ABSTRACT

LIAN, FEIER. Communication-Cost-Constrained Algorithms and Games for Multi-Agent Control Systems. (Under the direction of Alexandra Duel-Hallen and Aranya Chakrabortty).

This dissertation work focuses on communication-cost-constrained optimization and games in multi-agent control systems. In large-scale wide-area control systems, due to the need to feed back large volume of real-time sensor data to local and remote controllers, a dedicated wide-area communication infrastructure is required to guarantee secure and robust data communication, and that the economic efficiency of the communication network must be guaranteed. Throughout the thesis we take the communication cost of the underlying network as a constraint in the controller design for the wide-area control system. The controller design in this thesis is cast as an optimization problem to optimize a certain control performance index, subject to communication-cost constraints. In the multi-agent control system, when all agents have a global control performance objective, we solve a centralized (or social) optimization. When the agents have different objectives, we model the control strategies of multiple agents as a noncooperative game, and also develop cost-allocation algorithms. We first assume a known model and later in the thesis investigate control designs for systems with model uncertainty.

In Chapter 2, we investigate the social optimization and noncooperative games under communication-cost constraint for the multi-agent systems when the underlying system parameters are fixed and known to all agents. We model the communication cost as the number of communicating state/control input pairs across different nodes. Under the communication-cost constraint, we first investigate the scenario where the the agents cooperatively optimize a global performance objective. We develop a greedy algorithm for the sparsity-constrained centralized linear-quadratic regulator (LQR) problem based on gradient support pursuit (GraSP) method. Then, we consider the scenario where the agents aim to satisfy their own selfish objectives using a noncooperative differential game, and a distributed algorithm for solving the Nash equilibrium (NE) of the game is proposed. Moreover, we propose a cost allocation scheme, which fairly allocates costs of the communication infrastructure according to the agents' diverse needs for feedback and cooperation. The proposed algorithms and the cost allocation method are validated using numerical experiments on the Australian power system.

While in Chapter 2 we employ an idealistic assumption that the parameters for the underlying physical system are known to all agents, in real-world large-scale control networks, the system parameters might not be known exactly by the designer. To account for the model uncertainty, in Chapter 3, we investigate the centralized controller design problem with sparsity constraint under parametric uncertainty. We assume that the model uncertainty can be captured by the norm-bounded uncertainty in system state and input matrices, and we design a

sparsity-constrained mixed H_2/H_∞ controller, which is guaranteed to stabilize the closed-loop uncertain system, equivalent to minimizing the H_2 objective under the H_∞-norm and sparsity constraints. We solve the proposed minimization problem by first improving a previously proposed descent algorithm for mixed H_2/H_∞ control using a modified Zoutendijk's method. Then, we impose a sparsity constraint on this design by combining it with a greedy gradient support pursuit (GraSP) method. We illustrate the effectiveness of the proposed algorithm via a system of unstable nodes, where the proposed design can guarantee a predetermined level of sparsity while maintaining acceptable H_2 performance as well as H_∞ robustness.

Finally in Chapter 4, we extend the centralized sparsity-constrained design for uncertain systems in Chapter 3 to noncooperative games in uncertain multi-agent systems. First, we revisit the centralized mixed H_2/H_∞ control problem under the sparsity constraint. We improve on the GraSP-based method in Chapter 3 and propose a new algorithm based on the proximal alternating linearization method (PALM). Then we investigate a sparsity-constrained noncooperative game based on PALM for the mixed H_2/H_∞ control of multi-agent systems under a joint communication constraint. For this game we propose a best-response dynamics algorithm, which converges to an approximate generalized Nash equilibrium (GNE). Moreover, we show that when all agents have the same H_2 objective in this game, the algorithm for solving the approximate GNE serves as a partially distributed social optimization solution. Numerical experiments are run on the example of a network of unstable nodes, and the results show the advantages of the PALM-based algorithm over the GraSP-based algorithm as well as the convergence of the game.

Communication-Cost-Constrained Algorithms and Games for Multi-Agent Control Systems

by
Feier Lian

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Electrical Engineering

Raleigh, North Carolina

2019

APPROVED BY:

Dror Baron	Thayer Morrill
Alexandra Duel-Hallen Co-chair of Advisory Committee	Aranya Chakrabortty Co-chair of Advisory Committee

DEDICATION

To my parents.

BIOGRAPHY

Feier Lian received the B.E. degree in electrical engineering from Southeast University, Nanjing, China, in 2012. In the same fall, she joined the Ph.D. program of the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA, under the joint supervision of Dr. Alexandra Duel-Hallen and Dr. Aranya Chakrabortty. Her research interests include signal processing, control theory, power system, sparse optimization, noncooperative and cooperative game theory.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my co-advisors, Dr. Alexandra Duel-Hallen and Dr. Aranya Chakrabortty, for their guidance, support, patience, and insights. Thank you for sharing your expertise in communications, game theory, optimization, control theory and power systems with me, and I am grateful for your trust and encouragement during the difficult times of my research, your patience in helping me polish my writing and presentation. It is such an honor for me to work with you, I am so glad that we dived deep into the research problem together, it was a fulfilling journey.

I would also like to thank my committee members Dr. Dror Baron and Dr. Thayer Morrill for their encouragement, support, and insightful comments on my research work. Thanks for serving in my committee board. My extended thanks to all professors who have taught me in NC State University. Your excellent lectures have laid the foundations for me being able to explore new research problems.

I would like to thank my parents, without your continuous love and support I could not accomplish this much. Thank you for the encouraging words during my frustrated moments. I cherish the time we spent together during vacation, your homemade cooking and sense of humor can always cheer me up.

I would like to thank my fellow lab members, Yuan Lu, Mang Liao, Prathistha Shukla, Lu An, Amirhassan Fallah Dizche for your collaborative work and helpful discussions. I would also like to thank all other friends and fellow graduate students who study and work with me in Electrical and Computer Engineering department of NC State University. I remember the numerous hours we spent together attending lectures, working on homework and research problems, and sharing thoughts. Special thanks for my best friend Qian Ge for her support and company.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

Networked Control Systems (NCSs), such as power systems, sensor networks and autonomous vehicles, require real-time feedback of massive volumes of sensor data from one operating region of the physical network to controllers located in other regions [3]. One of the foremost requirements for NCS, is the need for a highly robust communication system that works in sync with the control functionalities. For example, the envisioned architecture of wide-area communication for the US power grid, often referred to as the North American Synchrophasor Initiative Network (NASPI-net) [4], involves sensors (Phasor Measurement Units (PMUs)) inside the operating boundaries of utility companies that send real-time data to local controllers via a local-area network and to remote controllers over a secure wide-area network. Each area is equipped with its own dedicated phasor gateway, which routes the incoming sensor datastreams to the respective controllers. In other applications of NCS such as disease monitoring, vision-based control of robots and autonomous vehicles [5], it is also economically desirable to optimize for sparse sensors or reduce the communication between sensors and controllers, due to either the high costs of sensors, or the need to identify key communication links to guarantee fast control decisions. This thesis is motivated by the following question - how can one guarantee the economic viability of this communication architecture? Installation of communication links for transporting feedback data from sensors to controllers requires a significant financial investment by the utilities.

Thanks to recent advances in sparse controller design, a few recent papers such as [6–12] have proposed economically lucrative NCS designs by sparsifying the number of communication links between sensors and controllers. In [6–8], sparsity-promoting static state feedback linear-quadratic regulator (LQR) controllers are designed using the alternating direction method of multipliers (ADMM), where the sparse controller design is cast as an optimization problem

to minimize the LQR cost together with an ℓ_1-norm penalty for the feedback matrix. Other works solve a cardinality minimization or cardinality constrained problem, by relaxing it to a convex problem, such as converting the cardinality constraint to a rank constraint [13, 14], or ℓ_1-norm constraint [15–17]. Another line of research investigates the design and performance of distributed control given the predefined sparsity and delay pattern [10, 18]. The sparsity promoting algorithms have been widely used in various NCSs for system identification or optimal sensor placement problems such as fluid flow control [5, 19] and biological networks [20].

In real world large-scale control systems, model uncertainties are bound to arise due to changing network operating conditions and topologies, or the mismatch between designers' knowledge and real network parameter values caused by idealistic assumptions or communication delay. The robust design of communication-cost-efficient controllers is investigated in [11, 15, 21–27]. The H_∞ control is used in [21, 22] to guarantee stability for closed-loop uncertain systems with parametric uncertainty, where the H_∞-norm constraint is translated into a linear matrix inequality (LMI). In [15], controllers with minimum number of communication links are designed for systems with polytopic uncertainty, which is equivalent to minimizing the cardinality of the feedback matrix subject to an H_∞ constraint. In [15], the ℓ_1 norm minimization problem is solved as a convex relaxation for the cardinality minimization problem. In [23], a sparse H_∞ controller is designed by exploiting the symmetric and sparse structure of the system state matrix. In [24], the robust stability analysis of sparsely interconnected uncertain systems is investigated using integral quadratic constraints, and it is shown that the sparse formulation of the stability analysis is equivalent to solving a set of sparse LMIs. In [11], a sparsity-promoting controller is designed for systems with norm-bounded uncertainty and delay, via minimizing the quadratic cost supplemented by the ℓ_1 norm of the feedback matrix under an LMI constraint. The work [26] introduces a notion of non-fragility for the state-feedback controller, which aims to provide stability guarantees for uncertain perturbations in the feedback matrix. Based on the non-fragility notion, a sparse state feedback controller is designed given the desired sparsity pattern. In [25], the robust non-fragility-based controller is designed with a row-column sparsity constraint, where the non-convex sparsity-constrained problem is reformulated as a convex rank-constrained problem.

In NCSs, the underlying physical dynamic system graph is usually spatially distributed, where the graph may be partitioned by geographical bounds or different economic priorities, and each such subsystem can be viewed as an agent. Multi-agent systems arise in many practical applications of NCS. In multi-agent systems, the agents can have the same objective, such as formation of autonomous vehicles [28], coordinating mobile senor networks [29], and motion planning for autonomous robots [30]. Contrary to seeking consensus, the agents may also have competing objectives, for example, cost allocation for communication cost in power systems [31, 32], congestion control in intelligent traffic systems [33], and resource allocation for wireless

networks [34]. Recently, significant research effort has been devoted to modeling decision making and analyzing interactions among agents using game theory [32, 33, 35–37]. In particular, the linear-quadratic differential game [38] is a family of games that is popular in control system applications where the system states evolve according to a differential equation, and each player chooses its own control strategy and has a quadratic cost function to minimize. The necessary conditions of Nash equilibrium (NE) for linear quadratic games and corresponding numerical algorithms are investigated in [39–41].

In multi-agent systems, robustness for control strategies has been investigated in [42–47]. In [42], the Nash-type differential game for known systems [38] was extended to systems with polytopic uncertainty, and the Nash equilibrium was found for guaranteed cost-control. In [43], the uncertain external disturbance is modeled as a fictitious player trying to maximize all players' costs, and the robust Nash strategies for all players are found by solving a coupled Riccati differential equation. The works [42, 43] extend the Nash-type differential game in [38] by finding robust Nash strategies while either considering polytopic uncertainty or formulating uncertain external disturbances as a fictitious player. The works [44–47] model the uncertainty in the multi-agent system using stochastic differential equations, and Nash strategies are found by solving cross-coupled matrix equations, using necessary optimality conditions or Karush-Kuhn-Tucker (KKT) conditions. In [44], the guaranteed cost control problem for uncertain stochastic systems with N decision makers is investigated. A similar work [45] investigates both the Pareto optimality and Nash equilibrium for a mixed H_2/H_∞ control problem for stochastic delay systems with multiple players. In [46], Stackelberg games for linear stochastic systems with multiple followers are studied, and the Nash strategies for noncooperative followers are developed. In [47], the Nash game and Pareto optimal optimization are studied for linear time-delay systems with Markovian jumping parameters. In [48], gain-scheduled Nash games with H_∞ constraints for stochastic linear parameter varying (LPV) systems are studied, where the Nash strategies are developed for the worst-case disturbance. Moreover, reinforcement learning (RL) has been applied to multi-agent control problems when the system parameters are completely or partially unknown [49–53], where the work [53] investigated sparsity in the feedback matrix.

Motivated by the challenge to develop an economically feasible communication network for NCSs, in this thesis we first develop a communication-cost constrained centralized feedback control design for dynamic systems with known model parameters. Then we investigate the communication-cost-constrained controller for systems with norm-bounded parametric uncertainty, which can guarantee closed-loop stability. Throughout our work, we consider linear static feedback for the control system, and the communication cost is defined on the number of sensor/controller feedback links. Moreover, we extend the communication-cost social optimization to multi-agent systems, where different agents have different performance objectives while satisfying to a global communication-cost constraint. For the communication-cost-constrained

noncooperative game, we first investigate the NE solution when model parameters are known to all agents, and then study the Generalized Nash Equilibrium of the noncooperative game when the model uncertainty is translated into a shared H_∞-norm bound on all the agents.

This thesis is organized as follows. First, in Chapter 2, we investigate the social optimization and noncooperative games under communication-cost constraints for multi-agent systems when the underlying system parameters are fixed and known to all agents. We model the communication cost as the number of communicating state/control input pairs across different nodes. Under the communication-cost constraint, we first investigate the scenario where the agents cooperatively optimize a global performance objective. We develop a greedy algorithm for the sparsity-constrained centralized LQR problem based on the gradient support pursuit (GraSP) method. Next, we consider the scenario where the agents aim to satisfy their own selfish objectives using a noncooperative differential game, and a distributed algorithm for solving the NE of the game is proposed. Moreover, we propose a cost allocation scheme, which allocates costs of the communication infrastructure fairly according to the agents' diverse needs for feedback and cooperation. The proposed algorithms and the cost allocation method are validated using numerical experiments on the Australian power system.

While in Chapter 2 we employ an idealistic assumption that the parameters for the underlying physical system are known to all agents, in real-world large-scale control networks, the system parameters might not be known exactly by the designer. To account for the model uncertainty, in Chapter 3 we investigate the centralized controller design problem with a sparsity constraint under parametric uncertainty. We assume that the model uncertainty can be captured by the norm-bounded uncertainty in the system state and control matrices, and design a sparsity-constrained mixed H_2/H_∞ controller, which is guaranteed to stabilize the closed-loop uncertain system, equivalent to minimizing the H_2 objective under the H_∞-norm and sparsity constraints. We solve the proposed minimization problem by first improving a previously proposed descent algorithm for mixed H_2/H_∞ control using a modified Zoutendijk's method. Then, we impose a sparsity constraint on this design by combining it with a greedy GraSP method. We illustrate the effectiveness of the proposed algorithm via a system of unstable nodes, where the proposed design can guarantee a predetermined level of sparsity while maintaining acceptable H_2 performance as well as H_∞ robustness.

Finally in Chapter 4 we extend the centralized sparsity-constrained design for uncertain systems of Chapter 3 to noncooperative games in uncertain multi-agent systems. First, we revisit the centralized mixed H_2/H_∞ control problem under the sparsity constraint. We improve on the GraSP-based method of Chapter 3 and propose a new algorithm based on the proximal alternating linearization method (PALM). Then we investigate a sparsity-constrained noncooperative game based on PALM for mixed H_2/H_∞ control of multi-agent systems under a joint communication constraint. For this game we propose a best-response dynamics algo-

rithm, which converges to an approximate generalized Nash equilibrium (GNE). Moreover, we show that when all agents have the same H_2 objective in this game, the algorithm for solving the approximate GNE serves as a partially distributed social optimization solution. Numerical experiments are run on the example of a network of unstable nodes, and the results show the advantages of the PALM-based algorithm over the GraSP-based algorithm as well as the convergence of the game.

1.2 Preliminaries

1.2.1 Strategic-form game and Nash Equilibrium

In this subsection, we review some basic concepts for strategic-form games, which are used in Chapters 2 and 4 of this thesis. In both chapters, we consider two kinds of games. In one case, the players act noncooperatively, where each player tries to optimize its own objective, i.e., noncooperative game. In the other case, the players act cooperatively, and optimize one social objective, i.e., cooperative game. In Chapter 2, we investigate cost allocation among the players when they form the cooperation, thus we develop the bargaining game.

Next, we briefly review games in strategic form, which are employed in Chapters 2 and 4 when we consider noncooperative games. To define a game in strategic form, we need to specify the set of players in the game, the set of actions available to each player, and the way the players' payoffs depend on the actions that they choose.

The strategic-form game has the following form,

$$\Gamma = (\mathcal{N}, (\mathcal{A}_i)_{i\in\mathcal{N}}, (u_i)_{i\in\mathcal{N}}), \tag{1.1}$$

where $\mathcal{N}$ is is the set of players in the game Γ, $\mathcal{A}_i$ is the set of strategies (or pure strategies) available to player i, and u_i is the payoff function, $\times_{j\in\mathcal{N}}\mathcal{A}_j \to \mathbb{R}$. When the strategic-form game Γ is played, each player i must choose one of the strategies in set $\mathcal{A}_i$. A strategy profile is a combination of strategies that the players in $\mathcal{N}$ might choose. Let $\mathcal{A}$ denote the set of all possible strategy profiles,

$$\mathcal{A} = \times_{i\in\mathcal{N}}\mathcal{A}_i. \tag{1.2}$$

For any strategy profile $a = (a_j)_{j\in\mathcal{N}}$, the value $u_i(a)$ represents the payoff that player i would obtain if a were the combination of strategies implemented by the players. In all strategic-form games studied in this thesis, the players choose their strategies simultaneously. This kind of game is also called a one-shot simultaneous-move game [54].

Under the assumptions that each player chooses his or her action according to the model of rational choice given his or her belief about other players' actions, and whether his or her

belief about other players' actions is correct [55], the NE can depict the steady state, where each player chooses the best strategy available based on other players' strategies. Let a be a strategy profile, and a_{-i} the combination of strategies in a of all players except i, then (a'_i, a_{-i}) denotes the strategy profile where player i choose a'_i, and all other players choose a_{-i}. The NE can be formally defined as follows [55]:

The strategy profile a^* in a strategic game is a Nash Equilibrium if for every player i and every strategy a_i of player i, a^* is at least as good according to player i's preferences as the strategy profile (a_i, a^*_{-i}), where player i chooses a_i while every other player j chooses a^*_j, i.e.,

$$u_i(a^*) \geq u_i(a_i, a^*_{-i}) \ \forall a_i \in \mathcal{A}_i. \tag{1.3}$$

1.2.2 Kurdyka-Łojasiewicz (KL) Property

The Kurdyka-Łojasiewicz (KL) property is used in Chapter 4 for convergence analysis. Here we present some basics of the KL property introduced in [56]. First we define some notation.

Definition 1.2.1. (Distance.) For any subset $\mathcal{S} \subset \mathbb{R}^d$ and any point $x \in \mathbb{R}^d$, the distance from x to $\mathcal{S}$ is defined and denoted by

$$\text{dist}(\mathbf{x}, \mathcal{S}) := \inf\{||\mathbf{y} - \mathbf{x}|| : \mathbf{y} \in \mathcal{S}\}. \tag{1.4}$$

When $\mathcal{S} = \emptyset$, we have $\text{dist}(\mathbf{x}, \mathcal{S}) = \infty$ for all $\mathbf{x}$.

Let $\eta \in [0, \infty]$. We denote by Φ_η the class of all concave and continuous functions $\varphi : [0, \eta) \to \mathbb{R}_+$ that satisfy the following conditions:
(i) $\varphi(0) = 0$;
(ii) φ has first order continuous derivative on $(0, \eta)$ and continuous at 0;
(iii) for all $s \in (0, \eta) : \varphi'(s) > 0$.

We first review the definition of proper and lower semicontinuous functions:

Definition 1.2.2. (Proper) The function $\sigma : \mathcal{S} \to \mathbb{R}$ is a proper function if $\sigma(\mathbf{x}) > -\infty$ for all $\mathbf{x} \in \mathcal{S}$, and $\sigma(\mathbf{x}) < \infty$ for at least one point $\mathbf{x} \in \mathcal{S}$.

Definition 1.2.3. (Lower semicontinuous) The function $\sigma : \mathcal{S} \to \mathbb{R}$ is lower semicontinuous at $\bar{\mathbf{x}} \in \mathcal{S}$ if for all $\epsilon > 0$ there exists a δ such that $\mathbf{x} \in \mathcal{S}$ and $||\mathbf{x} - \hat{\mathbf{x}}|| < \delta$ imply $\sigma(\mathbf{x}) - \sigma(\bar{\mathbf{x}}) > -\epsilon$.

For proper and lower semicontinous functions, the subdifferentials are defined below [56]:

Definition 1.2.4. (Subdifferentials) Let $\sigma : \mathbb{R}^d \to (-\infty, \infty]$ be a proper and lower semicontinous function.
(i) For a given $\mathbf{x} \in \text{dom}(\sigma)$, the Fréchet subdifferential of σ at x, written $\hat{\partial}\sigma(\mathbf{x})$, is the set of

all vectors $\mathbf{u} \in \mathbb{R}^d$ that satisfy

$$\lim_{\mathbf{y}\neq\mathbf{x}\,\mathbf{y}\to x} \inf \frac{\sigma(\mathbf{y}) - \sigma(\mathbf{x}) - \langle \mathbf{u}, \mathbf{y}-\mathbf{x}\rangle}{||\mathbf{y}-\mathbf{x}||} \geq 0. \tag{1.5}$$

When $\mathbf{x} \notin \text{dom}(\sigma)$, we set $\hat{\partial}\sigma(\mathbf{x}) = \emptyset$.
(ii) The limiting subdifferential, or subdifferential, of σ at $\mathbf{x} \in \mathbb{R}^d$, written $\partial\sigma(\mathbf{x})$, is defined through the following,

$$\partial\sigma(\mathbf{x}) := \left\{\mathbf{u} \in \mathbb{R}^d : \exists \mathbf{x}^k \to \mathbf{x}, \sigma(\mathbf{x}^k) \to \sigma(\mathbf{x}) \text{ and } \mathbf{u}^k \in \hat{\partial}\sigma(\mathbf{x}^k) \to \mathbf{u} \text{ as } k \to \infty\right\} \tag{1.6}$$

Note that points whose subdifferentials contain 0 are called *(limiting-)critical points.*

The KL property is defined as follows.

Definition 1.2.5. (Kurdyka-Łojasiewicz (KL) Property) Let $\sigma : \mathbb{R}^d \to (-\infty, +\infty]$ be proper and lower semicontinuous.
(i) The function σ is said to have the KL Property at $\bar{\mathbf{u}} \in \text{dom}(\partial\sigma) := \{\mathbf{u} \in \mathbb{R}^d : \partial\sigma(\mathbf{u}) \neq \emptyset\}$ if there exists $\eta \in (0, \infty]$, a neighborhood $\mathcal{U}$ of $\bar{\mathbf{u}}$ and a function $\varphi \in \Phi_\eta$, such that for all

$$\mathbf{u} \in \mathcal{U} \cap [\sigma(\bar{\mathbf{u}}) < \sigma(\mathbf{u}) < \sigma(\bar{\mathbf{u}}) + \eta], \tag{1.7}$$

the following inequality holds,

$$\varphi'(\sigma(\mathbf{u}) - \sigma(\bar{\mathbf{u}}))\text{dist}(0, \partial\sigma(\mathbf{u})) \geq 1. \tag{1.8}$$

(ii) If σ satisfies the KL property at each point of $\text{dom}(\partial\sigma)$, then σ is called a KL function.

It is shown in [56] that KL functions arise in many applications of optimization, in particular, semi-algebraic functions are KL functions. The definitions for semi-algebraic functions are given as follows.

Definition 1.2.6. (Semi-algebraic sets and functions). (i) A subset $\mathcal{S} \in \mathbb{R}^d$ is a real semi-algebraic set if there exists a finite number of real polynomial functions, $g_{ij}, h_{ij} : \mathbb{R}^d \to \mathbb{R}$, such that

$$\mathcal{S} = \cup_{j=1}^{p} \cap_{i=1}^{q} \left\{\mathbf{u} \in \mathbb{R}^d : g_{ij}(\mathbf{u}) = 0 \text{ and } h_{ij}(\mathbf{u}) < 0\right\}. \tag{1.9}$$

(ii) A function $h : \mathbb{R}^d \to (-\infty, +\infty]$ is called semi-algebraic if its graph

$$\left\{(\mathbf{u}, t) \in \mathbb{R}^{d+1} : h(\mathbf{u}) = t\right\} \tag{1.10}$$

is a semi-algebraic subset of $\mathbb{R}^{d+1}$.

Chapter 2

Game-Theoretic Multi-Agent Control and Network Cost Allocation under Communication Constraints

Multi-agent networked linear dynamic systems have attracted attention of researchers in power systems, intelligent transportation, and industrial automation. The agents might cooperatively optimize a global performance objective, resulting in social optimization, or try to satisfy their own selfish objectives using a noncooperative differential game. However, in these solutions, large volumes of data must be sent from system states to possibly distant control inputs, thus resulting in high cost of the underlying communication network. To enable economically-viable communication, a game-theoretic framework is proposed under the *communication cost*, or *sparsity*, constraint, given by the number of communicating state/control input pairs. As this constraint tightens, the system transitions from dense to sparse communication, providing the trade-off between dynamic system performance and information exchange. Moreover, using the proposed sparsity-constrained distributed social optimization and noncooperative game algorithms, we develop a method to allocate the costs of the communication infrastructure fairly and according to the agents' diverse needs for feedback and cooperation. Numerical results illustrate utilization of the proposed algorithms to enable and ensure economic fairness of wide-area control among power companies.

2.1 Introduction

Multi-agent networked dynamic systems arise in many practical scenarios where the communicating entities are spatially separated or have different economic priorities, e.g., in cyber-physical power networks, multi-vehicle formation, intelligent transportation, industrial automation, etc [57]. Social optimization can be performed when all agents aim to jointly optimize a system-wide objective while a noncooperative differential game is suitable when their objectives are different [38]. The linear-quadratic regulator (LQR) optimization objectives are frequently employed in the literature due to their tractability, feasibility of distributed implementation, and broad applicability of the quadratic utility function [38–40, 58].

A networked multi-agent dynamic system with multiple nodes is illustrated in Fig.2.1. Every agent owns a subset of nodes, where each node contains several states and control inputs. To achieve a desired performance objective, it is often necessary to employ state or output feedback. In this chapter, frequently used assumptions of LQR optimization and static state feedback are employed [58]. Without loss of generality, we define the feedback links from states to control inputs within one node as *local feedback links*, which incur negligible expense, and the feedback links across different nodes as *communication links*. The traditional state-feedback centralized LQR optimization [58] and the linear-quadratic games [38–40] require a dense feedback matrix and, thus, communication links from every state to every control input, which necessitates significant information exchange among the system nodes, and, thus, large communication infrastructure investment to assure desired rate and delay constraints. Due to wide

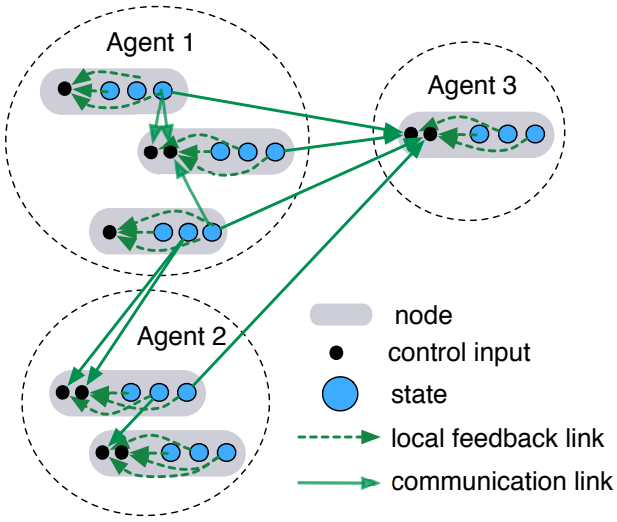


Figure 2.1: The communication structure of the multi-agent system.

applicability of proposed methods, we do not assume specific communication medium or network topology. Instead, we address the following question: how to reduce the *communication cost*, given by the number of communication links (see Fig.2.1), while maintaining desired control objectives? To answer this question, we develop a family of *sparse* designs, which provide a trade-off between the communication cost and control performance, and reveal the most critical communication links. By limiting the number of communicating state/control input pairs, we reduce the overall bill for leasing bandwidth in an existing network or investment in a dedicated communication infrastructure. However, computation of the actual economic benefit depends on specific communication technology and application and is beyond the scope of this chapter.

Sparsity-constrained optimization has been investigated in [59, 60], and the state-feedback optimal-control LQR algorithms for sparsity promotion were addressed in [6, 7]. However, these methods employ global optimization objectives and centralized implementation, which limit their applicability to multi-agent systems. Moreover, distributed approaches in the literature [10,61] cannot accommodate specified sparsity constraints and different optimization objectives of the agents.

In this chapter, we investigate LQR optimization for dynamic systems with linear state feedback *under the constraint on the communication cost*, i.e. the *sparsity* constraint, given by the number of communication links, expressed in terms of the off-diagonal cardinality of the state feedback matrix. First, to solve the *centralized* sparsity-constrained optimization problem, we employ the greedy Gradient Support Pursuit (GraSP) algorithm [59], which was shown to provide accurate approximations to sparsity-constrained solutions for a wide class of optimization functions. The proposed method also utilizes the restricted Newton step [62] to speed up convergence. Second, we develop a *noncooperative linear-quadratic game* among the agents, under the global communication cost constraint. To compute a Nash equilibrium of this game, we combine the ideas of GraSP and iterative gradient descent approaches [36, 63]. In the resulting algorithm, the computation of the players' (agents') utilities is *distributed* and requires limited information exchange. Third, we convert the proposed noncooperative game into a potential game [36] where the players' utilities agree, thus producing a sparsity-constrained *distributed social optimization*. The games developed in this chapter can be viewed as *Network Formation Games* (NFGs) [64] since players take strategic moves to form a network from states to control inputs. Moreover, using the above algorithms, we apply cooperative NFG theory [64] and the Nash Bargaining Solution (NBS) [65, 66] to allocate the costs of communication among the agents proportionally to the benefits they derive from sparsity-constrained feedback and cooperation. This *network cost allocation method* improves on our previous WAC cost allocation approaches [31, 67], which employed heuristics, relied on the centralized optimization in [7], and extrapolated the costs of a dense network [40] to sparse scenarios. Finally, we present numerical results for an example of wide-area control (WAC) of power systems, which helps in

suppression of inter-area power oscillations, but potentially requires a substantial investment in the communication network needed to exchange state feedback information [4, 7, 67–71]. These results are shown vs. the sparsity constraint, from dense feedback [40] to the decentralized implementation [67], thus illustrating the trade-off between the communication cost and the control performance.

The main contributions of this chapter are:

- Development and analysis of centralized and distributed social optimization algorithms and a noncooperative linear-quadratic game for multi-agent LQR optimal control with static linear state feedback under the constraint on the number of communicating state-control input pairs;
- Development of fair network cost allocation algorithm under sparsity constraints;
- Enabling sparsity-constrained designs and network cost allocation for a multi-area power system that employs wide-area control.

This chapter is organized as follows. The system model and the communication-cost-constrained social optimization is presented in Section 2.2. In Section 2.3, the multi-agent system model is developed, and the sparsity-constrained distributed differential games are discussed. Section 2.4 describes the proposed network cost allocation algorithm. In Section 2.5, we present an example of utilizing the proposed methods for WAC of power systems. Numerical results and discussion for the Australian power system example are contained in Section 2.6. Finally, some concluding remarks are made in Section 2.7.

2.2 System Model and the Communication-Cost-Constrained Centralized Optimization

The linear dynamic system with n nodes illustrated in Fig.2.1 is described by the following state-space equation.

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{D}w(t),\ \mathbf{x}_0(t) = \mathbf{0}. \tag{2.1}$$

where $\mathbf{x}(t)=(\mathcal{X}_1^{\mathrm{T}}(t), ..., \mathcal{X}_n^{\mathrm{T}}(t))^{\mathrm{T}}\in\mathbb{R}^{m\times 1}$ is the vector of states, $\mathcal{X}_j\in\mathbb{R}^{m_j\times 1}$ is the vector of states for node $j\in\{1, ..., n\}$, m_j is the number of states in node j, $m=\sum_{j=1}^{n} m_j$, $\mathbf{u}(t) = (\mathcal{U}_1(t)^{\mathrm{T}}, ..., \mathcal{U}_n(t)^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{q\times 1}$ is the vector of control inputs, $\mathcal{U}_j \in \mathbb{R}^{p_j\times 1}$ is the the vector of control input of node j, p_j is the number of control inputs in node j, $q=\sum_{j=1}^{n} p_j$, $w(t)$ is a scalar impulse disturbance input, and $\mathbf{A}, \mathbf{B}, \mathbf{D}$ are matrices with appropriate dimensions, among which matrix $\mathbf{A}$ determines the physical topology of the system [58].

We assume linear static feedback is employed, and thus the control input satisfies

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \tag{2.2}$$

where $\mathbf{K} \in \mathbb{R}^{q\times m}$ is the feedback gain matrix, with $\mathbf{u}(t) = (u_1(t), ..., u_q(t))^{\mathrm{T}}$, and $\mathbf{x}(t) = (x_1(t), ..., x_m(t))^{\mathrm{T}}$. If the coefficient $K_{ij} \neq 0$, the system (shown in Fig.2.1) contains a communication link that delivers the data of state j to control input i. We will refer to the tuple $(x_j(t), u_i(t))$ where $j = 1, ..., m$, and $i = 1, ..., q$, as a *state-control input link* in the rest of this chapter. Since the states $\mathbf{x}(t)$ and the control inputs $\mathbf{u}(t)$ are organized according to their physical locations, the matrix $\mathbf{K}$ is in the form

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \cdots & \mathbf{K}_{1n} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \cdots & \mathbf{K}_{2n} \\ & & \vdots & \\ \mathbf{K}_{n1} & \mathbf{K}_{n2} & \cdots & \mathbf{K}_{nn} \end{bmatrix} \tag{2.3}$$

where the block $\mathbf{K}_{ij} \in \mathbb{R}^{p_i\times m_j}$ represents feedback of the states of node j to the control inputs of node i, with $i = j$ corresponding to local feedback and $i \neq j$ — to communication links (see Fig.2.1). Without loss of generality, we define the communication cost as the number of communication links associated with the off-diagonal blocks of $\mathbf{K}$, given by

$$\mathrm{card}_{\mathrm{off}}(\mathbf{K}) = \sum_{i,j=1,i\neq j}^{n} \mathrm{nnz}(\mathbf{K}_{ij}) \tag{2.4}$$

where $\mathrm{nnz}(\cdot)$ operator counts the number of nonzero elements in a matrix. The proposed algorithms can be easily adapted to other sparsity criteria. Additional notation used in the algorithms is described in Table I.

