{aligned}r^{k+1}&=\sum_{i=1}^{M}(f^{i}(x^{i},\boldsymbol{u}^{i,k+1})+s^{i,k+1}-\frac{b(\varepsilon)}{M})\\&=\sum_{i=1}^{M}(2\rho|N_i|\lambda^{i,k+1}+2\sum_{j\in N_i}\alpha_{ij}^{k}-\rho\sum_{j\in N_i}(\lambda^{i,k}+\lambda^{j,k}))\\&=\sum_{i=1}^{M}(2\sum_{j\in N_i}\alpha_{ij}^{k+1}+\rho\sum_{j\in N_i}(\lambda^{i,k+1}+\lambda^{j,k+1}-\lambda^{i,k}-\lambda^{j,k}))\\&=\rho(\mathbf{1}_M\otimes I_{Np})^{T}\bar{\Gamma}(\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^{k}) \quad (5.59)\end{aligned}$$

The convergence of $\{\boldsymbol{\lambda}^{k}\}_{k=1}^{\infty}$ implies that $\{r^{k}\}_{k=1}^{\infty}$ converges to 0.

(iii) It follows from (5.58) that

$$\Delta J^{k+1}\geq-\|\lambda^{*}\|\|r^{k+1}\| \quad (5.60)$$

$$\Delta J^{k+1}\leq\|\lambda^{*}\|\|r^{k+1}\|+\frac{2}{\rho}\|\Delta\boldsymbol{\alpha}^{k+1}\|\|\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^{*}\|+\rho\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^{*}\|\|\bar{\Gamma}\|\|\Delta\boldsymbol{\lambda}^{k+1}\| \quad (5.61)$$

Hence,

$$|\Delta J^{k+1}| \leq \|\lambda^*\|\|r^{k+1}\| + \frac{2}{\rho}\|\Delta\boldsymbol{\alpha}^{k+1}\|\|\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^*\| + \rho\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|\|\bar{\Gamma}\|\|\Delta\boldsymbol{\lambda}^{k+1}\| \tag{5.62}$$

which implies that $|\Delta J^{k+1}| \leq C^{k+1}$. From the convergence of $\{\boldsymbol{\lambda}^k\}_{k=1}^{\infty}$, $\{\boldsymbol{\alpha}^k\}_{k=1}^{\infty}$, $\{\boldsymbol{u}^{i,k}\}_{k=1}^{\infty}$ and $\{r^k\}_{k=1}^{\infty}$, both ΔJ^k and C^k go to 0 as $k \to \infty$.

(iv) This result follows directly from (5.32). □

5.B The proof of Theorem 5.2

(i) Summing up (5.56) up to k, we can obtain

$$\begin{aligned}\sum_{\ell=1}^{k}\sum_{i=1}^{M}\mu^i\|\boldsymbol{u}^{i,\ell} - \boldsymbol{u}^{i*}\|^2 \leq \frac{1}{\rho}\|\boldsymbol{\alpha}^0 - \boldsymbol{\alpha}^*\|^2 + \frac{\rho}{2}\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|_{\bar{\Gamma}}^2 \\ - \frac{1}{\rho}\|\boldsymbol{\alpha}^k - \boldsymbol{\alpha}^*\|^2 - \frac{\rho}{2}\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|_{\bar{\Gamma}}^2\end{aligned}$$

Hence, from the convexity,

$$\begin{aligned}\sum_{i=1}^{M}\mu^i\|\tilde{\boldsymbol{u}}^{i,k} - \boldsymbol{u}^{i*}\|^2 &\leq \frac{1}{k}\sum_{\ell=1}^{k}\sum_{i=1}^{M}\mu^i\|\boldsymbol{u}^{i,\ell} - \boldsymbol{u}^{i*}\|^2 \\ &\leq \frac{1}{\rho k}\|\boldsymbol{\alpha}^0 - \boldsymbol{\alpha}^*\|^2 + \frac{\rho}{2k}\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|_{\bar{\Gamma}}^2\end{aligned}$$

Note that $\boldsymbol{\alpha}^0 = 0$ and $\boldsymbol{\lambda}^0 = 0$,

$$\|\tilde{\boldsymbol{u}}^{i,k} - \boldsymbol{u}^{i*}\|^2 \leq \frac{1}{k}\left(\frac{1}{\rho\mu^i}\|\boldsymbol{\alpha}^*\|^2 + \frac{\rho}{2\mu^i}\|\boldsymbol{\lambda}^*\|_{\bar{\Gamma}}^2\right)$$

(ii) Note that (5.57) can be rewritten as

$$\begin{aligned}&J^i(x^i, \boldsymbol{u}^{i*}) - J^i(x^i, \boldsymbol{u}^{i,k+1}) + \lambda^T(F^i(\boldsymbol{u}^{i*} - \boldsymbol{u}^{i,k+1}) + y^{i*} - s^{i,k+1}) \\ &+ (\lambda^{i,k+1} - \lambda)^T(F^i(\boldsymbol{u}^{i*} - \boldsymbol{u}^{i,k+1}) + y^{i*} - s^{i,k+1}) \geq 0\end{aligned}$$

for any λ. Consider the same treatment used in the proof of property (ii) of Theorem 5.1, we have

$$\begin{aligned}&\sum_{i=1}^{M}\left(J^i(x^i,\boldsymbol{u}^{i,k+1})-J^i(x^i,\boldsymbol{u}^{i*})+\lambda^T(F^i(\boldsymbol{u}^{i,k+1}-\boldsymbol{u}^{i*})+s^{i,k+1}-y^{i*})\right)\\ \leq&\frac{1}{\rho}\|\boldsymbol{\alpha}^k-\boldsymbol{\alpha}^*\|^2+\frac{\rho}{2}\|\boldsymbol{\lambda}^k-\boldsymbol{\lambda}\|_\Gamma^2-\frac{1}{\rho}\|\boldsymbol{\alpha}^{k+1}-\boldsymbol{\alpha}^*\|^2-\frac{\rho}{2}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}\|_\Gamma^2\end{aligned}$$

where $\boldsymbol{\lambda}=I_M\otimes\lambda$. Applying the same reasoning, we can obtain

$$\begin{aligned}&\sum_{i=1}^{M}\lambda^T(F^i(\boldsymbol{u}^{i,k+1}-\boldsymbol{u}^{i*})+s^{i,k+1}-y^{i*})\\ =&\lambda^T\left(\sum_{i=1}^{M}(f^i(\boldsymbol{u}^{i,k+1},x^i)+s^{i,k+1})-b(\varepsilon)\right)\end{aligned}$$

Therefore, we can get that

$$\begin{aligned}&\sum_{i=1}^{M}\left(J^i(x^i,\tilde{\boldsymbol{u}}^{i,k})-J^i(x^i,\boldsymbol{u}^{i*})\right)+\lambda^T\tilde{r}^k\\ \leq&\frac{1}{k}\sum_{\ell=1}^{k}\sum_{i=1}^{M}\left(J^i(x^i,\boldsymbol{u}^{i,\ell})-J^i(x^i,\boldsymbol{u}^{i*})\right)+\frac{1}{k}\sum_{\ell=1}^{k}\lambda^Tr^\ell\leq\frac{1}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2+\frac{\rho}{2}\|\boldsymbol{\lambda}\|_\Gamma^2\right)\end{aligned}\tag{5.63}$$

where $\tilde{r}^k=\sum_{i=1}^{M}(f^i(\tilde{\boldsymbol{u}}^{i,k},x^i)+\tilde{s}^{i,k})-b$. Following the choice of λ in [151], let $\lambda=\lambda^*+\tilde{r}^k/\|\tilde{r}^k\|$. Consider the saddle point condition of (5.18) and the fact that $(\lambda^*)^T\tilde{s}^{i,k}\geq 0$ for all $i\in\mathbb{Z}^M$, we know that

$$\begin{aligned}&\sum_{i=1}^{M}\left(J^i(x^i,\tilde{\boldsymbol{u}}^{i,k})-J^i(x^i,\boldsymbol{u}^{i*})\right)+(\lambda^*)^T\tilde{r}^k\\ \geq&\sum_{i=1}^{M}\left(J^i(x^i,\tilde{\boldsymbol{u}}^{i,k})-J^i(x^i,\boldsymbol{u}^{i*})\right)+(\lambda^*)^T\left(\sum_{i=1}^{M}f^i(\tilde{\boldsymbol{u}}^{i,k},x^i)-b(\varepsilon)\right)\geq 0\end{aligned}\tag{5.64}$$

Hence,

$$\begin{aligned}\|\tilde{r}^k\|\leq&\frac{1}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2+\frac{\rho}{2}\|\boldsymbol{\lambda}\|_\Gamma^2\right)\leq\frac{1}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2+\frac{\rho}{2}\lambda_{\max}(\Gamma)\|\boldsymbol{\lambda}\|^2\right)\\ \leq&\frac{1}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2+\frac{\rho}{2}\lambda_{\max}(\Gamma)M\|\lambda\|^2\right)\\ \leq&\frac{1}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2+\frac{\rho}{2}\lambda_{\max}(\Gamma)M(1+\|\lambda^*\|)^2\right)\end{aligned}$$

where $\Gamma = (\mathcal{A}+D)$. Therefore,

$$\begin{aligned}&\sum_{i=1}^{M} f^i(\tilde{\boldsymbol{u}}^{i,k},x^i) - b \leq \sum_{i=1}^{M}(f^i(\tilde{\boldsymbol{u}}^{i,k},x^i)+\tilde{s}^{i,k}) - b(\varepsilon)\\ \leq &\frac{1}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2 + \frac{\rho}{2}\lambda_{\max}(\Gamma)M(1+\|\lambda^*\|)^2\right)\mathbf{1}_{Np}\end{aligned}$$

(iii) From (5.64) in (ii), we know that

$$\begin{aligned}&\sum_{i=1}^{M}\left(J^i(x^i,\tilde{\boldsymbol{u}}^{i,k}) - J^i(x^i,\boldsymbol{u}^{i*})\right)\\ \geq &-\lambda^*\left(\sum_{i=1}^{M}(f^i(\tilde{\boldsymbol{u}}^{i,k},x^i)+\tilde{s}^{i,k}) - b(\varepsilon)\right)\\ \geq &-\|\lambda^*\|\|\sum_{i=1}^{M}(f^i(\tilde{\boldsymbol{u}}^{i,k},x^i)+\tilde{s}^{i,k}) - b(\varepsilon)\|\end{aligned}$$

From (5.63), we can observe that

$$\begin{aligned}&\sum_{i=1}^{M}\left(J^i(x^i,\tilde{\boldsymbol{u}}^{i,k}) - J^i(x^i,\boldsymbol{u}^{i*})\right)\\ \leq &\|\lambda\|\|\tilde{r}^k\| + \frac{1}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2 + \frac{\rho}{2}\|\boldsymbol{\lambda}\|_{\Gamma}^2\right)\\ \leq &(1+\|\lambda^*\|)\|\tilde{r}^k\| + \frac{1}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2 + \frac{\rho}{2}\|\boldsymbol{\lambda}\|_{\Gamma}^2\right)\\ \leq &\frac{2+\|\lambda^*\|}{k}\left(\frac{1}{\rho}\|\boldsymbol{\alpha}^*\|^2 + \frac{\rho}{2}\lambda_{\max}(\Gamma)M(1+\|\lambda^*\|)^2\right)\end{aligned}$$

which proves the result in (iii). □

5.C Proof of Lemma 5.1

The existence of $\bar{k}$ for the stated conditions follows from the convergence of r^k and $|\Delta J^k|$ of properties (ii) and (iii) of Theorem 5.1. Properties (iii) of Theorem 5.1 also implies $|\Delta J^{\bar{k}}| \leq \delta$ if $C^{\bar{k}} \leq \delta$. Also, $r^{\bar{k}} \leq \varepsilon M\mathbf{1}_{pN}$ implies $r^{\bar{k}} - \sum_{i=1}^{M} s^{i,\bar{k}} \leq \varepsilon M\mathbf{1}_{pN}$ as $s^{i,\bar{k}} \geq 0$ for all $i \in \mathbb{Z}^M$. It further implies that $\sum_{i=1}^{M} f^i(x^i,\boldsymbol{u}^{i,\bar{k}}) - b(\varepsilon) \leq \varepsilon M\mathbf{1}_{pN}$ since $r^{\bar{k}} - \sum_{i=1}^{M} s^{i,\bar{k}} = \sum_{i=1}^{M} f^i(x^i,\boldsymbol{u}^{i,\bar{k}}) - b(\varepsilon)$. □

5.D Proof of Lemma 5.2

By definition of $h_{\bar{\sigma}^i X_f^i}$, $\bar{F}_\ell^i x^i \le max_{y\in\bar{\sigma}^i X_f^i} \bar{F}_\ell^i y = h_{\bar{\sigma}^i X_f^i}(\bar{F}_\ell^i) = \bar{\sigma}^i h_{X_f^i}(\bar{F}_\ell^i)$ for any $x^i \in \bar{\sigma}^i X_f^i$ and any $\ell \in \mathbb{Z}^{Np}$. This fact, together with $\bar{\sigma}^i \le \frac{1}{M} b_\ell(\varepsilon)/h_{X_f^i}(\bar{F}_\ell^i)$ from the definition of $\{\bar{\sigma}^i\}_{i\in\mathbb{Z}^M}$, implies $\frac{b(\varepsilon)}{M} - \bar{F}^i x^i \ge 0$. Since $\bar{\sigma}^i X_f^i \subseteq \sigma_\varepsilon^i X_f^i$, $K_A^i x^i \in \mathcal{U}^i(x^i)$ from (5.5), (5.6) and (5.14). Hence, when $k=0$, $(K_A^i x^i, \frac{b(\varepsilon)}{M} - \bar{F}^i x^i)$ is a feasible solution to (5.32) because $v^{i,0} = 0$ from the initialization, $i \in \mathbb{Z}^M$. Since K^i and P^i are obtained from ARE, $K_A^i x^i$ is the optimal solution to $J^i(x^i, \boldsymbol{u}^i)$. Therefore, the solution when $k=0$ is $\boldsymbol{u}^{i,1} = K_A^i x^i$, and $s^{i,1} = \frac{b(\varepsilon)}{M} - \bar{F}^i x^i$ for all $i \in \mathbb{Z}^M$. Using these values in (5.35) and (5.33), it can be shown from (5.16) that $r^1 = \sum_{i=1}^M (f^i(x^i, K_A^i x^i) + s^{i,1}) - b(\varepsilon) = \sum_{i=1}^M (\bar{F}^i x^i + \frac{b(\varepsilon)}{M} - \bar{F}^i x^i) + b(\varepsilon) = 0$ and $\lambda^{i,1} = 0$ for all $i \in \mathbb{Z}^M$. This means that $\Delta\boldsymbol{\lambda}^1 = \boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^0 = 0$ and $\Delta\boldsymbol{\alpha}^1 = \boldsymbol{\alpha}^1 - \boldsymbol{\alpha}^0 = 0$ following the update of $\boldsymbol{\alpha}$ via (5.27). Hence, $r^1 = 0$ and $C^1 = 0$ and Algorithm 5.1 terminates at $\bar{k} = 1$. □

