

# Analysis, design, and optimization of large-scale networks of dynamical systems

A DISSERTATION  
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF SOUTHERN CALIFORNIA  
BY

Sepideh Hassan-Moghaddam

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
Doctor of Philosophy

Ming Hsieh Department of Electrical and Computer Engineering

December 2019

Analysis, design, and optimization of  
large-scale networks of dynamical  
systems

Copyright © 2019

by

Sepideh Hassan-Moghaddam

ALL RIGHTS RESERVED

*To Karen, Susan, Hojjat, and Neda*

# Acknowledgements

I would like to express my sincere gratitude to my Ph.D. advisor, Professor Mihailo Jovanović, for his continuous support and guidance through these years. He was always ready to help and never gave up on making me a better researcher. He was the perfect role model in research with his great vision, immense knowledge, intelligence, and patience. I am extremely fortunate to have the opportunity to work with him and I will always be grateful for all his advices and support. His kindness and friendship have made the hard days of the graduate school enjoyable for me.

I would like to thank Professors Antonio Ortega, Ketan Savla, Urbashi Mitra, and Meisam Razaviyayn for serving on my doctoral committee. I have received invaluable comments and feedback from all of them. Also, I would like to thank Professors Tryphon Georgiou, Andrew Lamperski, Yousef Saad, and Tom Luo for their extremely useful feedback on my research. Also, I am extremely lucky to have the opportunity to work with Professor Meyn. His insightful comments and suggestions have made our joint work successful.

I am very grateful to have my past and current lab mates, Xiaofan Wu, Neil Dhingra, Armin Zare, Yongxin Chen, Wei Ran, Hesam Mohammadi, and Dongsheng Ding for making these graduate school years more interesting. I would like to also thank the senior members Rashad Moarref, Fu Lin, and Binh Lieu who built the foundation of my research by their hard work.

I am incredibly fortunate for having my PhD program in Minneapolis and Los Angeles where I met my friends for life. I would like to thank Fatemeh Sheikholeslami, Pardees Azodanloo, Maziar Sanjabi, Roushanak Navvab, Farnaz Forootaninia, Pooyan Behnam, Mehdi Lamee, Behnaz Forootaninia, Nazila Haratipour, Mehrdad Hairani,

Maral Mousavi, Meisam Razaviyayn, Saghar Sadeghpour, Pooya Rezaei, Samira Tabar, Farnoosh Zough, and Rashad Moarref. Thank you for constantly providing love and support.

I would like to sincerely thank my family. My parents have always been teaching me to believe in myself. They have scarified a lot for me to be here and I am forever indebted to them. I am grateful to my sister Neda for her constant support and infinite love. Finally and most importantly, I would like to thank my wonderful husband, Karen, for being on my side, loving me, and supporting me through all these years.

# Abstract

Large-scale networks of dynamical systems are becoming increasingly important in science and engineering with applications ranging from economics, social networks, power systems, and robotics. As a consequence, network science has emerged as an interdisciplinary field that draws from diverse disciplines including graph theory, matrix theory, dynamical systems, optimization, and statistical mechanics. One of the major challenges is the development of fast and scalable methods for analysis and design of these systems. Such systems are usually large-scale, they involve large-scale interconnections of components but have limitations on communication power, and their dynamics may be unknown. In large-scale dynamical systems, centralized implementation of control and estimation policies is computationally expensive or even infeasible. Moreover, it is often the case that the underlying dynamics are unknown and only limited input-output measurements are available from the system. To overcome these challenges, we combine tools and ideas from control theory, distributed optimization, reinforcement learning, and compressive sensing to develop distributed estimation and control strategies that utilize limited information exchange between the individual subsystems and do not require knowledge of the underlying dynamics.

In Part I of this dissertation, we study the problem of optimal topology design for stochastically forced undirected consensus networks in order to optimally enhance closed-loop performance. The performance is given by the steady-state variance amplification of the network with additive stochastic disturbances. We use a sparsity promoting optimal control framework to avoid computational complexity. In particular, we introduce  $\ell_1$ -regularization into the optimal  $\mathcal{H}_2$  formulation and cast the design problem as a nonsmooth composite optimization problem. By exploiting the structure of the problem, we develop customized proximal gradient and Newton algorithms that are well-suited for large-scale problems. We illustrate that our algorithms can solve the problems with more than million edges in the controller graph in a few minutes, on a PC. We also exploit structure of connected resistive networks to demonstrate how additional edges can be systematically added in order to minimize the  $\mathcal{H}_2$  norm of the

closed-loop system. Moreover, we study the problem of performance enhancement in stochastically-forced directed consensus networks by adding edges to an existing topology. We formulate the problem as a feedback control design, and represent the links as the elements of the controller graph Laplacian matrix. To deal with the structural constraints that arise from the absence of absolute measurements, we introduce a coordinate transformation to eliminate the average mode and assure convergence of all states to the average of the initial node values. By exploiting structure of the optimization problem, we develop a customized algorithm based on the alternating direction method of multipliers to design a sparse controller network that improves the performance of the closed-loop system.