For the model (2.1, 2.2), the *social global* LQR objective function is given by [58]

$$J(\mathbf{K}) = \int_{t=0}^{\infty} [\mathbf{x}(t)^T \mathbf{Q}\mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R}\mathbf{u}(t)]dt \tag{2.5}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the positive semidefinite and positive definite design matrices with dimensions $m \times m$ and $q \times q$, respectively. The minimization of (2.5) with respect to $\mathbf{K}$ results in dense feedback, i.e., communication links from every state to every control input.

Table 2.1: Notation used in the Algorithms 2.1 and 2.2

Term	**Definition**
$\|\|\mathbf{K}\|\|_2$	Frobenius norm of the matrix $\mathbf{K}$, defined by trace($\mathbf{K}^{\mathrm{T}}\mathbf{K}$).
supp($\mathbf{K}$)	The support set of the matrix $\mathbf{K}$, i.e., the set of indices of the nonzero entries of matrix $\mathbf{K}$ [59].
$[\mathbf{K}]_s$	The matrix obtained by preserving only the s largest-magnitude entries of the matrix $\mathbf{K}$, and setting all other entries to zero.
$\mathbf{K}^{\mathrm{off}}$	The matrix obtained by preserving only the off-diagonal blocks of the matrix $\mathbf{K}$ (see (2.3)) and setting all other entries to zero.
$\mathbf{K}^{\mathrm{diag}}$	The matrix obtained by preserving the diagonal blocks of the matrix $\mathbf{K}$ and setting all other entries to zero.
$\nabla_{\mathbf{K}} J(\mathbf{K})$	The gradient of the scalar function $J(\mathbf{K})$ with respect to the matrix $\mathbf{K}$ [72]. Assuming $\mathbf{K} \in \mathbb{R}^{m\times n}$, $\nabla_{\mathbf{K}} J(\mathbf{K})$ is given by a $m \times n$ matrix with the elements $[\nabla_{\mathbf{K}} J(\mathbf{K})]_{ij} = \partial J/\partial K_{ij}$.
$\nabla_{\mathbf{K}} J(\mathbf{K})\|_{\mathcal{T}}$	The gradient of the scalar function $J(\mathbf{K})$ with respect to the matrix $\mathbf{K}$ projected onto the index set $\mathcal{T}$. The matrix $\nabla_{\mathbf{K}} J(\mathbf{K})\|_{\mathcal{T}}$ is obtained by preserving only the entries of $\nabla_{\mathbf{K}} J(\mathbf{K})$ with indices in the set $\mathcal{T}$ and setting other entries to zero.
$\Delta_{\mathrm{nwt}}(\mathbf{K}, \mathcal{T})$	The restricted Newton step of function $f(\mathbf{K})$ at matrix $\mathbf{K} \in \mathbb{R}^{m\times n}$ under the structural constraint supp($\mathbf{K}$) $\subset \mathcal{T}$. First, the $mn\times 1$ vector $\mathbf{x}$ is computed by stacking the columns of $\mathbf{K}$, and the function $g(\mathbf{x})$ is defined as $g(\mathbf{x}) \triangleq f(\mathbf{K})$. Then the $mn\times 1$ restricted Newton step vector $\Delta_{\mathrm{nwt}}(\mathbf{x}, \mathcal{T})$ of $g(\mathbf{x})$ at $\mathbf{x}$ [59] is computed using the conjugate gradient (CG) method [6]. The vector $\Delta_{\mathrm{nwt}}(\mathbf{x}, \mathcal{T})$ is then converted into an $m\times n$ matrix by stacking the consecutive $m\times 1$ segments of $\Delta_{\mathrm{nwt}}(\mathbf{x}, \mathcal{T})$.

Next, we formulate the *social optimization under the communication cost constraint s*:

$$\begin{aligned} \min_{\mathbf{K}} \quad & J(\mathbf{K}) \\ \text{s.t.} \quad & \mathrm{card}_{\mathrm{off}}(\mathbf{K}) \leq s \\ & \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{D}w(t) \\ & \mathbf{u}(t) = -\mathbf{K}x(t) \end{aligned} \tag{2.6}$$

The optimization (2.6) produces a system with at most s state-control input links. Direct solution of (2.6) can have combinatorial complexity [59], so we utilize a numerically efficient *GraSP* method [59] in the proposed *centralized Algorithm 2.1*. Given the overall sparsity constraint s, in each iteration of Step 2(1–4), the algorithm extends the matrix $\mathbf{K}$ along its steepest $2s$

Algorithm 2.1 Minimizing the centralized LQR objective under the global communication cost constraint s.

1 *Initialization*
$\mathbf{K} := \mathbf{K}_0$
2 *Iteration*
while stopping criterion are not met **do**
(0) $\mathbf{K}^{\text{prev}} := \mathbf{K}$
(1) Compute gradient of $J(\mathbf{K})$ w.r.t $\mathbf{K}$: $\mathbf{g} = \nabla_{\mathbf{K}} J(\mathbf{K})$
(2) Identify up to $2s$ off-diagonal block directions: $\mathcal{Z} = \text{supp}([\mathbf{g}^{\text{off}}]_{2s})$
(3) Merge support: $\mathcal{T} = \mathcal{Z} \cup \text{supp}(\mathbf{K})$.
(4) Descend using the Newton step of J restricted to $\mathcal{T}$: $\mathbf{K} := \mathbf{K} + \lambda \Delta_{\text{nwt}}(\mathbf{K}, \mathcal{T})$.
(5) Prune communication links: $\mathbf{K} := \mathbf{K}^{\text{diag}} + [\mathbf{K}^{\text{off}}]_s$
(6) Stopping criterion:
$||\mathbf{K} - \mathbf{K}^{\text{prev}}||_2 < \epsilon_{\text{abs}} \sqrt{qm} + \epsilon_{\text{rel}} ||\mathbf{K}^{\text{prev}}||_2$
end while
3 *Polishing*
$\mathcal{I} = \text{supp}(\mathbf{K})$
while not $||\nabla_{\mathbf{K}} J|_{\mathcal{I}}||_2 / \sqrt{qm} < \epsilon_2$ **do**
Descend using the Newton step of J restricted to $\mathcal{I}$:
$\mathbf{K} := \mathbf{K} + \lambda \Delta_{\text{nwt}}(\mathbf{K}, \mathcal{I})$.
end while

gradient-descent directions. In Step 2(1), the gradient of $\mathbf{J}(\mathbf{K})$ over $\mathbf{K}$ is computed as [62]

$$\nabla_{\mathbf{K}} J(\mathbf{K}) = 2(\mathbf{R}\mathbf{K} - \mathbf{B}^{\mathrm{T}}\mathbf{P})\mathbf{L} \tag{2.7}$$

where the matrices $\mathbf{P}$ and $\mathbf{L}$ are the unique solutions of the following Lyaponuv equations

$$\begin{aligned} (\mathbf{A} - \mathbf{B}\mathbf{K})^{\mathrm{T}}\mathbf{P} + \mathbf{P}(\mathbf{A} - \mathbf{B}\mathbf{K}) + \mathbf{Q} + \mathbf{K}^{\mathrm{T}}\mathbf{R}\mathbf{K} = 0 \\ (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{L} + \mathbf{L}(\mathbf{A} - \mathbf{B}\mathbf{K})^{\mathrm{T}} + \mathbf{D}\mathbf{D}^{\mathrm{T}} = 0 \end{aligned} \tag{2.8}$$

In Step 2(4), the matrix $\mathbf{K}$ is updated using the restricted Newton step, and the step size λ is chosen via the Armijo line search [73]. In Step 2(5), pruning is performed to impose the constraint s. To guarantee stability of the feedback matrix after pruning, we provide a backtracking option to return to a previously found stable solution, which has $s-1$ or fewer communication links. (The stopping criterion in Step 2(6) was also used to determine convergence in the sparsity-promotion algorithm [7].) Note that in Step 2(6), q is the total number of control inputs, and m is the total number of states. Note also that as s grows, the algorithm retains the links that are the most *critical* for minimizing the objective (2.5).

2.3 Multi-Agent System Model and Communication-Cost-Constrained Linear-Quadratic Games

Suppose that there are r agents, where the agent i owns n_i nodes in the system (2.1) as shown in Fig.2.1. Without loss of generality, the nodes are partitioned as follows

$$\begin{aligned} \mathcal{S}_1 = \{1, 2, ..., n_1\} &\Rightarrow \text{belongs to agent 1.} \\ \mathcal{S}_2 = \{n_1 + 1, n_1 + 2, ..., n_1 + n_2\} &\Rightarrow \text{belongs to agent 2.} \\ \dots &\Rightarrow \dots \\ \mathcal{S}_r = \{n_1 + n_2 + ... + n_{r-1} + 1, ..., n\} &\Rightarrow \text{belongs to agent } r. \end{aligned} \tag{2.9}$$

We can rewrite the states and control inputs of each agent i in (2.1) as follows:

$$\dot{\mathbf{x}}_i(t) = \sum_{k=1}^{r} \mathbf{A}_{ik}\mathbf{x}_k(t) + \sum_{k=1}^{r} \mathbf{B}_{ik}\mathbf{u}_k(t) + \mathbf{D}_i w(t) \tag{2.10}$$

$$\begin{bmatrix} \mathbf{u}_1(t) \\ \vdots \\ \mathbf{u}_r(t) \end{bmatrix} = - \begin{bmatrix} \mathbf{K}^1 \\ \vdots \\ \mathbf{K}^r \end{bmatrix} \cdot \mathbf{x}(t) \tag{2.11}$$

where $\mathbf{x}_i(t) = (\mathcal{X}^{\mathrm{T}}_{n_1+...+n_{i-1}+1}(t), ..., \mathcal{X}^{\mathrm{T}}_{n_1+...+n_i}(t))^{\mathrm{T}} \in \mathbb{R}^{M_i \times 1}$ is the vector of states for agent i, with $M_i = \sum_{j=n_1+...+n_{i-1}+1}^{n_1+...+n_i} m_j$; $\mathbf{u}_i(t) = [\mathcal{U}^{\mathrm{T}}_{n_1+...+n_{i-1}+1}(t), \dots, \mathcal{U}^{\mathrm{T}}_{n_1+...+n_i}(t)]^{\mathrm{T}} \in \mathbb{R}^{N_i \times 1}$ is the vector of control inputs for agent i, with $N_i = \sum_{j=n_1+...+n_{i-1}+1}^{n_1+...+n_i} p_j$, $\mathbf{K}^i$ is the submatrix of $\mathbf{K}$ associated with the control inputs of the agent i, $\mathbf{D}_i$ is the control matrix for the disturbance input that enters agent i, and $\mathbf{D} = \mathrm{col}(\mathbf{D}_1, ..., \mathbf{D}_r)$.

Next, we briefly summarize results on linear-quadratic games for this system [38, 40]. The agents in (2.9) are viewed as players that optimize their individual, or selfish, objectives J_i by selecting their control inputs $\mathbf{u}_i(t)$, for $i = 1, ..., r$. The *selfish* objective of the player i is given by

$$J_i(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_r) = \int_{t=0}^{\infty} [\mathbf{x}(t)^{\mathrm{T}}\mathbf{Q}_i\mathbf{x}(t) + \mathbf{u}_i(t)^{\mathrm{T}}\mathbf{R}_i\mathbf{u}_i(t)]dt \tag{2.12}$$

where $\mathbf{R}_i \in \mathbb{R}^{N_i \times N_i}$, and $\mathbf{Q}_i \in \mathbb{R}^{m \times m}$ are positive semidefinite and positive definite matrices, respectively, chosen to improve the i^{th} user's objective. A *Nash Equilibrium (NE)* is achieved when it is impossible for any player to improve its objective function by unilaterally changing its strategy. At a given NE, the players employ Nash strategies $(\mathbf{u}_1^*(t), \mathbf{u}_2^*(t), ..., \mathbf{u}_r^*(t))$ defined

as [38]

$$J_i(\mathbf{u}_i^*(t), \mathbf{u}_{-i}^*(t)) \leq J_i(\mathbf{u}_i(t), \mathbf{u}_{-i}^*(t)), \forall \mathbf{u}_i(t), t=[0, \infty) \tag{2.13}$$

for $\forall i \in \{1, ...r\}$, where $\mathbf{u}_{-i}(t) := (\mathbf{u}_1(t), \ldots, \mathbf{u}_{i-1}(t), \mathbf{u}_{i+1}(t), \ldots, \mathbf{u}_r(t))$ is the tuple of strategies formed by all players except for the player i. When state feedback is employed (2.11), the Nash strategies $\mathbf{u}_i^*(t)$ in eq.(2.13) can be determined by solving the cross-coupled algebraic Riccati equations (CARE) (eq.(8) in [40]), where the solution to CARE exists and is unique when the system is weakly coupled. However, CARE produces a dense feedback matrix [40], and, thus, communication links from every state to every control input.

To limit the communication cost in a noncooperative scenario, we formulate a *linear-quadratic noncooperative NFG* where the players can establish at most s state-control input communication links (see Fig.2.1) while each player aims to minimize its selfish LQR objective (4.37). A Nash Equilibrium of this game results in a communication network with cost bounded by s. Thus, Nash strategies $(\mathbf{u}_1^*(t), \mathbf{u}_2^*(t), ..., \mathbf{u}_r^*(t))$ satisfy for $\forall i \in \{1, ..., r\}$

$$\begin{aligned} & J_i(\mathbf{u}_i^*(t), \mathbf{u}_{-i}^*(t)) \leq J_i(\mathbf{u}_i(t), \mathbf{u}_{-i}^*(t)) \quad , \forall \mathbf{u}_i(t) \\ & \text{s.t. } \text{card}_{\text{off}}(\mathbf{K}) \leq s \end{aligned} \tag{2.14}$$

Equivalently, since linear static feedback (2.11) is employed, the strategy of player i is given by the submatrix $\mathbf{K}^i$ in (2.11), and Nash strategies are expressed as $(\mathbf{K}^{1^*}, \mathbf{K}^{2^*}, ..., \mathbf{K}^{r^*})$, which satisfy for each $i \in \{1, ..., r\}$

$$\begin{aligned} & J_i(\mathbf{K}^{i^*}, \mathbf{K}^{-i^*}) \leq J_i(\mathbf{K}^i, \mathbf{K}^{-i^*}) \ , \forall \mathbf{K}^i \\ & \text{s.t. } \text{card}_{\text{off}}(\mathbf{K}) \leq s \end{aligned} \tag{2.15}$$

where the tuple $\mathbf{K}^{-i} := (\mathbf{K}^1, ..., \mathbf{K}^{i-1}, \mathbf{K}^{i+1}, ..., \mathbf{K}^r)$ represents the strategies of all other players except i. In this case, as s increases, the optimization (2.15) retains the most critical state-control input links needed for the noncooperative optimization, characterized by the set of selfish objectives (4.37).

The proposed game is described in *Algorithm 2.2*. It is inspired by the iterative gradient descent methods in [36,63], where each player takes a small step towards minimizing its own objective while other players' strategies are fixed. In each step associated with player i, we use the GraSP algorithm [59] to update the strategic variable $\mathbf{K}^i$ while maintaining the overall sparsity constraint. Thus, in the submatrix $\mathbf{K}^i$ in (2.11), the elements representing local feedback, i.e., those in the blocks $\mathbf{K}_{jj} \in \mathbb{R}^{p_j \times m_j}$ in (2.3) for $j \in \mathcal{S}_i$, are free variables while the off-diagonal blocks in (2.3) are subject to the sparsity constraint. The computation of the gradient of J_i

w.r.t. player i's strategy $\mathbf{K}^i$ is similar to that in (2.7–2.8).

$$\nabla_{\mathbf{K}^i} J_i(\mathbf{K}^i, \mathbf{K}^{-i}) = 2(\mathbf{R}_i \mathbf{K}^i - \mathbf{D}^{\mathrm{T}} \bar{\mathbf{P}}_i)\bar{\mathbf{L}}_i \tag{2.16}$$

where $\bar{\mathbf{P}}_i$ and $\bar{\mathbf{L}}_i$ are the unique solutions of the following Lyapunov equations

$$\begin{aligned} (\bar{\mathbf{A}}_i - \mathbf{B}_i \mathbf{K}^i)^{\mathrm{T}} \bar{\mathbf{P}}_i + \bar{\mathbf{P}}_i (\bar{\mathbf{A}}_i - \mathbf{B}_i \mathbf{K}^i) + \mathbf{Q}_i + {\mathbf{K}^i}^{\mathrm{T}} \mathbf{R}_i \mathbf{K}^i = 0 \\ (\bar{\mathbf{A}}_i - \mathbf{B}_i \mathbf{K}^i)\bar{\mathbf{L}}_i + \bar{\mathbf{L}}_i (\bar{\mathbf{A}}_i - \mathbf{B}_i \mathbf{K}^i)^{\mathrm{T}} + \mathbf{D}\mathbf{D}^{\mathrm{T}} = 0 \end{aligned} \tag{2.17}$$

and $\bar{\mathbf{A}}_i = \mathbf{A} - \sum_{j=1, j\neq i}^{r} \mathbf{B}_j \mathbf{K}^j$, $\mathbf{B}_j = \mathrm{col}(\mathbf{B}_{1j}, ..., \mathbf{B}_{rj})$.

Algorithm 2.2 is *distributed* in the sense that player i individually updates its strategic variable $\mathbf{K}^i$. Each player has prior knowledge of the underlying physical system (i.e., the matrices $\mathbf{A}, \mathbf{B}, \mathbf{D}$ in (2.1)) and broadcasts its strategic move (the submatrix $\mathbf{K}^i$) after its strategy is

Algorithm 2.2 Noncooperative game under the global communication cost constraint s.

1 *Initialization*
$\mathbf{K} := \mathbf{K}_0$
Initialize the link constraint s_i for each player i , $i = 1, ..., r$, $\sum_{i=1}^{r} s_i = s$.
2 *Iteration*
while stopping criterion are not met **do**
  $\mathbf{K}^{\mathrm{prev}} := \mathbf{K}$
  for $i = 1 \dots r$ **do**
    (1) if $i > 1$, $s_i = s - \mathrm{card}_{\mathrm{off}}(\mathbf{K}^{-i})$.
    (2) Compute gradient of J_i in (4.37) w.r.t $\mathbf{K}^i$:
      $\mathbf{g}_i = \nabla_{\mathbf{K}^i} J_i(\mathbf{K}^i, \mathbf{K}^{-i})$
    (3) Identify up to $2s_i$ off-diagonal block directions: $\mathcal{Z} = \mathrm{supp}([\mathbf{g}_i^{\mathrm{off}}]_{2s_i})$
    (4) Merge support: $\mathcal{T} = \mathcal{Z} \cup \mathrm{supp}(\mathbf{K}^i)$.
    (5) Descend using the Newton step of J_i restricted to $\mathcal{T}$: $\mathbf{K}^i := \mathbf{K}^i + \lambda \Delta_{\mathrm{nwt}}(\mathbf{K}^i, \mathcal{T})$.
    (6) Prune among the off-diagonal block elements: $\mathbf{K}^i := \mathbf{K}^{i,\mathrm{diag}} + [\mathbf{K}^{i,\mathrm{off}}]_{s_i}$
  end for
  Construct global feedback matrix: $\mathbf{K} = [\mathbf{K}^1; ...; \mathbf{K}^r]$.
  Stopping criterion:
  $||\mathbf{K} - \mathbf{K}^{\mathrm{prev}}||_2 < \epsilon_{\mathrm{abs}} \sqrt{qm} + \epsilon_{\mathrm{rel}} ||\mathbf{K}^{\mathrm{prev}}||_2$.
end while
3 *Polishing*
$\mathcal{I}_i = \mathrm{supp}(\mathbf{K}^i)$, for $i = 1, ..., r$.
while not $\nabla_{\mathbf{K}^i} J_i|_{\mathcal{I}_i} < \epsilon_2 \ \forall i = 1, ..., r$ **do**
  for $i = 1, ..., r$ **do**
    Descend using the Newton step of J_i restricted to $\mathcal{I}_i$: $\mathbf{K}^i := \mathbf{K}^i + \lambda \Delta_{\mathrm{nwt}}(\mathbf{K}^i, \mathcal{I}_i)$.
  end for
end while

updated at the completion of Step 2(6) and each inner loop of Step 3.

Finally, note that the social optimization under the sparsity constraint (2.6) can also be implemented in a distributed fashion using a potential game [36], obtained from (2.15) by replacing the individual objectives in (2.15) with the common social objective (2.5) while the players' strategies are still defined as their control vectors. We refer to this game as *distributed social optimization*, and its equilibria are defined as

$$\begin{aligned} J(\mathbf{K}^{i*}, \mathbf{K}^{-i*}) \leq J(\mathbf{K}^{i}, \mathbf{K}^{-i*}) \ , \forall \mathbf{K}^{i} \\ \text{s.t. } \text{card}_{\text{off}}(\mathbf{K}) \leq s \end{aligned} \tag{2.18}$$

To compute (2.18), we use Algorithm 2, where the social objective $J()$ replaces $J_i()$ in Step 2(2) and in the Polishing Step 3.

Finally, we discuss the NEs of the proposed games. Since the LQR objective (2.5) is not convex in the feedback matrix $\mathbf{K}$ in general [72], the noncooperative game (2.14, 2.15) is not guaranteed to admit a pure-strategy NE [74]. On the other hand, the distributed social optimization (2.18) is an exact potential game, and, thus, a pure NE exists for this game [36]. Moreover, the optimal solution of the social optimization problem (2.6) constitutes a NE of (2.18) although the converse does not necessarily hold due to nonconvexity of the LQR objective. Finally, while CARE has a unique NE under the weakly-coupled system assumption [40], the cost-constrained games (2.14), (2.18) can have multiple NEs.

2.4 Network Cost Allocation

First, we review relevant results for cooperative NFGs with transferable utility [75] where the utility of a coalition is viewed as monetary value, which is distributed among the players. Suppose r players cooperatively form a network with the objective of maximizing their payoffs. Several approaches to fair payoff allocation have been proposed in the literature [65, 66, 76, 77]. We employ the Nash Bargaining Solution (NBS) due to its computational efficiency in NFGs [65, 66]. The NBS payoff allocation algorithm proceeds in *three steps* based on [66]:

(1) The players cooperate to construct a network that maximizes the *global social payoff* v_{soc};

(2) The *disagreement point* is computed as

$$\mathbf{v} = (v_1, v_2 ..., v_r) \tag{2.19}$$

where the *selfish payoff* v_i is the minimum payoff that the i^{th} player is willing to accept. For example, (2.19) can be computed as a NE of a noncooperative NFG where each player aims to

satisfy its own selfish objective;

(3) The overall payoff $v_{\rm soc}$ is split among the players, with the *allocated payoff* of player i given by [66]

$$\alpha_i = v_i + \frac{v_{\rm soc} - \sum_{k=1}^{r} v_k}{r}. \tag{2.20}$$

Note that bargaining is successful when the payoff of the social network is at least as large as the sum of the selfish payoffs, i.e.,

$$v_{\rm soc} \geq \sum_{i=1}^{r} v_i \tag{2.21}$$

or equivalently, each player's allocated payoff is at least as large as its selfish payoff:

$$\xi = \alpha_i - v_i \geq 0. \tag{2.22}$$

From (2.22), all players benefit equally by forming the social network.

Next, we describe the proposed *network cost allocation method for sparsity-constrained multi-agent dynamic systems*, which is summarized in *Algorithm 2.3*. Given the global communication cost constraint s, in steps 1–3, three different scenarios are analyzed, depending on

Algorithm 2.3 Network Cost Allocation under the Communication Cost Constraint s.

1 *Find social optimized objective* $J_{\rm soc}(s)$

$J_{\rm soc}(s)$ is the optimized objective of (2.6) or (2.18) under constraint s;

2 *Find decoupled optimized objectives at DNE.*

(1) The NE (2.15) for $s = 0$ is the *decoupled* NE (DNE);

(2) *Optimized selfish objective* at *DNE*:$J_i^{\rm D}, i = 1, ..., r$.

(3) *Total objective at DNE:* $\tilde{J}^{\rm D} := \sum_{i=1}^{r} J_i^{\rm D}$.

3 *Find coupled optimized objectives at CNE*

(1) The NE (2.15) under constraint s is *coupled* NE (CNE(s));

(2) *Optimized selfish objective at CNE*: $J_i^{\rm C}(s),\ i = 1, ..., r$

(3) *Total objective at* CNE(s): $\tilde{J}^{\rm C}(s) = \sum_{i=1}^{r} J_i^{\rm C}(s)$.

4 *Compute and allocate payoffs and network costs*

(1) The *social payoff*: $v_{\rm soc}(s) = \tilde{J}^{\rm D} - J_{\rm soc}(s)$

(2) The *selfish payoffs* (the *disagreement point*):
$v_i(s) = J_i^{\rm D} - J_i^{\rm C}(s),\ i = 1, ..., r$;

(3) The *allocated payoffs*:
$\alpha_i(s) = v_i(s) + \frac{v_{\rm soc}(s) - \sum_{k=1}^{r} v_k(s)}{r}$

(4) *Proportional allocation of the network cost*:
$C_i(s) = \alpha_i(s) / \sum_{i=1}^{r} \alpha_i(s),\ i = 1, ..., r.$

whether the agents employ: *(i)* non-local feedback, i.e., *communication bounded by cost* s; and *(ii) cooperation*. In Step 1, both *(i)* and *(ii)* are assumed, resulting in the social optimization, implemented using Algorithm 2.1 or 2.2 under the constraint s. In contrast, in Step 2 neither *(i)* nor *(ii)* are employed, and thus the agents play the *decoupled* noncooperative game, i.e., (2.14) with $s = 0$, which results in a *decentralized* system using the Polishing Step 3 of Algorithm 2.2 (see [67]). Finally, in Step 3, only *(i)* is employed, resulting in the *coupled* game, i.e., (2.14) with the global constraint s, implemented using Algorithm 2.2. Note that Steps 1 and 3 optimize systems with (at most) s communication links (see Fig.2.1), while Step 2 restricts communication to local feedback links.

In Steps 4(1) and 4(2), the *payoffs* $v_{\text{soc}(s)}$ and $v_i(s)$ represent the improvement in control performance, or *objective reduction*, provided by communication bounded by cost s with (in Step 1) and without (in Step 3) cooperation, respectively, relative to the decentralized implementation (Step 2). It is reasonable to model the minimum payoff an agent i expects from communication by its selfish objective reduction, or *selfish payoff*, $v_i(s)$, thus forming the *disagreement point*. The *allocated payoffs* $\alpha_i(s)$ are computed in Step 4(3). Note that the NBS algorithm interprets the *payoffs* $\alpha_i(s)$ and $v_i(s)$ as *values* (e.g., monetary payoffs) that agent i derives from communication with and without cooperation, respectively (see (2.19, 2.20)). If cooperation does not degrade the minimum acceptable value $v_i(s)$, i.e., if (2.21, 2.22) hold, then bargaining is successful, and the agents agree to use the links formed by the social optimization found in Step 1, which provides them with the social payoff $v_{\text{soc}}(s)$ and the individual payoffs $\alpha_i(s),\ i = 1...r$. The latter condition is satisfied for a practical scenario where, in (2.5) and (4.37), the matrix $\mathbf{R}$ is block-diagonal with diagonal blocks given by $\mathbf{R}_i$, $i = 1, ..., r$, and

$$\sum_{i=1}^{r} \mathbf{Q}_i = \mathbf{Q} \tag{2.23}$$

In this case, $\sum_{i=1}^{r} J_i(\mathbf{K}) = J(\mathbf{K})$[1], $\forall \mathbf{K}$, and thus a NE of the noncooperative game (2.13) is a feasible solution of the social optimization (2.6) for the same value of constraint s, resulting in $J_{\text{soc}}(s) \leq \tilde{J}^{\text{C}}(s)$ (see Step 1 and 3(3)), which is equivalent to (2.21).

We assume that the grand coalition forms in the proposed cooperative game among the agents. This assumption justifies Step 1 in Alg. 3, where all players cooperate. A sufficient condition for this assumption is efficiency of the grand coalition [78]. We show in appendix A.1 that a practical coalition-level identity similar to (2.23) guarantees efficiency of the grand coalition.

If cooperation is successful, the feedback data required for implementing the *social control*

[1] $J_i(\mathbf{K})$ is equivalent to (4.37). Since we use full state feedback, and the control input $\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$ is equivalent to (2.11), the objective $J_i(\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_r)$ is a function of $\mathbf{K}$ and can be expressed as $J_i(\mathbf{K})$.

strategy (as specified by the optimal feedback matrix in (2.6) or (2.18)) will be delivered by the communication network specific to the given medium and application, where *the constraint s is chosen based on the desired performance/cost tradeoff*. In the final Step 4(4), we compute each agent's share of the cost of this network proportionally to the allocated payoffs $\alpha_i(s)$. Note that if $\xi(s)$ in (2.22) is small relative to the average of the selfish payoffs $v_i(s)$ (or $J_{\rm soc}(s) \approx \tilde{J}^{\rm C}(s)$), the benefit of cooperation *(ii)* is small relative to that of communication *(i)*, and the agents' costs $C_i(s)$ can vary significantly due to the agents' diverse needs for feedback and cooperation as is illustrated in Section 3.5 and in [67]. On the other hand, when $\xi(s)$ is relatively large, i.e., $\alpha_i(s) \approx \xi(s)$ for all i, the payoffs equalize (2.22), resulting in equally split cost of the communication network among the agents.

2.5 Example: Sparsity-Constrained Wide-Area Control of Power Systems

We validate our algorithms using the example of wide-area control (WAC) of large-scale power system networks. In recent literature such as [7, 68], WAC has been shown to be very useful for suppressing low-frequency oscillations following small-signal disturbances in large power grids, at the cost of communicating real-time data from sensors at one operating region to controllers at others. The sensors are referred to as Phasor Measurement Units (PMUs), all of which are synchronized to each other via GPS. The envisioned architecture for PMU data exchange in the US power grid, also referred to as the North American Synchrophasor Initiative Network, or NASPInet [4], involves PMUs located at substations of different utility companies sending their measurements to controllers at remote generators through a wide-area communication network such as the Internet. To reduce the cost of this communication, sparsity promotion for WAC was studied in [6, 7]. However, these designs employed social centralized implementation. In contrast, our approach is to design distributed controllers for WAC that are efficient with respect to communication costs. Moreover, we address the question posed in [4] on financing the communication network among the power companies. While the envisioned NASPInet has many functions beyond WAC, fair allocation of network costs associated with delivery of feedback data is necessary to develop its overall pricing scheme.

We model a power transmission system using (2.1,2.9–2.11) and as in Fig.2.1, where a node represents a *generator* while an agent (player) represents the operating territory (*area*) owned by a utility company. Following [7, 67], we express WAC as an LQR problem for minimizing the closed-loop energy of the system states, which equivalently translates into reducing the oscillations in their dynamic response. The LQR objective in (2.5) aims to damp the power oscillations captured by the small-signal changes around the nominal values while spending a

reasonable amount of control effort [7], where the matrices $\mathbf{Q}$ and $\mathbf{R}$ in (2.5) are chosen to reduce the energies of the state and control vectors, respectively. Therefore, in the rest of this chapter we will refer to the objective (e.g., in (2.5), (4.37), Algorithm 2.3, etc.) as *energy* and to the payoffs $v_{\text{soc}}(s)$ and $v_i(s)$ in Algorithm 2.3 (Step 4(1) and 4(2)) as *energy savings*.

Typically a 3$^{\text{rd}}$-order model of synchronous generators, including two swing states and one excitation state (for details, please see [68]) suffices for solving most WAC problems since the goal is to influence the electro-mechanical dynamics of these generators. Note that the 3$^{\text{rd}}$-order model indicates $m_j = 3$ and $p_j = 1, \forall j = 1, ..., n$ in the general state-space model (2.1). We consider a power system with n synchronous generators, divided into r non-overlapping areas, and define the states of the j^{th} generator accordingly as

$$\mathcal{X}_j^{\mathrm{T}}(t) = (\Delta\delta_j(t), \Delta\omega_j(t), \Delta E_j(t)), \tag{2.24}$$

where the elements represent small-signal changes in phase angle, frequency, and excitation voltage, respectively. If higher-order models are considered, then the last term can be simply replaced by a vector $\mathbf{x}_j^-$ that collects all states except for the phase and the frequency. We assume full state availability, which can be achieved by placing PMUs at every generator bus, or by running a prior state estimation loop. We assume the matrix $\mathbf{D}$ in (2.1) to be an indicator matrix with all elements zero except for the one corresponding to the acceleration equation of the generator at which the fault $w(t)$ happens.

Next, we describe the choice of matrices in the social (2.5) and the selfish (4.37) energy optimization. We set $\mathbf{R}$ in (2.5) and $\mathbf{R}_i$ (4.37) as the identity matrix as to achieve the same weight for the energy of every control input. The matrix $\mathbf{Q}$ in (2.5) is chosen so that all generators arrive at a consensus in their small-signal changes in phase angles and frequencies, as dictated by the physical topology of the network [7, 67]. Considering the small-signal-model in Kron-reduced form [68], for the 3$^{\text{rd}}$ order model, $\mathbf{Q}$ in (2.5) is determined from (2.25)

$$\begin{aligned} E_{\text{states}} &= \begin{bmatrix} \Delta\boldsymbol{\delta} \\ \Delta\boldsymbol{\omega} \\ \Delta\mathbf{E} \end{bmatrix}^T \underbrace{\begin{bmatrix} \bar{\mathcal{L}} & & \\ & \bar{\mathcal{L}} & \\ & & \mathbf{I} \end{bmatrix}}_{\mathbf{Q}'} \begin{bmatrix} \Delta\boldsymbol{\delta} \\ \Delta\boldsymbol{\omega} \\ \Delta\mathbf{E} \end{bmatrix} \\ &= \mathbf{x}^T(\mathcal{P}^T\mathbf{Q}'\mathcal{P})\mathbf{x} = \mathbf{x}^T\mathbf{Q}\mathbf{x} \\ &= \sum_{k=1}^{n}\sum_{j=k+1}^{n}[(\Delta\delta_j - \Delta\delta_k)^2 + (\Delta\omega_j - \Delta\omega_k)^2] + \sum_{j=1}^{n}\Delta E_j^2, \end{aligned} \tag{2.25}$$

where $\mathcal{P}$ is a permutation matrix that rearranges the state vector $\mathbf{x}$ in (2.1) by stacking all the angles first, then all the frequencies, then the excitation voltages. For the general case (2.1), the

$\Delta\mathbf{E}^{\mathrm{T}}\Delta\mathbf{E}$ term in (2.25) is replaced by $(\mathbf{x}^{-})^{\mathrm{T}}\mathbf{x}^{-}$, where the latter is obtained by stacking the terms $\mathbf{x}_j^{-}$. The matrix $\bar{\mathcal{L}} = n\mathbf{I} - \mathbf{1}_n \cdot \mathbf{1}_n^T$ (appendix A.2, where $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ is the column vector of all ones, and $\mathbf{I}$ is the identity matrix.