5.E Proof of Theorem 5.3

(i) As given by (5.34), let $\bar{k}(t)$ be the stopping iteration of the Algorithm 5.1 at time t and the solution of (5.32) be $\boldsymbol{u}^{i,\bar{k}(t)} := \boldsymbol{u}_t^i := \{u_{0|t}^i, u_{1|t}^i, \cdots, u_{N-1|t}^i\}$. Denote the associated state sequence as $\boldsymbol{x}_t^i = \{x_{0|t}^i, x_{1|t}^i, \cdots, x_{N|t}^i\}$ for all $i \in \mathbb{Z}^M$. Since $\{\boldsymbol{u}_t^i\}_{i=1}^M$ are the output of Algorithm 5.1, $\{(x^i(t), \boldsymbol{u}_t^i)\}_{i\in\mathbb{Z}^M}$ satisfy the stopping condition of (5.42) and is a (ε,δ)-relaxed solution of $\mathbb{P}_\varepsilon(x(t))$, or from (5.35), they satisfy

$$\sum_{i=1}^M f^i(x^i(t), \boldsymbol{u}_t^i) - b(\varepsilon) \le \varepsilon M \mathbf{1}_{pN} \tag{5.65}$$

Rewriting $f^i(x^i(t), \boldsymbol{u}_t^i)$ and $b(\varepsilon)$ back in terms of $\{x_{0|t}^i, x_{1|t}^i, \cdots, x_{N|t}^i\}$ and $\{u_{0|t}^i, u_{1|t}^i, \cdots, u_{N-1|t}^i\}$ (note that $f^i(x^i(t), \boldsymbol{u}_t^i) - b$ are simplified expression of (5.9)), (5.65) is equivalent to

$$\sum_{i=1}^M \Psi_x^i x_{\ell|t}^i + \Psi_u^i u_{\ell|t}^i \le (1 - \varepsilon M(\ell+1))\mathbf{1}_p + \varepsilon M \mathbf{1}_p = (1 - \varepsilon M \ell)\mathbf{1}_p, \ \ \forall \ell \in \mathbb{Z}_0^{N-1} \tag{5.66}$$

For all $i \in \mathbb{Z}^M$, let a feasible control to the i^{th} system at $t+1$ be chosen as

$$\hat{\boldsymbol{u}}_{t+1}^i := \{\hat{u}_{0|t+1}^i, \hat{u}_{1|t+1}^i, \cdots, \hat{u}_{N-1|t+1}^i\} := \{u_{1|t}^i, u_{2|t}^i, \cdots, u_{N-1|t}^i, K^i x_{N|t}^i\}. \tag{5.67}$$

and the associated state sequence

$$\hat{\boldsymbol{x}}^i_{t+1} = \{\hat{x}^i_{0|t+1}, \hat{x}^i_{1|t+1}, \cdots, \hat{x}^i_{N|t+1}\} := \{x^i_{1|t}, x^i_{2|t}, \cdots, x^i_{N|t}, (A^i + B^i K^i) x^i_{N|t}\}$$

It follows from this choice of $\hat{\boldsymbol{u}}^i_{t+1}$ and (5.66) that

$$\sum_{i=1}^{M} \Psi^i_x \hat{x}^i_{\ell|t+1} + \Psi^i_u \hat{u}^i_{\ell|t+1} = \sum_{i=1}^{M} \Psi^i_x x^i_{\ell+1|t} + \Psi^i_u u^i_{\ell+1|t} \leq (1 - \varepsilon M(1+\ell))\mathbf{1}_p, \ \forall \ell \in \mathbb{Z}_0^{N-2} \tag{5.68}$$

$$\text{and where } \ell = N-1, \tag{5.69}$$

$$\sum_{i=1}^{M} (\Psi^i_x x^i_{N|t} + \Psi^i_u K^i x^i_{N|t}) = \sum_{i=1}^{M} \bar{\Psi} x^i_{N|t} \leq (1 - \varepsilon MN)\mathbf{1}_p \tag{5.70}$$

where the last inequality follows the fact that $x^i_{N|t} \in \sigma^i_\varepsilon X^i_f$ and (5.10) (with $T^i_f = \sigma^i_\varepsilon X^i_f$). In addition, $\hat{u}^i_{\ell|t+1} \in U^i$ for $\ell \in \mathbb{Z}_0^{N-2}$ since $u^i_{\ell+1|t} \in U^i$ because of (iv) of Theorem 5.1. The last control, $\hat{u}^i_{N-1|t+1} = K^i x^i_{N|t} \in U^i$ because $x^i_{N|t} \in \sigma^i_\varepsilon X^i_f$ and $\sigma^i_\varepsilon X^i_f$ satisfies (5.6). The constraints of $\hat{x}^i_{\ell|t+1} \in X^i, \ell \in \mathbb{Z}_0^{N-1}$ and $\hat{x}^i_{N|t+1} \in \sigma^i_\varepsilon X^i_f$ follow similar argument. These properties implies $\hat{\boldsymbol{u}}^i_{t+1} \in \mathcal{U}(x^i(t+1))$, $i \in \mathbb{Z}^M$ and $\{\hat{\boldsymbol{u}}^i_{t+1}\}_{i=1}^M$ is a feasible solution to $\mathbb{P}_\varepsilon(x(t+1))$.

(ii) This result follows from Lemma 5.1 since $\{x^i(t), \boldsymbol{u}^{i,\bar{k}(t)}\}_{i=1}^M$ is a (ε, δ)-relaxed solution of $\mathbb{P}_\varepsilon(x(t))$.

(iii) Let $V_\varepsilon(x(t)) = \sum_{i=1}^M J^i(x^i(t), \boldsymbol{u}^{i*}_t)$ (where $\{\boldsymbol{u}^{i*}_t\}_{i=1}^M$ is the optimal solution of (5.15)) be the Lyapunov function of the closed-loop system of (5.1) with input $u^i(t) = u^{i,\bar{k}(t)}_0$ given by (5.34). When Algorithm 5.1 terminates at time t, $\boldsymbol{u}^{i,\bar{k}(t)} := \boldsymbol{u}^i_t$ and the stopping condition of (5.42) states that

$$|\sum_{i=1}^{M} J^i(x^i(t), \boldsymbol{u}^i_t) - V_\varepsilon(x(t))| \leq \delta \tag{5.71}$$

Let $\hat{\boldsymbol{u}}^i_{t+1}$ be as defined in (5.67) of property (i) above, it follows from the standard argument in MPC,

$$\begin{aligned} J^i(x^i(t+1), \hat{\boldsymbol{u}}^i_{t+1}) - J^i(x^i(t), \boldsymbol{u}^i_t) &= -\|x^i(t)\|^i_{Q^i} - \|u^i(t)\|^2_{R^i} + \|x^i_{N|t}\|^2_{Q^i} + \|K^i x^i_{N|t}\|^2_{R^i} \\ &\quad + \|A^i_K x^i_{N|t}\|^2_{P^i} - \|x^i_{N|t}\|^2_{P^i} \\ &= -\|x^i(t)\|^i_{Q^i} - \|u^i(t)\|^2_{R^i} \end{aligned} \tag{5.72}$$

where the last equality is from the fact K^i, P^i satisfy the Algebraic Riccatii Equation of $(A^i_K)^T P^i A^i_K -$

$P^i = -(Q^i + K^i R^i K^i)$. Since $\hat{\boldsymbol{u}}^i_{t+1}$ may not be the optimal at $t+1$,

$$V_\varepsilon(x(t+1)) \le \sum_{i=1}^{M} J^i(x^i(t+1), \hat{\boldsymbol{u}}^i_{t+1}) = \sum_{i=1}^{M} \left(J^i(x^i(t), \boldsymbol{u}^i_t) - \|x^i(t)\|^i_{Q^i} - \|u^i(t)\|^2_{R^i} \right)$$
$$\le V_\varepsilon(x(t)) + \delta - \sum_{i=1}^{M} \|x^i(t)\|^2_{Q^i} = V_\varepsilon(x(t)) + \theta(t) \tag{5.73}$$

where the equality condition follows from (5.72) and the last inequality is due to (5.71). From the stated condition, choose any $\tilde{\varepsilon} > 0$ such that $\{x^i : \|x^i\|^2_{Q^i} \le \delta + \tilde{\varepsilon}\} \subseteq \bar{\sigma}^i X^i_f$. Consider the following two cases for (5.73): $\theta(t) < -\tilde{\varepsilon}$ for all t and $\theta(t) \ge -\tilde{\varepsilon}$ for at least one t. Suppose $\theta(t) < -\tilde{\varepsilon}$ for all t, then $\lim_{t\to\infty} V_\varepsilon(x(t))$ becomes negative and leads to a contradiction of non-negativity of $V_\varepsilon(x(t))$. Suppose there exists one t, denoted by t_f such that $\theta(t_f) \ge -\tilde{\varepsilon}$ or $\sum_{i\in\mathbb{Z}^M} \|x^i(t_f)\|^2_{Q^i} \le \delta + \tilde{\varepsilon}$. This implies that $\|x^i(t_f)\|^2_{Q^i} \le \delta + \tilde{\varepsilon}$ which, together with $\{x^i : \|x^i\|^2_{Q^i} \le \delta + \tilde{\varepsilon}\} \subseteq \bar{\sigma}^i X^i_f$, implies that $x^i(t_f) \in \bar{\sigma}^i X^i_f$ for all $i \in Z^M$.

(iv) Property (iii) states that a finite t_f exists such that $x^i(t_f) \in \bar{\sigma}^i X^i_f$, $i \in \mathbb{Z}^M$. When this happens, it follows from Lemma 5.2 that $\kappa^i(x(t_f)) = K^i x^i(t_f)$ and the closed-loop system becomes $x^i(t_f+1) = A^i_K x^i(t_f)$. Since $x^i(t_f+1) \in \bar{\sigma}^i X^i_f$ for any $x^i(t_f) \in \bar{\sigma}^i X^i_f$ from (5.6). As a result, $x^i(t+1) = A^i_K x^i(t)$ for all $t \ge t_f$ and the closed-loop system is exponentially stable. □

CHAPTER 6

A Distributed Fast Dual Gradient Algorithm for Distributed Model Predictive Control with Coupled Constraints

6.1 Introduction

This chapter proposes an accelerated DMPC approach for M discrete-time linear dynamical systems, given by (5.1)-(5.3). Like the approach in Chapter 5, this approach also solves the dual problem of the overall MPC problem and converts the dual problem into a distributed consensus optimization problem (DCOP). In Chapter 5, the ADMM is used for the DCOP problem. However, the ADMM can have slow convergence for highly accurate solutions. This work is motivated by the need for a faster solution of the DCOP problem on a connected network (not necessarily fully connected) and is a distributed implementation of the standard stand-alone Nestrov gradient method [97, 98]. The Nestrov gradient method for single MPC can be found in [176, 177]. The distributed Nestrov gradient implementations of DCOP have appeared [106, 107] but not for MPC. For distributed implementations, each system has a local copy of the dual variable. In [106], a constant step-size update for the iterates is used but such a choice does not ensure convergence to the optimal consensus variable. The work of [107] uses an inner loop for better convergence of the consensus variable. However, the number of consensus steps in the inner loop grows with the index of the outer loop and this leads to a significant increase in the information exchanges. Similar to [107], this chapter uses several consensus steps in the inner-loop to achieve the consensus of the local copies. However, the number of consensus steps is fixed and does not grow with the outer-loop index. Despite this, the computations of the proposed approach can be high for large-scale problems. This issue

is minimized by having premature termination of the proposed approach. Convergence of the proposed approach under the premature termination condition, together with recursive feasibility and stability of the closed-loop system are provided. Under reasonable assumptions, the approach converges faster than the ADMM approach [185] for the same accuracy. An important difference of this work to [185] is that the local copies of the dual variable reach exact consensus while the approach of [185] achieves only approximate consensus at each iteration. This feature, together with the premature termination consideration allows the proposed approach to generate a solution whose cost is upper bounded by the optimal cost and to have a simplified stopping condition. A comparison with [185] using an example is also provided. Not accounting for the information exchanges, the results from this and other examples show that the number of iterations needed to reach the same accuracy by the proposed approach is about 30% to 50% of that needed by the ADMM approach of [185].

The rest of this chapter is organized as follows. Section 6.2 reviews the dual formulation of the overall MPC problem. Section 6.3 presents the proposed approach and its convergence results. The recursive feasibility and stability results are given in Section 6.5. The performance of the approach is illustrated by a numerical example in Section 6.6 with the conclusions given in Section 6.7. The proofs are given in the appendices. The notations used in this chapter follows those in Chapter 5. Additional notations are introduced as required in the text.