Part II studies the problem of identifying sparse interaction topology using sample covariance matrix of the states of the network. We assume that the statistics are generated by a stochastically-forced undirected consensus network with unknown topology. We propose a method for identifying the topology using a regularized Gaussian maximum likelihood framework where the  $\ell_1$  regularizer is introduced as a means for inducing sparse network topology. The proposed algorithm employs a sequential quadratic approximation in which the Newton's direction is obtained using coordinate descent method. We also develop a method based on growing a Chow-Liu tree that is well-suited for identifying the underlying structure of large-scale systems in which some of the nodes may have access to their own states.

In Part III, we study the algorithms that can be used to solve nonsmooth composite optimization problems from control-theoretic point of view. We view proximal algorithms as dynamical systems and leverage techniques from control theory to study their global convergence properties. In particular, for problems with strongly convex objective functions, we utilize the theory of integral quadratic constraints to prove global exponential stability of the differential equations that govern the evolution of proximal gradient and Douglas-Rachford splitting flows. Moreover, we establish conditions for global exponential convergence even in the absence of strong convexity. We also study a class of nonsmooth composite optimization problems in which the convex objective function is given by a sum of differentiable and nondifferentiable terms. We propose a primal-descent dual-ascent gradient flow method that exploits separability of the objective function and is well-suited for in-network optimization. We prove global asymptotic

stability of the proposed algorithm and solve the problem of growing undirected consensus networks in a distributed manner to demonstrate its effectiveness.

Part IV focuses on the distributed design of structured feedback gains for large-scale systems. This is a challenging problem even in the absence of additional structural constraints on the feedback controller. We approach the structured optimal control problem via a data-driven framework that does not require knowledge of the system parameters and avoids the need to solve large-scale matrical equations. For the structured optimal  $H_2$  state-feedback problem, we show that the objective function and its gradient can be computed from data and develop customized proximal algorithms based on gradient descent and incremental gradient method. Moreover, we exploit separability of the objective function and utilize an ADMM-based consensus algorithm to solve the regularized optimal control problem in a distributed manner over multiple processors.



# Table of Contents

<b>Acknowledgements</b>	<b>ii</b>
<b>Abstract</b>	<b>iv</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Main topics of the dissertation . . . . .	3
1.1.1 Distributed design and control of consensus networks . . . . .	3
1.1.2 Network estimation and inference . . . . .	5
1.1.3 Control theoretic approach for analysis of optimization algorithms	7
1.1.4 Distributed in-network optimization . . . . .	8
1.1.5 Data-driven and model-free distributed control . . . . .	9
1.2 Dissertation structure . . . . .	10
1.3 Contributions of the dissertation . . . . .	12
<b>I Distributed design and control of networks</b>	<b>16</b>
<b>2 Topology design for stochastically forced undirected consensus net-</b>	<b>17</b>
<b>works</b>	
2.1 Introduction . . . . .	17
2.2 Problem formulation . . . . .	19
2.2.1 Design of optimal sparse topology . . . . .	22

2.2.2	Structured optimal control problem: debiasing step . . . . .	24
2.2.3	Gradient and Hessian of $J(x)$ . . . . .	25
2.3	Dual problem . . . . .	25
2.4	Customized algorithms . . . . .	29
2.4.1	Proximal gradient method . . . . .	29
2.4.2	Proximal Newton method . . . . .	30
2.5	Growing connected resistive networks . . . . .	34
2.5.1	Proximal gradient method . . . . .	35
2.5.2	Proximal Newton method . . . . .	36
2.6	Computational experiments . . . . .	37
2.6.1	Performance comparison . . . . .	38
2.6.2	Large-scale Facebook network . . . . .	39
2.6.3	Random disconnected network . . . . .	41
2.6.4	Path and ring networks . . . . .	43
2.7	Concluding remarks . . . . .	43
<b>3</b>	<b>Edge addition in directed consensus networks</b>	<b>46</b>
3.1	Introduction . . . . .	46
3.2	Motivation and background . . . . .	47
3.2.1	Directed graphs . . . . .	48
3.2.2	Feedback design in consensus networks . . . . .	48
3.2.3	Elimination of the average mode . . . . .	50
3.3	Topology design for directed networks . . . . .	52
3.3.1	Sparsity-promoting optimal control problem . . . . .	52
3.3.2	Structured optimal control problem . . . . .	53
3.4	An ADMM-based algorithm . . . . .	53
3.4.1	$F$ -minimization step . . . . .	54
3.4.2	$L_x$ -minimization step . . . . .	56
3.5	Computational experiments . . . . .	56
3.5.1	Synthetic example . . . . .	56
3.5.2	Performance improvement . . . . .	57
3.6	Concluding remarks . . . . .	59