To quantify the i^{th} player's selfish objective (4.37), we define two quantities, namely, *intra-area energy* and *inter-area energy*, from the perspective of the i^{th} player as follows:

$$E_i^{\text{intra}}(\mathbf{x}) := \sum_{k\in s_i} \sum_{\substack{j\in s_i \\ j>k}} \left[(\Delta\delta_k - \Delta\delta_j)^2 + (\Delta\omega_k - \Delta\omega_j)^2\right] + \sum_{k\in s_i} \Delta E_k^2 \tag{2.26}$$

$$E_i^{\text{inter}}(\mathbf{x}) := \frac{1}{2} \sum_{k\in s_i} \sum_{\substack{j=1,\ldots,n, \\ j\notin s_i}} \left[(\Delta\delta_k - \Delta\delta_j)^2 + (\Delta\omega_k - \Delta\omega_j)^2\right] \tag{2.27}$$

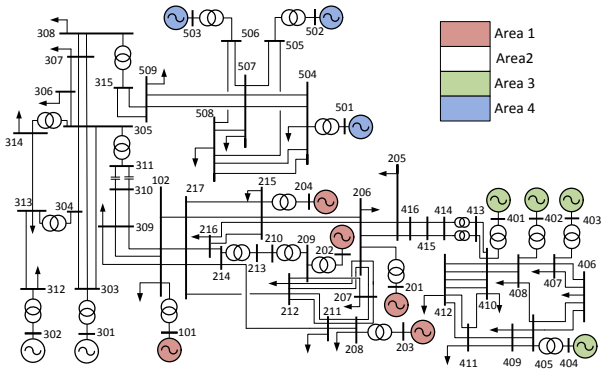
It can be seen that the intra-area energy of area i is designed for the consensus in the phase angle and frequency states of the generators in area i. The inter-area energy is modeled by collecting the power transfer terms associated with a generator in area i and a generator in another area, and attributing $1/2$ of this energy to area i. The total *state energy* associated with area $i \in \{1, ..., r\}$ is

$$\mathbf{x}^T\mathbf{Q}_i\mathbf{x} = E_i^{\text{intra}}(\mathbf{x}) + E_i^{\text{inter}}(\mathbf{x}), \tag{2.28}$$

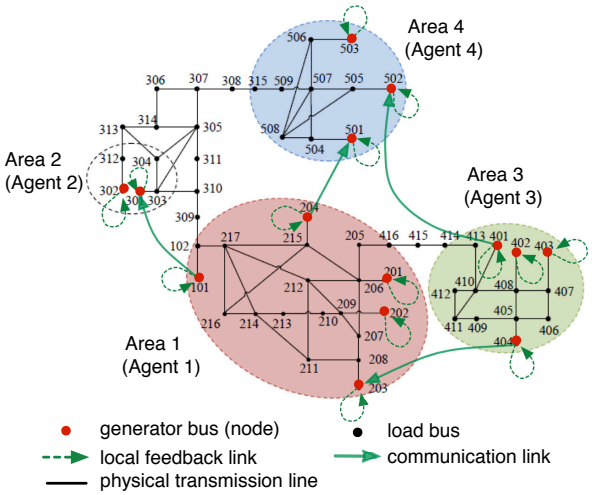
where $\mathbf{Q}_i \geq 0$, since (2.28) is the quadratic form of the states. Detailed derivations of $\mathbf{Q}$ and $\mathbf{Q}_i$ can be found in [79]. It is easy to show that the resulting LQR objectives (2.5) and (4.37) satisfy (2.23), thus assuring successful cooperation among the power companies. We illustrate this fact in the next section by numerical simulation of a 50-bus power system model.

2.6 Numerical Results and Performance Analysis for the Australian Power System Model

We validate our results using a 50-bus Australian power system model shown in Fig.2.2(a), which consists of 14 synchronous generators, divided into 4 coherent areas, and is a reasonably accurate representation of the power grid in south-eastern Australia [1]. The area distribution is shown in different colors in Fig.2.2(b), with the red dots denoting generator buses. Generators 1 to 5 belong to area 1, 6 and 7 – to area 2, 8 to 11 – to area 3, and 12 to 14 – to area 4. Each generator is modeled by up to 17 states, namely the generator phase angle, the generator speed, direct and quadrature axis components of the internal voltage of the generator, direct and quadrature axis components of the internal flux of the generator, the field excitation voltage, three states contributed by the automatic voltage regulator (AVR), three states contributed by the power system stabilizer (PSS), one state contributed by the stabilizing transformer, and finally three states contributed by supporting induction generators. The small-signal linearized



(a) Line diagram of the Australian 50-bus system.



(b) Simplified Australian power system with feedback links.

Figure 2.2: A simplified 50-bus representation of the southeast Australian power system [1].

model is extracted using the MATLAB PST toolbox [80]. However, since we are primarily interested in the electro-mechanical states, we perform an initial round of model reduction using singular perturbation and thereby eliminate the non-electromechanical states with very low participation in the swing dynamics. The exact expressions of the model matrices are not included in this chapter for brevity and are provided in [79]. The small-signal model is excited by impulsive disturbance inputs entering through the acceleration equation of the generators, and the proposed LQR controller is actuated through the field excitation voltages, using state feedback from all generators. Fig.2.2(b) also illustrates the communication and local feedback links between the generators.

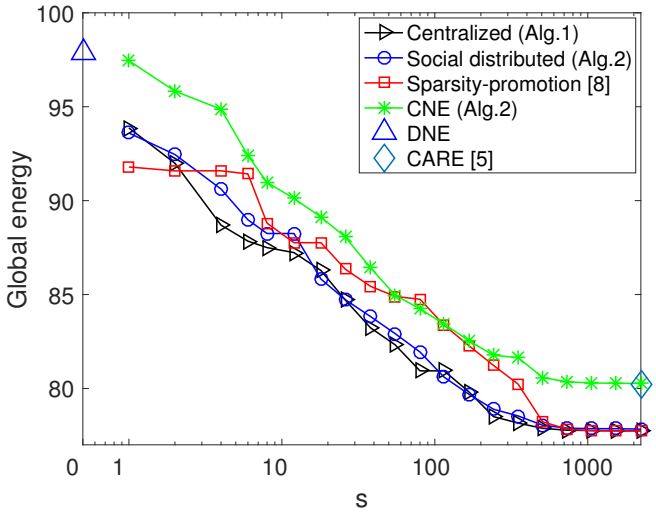


Figure 2.3: Global energies of social optimization and noncooperative games vs. communication cost constraint.

2.6.1 Global Energies

Fig.2.3 shows the global closed-loop energies for the 50-bus system with respect to the communication cost constraints for the various centralized and distributed algorithms discussed in the previous sections. Global energies of the centralized optimization (2.6) using Algorithm 2.1, the coupled noncooperative game (CNE) in Step 3 of Algorithm 2.3, the social distributed optimization (2.18) using Algorithm 2, and of the decoupled game (DNE) in Step 2 of Algorithm 2.3 are included. We also show the global energy of the iterative dense-feedback method that solves CARE [40] (note that our system satisfies the weakly coupled condition in [40]) and of the centralized sparsity-promoting ADMM method [7], modified to satisfy the sparsity constraint. Since the energy of the method in [7] is nondecreasing with the sparsity parameter γ, a bisection search on γ yields the smallest value of γ for which the off-diagonal cardinality of the feedback matrix produced by the ADMM algorithm satisfies the constraint. The choice of the l_1-metric weights for this ADMM implementation is described in [67].

The figure shows that the global closed-loop energies resulting from the social optimization using Algorithm 2.1 or 2.2 are smaller than those of the noncooperative game (CNE using Algorithm 2.2) for any value of s. This testifies to the fact that the dynamic performance of the grid improves when companies cooperate with each other to jointly design the controller, which constitutes a sufficient condition for the cooperation to form according to (2.21). We also observe that the global CNE objective (Algorithm 2.3, Step 3(3)) tends to those of the DNE (Algorithm 2.3, Step 2(3)) and CARE as s approaches 0 and its largest value $s = 2223$,

respectively. The former corresponds to the decentralized case [67] while the latter requires dense communication [40].

Moreover, the energy J_{soc} of the distributed social optimization (using Algorithm 2) closely approximates that of the centralized Algorithm 1 for most of the s-range. We also observe that the closed-loop energy of social optimization using Algorithms 2.1 or 2.2 is smaller than that using the sparsity-promoting algorithm in a moderately sparse region, thus providing better reduction of both intra- and inter-area oscillations when given an exact communication cost constraint.

While the global energies of all sparsity-constrained methods are theoretically nonincreasing with s, the social algorithms might occasionally produce a larger energy as s increases. This happens when an algorithm converges to a local minimum since the optimization objective (2.5) is locally, but not necessarily globally, convex in $\mathbf{K}$ [72]. If the algorithm results in $J_{\text{soc}}(s_2) > J_{\text{soc}}(s_1)$ for $s_2 > s_1$, we choose a suboptimal solution $J_{\text{soc}}(s_2) \triangleq J_{\text{soc}}(s_1)$ (see Algorithm 2.3, Step 1), which produces a nonincreasing $J_{\text{soc}}(s)$. Finally, note that the global objectives (energies) of all algorithms saturate to the same asymptotic value when s exceeds 740, implying that the communication cost can be reduced by a factor of 3 relative to the cost of the dense LQR network without compromising the control performance. Thus, in the next figure, we show results only for small and moderate values of s, where the energies vary significantly.

2.6.2 Selfish Energies and Cost Allocation

In Fig.2.4, we show performance of *noncooperative* games, as well as payoffs and costs in Algorithm 2.3. Fig.2.4(a) shows the individual energy objectives of the four areas at CNE (Step 3(2)). In Fig.2.4(b), the selfish energy savings $v_i(s)$ (Step 4(2)) for each company, their sum, the social energy savings $v_{\text{soc}}(s)$ (Step 4(1)), and the payoff increase $\xi(s)$ (2.22) are illustrated. We observe that bargaining is successful (see (2.21)), and there is modest payoff increase due to cooperation.

Note that while the overall global CNE objective $\tilde{J}^{\text{C}}(s)$ (Step 3(3)) is theoretically nonincreasing with s, this is not necessarily true for individual selfish energies $J_i^{\text{C}}(s)$ in Step 3(2), which means that increasing the overall communication budget might degrade some areas' energies when they act noncooperatively. This phenomenon can result in decreasing and possibly negative payoffs $v_i(s)$ and $\alpha_i(s)$ over some regions of the constraint s, which might dissatisfy the affected players and would require a revision of the proposed cost allocation method in Step 4(4) of Algorithm 2.3. See, e.g., slightly increasing selfish objective $J_3^{\text{C}}(s)$ and negative selfish payoff $v_3(s)$ of area 3 in Fig.2.4 (a) and (b), respectively, for small values of s. Nevertheless, the allocated payoffs $\alpha_i(s)$ in Algorithm 2.3 (Step 4(3)) are nonnegative for all companies for the Australian power system scenario due to sufficient cooperation gain $\xi(s)$. A slightly modified

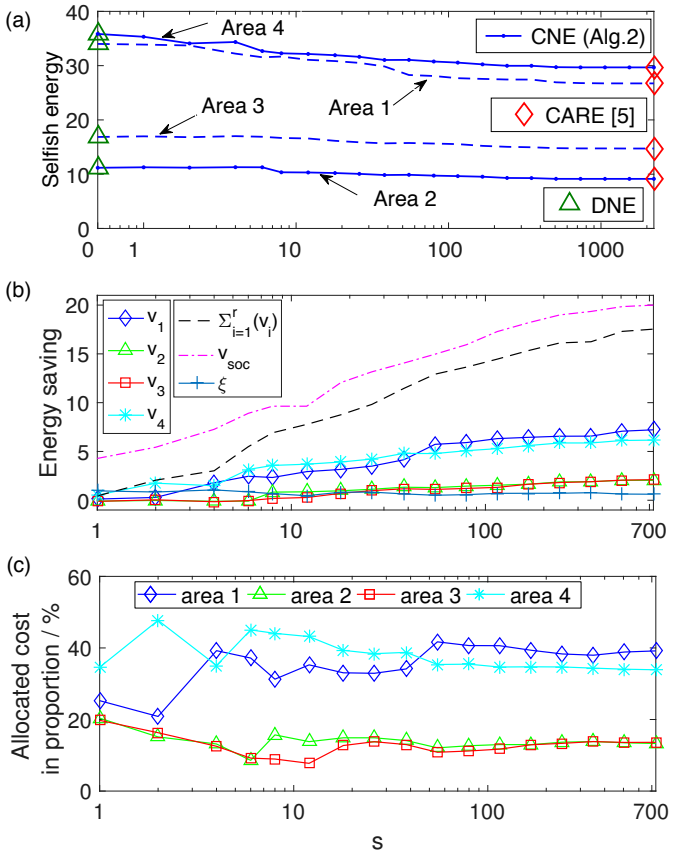


Figure 2.4: Selfish energies (objectives), payoffs and cost allocation in Algorithm 2.3 vs. communication cost constraint s. (a) Selfish energies of noncooperative games for areas 1–4. (b) The energy savings (payoffs) $v_i(s)$ and $v_{\rm soc}(s)$ and payoff increase $\xi(s)$. (c) Proportional cost allocation.

payoff computation method that guarantees nondecreasing payoffs is presented in Appendix A.1.

Fig.2.4 demonstrates significant disparity in the selfish energies $J_i^{\rm C}(s)$, the $v_i(s)$ values, and the allocated proportional costs $C_i(s)$ among the areas, which is due to the grid topology. For example, large selfish energy and allocated cost of area 1 can be explained by its large number of generators. However, area size is not the only indicator, e.g., area 4 has fewer generators than area 3, but much larger selfish energy $J_4^{\rm C}(s)$, energy savings $v_4(s)$, and allocated cost $\alpha_4(s)$, which even exceed those of area 1 for smaller values of s. In summary, areas 1 and 4 pay a much greater share of the overall network cost than areas 2 and 3 due to the former areas' greater needs for feedback and cooperation, which is consistent with relatively steep decline of their selfish energies with s in Fig.2.4(a) and with the fact that in the social optimization, links among the generators in these areas are the first to be added as s increases. Thus, these links

are the most valuable for achieving energy reduction when using WAC [7].

2.6.3 Algorithm Convergence and Implementation Issues

First, we focus on the numerical properties of Algorithms 1 and 2. Since the LQR objective does not satisfy the Stable Restricted Hessian condition [59], convergence of these algorithms is not assured in general. However, if Step 2 of Algorithm 1 converges and yields a stabilizing feedback matrix $\mathbf{K}$, then Step 3 will also converge due to the local convexity of $J(\mathbf{K})$. At convergence, Step 3 produces a feedback matrix $\hat{\mathbf{K}}$ that satisfies the basic feasibility property $\nabla_{\mathbf{K}} J(\mathbf{K})|_{\text{supp}(\hat{\mathbf{K}})}(\mathbf{K} = \hat{\mathbf{K}}) = 0$, which is a weak necessary condition for the optimality of problem (2.6) [60]. Similar arguments demonstrate that Algorithm 2 produces a feedback matrix that satisfies the basic feasibility property of each individual minimization in the CNE problem (2.15) (or (2.18) for social optimization). Note that the decoupled game in Step 2 of Algorithm 2.3 represents an unconstrained optimization with respect to each player's strategy, which is given by its local feedback matrix [67]. This game is implemented using the polishing Step 3 of Algorithm 2.2. When this implementation converges, the resulting equilibrium point is a local NE [63] due to local convexity of the LQR objective [72], i.e., the strategy of each player is a local optimum given other players' strategies. From these observations, we conclude that convergence properties of proposed algorithms resemble those of the ADMM-based methods [7]. In both cases, while theoretical guarantees are not always feasible, extensive numerical experience demonstrates that the algorithms converge and provide desirable minimization solutions over a range of sparsity parameters. We found that the proposed sparsity-constrained algorithms also converge and exhibit similar performance and complexity trends to those shown in this chapter for the New England power system model used in [7].

Next, we describe the algorithm implementation details for the results shown in Fig.2.3–2.4. We found that the proposed algorithms can converge to different stabilizing feedback matrices given different initial settings. Moreover, for $s > 0$, the energies of these solutions can differ significantly. However, we found that the energies of different equilibria were very similar to each other for the decoupled game implemented using Algorithm 2.3 (Step 2). We have employed $\epsilon_{\text{abs}} = \epsilon_{\text{rel}} = 10^{-4}$ and $\epsilon_2 = 10^{-4}$ and 10^{-3} in Algorithms 1 and 2, respectively, to achieve comparable performance for distributed and centralized social optimizations. For $s = 1$, both algorithms were initialized with a stabilizing matrix $\mathbf{K}_0$ obtained by preserving the block-diagonal entries of the dense LQR feedback matrix that optimizes (2.5) and setting other entries to zero. For larger s, $\mathbf{K}_0$ was chosen as the optimized feedback matrix obtained in a previous computation for a smaller value of s. We found that this initialization produced the lowest energies over the entire s-range. If a stable block-diagonal matrix cannot be found, $\mathbf{K}_0$ can be obtained using the sparsity-promotion algorithm [7] with the largest γ that produces a

stabilizing feedback matrix (Similarly, stabilizing feedback might not exist for small values of the constraint s). Moreover, Algorithm 2 had the best performance when the initial link settings s_i were chosen proportionally to the number of nodes n_i in (2.9).

Finally, we found that the computational load of the polishing step dominates the overall runtime for both algorithms, and the Newton step using the CG method (see Table 2.1, last entry) is the most computation-intensive operation, which has polynomial complexity in s and the number of states [6]. In our experiments, all algorithms in Fig.2.3 converged in less than 10^3 seconds for any value of s although this did not include the bisection search time for the modified ADMM method [7], which is very computation-intensive. Moreover, the distributed social implementation using Algorithm 2 converged much faster than the centralized method (Algorithm 1)[2].

2.7 Conclusion

LQR optimization under the communication-cost constraint was investigated for multi-agent dynamic systems with linear static state feedback. First, a communication-cost-constrained centralized social optimization algorithm was developed. Second, distributed game-theoretic algorithms were investigated for both selfish and social optimization under the sparsity constraint. Finally, cooperative NFG theory was employed to allocate the costs of the communication infrastructure in a multi-agent dynamic system. Using a 50-bus power system model divided into 4 areas, we demonstrated convergence of proposed algorithms and desirable performance and complexity features of distributed methods over the range of the sparsity constraint, thus providing a trade-off between the communication cost and the control performance. Furthermore, we discussed the relationship between the proportionally allocated costs and power companies' needs for feedback and cooperation, and showed that the proposed cost allocation algorithm is rooted in the physical topology of the power grid. Our current research focuses on applying proposed algorithms to systems with uncertainty and communication delays. Moreover, we plan to extend the concepts of this chapter to diverse multi-agent dynamic system applications, robust optimization objectives, output feedback, and different communication network architectures.

[2]The experiments are run using MATLAB on a MacBook Pro with Yosemite operating system, 2.6 GHz Intel core i5 processor, and 8 GB 1600 MHz DDR3 memory.

Chapter 3

Sparsity-Constrained Mixed H_2/H_∞ Control

We propose an algorithm for designing sparsity-constrained controllers for linear time-invariant systems with model uncertainties using a mixed H_2/H_∞ objective. We first improve a previously proposed descent algorithm for mixed H_2/H_∞ control using a modified Zoutendijk's method. Thereafter, we impose a sparsity constraint on this design by combining it with a greedy gradient support pursuit (GraSP) method. The proposed algorithm guarantees a predetermined level of sparsity while maintaining acceptable H_2 performance as well as H_∞ robustness to uncertainties in the open-loop model. The effectiveness of the proposed method is illustrated through simulations.

3.1 Introduction

Recent papers such as [2, 8, 81, 82] have proposed various sparse LQR optimal control designs for reducing communication costs for controlling large multi-agent LTI networks. However, this research ignores model uncertainties, which are bound to arise in all practical large-scale systems since the network operating conditions and the resulting network topology might change rapidly over time. Even if the system is static, the designer might not know all model parameters or have mismatched parameter values due to idealistic design assumptions in the underlying communication network. This problem is addressed in [23], where a sparse H_∞ controller is derived for a class of system with sparse input matrix by exploiting the symmetric structure of the state matrix. However, to provide robustness while optimizing the closed-loop performance, is it desirable to use the mixed H_2/H_∞ control objective [83, 84]. In [22], this objective was combined with sparsity promotion by minimizing the weighted sum of H_2-norm bound, H_∞-norm bound, and the ℓ_1-norm for the feedback gain matrix. However, the latter method can

only promote sparsity (as in [8,81]) and robustness, but does not guarantee that a specific H_∞ bound or the number of feedback links is achieved.

In this chapter, we develop a controller that minimizes the H_2 norm while satisfying a predetermined sparsity and the H_∞ bound constraints. To achieve this goal, we first employ a modified Zoutendijk's method in [85] to extend the gradient descent approach for solving the mixed H_2/H_∞ control problem. We also employ the LMI interpretation of the H_∞-norm constraint from [86] to avoid computing the gradient of the constraint function. Since the above approaches are designed for dense feedback, we next impose a sparsity constraint on the modified algorithm using the GraSP method [2, 59].

The *main contributions* of this chapter are:

- Derivation of an improved descent algorithm for the mixed H_2/H_∞ control.
- Development of a sparsity-constrained mixed H_2/H_∞ control algorithm.

This chapter is organized as follows. Section 3.2 reviews mixed H_2/H_∞ control. In Section 3.3, the gradient-descent method for mixed H_2/H_∞ control is extended using a modified Zoutendijk's method of feasible direction. In Section 3.4, a GraSP-based algorithm for the sparsity-constrained mixed H_2/H_∞ control is developed. In Section 3.5, we illustrate the effectiveness of the proposed algorithm using a network of unstable nodes. Finally, some concluding remarks are made in Section 3.6.

Throughout this chapter, matrices are denoted with boldface capital letters. If $\mathbf{M}$ is a symmetric matrix, the upper block matrices are sometimes denoted by $*$ to save space. The notation used in this chapter is summarized in Table 3.1.

3.2 Mixed H_2/H_∞ control

Consider the following class of linear time-invariant uncertain systems,

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \underbrace{(\mathbf{A}+\Delta\mathbf{A})}_{\hat{\mathbf{A}}}\mathbf{x}(t) + \underbrace{(\mathbf{B}+\Delta\mathbf{B})}_{\hat{\mathbf{B}}}\mathbf{u}(t) + \mathbf{B}_2\mathbf{w}_2(t)\\ \mathbf{z}_2(t) &= \mathbf{C}_2\mathbf{x}(t) + \mathbf{D}_2\mathbf{u}(t) + \mathbf{D}_{22}\mathbf{w}_2(t)\end{aligned} \tag{3.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n\times 1}$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^{m\times 1}$ is the control input vector, $\mathbf{w}_2(t) \in \mathbb{R}^{m_2\times 1}$ is the exogenous input, and $\mathbf{z}_2(t) \in \mathbb{R}^{p_2\times 1}$ is the performance output. Matrices $\mathbf{C}_2$ and $\mathbf{D}_2$ have the form $\mathbf{C}_2 = \begin{bmatrix}\mathbf{C}_2^1\\ \mathbf{0}\end{bmatrix}$, $\mathbf{D}_2 = \begin{bmatrix}\mathbf{0}\\ \mathbf{D}_2^2\end{bmatrix}$, and $\mathbf{C}_2^T\mathbf{D}_2 = 0$. We define

$$\mathbf{Q} = (\mathbf{C}_2^1)^T\mathbf{C}_2^1 \succeq 0,\ \mathbf{R} = (\mathbf{D}_2^2)^T\mathbf{D}_2^2 \succ 0. \tag{3.2}$$

Table 3.1: Notation used in Chapter 3

Term	Definition
$\mathbf{M}\succ 0(\succeq 0)$	Matrix $\mathbf{M}$ is positive definite (semidefinite)
$\mathbf{M}\prec 0(\preceq 0)$	$\mathbf{M}$ is negative definite (semidefinite)
$\mathrm{sym}(\mathbf{M})$	$\mathbf{M}+\mathbf{M}^T$
$\sigma_{\max}(\mathbf{M})$	maximum singular value of $\mathbf{M}$
$\|\|\mathbf{K}\|\|_F$	Frobenius norm of the matrix $\mathbf{K}$, defined by $\sqrt{\mathrm{trace}(\mathbf{K}^{\mathrm{T}}\mathbf{K})}$.
$\mathrm{card}(\mathbf{K})$	Cardinality of matrix $\mathbf{K}$, defined by the number of nonzero elements in $\mathbf{K}$.
$\mathrm{supp}(\mathbf{K})$	The support set of the matrix $\mathbf{K}$, i.e., the set of indices of the nonzero entries of matrix $\mathbf{K}$ [59].
$[\mathbf{K}]_s$	The matrix obtained by preserving only the s largest-magnitude entries of the matrix $\mathbf{K}$ and setting all other entries to zero.
$\nabla_{\mathbf{K}} J(\mathbf{K})$	The gradient of the function $J(\mathbf{K})$ with respect to the matrix $\mathbf{K}$, where element $[\nabla_{\mathbf{K}} J(\mathbf{K})]_{ij} = \partial J/\partial K_{ij}$.
$\Delta_{\mathrm{nwt}}(\mathbf{K},\mathcal{T})$	The restricted Newton step of function $f(\mathbf{K})$ at matrix $\mathbf{K}\in\mathbb{R}^{m\times n}$ under the structural constraint $\mathrm{supp}(\mathbf{K})\subset\mathcal{T}$. The computation of $\Delta_{\mathrm{nwt}}(\mathbf{K},\mathcal{T})$ is detailed in [2].
$\mathbf{K}^*_{\mathrm{nom}}(s)$	Solution for the feedback matrix that minimizes the H_2 norm (4.8) of the nominal system subject to the sparsity constraint s [2].
$\mathbf{K}^*_{\mathrm{rob}}(s)$	Solution for the robust feedback matrix in (P2) with $\gamma = 1$, sparsity constraint s.
$\mathbf{K}^*_{\mathrm{lqr}}$	Solution for the standard LQR problem which minimizes the H_2 norm (4.8) for the nominal system (4.4).
H_2 norm	The system H_2 norm in the time domain is $\|\|G\|\|_2 = \left(\int_0^\infty \left[\mathrm{trace}(H(t)^T H(t))\right] dt\right)^{1/2}$, where $H(t)$ is the impulse response of the system.
H_∞ norm	The system H_∞ norm $\|\|G\|\|_\infty = \sup_\omega \sigma_{\max}(G(j\omega))$, where $G(j\omega)$ is the system transfer matrix.

The matrices $\mathbf{A}$ and $\mathbf{B}$ are the nominal values of the state and input matrices, respectively, while $\Delta\mathbf{A}$ and $\Delta\mathbf{B}$ model the respective uncertainties. We assume that the pair $(\mathbf{A},\mathbf{B})$ is controllable, $(\mathbf{C}_2,\mathbf{A})$ is observable, and $\Delta\mathbf{A}$ and $\Delta\mathbf{B}$ have the form

$$[\Delta\mathbf{A}\ \Delta\mathbf{B}] = \mathbf{B}_1\Delta\delta[\mathbf{C}_1\ \mathbf{D}_1], \tag{3.3}$$

where $\mathbf{B}_1\in\mathbb{R}^{n\times m_1}$, $\mathbf{C}_1\in\mathbb{R}^{p_1\times n}$, $\mathbf{D}_1\in\mathbb{R}^{p_1\times m}$ are known matrices, $\Delta\delta\in\mathbb{R}^{m_1\times p_1}$ is an unknown matrix which is norm-bounded, satisfying $\Delta\delta^T\Delta\delta\preceq\rho^2\mathbf{I}$. The system (3.1) can be expressed as

the feedback interconnection of the following two subsystems:

$$\Sigma : \begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{B}_1\mathbf{w}_1(t) + \mathbf{B}_2\mathbf{w}_2(t) \\ \mathbf{z}_1(t) = \mathbf{C}_1\mathbf{x}(t) + \mathbf{D}_1\mathbf{u}(t) \\ \mathbf{z}_2(t) = \mathbf{C}_2\mathbf{x}(t) + \mathbf{D}_2\mathbf{u}(t) + \mathbf{D}_{22}\mathbf{w}_2(t) \end{cases} \tag{3.4}$$

$$\Sigma_K : \Big\{ \mathbf{w}_1(t) = \Delta\delta\mathbf{z}_1(t), \tag{3.5}$$

where $z_1(t) \in \mathbb{R}^{p_1 \times 1}$, $w_1(t) \in \mathbb{R}^{m_1 \times 1}$.

Our goal is to find a linear static state-feedback controller

$$\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t) \tag{3.6}$$

which not only stabilizes the system (3.1) despite the uncertainties, by imposing the requirement that the H_∞ norm from $\mathbf{w}_1$ to $\mathbf{z}_1$ is bounded below $\sigma_{\max}(\Delta\delta)$ [87], but also guarantees a desirable closed-loop performance. This dual objective is posed as a mixed H_2/H_∞ design problem, as follows.

Consider the system in (4.4–3.6). For simplicity and without loss of generality, we set $\mathbf{D}_{22} = \mathbf{0}$ [87] in (3.1). This closed-loop system is equivalent to

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}_{cl}\mathbf{x}(t) + \mathbf{B}_1\mathbf{w}_1(t) + \mathbf{B}_2\mathbf{w}_2(t) \\ \mathbf{z}_1(t) &= \mathbf{C}_{cl1}\mathbf{x}(t) \\ \mathbf{z}_2(t) &= \mathbf{C}_{cl2}\mathbf{x}(t) \end{aligned} \tag{3.7}$$

$$\begin{aligned} \mathbf{A}_{cl} &:= \mathbf{A} - \mathbf{B}\mathbf{K} \\ \mathbf{C}_{cli} &:= \mathbf{C}_i - \mathbf{D}_i\mathbf{K},\ i = 1, 2 \end{aligned} \tag{3.8}$$

The mixed H_2/H_∞ control problem [84] is defined as follows: Given an achievable H_∞-norm bound γ, find a feedback controller $\mathbf{K}$ that solves

$$\underset{\mathbf{K}}{\text{Minimize}}\ ||T_{z_2w_2}(\mathbf{K})||_2,\ \text{s.t.}||T_{z_1w_1}(\mathbf{K})||_\infty < \gamma. \tag{P1}$$

where $T_{z_iw_i}, i, = 1, 2$ denotes the closed-loop transfer function from $\mathbf{w}_i$ to $\mathbf{z}_i$.

3.3 Descent algorithm for mixed H_2/H_∞ control

3.3.1 The gradient method for mixed H_2/H_∞ control

Following standard robust control results such as in [84], it can be shown that the squared H_2 norm for the above problem is

$$||T_{z_2w_2}(\mathbf{K})||_2^2 = J(\mathbf{K}) := \text{trace}(\mathbf{B}_2^T\mathbf{P}\mathbf{B}_2) \tag{3.9}$$

where $\mathbf{P}$ is the solution of the Lyapunov equation

$$\mathbf{P}\mathbf{A}_{cl} + \mathbf{A}_{cl}^T\mathbf{P} + \mathbf{C}_{cl2}^T\mathbf{C}_{cl2} = 0. \tag{3.10}$$

The H_∞-norm constraint can be transformed into an LMI condition as follows [84]:

Theorem 1. The H_∞-norm constraint $||T_{z_1w_1}(\mathbf{K})||_\infty < \gamma$ if and only if there exists an $\mathbf{X} = \mathbf{X}^T$ that satisfies the linear matrix inequality (LMI) on $\mathbf{X}$ and γ^2

$$\begin{bmatrix} \mathbf{A}_{cl}\mathbf{X} + \mathbf{X}\mathbf{A}_{cl}^T + \mathbf{B}_1\mathbf{B}_1^T & \mathbf{X}\mathbf{C}_{cl1}^T \\ \mathbf{C}_{cl1}\mathbf{X} & -\gamma^2\mathbf{I} \end{bmatrix} \prec 0, \qquad \mathbf{X} \succ 0. \tag{3.11}$$

The authors of [84] proposed a gradient method to solve (4.7). The basic idea is to descend along the gradient of $J(\mathbf{K})$ (4.8) from an interior point of the feasible region of (P1) until the solution reaches the stationary point in the interior or on the boundary of the constraint set. Accordingly, the outcome $\mathbf{K}^*$ of this algorithm is a local minimum of the H_2 cost in the interior of or on the boundary of the H_∞-norm constraint set. However, this result might not be satisfactory if the H_∞-norm value of the initial guess is close to the boundary of the actual H_∞-norm constraint set, and the gradient-descent direction violates the H_∞-norm constraint even with a very small step size, thus causing an insufficient decrease of the H_2 norm. Our first goal is to find a remedy for this problem by selecting a different feasible direction for reducing the H_2 norm while simultaneously moving away from the boundary of the H_∞-constraint set. This approach, inspired by the Zoutendijk's method described in [85], is elaborated in the following section and applied to solve (P1).

3.3.2 Modified Zoutendijk's feasible direction method

In this section, we recall the Zoutendijk's method [85] as the foundation for general feasible direction methods, which requires evaluating the gradients of both the objective and the constraints functions. Moreover, to overcome the difficulty typically encountered while computing

the gradient of an H_∞ norm, we utilize the concept of level sets as in [86], as well as their LMI interpretation, to develop an alternative condition relative to the original formulation in the Zoutendijk's method.