6.2 Preliminaries

This section reviews the dual problem of the overall MPC problem (5.15). Let $\lambda \in \mathbb{R}^{Np}$ be the dual variable associated with the coupled constraint (5.15c). The Lagrangian of (5.15) is $\mathcal{L}(\{\boldsymbol{u}^i\},\lambda) = \sum_{i=1}^{M} J^i(x^i,\boldsymbol{u}^i) + \lambda^T(\sum_{i=1}^{M} f^i(x^i,\boldsymbol{u}^i) - b(\varepsilon))$ for all $\boldsymbol{u}^i \in \mathcal{U}^i(x^i), i \in \mathbb{Z}^M$ and the dual problem is

$$\max_{\lambda \geq 0} \Phi(x,\lambda) \tag{6.1}$$

where $\Phi(x,\lambda) = \min_{\{\boldsymbol{u}^i \in \mathcal{U}^i(x^i), i\in\mathbb{Z}^M\}} \mathcal{L}(\{\boldsymbol{u}^i\},\lambda)$. This dual problem is also equivalent to

$$\min_{\lambda\geq 0} \max_{\{\boldsymbol{u}^i \in \mathcal{U}^i(x^i), i\in\mathbb{Z}^M\}} -\mathcal{L}(\{\boldsymbol{u}^i\},\lambda) := \min_{\lambda\geq 0} \sum_{i=1}^{M} g_i(\lambda) \tag{6.2}$$

where

$$g_i(\lambda) := \max_{\boldsymbol{u}^i \in \mathcal{U}^i(x^i)} -J^i(x^i,\boldsymbol{u}^i) - \lambda^T (f^i(x^i,\boldsymbol{u}^i) - \frac{b(\varepsilon)}{M}). \tag{6.3}$$

Let $\boldsymbol{u}^i(\lambda) = \arg\max_{\boldsymbol{u}^i \in \mathcal{U}^i(x^i)} g_i(\lambda)$. Then, it can be verified that $g_i(\lambda)$ is convex and the gradient of $g_i(\lambda)$ is $\nabla g_i(\lambda) = -(f^i(x^i,\boldsymbol{u}^i(\lambda)) - \frac{b(\varepsilon)}{M})$ (see Danskin's Theorem of [149]). In addition, $\nabla g_i(\lambda)$ is Lipschitz continuous with Lipschitz constant $\frac{\|F^i\|^2}{\mu^i}$, where $\mu_i > 0$ is such that $\nabla^2_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^i) \geq \mu_i I$ for all x^i and $\boldsymbol{u}^i$ (see [100]). Let $L_g = \max_{i\in\mathbb{Z}^M}\{\frac{\|F^i\|^2}{\mu^i}\}$. Note that while the optimal solution of (5.15) is unique as $J^i(x^i,\boldsymbol{u}^i)$ is stictly convex in $\boldsymbol{u}^i$, the optimal λ of (6.2) may not be [177, 179]. Let

$$\Lambda(x) = \{\lambda : \lambda \text{ is an optimal solution of } (6.1)\} \tag{6.4}$$

be the collection of all possible optimal λ. Despite the nonuniqueness of λ, the proposed algorithm (Algorithm 5.1) will converge to an unique solution of (5.15), see Theorem 6.1.

6.3 The Main Results

This section discusses the proposed distributed fast dual gradient algorithm. The overall problem (5.15) is considered as the primal problem.

6.3.1 Distributed Fast Dual Gradient Algorithm

The standard stand-alone Nesterov gradient algorithm [97, 98, 177] for $\min_{\lambda\geq 0}\sum_{i=1}^{M} g_i(\lambda)$ of (6.2) consists of the following iterates

$$\tilde{\lambda}^k = \lambda^k + \theta_k(\theta_{k-1}^{-1}-1)(\lambda^k-\lambda^{k-1}) \tag{6.5a}$$

$$\lambda^{k+1} = [\tilde{\lambda}^k - \frac{1}{L_g}\sum_{i=1}^{M}\nabla g_i(\tilde{\lambda}^k)]_+ \tag{6.5b}$$

$$\theta_{k+1} = (\sqrt{\theta_k^4+4\theta_k^2}-\theta_k^2)/2 \tag{6.5c}$$

where $[x]_+ = \max\{0,x\}$, and $\lambda^{-1}=\lambda^0=0$ and $\theta_{-1}=\theta_0=1$ are the needed initializations. From (6.5c), the sequence $\{\theta_k\}$ satisfies [177]

$$\frac{1-\theta_{k+1}}{\theta_{k+1}^2} = \frac{1}{\theta_k^2},\ \ \frac{1}{\theta_k^2} = \sum_{\ell=0}^{k}\theta_\ell^{-1},\ \text{ and } \theta_k \leq \frac{2}{k+2} \tag{6.6}$$

Note that (6.5b) requires the gradients from all M systems. In order to implement a fully distributed computation, each system i makes a local copy of λ, λ^i. Correspondingly, (6.5a) and (6.5b) are replaced by

$$\tilde{\lambda}^{i,k} = \lambda^{i,k} + \theta_k(\theta_{k-1}^{-1}-1)(\lambda^{i,k}-\lambda^{i,k-1}),\ \ \forall i\in\mathbb{Z}^M \tag{6.7a}$$

$$\lambda^{i,k+1} = [\frac{1}{M}\sum_{i=1}^{M}\left(\tilde{\lambda}^{i,k} - \frac{1}{L_g}\nabla g_i(\tilde{\lambda}^{i,k})\right)]_+,\ \ \forall i\in\mathbb{Z}^M \tag{6.7b}$$

where $\nabla g_i(\tilde{\lambda}^{i,k}) = -(f^i(x^i,\tilde{\boldsymbol{u}}^{i,k}) - \frac{b}{M})$ with

$$\tilde{\boldsymbol{u}}^{i,k} = \arg\min_{\boldsymbol{u}^i\in\mathcal{U}^i(x^i)} J^i(x^i,\boldsymbol{u}^i) + (\tilde{\lambda}^{i,k})^T(f^i(x^i,\boldsymbol{u}^i) - \frac{b(\varepsilon)}{M}). \tag{6.8}$$

For all $i\in\mathbb{Z}^M$, let

$$\bar{\boldsymbol{u}}^{i,k} := \theta_k^2\sum_{\ell=0}^{k}\theta_\ell^{-1}\tilde{\boldsymbol{u}}^{i,\ell} = (1-\theta_k)\bar{\boldsymbol{u}}^{i,k-1} + \theta_k\tilde{\boldsymbol{u}}^{i,k} \tag{6.9}$$

where $\bar{\boldsymbol{u}}^{i,-1}=0$ and $\tilde{\boldsymbol{u}}^{i,k}$ is obtained from (6.8). Note that (6.7b) requires the quantity $\left(\tilde{\lambda}^{i,k} - \frac{1}{L_g}\nabla g_i(\tilde{\lambda}^{i,k})\right)$ from all $i\in\mathbb{Z}^M$ and, hence, is not fully distributed. To handle this problem, the finite-time consensus

algorithm mentioned in Section 3.3.2 is used to compute the quantity $\frac{1}{M}\sum_{i=1}^{M}\left(\tilde{\lambda}^{i,k}-\frac{1}{L_g}\nabla g_i(\tilde{\lambda}^{i,k})\right)$ in (6.7b). Specifically, for each $i\in\mathbb{Z}^M$, introduce variable $y^i(\ell,k)$ with

$$y^i(0,k):=\tilde{\lambda}^{i,k}-\frac{1}{L_g}\nabla g_i(\tilde{\lambda}^{i,k}) \tag{6.10}$$

$$y^i(\ell+1,k)=W^{ii}y^i(\ell,k)+\sum_{j\in N_i}W^{ij}y^j(\ell,k),\quad \ell\in\mathbb{Z}_0^{T-2}, \tag{6.11}$$

where T is the order of the minimal polynomial of W and (6.11) is the i^{th} component of the consensus dynamics given by (3.37). From (3.39), $\sum_{\ell=0}^{T-1}\tau_\ell y^i(\ell,k)=\frac{1}{M}\sum_{i=1}^{M}\left(\tilde{\lambda}^{i,k}-\frac{1}{L_g}\nabla g_i(\tilde{\lambda}^{i,k})\right)$ for all $i\in\mathbb{Z}^M$. With this property, (6.7b) is replaced by

$$\lambda^{i,k+1}=[\sum_{\ell=0}^{T-1}\tau_\ell y^i(\ell,k)]_+,\quad i\in\mathbb{Z}^M \tag{6.12}$$

This process is then repeated by incrementing k. The stopping criterion of this distributed fast dual gradient algorithm is discussed in Section 6.4 and suppose it terminates at iteration $\bar{k}$. Then, the solution from this algorithm is $\bar{\boldsymbol{u}}^{i,\bar{k}}:=\{\bar{u}_0^{i,\bar{k}},\bar{u}_1^{i,\bar{k}},\cdots,\bar{u}_{N-1}^{i,\bar{k}}\}$, as defined by (6.9), $i\in\mathbb{Z}^M$. Correspondingly, the MPC control law applied on the i^{th} system is

$$\kappa^i(x)=\bar{u}_0^{i,\bar{k}},i\in\mathbb{Z}^M \tag{6.13}$$

6.3.2 Convergence Analysis

The convergence results of the distributed fast dual gradient algorithm are discussed in this section. The convergence to an optimal dual solution is stated as follows.

Lemma 6.1. *For any $x\in\mathcal{D}_\varepsilon$, let $\{\lambda^{i,k},\tilde{\lambda}^{i,k}\}_{i=1}^M$ be generated from (6.7a) and (6.7b) with $\lambda^{i,-1}=\lambda^{i,0}=0$. Then, for any $\lambda^*\in\Lambda(x)$, the following results hold.*
(i) For any $k\geq 0$, the objective $\sum_{i=1}^{M}g_i(\lambda^{i,k+1})$ is bounded by

$$0\leq\sum_{i=1}^{M}g_i(\lambda^{i,k+1})-\sum_{i=1}^{M}g_i(\lambda^*)\leq\frac{L_gM\theta_k^2}{2}\|\lambda^*\|^2 \tag{6.14}$$

(ii) Let $\bar{\lambda}^{i,k} := \lambda^{i,k-1} + \theta_{k-1}^{-1}(\lambda^{i,k} - \lambda^{i,k-1})$ *for all* $k \geq 0$ *and* $i \in \mathbb{Z}^M$. *The sequence* $\{\bar{\lambda}^{i,k}\}$ *satisfies*

$$\sum_{i=1}^{M} \|\bar{\lambda}^{i,k+1} - \lambda^*\|^2 \leq M\|\lambda^*\|^2 \tag{6.15}$$

(iii) Consider the sequences $\{\tilde{\boldsymbol{u}}^{i,k}\}_{i=1}^M$ *and* $\{\bar{\boldsymbol{u}}^{i,k}\}_{i=1}^M$ *generated from (6.8) and (6.8) respectively. For any* $k \geq 0$, *the coupled constraint using* $\{\bar{\boldsymbol{u}}^{i,k}\}_{i=1}^M$ *satisfies the inequality*

$$\sum_{i=1}^{M} f^i(x^i, \bar{\boldsymbol{u}}^{i,k}) - b \leq \left(\frac{4M(\sqrt{M}+1)L_g\|\lambda^*\|}{(k+2)^2}\right)\mathbf{1}_{Np} \tag{6.16}$$

Property (iii) of Lemma 6.1 provides the decreasing upper bound on the violation of the coupled constraint. On the basis of the convergence of the dual variable, the primal convergence result is stated in the following theorem, which is a modification of Theorem 5 in [177].

Theorem 6.1. *For any* $x \in \mathcal{D}_\varepsilon$, *suppose* $\{\boldsymbol{u}^{i*}\}_{i=1}^M$ *is the optimal solution of* $\mathbb{P}_\varepsilon(x)$. *Then, for any* $k \geq 0$ *and* $\lambda^* \in \Lambda(x)$, *it holds that*

$$-\frac{4M(\sqrt{M}+1)L_g\|\lambda^*\|^2}{(k+2)^2} \leq \sum_{i=1}^{M} J^i(x^i, \bar{\boldsymbol{u}}^{i,k}) - \sum_{i=1}^{M} J^i(x^i, \boldsymbol{u}^{i*}) \leq 0 \tag{6.17}$$

6.3.3 Primal Suboptimality and Feasibility

As mentioned before, a premature termination condition is used to reduce the computational load for the solution of (6.1). For this purpose, the relaxed solution of (5.15) is defined as follows.

Definition 6.1. *Given any* $\varepsilon > 0$, *the set* $\{x^i, \boldsymbol{u}^i\}_{i=1}^M$ *is a* ε*-relaxed solution of (5.15) if*

$$\boldsymbol{u}^i \in \mathcal{U}^i(x^i), i \in \mathbb{Z}^M, \quad \sum_{i=1}^{M} f^i(x^i, \boldsymbol{u}^i) - b(\varepsilon) \leq \varepsilon M \mathbf{1}_{pN} \tag{6.18}$$

where $\{\boldsymbol{u}^{i*}\}_{i=1}^M$ *is the optimal solution of (5.15). In addition, for any* $\varepsilon, \delta > 0$, *the set* $\{x^i, \boldsymbol{u}^i\}_{i=1}^M$ *is a* (ε, δ)*-suboptimal solution of (5.15) if it is a* ε*-relaxed solution and*

$$\sum_{i=1}^{M} \left((J^i(x^i, \boldsymbol{u}^i) - J^i(x^i, \boldsymbol{u}^{i*})\right) \leq \delta \tag{6.19}$$

The following lemma discusses the existence of the suboptimal solution.

Lemma 6.2. *For any $x \in \mathcal{D}_\varepsilon$, let $\{\tilde{\boldsymbol{u}}^{i,k}\}_{i=1}^M$ and $\{\bar{\boldsymbol{u}}^{i,k}\}_{i=1}^M$ be generated from (6.8) and (6.9) respectively. Then, it holds that:*

(i) there exists a finite k such that $\{x^i, \bar{\boldsymbol{u}}^{i,k}\}_{i=1}^M$ is a $(\varepsilon, 0)$-suboptimal solution of (5.15);

(ii) $\{x^i, \bar{\boldsymbol{u}}^{i,k}\}_{i=1}^M$ is a $(\varepsilon, 0)$-suboptimal if and only if it is a ε-relaxed solution.

Proof of Lemma 6.2: (i) Since $\tilde{\boldsymbol{u}}^{i,k} \in \mathcal{U}^i(x^i)$ for all $i \in \mathbb{Z}^M$ and k from (6.8), it holds that $\bar{\boldsymbol{u}}^{i,k} \in \mathcal{U}^i(x^i)$ for all $i \in \mathbb{Z}^M$ and k. From property (iii) of Lemma 6.1, there always exits a k such that (6.18) is satisfied.

(ii) The equivalence between the $(\varepsilon, 0)$-suboptimal solution and the ε-relaxed solution follows from Definition 6.1 and Theorem 6.1. □

The next theorem shows the existence of a ε-relaxed solution ensures the recursive feasibility of (5.15).

Theorem 6.2. *Suppose $\{x^i, \boldsymbol{u}^i\}_{i=1}^M$ is a ε-relaxed solution of (5.15) as defined by Definition 6.1 with $\boldsymbol{u}^i = \{u_0^i, u_1^i, \cdots, u_{N-1}^i\}$ for all $i \in \mathbb{Z}^M$. Let the state sequence associated with this solution be $\{x_0^i, x_1^i, \cdots, x_N^i\}$, $x^{i+} = A^i x^i + B^i u_0^i$ and $\boldsymbol{u}^{i+} = \{u_1^i, \cdots, u_{N-1}^i, K^i x_N^i\}$ for all $i \in Z^M$. Then, the following results hold.*

(i) $\{\boldsymbol{u}^{i+}\}_{i=1}^M$ is a feasible solution to $\mathbb{P}_\varepsilon(x^+)$.