<b>II</b>	<b>Network estimation and inference</b>	<b>60</b>
<b>4</b>	<b>Topology identification of undirected consensus networks via sparse inverse covariance estimation</b>	<b>61</b>
4.1	Introduction . . . . .	61
4.2	Problem formulation . . . . .	62
4.2.1	Undirected consensus networks . . . . .	62
4.2.2	Topology identification . . . . .	64
4.3	Customized algorithm based on sequential quadratic approximation . .	65
4.3.1	Structured problem: debiasing step . . . . .	66
4.3.2	Gradient and Hessian of $J(x)$ . . . . .	66
4.3.3	Algorithm . . . . .	67
4.4	Computational experiments . . . . .	70
4.4.1	Network identification . . . . .	70
4.4.2	Network sparsification . . . . .	72
4.5	Concluding Remarks . . . . .	75
<b>5</b>	<b>Topology identification via growing a Chow-Liu tree network</b>	<b>76</b>
5.1	Introduction . . . . .	76
5.2	Topology identification via structured graphical-LASSO . . . . .	77
5.3	Growing a Chow-Liu tree network . . . . .	80
5.4	Computational experiments . . . . .	83
5.4.1	Structured graphical-LASSO algorithm . . . . .	83
5.4.2	Topology identification via growing a Chow-Liu tree . . . . .	84
5.4.3	FMRI dataset . . . . .	85
5.5	Concluding remarks . . . . .	87
<b>III</b>	<b>Control theoretic approach for analysis of optimization algorithms</b>	<b>88</b>
<b>6</b>	<b>Proximal gradient flow and Douglas-Rachford splitting dynamics: global exponential stability via integral quadratic constraints</b>	<b>89</b>
6.1	Introduction . . . . .	90

6.2	Problem formulation and background . . . . .	91
6.2.1	Proximal operators and the associated envelopes . . . . .	91
6.2.2	Proximal augmented Lagrangian . . . . .	94
6.2.3	Strong convexity and Lipschitz continuity . . . . .	95
6.2.4	Proximal Polyak-Lojasiewicz inequality . . . . .	95
6.3	Exponential stability of proximal algorithms . . . . .	96
6.3.1	Strongly convex problems . . . . .	98
6.3.2	Proximal Polyak-Lojasiewicz condition . . . . .	100
6.4	Global exponential stability of Douglas-Rachford splitting dynamics . .	102
6.4.1	Non-smooth composite optimization problem . . . . .	102
6.4.2	Douglas-Rachford splitting on the dual problem . . . . .	107
6.5	Concluding remarks . . . . .	108
<b>7</b>	<b>Distributed proximal augmented Lagrangian method for nonsmooth composite optimization</b>	<b>109</b>
7.1	Introduction . . . . .	109
7.2	Problem formulation . . . . .	111
7.2.1	Motivating application . . . . .	112
7.2.2	Background . . . . .	114
7.3	Proximal augmented Lagrangian . . . . .	116
7.4	Arrow-Hurwicz-Uzawa gradient flow . . . . .	118
7.5	Distributed implementation . . . . .	122
7.5.1	Primal update . . . . .	122
7.5.2	Dual updates . . . . .	123
7.6	Computational experiments . . . . .	123
7.7	Concluding remarks . . . . .	124
<b>IV</b>	<b>Data-driven and model-free distributed control</b>	<b>125</b>
<b>8</b>	<b>Data-driven proximal algorithms for the design of structured optimal feedback gains</b>	<b>126</b>
8.1	Introduction . . . . .	126