Following the Zoutendijk's method, the inequality trace$[(\nabla_{\mathbf{K}} J(\mathbf{K}))^T \cdot \Delta\mathbf{K}]<0$ guarantees that an update direction $\Delta\mathbf{K}$ decreases the H_2 norm, the inequality trace$[(\nabla_{\mathbf{K}} ||T_{z_1 w_1}(\mathbf{K})||_\infty)^T \cdot \Delta\mathbf{K}] < 0$ can be used in the mixed H_2/H_∞ problem (P1) to check if $\Delta\mathbf{K}$ moves away from the H_∞ bound, where $J(\mathbf{K})$ is defined in (4.8). However, since it is difficult to compute the gradient $\nabla_{\mathbf{K}}$ of the H_∞ norm with respect to $\mathbf{K}$, we choose another condition for reduction of both the H_2 and H_∞ norm, by employing the descent method for the H_∞ norm based on level sets [86]. In the gain space of $\mathbf{K} \in \mathbb{R}^{m\times n}$, the set of all stabilizing $\mathbf{K}$ which satisfy (4.6), i.e., with H_∞ norm smaller than γ, is a level set

$$\mathcal{K}(\gamma) := \{\mathbf{K}|\ ||T_{z_1 w_1}(\mathbf{K})||_\infty < \gamma\}. \tag{3.12}$$

Given a stabilizing gain $\mathbf{K}_0$ and a sufficiently small γ_0 such that $\mathbf{K}_0 \in \mathcal{K}(\gamma_0)$, the method in [86] provides a way to construct a convex set $\hat{\mathcal{K}}(\gamma_0)$, which contains $\mathbf{K}_0$ near the boundary, where for all $\mathbf{K} \in \hat{\mathcal{K}}(\gamma_0)$, we have $||T_{w_1 z_1}(\mathbf{K})||_\infty < \gamma_0$. Furthermore, it gives rise to a sufficient LMI condition for the above $\mathbf{K}_0$, γ_0, and another gain $\mathbf{K}_{in}$, such that $\mathbf{K}_{in}$ is an inner point of $\hat{\mathcal{K}}(\gamma_0)$:

$$\mathbf{G}(\mathbf{K}_{in}; \mathbf{K}_0) \succ 0 \Rightarrow \mathbf{K}_{in} \in \hat{\mathcal{K}}(\gamma_0). \tag{3.13}$$

where $\mathbf{G}(\mathbf{K}_{in}; \mathbf{K}_0)$ is a matrix function of $\mathbf{K}_{in}$ and $\mathbf{K}_0$, γ_0. This matrix is positive definite if and only if $\mathbf{K}_{in}$ is in the level set $\hat{\mathcal{K}}(\gamma_0)$. The details of computing $\mathbf{G}(\mathbf{K}_{in}; \mathbf{K}_0)$ are provided in Appendix B.1.

Given a feasible point $\mathbf{K}$ and the gradient $\nabla_{\mathbf{K}} J(\mathbf{K})$ of the cost function $\mathbf{J}(\mathbf{K})$ in (4.8) at $\mathbf{K}$,

Algorithm 3.4 Improved version of the descent algorithm in [84] for mixed H_2/H_∞ control

1 Initialization: $\mathbf{K}_0$ feasible with $T_\infty(\mathbf{K}_0) < \gamma$.
2 Gradient descent on H_2 norm:
while $T_\infty(\mathbf{K}_i) < \gamma - \epsilon_1$ **do**
  $\mathbf{K}_{i+1} := \mathbf{K}_i + d_i \nabla_{\mathbf{K}} J(\mathbf{K}_i)$.
end while
3 Feasible direction search:
while $|J(\mathbf{K}_{i+1}) - J(\mathbf{K}_i)| < \epsilon_3$ **do**
  3.1 Solve for an inner point $\mathbf{K}_{in}$ using (4.34) with $\mathbf{K} = \mathbf{K}_i$, and set the feasible direction $\Delta\mathbf{K} := \mathbf{K}_{in} - \mathbf{K}_i$.
  3.2 Let $\mathbf{K}_{i+1} = \mathbf{K}_i + d_i \Delta\mathbf{K}$.
end while

we can find an improved direction for the mixed H_2/H_∞ problem (4.7) by solving the following optimization problem

$$\begin{aligned} \underset{z,\mathbf{K}_{in}}{\text{Maximize}} \quad & z \\ \text{s.t.} \quad & \text{trace}[(\nabla_{\mathbf{K}} J(\mathbf{K}))^T(\mathbf{K}_{in}-\mathbf{K})] + z \le 0 \\ & \mathbf{G}(\mathbf{K}_{in};\mathbf{K}) - \theta z \cdot \mathbf{I} \succeq 0 \end{aligned} \tag{3.14}$$

where $\theta \ge 0$ is a predetermined factor that controls how far $\mathbf{K}$ moves away from the H_∞-norm boundary. The value of θ determines the speed of reduction of the H_∞ norm. Thus, a small value of θ results in a less aggressive shrinkage of the H_∞ norm. If the solution z^* in (4.34) is positive, then $\mathbf{K}_{in} - \mathbf{K}$ is an improving feasible direction, otherwise an improving feasible direction cannot be found.

We next apply (4.34) for improving on the gradient method of [84] for solving (P1). Algorithm 3.4 summarizes our proposed method. For simplicity, we denote $||T_{z_1w_1}(\mathbf{K})||_\infty$ as $T_\infty(\mathbf{K})$ and $||T_{z_2w_2}(\mathbf{K})||_2$ as $T_2(\mathbf{K})$.

Note that Algorithm 3.4 does not necessarily start in the feasible region. Starting with any stabilizing gain $\mathbf{K}_0^{\text{ext}}$, we can find a gain matrix $\mathbf{K}_0$ with $T_\infty(\mathbf{K}_0) < \gamma$ by iteratively solving (4.32) to determine a sequence of inner points with decreasing H_∞ norms until γ is achieved [86]. In Step 2 and 3, the step size d_i is determined by a backtracking line search using the Armijo condition [85]

$$J(\mathbf{K}_i + d_i\mathbf{\Delta K}) \le J(\mathbf{K}_i) + d_i\beta \cdot \text{trace}[(\nabla_{\mathbf{K}} J(\mathbf{K}_i))^T \Delta\mathbf{K}], \tag{3.15}$$

which is less expensive computationally than the exact line search in [84]. In (3.15), $\beta \in (0,1)$ is a predetermined constant. In this case we do not need to check $T_\infty(\mathbf{K}_i + d_i\mathbf{\Delta K}) < \gamma$ if the backtracking starts at $d_i = 1$. Since $\hat{\mathcal{K}}(\gamma_i)$ is a convex set with $T_\infty(\mathbf{K}_i) < \gamma_i$, if $\mathbf{K}_{in} = \mathbf{K}_i + \Delta\mathbf{K}$ is an inner point, then any $\mathbf{K}_i + d_i\Delta\mathbf{K}$ is also an inner point of $\hat{\mathcal{K}}(\gamma_i)$ for $d_i \in [\epsilon_2, 1)$. The parameters ϵ_1, ϵ_2, ϵ_3 are sufficiently small positive constants, where ϵ_1 is the margin to guard H_∞ norm within the γ-constraint for numerical stability, ϵ_2 is the smallest step size in the line search, and ϵ_3 is the smallest change in the cost function $J(\mathbf{K})$ in two consecutive iterations.

In Step 2, if Algorithm 3.4 achieves sufficiently small $\nabla_{\mathbf{K}} J(\mathbf{K}_i)$ within the H_∞-norm constraint, the algorithm terminates and the resulting feedback matrix is a local minimum of H_2 norm in the interior of the H_∞-norm constraint. Otherwise, when $\mathbf{K}$ is sufficiently near the H_∞-norm boundary and the algorithm exits Step 2, the direction-search Step 3 will change the current gradient-descent direction and continue to decrease the H_2 norm.

3.4 Sparsity-constrained H_2/H_∞ control

In this section, we design sparse controllers for the mixed H_2/H_∞ problem. For the linear static state feedback (3.6), $\mathbf{K}$ is a $m \times n$ matrix, with coefficient K_{ij} being the feedback gain from state j to control input i. If the coefficient $K_{ij} \neq 0$, a communication link that delivers the data from state j to control input i is required. One way to save communication cost, therefore, is to promote sparsity in $\mathbf{K}$. The level of sparsity can be quantified by the cardinality of the $\mathbf{K}$ matrix, denoted by $\text{card}(\mathbf{K})$.

The sparsity criterion can further be adapted to quantify the communication cost. For example, if the system consists of N nodes and each node i has n_i states and m_i control inputs, the communication cost can be defined as the number of communication links connecting different nodes. Specifically, if the states $\mathbf{x}(t)$ and the control inputs $\mathbf{u}(t)$ in (3.6) are organized according to their physical locations, the matrix $\mathbf{K}$ is in the form

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \dots & \mathbf{K}_{1N} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \dots & \mathbf{K}_{2N} \\ & & \vdots & \\ \mathbf{K}_{N1} & \mathbf{K}_{N2} & \dots & \mathbf{K}_{NN} \end{bmatrix}, \tag{3.16}$$

where the block $\mathbf{K}_{ij} \in \mathbb{R}^{m_i \times n_j}$ represents feedback from the states of node j to the control inputs of node i [2]. For the model (3.16), we define the communication cost as the number of communication links associated with the off-diagonal blocks of $\mathbf{K}$, given by

$$\text{card}_{\text{off}}(\mathbf{K}) = \textstyle\sum_{i,j=1,i\neq j}^{N} \text{nnz}(\mathbf{K}_{ij}). \tag{3.17}$$

where $\text{nnz}(\cdot)$ counts the number of nonzero elements in a matrix. The communication cost (3.17) will be employed in the numerical example of Section 3.5.

We are now ready to address the sparsity-constrained version of problem (P1), stated as follows:

The sparsity-constrained mixed H_2/H_∞ control problem: Given an achievable H_∞-norm bound γ and a sparsity level $s \geq 0$, find a feedback controller $\mathbf{K}$ which solves

$$\begin{aligned} &\min_{\mathbf{K}} \ ||T_{z_2 w_2}(\mathbf{K})||_2, \\ &\text{s.t. } ||T_{z_1 w_1}(\mathbf{K})||_\infty < \gamma, \ \text{card}(\mathbf{K}) \leq s \end{aligned} \tag{P2}$$

The problem (P2) reduces to (P1) if the constraint $\text{card}(\mathbf{K}) \leq s$ is removed in (P2). To address the sparsity constraint in (P2), we combine the approach in Algorithm 3.4 with the GraSP-based algorithm in [2]. The resulting procedure is presented in Algorithm 3.5. Given a sparsity

constraint $s \geq 0$, we start with a stabilizing gain $\mathbf{K}_0^{\text{ext}}$ with $\text{card}(\mathbf{K}_0^{\text{ext}}) \leq s$, but not necessarily satisfying the H_∞-norm constraint. If the open-loop system $\mathbf{A}$ is stable, $\mathbf{K}_0^{\text{ext}}$ can be chosen as the zero matrix. Otherwise, the sparsity-promoting method in [81] can be used with a large ℓ_1-penalty, and $\mathbf{K}_0^{\text{ext}}$ can be the feedback with the sparsest structure. Based on $\mathbf{K}_0^{\text{ext}}$, in Step 1, we find an improved initial point by keeping the nonzero structure in $\mathbf{K}_0^{\text{ext}}$ and decrease the H_∞ norm by iteratively finding an inner point of the current H_∞-norm level set. Note that the original descent method in [86] did not include any structural or sparsity constraint. We extend the LHS of the LMI condition (4.32) to impose the sparsity structure by adding another linear constraint

$$G(\mathbf{K}; \mathbf{K}_0) \succ 0 \tag{3.18a}$$

$$\mathbf{K} \odot \bar{\mathbf{I}}_s = 0 \tag{3.18b}$$

where $\mathbf{K}$ is the optimization variable, $\mathbf{I}_s \in \mathbb{R}^{m \times n}$ is the structural indicator matrix which has elements 1 and 0 at the positions of nonzero and zero entries of $\mathbf{K}_0$, respectively, and $\bar{\mathbf{I}}_s$ is the complement of $\mathbf{I}_s$. The operator $\odot$ is the element-wise product of the two matrices. This procedure for tuning values of nonzero entries of the gain matrix to satisfy the H_∞-norm constraint is summarized in the subroutine "TuneGain" in Algorithm 3.5. If a feasible solution $\mathbf{K}_0$ cannot be achieved by "TuneGain", we can instead promote row sparsity in $\mathbf{K}$ subject to (4.6) by a r_1-norm minimization as in [21].

The greedy selection and update steps are similar to those in the sparsity-constrained LQR method in [2]. In each iteration of Step 2.1–2.4, the algorithm extends the matrix $\mathbf{K}$ along its steepest $2s$ gradient-descent directions. In Step 2.1, the gradient of $\mathbf{J}(\mathbf{K})$ over $\mathbf{K}$ is computed as [81]

$$\nabla_{\mathbf{K}} J(\mathbf{K}) = 2(\mathbf{R}\mathbf{K} - \mathbf{B}^{\mathrm{T}}\mathbf{P})\mathbf{L} \tag{3.19}$$

where the matrices $\mathbf{P}$ and $\mathbf{L}$ are the unique solutions of the following Lyaponuv equations

$$\begin{aligned} &(\mathbf{A}-\mathbf{B}\mathbf{K})^{\mathrm{T}}\mathbf{P}+\mathbf{P}(\mathbf{A}-\mathbf{B}\mathbf{K}) + \mathbf{Q}+\mathbf{K}^T\mathbf{R}\mathbf{K} = 0 \\ &(\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{L} + \mathbf{L}(\mathbf{A} - \mathbf{B}\mathbf{K})^T + \mathbf{B}_2\mathbf{B}_2^{\mathrm{T}} = 0 \end{aligned} \tag{3.20}$$

In Step 2.4, the matrix $\mathbf{K}$ is updated using the restricted Newton step, and the step size d_i is chosen similarly to (3.15), with the additional constraint that it must satisfy $T_\infty(\mathbf{K}_i + d_i\Delta_{nwt}(\mathbf{K}_i, \mathcal{T})) < \gamma$. In Step 2.5, pruning is performed to impose the constraint s where we only keep the nonzero entries with the s largest magnitudes and truncate other entries to zero. This process can change the H_2 norm and the H_∞ norm of the feedback matrix achieved at Step 2.4. If the pruned result violates the H_∞-norm constraint, we use the subroutine "TuneGain" to tune the nonzero values in the gain matrix similarly to Step 1. In case of instabilizing gain after the

pruning step, we adopt a backtracking option to return to a previously found stable solution, which has cardinality less than s. This option can also be used when a suitable $\mathbf{K}_0$ cannot be found in Step 1. The stopping criterion for the iteration Step 2 is given by a sufficiently small change in the feedback matrix, where ϵ_{abs} and ϵ_{rel} are small constants that quantify the absolute and relative change in the feedback matrix, respectively. When Step 2 converges, we have selected the nonzero entries of the gain matrix, and Step 3 further tunes the values of the nonzero entries for better H_2 performance. In Step 3, given the gain matrix $\mathbf{K}_i$ after Step 2 and the associated structural indicator matrix $\mathbf{I}_s$, we iteratively find a feasible direction using (4.34) and (3.18b). In Step 3.2, the step size $d_i \in [\epsilon_1, 1)$ is determined by backtracking line search using condition (3.15). Similarly to Algorithm 1, we stop the polishing step when there is sufficiently small decrease in the H_2 objective, quantified by the small constant ϵ_3.

Algorithm 3.5 Mixed H_2/H_∞ control algorithm with sparsity constraint

Given s: sparsity constraint, γ: H_∞-norm bound,
$\mathbf{K}_0^{\text{ext}}$: stabilizing feedback gain with $\text{card}(\mathbf{K}_0^{\text{ext}}) \leq s$.
1 Initialization: $\mathbf{K}_0 = \text{TuneGain}(\mathbf{K}_0^{\text{ext}}, \gamma)$.
2 Iteration:
while $||\mathbf{K}_{i+1} - \mathbf{K}_i||_{\text{F}} > \epsilon_{\text{abs}}\sqrt{mn} + \epsilon_{\text{rel}}||\mathbf{K}_i||_{\text{F}}$ **do**
2.1 Compute gradient of $J(\mathbf{K})$ w.r.t $\mathbf{K}$ at $\mathbf{K}_i$: $\mathbf{g} = \nabla_{\mathbf{K}} J(\mathbf{K}_i)$.
2.2 Identify up to $2s$ directions: $\mathcal{Z} = \text{supp}([\mathbf{g}]_{2s})$
2.3 Merge support: $\mathcal{T} = \mathcal{Z} \cup \text{supp}(\mathbf{K}_i)$
2.4 Descent along the Newton step of $J(\mathbf{K})$ restricted to $\mathcal{T}$: $\mathbf{K}_{i+1} = \mathbf{K}_i + \lambda \Delta_{nwt}(\mathbf{K}_i, \mathcal{T})$.
2.5 Prune $\mathbf{K}_{i+1}$: $\mathbf{K}_{i+1} = [\mathbf{K}_{i+1}]_s$. If $T_\infty(\mathbf{K}_{i+1}) \geq \gamma$, set $\mathbf{K}_{i+1} = \text{TuneGain}(\mathbf{K}_{i+1}, \gamma)$.
end while
3 Polishing:
Set $\mathbf{I}_s$ to be the structural indicator matrix associated with $\mathbf{K}_i$ at the end of Step 2.
while $|J(\mathbf{K}_{i+1}) - J(\mathbf{K}_i)| > \epsilon_3$ **do**
3.1 Solve for $\mathbf{K}_{in}$ using (4.34) with additional constraint $\mathbf{K}_{in} \odot \bar{\mathbf{I}}_s = 0$ with $\mathbf{K} = \mathbf{K}_i$, and let $\Delta\mathbf{K} = \mathbf{K}_{in} - \mathbf{K}_i$.
3.2 Let $\mathbf{K}_{i+1} = \mathbf{K}_i + d_i \Delta\mathbf{K}$, where $d_i <= 1$ is determined by backtracking line search using (3.15).
end while

procedure TuneGain($\mathbf{K}_0$, γ)
while True **do**
Find γ^* which is the minimizer of γ in (4.6) with $\mathbf{K} = \mathbf{K}_0$ fixed.
If $\gamma^* < \gamma$, return $\mathbf{K}_0$.
Set $\mathbf{K}_0 := \mathbf{K}$, where $\mathbf{K}$ is the solution to (3.18).
end while
end procedure

Note that the objectives in [2,81], as well as in the algorithms of this chapter, are not convex. Thus, convergence of these methods cannot be guaranteed, and they can produce local minima. However, extensive simulations were carried out to verify convergence of the proposed method empirically, as illustrated in the next section.

3.5 Numerical Example

Consider a network that consists of N connected nodes distributed randomly on a L unit by L unit square area [81, 88]. Each node is an unstable second-order system, with the state-space representation:

$$\begin{bmatrix} \dot{x}_{1i} \\ \dot{x}_{2i} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} + \sum_{j \neq i} e^{-l^0(i,j)} \begin{bmatrix} x_{1j} \\ x_{2j} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (d_i + u_i). \tag{3.21}$$

To account for model uncertainty, the actual state-space representation of the i^{th} node is written as

$$\begin{bmatrix} \dot{x}_{1i} \\ \dot{x}_{2i} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} + \sum_{j \neq i} e^{-\hat{l}(i,j)} \begin{bmatrix} x_{1j} \\ x_{2j} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (d_i + u_i), \tag{3.22}$$

where $a_{ij} = a_{ij}^0 \cdot (1+\theta_{ij})$, and a_{ij}^0 is the nominal value in (3.21). The Euclidean distance between the nodes i and j is given by $\hat{l}(i,j)$, where $\hat{l}(i,j) = l^0(i,j) \cdot (1 + \delta_{ij})$, and $l^0(i,j)$ is the known nominal value of the distance between the nodes i and j. In general, as L increases, the nodes spread farther apart and the $l^0(i,j)$ values grow. The random perturbations δ_{ij} and θ_{ij} are uniformly distributed in the range $\pm 20\%$.

Define $\hat{\mathbf{A}}$ such that $\hat{\mathbf{A}} - \mathbf{A} = \mathbf{B}_1 \Delta \delta \mathbf{C}_1$. To stabilize the system in (4.53), we numerically compute the worst-case $\hat{\mathbf{A}}$ as $\hat{\mathbf{A}}_{\text{worst}} = \arg\max_{\hat{\mathbf{A}}} \sigma_{\max}(\hat{\mathbf{A}} - \mathbf{A})$. By the singular value decomposition $\mathbf{U}\mathbf{S}\mathbf{V}^T = \hat{\mathbf{A}}_{\text{worst}} - \mathbf{A}$ and normalizing $\mathbf{S}$ by $\sigma_{\max}(\mathbf{S})$, we set $\mathbf{B}_1 = \sqrt{\sigma_{\max}(\mathbf{S})}\mathbf{U}$, $\mathbf{C}_1 = \sqrt{\sigma_{\max}(\mathbf{S})}\mathbf{V}^T$. Due to this normalization, $\gamma = 1$.

We apply the proposed Algorithm 3.5 to (4.53) with $N = 5$ and $L = 2$. In this chapter, we solve all the LMIs using the CVX package [89]. For the sparsity constraint, as discussed in Section 3.4, we are only concerned with the communication cost quantified by the number of nonzero entries on the off-diagonal blocks of $\mathbf{K}$ (3.16), representing feedback from $x_{ki}, k = 1, 2$ to u_j for $i \neq j$. We examine the performance of Algorithm 2 over a range of s-values from completely decentralized feedback ($s = 0$) to dense feedback ($s = 2N(N-1)$) under the constraint $\gamma = 1$, and the resulting feedback matrix is denoted as $K_{\text{rob}}^*(s)$. The initial value $\mathbf{K}_0^{\text{ext}}$ is chosen as a fully decentralized stabilizing feedback matrix, i.e., $\text{card}_{\text{off}}(\mathbf{K}_0^{\text{ext}}) = 0$. For comparison, we also use Algorithm 1 in [2] to obtain the sparsity-constrained LQR

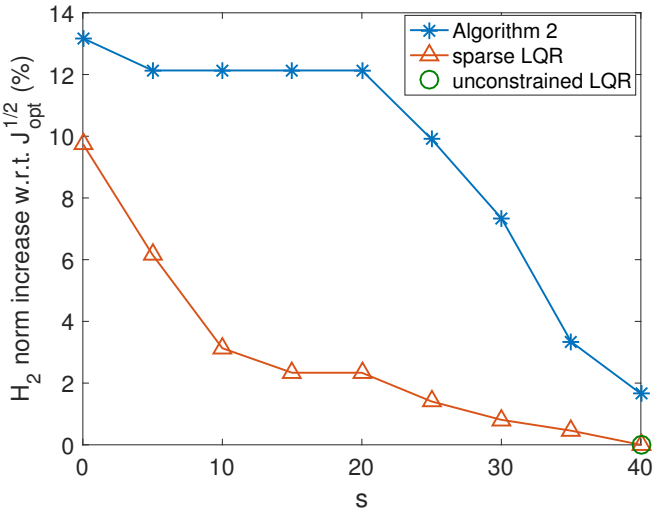


Figure 3.1: H_2 norm increase relative to the H_2 norm of standard LQR vs. sparsity constraint s for sparse LQR [2] and Algorithm 2.

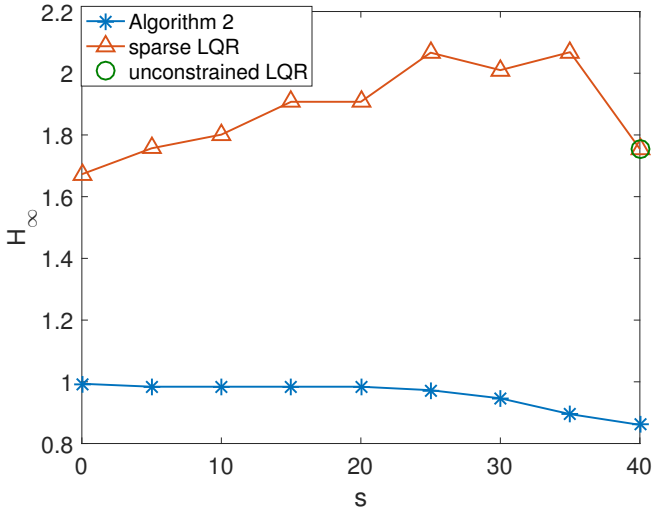


Figure 3.2: H_∞ norm vs. sparsity constraint s for sparse LQR [2] and Algorithm 2.

optimization for the same range of s-values. The latter algorithm does not consider H_∞ norm in the optimization. The solution of the feedback gain matrix for the sparsity-constrained LQR optimization is denoted by $\mathbf{K}^*_{\text{nom}}(s)$.

In Figure 3.1, the blue curves show increase in the H_2 norm relative to the optimal LQR metric $J^{1/2}_{\text{opt}}$ (4.8) for the system (4.53) for $\mathbf{K}^*_{\text{rob}}(s)$ and $\mathbf{K}^*_{\text{nom}}(s)$ vs. the sparsity constraint s. Note that the H_2 norm decreases with decreasing sparsity, meaning that increasing sparsity

in the feedback structure would sacrifice the H_2-norm performance to some extent. Moreover, in the case of dense feedback ($s = 40$), the H_2 norms of $\mathbf{K}^*_{\text{nom}}(s = 40)$ and the standard LQR feedback are the same and are given by $J_{opt}^{1/2}$ (4.8), while using $\mathbf{K}^*_{\text{rob}}(s = 40)$ results in 1.7% increase. The red curves in Figure 3.1 show the H_∞ norm for $\mathbf{K}^*_{\text{rob}}(s)$ and $\mathbf{K}^*_{\text{nom}}(s)$ vs. s. Note that the H_∞ norms of the standard and sparse LQR controllers exceed $\gamma = 1$, and, thus, cannot guarantee stability for the uncertain system (4.53) while for the robust feedback, $T(\mathbf{K}^*_{\text{rob}}(s)) < 1{=}\gamma$ for all s-values, which is guaranteed to stabilize (4.53) with perturbations in the $\mathbf{A}$ matrix. Finally, we observe that for all s-values, the H_2 norm $T_2(\mathbf{K}^*_{\text{rob}}(s)) \geq T_2(\mathbf{K}^*_{\text{nom}}(s))$, which is due to the additional H_∞-norm constraint in (P2) relative to the sparsity-constrained LQR problem in [2].

We show the effectiveness of the proposed approach for stabilizing uncertain systems by plotting the closed-loop poles of (4.53) using robust and nominal feedback for random realizations of the uncertain matrix $\mathbf{A}$ in (4.53). We generated 200 random realizations $\hat{\mathbf{A}}^i$, $i = 1, ..., 200$ and applied the robust feedback $\mathbf{K}^*_{\text{rob}}(s)$ and the nominal feedback $\mathbf{K}^*_{\text{nom}}(s)$ to $\hat{\mathbf{A}}^i$'s. The resulting closed-loop poles for several s-values are shown in Figure 3.3. In the left two plots in Figure 3.3, the nominal feedback $\mathbf{K}^*_{\text{nom}}(s)$ is applied. We observe that all the closed-loop poles of the nominal system have negative real parts, which indicates that the closed-loop nominal system without uncertainty with nominal sparse feedback [2] is stable. However, when $\mathbf{A}$ is perturbed, the nominal feedback becomes mismatched, and several poles cross over to the right of the imaginary axis, destabilizing the closed-loop system. In the right three plots of Figure 3.3, the proposed robust feedback $\mathbf{K}^*_{\text{rob}}(s)$ is applied. In this case, all closed-loop poles have negative real parts for both the nominal and the uncertain models.

3.6 Conclusions

A sparsity-constrained mixed H_2/H_∞ controller was developed by combining an improved descent approach with GraSP method to achieve a predetermined level of sparsity. The robustness properties of the proposed controller are illustrated using simulations of an open-loop unstable plant. It is demonstrated that this controller maintains stability over a range of sparsity constraints with a modest degradation of the H_2 norm performance relative to the sparse controller for the nominal system. Our future work will include investigation of convergence properties of Algorithm 3.4, extension of this method to output-feedback, distributed, and multi-agent implementations, as well as applying it in large-scale systems, such as wide-area control of power systems with parametric uncertainties.

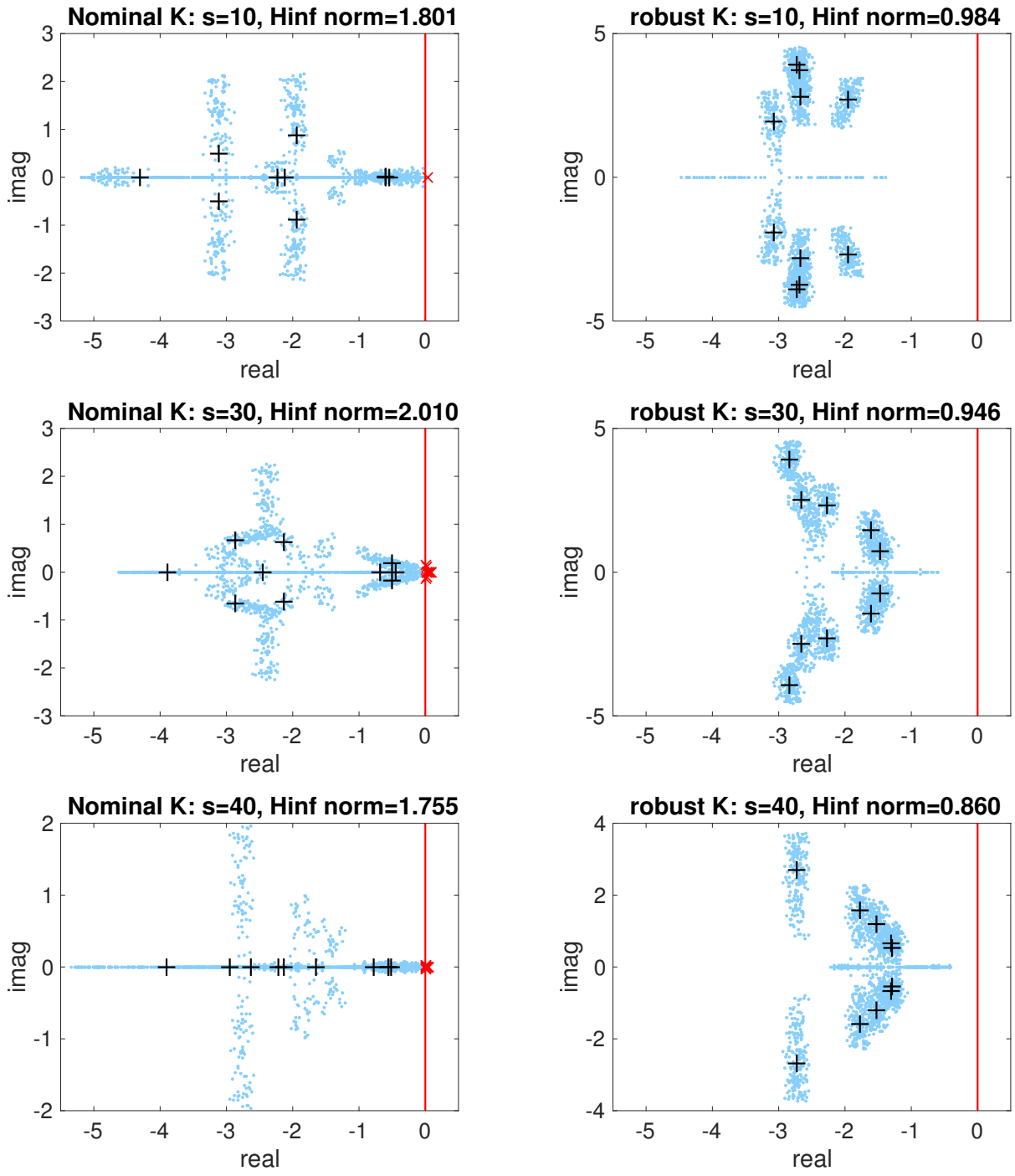


Figure 3.3: The closed-loop poles for the unstable nodes system using the nominal feedback $\mathbf{K}^*_{\text{nom}}(s)$ and the robust feedback $\mathbf{K}^*_{\text{rob}}(s)$ for two sparsity constraint values s. The blue circles and the red crossmarks represent the stable and unstable closed-loop poles of the uncertain $\hat{\mathbf{A}}^i$'s, respectively. The black pluses represent the closed-loop poles of the nominal system without uncertainty. The imaginary axis is shown by the red vertical line.

Chapter 4

Game-Theoretic Mixed H_2/H_∞ Control with Sparsity Constraint for Multi-agent Networked Control Systems

Multi-agent NCSs are often subject to model uncertainty and are limited by large communication cost, associated with feedback of data between the system nodes. To provide robustness against model uncertainty and to reduce the communication cost, this chapter investigates the mixed H_2/H_∞ control problem for NCS under the sparsity constraints. First, PALM is employed to solve the centralized social optimization algorithm where the agents have the same optimization objective that . Next, we investigate a sparsity-constrained noncooperative game which accommodates different control-performance criteria of multiple agents, and propose a best-response dynamics algorithm based on PALM that converges to an approximate GNE of this game. A special case of this game, where the agents have the same H_2 objective, produces a partially-distributed social optimization solution. We validate the proposed algorithms using a network with unstable node dynamics and demonstrate the superiority of the proposed PALM-based method to a previously investigated sparsity-constrained mixed H_2/H_∞ controller.

4.1 Introduction

Recent papers, such as [2, 8, 81, 82, 90, 91], have proposed various sparse H_2 control designs for reducing communication costs of controlling large multi-agent linear time invariant (LTI) networks. However, the majority of these papers ignore the effects of model uncertainties, which

are bound to arise in all practical large-scale systems since the network operating conditions and the resulting network topology might change frequently over time. Even if the topology is fixed, the designer might not know all model parameters or have mismatched parameter values. To address this problem, several works, e.g. [11,22,23,26,92], consider designs robust to uncertainty in the system dynamics or controller implementation. In particular, [22,23,26,92] focuses on the H_∞-controller suitable for norm-bounded parameter uncertainty in the system dynamics. Our objective in this chapter is twofold. First, to design sparse controllers for an uncertain system; Second, to examine this design in light of a non-cooperative game to understand how individual agent dynamics and uncertainty are playing a role in the sparse structure of the controller.

While a global control performance cost is often employed in dynamic systems, this metric does not address the individual objectives of multiple agents. Decision-making in multi-agent systems with uncertainty received significant attention due to its wide application in various domains, such as wide-area control of power systems [2], multi-robot coordination, multi-access broadcast channel, vehicle formation, and wireless sensor network [93, 94]. In these problems, game theory becomes a powerful tool, with different control inputs modeled as game players, where each player tries to optimize its individual objective using an associated control policy. Differential games were investigated for uncertain multi-agent systems, and algorithms for finding an equilibrium point were proposed based on solving a set of coupled optimization problems. The works [42, 43] extend Nash-type differential game in [38] by finding robust Nash strategies while either considering polytopic uncertainty or formulating uncertain external disturbance as a fictitious player. The works [44–47] model the uncertainty in the multi-agent system stochastic differential equations, and the Nash strategies are found by solving cross-coupled matrix equations, using necessary optimality conditions or Karush-Kuhn-Tucker (KKT) conditions. Moreover, reinforcement learning has been applied to multi-agent control problems when the system parameters are completely or partially unknown [49–52].