(ii) Consider the solution of $\mathbb{P}_\varepsilon(x^+)$ and let $\{\bar{\boldsymbol{u}}^{i,k}\}_{i=1}^M$ be generated from (6.8) and (6.9) with the states $\{x^{i+}\}_{i=1}^M$. Then, there exists a finite k such that $\{x^{i+}, \bar{\boldsymbol{u}}^{i,k}\}_{i=1}^M$ is a ε-relaxed solution of (5.15).

6.4 The Overall DMPC Scheme

The overall DMPC scheme is now presented in this section. First, a proper stopping criterion for the distributed fast dual gradient algorithm is needed. This condition is based on the results of Lemma 6.2. Specifically, the algorithm terminates at the first k, denoted as $\bar{k}$, such that a ε-relaxed solution is achieved. Following Definition 6.1, the stopping criterion corresponds to $\sum_{i=1}^M f^i(x^i, \bar{\boldsymbol{u}}^{i,k}) - b(\varepsilon) \leq \varepsilon M \mathbf{1}_{pN}$. This condition should be checked in a fully distributed manner. Again, the finite-time

consensus algorithm in Section 3.3.2 is used. For each $i \in \mathbb{Z}^M$, introduce the variable $z^i(\ell,k)$ with

$$z^i(0,k) := f^i(x^i,\bar{\boldsymbol{u}}^{i,k}) - \frac{b(\varepsilon)}{M} \tag{6.20}$$

$$z^i(\ell+1,k) = W^{ii}z^i(\ell,k) + \sum_{j\in N_i} W^{ij}z^j(\ell,k), \quad \ell \in \mathbb{Z}_0^{T-2} \tag{6.21}$$

The stopping criterion is satisfied if $\sum_{\ell=0}^{T-1}\tau_\ell z^i(\ell,k) \le \varepsilon$ as $\sum_{\ell=0}^{T-1}\tau_\ell z^i(\ell,k) = \frac{1}{M}\left(\sum_{i=1}^{M} f^i(x^i,\bar{\boldsymbol{u}}^{i,k}) - b(\varepsilon)\right)$ for all $i \in \mathbb{Z}^M$. The distributed fast dual gradient algorithm with the finite-time consensus is summarized in Algorithm 6.1.

Algorithm 6.1: Distributed fast dual gradient algorithm

Input: x^i, $i \in \mathbb{Z}^M$

Output: $\bar{\boldsymbol{u}}^{i,\bar{k}}$, $i \in \mathbb{Z}^M$

Initialization: set $k=0$, $\bar{\boldsymbol{u}}^{i,-1}=0, \lambda^{i,-1}=\lambda^{i,0}=0$ and $\theta_{-1}=\theta_0=1$, for all $i \in \mathbb{Z}^M$;

repeat

for all $i \in \mathbb{Z}^M$ (in parallel) **do**

Obtain $\tilde{\lambda}^{i,k}$ and $\tilde{\boldsymbol{u}}^{i,k}$ from (6.7a) and (6.8) respectively;

Perform the finite-consensus steps in (6.11) with $y^i(0,k)$ being given in (6.10);

Obtain $\lambda^{i,k+1}$ from (6.12);

end for

Set up $z^i(0,k)$ using $\bar{\boldsymbol{u}}^{i,k}$ which is defined in (6.9), $i \in \mathbb{Z}^M$

for all $i \in \mathbb{Z}^M$ (in parallel) **do**

Obtain $z^i(1,k),\cdots,z^i(T-1,k)$ using (6.20) and (6.21);

end for

$\theta_{k+1} = (\sqrt{\theta_k^4+4\theta_k^2} - \theta_k^2)/2$;

$k \leftarrow k+1$

until $\sum_{\ell=0}^{T-1}\tau_\ell z^i(\ell,k) \le \varepsilon$

The overall procedure of the DMPC algorithm is summarized in Algorithm 6.2.

Algorithm 6.2: The overall DMPC algorithm

1: At time t, every system i measures it own state $x^i(t)$;

2: Every system i calls Algorithm 5.1 with $x^i(t)$ and obtain $\bar{\boldsymbol{u}}^{i,\bar{k}(t)}$ as its output.

3: Every system obtains $\kappa^i(x(t))$ from $\bar{\boldsymbol{u}}^{i,\bar{k}(t)}$ via (6.13) and apply $\kappa^i(x(t))$ to the i^{th} system.

4: Wait for next sampling time, let $t=t+1$ and go to step 1.

6.5 Recursive Feasibility and Stability

This section discusses the recursive feasibility and stability results of the proposed DMPC formulation. The next lemma pertains to a property of the terminal set of the overall system and is needed for stability of the closed-loop MPC system.

Lemma 6.3. *Let*

$$\bar{\sigma}^i := \min\{\sigma_\varepsilon^i, \frac{1}{M}\min_{\ell\in\mathbb{Z}^{Np}}\{b_\ell(\varepsilon)/h_{X_f^i}(\bar{F}_\ell^i)\}\} \tag{6.22}$$

where $h_{X_f^i}(\cdot)$ is the support function of X_f^i, $b_\ell(\varepsilon)$ denotes the ℓ^{th} element of $b(\varepsilon)$, $\bar{F}^i := F^iK_A^i + H^i$ from (5.16) with K_A^i defined by (6.23) below, and $\bar{F}_\ell^i$ denotes the ℓ^{th} row of $\bar{F}^i$. For any $x^i \in \bar{\sigma}^i X_f^i$, the optimal solution to Algorithm 5.1 for the i^{th} system is

$$\bar{\boldsymbol{u}}^{i,\bar{k}} = \{\bar{u}_0^{i,\bar{k}}, \cdots, \bar{u}_{N-1}^{i,\bar{k}}\} = \{K^ix^i, K^iA_K^ix^i, \cdots, K^i(A_K^i)^{N-1}x^i\} := K_A^ix^i \tag{6.23}$$

with $\bar{k} = 0$.

The recursive feasible and stability results of the proposed DMPC approach are stated in the following theorem.

Theorem 6.3. *Suppose A5.1-A5.3 hold and $\mathbb{P}_\varepsilon(x(t))$ of (5.15) has a feasible solution at time t and that the MPC law of (6.13) is applied to the i^{th} system of (5.2) for all $i \in \mathbb{Z}^M$. Then, the following results hold.*

(i) $\mathbb{P}_\varepsilon(x(t+1))$ has a feasible solution at time $t+1$.

(ii) For all $t \geq 0$, $\sum_{i=1}^M \left(J^i(x^i(t), \bar{\boldsymbol{u}}^{i,\bar{k}(t)}) - J^i(x^i(t), \boldsymbol{u}_t^{i*})\right) \leq 0$, where $\{\boldsymbol{u}_t^{i*}\}_{i=1}^M$ is the optimal solution of (5.15).

(iii) The closed-loop system (5.1) with the MPC law (6.13) is exponentially stable.

Remark 6.1. *It can be shown that the true LQ cost is upper bounded by the predicted cost of the initial state. For any $x \in \mathcal{D}_\varepsilon$, let the infinite true LQ cost associated with the control law (6.13) be*

$$J_\varepsilon^\infty(x) = \sum_{t=0}^{\infty}\sum_{i=1}^{M}(\|x^i(t)\|_{Q^i}^2 + \|\kappa^i(x(t))\|_{R^i}^2) \tag{6.24}$$

where $x(0) = x$ *and* $x^i(t+1) = A^i x^i(t) + B^i \kappa^i(x(t))$ *for all* $i \in \mathbb{Z}^M$. *From (6.48),*

$$V_\varepsilon(x(t+1)) \leq V_\varepsilon(x(t)) - \sum_{i=1}^{M} \left(\|x^i(t)\|^i_{Q^i} + \|\kappa^i(x(t))\|^2_{R^i} \right), \tag{6.25}$$

which, when summed up from $t = 0$ *to* ∞, *implies that* $J^\infty_\varepsilon(x) \leq V_\varepsilon(x)$. *This gives a performance bound for the infinite LQ cost of the closed-loop system.*

6.6 Numerical Results

The example chosen is a four-agent system where every agent consists of two coupled water tanks [186]. The target is to regulate the water levels to some given references with a limited total input flow rate. As shown in Figure 6.1, q^i is the input flow and h^i_1 and h^i_2 are the water levels for system i. Suppose the targeted water levels are $\tilde{h}^i_1$ and $\tilde{h}^i_2$ with the steady input flow $\tilde{q}^i$ of system i. Let $x^i_1 = h^i_1 - \tilde{h}^i_1$, $x^i_2 = h^i_2 - \tilde{h}^i_2$ and $u^i = q^i - \tilde{q}^i$ for $i \in \mathbb{Z}^4$.

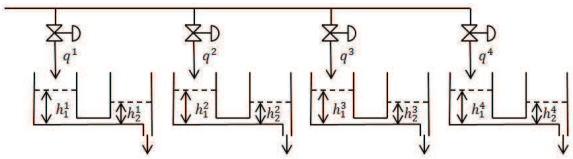


Figure 6.1: The water tanks system

Given the parameters: $\tilde{h}^i_1 = 1, \tilde{h}^i_2 = 0.64, \tilde{q}^i = 0.3$, $i \in \mathbb{Z}^4$, a linearized and discretized model is given by

$$x^i(t+1) = \begin{pmatrix} 0.8750 & 0.1250 \\ 0.1250 & 0.8047 \end{pmatrix} x^i(t) + \begin{pmatrix} 0.3 \\ 0 \end{pmatrix} u^i(t), i \in \mathbb{Z}^4$$

All the agents have the same local constraints $X^i := \{x^i \in \mathbb{R}^2 : |x^i_1| \leq 1, |x^i_2| \leq 0.64\}$ and $U^i := \{u^i \in \mathbb{R} : |u^i| \leq 0.3\}$. Suppose the maximal total input flow rate is 2, the coupled constraint can be given by $\sum_{i=1}^{4} q^i \leq 2$, which implies that $\sum_{i=1}^{4} u^i \leq 2 - \sum_{i=1}^{4} \tilde{q}^i = 0.8$. The values of K^i and P^i obtained from the

discrete-time ARE, with $Q^i = 10I_2$ and $R^i = 1$, $i \in \mathbb{Z}^4$, are

$$K^i = \begin{pmatrix} -1.7916 & -0.7337 \end{pmatrix}, P^i = \begin{pmatrix} 31.7459 & 9.8300 \\ 9.8300 & 56.3415 \end{pmatrix}$$

for all $i \in \mathbb{Z}^4$. Consider the network connection of a ring and $W = I - 0.1L(G)$. The minimal polynomial of W is $t^3 - 2.4t^2 + 1.88t - 0.48 = 0$. Hence, value of T of (3.39) is 3 and the finite-time consensus in (3.39) can be obtained 2 steps. The initial conditions are: $x^1(0) = [-0.3241 \ \ -0.5977]^T$, $x^2(0) = [0.4390 \ \ -0.4667]^T$, $x^3(0) = [-0.4391 \ \ -0.5818]^T$, $x^4(0) = [-0.5337 \ \ -0.4347]^T$ and the horizon length $N = 8$. The performance of the proposed DMPC approach is presented for several choices of ε and the comparison is made between the results of the proposed approach and that obtained by solving (5.15) with $\varepsilon = 0$ using a single centralized computer, known as the centralized MPC (CMPC) solution. The terminal sets $\{\sigma_s^i X_f^i\}_{i=1}^M$ in CMPC are obtained from $\min\{\sum_{i=1}^M (1-\sigma^i) : (5.13)\}$ with $\varepsilon = 0$: $\sigma_s^i = 0.6667, i \in \mathbb{Z}^4$. Consider the case of $\varepsilon = 0.01$ in DMPC. The overall input is shown in Figure 6.2.

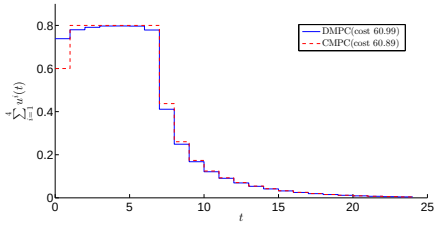


Figure 6.2: The overall input trajectories:DMPC($\varepsilon = 0.01$) and CMPC

The following table gives the real LQ cost $J_\varepsilon^\infty(x(0))$(as defined in Remark 6.1) for different choices of ε. The values of $\{\sigma_\varepsilon^i\}_{i\in\mathbb{Z}^4}$, obtained from $\min\{\sum_{i=1}^M (1-\sigma^i) : (5.13)\}$, are also shown in Table 6.1. It can be seen that the performance of the DMPC approach is close to that of CMPC because they have similar overall input trajectories and the degradation in cost is less than 0.2% even in the case of $\varepsilon = 0.01$. The fact that $J_\varepsilon^\infty(x(0))$ is upper bounded by $V_\varepsilon(x(0))$ can also be verified in Table 6.1.

ε	0.01	0.005	0.001
$\sigma_\varepsilon^i, i \in \mathbb{Z}^4$	0.4533	0.56	0.6453
$V_\varepsilon(x(0))$	61.87	61.26	60.95
$J_\varepsilon^\infty(x(0))$	60.99	60.91	60.89

Table 6.1: The values of $J_\varepsilon^\infty(x(0))$ for different choices of ε

A comparison between the proposed approach and the ADMM-based approach in [185] is made. The number of iterations of this proposed approach at each t along the trajectories is denoted by $\bar{k}_F(t) := \bar{k}(t) + 1$, whose values are shown in Table 6.2. The number of iterations of the ADMM-based approach, denoted by $\bar{k}_A(t)$ is also shown in this table. Notice that for $t \geq 7$, $\bar{k}(t)$ becomes 0 because the global constraints are not active. It can be seen from the Table 6.2 that the number of iterations needed to reach the same accuracy by the proposed approach is about 30% to 50% of that needed by the ADMM-based approach. The number of communications needed is $(2T-1)\bar{k}_F(t)$ for this approach and $T\bar{k}_A(t)$ for the ADMM-based approach. The number of optimization problems solved at each time instant t is $\bar{k}_F(t)$ by the proposed approach and $\bar{k}_A(t)$ by the ADMM-based approach. Although $(2T-1)\bar{k}_F(t)$ and $T\bar{k}_A(t)$ may be close and this approach may even require more communications in some cases, this approach solves fewer optimization problems.

	ε	$t=0$	$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$
$\bar{k}_F(t)$	0.01	23	21	19	16	13	9	1	1
	0.005	24	25	23	20	17	12	1	1
	0.001	49	50	48	46	42	30	6	1
$\bar{k}_A(t)$	0.01	88	88	72	67	71	74	94	1
	0.005	96	96	80	79	79	83	105	1
	0.001	120	120	112	106	108	115	131	1

Table 6.2: The number of iterations along the trajectories starting from $x(0)$ for different choices of ε

For the rest of this section, results of the proposed approach for problem (6.2) are compared with those obtained using other distributed algorithms in the literature: the distributed subgradient(D-SG) algorithm of [187] and the distributed Nesterov gradient(D-NG) algorithm of [106, 107]. Although [107] also proposed an algorithm that has a better convergence rate than D-SG and D-NG, it is unclear if it can be extended to constrained problems. For this reason, the comparison is only made with D-SG and D-NG. Consider the DMPC problem (5.15) with $\varepsilon = 0.01$ and the initial state $x(0)$. The proposed approach and D-NG use the weighted running average $\bar{\boldsymbol{u}}^{i,k}$ in (6.9) to

compute the primal cost after each iteration as shown in Algorithm 5.1, while D-SG uses the latest the primal variable $\tilde{\boldsymbol{u}}^{i,k}$ in (6.8). As the stepsize affects the performance of D-SG and D-NG, some reasonable stepsizes are chosens after a few trials: 0.2 and $\frac{7}{k+1}$ for D-SG; 0.1 and $\frac{5}{k+1}$ for D-NG. The convergence curves of these algorithms are shown in Figure 6.3. It can be seen that the propose approach converges faster than these two algorithms.

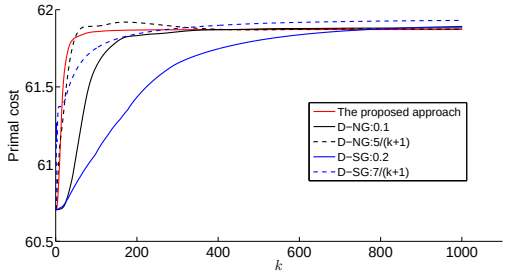


Figure 6.3: Convergence curves of different distributed algorithms

6.7 Conclusions

A novel DMPC approach is proposed for a group of linear systems with local and global constraints. The proposed approach relies on the dual problem of the overall MPC problem and uses a distributed fast dual gradient algorithm for its solution. This is made possible by introducing local copies of the dual variables in individual system and enforcing all the local copies to achieve consensus at each iteration. Provision for computational expediency is made via early termination of the proposed algorithm where the inaccuracy depends on the prescribed violation of the coupled constraint. Termination condition is checked using a finite-time consensus algorithm. Under mild assumptions, a suboptimal solution of the overall MPC problem can be obtained so long as the network of systems are connected. Recursive feasibility and exponential stability of the closed-loop system are ensured. The performance of the proposed approach is demonstrated by a 4-tank networked system with a limited total input flow rate. Compared to the ADMM-based approach of the same problem, this approach achieves convergence of about 2 to 3 times faster and invokes fewer quadratic optimization solvers, but may require more communications among systems. This communication issue is minimized by the use of a finite-time consensus based on the minimal polynomial extracted from the network. Comparisons of convergence results are also made with the distributed subgradient al-

gorithm and distributed Nesterov gradient algorithm. In both cases, the proposed method has faster convergence.

6.A Proof of Lemma 6.1

(i) The problem (6.2) can be rewritten as

$$\min_{\lambda^i \geq 0, i \in \mathbb{Z}^M} \sum_{i=1}^{M} g_i(\lambda^i) \quad \text{s.t. } \lambda^1 = \lambda^2 = \cdots = \lambda^M \tag{6.26}$$

which is equivalent to $\min_{\boldsymbol{\lambda} \in \Omega} g(\boldsymbol{\lambda})$, where $g(\boldsymbol{\lambda}) := \sum_{i=1}^{M} g_i(\lambda^i)$, $\boldsymbol{\lambda} = (\lambda^1, \lambda^2, \cdots, \lambda^M)$ and $\Omega = \{\boldsymbol{\lambda} \geq 0 : \lambda^1 = \lambda^2 = \cdots = \lambda^M\}$. Then, (6.7a)-(6.7b) can be written in a compact form

$$\tilde{\boldsymbol{\lambda}}^k = \boldsymbol{\lambda}^k + \theta_k(\theta_{k-1}^{-1} - 1)(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-1}) \tag{6.27a}$$

$$\boldsymbol{\lambda}^{k+1} = P_\Omega[\tilde{\boldsymbol{\lambda}}^k - \frac{1}{L_g}\nabla g(\boldsymbol{\lambda}^k)] \tag{6.27b}$$

It can be easily verified that $\nabla g(\boldsymbol{\lambda})$ is Lipschitz continuous with the constant L_g. For any $\lambda \geq 0$, from Proposition 6.9.2 in [98] and the fact that $\mathbf{1}_M \otimes \lambda \in \Omega$, it holds that

$$\sum_{i=1}^{M} g_i(\lambda^{i,k+1}) - \sum_{i=1}^{M} g_i(\lambda) + \theta_k^2 \sum_{\ell=0}^{k} \theta_\ell^{-1} \sum_{i=1}^{M} (g_i(\lambda) - \Delta^i(\lambda, \tilde{\lambda}^{i,\ell})) + \frac{L_g \theta_k^2}{2} \sum_{i=1}^{M} \|\bar{\lambda}^{i,k+1} - \lambda\|^2 \leq \frac{L_g M \theta_k^2}{2} \|\lambda\|^2 \tag{6.28}$$

where

$$\bar{\lambda}^{i,k} := \lambda^{i,k-1} + \theta_{k-1}^{-1}(\lambda^{i,k} - \lambda^{i,k-1}) \tag{6.29}$$

$$\Delta^i(\lambda, \tilde{\lambda}) := g_i(\tilde{\lambda}) + \nabla g_i^T(\tilde{\lambda})(\lambda - \tilde{\lambda}) \tag{6.30}$$

Let λ in (6.28) be any $\lambda^* \in \Lambda(x)$. The the first inequality of (6.14) holds because λ^* is a minimizer of (6.2) and the second inequality (6.14) holds due to the fact that $\Delta^i(\lambda^*, \tilde{\lambda}^{i,k}) \leq g_i(\lambda^*)$ from the convexity of $g_i(\cdot)$.

(ii) The inequality (6.15) follows from (6.28) because $\sum_{i=1}^{M} g_i(\lambda^{i,k+1}) - \sum_{i=1}^{M} g_i(\lambda^*) \geq 0$ and $\Delta^i(\lambda^*, \tilde{\lambda}^{i,k}) \leq g_i(\lambda^*)$.

(iii) Some intermediate results are needed to prove (6.16). Using the auxiliary variable $\bar{\lambda}^{i,k} = \lambda^{i,k-1} + \theta_{k-1}^{-1}(\lambda^{i,k} - \lambda^{i,k-1})$, $\lambda^{i,k+1}$ can be written as $\lambda^{i,k+1} = \theta_k(\bar{\lambda}^{i,k+1} - \lambda^{i,k}) + \lambda^{i,k}$. This, together

with $\tilde{\lambda}^{i,k} = \lambda^{i,k} + \theta_k(\theta_{k-1}^{-1} - 1)(\lambda^{i,k} - \lambda^{i,k-1})$, implies that

$$\begin{aligned}\lambda^{i,k+1} - \tilde{\lambda}^{i,k} &= \theta_k(\bar{\lambda}^{i,k+1} - \lambda^{i,k}) - \theta_k(\theta_{k-1}^{-1} - 1)(\lambda^{i,k} - \lambda^{i,k-1}) \\ &= \theta_k(\bar{\lambda}^{i,k+1} - \bar{\lambda}^{i,k}) \end{aligned} \tag{6.31}$$

Now, we can prove (6.16). Consider that $\lambda^{1,k+1} = \lambda^{2,k+1} = \cdots = \lambda^{M,k+1}$ and $\lambda^{i,k+1} = [\sum_{\ell=0}^{T-1} \tau_\ell y^i(\ell,k)]_+ \geq \sum_{\ell=0}^{T-1} \tau_\ell y^i(\ell,k) = \frac{1}{M}\sum_{i=1}^{M} y^i(0,k)$, it implies that

$$\sum_{i=1}^{M} \lambda^{i,k+1} \geq \sum_{i=1}^{M} \tilde{\lambda}^{i,k} + \frac{1}{L_g}(\sum_{i=1}^{M} f^i(x^i, \tilde{\boldsymbol{u}}^{i,k}) - b(\varepsilon)) \tag{6.32}$$

Using (6.31), (6.32) can be rewritten as

$$\sum_{i=1}^{M} \bar{\lambda}^{i,k+1} \geq \sum_{i=1}^{M} \bar{\lambda}^{i,k} + \theta_k^{-1}\frac{1}{L_g}(\sum_{i=1}^{M} f^i(x^i, \tilde{\boldsymbol{u}}^{i,k}) - b(\varepsilon)) \tag{6.33}$$

which implies that

$$\sum_{i=1}^{M} \bar{\lambda}^{i,k+1} \geq \sum_{\ell=1}^{k} \theta_\ell^{-1}\frac{1}{L_g}(\sum_{i=1}^{M} f^i(x^i, \tilde{\boldsymbol{u}}^{i,\ell}) - b(\varepsilon)) \tag{6.34}$$

Using (6.6) and (6.9), the inequality above can be again rewritten

$$\theta_k^2 \sum_{i=1}^{M} \bar{\lambda}^{i,k+1} \geq \frac{1}{L_g}(\sum_{i=1}^{M} f^i(x^i, \bar{\boldsymbol{u}}^{i,k}) - b(\varepsilon)) \tag{6.35}$$

From (6.15), we can know that $\|\bar{\lambda}^{i,k+1} - \lambda^*\| \leq \sqrt{M}\|\lambda^*\|$ for all $i \in \mathbb{Z}^M$, which from $\|\bar{\lambda}^{i,k+1} - \lambda^*\| \geq \|\bar{\lambda}^{i,k+1}\| - \|\lambda^*\|$ implies that $\|\bar{\lambda}^{i,k+1}\| \leq (\sqrt{M}+1)\|\lambda^*\|$. Consider that $\|\sum_{i=1}^{M} \bar{\lambda}^{i,k+1}\| \leq \sum_{i=1}^{M} \|\bar{\lambda}^{i,k+1}\|$. Therefore, from (6.35), we can get that

$$\frac{1}{L_g}\|[\sum_{i=1}^{M} f^i(x^i, \bar{\boldsymbol{u}}^{i,k}) - b(\varepsilon)]_+\| \leq \theta_k^2 \|\sum_{i=1}^{M} \bar{\lambda}^{i,k+1}\| \leq \theta_k^2 M(\sqrt{M}+1)\|\lambda^*\| \tag{6.36}$$

This, together with $\theta_k \leq \frac{2}{k+2}$ and $\|[\sum_{i=1}^{M} f^i(x^i, \bar{\boldsymbol{u}}^{i,k}) - b(\varepsilon)]_+\| \geq \|[\sum_{i=1}^{M} f^i(x^i, \bar{\boldsymbol{u}}^{i,k}) - b(\varepsilon)]_+\|_\infty$, implies (6.16). □

6.B Proof of Theorem 6.1:

Consider $g_i(\tilde{\lambda}^{i,k}) = -J^i(x^i,\tilde{\boldsymbol{u}}^{i,k}) - (\tilde{\lambda}^{i,k})^T(f^i(x^i,\tilde{\boldsymbol{u}}^{i,k}) - \frac{b(\varepsilon)}{M})$ and $\nabla g_i(\tilde{\lambda}^{i,k}) = -(f^i(x^i,\tilde{\boldsymbol{u}}^{i,k}) - \frac{b(\varepsilon)}{M})$ for all $i \in \mathbb{Z}^M$ and $k \geq 0$. For any $\lambda \geq 0$ in (6.28), it can be shown that $\Delta^i(\lambda,\tilde{\lambda}^{i,k}) = -J^i(x^i,\tilde{\boldsymbol{u}}^{i,k}) - \lambda^T(f^i(x^i,\tilde{\boldsymbol{u}}^{i,k}) - \frac{b(\varepsilon)}{M})$ where the notation $\Delta^i(\cdot,\cdot)$ is given in (6.30). Substitute this into (6.28), it yields

$$\sum_{i=1}^{M} g_i(\lambda^{i,k+1}) + \theta_k^2 \sum_{\ell=0}^{k} \theta_\ell^{-1} \sum_{i=1}^{M} (J^i(x^i,\tilde{\boldsymbol{u}}^{i,\ell}) + \lambda^T(f^i(x^i,\tilde{\boldsymbol{u}}^{i,\ell}) - \frac{b(\varepsilon)}{M}))) \leq \frac{L_g M \theta_k^2}{2}\|\lambda\|^2 \tag{6.37}$$

by dropping the quadratic term on the right-hand side. Consider (6.9) and the fact that

$$\theta_k^2 \sum_{\ell=0}^{k} \theta_\ell^{-1} \sum_{i=1}^{M} J^i(x^i,\tilde{\boldsymbol{u}}^{i,\ell}) \geq \sum_{i=1}^{M} J^i(x^i,\bar{\boldsymbol{u}}^{i,k}),$$

(6.37) implies

$$\sum_{i=1}^{M} g_i(\lambda^{i,k+1}) + \sum_{i=1}^{M} (J^i(x^i,\bar{\boldsymbol{u}}^{i,k}) + \lambda^T(f^i(x^i,\bar{\boldsymbol{u}}^{i,k}) - \frac{b(\varepsilon)}{M}))) \leq \frac{L_g M \theta_k^2}{2}\|\lambda\|^2 \tag{6.38}$$

Let $\lambda = 0$ in (6.38). It holds that

$$\sum_{i=1}^{M} g_i(\lambda^{i,k+1}) + \sum_{i=1}^{M} J^i(x^i,\bar{\boldsymbol{u}}^{i,k}) \leq 0 \tag{6.39}$$