In this chapter, we investigate the controller design that aims to reduce the H_2 cost under the H_∞ and sparsity constraints. While we focus on the norm-bounded parameter uncertainty in the system translated into an H_∞ constraint as in [26, 92], the sparse controller designed in this chapter does not depend on any sparse structure of the state matrix and satisfies the hard constraint on the cardinality of the feedback gain. We employ the proximal alternating linearization method (PALM) [90], which has proven to be effective for optimization for nonconvex nonsmooth problems [56] and was utilized in [90] in a sparsity-constrained output-feedback co-design problem. First, a centralized sparsity-constrained mixed H_2/H_∞ controller is investigated, which can be viewed as the social optimization where all agents have a consensus control objective. In [92], we developed a centralized controller under the sparsity and H_∞ constraints using a greedy gradient support pursuit (GraSP) method [59]. However, the algorithm in [92] requires the knowledge of an initial stabilizing feedback gain that satisfies the sparsity con-

Table 4.1: Notation

Term	Definition
$\mathbf{M}\succ 0(\succeq 0)$	Matrix $\mathbf{M}$ is positive definite (semidefinite)
$\mathbf{M}\prec 0(\preceq 0)$	$\mathbf{M}$ is negative definite (semidefinite)
$\sigma_{\max}(\mathbf{M})$	maximum singular value of $\mathbf{M}$
$\|\|\mathbf{K}\|\|_F$	Frobenius norm of the matrix $\mathbf{K}$, defined by $\sqrt{\text{trace}(\mathbf{K}^{\mathrm{T}}\mathbf{K})}$.
$\text{card}(\mathbf{K})$	Cardinality of matrix $\mathbf{K}$, defined by the number of nonzero elements in $\mathbf{K}$.
$\nabla_{\mathbf{K}} J(\mathbf{K})$	The gradient of the scalar function $J(\mathbf{K})$ with respect to the matrix $\mathbf{K}$. Assuming $\mathbf{K} \in \mathbb{R}^{m\times n}$, $\nabla_{\mathbf{K}} J(\mathbf{K})$ is given by a $m \times n$ matrix with the elements $[\nabla_{\mathbf{K}} J(\mathbf{K})]_{ij} = \partial J/\partial K_{ij}$.
$[\mathbf{K}]_s$	The matrix obtained by preserving only the s largest-magnitude entries of the matrix $\mathbf{K}$ and setting all other entries to zero.
H_2 norm	The system H_2 norm in the time domain is $\|\|G\|\|_2 = \left(\int_0^\infty \left[\text{trace}(H(t)^T H(t))\right] dt\right)^{1/2}$, where $H(t)$ is the impulse response of the system.
H_∞ norm	The system H_∞ norm $\|\|G\|\|_\infty = \sup_\omega \sigma_{\max}(G(j\omega))$, where $G(j\omega)$ is the system transfer matrix.

straint. We eliminate this requirement, compare sparse controllers based on GraSP [92] and PALM, and show the advantage of the latter in terms of the quadratic cost. Second, we extend the proposed design to the multi-agent scenario where each agent designs its own part of the feedback matrix subject to a shared global H_∞-norm and sparsity constraints. Since each agent has different individual cost, the controller design for this multi-agent system is modeled as a noncooperative game with shared constraints. We develop a numerical algorithm to find the generalized Nash equilibrium (GNE) [95] of the game. The proposed algorithm has partially-distributed computation, i.e., in the first stage, each player computes its own feedback matrix, while in the second stage, the sparse links are chosen globally based on the results from the first stage. Third, assuming all players of the game above have the same H_2-optimization objective, we develop a potential game that yields a partially-distributed multi-agent implementation of the social optimization. Finally, we briefly analyze convergence of the proposed centralized and game-theoretic algorithms.

The main contributions of this chapter are:

- Development and analysis of a centralized, sparsity-constrained mixed H_2/H_∞ controller for social optimization of multi-agent systems with norm-bounded uncertainty.
- Development of game-theoretic, partially-distributed algorithms that aim to minimize the agents' H_2-norms under shared sparsity and H_∞-norm constraints.

This chapter is organized as follows. Section 4.2 presents the system model with parametric uncertainty and develops a centralized PALM algorithm for sparsity-constrained mixed H_2/H_∞ control. Section 4.3 describes a multi-agent system with parametric uncertainty, proposes a non-cooperative game with shared sparsity and H_∞ constraints, and develops partially-distributed numerical algorithms for this game and for social optimization. Section 4.4 demonstrates effectiveness of the proposed algorithms using numerical simulations. Finally, Section 4.5 discusses future directions and concludes this chapter.

Throughout this chapter, matrices are denoted with boldface capital letters. If $\mathbf{M}$ is a symmetric matrix, the upper block matrices are sometimes denoted by $*$ to save space. The notation used in this chapter is summarized in Table 4.1.

4.2 PALM algorithm for centralized sparsity-constrained mixed H_2/H_∞ control

4.2.1 System model and mixed H_2/H_∞ control

In this section, we assume centralized social optimization where all agents share the same control objective. To model this scenario, it is not necessary to consider individual agents. Thus, in this section we investigate the following linear time invariant system with model uncertainty

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \underbrace{(\mathbf{A}+\Delta\mathbf{A})}_{\hat{\mathbf{A}}}\mathbf{x}(t) + \underbrace{(\mathbf{B}+\Delta\mathbf{B})}_{\hat{\mathbf{B}}}\mathbf{u}(t) + \mathbf{B}_2\mathbf{w}_2(t)\\ \mathbf{z}_2(t) &= \mathbf{C}_2\mathbf{x}(t) + \mathbf{D}_2\mathbf{u}(t) + \mathbf{D}_{22}\mathbf{w}_2(t)\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned} \tag{4.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n\times 1}$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^{m\times 1}$ is the control input vector, $\mathbf{w}_2(t) \in \mathbb{R}^{m_2\times 1}$ is the exogenous input, $\mathbf{z}_2(t) \in \mathbb{R}^{p_2\times 1}$ is the performance output, and $\mathbf{y}(t)\in\mathbb{R}^{p\times 1}$ is the measured output. The matrices $\mathbf{A}$ and $\mathbf{B}$ are the nominal values of the state and input matrices, respectively while $\Delta\mathbf{A}$ and $\Delta\mathbf{B}$ model the respective uncertainties.

We employ the following assumptions:

Assumption 4.2.1: (i) The pair $(\mathbf{A},\mathbf{B})$ is controllable, $(\mathbf{C},\mathbf{A})$ is observable, $(\mathbf{C}_2,\mathbf{A})$ is observable.
(ii) $\Delta\mathbf{A}$ and $\Delta\mathbf{B}$ have the form [26]

$$[\Delta\mathbf{A}\ \Delta\mathbf{B}] = \mathbf{B}_1\Delta\boldsymbol{\delta}[\mathbf{C}_1\ \mathbf{D}_1], \tag{4.2}$$

where $\mathbf{B}_1\in\mathbb{R}^{n\times m_1}$, $\mathbf{C}_1\in\mathbb{R}^{p_1\times n}$, $\mathbf{D}_1\in\mathbb{R}^{p_1\times m}$ are known matrices, $\Delta\boldsymbol{\delta}\in\mathbb{R}^{m_1\times p_1}$ is an unknown

matrix which is norm-bounded, satisfying $\Delta\boldsymbol{\delta}^T\Delta\boldsymbol{\delta} \preceq \rho^2\mathbf{I}$.

Assumption 4.2.2: Matrices $\mathbf{C}_2$ and $\mathbf{D}_2$ have the following form:

$$\mathbf{C}_2 = \left[\mathbf{C}_2^1, \mathbf{0}\right], \mathbf{D}_2 = \begin{bmatrix}\mathbf{0}\\ \mathbf{D}_2^2\end{bmatrix}, \mathbf{C}_2^T\mathbf{D}_2 = 0, \tag{4.3}$$

The system (4.1) can be expressed as the feedback interconnection of the following two subsystems:

$$\Sigma : \begin{cases}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{B}_1\mathbf{w}_1(t) + \mathbf{B}_2\mathbf{w}_2(t)\\ \mathbf{z}_1(t) = \mathbf{C}_1\mathbf{x}(t) + \mathbf{D}_1\mathbf{u}(t)\\ \mathbf{z}_2(t) = \mathbf{C}_2\mathbf{x}(t) + \mathbf{D}_2\mathbf{u}(t) + \mathbf{D}_{22}\mathbf{w}_2(t)\\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)\end{cases} \tag{4.4}$$

$$\Sigma_K : \left\{\mathbf{w}_1(t) = \Delta\boldsymbol{\delta}\mathbf{z}_1(t), \right. \tag{4.5}$$

where $\mathbf{z}_1(t) \in \mathbb{R}^{p_1\times 1}$, $\mathbf{w}_1(t) \in \mathbb{R}^{m_1\times 1}$.

Our goal is to find a linear static output-feedback controller which is stabilizing to (4.1), i.e., $||T_{z_1w_1}(\mathbf{K})||_\infty < 1/\rho$ [87]. The H_∞-norm constraint can be transformed into an linear matrix inequality (LMI) condition as follows [84]:

Theorem 4.2.1: The H_∞-norm constraint $||T_{z_1w_1}(\mathbf{K})||_\infty < \gamma$ holds if and only if there exists an $\mathbf{X} = \mathbf{X}^T$ that satisfies the LMI on $\mathbf{X}$ and γ^2

$$\begin{bmatrix}\mathbf{A}_{cl}(\mathbf{K})\mathbf{X} + \mathbf{X}\mathbf{A}_{cl}(\mathbf{K})^T + \mathbf{B}_1\mathbf{B}_1^T & \mathbf{X}\mathbf{C}_{cl1}(\mathbf{K})^T\\ \mathbf{C}_{cl1}(\mathbf{K})\mathbf{X} & -\gamma^2\mathbf{I}\end{bmatrix} \prec 0$$
$$\mathbf{X} \succ 0 \tag{4.6}$$

where $\mathbf{A}_{cl}(\mathbf{K}) = \mathbf{A} - \mathbf{B}\mathbf{K}\mathbf{C}$, $\mathbf{C}_{cli}(\mathbf{K}) = \mathbf{C}_i - \mathbf{D}_i\mathbf{K}\mathbf{C}$ for $i = 1, 2$.

The mixed H_2/H_∞ control problem [84]: Given an achievable H_∞-norm bound γ, find a feedback controller $\mathbf{K}$ that solves

$$\begin{aligned}&\underset{\mathbf{K}}{\text{Minimize}}\ ||T_{z_2w_2}(\mathbf{K})||_2,\\ &\text{s.t.}||T_{z_1w_1}(\mathbf{K})||_\infty < \gamma.\\ &\mathbf{u}(t) = -\mathbf{K}\mathbf{y}(t), (4.4).\end{aligned} \tag{4.7}$$

where $T_{z_iw_i}, i, = 1, 2$ denotes the closed-loop transfer function from $\mathbf{w}_i$ to $\mathbf{z}_i$.

For simplicity and without loss of generality, we set $\mathbf{D}_{22} = \mathbf{0}$ in (4.1). Following standard robust control results, such as in [84], it can be shown that the squared H_2 norm from $\mathbf{w}_2$ to $\mathbf{z}_2$ for the system (4.4) is

$$||T_{z_2w_2}(\mathbf{K})||_2^2 = J(\mathbf{K}) := \text{trace}(\mathbf{B}_2^T\mathbf{P}\mathbf{B}_2) \tag{4.8}$$

where $\mathbf{P}$ is the solution of the Lyapunov equation

$$\mathbf{P}\mathbf{A}_{cl}(\mathbf{K}) + \mathbf{A}_{cl}(\mathbf{K})^T\mathbf{P} + \mathbf{C}_{cl2}(\mathbf{K})^T\mathbf{C}_{cl2}(\mathbf{K}) = 0, \tag{4.9}$$

We can define

$$\mathbf{Q} = (\mathbf{C}_2^1)^T\mathbf{C}_2^1 \succeq 0,\ \mathbf{R} = (\mathbf{D}_2^2)^T\mathbf{D}_2^2 \succ 0. \tag{4.10}$$

such that the objective $J(\mathbf{K})$ in (4.8) can also be written as

$$J(\mathbf{K}) = \int_{t=0}^{\infty} \left[\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\right] dt. \tag{4.11}$$

4.2.2 Sparsity-constrained mixed H_2/H_∞ control

The solution for problem (4.7) is the feedback matrix $\mathbf{K}$ that optimizes the H_2 performance while satisfying the H_∞ constraint [84]. This solution is dense, i.e., it requires a feedback link from every output to every control input, which must incur large communication cost in a cyber-physical setting. To reduce this cost, we impose a sparsity constraint on the feedback matrix [2, 92], resulting in the following sparsity-constrained mixed H_2/H_∞ problem:

$$\begin{aligned} &\min_{\mathbf{K}} \ ||T_{z_2w_2}(\mathbf{K})||_2, \\ &\text{s.t. } ||T_{z_1w_1}(\mathbf{K})||_\infty < \gamma,\ \text{card}(\mathbf{K}) \leq s. \end{aligned} \tag{4.12}$$

In this work, for simplicity, we define each nonzero entry in the feedback gain matrix as one communication link. Alternative definitions and the effect of the feedback matrix sparsity on the actual cost of communication are discussed in [2].

4.2.3 Overview of the centralized PALM algorithm

The sparsity-constrained mixed H_2/H_∞ problem (4.12) can be solved using a greedy GraSP algorithm [92], when an s-sparse initial value for $\mathbf{K}$ can be found in the feasible region. In this section, we introduce a sparsity-constrained optimization algorithm based on PALM, where sparse feasible initial value is not needed. To solve (4.12) using PALM [90], we transform it into a problem with two optimization variables, $\mathbf{K}$ and $\mathbf{F}$, where $\mathbf{K}$ is the feedback matrix that

minimizes the H_2 norm subject to the H_∞ constraint, and $\mathbf{F}$ represents the sparse feedback matrix that satisfies the cardinality constraint. The problem (4.12) can be reformulated as follows:

$$\begin{aligned} &\min_{\mathbf{K},\mathbf{F}} J(\mathbf{K}) + \frac{\rho}{2}||\mathbf{K}-\mathbf{F}||_F^2, \\ &\text{s.t. } ||T_{z_1 w_1}(\mathbf{K})||_\infty < \gamma, \\ &\quad \text{card}(\mathbf{F}) \leq s. \end{aligned} \tag{4.13}$$

where the penalty term $\rho/2||\mathbf{K}-\mathbf{F}|_F^2$ is used to regularize on the difference between $\mathbf{K}$ and $\mathbf{F}$. When the parameter ρ is chosen large enough, the penalty term can be reduced sufficiently. There are two constrained variables in (4.13). We transform (4.13) to an unconstrained optimization problem by defining the following indicator functions.

$$g(\mathbf{K}) = \begin{cases} 0, & T_\infty(\mathbf{K}) < \gamma \\ +\infty, & O.W. \end{cases} \tag{4.14}$$

$$f(\mathbf{F}) = \begin{cases} 0, & \text{card}(\mathbf{F}) \leq s \\ +\infty, & O.W. \end{cases} \tag{4.15}$$

Then, the problem (4.13) is expressed

$$\min_{\mathbf{K},\mathbf{F}} \Phi(\mathbf{K},\mathbf{F}), \tag{4.16}$$

where

$$\Phi(\mathbf{K},\mathbf{F}) = J(\mathbf{K}) + g(\mathbf{K}) + f(\mathbf{F}) + H(\mathbf{K},\mathbf{F}) \tag{4.17}$$

and

$$H(\mathbf{K},\mathbf{F}) = \frac{\rho}{2}||\mathbf{K}-\mathbf{F}||_F^2 \tag{4.18}$$

is the coupling function for $\mathbf{K}$ and $\mathbf{F}$ for the equality constraint.

The PALM algorithm proceeds by alternating the minimization on the variables $(\mathbf{K},\mathbf{F})$ through separate subproblems [56], which simplifies solving (4.12), as described below. When $\mathbf{K}$ is fixed, the optimization (4.16) reduces to minimizing the sum of a nonsmooth function $f(\mathbf{F})$ and a smooth function $H(\mathbf{K},\mathbf{F})$ of $\mathbf{F}$. From the result of proximal forward-backward splitting algorithm [56], minimizing $f + H$ can be relaxed as iteratively upper bounding the objective

and minimizing the upper bound [96], and the iteration can be written as

$$\mathbf{F}^{k+1} = \arg\min_{\mathbf{F}} \left\{ \langle \mathbf{F} - \mathbf{F}^k, \nabla_{\mathbf{F}} H(\mathbf{F}, \mathbf{K}) \rangle + \frac{t}{2} ||\mathbf{F} - \mathbf{F}^k||_F^2 + f(\mathbf{F}) \right\},$$

where $\langle , \rangle$ denotes the inner product. Minimizing the first two terms $\langle \mathbf{F} - \mathbf{F}^k, \nabla_{\mathbf{F}} H(\mathbf{F}, \mathbf{K}) \rangle + \frac{t}{2} ||\mathbf{F} - \mathbf{F}^k||_F^2$ is equivalent to minimizing the first order (linear) approximation of $H(\mathbf{K}, \mathbf{F})$ at $\mathbf{F} = \mathbf{F}^k$, regularized by a trust-region penalty near point $\mathbf{F}^k$. When $t \in (L, \infty)$ and L is the Lipschitz constant [85] (see Appendix B.3) for $\nabla_{\mathbf{F}} H(\mathbf{K}, \mathbf{F})$, the regularized linear approximation provides an upper bound on $H(\mathbf{K}, \mathbf{F})$ [96].

Eq. (**??**) can be rewritten compactly using the definition of a proximal map

$$\mathbf{F}^{k+1} \in \text{prox}_t^f \left(\mathbf{F}^k - 1/t \nabla_{\mathbf{F}} H(\mathbf{K}, \mathbf{F}^k) \right). \tag{4.19}$$

where for $\sigma : \mathbb{R}^d \to (\infty, \infty]$, a proper and lower semicontinuous function, $x \in \mathbb{R}^d$ and $t > 0$, the proximal map associated with σ at point $\mathbf{x}$ is

$$\text{prox}_t^{\sigma}(\mathbf{x}) = \arg\min_{\mathbf{u} \in \mathbb{R}^d} \left\{ \sigma(\mathbf{u}) + \frac{t}{2} ||\mathbf{u} - \mathbf{x}||^2 \right\}. \tag{4.20}$$

Similar analysis can be carried out for the minimization of (4.16) when $\mathbf{F}$ is fixed. In summary, the PALM algorithm minimizes (4.16) by alternatively finding the proximal maps:

$$\mathbf{F}^{k+1} \in \text{prox}_{a_k}^f \left(\mathbf{F}^k - 1/a_k \nabla_{\mathbf{F}} H(\mathbf{K}^k, \mathbf{F}^k) \right) \tag{4.21}$$

$$\mathbf{K}^{k+1} \in \text{prox}_{b_k}^{J+g} \left(\mathbf{K}^k - 1/b_k \nabla_{\mathbf{K}} H(\mathbf{K}^k, \mathbf{F}^{k+1}) \right). \tag{4.22}$$

where a_k and b_k are positive constants that are greater than the Lipschitz constants $L_1(\mathbf{K}^k)$ and $L_2(\mathbf{F}^{k+1})$ of $\nabla_{\mathbf{F}} H(\mathbf{K}^k, \mathbf{F})$ and $\nabla_{\mathbf{K}} H(\mathbf{K}, \mathbf{F}^{k+1})$, respectively.

4.2.4 Algorithm description

We summarize the PALM algorithm for sparsity-constrained mixed H_2/H_∞ control in Algorithm 4.6. In Step 2 and 3 of Algorithm 4.6, $\mathbf{F}$-*minimization* (4.21) and $\mathbf{K}$-*minimization* (4.22) are performed, respectively. In step 2, we perform iterative $\mathbf{F}$-*minimization* (4.21), rewritten as (4.20):

$$\mathbf{F}^{k+1} = \arg\min_{\mathbf{F}} \left\{ f(\mathbf{F}) + \frac{a_k}{2} ||\mathbf{F} - \mathbf{Z}^k||_F^2 \right\} \tag{4.23}$$

Algorithm 4.6 PALM algorithm for the mixed H_2/H_∞ control algorithm with sparsity constraint

Given s: sparsity constraint, γ: H_∞-norm bound.
1. **Initialization**:
$\mathbf{K}^0$: any stabilizing feedback gain with $T_\infty(\mathbf{K}^0) < \gamma$.
$\mathbf{F}^0$: any stabilizing feedback gain $\mathbf{F}^0$.
Compute $a := \gamma_1\rho$, $b := \gamma_2\rho$.
for $k = 1, 2, ...k_{\max}$ until $||\mathbf{K}^{k+1} - \mathbf{K}^k||_F < \epsilon_1$ or $||\mathbf{F}^{k+1} - \mathbf{F}^k||_F < \epsilon_2$ **do**
  // 2. **F**-minimization step
  2.1 Compute $\mathbf{Z}^k := \mathbf{F}^k - \frac{1}{a}\nabla_{\mathbf{F}} H(\mathbf{K}^k, \mathbf{F}^k)$
  2.2 Prune $\mathbf{Z}^k$: $\mathbf{F}^{k+1} := [\mathbf{Z}^k]_s$.
  // 3. **K**-minimization step
  3.1 Compute $\mathbf{X}^k := \mathbf{K}^k - \frac{1}{b}\nabla_{\mathbf{K}} H(\mathbf{K}^k, \mathbf{F}^{k+1})$.
  3.2 Update $\mathbf{K}^{k+1}$: $\mathbf{K}^{k+1} := \textsc{KproxOp}(\mathbf{K}^k, \mathbf{X}^k, b)$.
end for

where $\mathbf{Z}^k$ is the point within the parenthesis in (4.21), found in Step 2.2. It is easy to see that the partial gradients of $H(\mathbf{K}, \mathbf{F})$ (4.18) are

$$\begin{aligned} \nabla_{\mathbf{K}} H(\mathbf{K}, \mathbf{F}) &= \rho(\mathbf{K} - \mathbf{F}) \\ \nabla_{\mathbf{F}} H(\mathbf{K}, \mathbf{F}) &= \rho(\mathbf{F} - \mathbf{K}), \end{aligned} \tag{4.24}$$

From (4.24), the Lipschitz constant $L_1(\mathbf{K}^k) = \rho$, and thus the constant a_k in (4.21) and (4.23) is defined as

$$a_k = a = \gamma_1\rho \tag{4.25}$$

with $\gamma_1 > 1$.

In Step 3 of Algorithm 4.6, we perform iterative **K**-minimization.

$$\mathbf{K}^{k+1} = \arg\min_{\mathbf{K}} \left\{ J(\mathbf{K}) + g(\mathbf{K}) + \frac{b_k}{2}||\mathbf{K} - \mathbf{X}^k||_F^2 \right\}, \tag{4.26}$$

which is equivalent to (4.22), and b_k is chosen as

$$b_k = b = \gamma_2\rho, \tag{4.27}$$

with $\gamma_2 > 1$.

In the following, we present the solutions for Eq. (4.23) and (4.26) used in Steps 2 and 3 of Algorithm 4.6.

Algorithm 4.7 KproxOp: Subroutine to solve (4.29)

procedure KproxOp($\mathbf{K}^{\text{cur}}$, $\mathbf{X}^k$, b)
  while True **do**
    $\mathbf{K}^{\text{prev}} := \mathbf{K}^{\text{cur}}$
    if $||\nabla_{\mathbf{K}} h(\mathbf{K}^{\text{cur}})||_F < \epsilon_3$ **then**
      // Stationary point in the interior of the H_∞-constraint set.
      break
    end if
    // Take a gradient-descent step in the interior of the H_∞-constraint set.
    $\mathbf{K}^{\text{cur}} := \mathbf{K}^{\text{prev}} - d\nabla_{\mathbf{K}} h(\mathbf{K}^{\text{prev}})$, where step size $d > 0$ is chosen by backtracking line search [97] s.t. $T_\infty(\mathbf{K}^{\text{cur}}) < \gamma$
    if $d < \epsilon_2$ **then**
      // $\mathbf{K}^{\text{prev}}$ is near the boundary of the H_∞-constraint set
      Solve for $\mathbf{K}^{\text{in}}$ using (4.34). Let $\Delta\mathbf{K}^{\text{cur}} := \mathbf{K}^{\text{in}} - \mathbf{K}^{\text{prev}}$.
      $\mathbf{K}^{\text{cur}} := \mathbf{K}^{\text{prev}} + d'\Delta\mathbf{K}^{\text{cur}}$, where d' is determined by backtracking line search [97].
    end if
    if $||\mathbf{K}^{\text{cur}} - \mathbf{K}^{\text{prev}}||_F < \epsilon_1$ **then**
      break
    end if
  end while
end procedure

F-minimization

Applying the proximal operator (4.21) of function f is equivalent to minimizing a regularized version of f. In (4.23), f is an indicator function of the set $\mathcal{X} = \{\mathbf{F}|\text{card}(\mathbf{K}) \leq s\}$ (4.15), so the proximal operator in (4.23) (Step 2.3 in Algorithm 4.6) is equivalent to the projection of Z^k onto the set $\mathcal{X}$, i.e., we can rewrite (4.23) as

$$\begin{aligned} \mathbf{F}^{k+1} = \arg\min_{\mathbf{F}} \ & ||\mathbf{F} - \mathbf{Z}^k||_F^2 \\ \text{s.t.} \ & \text{card}(\mathbf{F}) \leq s. \end{aligned} \tag{4.28}$$

As shown in [56,90], the solution to (4.28) is $[\mathbf{Z}^k]_s$ (see Table 4.1), which is Step 2.3 af Algorithm 4.6.

K-minimization

Next we focus on the proximal operator for (4.26), which is equivalent to

$$\min_{\mathbf{K}} \ h(\mathbf{K})$$

$$\text{s.t. } T_\infty(\mathbf{K}) < \gamma. \tag{4.29}$$

where

$$h(\mathbf{K}) \triangleq \left(J(\mathbf{K}) + \frac{b_k}{2} ||\mathbf{K} - \mathbf{X}^k||_F^2 \right) \tag{4.30}$$

We propose to solve (4.29) using a feasible direction method in the gain space of $\mathbf{K}$, summarized in Algorithm 4.7. Starting from an interior point of the feasible region of the problem (In step 3.2 of Algorithm 4.6, $\mathbf{K}^k$ always satisfies $T_\infty(\mathbf{K}^k) < \gamma$), the algorithm first descends along the gradient of $h(\mathbf{K})$ until the solution reaches a stationary point in the interior (line 6) or on the boundary of the constraint set. When the solution is in the interior of the feasible region, a gradient-descent update step is used (line 9). When the current solution is at the boundary and the gradient-descent direction violates the H_∞-norm constraint, we seek a direction that reduces the minimization objective and simultaneously moves the solution away from the boundary of the H_∞-norm constraint set to its interior (lines 12-13 in Algorithm 4.7).

In lines 12-13, we find the improving feasible direction for (4.29) when the solution is at the boundary of the feasible region. We recall the Zoutendijk's method [85] as the foundation for general feasible direction methods, which requires evaluating the gradients of both the objective and the constraints functions, i.e., $\nabla_{\mathbf{K}} h(\mathbf{K})$ and $\nabla_{\mathbf{K}} T_\infty(\mathbf{K})$. In the original formulation of Zoutendijk's method (Appendix B.2), the gradient of the constraint function is evaluated in order to form conditions for the improving feasible direction. However, due to the difficulty in evaluating the gradient of an H_∞ norm, we utilize the concept of level sets as in [86], as well as their LMI interpretation, to develop an alternative condition.

In each step of the Zoutendijk's method, a linear programming subproblem is solved to find the improving feasible direction. The inequality $\text{trace}[(\nabla_{\mathbf{K}} h(\mathbf{K}))^T \cdot \Delta\mathbf{K}] < 0$ guarantees that an update direction $\Delta\mathbf{K}$ decreases $h(\mathbf{K})$ in (4.29). Moreover, the inequality which involves gradient of the H_∞ norm, $\text{trace}[(\nabla_{\mathbf{K}} ||T_{z_1 w_1}(\mathbf{K})||_\infty)^T \cdot \Delta\mathbf{K}] < 0$ can be used to check if $\Delta\mathbf{K}$ moves away from the H_∞ bound. However, since it is difficult to compute the gradient $\nabla_{\mathbf{K}}$ of the H_∞ norm with respect to $\mathbf{K}$, we choose another condition for reduction of H_∞ norm by employing the descent method for the H_∞ norm developed in [86]. The idea of the optimization in [86] is based on level sets. In the gain space of $\mathbf{K} \in \mathbb{R}^{m \times n}$, the set of all stabilizing $\mathbf{K}$ which satisfy (4.6), i.e., with H_∞ norm smaller than γ, is a level set

$$\mathcal{K}(\gamma) := \{\mathbf{K} |\ ||T_{z_1 w_1}(\mathbf{K})||_\infty < \gamma\}. \tag{4.31}$$

Given a stabilizing gain $\mathbf{K}^0$, the algorithm in [86] proceeds by first finding a sufficiently small γ_0 such that $\mathbf{K}^0 \in \mathcal{K}(\gamma_0)$. Next, a convex subset $\hat{\mathcal{K}}(\gamma_0)$ of $\mathcal{K}(\gamma_0)$, which also contains $\mathbf{K}^0$ near the boundary, can be formed using an LMI sufficient condition. Then an inner point $\mathbf{K}^{\text{in}}$ of $\hat{\mathcal{K}}(\gamma_0)$ is found using the following sufficient LMI condition:

For the above $\mathbf{K}^0$, γ_0, the $\mathbf{K}^{\text{in}}$ is an inner point of $\hat{\mathcal{K}}(\gamma_0)$ if the matrix function $\mathbf{G}(\mathbf{K}^{\text{in}};\mathbf{K}^0)$ is positive definite, i.e.,

$$\mathbf{G}(\mathbf{K}^{\text{in}};\mathbf{K}^0) \succ 0 \Rightarrow \mathbf{K}^{\text{in}} \in \hat{\mathcal{K}}(\gamma_0). \tag{4.32}$$

where $\mathbf{G}(\mathbf{K}^{\text{in}};\mathbf{K}^0)$ is a matrix function of $\mathbf{K}^{\text{in}}$, $\mathbf{K}^0$ and γ_0. The details of computing $\mathbf{G}(\mathbf{K}^{\text{in}};\mathbf{K}^0)$ are provided in [92].

We combine the LMI condition (4.32) and the gradient of $h(\mathbf{K})$ in (4.29) to form the iterative algorithm to solve (4.29). The gradient of $h(\mathbf{K})$ is

$$\nabla_{\mathbf{K}} h(\mathbf{K}) = 2(\mathbf{RKC} - \mathbf{B}^T\mathbf{P})\mathbf{LC}^T + b_k(\mathbf{K} - \mathbf{X}^k) \tag{4.33}$$

Thus, given a current solution $\mathbf{K}^{\text{cur}}$ near the boundary of H_∞-norm constraint set, an improving feasible point $\mathbf{K}^{\text{in}}$ can be found by solving the following linear matrix inequality:

$$\begin{aligned} \max_{z,\mathbf{K}^{\text{in}}} \quad & z \\ \text{s.t.} \quad & \text{trace}[(\nabla_{\mathbf{K}} h(\mathbf{K}^{\text{cur}}))^T(\mathbf{K}^{\text{in}} - \mathbf{K}^{\text{cur}})] + z \le 0 \\ & \mathbf{G}(\mathbf{K}^{\text{in}};\mathbf{K}^{\text{cur}}) - \theta z \cdot \mathbf{I} \succeq 0 \end{aligned} \tag{4.34}$$

The parameter $\theta \ge 0$ is a predetermined factor that controls how far $\mathbf{K}$ moves away from the H_∞-norm boundary. The value of θ determines the speed of reduction of the H_∞ norm, with a small value of θ resulting in a less aggressive shrinkage of the H_∞ norm. If the solution z^* in (4.34) is positive, then $\mathbf{K}^{in} - \mathbf{K}^{\text{cur}}$ is an improving feasible direction, otherwise an improving feasible direction cannot be found.

Given the current solution $\mathbf{K}^{\text{cur}}$ and the inner point $\mathbf{K}^{\text{in}}$ solved in (4.34), the update rule is given in lines 12–13 of Algorithm 4.7, where $d' \le 1$ is the step size found by a backtracking line search using the Armijo condition [85].

4.3 Sparsity-constrained noncooperative games for multi-agent control

4.3.1 Multi-agent system model and generalized Nash equilibrium

Next, we extend the optimization in section 4.2 to the case when the agents have different optimization objectives. To accommodate this scenario, we consider the following multi-agent system with model uncertainty in the system state matrix and the control matrix. Assume there are N agents, and agent i employs its control strategy $\mathbf{u}_i(t) \in \mathbb{R}^{q_i \times 1}$, $i = 1, ..., N$. Thus, (4.1)

becomes

$$\begin{aligned}
\dot{\mathbf{x}}(t) &= (\mathbf{A}+\Delta\mathbf{A})\mathbf{x}(t) + \sum_{i=1}^{N} \left(\mathbf{B}_{(i)}+\Delta\mathbf{B}_{(i)}\right)\mathbf{u}_i(t) + \mathbf{B}_2\mathbf{w}_2(t) \\
\mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \\
\mathbf{u}_i(t) &= -\mathbf{K}_i\mathbf{y}(t),\ i = 1, ..., N.
\end{aligned} \tag{4.35}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B}_{(i)} \in \mathbb{R}^{n\times q_i}$ represents the nominal values of the state and control matrix, respectively, for the i-th control input. We assume all agents know $\mathbf{A}$ and $\mathbf{B}_{(i)}$ for $i = 1, ..., N$, and the uncertain matrices $\Delta\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\Delta\mathbf{B} \triangleq [\Delta\mathbf{B}_{(1)}, \Delta\mathbf{B}_{(2)}, ..., \Delta\mathbf{B}_{(N)}]$ satisfy the norm-bounded assumption (4.2), where $\Delta\mathbf{B}_{(i)} \in \mathbb{R}^{q_i\times n}$. Using linear static feedback, the control strategy for agent i is $\mathbf{u}_i$ or $\mathbf{K}_i$. Note that in the multi-agent system, $\mathbf{B}_{(i)}$ is the column block of $\mathbf{B}$ in (4.1), with $\sum_{i=1}^{N} \mathbf{B}_{(i)}\mathbf{u}_i = \mathbf{B}\mathbf{u}$, and $\mathbf{K}_i \in \mathbb{R}^{q_i\times p}$ is the row block of $\mathbf{K}$ in (4.7) associated with the rows corresponding to the control inputs for agent i. Thus, the multi-agent system can be expressed in the form (4.4–4.5), with the first equation in (4.4) replaced by

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \sum_{i=1}^{N} \mathbf{B}_{(i)}\mathbf{u}_i(t) + \mathbf{B}_1\mathbf{w}_1(t) + \mathbf{B}_2\mathbf{w}_2(t) \tag{4.36}$$

We introduce the following notation. Let $\mathbf{K}_{-i}$ denote the set of strategies $j \neq i, j = 1, ..., N$. When agent i chooses its strategy $\mathbf{K}_i$ in (4.35) given $\mathbf{K}_{-i}$, we refer to the resulting feedback gain matrix $\mathbf{K}$ in (4.7) as $\{\mathbf{K}_i; \mathbf{K}_{-i}\}$.