From (6.1) and (6.2), it can be easily verify from the dual problem (6.1) that $\sum_{i=1}^{M} g_i(\lambda^{i,k+1}) \geq -\Phi(x,\lambda^*) = -\sum_{i=1}^{M} J^i(x^i,\boldsymbol{u}^{i*})$ since $\lambda^{1,k+1} = \lambda^{2,k+1} = \cdots = \lambda^{M,k+1}$. Using $\theta_k \leq \frac{2}{k+2}$, the second inequality of (6.17) holds. Now let us consider the proof of the first inequality of (6.17). Note that

$$\begin{aligned}
\sum_{i=1}^{M} J^i(x^i,\bar{\boldsymbol{u}}^{i,k}) &= \mathcal{L}(\{\bar{\boldsymbol{u}}^{i,k}\},\lambda^*) - (\lambda^*)^T(\sum_{i=1}^{M} f^i(x^i,\bar{\boldsymbol{u}}^{i,k}) - b(\varepsilon)) \\
&\geq \Phi(x,\lambda^*) - (\lambda^*)^T(\sum_{i=1}^{M} f^i(x^i,\bar{\boldsymbol{u}}^{i,k}) - b(\varepsilon)) \\
&= \sum_{i=1}^{M} J^i(x^i,\boldsymbol{u}^{i*}) - (\lambda^*)^T(\sum_{i=1}^{M} f^i(x^i,\bar{\boldsymbol{u}}^{i,k}) - b(\varepsilon))
\end{aligned} \tag{6.40}$$

Consider the following inequalities

$$\begin{aligned}(\lambda^*)^T(\sum_{i=1}^{M} f^i(x^i,\bar{\boldsymbol{u}}^{i,k})-b(\varepsilon)) &\le (\lambda^*)^T[\sum_{i=1}^{M} f^i(x^i,\bar{\boldsymbol{u}}^{i,k})-b(\varepsilon)]_+\\ &\le \|\lambda^*\|\|[\sum_{i=1}^{M} f^i(x^i,\bar{\boldsymbol{u}}^{i,k})-b(\varepsilon)]_+\|\\ &\le \theta_k^2 L_g M(\sqrt{M}+1)\|\lambda^*\|^2\end{aligned}$$

where the last inequality is from (6.36). This inequality, together with (6.40), implies the first inequality of (6.17). □

6.C Proof of Theorem 6.2

(i) Since $\{x^i,\boldsymbol{u}^i\}_{i=1}^M$ is a ε-relaxed solution, it satisfies

$$\boldsymbol{u}^i\in\mathcal{U}^i(x^i), i\in\mathbb{Z}^M,\ \ \sum_{i=1}^{M} f^i(x^i,\boldsymbol{u}^i)-b(\varepsilon)\le \varepsilon M\mathbf{1}_{pN} \tag{6.41}$$

Rewriting $f^i(x^i,\boldsymbol{u}^i)$ and $b(\varepsilon)$ back in terms of $\{x_0^i,x_1^i,\cdots,x_N^i\}$ and $\{u_0^i,u_1^i,\cdots,u_{N-1}^i\}$ (note that $f^i(x^i,\boldsymbol{u}^i)-b$ are simplified expression of (5.9)), (6.41) is equivalent to

$$\sum_{i=1}^{M}\Psi_x^i x_{\ell|t}^i+\Psi_u^i u_{\ell|t}^i\le(1-\varepsilon M(\ell+1))\mathbf{1}_p+\varepsilon M\mathbf{1}_p=(1-\varepsilon M(\ell+1))\mathbf{1}_p,\ \ \forall\ell\in\mathbb{Z}_0^{N-1} \tag{6.42}$$

For all $i\in\mathbb{Z}^M$, let a feasible control to the i^{th} system at next time instant be chosen as

$$\boldsymbol{u}^{i+}:=\{u_0^{i+},u_1^{i+},\cdots,u_{N-1}^{i+}\}:=\{u_1^i,u_2^i,\cdots,u_{N-1}^i,K^i x_N^i\}. \tag{6.43}$$

and the associated state sequence $\{x_0^{i+},x_1^{i+},\cdots,x_N^{i+}\} := \{x_1^i,x_2^i,\cdots,x_N^i,(A^i+B^iK^i)x_N^i\}$. It follows from this choice of $\boldsymbol{u}^{i+}$ and (6.42) that

$$\sum_{i=1}^{M}\Psi_x^i x_\ell^{i+}+\Psi_u^i u_\ell^{i+}=\sum_{i=1}^{M}\Psi_x^i x_{\ell+1}^i+\Psi_u^i u_{\ell+1}^i\le(1-\varepsilon M(1+\ell))\mathbf{1}_p,\ \forall \ell\in\mathbb{Z}_0^{N-2} \tag{6.44}$$

and where $\ell=N-1$,

$$\sum_{i=1}^{M}(\Psi_x^i x_N^i+\Psi_u^i K^i x_N^i)=\sum_{i=1}^{M}\bar{\Psi}x_N^i\le(1-\varepsilon MN)\mathbf{1}_p \tag{6.45}$$

where the last inequality follows the fact that $x_N^i\in\sigma_\varepsilon^i X_f^i$ and (5.10) (with $T_f^i=\sigma_\varepsilon^i X_f^i$). In addition, $u_\ell^{i+}\in U^i$ for $\ell\in\mathbb{Z}_0^{N-2}$ since $u_{\ell+1}^i\in U^i$ because of (6.41). The last control, $u_{N-1}^{i+}=K^i x_N^i\in U^i$ because $x_N^i\in\sigma_\varepsilon^i X_f^i$ and $\sigma_\varepsilon^i X_f^i$ satisfies (5.6). The constraints of $x_\ell^{i+}\in X^i,\ell\in\mathbb{Z}_0^{N-1}$ and $x_N^{i+}\in\sigma_\varepsilon^i X_f^i$ follow similar argument. These properties implies $\boldsymbol{u}^{i+}\in\mathcal{U}(x^{i+})$, $i\in\mathbb{Z}^M$ and $\sum_{i=1}^{M} f^i(x^{i+},\boldsymbol{u}^{i+})\le b(\varepsilon)$.
(ii) This result follows from Lemma 6.2 since $x^+\in\mathcal{D}_\varepsilon$. □

6.D Proof of Lemma 6.3

By definition of $h_{\bar{\sigma}^i X_f^i}$, $\bar{F}_\ell^i x^i\le max_{y\in\bar{\sigma}^i X_f^i}\bar{F}_\ell^i y=h_{\bar{\sigma}^i X_f^i}(\bar{F}_\ell^i)=\bar{\sigma}^i h_{X_f^i}(\bar{F}_\ell^i)$ for any $x^i\in\bar{\sigma}^i X_f^i$ and any $\ell\in\mathbb{Z}^{Np}$. This fact, together with $\bar{\sigma}^i\le\frac{1}{M}b_\ell(\varepsilon)/h_{X_f^i}(\bar{F}_\ell^i)$ from the definition of $\{\bar{\sigma}^i\}_{i\in\mathbb{Z}^M}$, implies $\frac{b(\varepsilon)}{M}-\bar{F}^i x^i\ge 0$. Since $\bar{\sigma}^i X_f^i\subseteq\sigma_\varepsilon^i X_f^i$, $K_A^i x^i\in\mathcal{U}^i(x^i)$ from (5.5), (5.6) and (5.14). Hence, when $k=0$, $K_A^i x^i$ is a feasible solution to $\min_{\boldsymbol{u}^i\in\mathcal{U}^i(x^i)} J^i(x^i,\boldsymbol{u}^i)$, $i\in\mathbb{Z}^M$. Since K^i and P^i are obtained from ARE, $K_A^i x^i$ is the optimal solution to $\min_{\boldsymbol{u}^i} J^i(x^i,\boldsymbol{u}^i)$. Therefore, the solution when $k=0$ is $\tilde{\boldsymbol{u}}^{i,0}=K_A^i x^i$ for all $i\in\mathbb{Z}^M$. This suggests that $\bar{\boldsymbol{u}}^{i,0}=K_A^i x^i$ and $f^i(x^i,K_A^i x^i)-\frac{b(\varepsilon)}{M}\le 0$ for all $i\in\mathbb{Z}^M$, which means that $\{x^i,\bar{\boldsymbol{u}}^{i,0}\}_{i=1}^M$ is a ε-relaxed solution and Algorithm 5.1 terminates at $k=0$. □

6.E Proof of Theorem 6.3

(i) Consider the ε-relaxed solution $\{x^i(t),\bar{\boldsymbol{u}}^{i,\bar{k}(t)}\}_{i=1}^M$ at time t. Let $\boldsymbol{u}_t^i:=\{u_{0|t}^i,u_{1|t}^i,\cdots,u_{N-1|t}^i\}=\bar{\boldsymbol{u}}^{i,\bar{k}(t)}$ with the associated predictive state sequence $\boldsymbol{x}_t^i:=\{x_{0|t}^i,x_{1|t}^i,\cdots,x_{N|t}^i\}$ for all $i\in\mathbb{Z}^M$. Define the shifted predicted sequence at next time instant $\hat{\boldsymbol{u}}_{t+1}^i=\{u_{1|t}^i,\cdots,u_{N-1|t}^i,K^i x_{N|t}^i\}$, $i\in\mathbb{Z}^M$. From property (i) of Theorem 6.2, it suggests that $\{\hat{\boldsymbol{u}}_{t+1}^i\}_{i=1}^M$ is a feasible solution to $\mathbb{P}_\varepsilon(x(t+1))$ because

$x^i(t+1) = A^i x^i(t) + B^i u^i_{0|t}$ using the control law (6.13).

(ii) This result holds since $\{x^i(t), \bar{\boldsymbol{u}}^{i,\bar{k}(t)}\}_{i=1}^M$ is a $(\varepsilon, 0)$-upper-relaxed solution of $\mathbb{P}_\varepsilon(x(t))$ for all $t \geq 1$.

(iii) Let $V_\varepsilon(x(t)) = \sum_{i=1}^M J^i(x^i(t), \boldsymbol{u}_t^{i*})$ (where $\{\boldsymbol{u}_t^{i*}\}_{i=1}^M$ is the optimal solution of (5.15)) be the Lyapunov function of the closed-loop system of (5.1) with input $u^i(t) = \bar{u}_0^{i,\bar{k}(t)}$ given by (6.13). When Algorithm 5.1 terminates at time t, $\boldsymbol{u}^{i,\bar{k}(t)} := \boldsymbol{u}_t^i$ and it follows property (iii) of Theorem 6.2

$$\sum_{i=1}^M J^i(x^i(t), \boldsymbol{u}_t^i) - V_\varepsilon(x(t)) \leq 0 \tag{6.46}$$

Using the same shifted shifted control sequence $\hat{\boldsymbol{u}}_{t+1}^i$ at time $t+1$ as defined in (i), it follows from the standard argument in MPC,

$$\begin{aligned} J^i(x^i(t+1), \hat{\boldsymbol{u}}_{t+1}^i) - J^i(x^i(t), \boldsymbol{u}_t^i) &= -\|x^i(t)\|^i_{Q^i} - \|u^i(t)\|^2_{R^i} \\ &\quad + \|x^i_{N|t}\|^2_{Q^i} + \|K^i x^i_{N|t}\|^2_{R^i} + \|A^i_K x^i_{N|t}\|^2_{P^i} - \|x^i_{N|t}\|^2_{P^i} \\ &= -\|x^i(t)\|^i_{Q^i} - \|u^i(t)\|^2_{R^i} \end{aligned} \tag{6.47}$$

where the last equality is from the fact K^i, P^i satisfy the Algebraic Riccatii Equation of $(A^i_K)^T P^i A^i_K - P^i = -(Q^i + K^i R^i K^i)$. Since $\hat{\boldsymbol{u}}_{t+1}^i$ may not be the optimal at $t+1$,

$$\begin{aligned} V_\varepsilon(x(t+1)) &\leq \sum_{i=1}^M J^i(x^i(t+1), \hat{\boldsymbol{u}}_{t+1}^i) = \sum_{i=1}^M \left(J^i(x^i(t), \boldsymbol{u}_t^i) - \|x^i(t)\|^i_{Q^i} - \|u^i(t)\|^2_{R^i} \right) \\ &= V_\varepsilon(x(t)) - \sum_{i=1}^M (\|x^i(t)\|^2_{Q^i} + \|u^i(t)\|^2_{R^i}) \qquad (6.48) \\ &\leq V_\varepsilon(x(t)) - \sum_{i=1}^M \|x^i(t)\|^2_{Q^i} \qquad (6.49) \end{aligned}$$

where the equality condition follows from (6.47) and the last inequality is due to (6.46). Therefore, $x^i(t)$ goes to 0 as $t \to \infty$ for all $i \in \mathbb{Z}^M$. This means that there exists a finite t_f such that $x^i(t_f) \in \bar{\sigma}^i X_f^i$ for all $i \in Z^M$. When this happens, it follows from Lemma 6.3 that $\kappa^i(x(t_f)) = K^i x^i(t_f)$ and the closed-loop system becomes $x^i(t_f+1) = A^i_K x^i(t_f)$. Since $x^i(t_f+1) \in \bar{\sigma}^i X_f^i$ for any $x^i(t_f) \in \bar{\sigma}^i X_f^i$ from (5.6). As a result, $x^i(t+1) = A^i_K x^i(t)$ for all $t \geq t_f$ and the closed-loop system is exponentially stable. □

CHAPTER 7

Conclusions and Future Work

This chapter summarizes the main contributions of this thesis and provides possible directions for future work.

7.1 Contributions of This Thesis

A summary of the main contributions is presented below. The first contribution of this thesis is a DMPC approach for a network of dynamically-coupled linear systems. This approach is less conservative compared with the existing DMPC approaches because of the choices of the terminal cost and the terminal set.

- Unlike other DMPC approaches, where the terminal cost function depends on a block diagonal Lyapunov matrix, this approach uses a terminal cost function that depends on a non-block diagonal Lyapunov matrix that conforms to the structural constraint imposed by the network.
- The terminal set is obtained from the maximal constraint admissible invariant set of the overall system. More exactly, the approach determines a time-varying terminal set that moves within the maximal constraint admissible invariant set, changing in both size and location at each time.
- The computation of the time-varying terminal set depends on the topology of the communication network. When the network is fully connected or a central collector exists, the terminal set can be easily computed and local exponential stability is achieved. If the network is only connected without a central collector, the computations of the terminal set require a series of linear programming (LP) problems; the computations of which can be speeded up, via a

preprocessing step, so that the approach is suitable for online computations. In this case, the overall closed-loop system can only be guaranteed to be asymptotically stable.

The second contribution of this thesis is a DMPC approach for a group of linear systems with local and global constraints.