In the multi-agent system (4.36), the single performance output $\mathbf{z}_2$ in (4.1) is replaced by N individual performance outputs of the agents $\mathbf{z}_{2,(i)}$. Assuming that each performance output $\mathbf{z}_{2,(i)} = \mathbf{C}_{2,(i)}\mathbf{x} + \mathbf{D}_{2,(i)}\mathbf{u}_i$ has a form that satisfies (4.3), the H_2 cost from $\mathbf{w}_2$ to agent i's performance output can equivalently be defined as the individual LQR cost of agent i:

$$\begin{aligned}
J_i(\mathbf{K}) &= \int_{t=0}^{\infty} \left[\mathbf{x}^T(t)\mathbf{Q}_i\mathbf{x}(t) + \mathbf{u}_i^T(t)\mathbf{R}_i\mathbf{u}_i(t)\right] dt \\
\text{s.t.} \quad & \mathbf{w}_1(t) = \mathbf{0}, \mathbf{w}_2(t) = \boldsymbol{\delta}(t)
\end{aligned} \tag{4.37}$$

where $\mathbf{Q}_i \in \mathbb{R}^{n\times n} \succeq 0$ and $\mathbf{R}_i \in \mathbb{R}^{q_i\times q_i} \succ 0$ are weight matrices for state and control input for agent i, respectively, and $\mathbf{w}_2(t)$ is an impulse disturbance. Moreover, similarly to the centralized case, for stabilization of (4.35), the joint control strategy $\mathbf{K}$ needs to satisfy (4.6).

In addition, we are interested in implementing a sparse controller subject to a global sparsity constraint. In the following, we develop a noncooperative game where agent i is modeled as a game player, with its strategy given by the control policy represented by $\mathbf{K}_i$. The joint strategies $\{\mathbf{K}_1, \mathbf{K}_2, ..., \mathbf{K}_N\}$ must guarantee stability of the uncertain system with at most s

communication links in total. Thus, the set of admissible strategies $\{\mathbf{K}_1, \mathbf{K}_2, ..., \mathbf{K}_N\}$ must satisfy $||T_{z_1 w_1}(\{\mathbf{K}_1, \mathbf{K}_2, ..., \mathbf{K}_N\})||_\infty < \gamma\}$ and $\text{card}(\{\mathbf{K}_1, \mathbf{K}_2, ..., \mathbf{K}_N\}) \leq s$, and the set of feasible strategies for player i, given other players' strategies $\mathbf{K}_{-i}$, must satisfy

$$\begin{aligned}\mathcal{G}_i(\mathbf{K}_{-i}) = \{\mathbf{K}_i | &\text{card}(\{\mathbf{K}_i; \mathbf{K}_{-i}\}) \leq s, \\ &||T_{z_1 w_1}(\{\mathbf{K}_i; \mathbf{K}_{-i}\})||_\infty < \gamma\},\end{aligned} \tag{4.38}$$

Given $\mathbf{K}_{-i}$, the player i solves the following optimization:

$$\begin{aligned}&\min_{\mathbf{K}_i} J_i(\{\mathbf{K}_i; \mathbf{K}_{-i}\}) \\ &\text{s.t.} \mathbf{K}_i \in \mathcal{G}_i(\mathbf{K}_{-i}).\end{aligned} \tag{4.39}$$

The set of strategies $(\mathbf{K}_1^*, \mathbf{K}_2^*, ..., \mathbf{K}_N^*)$ is the Generalized Nash Equilibrium (GNE) [95] if

$$\begin{aligned}J_i(\{\mathbf{K}_i^*; \mathbf{K}_{-i}^*\}) \leq J_i(\{\mathbf{K}_i; \mathbf{K}_{-i}^*\}),\ \forall \mathbf{K}_i \in \mathcal{G}_i(\mathbf{K}_{-i}^*), \\ i = 1, ..., N.\end{aligned} \tag{4.40}$$

In GNE, no user can unitarily deviate from the equilibrium to improve his utility given that the strategy satisfies the global constraint [95]. A GNE differs from Nash equilibrium (NE) due to the presence of global constraints.

4.3.2 PALM algorithm for GNE

We propose to solve the generalized Nash strategies (4.40) using best-response dynamic (4.39) as described in Algorithm 4.8. Recall Algorithm 4.6, where the tuple $\mathbf{K}, \mathbf{F}$ was iteratively optimized to solve the penalized optimization (4.16). Similarly, given $\mathbf{K}_{-i}$, player i's optimization (4.39) can be written in the penalized form using indicator functions

$$\min_{\mathbf{K}_i, \mathbf{F}} \Phi_i(\mathbf{K}_i, \mathbf{F}; \mathbf{K}_{-i}) \tag{4.41}$$

with

$$\begin{aligned}\Phi_i(\mathbf{K}_i, \mathbf{F}; \mathbf{K}_{-i}) \triangleq \quad &J_i(\{\mathbf{K}_i; \mathbf{K}_{-i}\}) + h(\{\mathbf{K}_i; \mathbf{K}_{-i}\}) \\ &+ f(\mathbf{F}) + H(\{\mathbf{K}_i; \mathbf{K}_{-i}\}, \mathbf{F}).\end{aligned} \tag{4.42}$$

where the indicator functions $h(\cdot)$ and $f(\cdot)$ are given by (4.14,4.15), and the matrix $\{\mathbf{K}_i; \mathbf{K}_{-i}\}$ is defined after (4.36) . In this optimization, $\mathbf{K}_i \in \mathbb{R}^{q_i \times n}$ is viewed as the feedback gain of agent i that satisfies $||T_{z_1 w_1}(\{\mathbf{K}_i; \mathbf{K}_{-i}\})||_\infty < \gamma$, and $\mathbf{F} \in \mathrm{R}^{m \times n}$ represents the system-wide sparse

feedback gain matrix that satisfies the global sparsity constraint. In the function $\Phi(\mathbf{K}, \mathbf{F})$ (4.16), the variables $\mathbf{K}$ and $\mathbf{F}$ were of the same size, and they represented the same global sparsity-constrained feedback gain. However, when minimizing $\Phi_i(\mathbf{K}_i, \mathbf{F}; \mathbf{K}_{-i})$ (4.41), the variable $\mathbf{K}_i$ is the robust feedback gain for player i while $\mathbf{F}$ represents the global feedback gain that satisfies the sparsity constraint. In the best-response dynamic, in each round the players take turns to minimize Φ_i functions over $\mathbf{K}_i$ and $\mathbf{F}$. The equilibrium point is achieved when no player can improve its Φ_i using $\mathbf{K}_i$ and $\mathbf{F}$ while $\mathbf{K}_j$ is fixed for $j \neq i$. Note that in the initial best response update steps, given non-sparse $\mathbf{K}_{-i}$, the minimization objective (4.41) cannot drive the coupling function $H(\{\mathbf{K}_i; \mathbf{K}_{-i}\}, \mathbf{F})$ (4.18) to zero, since $\sum_{i \neq j} ||\mathbf{K}_i - (\mathbf{F})_i||_F^2 \approx 0$ only when $\mathbf{K}_{-i}$ approaches the desired level of sparsity, where $(\mathbf{F})_i \in \mathbb{R}^{q_i \times n}$ denotes the row block of $\mathbf{F}$ which corresponds to the feedback gain of the i's player.

The minimization of (4.41) is similar to the minimization of (4.16). Thus, modified Algorithm 4.6 is used in line 9 of Algorithm 4.8 to solve (4.41). Given its $\mathbf{K}_i^l$, $\mathbf{F}^l$ at iteration l, the following proximal operators are performed by player i in the minimization of line 9 of Algorithm 4.8.

F-minimization:

Compute the proximal point $\mathbf{Z}^k$ for $\mathbf{F}^k$:

$$\begin{aligned} \mathbf{Z}^k &= \mathbf{F}^k - \frac{1}{a} \nabla_{\mathbf{F}} H(\mathbf{K}^k, \mathbf{F}^k) \\ &= \mathbf{F}^k - \frac{\gamma}{a} (\mathbf{F}^k - \{\mathbf{K}_i^k; \mathbf{K}_{-i}\}) \end{aligned} \tag{4.43}$$

Algorithm 4.8 PALM algorithm for computing GNE (4.40)

1: **Given** s: global sparsity constraint, γ: H_∞-norm bound.
2: **Initialization**:
3: $\mathbf{K}^0$: any stabilizing feedback gain with $T_\infty(\mathbf{K}^0) < \gamma$.
4: $\mathbf{F}^0$: any stabilizing feedback gain $\mathbf{F}^0$.
5: **for** $l = 1...l_{\max}$ until $||\mathbf{F}^1 - \mathbf{F}^{l-1}||_F < \epsilon_3$ **do**
6: $\quad \mathbf{K}^l := \mathbf{K}^{l-1}$, $\mathbf{F}^l := \mathbf{F}^{l-1}$
7: $\quad$ **for** $i = 1...N$ **do**
8: $\quad\quad$ // Solve using (4.43–4.46) with $\mathbf{K}_i^l$, $\mathbf{F}^l$ as the initial values:
9: $\quad\quad \hat{\mathbf{K}}_i, \hat{\mathbf{F}} = \arg\min_{\mathbf{K}_i, \mathbf{F}} \Phi_i(\mathbf{K}_i, \mathbf{F}; \mathbf{K}_{-i}^l)$
10: $\quad\quad$ // Update $\mathbf{K}^l$ and $\mathbf{F}^l$:
11: $\quad\quad \mathbf{K}^l = \{\hat{\mathbf{K}}_i; \mathbf{K}_{-i}^l\}$
12: $\quad\quad \mathbf{F}^l = \hat{\mathbf{F}}$
13: $\quad$ **end for**
14: **end for**
15: **Output:** $\mathbf{K}^{\mathrm{GNE}}(s) := \mathbf{F}^l$.

Solve the proximal operator:

$$\begin{aligned}\mathbf{F}^{k+1} = \quad & \arg\min_{\mathbf{F}} \frac{a}{2}||\mathbf{F}-\mathbf{Z}^k||_F^2 \\ & \text{s.t. } \text{card}(\mathbf{F}) \leq s,\end{aligned} \tag{4.44}$$

and get $\mathbf{F}^{k+1} = [\mathbf{Z}^k]_s$, similarly to Step 2 of Algorithm 4.6.

K-minimization:

Compute proximal point $\mathbf{X}_i^k$ for $\mathbf{K}_i$:

$$\begin{aligned}\mathbf{X}_i^k &= \mathbf{K}_i^k - \frac{1}{b}\nabla_{\mathbf{K}_i} H(\mathbf{K}_i^k - (\mathbf{F}^{k+1})_i) \\ &= \mathbf{K}_i^k - \frac{\rho}{b}(\mathbf{K}_i^k - (\mathbf{F}^{k+1})_i).\end{aligned} \tag{4.45}$$

Solve the proximal operator:

$$\begin{aligned}\mathbf{K}_i^{k+1} = \quad & \arg\min_{\mathbf{K}_i} \left\{ J_i(\{\mathbf{K}_i;\mathbf{K}_{-i}\}) + \frac{b}{2}||\mathbf{K}_i - \mathbf{X}_i^k||_F^2 \right\} \\ & \text{s.t. } ||T_{z_1 w_1}(\{\mathbf{K}_i;\mathbf{K}_{-i}\})||_\infty < \gamma.\end{aligned} \tag{4.46}$$

Solving (4.46) is similar to solving (4.29). In (4.46), player i aims to update its control strategy $\mathbf{K}_i$ given other players' strategies $\mathbf{K}_{-i}$. Algorithm 4.7 is applied to solve (4.46) with several modifications. The minimization cost in (4.46) is defined as $h_i(\mathbf{K}_i,\mathbf{K}_{-i}) \triangleq J_i(\{\mathbf{K}_i;\mathbf{K}_{-i}\}) + \frac{b}{2}||\mathbf{K}_i - \mathbf{X}_i^k||_F^2$, and the gradient with respect to $\mathbf{K}_i$ in line 4 of Algorithm 4.7 is replaced by

$$\begin{aligned}&\nabla_{\mathbf{K}_i} h_i(\mathbf{K}_i,\mathbf{K}_{-i}) = \\ &2(\mathbf{R}_i\mathbf{K}_i\mathbf{C} - \mathbf{B}_{(i)}^T\mathbf{P}(\mathbf{K}_i,\mathbf{K}_{-i}))\mathbf{L}(\mathbf{K}_i,\mathbf{K}_{-i})\mathbf{C}^T + b(\mathbf{K}_i - \mathbf{X}_i^k)\end{aligned} \tag{4.47}$$

where $L(\mathbf{K}_i,\mathbf{K}_{-i})$ and $P(\mathbf{K}_i,\mathbf{K}_{-i})$ are the solution of the following set of equations:

$$\begin{aligned}&(\bar{\mathbf{A}}_{cl}(\mathbf{K}_i,\mathbf{K}_{-i}))^{\mathrm{T}}\boldsymbol{P}(\mathbf{K}_i,\mathbf{K}_{-i}) + \boldsymbol{P}(\mathbf{K}_i,\mathbf{K}_{-i})\bar{\mathbf{A}}_{cl}(\mathbf{K}_i,\mathbf{K}_{-i}) + \bar{\mathbf{Q}}_i(\mathbf{K}_i,\mathbf{K}_{-i}) = 0 \\ &\bar{\mathbf{A}}_{cl}(\mathbf{K}_i,\mathbf{K}_{-i})\boldsymbol{L}(\mathbf{K}_i,\mathbf{K}_{-i}) + \boldsymbol{L}(\mathbf{K}_i,\mathbf{K}_{-i})(\bar{\mathbf{A}}_{cl}(\mathbf{K}_i,\mathbf{K}_{-i}))^T + \boldsymbol{B}_2\boldsymbol{B}_2^{\mathrm{T}} = 0\end{aligned} \tag{4.48}$$

and

$$\begin{aligned}&\bar{\mathbf{A}}_{cl}(\mathbf{K}_i,\mathbf{K}_{-i}) = \mathbf{A} - \sum_{j\neq i}\mathbf{B}_{(j)}\mathbf{K}_j\mathbf{C} - \boldsymbol{B}_{(i)}\boldsymbol{K}_i\mathbf{C} \\ &\bar{\mathbf{Q}}_i(\mathbf{K}_i,\mathbf{K}_{-i}) = \mathbf{Q}_i + \mathbf{C}^T(\sum_{j\neq i}(\mathbf{K}_j)^T\mathbf{R}_j\mathbf{K}_j + \mathbf{K}_i^T\boldsymbol{R}_i\boldsymbol{K}_i)\mathbf{C}.\end{aligned} \tag{4.49}$$

where $\mathbf{Q}_i$ and $\mathbf{R}_i$ are defined in (4.37). Similarly to lines 10–14 in Algorithm 4.7, when player i's strategy $\mathbf{K}_i^{\text{cur}}$ is near the boundary of the H_∞-norm constraint given other players' strategies $\mathbf{K}_{-i}$, an improving feasible direction for $\mathbf{K}_i$ can be found by solving an LMI like (4.34) for scalar z and $\mathbf{K}_i^{in} \in \mathbb{R}^{q_i \times n}$.

$$\begin{aligned} &\underset{z,\mathbf{K}_i^{\text{in}}}{\text{Maximize}}\ z \\ &\text{s.t. trace}[(\nabla_{\mathbf{K}_i} h_i(\mathbf{K}_i^{\text{cur}}, \mathbf{K}_{-i})^T(\mathbf{K}_i^{\text{in}} - \mathbf{K}_i^{\text{cur}})] + z \leq 0 \\ &\qquad \mathbf{G}\left(\{\mathbf{K}_i^{\text{in}}; \mathbf{K}_{-i}\}; \{\mathbf{K}_i^{\text{cur}}; \mathbf{K}_{-i}\}\right) - \theta z \cdot \mathbf{I} \succeq 0 \end{aligned} \tag{4.50}$$

where θ is the factor to control the speed of reduction of H_∞ norm, and G is defined in (4.32). Then the update direction of $\mathbf{K}_i$ can be formed: $\Delta\mathbf{K}_i = \mathbf{K}_i^{\text{in}} - \mathbf{K}_i^{\text{cur}}$. We note that in **K**-minimization step, each player updates its own strategy $\mathbf{K}_i$, while in **F**-minimization step, the strategies of all the players are jointly updated. Thus, Algorithm 4.8 has partially distributed computation.

Finally, we note that the centralized problem (4.12) can be represented as a potential game by modifying the noncooperative game (4.40). A game $\{N, \{\mathcal{A}_i\}, \{J_i\}\}$ with N players, action set $\{\mathcal{A}_i\}_{i=1}^N$ and utilities $\{J_i\}_{i=1}^N$, is an exact potential game [36] if there exists a global function Φ, such that for every player $i \in N$, $a_{-i} \in \mathcal{A}_{-i}$ and $a_i', a_i'' \in \mathcal{A}_i$,

$$J_i(a_i', a_{-i}) - J_i(a_i'', a_{-i}) = \Phi(a_i', a_{-i}) - \Phi(a_i'', a_{-i}). \tag{4.51}$$

We employ a common assumption that the input penalty of each user is uncorrelated, i.e., $\mathbf{u}^T\mathbf{R}\mathbf{u} = \sum_{i=1}^N \mathbf{u}_i^T \mathrm{R}_{(i)} \mathbf{u}_i$, where $\mathbf{R}_{(i)}$ is the submatrix of $\mathbf{R}$ that represents the weight matrix for $\mathbf{u}_i(t)$. Thus, the objective $J(\mathbf{K})$ in (4.11) can be expressed as

$$J(\mathbf{K}) = J(\{\mathbf{K}_i, \mathbf{K}_{-i}\}) = \int_0^\infty \left[\mathbf{x}^T \left(\mathbf{Q} + \mathbf{C}^T (\sum_{j \neq i} \mathbf{K}_j^T \mathbf{R}_{(j)} \mathbf{K}_j) \mathbf{C} \right) \mathbf{x} + \mathbf{u}_i^T \mathbf{R}_{(i)} \mathbf{u}_i \right] dt,$$

with $\mathbf{u}_i = -\mathbf{K}_i \mathbf{y}$. Thus, the minimization objectives $J_i(\{\mathbf{K}_i; \mathbf{K}_{-i}\})$ of all players in (4.39) are replaced by the global LQR cost (4.52). To convert the game in (4.39) into an exact potential game, we set $\mathbf{Q}_i = \mathbf{Q} + \mathbf{C}^T(\sum_{j \neq i} \mathbf{K}_j^T \mathbf{R}_{(j)} \mathbf{K}_j)\mathbf{C}$, $\mathbf{R}_i = \mathbf{R}_{(i)}$ in the individual cost (4.37), which is consistent with (4.52). GNE strategies $\mathbf{K}_1^*, \mathbf{K}_2^*, ..., \mathbf{K}_N^*$ for the potential game are

$$\begin{aligned} &J(\{\mathbf{K}_i^*; \mathbf{K}_{-i}^*\}) \leq J(\{\mathbf{K}_i; \mathbf{K}_{-i}^*\}), \\ &\forall \mathbf{K}_i \in \mathcal{G}_i(\mathbf{K}_{-i}^*), i = 1, ..., N. \end{aligned} \tag{4.52}$$

We employ Algorithm 4.8 to compute (4.52), where players update their control strategies in

K-minimization step distributedly, and jointly update their strategies in **F**-minimization step, thus obtaining a partially distributed implementation of the centralized sparsity-constrained problem (4.12).

4.4 Numerical results and convergence analysis

4.4.1 Experiment setup

Consider a network that consists of N connected nodes distributed randomly on a L unit by L unit square area. Each node is an unstable second-order system coupled with other nodes through an exponentially decaying function of the Euclidean distance $\hat{l}(i,j)$ [81, 88]. The state-space representation of node i is given:

$$\begin{bmatrix} \dot{x}_{1i} \\ \dot{x}_{2i} \end{bmatrix} = \hat{\mathbf{A}}_{ii} \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} + \sum_{j \neq i} e^{-\hat{l}(i,j)} \begin{bmatrix} x_{1j} \\ x_{2j} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (d_i + u_i). \tag{4.53}$$

In the above state-space representation, the state matrix $\hat{\mathbf{A}}_{ii},\ i = 1, ..., N$ and the Euclidean distances $\hat{l}(i,j)$ are not known exactly, and

$$\begin{aligned} \hat{\mathbf{A}}_{ii} &= \mathbf{A}_{ii} + \mathbf{A}_{ii} \odot \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix} \\ \hat{l}(i,j) &= l(i,j) \cdot (1 + \delta_{i,j}), \end{aligned} \tag{4.54}$$

where $\mathbf{A}_{ii}$ and $l(i,j)$ are the nominal values, and δ_{ij} and θ_{ij} are independent random perturbations, uniformly distributed in the range $\pm 20\%$. The operator $\odot$ denotes element-wise multiplication. As in (4.1), $\mathbf{A}$ denotes the nominal value of the state matrix of this N-node unstable system, and $\hat{\mathbf{A}}$ denotes one realization of the perturbed state matrix. The uncertain matrix $\Delta\mathbf{A} = \hat{\mathbf{A}} - \mathbf{A}$ in (4.1), $\hat{\mathbf{B}} = \mathbf{B} = \mathbf{1}_N \otimes \mathbf{B}_{ii}$ since the control matrix is exactly known, and $\mathbf{B}_{ii} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$, $\otimes$ denotes the Kronecker product [98]. In the simulation, we collected 200 random samples of $\hat{\mathbf{A}}$. To guarantee closed-loop stability of the system in (4.53), according to (4.2), we numerically compute the worst-case $\hat{\boldsymbol{A}}$ as,

$$\hat{\boldsymbol{A}}_{\text{worst}} = \arg\max_{\hat{\boldsymbol{A}}} \sigma_{\max}(\hat{\boldsymbol{A}} - \boldsymbol{A}) \tag{4.55}$$

Using the singular value decomposition, we obtain $\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \hat{\boldsymbol{A}}_{\text{worst}} - \boldsymbol{A}$. Normalizing $\boldsymbol{S}$ by $\sigma_{\max}(\boldsymbol{S})$, we set $\boldsymbol{B}_1 = \sqrt{\sigma_{\max}(\boldsymbol{S})}\boldsymbol{U}$, $\boldsymbol{C}_1 = \sqrt{\sigma_{\max}(\boldsymbol{S})}\boldsymbol{V}^T$ in (4.2). Due to this normalization, $\gamma = 1$.

The following parameters are employed in the simulations. We set $L = 2$ and $N = 5$,

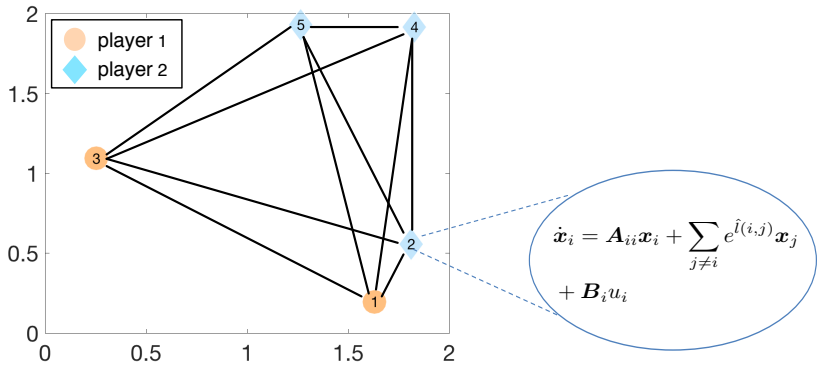


Figure 4.1: The unstable node system.

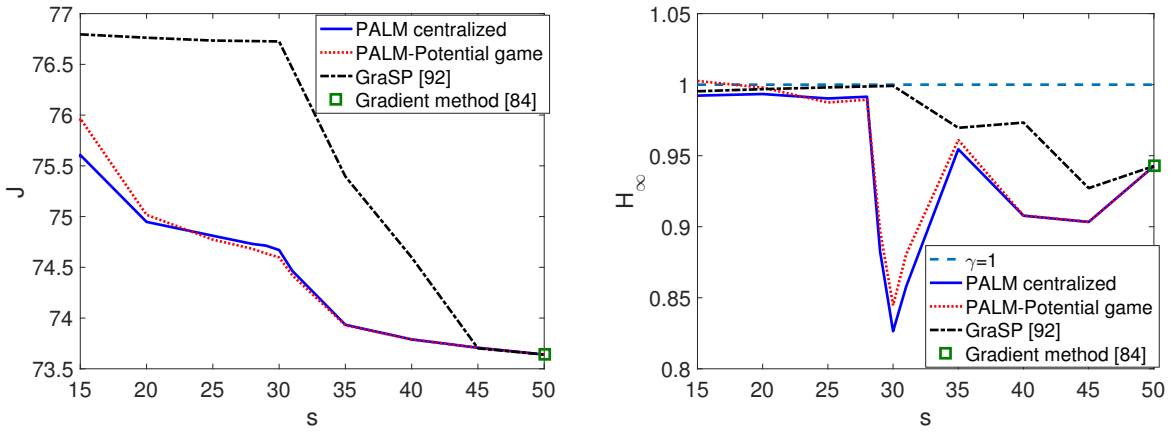
thus $\mathbf{A} \in \mathbb{R}^{10\times 10}$, $\mathbf{B} \in \mathbb{R}^{10\times 5}$. The output matrix $\mathbf{C} = \mathbf{I}_{10}$. The dense feedback matrix K has $\text{card}(\mathbf{K}) = 50$. When the feedback controller is completely decentralized, i.e., feedback links only exist between states and controllers within the same node, $\text{card}(\mathbf{K}) = 10$. The performance index for the LQR cost employs $\mathbf{Q} = 100 \cdot \mathbf{I}$ and $\mathbf{R} = \mathbf{I}$ in (4.10) for the centralized problem (4.12). For the noncooperative game (4.40), we consider a two-player game as shown in Figure 4.1, where player 1 is in charge of the control inputs in nodes 1 and 3 and player 2 is in charge of the control inputs in nodes 2, 4, 5. The performance index matrices $\mathbf{Q}_i, \mathbf{R}_i$, $i = 1, 2$ for the LQR cost in (4.37) satisfy:

$$\begin{aligned}
&\mathbf{x}^T\mathbf{Q}_1\mathbf{x}+\mathbf{u}_1^T\mathbf{R}_1\mathbf{u}_1 = 100[(x_{11}-x_{13})^2+(x_{21}-x_{23})^2]+u_1^2+u_3^2 \\
&\mathbf{x}^T\mathbf{Q}_2\mathbf{x} + \mathbf{u}_2^T\mathbf{R}_2\mathbf{u}_2 = 100 \sum_{j=2,4,5} (x_{1j}^2 + x_{2j}^2) + \sum_{j=2,4,5} u_j^2.
\end{aligned} \tag{4.56}$$

In this chapter, we solve all the LMIs using the CVX package [89].

4.4.2 Social optimization

First, we present simulation results for the problem (4.12) applied to the system in (4.53) with $\gamma = 1$ in (4.12) over a range of s-values. First, we implement Algorithm 4.6, and the resulting feedback matrix is denoted as $\mathbf{K}^*_{\text{palm}}(s)$. For the same problem (4.12), we also use Algorithm 4.8 applied to the potential game (4.52), with the solution denoted by $\mathbf{K}^*_{\text{PALMPG}}(s)$ given the sparsity-constraint s. For comparison, we also run the GraSP algorithm in [92], and the resulting feedback is denoted by $\mathbf{K}^*_{\text{GraSP}}(s)$ under the sparsity constraint s, initialized by a stabilizing decentralized controller $\mathbf{K}_{\text{dec}}$ with $\text{card}(\mathbf{K}_{\text{dec}}) = 10$. In general, the GraSP algorithm in [92] must be initialized by a $\mathbf{K}_0$ that satisfies $\text{card}(\mathbf{K}_0) \leq s$ and $T_\infty(\mathbf{K}_0) < \gamma$, which might be difficult to find. On the contrary, the PALM-based Algorithm 4.6 of this chapter does not rely



(a) J vs. sparsity constraint s.

(b) H_∞ norm v.s. sparsity constraint s.

Figure 4.2: The LQR cost J and H_∞ norm vs. sparsity constraint s.

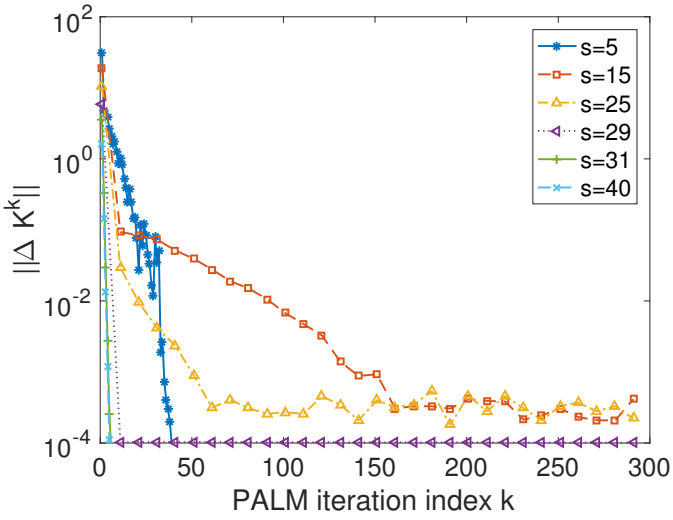


Figure 4.3: The error in $\mathbf{K}$ vs. iteration k in PALM Algorithm 4.6 Step 2 and 3 for different s-values.

on sparse initialization. Finally, we show performance of the dense mixed H_2/H_∞ controller using the simple gradient method in [84].

Figure 4.2 illustrates the optimal LQR cost J in problem (4.12) and the associated H_∞ norm vs. sparsity constraint s. For $15 \leq s \leq 50$, the centralized Algorithm 4.6 and the potential game using Algorithm 4.8 both converge to a solution with sufficiently small coupling function in (4.16), which indicates $\mathbf{F} \approx \mathbf{K}$. From Figure 4.2(a), we observe that the H_2 norms

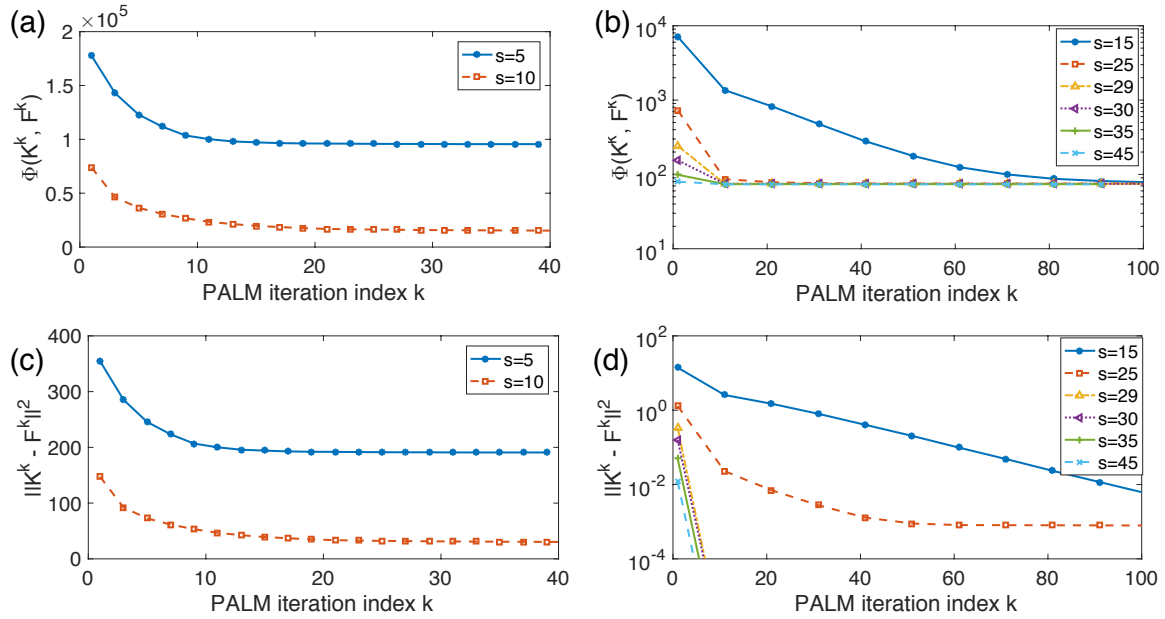


Figure 4.4: The penalized cost function $\Phi(\mathbf{K}^k, \mathbf{F}^k)$ and the coupling function $||\mathbf{K}^k - \mathbf{F}^k||_F^2$ vs iteration k in the end of Step 3 of Algorithm 4.6 for multiple s-values.

of all sparsity-constrained methods decrease as s is relaxed, and approach to that of the dense controller [8]. However, the PALM-based methods have similar LQR costs and outperform significantly the greedy GraSP algorithm in [92]. In GraSP, the choice of active coordinates only depends on the gradient information of the function J. At convergence, the solution of the mixed H_2/H_∞ problem has the sparsity structure given by the greedy selection step. For the PALM algorithm, since we iteratively compute the proximal map on $\mathbf{X}^k$ and $\mathbf{Z}^k$, the support is chosen based on the information on both the LQR cost $J(\mathbf{K})$ and the H_∞-norm constraint $T_\infty(\mathbf{K})$. Thus, at convergence, PALM method finds a critical point of problem (4.12) while GraSP does not necessarily achieve it. Figure 4.2(b) shows the H_∞ norms of $\mathbf{K}^*_{\text{PALM}}(s)$, $\mathbf{K}^*_{\text{PALMPG}}(s)$ and $\mathbf{K}^*_{\text{GraSP}}(s)$. We observe that for both GraSP and PALM methods, the solution is found in the interior of the H_∞-norm constraint for $s \geq 30$, and on the boundary for $s \leq 25$, which indicates that when the sparsity constraint becomes more stringent, satisfying the sparsity and H_∞-norm constraints simultaneously becomes challenging.