- In most previous DMPC approaches for the same setting, the optimality properties of the overall system are not explicitly pursued. This approach is able to achieve the optimality of the overall MPC problem within some prescribed accuracy. This is possibly because the approach is based on the dual problem of the overall MPC problem, which is solved by a distributed ADMM algorithm. The distributed implementation is made possible by introducing local copies of the dual variables in individual system and enforcing all the local copies to achieve consensus value. One important property of this distributed ADMM formulation is that it converges fast to modest accuracy.

- Besides the distributed formulation, this approach also proposes a stopping criterion that allows early termination of the distributed ADMM algorithm where the inaccuracy depends on the violation of the coupled constraint and the primal cost gap. The characterization of the violation of the coupled constraint and the primal cost gap is given from the sequences generated from the distributed ADMM algorithm.

- The stopping criterion based on the violation of the coupled constraint and the primal cost gap is provided and is checked using a finite-time consensus algorithm.

- Under mild assumptions, the DMPC converges to some small neighborhood of the optimal so long as the network of systems is connected. Recursive feasibility and exponential stability of the closed-loop system are ensured.

The last contribution of this thesis is an accelerated distributed dual gradient algorithm for DMPC of a group of linear systems with coupled constraints. The main advantage of this approach is that it converges faster than the ADMM-based approach.

- Similar to the ADMM-based approach, this approach also relies on the dual problem of the overall MPC problem. The dual problem is then solved by a distributed fast dual gradient algorithm. Unlike the ADMM-based approach, the local copies of the dual variables in this

approach reach the consensus at each iteration due to the use of a finite-time consensus algorithm.

- Another improvement of this approach is that the stopping criterion is based on the violation of the coupled constraint. There is no requirement to check the primal cost gap. This is possibly because the overall cost function is upper bounded by the optimal overall cost function at each iteration. Again, this approach uses a finite-time consensus algorithm to check the stopping criterion.
- Under mild assumptions, a suboptimal solution of the overall MPC problem can be obtained so long as the network of systems is connected. It is shown in the experiment that this approach converges 2-3 times faster than the ADMM-based approach.

7.2 Future Work

Several possible directions for the future research of this thesis are outlined below.

7.2.1 Stabilization with Structural Constraints

The stabilization of linear systems with arbitrary structural constraints is a fundamental problem. The stabilizability of (A,B) is not enough to ensure the solution of a stabilizing K that conforms to the structural constraint imposed by the communication network. For arbitrary structural constraints, it is generally difficult to establish the necessary and sufficient condition for existence of the structural stabilizing K. In the thesis, the numerical solution of the structural K is obtained by solving the bilinear optimization problem (4.21) using ADMM, as shown in Chapter 4. However, the solution is not guaranteed since (4.21) is nonconvex. The work of this thesis can be extended to discuss the theoretical and numerical solution of the structural stabilizing K. In addition, it is also more desirable to solve (4.21) in a distributed manner for large-scale systems.

7.2.2 Disturbance and Transmission Errors

Throughout the thesis, all systems are assumed to be free of disturbance and the information transmission among systems is perfect. One direction of the future research is to extend the formulations of this thesis to the disturbed case where the systems are of the form

$$x(t+1) = Ax(t) + Bu(t) + w(t), \quad t \in \mathbb{Z}_+ \tag{7.1}$$

where $w(t)$ is the disturbance. The robust MPC can be used if the disturbance is assumed to be bounded by

$$w(t) \in W \tag{7.2}$$

where W is a bounded convex polyhedral set. In practice, there is always uncertainty in the transmission, such as time-delay, transmission breakdown, and information errors. The DMPC approaches in this thesis should be adapted to account for these problems. The difficulty is to achieve recursive feasibility and stability under these situations.

7.2.3 Time-Varying Communication

This thesis only considers static and fixed networks. More complications arise when the topology of the network is time-varying. Distributed control under time-varying network is a popular topic and has applications in areas such as unmanned air vehicles(UAVs), formation control, and congestion control in communication networks. The work of this thesis can be extended in the direction of time-varying networks. As far as the author can see, conditions on the connectivity of the time-varying network is needed to ensure recursive feasibility and stability of the overall system. This may need results in consensus and optimization in multi-agent time-varying network [85, 86, 96].

References

[1] D. D. Šiljak. *Decentralized control of complex systems*. Cambridge University Press, 1991.

[2] J. Lunze. *Feedback control of large scale systems*. Prentice-Hall, 1992.

[3] L. Bakule. Decentralized control: An overview. *Annual Reviews in Control*, 32(1):87–98, 2008.

[4] A.I. Zečeić and D. D. Šiljak. *Control of Complex Systems:Structural Constraints and Uncertainty*. Springer, 2010.

[5] M. Hovd and S. Skogestad. Sequential design of decentralized controllers. *Automatica*, 30(10):1601–1607, 1994.

[6] E. J. Davison and Ian J. Ferguson. The design of controllers for the multivariable robust servomechanism problem using parameter optimization methods. *IEEE Transactions on Automatic Control*, 26(1):93–110, 1981.

[7] R. Scattolini and N. Schiavoni. A parameter optimization approach to the design of structurally constrained regulators for discrete-time systems. *International Journal of Control*, 42(1):177–192, 1985.

[8] M. Ikeda, D. D. Šiljak, and D. E. White. Decentralized control with overlapping information sets. *Journal of Optimization Theory and Applications*, 34(2):279–310, 1981.

[9] A. Iftar. Decentralized estimation and control with overlapping input, state, and output decomposition. *Automatica*, 29(2):511–516, 1993.

[10] M. E. Sezer and D. D. Šiljak. Nested ε-decompositions and clustering of complex systems. *Automatica*, 22(3):321–331, 1986.

[11] M. Hodžić and D. D. Šiljak. Decentrahzed estimation and control with overlapping information sets. *IEEE Transactions on Automatic Control*, 31(1):83–86, 1986.

[12] N. R. Sandell, P. Varaiya, M. Athans, and M. G. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, 23(2):108–128, 1978.

[13] D. D. Šiljak. Decentralized control and computations: status and prospects. *Annual Reviews in Control*, 20:131ÍC141, 1996.

[14] D. D. Šiljak and A. I. Zečeić. Control of large-scale systems: Beyond decentralized feedback. *Annual Reviews in Control*, 29:169–179, 2005.

[15] R. Scattolini. Architectures for distributed and hierarchical model predictive control - A review. *Journal of Process Control*, 19(5):723–731, 2009.

[16] A.I. Zečeić and D. D. Šiljak. A new approach to control design with overlapping information structure constraints. *Automatica*, 41(2):265–272, 2005.

[17] A.I. Zečeić and D. D. Šiljak. Control design with arbitrary information structure constraints. *Automatica*, 44(10):2642–2647, 2008.

[18] R. Cogill and S. Lall. Control design for topology-independent stability of interconnected systems. In *Proceedings of the American Control Conference*, pages 3717–3722, Boston, MA, USA, June 2004.

[19] Y. Ebihara and T. Hagiwara. Structured controller synthesis using lmi and alternating projection method. In *Proc. of Conference on Decision and Control*, pages 5632–5637, Maui, Hawaii USA, December 2003.

[20] J. Lavaei and A. G. Aghdam. Control of continuous-time lti systems by means of structurally constrained controllers. *Automatica*, 44(1):141ÍC148, 2008.

[21] J. Lavaei and A. G. Aghdam. Overlapping control design for multi-channel systems. *Automatica*, 45(5):1326–1331, 2009.

[22] C. H. Papadimitriou and J. Tsitsiklis. Intractable problems in control theory. *SIAM journal on control and optimization*, 24(4):639–654, 1986.

[23] V. Blondel and J.N. Tsitsiklis. Np-hardness of some linear control design problems. *SIAM Journal on Control and Optimization*, 35(6):2118–2127, 1997.

[24] V. D. Blondel and J. N. Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

[25] M. Rotkowitz and S. Lall. Decentralized control information structures preserved under feedback. In *41th IEEE Conference on Decision and Control*, pages 569–575, 2002.

[26] M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51(2):274–286, 2006.

[27] L. Lessard and S. Lall. Internal quadratic invariance and decentralized control. In *Proceedings of the 2010 American Control Conference*, pages 5596–5601, 2010.

[28] D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, 1990.

[29] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[30] D. Q. Mayne. Control of constrained dynamic systems. *European Journal of Control*, 7:87–99, 2001.

[31] E. F. Camacho and A. C. Bordons. *Model Predictive Control*. Springer Verlag, 2004.

[32] E.G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36:1008–1020, 1991.

[33] Donald Chmielewski and V Manousiouthakis. On constrained infinite-time linear quadratic optimal control. In *Proceedings of the 35th IEEE Conference on Decision and Control*, volume 2, pages 1319–1324. IEEE, 1996.

[34] P. O.M. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 43(8):1163–1169, 1998.

[35] T. Keviczky, F. Borrelli, and G. J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42(12):2105–2115, 2006.

[36] J. B. Rawlings and B. T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control*, 18(9):839–845, 2008.

[37] A. Alessio, D. Barcelli, and A. Bemporad. Decentralized model predictive control of dynamically coupled linear systems. *Journal of Process Control*, 21(5):705–714, 2011.

[38] P. D. Christofides, R. Scattolini, D. M Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers and Chemical Engineering*, 51:21–41, 2013.

[39] I. Necoara and D. Clipici. Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: Application to distributed mpc. *Journal of Process Control*, 23(3):243–253, 2013.

[40] J. M. Maestre and R. R. Negenborn. *Distributed model predictive control made easy*. Springer, 2014.

[41] R.R. Negenborn and J.M. Maestre. Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems*, 34(4):87–97, 2014.

[42] X. Du, Y. Xi, and S. Li. Distributed model predictive control for large-scale systems. In *Proceedings of the American Control Conference*, pages 3142–3143, Arlington, VA, USA, June 2001.

[43] W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formatiom stabilization. *Automatica*, 42(4):549ÍC558, 2007.

[44] A. Richardsa and J. P. How. Robust distributed model predictive control. *International Journal of Control*, 80(9):1517–1531, 2007.

[45] A. Alessio and A. Bemporad. Stability conditions for decentralized model predictive control under packet drop communication. In *Proceedings of the American Control Conference*, pages 3577–3582, Seattle, Washington, USA, 2008.

[46] C. Wang and C. J. Ong. Distributed model predictive control of dynamically decoupled systems with coupled cost. *Automatica*, 46(12):2053–2058, 2010.

[47] M. Farina and R. Scattolini. Distributed non-cooperativeMPC with neighbor-to-neighbor communication. In *Proceedings of the 18th IFAC World Congress*, pages 404–409, Milano, Italy, September 2011.

[48] S. Riverso and G. Ferrari-Trecate. Tube-based distributed control of linear constrained systems. *Automatica*, 48(11):2860–2865, 2012.

[49] S. Riverso, M. Farina, and G. Ferrari-Trecate. Plug-and-play decentralized model predictive control for linear systems. *IEEE Transactions on Automatic Control*, 58(10):2608–2614, 2013.

[50] W. B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, 52(7):1249–1262, 2007.

[51] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Stability and optimality of distributed model predictive control. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pages 6680–6685, 2005.

[52] B. T. Stewart, S. J. Wright, and J. B. Rawlings. Cooperative distributed model predictive control for nonlinear systems. *Journal of Process Control*, 21(5):698–704, 2011.

[53] M. A. Müller, M. Reble, and F. Allg ower. Cooperative control of dynamically decoupled systems via distributed model predictive control. *International Journal of Robust and Nonlinear Control*, 22(12):1376ÍC1397, 2012.

[54] A. Ferramosca, D. Limon, I. Alvarado, and E. F. Camacho. Cooperative distributed MPC for tracking. *Automatica*, 49(4):906ÍC914, 2013.

[55] T. S. Basar and J. G. Olsder. *Dynamic Noncooperative Game Theory*. SIAM, Philadelphia, 1999.

[56] P. Trodden, D. Nicholson, and A. Richards. Distributed model predictive control as a game with coupled constraints. In *Proceedings of the European Control Conference*, pages 2996–3001, 2009.

[57] J. M. Maestre, D. M. Peña, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32(2):153ÍC176, 2010.

[58] J. Lee and D. Angeli. Cooperative distributed model predictive control for linear plants subject to convex economic objectives. In *Proceedings of 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 3434–3439, 2011.

[59] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems*, 22(1):44–52, 2002.

[60] Paul Trodden and Arthur Richards. Cooperative distributed MPC of linear systems with coupled constraints. *Automatica*, 49(2):479–487, 2013.

[61] A. N. Venkat, J. B. Rawlings, and S. J. Wright. Implementable distributed model predictive control with guaranteed performance properties. In *Proceedings of the American Control Conference*, Minneapolis, MN, USA, 2006.

[62] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed mpc strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6):1192–1206, 2008.

[63] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460–469, 2010.

[64] P. Giselsson and A. Rantzer. On feasibility, stability and performance in distributed model predictive control. *IEEE Transactions on Automatic Control*, 59(4):1031–1036, 2014.

[65] B. Johansson, A. Speranzon, M. Johansson, and K. H. Johansson. On decentralized negotiation of optimal consensus. *Automatica*, 44(4):1175–1179, 2008.

[66] C. Conte, N. Voellmy, M. N. Zeilinger, M. Morari, and C. N. Jones. Distributed synthesis and control of constrained linear systems. In *Proceedings of the American Control Conference*, pages 6017–6022, Montreal, Canada, June 2012.

[67] C. Conte, N. Voellmy, M. N. Zeilinger, M. Morari, and C. N. Jones. Cooperative distributed tracking mpc for constrained linear systems: Theory and synthesis. In *Proceedings of the Decision and Control (CDC)*, pages 3812–3817, Firenze, Italy, December 2013.

[68] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

[69] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics (SIAM), 1994.

[70] I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4:317–367, 1998.

[71] R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(2):41–76, 1975.

[72] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[73] D. Gabay. Chapter ix applications of the method of multipliers to variational inequalities. *Studies in mathematics and its applications*, 15:299–331, 1983.

[74] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[75] S. Setzer. Operator splittings, bregman methods and frame shrinkage in image processing. *International Journal of Computer Vision*, 92(3):265–280, 2011.

[76] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang. An admm algorithm for a class of total variation regularized estimation problems. In *Proceedings of the16th IFAC Symposium on System Identification*, page 83Ű88, 2012.

[77] S. Joshi, M. Codreanu, and M. Latva-aho. Distributed sinr balancing for miso downlink systems via the alternating direction method of multipliers. In *International Symposium on Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt)*, pages 318–325. IEEE, 2013.

[78] J. Eckstein and D. P. Bertsekas. On the douglasŮrachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–

318, 1992.

[79] B. He and X. Yuan. On the o(1/n) convergence rate of the douglas-rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

[80] Deren Han and Xiaoming Yuan. A note on the alternating direction method of multipliers. *Journal of Optimization Theory and Applications*, 155(1):227–238, 2012.

[81] W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, pages 1–28, 2012.

[82] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson. Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems. *IEEE Transactions on Automatic Control*, 60(3):644–658, 2015.

[83] Robert Nishihara, Laurent Lessard, Benjamin Recht, Andrew Packard, and Michael I Jordan. A general analysis of the convergence of admm. In *Proceedings of the International Conference on Machine Learning*, page 343Ű352, 2015.

[84] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[85] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[86] A. Nedić, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

[87] H. Terelius, U. Topcu, and R. Murray. Decentralized multi-agent optimization via dual decomposition. In *18th World Congress of the International Federation of Automatic Control(IFAC)*, pages 11245–11251, 2011.

[88] Tyler H Summers and John Lygeros. Distributed model predictive consensus via the alternating direction method of multipliers. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 79–84, 2012.

[89] M. Hong and Z. Luo. On the linear convergence of the alternating direction method of multipliers. *arXiv preprint arXiv:1208.3922*, 2012.

[90] B. He, M. Tao, and X. Yuan. Alternating direction method with Gaussian back substitution for separable convex programming. *SIAM Journal on Optimization*, 22(2):313–340, 2012.

[91] W. Deng, M. Lai, Z. Peng, and W. Yin. Parallel multi-block admm with o (1/k) convergence. *arXiv preprint arXiv:1312.3040*, 2013.

[92] M. Tao and X. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.

[93] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2233–2246, 2012.

[94] C. Chen, B. He, Y. Ye, and X. Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016.

[95] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Kluwer, Boston, 2004.

[96] S. S. Ram, A. Nedić, and V. V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of optimization theory and applications*, 147(3):516–545, 2010.

[97] Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[98] D. P. Bertsekas. *Convex optimization theory*. Athena Scientific, 2009.

[99] Y. Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Mateaticheskie Metody*, 24(3):509–517, 1988.

[100] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[101] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report, Department of Mathematics, University of Washington, 2008.

[102] D. Goldfarb, S. Ma, and K. Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex functions. *Mathematical Programming*, 141(1-2):349–382, 2013.

[103] S. Azadi and S. Sra. Towards an optimal stochastic alternating direction method of multipliers. In *Proceedings of the 31st International Conference on Machine Learning*, pages 620–628, 2014.

[104] Y. Ouyang, Y. Chen, G. Lan, and E. Pasiliao Jr. An accelerated linearized alternating direction method of multipliers. *SIAM Journal on Imaging Sciences*, 8(1):644–681, 2015.

[105] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.

[106] D. Jakovetic, J. M.F. Moura, and J. Xavier. Distributed nesterov-like gradient algorithms. In *Proceedings of Conference on Decision and Control*, pages 5459–5464. IEEE, 2012.

[107] D. Jakovetic, J. Xavier, and J. M.F. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

[108] D. Jakovetic, F. Xavier, J. Manuel, and J. M.F. Moura. Convergence rates of distributed nesterov-like gradient methods on random networks. *IEEE Transactions on Signal Processing*, 62(4):868–882, 2014.

[109] Albert I Chen and Asuman Ozdaglar. A fast distributed proximal-gradient method. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 601–608. IEEE, 2012.

[110] A. M. Borbely and J. F. Kreider. *Distributed generation: the power paradigm for the new millennium*. CRC press, 2001.

[111] V. Sakizlis, J. D. Perkins, and E. N. Pistikopoulos. Recent advances in optimization-based simultaneous process and control design. *Computers & Chemical Engineering*, 28(10):2069–2086, 2004.

[112] H. Sarimveis, P. Patrinos, C. D. Tarantilis, and C. T. Kiranoudis. Dynamic modeling and control of supply chain systems: A review. *Computers & Operations Research*, 35(11):3530–3561, 2008.

[113] L Acar. Boundaries of the receding horizon control for interconnected systems. *Journal of Optimization Theory and Applications*, 84(2):251–271, 1995.

[114] Nader Motee and Bijan Sayyar-Rodsari. Optimal partitioning in distributed model predictive control. In *Proceedings of the American Control Conference*, volume 6, pages 5300–5305. IEEE, 2003.

[115] D. Jia and B. Krogh. Min-max feedback model predictive control for distributed control with communication. In *Proceedings of the 2002 American Control Conference*, volume 6, pages 4507–4512. IEEE, 2002.

[116] D. Q. Mayne, M. M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[117] David H Shim, H Jin Kim, and Shankar Sastry. Decentralized nonlinear model predictive control of multiple flying robots. In *Proceedings of the 42nd Annual Conference on Decision and Control*, volume 4, pages 3621–3626. IEEE, 2003.

[118] Steven L Waslander, Gokhan Inalhan, and Claire J Tomlin. Decentralized optimization via nash bargaining. In *Conference on Cooperative Control and Optimization*, page 565Ű583, 2003.

[119] C. Conte, C. N. Jones, M. Morari, and M. N. Zeilinger. Distributed synthesis and stability of cooperative distributed model predictive control for linear systems. *Automatica*, 69:117–125, 2016.

[120] A. Richards and J. How. Robust stable model predictive control with constraint tightening. In *Proceedings of the American Control Conference*, page 1557Ű1562. IEEE, 2006.

[121] Paul Trodden and Arthur Richards. Distributed model predictive control of linear systems with persistent disturbances. *International Journal of Control*, 83(8):1653–1663, 2010.

[122] Paul Trodden. Feasible parallel-update distributed MPC for uncertain linear systems sharing convex constraints. *Systems & Control Letters*, 74:98–107, 2014.

[123] A. Richards, J. Bellingham, M. Tillerson, and J. How. Coordination and control of multiple uavs. In *AIAA Conference on Guidance, Navigation, and Control*, 2002.

[124] P.E. Hagen, O. Midtgaard, , and O. Hasvold. Making auvs truly autonomous. In *OCEANS 2007*, pages 1–4. IEEE, 2007.

[125] O. Hegrenaes, E. Berglund, and O. Hallingstad. Model-aided inertial navigation for underwater vehicles. In *IEEE International Conference on Robotics and Automation*, pages 1069–1076. IEEE, 2008.

[126] D. M. Stipanović, G. Inalhan, R. Teo, and C. J. Tomlin. Decentralized overlapping control of a formation of unmanned aerial vehicles. *Automatica*, 40(8):1285–1296, 2004.

[127] R. M. Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, 2007.

[128] R. Muñoz-Salinas, E. Aguirre, M. García-Silvente, and M. Gómez. A multi-agent system architecture for mobile robot navigation based on fuzzy and visual behaviour. *Robotica*, 23(06):689–699, 2005.

[129] Fernando Paganini, John Doyle, and Steven Low. Scalable laws for stable network congestion control. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 1, pages 185–190. IEEE, 2001.

[130] R. Olfati-Saber, A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[131] W. Ren and R. W. Beard. *Distributed consensus in multi-vehicle cooperative control*. Springer, 2008.

[132] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

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

[134] F. R. K. Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[135] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[136] A. Agarwal, M. J. Wainwright, and J. C. Duchi. Distributed dual averaging in networks. In *Advances in Neural Information Processing Systems*, pages 550–558, 2010.

[137] John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2012.

[138] K. I. Tsianos, S. Lawlor, and M. G. Rabbat. Push-sum distributed dual averaging for convex optimization. In *Proceedings of the 51st Annual Conference on Decision and Control*, pages 5453–5458. IEEE, 2012.

[139] D. Jakovetic, J. Xavier, and J. M.F. Moura. Cooperative convex optimization in networked systems: Augmented lagrangian algorithms with directed gossip communication. *IEEE Transactions on Signal Processing*, 59(8):3889–3902, 2011.

[140] João FC Mota, João MF Xavier, Pedro MQ Aguiar, and Markus Püschel. Distributed basis pursuit. *IEEE Transactions on Signal Processing*, 60(4):1942–1956, 2012.

[141] D. Jakovetic, J. M.F. Moura, and J. Xavier. Linear convergence rate of a class of distributed augmented lagrangian algorithms. *IEEE Transactions on Automatic Control*, 60(4):922–936, 2015.

[142] N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos. An augmented lagrangian method for distributed optimization. *Mathematical Programming*, 152(1-2):405–434, 2015.

[143] Tomaso Erseghe, Davide Zennaro, Emiliano Dall'Anese, and Lorenzo Vangelista. Fast consensus by the alternating direction multipliers method. *IEEE Transactions on Signal Processing*, 59(11):5523–5537, 2011.

[144] Ermin Wei and Asuman Ozdaglar. Distributed alternating direction method of multipliers. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 5445–5450, 2012.

[145] João FC Mota, João MF Xavier, Pedro MQ Aguiar, and Markus Püschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.

[146] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

[147] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.

[148] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

[149] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[150] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, volume 30. Siam, 1970.

[151] T.H. Chang. A proximal dual consensus admm method for multi-agent constrained optimization. *arXiv preprint arXiv:1409.3307*, 2014.

[152] T.H. Chang, M. Hong, and X. Wang. Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Transactions on Signal Processing*, 63(2):482–497, 2015.

[153] S. P. Bhat and D. S. Bernstein. Finite-time stability of continuous autonomous systems. *SIAM Journal on Control and Optimization*, 38(3):751–766, 2000.

[154] J. Cortés. Finite-time convergent gradient flows with applications to network consensus. *Automatica*, 42(11):1993–2000, 2006.

[155] S. Sundaram and C. N. Hadjicostis. Finite-time distributed consensus in graphs with time-invariant topologies. In *Proceedings of American Control Conference*, pages 711–716, 2007.

[156] L. Wang and F. Xiao. Finite-time consensus problems for networks of dynamic agents. *IEEE Transactions on Automatic Control*, 55(4):950–955, 2010.

[157] T. Charalambous, Y. Yuan, T. Yang, W. Pan, C. N Hadjicostis, and M. Johansson. Decentralised minimum-time average consensus in digraphs. In *Proceedings of 52nd IEEE Conference on Decision and Control*, pages 2617–2622. IEEE, 2013.

[158] Y. Yuan, G. Stan, L. Shi, M. Barahona, and J. Goncalves. Decentralised minimum-time consensus. *Automatica*, 49(5):1227–1235, 2013.

[159] J. M. Hendrickx, R. M. Jungers, A. Olshevsky, and G. Vankeerberghen. Graph diameter, eigenvalues, and minimum-time consensus. *Automatica*, 50(2):635–640, 2014.

[160] J. M. Hendrickx, G. Shi, and K. H. Johansson. Finite-time consensus using stochastic matrices with positive diagonals. *IEEE Transactions on Automatic Control*, 60(4):1070–1073, 2015.

[161] S. Safavi and U. A. Khan. Revisiting finite-time distributed algorithms via successive nulling of eigenvalues. *IEEE Signal Processing Letters*, 22(1):54–57, 2015.

[162] C.T. Chen. *Linear system theory and design*. Oxford University Press, 1995.

[163] M. Farina and R. Scattolini. Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica*, 48(6):1088Ű–1096, 2012.

[164] T. H. Summers and J. Lygeros. Distributed model predictive consensus via the alternating direction method of multipliers. *arXiv:1212.1296 [math.OC]*, 2012.

[165] P. Giselsson and A. Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Proc. of Conference on Decision and Control*, pages 7272–7277, Georgia, USA, December 2010.

[166] P. Giselsson, M. D. Doan, T. Keviczky, B. D. Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49(3):829ÍC833, 2013.

[167] J. Hopcroft and R. Tarjan. Efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1971.

[168] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563Ű581, 1977.

[169] R. Lougee-Heimer et al. The coin-or initiative: Open source accelerates operations research progress. *ORMS Today*, 28:20–22, 2001.

[170] C. J. Ong and Z. Wang. Reducing variables in model predictive control of linear system with disturbances using singular value decomposition. *Systems & Control Letters*, 71:62–68, 2014.

[171] Z. Wang and C. J. Ong. Distributed MPC of constrained linear systems with time-varying terminal sets. *Systems & Control Letters*, 88:14–23, 2016.

[172] Steven H Low and David E Lapsley. Optimization flow controlŮi: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, 1999.

[173] Vedrana Spudić, Christian Conte, Mato Baotić, and Manfred Morari. Cooperative distributed model predictive control for wind farms. *Optimal Control Applications and Methods*, 36(3):333–352, 2015.

[174] R. Halvgaard, L. Vandenberghe, N. K. Poulsen, H. Madsen, and J.B. Jorgensen. Distributed model predictive control for smart energy systems. *IEEE Transactions on Smart Grid*, 7(3):1675–1682, 2016.

[175] João FC Mota, João MF Xavier, Pedro MQ Aguiar, and Markus Püschel. Distributed admm for model predictive control and congestion control. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 5110–5115, 2012.

[176] M. Rubagotti, P. Patrinos, and A. Bemporad. Stabilizing linear model predictive control under inexact numerical optimization. *Automatic Control, IEEE Transactions on*, 59(6):1660–1666, 2014.

[177] P. Patrinos and A. Bemporad. An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Transactions on Automatic Control*, 59(1):18–33, 2014.

[178] Ion Necoara and Valentin Nedelcu. Rate analysis of inexact dual first-order methods application to dual decomposition. *IEEE Transactions on Automatic Control*, 59(5):1232–1243, 2014.

[179] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[180] Hao Zhu, Georgios B. Giannakis, and Alfonso Cano. Distributed in-network channel decoding. *IEEE Transactions on Signal Processing*, 57(10):3970–3983, 2009.

[181] D. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex analysis and optimization*. Athena Scientific, 2003.

[182] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[183] B. He and X. Yuan. On the O(1/n) convergence rate of the douglas-rachford alternating direction method. *SIAM Journal on Numerical Analysis*, 50(2):700–709, 2012.

[184] Z. Wang and C. J. Ong. Distributed model predictive control of linear discrete-time systems with coupled constraints. In *submitted to Proceedings of the 55st IEEE Conference on Decision and Control*, 2016.

[185] Z. Wang and C. J. Ong. Distributed model predictive control of linear discrete-time systems with local and global constraints. *submitted to Automatica*, 2016.

[186] P.P. Wellstead. *Introduction to Physical System Modelling*. Academic Press, 1979.

[187] A. Nedić, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.