We found that both Algorithm 4.6 (the social optimization) and Algorithm 4.8 (the potential game) converge for all s-values in this system. Figure 4.3 shows the error in consecutive steps for variable $\mathbf{K}$ at the end of step 3 of Algorithm 4.6 as a function of iteration step, for different s-values. We found that $\Delta\mathbf{F}_k$ has a similar trend to $\Delta\mathbf{K}_k$. The errors in consecutive steps are defined as $\Delta\mathbf{K}^k \triangleq \mathbf{K}^k - \mathbf{K}^{k-1}$ and $\Delta\mathbf{F}^k \triangleq \mathbf{F}^k - \mathbf{F}^{k-1}$. We note that the error converges faster for larger s-values, which might be explained by the fact that that for $s > 25$, the minima

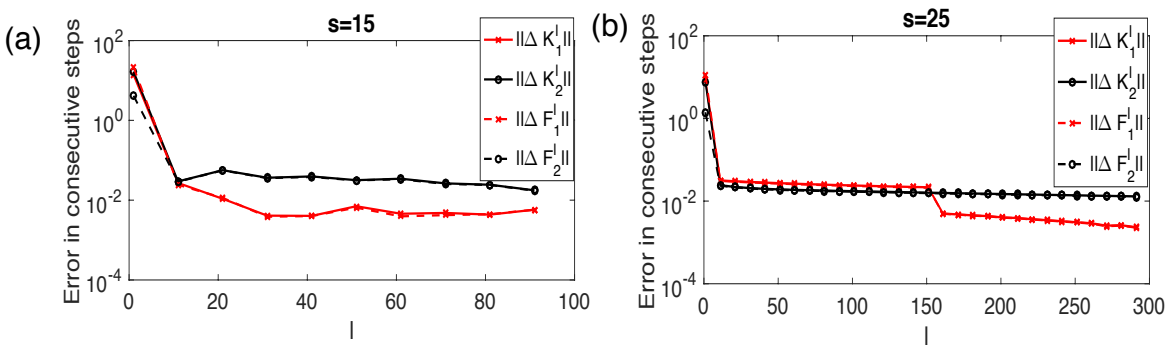
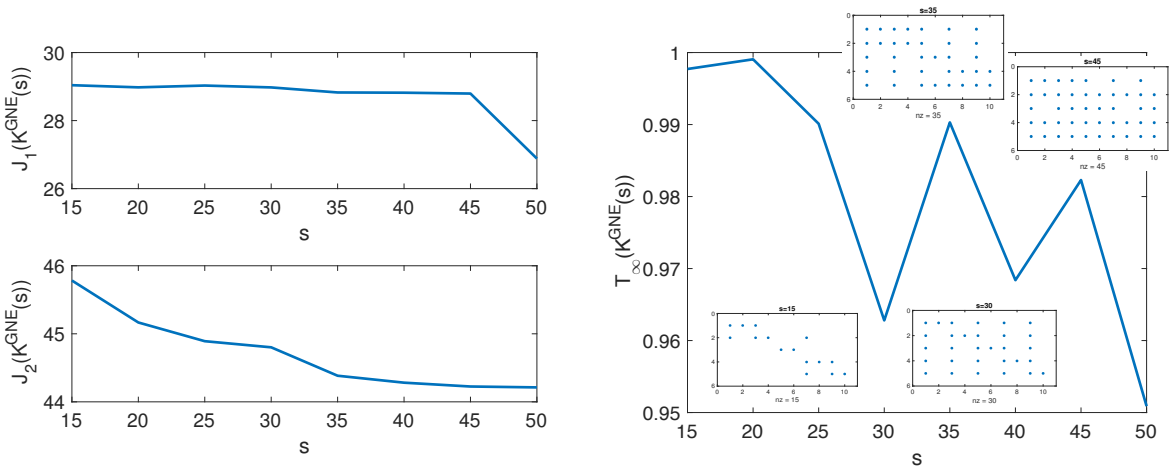


Figure 4.5: Errors in consecutive steps of $\mathbf{K}_i^l$ and $\mathbf{F}_i^l$ for players $i = 1, 2$ vs. step l in Algorithm 4.8 (the noncooperative game).

are found in the interior of the H_∞-norm constraint set (see Figure 4.2). For Algorithm 4.8 (potential game), the penalized cost function Φ_i and $||\mathbf{K} - \mathbf{F}||_F^2$ (line 9) have similar trends to those for Algorithm 4.6. Moreover, it is demonstrated in Fig 4.4 that although Algorithm 4.6 converges to a critical point of $\mathbf{\Phi}(\mathbf{K}, \mathbf{F})$, the coupling function $H(\mathbf{K}, \mathbf{F}) > 0$ for $s < 15$. As a result, when Algorithm 4.6 converges for these s-values, $\mathbf{K} \neq \mathbf{F}$, so a sparse feedback solution that satisfies (4.12) cannot be found. Thus, in Figure 4.2, we only show the LQR cost and H_∞-norm for $15 \leq s \leq 50$.

4.4.3 The noncooperative game

We investigate performance of Algorithm 4.8 for the noncooperative game with different individual costs (4.56) for the system (4.53). We use $\mathbf{K}^{\text{GNE}}(s) = \{\mathbf{K}_1^{\text{GNE}}(s), \mathbf{K}_2^{\text{GNE}}(s)\}$ to denote the two players' feedback produced by Algorithm 4.8 when the sparsity constraint is given by s. Figure 4.5 shows the errors in consecutive steps of player i's strategic variables $\mathbf{K}_i, \mathbf{F}_i$ for $i = 1, 2$ vs iteration round l in Algorithm 4.8. We observe that both $||\Delta\mathbf{K}_i||_F$ and $||\Delta\mathbf{F}_i||_F$ decrease significantly within the first 10 iterations and then saturate to small values as l grows, resulting in the saturation of the penalized cost function Φ_i in line 9 of Algorithm 4.8, which corresponds to an approximate equilibrium point as discussed in section 4.4.4. The normalized coupling function $\frac{1}{\rho}H(\mathbf{K}^l, \mathbf{F}^l) = ||\mathbf{K}^l - \mathbf{F}^l||_F^2$ (4.18) decreases with iteration l, following the trend in Figure 4.4. For $s > 20$, the square error $||\mathbf{K}^l - \mathbf{F}^l||_F^2$ reaches sufficiently small value ($< 10^{-4}$) at the equilibrium point, while for $s \leq 20$, the square error is larger, causing $T_\infty(\mathbf{K}^{\text{GNE}}(s)) > T_\infty(\mathbf{K}^l)$, which results in $T_\infty(\mathbf{K}^{\text{GNE}}(s)) > 1$ at convergence. To guarantee stabilization of the closed-loop uncertain system despite this discrepancy, we can replace γ in (4.14) with $\gamma - \epsilon$ to provide a margin that compensates for the square error between $\mathbf{K}$ and $\mathbf{F}$. In this example, we set $\epsilon = 0.01$, so that $T_\infty(\mathbf{K}^{\text{GNE}}(s)) < 0.99$.



(a) $J_i(\mathbf{K}^{\mathrm{GNE}}(s))$ vs. sparsity constraint s at GNE for $i =$ 1, 2. (b) $T_\infty(\mathbf{K}^{\mathrm{GNE}}(s))$ vs. the sparsity constraint s at GNE.

Figure 4.6: The individual LQR cost and global H_∞ norm of $\mathbf{K}^{\mathrm{GNE}}(s)$ vs sparsity constraint s at GNE, and the sparsity pattern of $\mathbf{K}^{\mathrm{GNE}}(s)$ for different s values.

Figure 4.6 illustrates the individual LQR costs J_i (4.37) and the global H_∞ norm when the feedback gains of Nash strategies in $\mathbf{K}^{\mathrm{GNE}}(s)$ are implemented. We observe that in Figure 4.6 (a), for each player i, the LQR cost achieved at the equilibrium point $J_i(\mathbf{K}^{\mathrm{GNE}}(s))$ tends to decrease with s, which indicates that there is a trade-off between the selfish LQR cost and the global shared sparsity constraint. From Figure 4.6 (b), $T_\infty(\mathbf{K}^{\mathrm{GNE}}(s)) < 1$ for $15 \leq s \leq 45$, so the Nash strategies in $\mathbf{K}^{\mathrm{GNE}}(s)$ are guaranteed to stabilize the uncertain system in (4.53).

4.4.4 Algorithm Convergence and Complexity

Global convergence of PALM algorithm for nonconvex nonsmooth functions was studied in [56]. Moreover, the global convergence property of PALM algorithm output feedback co-design problem under block-sparsity constraints has been established in [90]. Results in [56, 90] are extended to analyze the convergence properties of Algorithm 4.6 in Appendix B.5.

It has been proved in [56] that if Lemma B.5.1–B.5.3 in B.5 hold, then the sequence generated by PALM algorithm globally converges. In addition, if Lemma B.5.4 of Appendix B.5 holds, the sequence converges to a critical point [56] of Φ. This confirms convergence of Algorithm 4.6 to a sparsity-constrained mixed H_2/H_∞ controller, which corresponds to a critical point of Φ under mild assumptions on the functions J and g.

Next, we briefly discuss the convergence properties of Algorithm 4.8. When GNE (4.40) is achieved at point $(\mathbf{K}_1^*, ..., \mathbf{K}_N^*)$, then the following must hold for each player i, $i = 1, ..., N$ [99]:

$$\nabla_{\mathcal{G}_i(\mathbf{K}^*_{-i}),\eta} J_i(\{\mathbf{K}^*_i; \mathbf{K}^*_{-i}\}) = 0, \tag{4.57}$$

where $\nabla_{\mathcal{G}_i(\mathbf{K}^*_{-i}),\eta} J_i(\{\mathbf{K}_i; \mathbf{K}^*_{-i}\})$ is the projected gradient of cost J_i (4.37) onto the constraint set $\mathcal{G}_i$ (4.38) for player i, defined as [99]

$$\begin{aligned} &\nabla_{\mathcal{G}_i(\mathbf{K}^*_{-i}),\eta} J_i(\{\mathbf{K}_i; \mathbf{K}^*_{-i}\}) \triangleq \\ &\frac{1}{\eta}(\mathbf{K}_i - \Pi_{\mathcal{G}_i(\mathbf{K}^*_{-i})}[\mathbf{K}_i - \eta \nabla_{\mathbf{K}_i} J_i(\{\mathbf{K}_i, \mathbf{K}^*_{-i}\})]) \end{aligned} \tag{4.58}$$

where $\eta > 0$ and the operator $\Pi_{\mathcal{K}}(\cdot)$ denotes the projection onto set $\mathcal{K}$.

In line 9 of Algorithm 4.8, a necessary condition for Φ_i to achieve its minimum is that the projected gradient $\nabla_{\mathcal{G}_i(\mathbf{K}^l_{-i}),\eta} J_i(\{\mathbf{K}_i; \mathbf{K}^l_{-i}\}) = 0$. In Algorithm 4.8, instead of seeking an exact equilibrium point as GNE, we assume convergence when the projected gradient is sufficiently small, which corresponds to an approximate local equilibrium [99]. At iteration l at Algorithm 4.8, the $\hat{\mathbf{K}}_i$ at line 9 can be viewed as an approximation of $\Pi_{\mathcal{G}_i(\mathbf{K}^l_{-i})}[\mathbf{K}^{l-1}_i - \eta \nabla_{\mathbf{K}_i} J_i(\{\mathbf{K}^{l-1}_i, \mathbf{K}^l_{-i})\}]$. Thus, the norm of the projected gradient is proportional to $||\mathbf{K}^l_i - \mathbf{K}^{l-1}_i||$. Therefore, in Algorithm 3, small $\Delta\mathbf{K}^l_i$ and $\Delta\mathbf{F}^l_i$ indicate convergence of Algorithm 4.8, as shown in Figure 4.5. Although there is no theoretical guarantee for the existence of GNE for the game in (4.39), if a GNE exists for the potential game (4.52), this GNE satisfies the necessary condition for the minimizer of (4.12).

Finally, the complexity of Algorithms 4.6 and 4.8 is dominated by the the $\mathbf{K}$-minimization step (Step 3 of Algorithm 4.6 and line 9 of Algorithm 4.8), which has polynomial complexity on the number of variables in the feedback matrix [100].

4.5 Conclusion

The PALM method was exploited to solve the sparsity-constrained mixed H_2/H_∞ control problem for multi-agent systems. First, centralized social-optimization algorithm was investigated. Second, we developed noncooperative and potential games that have partially-distributed computation. The proposed algorithms were validated using a network of unstable nodes system. It was demonstrated that the centralized PALM method outperforms the GraSP-based method for most sparsity constraint values and converges both theoretically and in simulation results. Moreover, a best-response dynamics algorithm for proposed games converges to an approximate GNE point, and performance of the potential game for social optimization approximates closely that of the centralized algorithm.

Chapter 5

Contributions and Future Directions

In this thesis, several algorithms are discussed for sparse controller design, sparse noncooperative games and network cost allocation for centralized and multi-agent control systems under different scenarios. Our work is built upon foundations and advances in multiple areas, including recent results that bridge the gap between compressed sensing and optimal control [6–8, 10, 13, 22, 62, 81], game theory [101], especially differential games [38], H_∞ control theory [87], and LMI [102]. Compared to earlier works, this thesis considers the communication cost in the control network as a constraint on the controller design and investigates the game in the multi-agent control system under this constraint. We summarize our contributions as follows:

1. Development of GraSP-based algorithms for centralized social optimization and noncooperative linear-quadratic games for multi-agent control systems under a feedback constraint on the number of communicating state-control input pairs;
2. Development of a fair network cost allocation algorithm under sparsity constraints;
3. Development of GraSP-based and PALM-based algorithms for centralized social optimization and noncooperative games for multi-agent control systems with norm-bounded parametric uncertainty under communication-cost constraints.

Part of this thesis was published, including [2, 67, 92]. Before concluding this thesis, let us discuss several possible future directions:

Sparse linear-quadratic-Gaussian control

In this thesis, we mainly focus on static state feedback $\mathbf{u} = -\mathbf{K}\mathbf{x}$ (Chapter 2 and Chapter 3) and static output feedback $\mathbf{u} = -\mathbf{K}\mathbf{y}, \mathbf{y} = \mathbf{C}\mathbf{x}$ (Chapter 4). The designed sparse controller satisfies the communication constraint represented by the number of state(output)-control input pairs. Recently, a linear-quadratic Gaussian (LQG) control for observer-based controller, which jointly promotes sparsity on the Kalman gain and the state feedback gain [103], was designed. The

method in our thesis Chapter 4 can be extended to the case of observer-based output feedback with uncertainty.

Uncertainty in multi-agent systems with no or partial a priori information

In Chapter 4, we consider norm-bounded parametric uncertainty, with the nominal values and uncertain norm bounds known to all agents. This assumption has the advantage of a small set of parameters, and can fit nicely into the H_∞ control framework. However, for many real-world applications, the system dynamics can change over time, or the bound for uncertainty may not be known to all agents. To adapt our method to these applications, we can consider model-free approaches such as Q-learning to learn system dynamics and select the optimal control strategy at the same time, similar to the method proposed in [53].

Solution concepts beyond Nash Equilibrium

In this thesis, we consider (Generalized) Nash Equilibrium as the solution concept for the N-player noncooperative games, and we admit that NE or GNE does not necessarily exist theoretically, since the utility of the game is in general non-convex. Recently, new solution concepts with more efficient computation, such as regret minimization, were investigated in games, when the agent is uncertain of the true state or other players' strategies. In particular, regret minimization for non-convex games is studied [99] in both offline and stochastic optimization scenarios. Based on these results, we can further investigate the players' strategies (the sparse feedback matrices), assuming they follow the regret-minimization dynamics.

BIBLIOGRAPHY

[1] M. Gibbard and D. Vowles, "Simplified 14-generator model of the se australian power system," *Technical Report, The University of Adelaide, South Australia*, pp. 1–45, 2010.

[2] F. Lian, A. Chakrabortty, and A. Duel-Hallen, "Game-theoretic multi-agent control and network cost allocation under communication constraints," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 330–340, 2017.

[3] A. G. Phadke and J. S. Thorp, *Synchronized phasor measurements and their applications*. Springer Science & Business Media, 2008.

[4] P. T. Myrda and K. Koellner, "Naspinet - the internet for synchrophasors," in *43rd Hawaii International Conference on System Sciences (HICSS)*. Hawaii, 2010.

[5] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 63–86, June 2018.

[6] F. Lin, M. Fardad, and M. R. Jovanovic, "Design of optimal sparse feedback gains via the Alternating Direction Method of Multipliers," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2426–2431, 2013.

[7] F. Dörfler, M. R. Jovanović, M. Chertkov, and F. Bullo, "Sparse and optimal wide-area damping control in power networks," in *American Control Conference (ACC)*, 2013, pp. 4289–4294.

[8] F. Dörfler, M. R. Jovanović, M. Chertkov, and F. Bullo, "Sparsity-promoting optimal wide-area control of power networks," *IEEE Trans. on Power Systems*, vol. 29, no. 5, pp. 2281–2291, 2014.

[9] M. Wytock and J. Z. Kolter, "A fast algorithm for sparse controller design," *arXiv preprint arXiv:1312.4892*, 2013.

[10] A. Lamperski and L. Lessard, "Optimal decentralized state-feedback control with sparsity and delays," *Automatica*, vol. 58, pp. 143–151, 2015.

[11] M. Bahavarnia, C. Somarakis, and N. Motee, "State feedback controller sparsification via a notion of non-fragility," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 4205–4210.

[12] S. Zhang and V. Vittal, "Design of wide-area power system damping controllers resilient to communication failures," *IEEE Transactions on Power Systems*, vol. 28, p. 4, November 2013.

[13] R. Arastoo, N. Motee, and M. V. Kothare, "Optimal sparse output feedback control design: a rank constrained optimization approach," *arXiv preprint arXiv:1412.8236*, 2014.

[14] Y. Wang, J. Lopez, and M. Sznaier, "Sparse static output feedback controller design via convex optimization," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 376–381.

[15] M. S. Sadabadi and A. Karimi, "Fixed-structure sparse control of interconnected systems with polytopic uncertainty," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 2588–2593, 2014.

[16] R. P. Aguilera, G. Urrutia, R. A. Delgado, D. Dolz, and J. C. Agüero, "Quadratic model predictive control including input cardinality constraints," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3068–3075, 2017.

[17] E. Bakolas, "A solution to the minimumℓ_1-norm controllability problem for discrete-time linear systems via iteratively reweighted least squares," in *2018 Annual American Control Conference (ACC)*, June 2018, pp. 1244–1249.

[18] Y.-S. Wang, "Localized lqr with adaptive constraint and performance guarantee," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 2769–2776.

[19] S. L. Brunton and B. R. Noack, "Closed-loop turbulence control: progress and challenges," *Applied Mechanics Reviews*, vol. 67, no. 5, p. 050801, 2015.

[20] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Inferring biological networks by sparse identification of nonlinear dynamics," *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, vol. 2, no. 1, pp. 52–63, 2016.

[21] B. Polyak, M. Khlebnikov, and P. Shcherbakov, "An LMI approach to structured sparse feedback design in linear control systems," in *European Control Conference*. IEEE, 2013, pp. 833–838.

[22] R. Arastoo, M. Bahavarnia, M. V. Kothare, and N. Motee, "Closed-loop feedback sparsification under parametric uncertainties," in *IEEE 55th Conference on Decision and Control*, 2016, pp. 123–128.

[23] C. Lidström and A. Rantzer, "Optimal H_∞ state feedback for systems with symmetric and hurwitz state matrix," in *American Control Conference*, 2016, pp. 3366–3371.

[24] M. S. Andersen, S. K. Pakazad, A. Hansson, and A. Rantzer, "Robust stability analysis of sparsely interconnected uncertain systems," *IEEE Trans. on Aut. Ctrl.*, vol. 59, no. 8, pp. 2151–2156, 2014.

[25] M. Bahavarnia and N. Motee, "Row-column sparse linear quadratic controller design via bi-linear rank penalty technique and non-fragility notion," in *Control and Automation (MED), 2017 25th Mediterranean Conference on*. IEEE, 2017, pp. 1165–1169.

[26] M. Bahavarnia and N. Motee, "Sparse memoryless LQR design for uncertain linear time-delay systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 395–10 400, 2017.

[27] M. Siami and N. Motee, "Network abstraction with guaranteed performance bounds," *IEEE Transactions on Automatic Control*, 2018.

[28] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 3. IEEE, 2001, pp. 2968–2973.

[29] N. E. Leonard, D. A. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. E. Davis, "Collective motion, sensor networks, and ocean sampling," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, 2007.

[30] M. M. Zavlanos and G. J. Pappas, "Dynamic assignment in distributed motion planning with local coordination," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 232–242, 2008.

[31] F. Lian, A. Duel-Hallen, and A. Chakrabortty, "Cost allocation strategies for wide-area control of power systems using Nash bargaining solution," in *IEEE 53rd Annual Conference on Decision and Control*, Dec 2014, pp. 1701–1706.

[32] P. Chakraborty, E. Baeyens, P. P. Khargonekar, and K. Poolla, "A cooperative game for the realized profit of an aggregation of renewable energy producers," in *Decision and Control (CDC), 2016 IEEE 55th Conference on.* IEEE, 2016, pp. 5805–5812.

[33] M. Elhenawy, A. A. Elbery, A. A. Hassan, and H. A. Rakha, "An intersection game-theory-based traffic control algorithm in a connected vehicle environment," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sep. 2015, pp. 343–347.

[34] H. Pervaiz and J. Bigham, "Game theoretical formulation of network selection in competing wireless networks: An analytic hierarchy process model," in *Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST'09. Third International Conference on.* IEEE, 2009, pp. 292–297.

[35] S. Li, W. Zhang, J. Lian, and K. Kalsi, "Constrained linear quadratic Stackelberg games with applications in demand response," *arXiv preprint arXiv:1511.08838*, 2015.

[36] N. Li and J. R. Marden, "Designing games for distributed optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 230–242, 2013.

[37] F. Adib Yaghmaie, F. L. Lewis, and R. Su, "Output regulation of heterogeneous linear multi-agent systems with differential graphical game," *International Journal of Robust and Nonlinear Control*, vol. 26, no. 10, pp. 2256–2278, 2016.

[38] T. Başar and G. J. Olster, *Dynamic noncooperative game theory*. SIAM, 1995, vol. 200.

[39] D. Lukes and D. Russell, "A global theory for linear-quadratic differential games," *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 96–123, 1971.

[40] H. Mukaidani, "A numerical analysis of the Nash strategy for weakly coupled large-scale systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1371–1377, Aug. 2006.

[41] J. Engwerda, "A numerical algorithm to find soft-constrained nash equilibria in scalar lq-games," *International Journal of Control*, vol. 79, no. 06, pp. 592–603, 2006.

[42] M. Jungers, E. B. Castelan, E. R. De Pieri, and H. Abou-Kandil, "Bounded nash type controls for uncertain linear systems," *Automatica*, vol. 44, no. 7, pp. 1874–1879, 2008.

[43] N. de la Cruz and M. Jimenez-Lizarraga, "Finite time robust feedback nash equilibrium for linear quadratic games," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 794–11 799, 2017.

[44] H. Mukaidani, "Robust guaranteed cost control for uncertain stochastic systems with multiple decision makers," *Automatica*, vol. 45, no. 7, pp. 1758–1764, 2009.

[45] H. Mukaidani, "H_2/H_∞ control problem for stochastic delay systems with multiple decision makers," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*. IEEE, 2014, pp. 2648–2653.

[46] H. Mukaidani and H. Xu, "Stackelberg strategies for stochastic systems with multiple followers," *Automatica*, vol. 53, pp. 53–59, 2015.

[47] H. Mukaidani, H. Xu, and V. Dragan, "Dynamic games for markov jump stochastic delay systems," in *Recent Results on Time-Delay Systems*. Springer, 2016, pp. 207–227.

[48] H. Mukaidani, M. Unno, H. Xu, and V. Dragan, "Gain-scheduled nash games with h constraint for stochastic lpv systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1478–1483, 2017.

[49] D. Vrabie and F. Lewis, "Integral reinforcement learning for online computation of feedback nash strategies of nonzero-sum differential games," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 3066–3071.

[50] K. G. Vamvoudakis, "Non-zero sum nash q-learning for unknown deterministic continuous-time linear systems," *Automatica*, vol. 61, pp. 274–281, 2015.

[51] R. Song, F. L. Lewis, and Q. Wei, "Off-policy integral reinforcement learning method to solve nonlinear continuous-time multiplayer nonzero-sum games," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 3, pp. 704–713, 2017.

[52] K. G. Vamvoudakis, H. Modares, B. Kiumarsi, and F. L. Lewis, "Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online," *IEEE Control Systems*, vol. 37, no. 1, pp. 33–52, 2017.

[53] A. F. Dizche, A. Chakrabortty, and A. Duel-Hallen, "Sparse wide-area control of power systems using data-driven reinforcement learning," *arXiv preprint arXiv:1804.09827*, 2018.

[54] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge University Press, 2007.

[55] M. J. Osborne *et al.*, *An introduction to game theory*. Oxford university press New York, 2004, vol. 3, no. 3.

[56] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization or nonconvex and nonsmooth problems," *Mathematical Programming*, vol. 146, no. 1-2, pp. 459–494, 2014.

[57] J. Sztipanovits, X. Koutsoukos, G. Karsai, N. Kottenstette, P. Antsaklis, V. Gupta, B. Goodwine, J. Baras, and S. Wang, "Toward a science of cyber–physical system integration," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 29–44, 2012.

[58] F. L. Lewis and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 1995.

[59] S. Bahmani, B. Raj, and P. T. Boufounos, "Greedy sparsity-constrained optimization," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 807–841, 2013.

[60] A. Beck and Y. C. Eldar, "Sparsity constrained nonlinear optimization: Optimality conditions and algorithms," *SIAM Journal on Optimization*, vol. 23, no. 3, pp. 1480–1509, 2013.

[61] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Puschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.

[62] M. Fardad, F. Lin, and M. R. Jovanović, "On the optimal design of structured feedback gains for interconnected systems," in *Proceedings of the 48th IEEE Conference on Decision and Control*, 2009, pp. 978–983.

[63] L. J. Ratliff, S. A. Burden, and S. S. Sastry, "Characterization and computation of local Nash equilibria in continuous games," in *51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2013, pp. 917–924.

[64] A. van den Nouweland, "Models of network formation in cooperative games," *Group formation in economics*, pp. 58–88, 2005.

[65] K. Avrachenkov, J. Elias, F. Martignon, G. Neglia, and L. Petrosyan, "Cooperative network design: A Nash bargaining solution approach," *Computer Networks*, 2015.

[66] T. Kawamori and T. Miyakawa, "Nash bargaining solution under externalities," *Mathematical Social Sciences*, vol. 84, pp. 1–7, 2016.

[67] F. Lian, A. Duel-Hallen, and A. Chakrabortty, "Ensuring economic fairness in wide-area control for power systems via game theory," in *American Control Conference*, 2016.

[68] A. Chakrabortty and P. Khargonekar, "Introduction to wide-area control of power systems," in *American Control Conference, DC*, 2013.

[69] Y. Deng, H. Lin, A. G. Phadke, S. Shukla, J. S. Thorp, and L. Mili, "Communication network modeling and simulation for wide area measurement applications," in *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*. IEEE, 2012, pp. 1–6.

[70] M. Chenine, K. Zhu, and L. Nordstrom, "Survey on priorities and communication requirements for pmu-based applications in the nordic region," in *PowerTech, 2009 IEEE Bucharest*. IEEE, 2009, pp. 1–8.

[71] A. G. Phadke and J. S. Thorp, *Synchronized phasor measurements and their applications*. Springer Science & Business Media, 2008.

[72] T. Rautert and E. W. Sachs, "Computational design of optimal output feedback controllers," *SIAM Journal on Optimization*, vol. 7, no. 3, pp. 837–852, 1997.

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

[74] J.-S. Pang and G. Scutari, "Nonconvex games with side constraints," *SIAM Journal on Optimization*, vol. 21, no. 4, pp. 1491–1522, 2011.

[75] H. Peters, *Cooperative Games with Transferable Utility*. Springer Berlin Heidelberg, 2008, pp. 121–131.

[76] W. Saad, Z. Han, M. Debbah, and A. Hjørungnes, "A distributed coalition formation framework for fair user cooperation in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4580–4593, 2009.

[77] R. B. Myerson, "Conference structures and fair allocation rules," *International Journal of Game Theory*, vol. 9, no. 3, pp. 169–182, 1980.

[78] I. E. Hafalir, "Efficiency in coalition games with externalities," *Games and Economic Behavior*, vol. 61, no. 2, pp. 242 – 258, 2007.

[79] F. Lian. Supplementary materials for JSAC 2016. [Online]. Available: http://www4.ncsu.edu/~flian2/jsac2016_supplement.html

[80] J. Chow and K. Cheung, "A toolbox for power system dynamics and control engineering education and research," *IEEE Trans. Power Syst.*, vol. 7, no. 4, pp. 1559–1564, Nov 1992.

[81] F. Lin, M. Fardad, and M. R. Jovanovic, "Augmented lagrangian approach to design of structured optimal state feedback gains," *IEEE Transactions on Automatic Control*, vol. 56, p. 12, 2011.

[82] N. Monshizadeh, H. L. Trentelman, and M. K. Camlibel, "Projection-based model reduction of multi-agent systems using graph partitions," *IEEE Trans. on Control of Network Systems*, vol. 1, no. 2, pp. 145–154, 2014.

[83] P. P. Khargonekar and M. A. Rotea, "Mixed H_2/H_∞ control: a convex optimization approach," *IEEE Trans. on Aut. Ctrl.*, vol. 36, no. 7, pp. 824–837, 1991.

[84] Y. Kami and E. Nobuyama, "A gradient method for the static output feedback mixed H_2/H_∞ control," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 7838–7842, 2008.

[85] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

[86] M. Saeki, "Static output feedback design for H_∞ control by descent method," in *45th IEEE Conference on Decision and Control*, 2006, pp. 5156–5161.

[87] K. Zhou and J. C. Doyle, *Essentials of robust control*. Prentice hall Upper Saddle River, NJ, 1998, vol. 104.

[88] N. Motee and A. Jadbabaie, "Optimal control of spatially distributed systems," *IEEE Trans. on Aut. Ctrl.*, vol. 53, no. 7, pp. 1616–1629, 2008.

[89] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[90] F. Lin and V. Adetola, "Co-design of sparse output feedback and row/column-sparse output matrix," in *American Control Conference (ACC), 2017*. IEEE, 2017, pp. 4359–4364.

[91] N. Matni and V. Chandrasekaran, "Regularization for design," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 3991–4006, 2016.

[92] F. Lian, A. Chakrabortty, F. Wu, and A. Duel-Hallen, "Sparsity-constrained mixed H_2/H_∞ control," in *American Control Conference (ACC), 2018*, 2018.

[93] S. Seuken and S. Zilberstein, "Formal models and algorithms for decentralized decision making under uncertainty," *Autonomous Agents and Multi-Agent Systems*, vol. 17, no. 2, pp. 190–250, 2008.

[94] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic control*, vol. 49, no. 8, pp. 1292–1302, 2004.

[95] D. Paccagnan, B. Gentile, F. Parise, M. Kamgarpour, and J. Lygeros, "Distributed computation of generalized nash equilibria in quadratic aggregative games with affine coupling constraints," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 6123–6128.

[96] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[97] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer, 1984, vol. 2.

[98] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 2.

[99] E. Hazan, K. Singh, and C. Zhang, "Efficient regret minimization in non-convex games," *arXiv preprint arXiv:1708.00075*, 2017.

[100] P. Gahinet, A. Nemirovskii, A. J. Laub, and M. Chilali, "The lmi control toolbox," in *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, vol. 3. IEEE, 1994, pp. 2038–2041.

[101] G. Owen, *Game theory*. Academic Press, 1995.

[102] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Siam, 1994, vol. 15.

[103] F. Lin and S. D. Bopardikar, "Sparse linear-quadratic-gaussian control in networked systems," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 10 748–10 753, 2017.

[104] W. Saad, "Coalitional game theory for distributed cooperation in next generation wireless networks," Ph.D. dissertation, University of Oslo, 2010.

[105] T. Netzer, "Real algebraic geometry and its applications," *arXiv preprint arXiv:1606.07284*, 2016.

APPENDICES

Appendix A

A.1 Definition of Nondecreasing Selfish Payoffs for Algorithm 2.3

In Section 2.6.2, we comment that it is possible to define nondecreasing selfish payoffs in Step 4(2) of Algorithm 2.3. Given a cost constraint s, such alternative definition is

$$v_i^*(s) = J_i^{\mathrm{D}} - \underbrace{\min(\{J_i^{\mathrm{C}}(s') | s' \leq s\})}_{J_i^*(s)}, i = 1, ..., r \tag{A.1}$$

which is the maximum objective improvement an agent i can obtain by searching over the set of its selfish objectives $J_i^{\mathrm{C}}(s')$ with constraints s' that do not exceed s. If there exists some $s' < s$ such that $J_i^{\mathrm{C}}(s') < J_i^{\mathrm{C}}(s)$, the agent i might argue that the smaller selfish objective $J_i^{\mathrm{C}}(s')$, not $J_i^{\mathrm{C}}(s)$, should be used to compute its payoff since it also satisfies the constraint $(s' < s)$. It is easy to show that $v_i^*(s)$ in (A.1) is non-decreasing with s and $v_i^*(s) \geq 0, \forall s \geq 0$.

Note that the disagreement point (A.1) is hypothetical in a sense that a communication network with the energies $J_i^*(s)$ in (A.1) might not be feasible (different agents might have different s' values in (A.1) for a fixed constraint s). However, noncompatible selfish payoffs are often employed in the literature to reflect the player's subjective preferences and are not required to represent a feasible scenario [65, 66]. Moreover, successful cooperation (2.21) is not guaranteed for the payoffs (A.1). However, we found that bargaining was successful for the power system example in Section 3.5, and the payoffs (A.1) were very similar to the payoffs $v_i(s)$ defined in Algorithm 2.3 (Step 4(2)), which were shown in Fig.2.4(b).

A.2 Derivations of Q and Q_i matrices in Section 2.5, eq. (2.25, 2.28)

A.2.1 Matrix Q in eq. (2.25)

The permutation matrix $\mathcal{P}$ in eq. (2.25) is

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_1 \\ \hline \mathcal{P}_2 \end{bmatrix} \tag{A.2}$$

where

$$\begin{aligned} \mathcal{P}_2 &= \text{diag}(\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n). && \text{(A.3)} \\ \mathcal{T}_i &= \begin{bmatrix} \mathbf{0}_{(m_i-2)\times 2} & \mathbf{I}_{(m_i-2)\times(m_i-2)} \end{bmatrix} && \text{(A.4)} \\ \mathcal{P}_1 &= (p_{ij})_{2n\times s} && \text{(A.5)} \end{aligned}$$

and

$$\begin{aligned} p_{ij} &= \begin{cases} \delta_{j,k_i} & , 1 \le i \le n \\ \delta_{j,k_i+1} & , n+1 \le i \le 2n \end{cases} \\ k_i &= 1 + \sum_{k=1}^{i-1} m_k. && \text{(A.6)} \end{aligned}$$

Recall that n is the number of nodes in the system, m_i is the number of states belonging to node i, s is the total number of states in the network in (2.1), and δ_{ij} is the Kronecker delta function.

The phase angle terms in (2.25) are given by

$$\begin{aligned} &\sum_{k=1}^{n} \sum_{j=k+1}^{n} (\Delta\delta_j - \Delta\delta_k)^2 \\ &= \frac{1}{2} \sum_{k=1}^{n} \sum_{j=1}^{n} (\Delta\delta_j - \Delta\delta_k)^2 \\ &= \frac{1}{2} \sum_{k=1}^{n} |\Delta\delta_k \mathbf{1}_n - \Delta\delta|^2 \\ &= \frac{1}{2} \sum_{k=1}^{n} n\Delta\delta_k^2 - 2\Delta\delta_i \mathbf{1}_n^{\mathrm{T}} \Delta\delta + \Delta\delta^{\mathrm{T}} \Delta\delta \\ &= \frac{1}{2} [n\Delta\delta^{\mathrm{T}} \Delta\delta - 2\Delta\delta^{\mathrm{T}} \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}} \Delta\delta + n\Delta\delta^{\mathrm{T}} \Delta\delta] \end{aligned}$$

$$= \Delta\delta^{\mathrm{T}} \left(n\mathbf{I}_{n\times n} - \mathbf{1}_n \mathbf{1}^{\mathrm{T}} \right) \Delta\delta = \Delta\delta^{\mathrm{T}} \bar{\mathcal{L}} \Delta\delta \tag{A.7}$$

Thus

$$\bar{\mathcal{L}} = n\mathbf{I}_{n\times n} - \mathbf{1}_n \mathbf{1}_n^{\mathrm{T}}. \tag{A.8}$$

A.2.2 Matrix $\mathbf{Q}_i$ in eq. (2.28)

In eq.(24), the intra-area energy for agent i

$$\begin{aligned} E_i^{\text{intra}}(\mathbf{x}) &:= \sum_{k\in s_i} \sum_{\substack{j\in s_i \\ j>k}} (\Delta\delta_k - \Delta\delta_j)^2 + (\Delta\omega_k - \Delta\omega_j)^2 + \sum_{k\in s_i} \Delta E_k^2 \\ &= \begin{bmatrix} \Delta\delta \\ \Delta\omega \\ \Delta\mathbf{E} \end{bmatrix}^T \begin{bmatrix} \bar{\mathcal{L}}_i^{\text{intra}} & & \\ & \bar{\mathcal{L}}_i^{\text{intra}} & \\ & & \mathcal{I}_i^{\text{intra}} \end{bmatrix} \begin{bmatrix} \Delta\delta \\ \Delta\omega \\ \Delta\mathbf{E} \end{bmatrix} \end{aligned} \tag{A.9}$$

where $\mathcal{I}_i^{\text{intra}}$ is a block diagonal matrix with the identity matrix $\mathbf{I}_{n_i\times n_i}$ at the i^{th} diagonal block, and zeros elsewhere.

$$\mathcal{I}_i^{\text{intra}} = \text{blkdiag}(\mathbf{0}_{n_1\times n_1}, ..., \underbrace{\mathbf{I}_{n_i\times n_i}}_{\text{the } i\text{-th block}}, ..., \mathbf{0}_{n_r\times n_r}) \tag{A.10}$$

where $\text{blkdiag}(\mathbf{M}_1, ..., \mathbf{M}_n)$ represents the block-diagonal matrix with matrices $\mathbf{M}_1, ..., \mathbf{M}_n$ on the diagonal blocks. The phase angle terms of (2.26) are given by

$$\sum_{k\in s_i} \sum_{\substack{j\in s_i \\ j>k}} (\Delta\delta_k - \Delta\delta_j)^2 = \Delta\delta^{\mathrm{T}} \bar{\mathcal{L}}_i^{\text{intra}} \Delta\delta. \tag{A.11}$$

where

$$\begin{aligned} \text{LHS} &= \Delta\delta_i^{\mathrm{T}} \left(n_i \mathbf{I}_{n_i\times n_i} - \mathbf{1}_{n_i} \mathbf{1}_{n_i}^{\mathrm{T}} \right) \Delta\delta_i \\ &= \Delta\delta^{\mathrm{T}} \cdot \text{blkdiag}(\mathbf{0}_{n_1\times n_1}, ..., \underbrace{n_i \mathbf{I}_{n_i\times n_i} - \mathbf{1}_{n_i} \mathbf{1}_{n_i}^{\mathrm{T}}}_{\text{the } i\text{-th block}}, ..., \mathbf{0}_{n_r\times n_r}) \cdot \Delta\delta \\ &= \text{RHS} \end{aligned} \tag{A.12}$$

Thus

$$\bar{\mathcal{L}}_i^{\text{intra}} = \text{blkdiag}(\mathbf{0}_{n_1\times n_1}, ..., \underbrace{n_i \mathbf{I}_{n_i\times n_i} - \mathbf{1}_{n_i} \mathbf{1}_{n_i}^{\mathrm{T}}}_{\text{the } i\text{-th block}}, ..., \mathbf{0}_{n_r\times n_r}) \tag{A.13}$$

In eq.(2.27), the inter-area energy for agent i

$$E_i^{\text{inter}}(\mathbf{x}) = \frac{1}{2}\sum_{k\in s_i}\sum_{\substack{j=1,\dots,n,\\ j\notin s_i}} (\Delta\delta_k - \Delta\delta_j)^2 + (\Delta\omega_k - \Delta\omega_j)^2$$
$$= \begin{bmatrix}\Delta\delta\\ \Delta\omega\end{bmatrix}^T \begin{bmatrix}\bar{\mathcal{L}}_i^{\text{inter}} & \\ & \mathcal{I}_i^{\text{inter}}\end{bmatrix} \begin{bmatrix}\Delta\delta\\ \Delta\omega\end{bmatrix} \tag{A.14}$$

The phase angle terms of (2.27) are given by

$$\frac{1}{2}\sum_{k\in s_i}\sum_{\substack{j=1,\dots,n,\\ j\notin s_i}} (\Delta\delta_k - \Delta\delta_j)^2 = \Delta\delta^{\mathrm{T}}\bar{\mathcal{L}}_i^{\text{inter}}\Delta\delta, \tag{A.15}$$

We express the LHS as

$$\begin{aligned}
\text{LHS} &= \frac{1}{2}\sum_{k\in\mathcal{S}_i}\sum_{j=1,j\neq i}^{r} |\Delta\delta_k\mathbf{1}_{n_j} - \Delta\delta_j|^2\\
&= \frac{1}{2}\sum_{k\in\mathcal{S}_i}\sum_{j=1,j\neq i}^{r} \left(\Delta\delta_k\mathbf{1}_{n_j} - \Delta\delta_j\right)^{\mathrm{T}}\left(\Delta\delta_k\mathbf{1}_{n_j} - \Delta\delta_j\right)\\
&= \frac{1}{2}\sum_{k\in\mathcal{S}_i}\sum_{j=1,j\neq i}^{r} \left(n_j\Delta\delta_k^2 - 2\Delta\delta_k\mathbf{1}_{n_j}^{\mathrm{T}}\Delta\delta_j + \Delta\delta_j^{\mathrm{T}}\Delta\delta_j\right)\\
&= \frac{1}{2}\sum_{j=1,j\neq i}^{r} \left(n_j\Delta\delta_i^{\mathrm{T}}\Delta\delta_i - 2\Delta\delta_i^{\mathrm{T}}\mathbf{1}_{n_i}\mathbf{1}_{n_j}^{\mathrm{T}}\Delta\delta_j + n_i\Delta\delta_j^{\mathrm{T}}\Delta\delta_j\right)\\
&= \frac{1}{2}[(n-2n_i)\Delta\delta_i^{\mathrm{T}}\Delta\delta_i + n_i\Delta\delta^{\mathrm{T}}\Delta\delta - 2(\Delta\delta_i^{\mathrm{T}}\mathbf{1}_{n_i})(\mathbf{1}_n^{\mathrm{T}}\Delta\delta - \mathbf{1}_{n_i}^{\mathrm{T}}\Delta\delta_i)]\\
&= \Delta\delta^{\mathrm{T}}(\frac{n-2n_i}{2}\mathcal{I}_i^{\text{intra}})\Delta\delta + \Delta\delta^{\mathrm{T}}(\frac{n_i}{2}\mathbf{I}_{n\times n})\Delta\delta - \Delta\delta^{\mathrm{T}}(\mathcal{I}_i^{\text{intra}}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}(\mathbf{I}_{n\times n} - \mathcal{I}_i^{\text{intra}}))\Delta\delta\\
&= \Delta\delta^{\mathrm{T}}[\frac{n-2n_i}{2}\mathcal{I}_i^{\text{intra}} + \frac{n_i}{2}\mathbf{I}_{n\times n} + \mathcal{I}_i^{\text{intra}}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}(\mathbf{I}_{n\times n} - \mathcal{I}_i^{\text{intra}})]\Delta\delta = \text{RHS},
\end{aligned} \tag{A.16}$$

with

$$\bar{\mathcal{L}}_i^{\text{inter}} = \frac{n-2n_i}{2}\mathcal{I}_i^{\text{intra}} + \frac{n_i}{2}\mathbf{I}_{n\times n} + \mathcal{I}_i^{\text{intra}}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}(\mathbf{I}_{n\times n} - \mathcal{I}_i^{\text{intra}}). \tag{A.17}$$

Thus, according to eq.(2.28),

$$\mathbf{x}^T\mathbf{Q}_i\mathbf{x} = \sum_{k\in s_i}\sum_{\substack{j\in s_i\\ j>k}} (\Delta\delta_k - \Delta\delta_j)^2 + (\Delta\omega_k - \Delta\omega_j)^2 + \sum_{k\in s_i}\Delta E_k^2$$

$$= \begin{bmatrix} \Delta\delta \\ \Delta\omega \\ \Delta\mathbf{E} \end{bmatrix}^T \underbrace{\begin{bmatrix} \bar{\mathcal{L}}_i^{\text{intra}} + \bar{\mathcal{L}}_i^{\text{inter}} & & \\ & \bar{\mathcal{L}}_i^{\text{intra}} + \bar{\mathcal{L}}_i^{\text{inter}} & \\ & & \mathcal{I}_i^{\text{intra}} \end{bmatrix}}_{\mathbf{Q}'_i} \begin{bmatrix} \Delta\delta \\ \Delta\omega \\ \Delta\mathbf{E} \end{bmatrix}$$

$$= \mathbf{x}^T(\mathcal{P}^T \mathbf{Q'}_i \mathcal{P})\mathbf{x}, \tag{A.18}$$

and

$$\mathbf{Q}_i = \mathcal{P}^T \mathbf{Q'}_i \mathcal{P} \tag{A.19}$$

A.3 Efficiency of the grand coalition

Consider a coalitional game where the players within each coalition cooperate while different coalitions compete. Given a coalitional structure $\rho = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_l\}$ and a set of players $\mathcal{N} = \{1, ..., r\}$, ρ is defined as a *partition* if $\forall i \neq j$, $\mathcal{S}_i \cap \mathcal{S}_j = \phi$, and $\cup_{i=1}^{l} \mathcal{S}_i = \mathcal{N}$ [104]. In the multi-agent control problem, the control objective of each coalition $\mathcal{S} \subset \rho$ is denoted as $J_{\mathcal{S}}$, given by

$$J_{\mathcal{S}}(\mathbf{K}^{\mathcal{S}}, \mathbf{K}^{-\mathcal{S}}) = \int_{t=0}^{\infty} [\mathbf{x}^{\mathrm{T}}(t)\mathbf{Q}_{\mathcal{S}}\mathbf{x}(t) + \sum_{j \in \mathcal{S}} \mathbf{u}_j^{\mathrm{T}}(t)\mathbf{R}_j\mathbf{u}_j(t)]dt \tag{A.20}$$

where $\mathbf{K}^{\mathcal{S}}$ is the submatrix of the feedback matrix $\mathbf{K}$ that represents the strategy of the coalition $\mathcal{S}$ and is given by the union of the submatrices $\mathbf{K}^j$ in (2.11) associated with agents $j \in \mathcal{S}$. Under the sparsity constraint s, the Nash strategies of the coalitions in ρ are expressed as

$$\begin{aligned} &J_{\mathcal{S}}(\mathbf{K}^{\mathcal{S}^*}, \mathbf{K}^{-\mathcal{S}^*}) \leq J_{\mathcal{S}}(\mathbf{K}^{\mathcal{S}}, \mathbf{K}^{-\mathcal{S}^*}) \ , \forall \mathbf{K}^{\mathcal{S}} \\ &\text{s.t. } \text{card}_{\text{off}}(\mathbf{K}) \leq s \end{aligned} \tag{A.21}$$

Suppose $\mathbf{K}_\rho = (\mathbf{K}^{\mathcal{S}_1{}^*}, \mathbf{K}^{\mathcal{S}_2{}^*}, ..., \mathbf{K}^{\mathcal{S}_l{}^*})$ is the feedback matrix when the strategies of the coalitions in $\rho = \{\mathcal{S}_1, ..., \mathcal{S}_l\}$ are at a Nash Equilibrium.

The value of a coalition $\mathcal{S}$ in the partition ρ is defined as the objective reduction of $\mathcal{S}$, with respect to the decoupled game, i.e.

$$v_\rho(\mathcal{S}) = \sum_{i \in \mathcal{S}} J_i^D - J_{\mathcal{S}}(\mathbf{K}_\rho). \tag{A.22}$$

The above coalitional game is in partition form [78] since the value of each coalition depends on the composition of other coalitions. It is shown in [78] that for coalitonal games in partition form, the grand coalition $\mathcal{N} = \{1, ..., r\}$ forms when it is efficient, i.e., for any partition ρ, the

value of $\mathcal{N}$ is not exceeded by the combined values of the coalitions in ρ:

$$v_{\mathcal{N}}(\mathcal{N}) \geq \sum_{\mathcal{S}\subset\rho} v_\rho(\mathcal{S}),\ \forall\rho. \tag{A.23}$$

Next, suppose the matrices $\mathbf{Q}_\mathcal{S}$ in (A.20) satisfy

$$\sum_{\mathcal{S}\subset\rho} \mathbf{Q}_\mathcal{S} = \mathbf{Q}, \tag{A.24}$$

which is a coalition-level equivalent of (2.23). Then, for any partition ρ of $\mathcal{N}$, the sum of the values of the coalitions in ρ

$$\sum_{\mathcal{S}\subset\rho} v_\rho(\mathcal{S}) = \tilde{J}^{\mathrm{D}} - J(\mathbf{K}_\rho) \leq \tilde{J}^{\mathrm{D}} - J(\mathbf{K}_\mathcal{N}) = v_\mathcal{N}(\mathcal{N}) \tag{A.25}$$

where $\mathbf{K}_\mathcal{N}$ is the feedback matrix that satisfies the social optimization (2.6). To prove (A.25), note that $\sum_{\mathcal{S}\subset\rho} J_\mathcal{S}(\mathbf{K}_\rho) = J(\mathbf{K}_\rho)$ when (A.24) holds, and thus $\mathbf{K}_\rho$ represents a suboptimal solution to (2.6) under the constraint s, resulting in $J(\mathbf{K}_\rho) \geq J(\mathbf{K}_\mathcal{N})$. Therefore, for any partition ρ, the value of the grand coalition is at least as large as the sum of the values of the coalitions in ρ, i.e., (A.23) holds, and the grand coalition is efficient, which guarantees the formation of the grand coalition in the cooperative game and justifies Step 1 of Alg. 3 (social optimization) under the assumption (A.24).

Appendix B

B.1 Computation of (4.32)

Given a stabilizing gain $\boldsymbol{K}_0$, a sufficient condition that a point $\boldsymbol{K}_{in}$ is an inner point for the level set $\hat{\mathcal{K}}(\gamma_0)$ is [86]:

$$\boldsymbol{G}(\boldsymbol{K}_{in};\boldsymbol{K}_0)=\begin{bmatrix}\boldsymbol{H}-\boldsymbol{Z}_q\boldsymbol{Z}_q^T-\boldsymbol{R}_{\text{aug}} & * \\ \boldsymbol{U}_2^T\boldsymbol{X}_{\text{aug}}^T\boldsymbol{M}^T & \Lambda_2^{-1}\end{bmatrix}\succ 0 \tag{B.1}$$

$$\boldsymbol{H}:=\text{sym}\left\{(\boldsymbol{Z}_q\boldsymbol{\Lambda}_1^{1/2}\boldsymbol{U}_1^T-\boldsymbol{N})\boldsymbol{X}_{\text{aug}}^T\boldsymbol{M}^T\right\} \tag{B.2}$$

where

$$\boldsymbol{Z}_q=\boldsymbol{M}\boldsymbol{X}_{\text{aug}}^0\boldsymbol{U}_1\Lambda_1^{1/2}; \tag{B.3}$$

$$\boldsymbol{X}_{\text{aug}}^0=\begin{bmatrix}\boldsymbol{X}_0 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K}_0^T\end{bmatrix},\boldsymbol{X}_{\text{aug}}=\begin{bmatrix}\boldsymbol{X}_0 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K}_{in}^T\end{bmatrix}, \tag{B.4}$$

and γ_0 is the minimum γ such that (4.6) holds, $\boldsymbol{X}_0$ is a matrix $\mathbf{X}$ attained by (4.6) given $\gamma=\gamma_0$, with $\boldsymbol{K}=\boldsymbol{K}_0$ in $\boldsymbol{A}_{cl}$ and $\boldsymbol{C}_{cl}$.

The $\boldsymbol{M}$, $\boldsymbol{N}$, $\boldsymbol{R}_{\text{aug}}$, Q_{aug} matrices are given as

$$\boldsymbol{M}=\begin{bmatrix}\boldsymbol{I}_n & \boldsymbol{I}_n\\ \boldsymbol{0}_{p_1\times n} & \boldsymbol{0}_{p_1\times n}\\ \boldsymbol{0}_{m_1\times n} & \boldsymbol{0}_{m_1\times n}\end{bmatrix},\boldsymbol{N}^T=\begin{bmatrix}\boldsymbol{A} & \boldsymbol{B}_1 & \boldsymbol{0}_{n\times p_1}\\ \boldsymbol{0}_{m\times n} & \boldsymbol{0}_{m\times m_1} & \boldsymbol{D}_1^T\end{bmatrix}$$

$$\boldsymbol{R}_{\text{aug}}=\begin{bmatrix}\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{C}_1^T\\ \boldsymbol{0} & -\gamma_0\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{C}_1 & \boldsymbol{0} & -\gamma_0\boldsymbol{I}\end{bmatrix},\boldsymbol{Q}_{\text{aug}}=\begin{bmatrix}0 & \boldsymbol{B}\\ \boldsymbol{B}^T & 0\end{bmatrix} \tag{B.5}$$

The matrices $\boldsymbol{U}_1$, $\boldsymbol{U}_2$, $\boldsymbol{\Lambda}_1$, $\boldsymbol{\Lambda}_2$ are given by the eigenvalue decomposition for $\boldsymbol{Q}_{\text{aug}}$.

$$\boldsymbol{Q}_{\text{aug}} = \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 & \boldsymbol{U}_3 \end{bmatrix} \begin{bmatrix} -\Lambda & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{n-m} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_1^T \\ \boldsymbol{U}_2^T \\ \boldsymbol{U}_3^T \end{bmatrix} \tag{B.6}$$

where $\Lambda_1 = \Lambda_2 = \Lambda = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_m) \succ 0$, and $\sigma_i, i = 1, ..., m$ are the nonzero singular values of $\boldsymbol{B}$.

B.2 Overview of Zoutendijk's method

The Zoutendijk's method [85] is an approach to constrained optimization, where an improving feasible direction is generated by solving a subproblem, usually a linear program. We hereby briefly overview Zoutendijk's method for the case of nonlinear inequality constraints.

Consider the following constrained optimization problem:

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \ i = 1, ..., m \ , \end{aligned} \tag{B.7}$$

where $\mathbf{x} \in \mathbb{R}^{n \times 1}$ and $f(\mathbf{x})$ and $g_i(\mathbf{x})$ are differentiable at $\mathbf{x}$. At point $\mathbf{x}$, I is the set of active constraint $I = \{i | g_i(\mathbf{x}) = 0\}$. An improving feasible direction $\mathbf{d}$ can be found by the following linear programming problem [85]:

$$\begin{aligned} \underset{z,\mathbf{d}}{\text{Maximize}} \quad & z \\ \text{s. t.} \quad & \nabla f(\mathbf{x})^T \mathbf{d} + z \leq 0, \\ & \nabla g_i(\mathbf{x})^T \mathbf{d} + z \leq 0 \ \forall i \in I, \\ & -1 \leq d_j \leq 1, \ \forall j = 1, ..., n, \end{aligned} \tag{B.8}$$

where the third normalizing constraint prevents the optimal z from approaching ∞. It was shown [85] that if the optimal value of z, denoted as z^*, satisfies $z^* > 0$, then $\mathbf{d}$ is an improving direction since $\mathbf{d}$ satisfies $\nabla f(\mathbf{x})^T \mathbf{d} < 0$ and $\nabla g_i(\mathbf{x})^T \mathbf{d} < 0 \ \forall i \in I$. Otherwise if $z^* = 0$, then the current $\mathbf{x}$ is a Fritz John point [85], which satisfies the necessary condition for the local minimum of (B.7).

B.3 Definitions of Terms in Section 4.2.3

Definition B.3.1. (Lipschitz constant) A function $f : \mathbb{R}^d \to \mathbb{R}$ with the gradient function ∇f is Lipschitz continuous with Lipschitz constant L on $\mathcal{S} \in \mathbb{R}^d$ if $||\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})|| \leq L||\mathbf{x} - \mathbf{y}||$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{S}$ [85].

Definition B.3.2. (Proper) The function $\sigma : \mathcal{S} \to \mathbb{R}$ is a proper function if $\sigma(\mathbf{x}) > -\infty$ for all $\mathbf{x} \in \mathcal{S}$, and $\sigma(\mathbf{x}) < \infty$ for at least one point $\mathbf{x} \in \mathcal{S}$.

Definition B.3.3. (Lower semicontinuous) The function $\sigma : \mathcal{S} \to \mathbb{R}$ is lower semicontinuous at $\bar{\mathbf{x}} \in \mathcal{S}$ if for all $\epsilon > 0$ there exists a δ such that $\mathbf{x} \in \mathcal{S}$ and $||\mathbf{x} - \hat{\mathbf{x}}|| < \delta$ imply $\sigma(\mathbf{x}) - \sigma(\bar{\mathbf{x}}) > -\epsilon$.

B.4 Notation used in Kurdyka-Łojasiewicz (KL) Property, employed in convergence analysis of Algorithm 4.6

Definition B.4.1. (Distance.) For any subset $\mathcal{S} \subset \mathbb{R}^d$ and any point $x \in \mathbb{R}^d$, the distance from x to $\mathcal{S}$ is defined and denoted by

$$\text{dist}(\mathbf{x}, \mathcal{S}) := \inf\{||\mathbf{y} - \mathbf{x}|| : \mathbf{y} \in \mathcal{S}\}. \tag{B.9}$$

When $\mathcal{S} = \emptyset$, we have $\text{dist}(\mathbf{x}, \mathcal{S}) = \infty$ for all $\mathbf{x}$.

Let $\eta \in [0, \infty]$. We denote by Φ_η the class of all concave and continuous functions $\varphi : [0, \eta) \to \mathbb{R}_+$ which satisfy the following conditions:
(i) $\varphi(0) = 0$.
(ii) φ has first order continous derivative on $(0, \eta)$ and continous at 0;
(iii) for all $s \in (0, \eta) : \varphi'(s) > 0$.
For proper and lower semicontinous functions, the subdifferentials are defined below [56]:

Definition B.4.2. (Subdifferentials) Let $\sigma : \mathbb{R}^d \to (-\infty, \infty]$ be a proper and lower semicontinous function.
(i) For a given $\mathbf{x} \in \text{dom}\sigma$, the Fréchet subdifferential of σ at x, written $\hat{\partial}\sigma(\mathbf{x})$, is the set of all vectors $\mathbf{u} \in \mathbb{R}^d$ which satisfy

$$\lim_{\mathbf{y} \neq \mathbf{x}} \inf_{\mathbf{y} \to x} \frac{\sigma(\mathbf{y}) - \sigma(\mathbf{x}) - \langle \mathbf{u}, \mathbf{y} - \mathbf{x} \rangle}{||\mathbf{y} - \mathbf{x}||} \geq 0. \tag{B.10}$$

When $\mathbf{x} \notin \text{dom}\sigma$, we set $\hat{\partial}\sigma(\mathbf{x}) = \emptyset$.
(ii) The limiting subdifferential, or subdifferential, of σ at $\mathbf{x} \in \mathbb{R}^d$, written $\partial\sigma(\mathbf{x})$, is defined

through the following

$$\partial\sigma(\mathbf{x}) := \left\{\mathbf{u} \in \mathbb{R}^d : \exists \mathbf{x}^k \to \mathbf{x}, \sigma(\mathbf{x}^k) \to \sigma(\mathbf{x}) \text{ and } \mathbf{u}^k \in \hat{\partial}\sigma(\mathbf{x}^k) \to \mathbf{u} \text{ as } k \to \infty\right\} \tag{B.11}$$

Note that points whose subdifferentials contains 0 are called *(limiting-)critical points.*

Definition B.4.3. (Kurdyka-Łojasiewicz (KL) Property) Let $\sigma : \mathbb{R}^d \to (-\infty, +\infty]$ be proper and lower semicontinuous.
(i) The function σ is said to have the Kurdyka-Łojasiewicz (KL) Property at $\bar{\mathbf{u}} \in \text{dom}\partial\sigma := \{\mathbf{u} \in \mathbb{R}^d : \partial\sigma(\mathbf{u}) \neq \emptyset\}$ if there exist $\eta \in (0, \infty]$, a neighorhood $\mathcal{U}$ of $\bar{\mathbf{u}}$ and a function $\varphi \in \Phi_\eta$, such that for all

$$\mathbf{u} \in \mathcal{U} \cap [\sigma(\bar{\mathbf{u}}) < \sigma(\mathbf{u}) < \sigma(\bar{\mathbf{u}}) + \eta], \tag{B.12}$$

the following inequality holds

$$\varphi'(\sigma(\mathbf{u}) - \sigma(\bar{\mathbf{u}}))\text{dist}(0, \partial\sigma(\mathbf{u})) \geq 1. \tag{B.13}$$

(ii) If σ satisfies the KL property at each point of $\text{dom}\partial\sigma$, then σ is called a KL function.

It is shown in [56] that KL functions arise in many applications for optimization, in particular, semi-algebraic functions are KL functions. The definitions for semi-algebraic function is given as follows.

Definition B.4.4. (Semi-algebraic sets and functions). (i) A subset $\mathcal{S} \in \mathbb{R}^d$ is real semi-algebraic set if there exists a finite number of real polynomial functions $g_{ij}, h_{ij} : \mathbb{R}^d \to \mathbb{R}$ such that

$$\mathcal{S} = \cup_{j=1}^{p} \cap_{i=1}^{q} \left\{\mathbf{u} \in \mathbb{R}^d : g_{ij}(\mathbf{u}) = 0 \text{ and } h_{ij}(\mathbf{u}) < 0\right\}. \tag{B.14}$$

(ii) A function $h : \mathbb{R}^d \to (-\infty, +\infty]$ is called semi-algebraic if its graph

$$\left\{(\mathbf{u}, t) \in \mathbb{R}^{d+1} : h(\mathbf{u}) = t\right\} \tag{B.15}$$

is a semi-algebraic subset of $\mathbb{R}^{d+1}$.

B.5 Proof of Global Convergence of Algorithm 4.6

In this section, we employ results in [56,90] to analyze convergence of Algorithm 4.6. To simplify notation, we define $\tilde{g}(\mathbf{K}) \triangleq J(\mathbf{K}) + g(\mathbf{K})$, where $J(\mathbf{K})$ is the performance index of H_2 cost (4.8) and $g(\mathbf{K})$ is the indicator function for the H_∞ constraint (4.14).

Lemma B.5.1: $\tilde{g} : \mathbb{R}^{m\times p} \to (-\infty, \infty]$ and $f : \mathbb{R}^{m\times p} \to (-\infty, \infty]$ are proper and lower semi-continuous functions.

Proof. In problem (4.16) the function $J(\mathbf{K})$ is the LQR cost when using the feedback gain $\mathbf{K}$. Clearly $f(\mathcal{K}) > -\infty$, and $J(\mathbf{K}) < +\infty$ if $\mathbf{K}$ is stabilizing, thus function J is proper. In addition, $J(\mathbf{K})$ is continuous in $\mathbf{K}$ [72], thus lower semicontinuous.

The function $g(\mathbf{K})$ (4.14) is the indicator function for the level set $\mathcal{K}(\gamma)$ (4.31), and thus can take either 0 or $+\infty$, with $g(\mathbf{K}) = 0$ whenever $\mathbf{K} \in \mathcal{K}(\gamma)$. Thus $\tilde{g}(\mathbf{K})$ is proper. In addition, $g(\mathbf{K})$ is an indicator function of an open set, thus it is lower semicontinuous. Given J and g are both proper and lower semicontinuous, the summation $\tilde{g} = J + g$ is proper and lower semicontinuous. Similarly, $f(\mathbf{F})$ (4.15) is a proper function. Moreover, it is shown in [56] that it is lower semicontinuous. □

Lemma B.5.2: $H : \mathbb{R}^{m\times p} \times \mathbb{R}^{m\times p} \to \mathbb{R}$ is a continuously differentiable function, i.e., $H \in C^1$.

Proof. The gradient of $H(\mathbf{K}, \mathbf{F})$ (4.24) is countinous in $\mathbf{K}, \mathbf{F}$. Thus, $H \in C^1$. □

Lemma B.5.3: (i) $\inf_{\mathbb{R}^{m\times p},\mathbb{R}^{m\times p}} \Phi > -\infty$, $\inf_{\mathbb{R}^{m\times p}} f > -\infty$, and $\inf_{\mathbb{R}^{m\times p}} \tilde{g} > -\infty$, where Φ is given by (4.17).
(ii) The partial gradient $\nabla_{\mathbf{K}} H(\mathbf{K}, \mathbf{F})$ is globally Lipschitz with moduli $L_1(\mathbf{F})$, that is [56],

$$||\nabla_{\mathbf{K}} H(\mathbf{K}_1, \mathbf{F}) - \nabla_{\mathbf{K}} H(\mathbf{K}_2, \mathbf{F})|| \leq L_1(\mathbf{F})||\mathbf{K}_1 - \mathbf{K}_2||.$$

Likewise, the partial gradient $\nabla_{\mathbf{F}} H(\mathbf{K}, \mathbf{F})$ is globally Lipschitz with moduli $L_2(\mathbf{K})$.
(iii) There exist bounds λ_i^-, λ_i^+, $i = 1, 2$ such that

$$\begin{aligned} \inf\{L_1(\mathbf{F}^k) : k \in \mathbb{N}\} \geq \lambda_1^-, \inf\{L_2(\mathbf{K}^k) : k \in \mathbb{N}\} \geq \lambda_2^- \\ \sup\{L_1(\mathbf{F}^k) : k \in \mathbb{N}\} \leq \lambda_1^+, \sup\{L_2(\mathbf{K}^k) : k \in \mathbb{N}\} \leq \lambda_2^+ \end{aligned} \tag{B.16}$$

(iv) $\nabla H \triangleq (\nabla_{\mathbf{K}} H, \nabla_{\mathbf{F}} H)$ is Lipschitz continuous [97] on bounded subsets of $\mathbb{R}^{m\times p} \times \mathbb{R}^{m\times p}$. That is, for each bounded subset $\mathcal{B}_1 \times \mathcal{B}_2$ of $\mathbb{R}^{m\times p} \times \mathbb{R}^{m\times p}$ there exists $M > 0$, such that for all $(\mathbf{K}_i, \mathbf{F}_i) \in (\mathcal{B}_1, \mathcal{B}_2)$,

$$\begin{aligned} &||\nabla_{\mathbf{K}} H(\mathbf{K}_1, \mathbf{F}_1) - \nabla_{\mathbf{K}} H(\mathbf{K}_2, \mathbf{F}_2)||_F^2 \\ &+||\nabla_{\mathbf{F}} H(\mathbf{K}_1, \mathbf{F}_1) - \nabla_{\mathbf{F}} H(\mathbf{K}_2, \mathbf{F}_2)||_F^2 \\ &\leq M(||\mathbf{K}_1 - \mathbf{K}_2||_F^2 + ||\mathbf{F}_1 - \mathbf{F}_2||_F^2) \end{aligned} \tag{B.17}$$

Proof. (i)–(iv) are stated as assumptions in [56]. We show that these assumptions hold for our sparsity-constrained mixed H_2/H_∞ problem. It is easy to see that (i) holds since f and g are indicator functions. Since J is the LQR performance index, $J(\mathbf{K}) > 0$. Thus $\tilde{g}(\mathbf{K}) > -\infty$. In (4.18), $H(\mathbf{K}, \mathbf{F}) \geq 0$, so $\Phi(\mathbf{K}, \mathbf{F}) > -\infty$. Properties (ii) and (iii) require the partial gradient of H to be globally Lipschitz, and the Lipschitz constant be upper and lower bounded, which is easy

to verify since $L_1(\mathbf{F}^k) = L_2(\mathbf{K}^k) = \rho$ (4.24). Property (iv) holds since the left-hand side of (B.17) can be expressed as: $LHS = 2\rho^2||(\mathbf{K}_1-\mathbf{K}_2)-(\mathbf{F}_1-\mathbf{F}_2)||_F^2 \leq 4\rho^2(||\mathbf{K}_1-\mathbf{K}_2||_F^2+||\mathbf{F}_1-\mathbf{F}_2||_F^2)$. □

Assumption B.5.1: Function J is a semi-algebraic function [56].

Lemma B.5.4: The objective function Φ of (4.16) is a Kurdyka-Łojasiewicz (KL) function [56].

Remark B.5.1. A broad class of functions satisfy the semi-algebraic property, including polynomial functions, ℓ_0-norm function and indicator function of positive semidefinite cones [56]. The function $f(\mathbf{F})$ is the indicator function for the semi-algebraic set $\{\mathbf{F}|\text{card}(\mathbf{F}) \leq s\}$. Thus, function f is semi-algebraic [56, 90]. The function $g(\mathbf{K})$ is the indicator function for the level set $\mathcal{K}(\gamma)$, which is approximated by the convex level set $\hat{\mathcal{K}}(\gamma_0)$, represented by the LMI (4.32), and $\hat{\mathcal{K}}(\gamma_0)$ is a semi-algebraic set [105]. The coupling function H is polynomial so it is semi-algebraic [56]. Moreover, J is a semi-algebraic function by Assumption B.5.1. Thus, each term of Φ is semi-algebraic, and since a finite sum of semi-algebraic functions is also semi-algebraic, Φ is semi-algebraic. It is shown in Theorem 5.1 in [56] that a semi-algebraic function satisfies the KL property at any point in its domain. Thus, Φ is KL.

It has been proved in [56] that if Lemma B.5.1–B.5.3 hold, then the sequence generated by PALM algorithm globally converges. In addition, if Lemma B.5.4 holds, the sequence converges to a critical point [56] of Φ. This confirms convergence of Algorithm 4.6 to a sparsity-constrained mixed H_2/H_∞ controller, which corresponds to a critical point of Φ under mild assumptions on the functions J and g.