8.2	Problem formulation . . . . .	128
8.3	Computation from data . . . . .	130
8.3.1	Computation of $P$ and $L$ from data . . . . .	131
8.3.2	Computation of $PL$ from data . . . . .	132
8.4	Systems with special structure . . . . .	134
8.5	Proximal algorithms . . . . .	138
8.5.1	Proximal gradient descent . . . . .	138
8.5.2	Incremental proximal gradient . . . . .	139
8.6	An ADMM-based consensus algorithm . . . . .	140
8.6.1	Computational experiments . . . . .	143
8.7	Concluding remarks . . . . .	146
<b>9</b>	<b>Conclusions</b>	<b>148</b>
	<b>References</b>	<b>150</b>
	<b>Appendix A. Proximal PL condition</b>	<b>169</b>
A.1	Pointwise quadratic inequality . . . . .	171
	<b>Appendix B. On the exponential convergence rate of proximal gradient flow algorithms</b>	<b>172</b>
B.0.1	Distributed optimization . . . . .	172

# List of Tables

2.1	Comparison of our customized proximal algorithms . . . . .	33
2.2	Algorithmic properties of the primal dual IP method. . . . .	33
2.3	Comparison of algorithms (solve times in seconds/number of iterations) for the problem of growing connected resistive Erdős-Rényi networks with different number of nodes $n$ , edge probability $1.05 \log(n)/n$ , and $\gamma =$ $0.8 \gamma_{\max}$ . . . . .	38
2.4	Table 2.3 continued . . . . .	38

# List of Figures

1.1	(a) Social network. (b) Computer network. (c) Vehicular formation. (d) Power network. . . . .	2
2.1	(a) Solve times (in seconds); and (b) performance degradation (in percents) of proximal gradient and greedy algorithms relative to the optimal centralized controller. . . . .	39
2.2	(a) Sparsity level; and (b) optimal tradeoff curves resulting from the application of proximal gradient algorithm and a heuristic strategy for the Facebook network. . . . .	40
2.3	Topologies of the plant (blue lines) and controller graphs (red lines) for an unweighted random network with three disconnected subgraphs. . . .	42
2.4	(a) Sparsity level; (b) performance degradation; and (c) the optimal tradeoff curve between the performance degradation and the sparsity level of optimal sparse $x$ compared to the optimal centralized vector of the edge weights $x_c$ . The results are obtained for unweighted random disconnected plant network with topology shown in Fig. 2.3. . . . .	43
2.5	The problems of growing unweighted path (top row) and ring (bottom row) networks. Blue lines identify edges in the plant graph, and red lines identify edges in the controller graph. . . . .	44
3.1	Topologies of the plant (blue lines) and controller graphs (red lines) for a randomly generated weighted network. . . . .	58

3.2	(a) Sparsity level; (b) performance degradation; and (c) the optimal tradeoff between the performance degradation and the sparsity level of the optimal sparse $L_x$ compared to the optimal centralized controller $L_c$ . The results are obtained for the weighted random plant network with topology shown in Fig. 3.1a. . . . .	59
4.1	The problem of identification of sparse networks using sample covariance matrix for a network with $n = 12$ nodes. . . . .	71
4.2	The problem of identification of sparse networks using sample covariance matrix for a network with $n = 12$ nodes. . . . .	72
4.3	The problem of sparsification of a network with $n = 50$ nodes using sample covariance matrix. . . . .	73
4.4	Relative error in log-scale for the sparsification problem of a network with $n = 50$ nodes using sample covariance matrix. . . . .	74
5.1	The RC tree network with $n = 10$ nodes. . . . .	83
5.2	(a) Plant network; (b) constructed Chow-Liu tree network; and (c) the identified network from growing the Chow-Liu tree. . . . .	84
5.3	The identified networks for different values of $\gamma$ . . . . .	86
7.1	Speedup ratio versus the number of the cores used for growing connected resistive networks with $n$ nodes. . . . .	124
8.1	A mass-spring-damper system on a line. . . . .	143
8.2	Structure of feedback gains resulting from the solution to (8.6) for different values of the regularization parameter $\gamma$ for an MSD system with $N = 10$ masses. The number of non-zero elements in $K$ is denoted by $n_z$ . . . . .	144
8.3	(a) Sparsity level; (b) performance degradation; and (c) the optimal tradeoff between the performance degradation and the sparsity level of the optimal sparse $x$ compared to the optimal centralized controller $x_c$ . The results are obtained for a randomly generated Erdős-Rényi network with $n = 20$ nodes. . . . .	145
8.4	Speedup ratio versus the number of the cores used for growing connected resistive networks with $n$ nodes. . . . .	146



# Chapter 1

## Introduction

Traditional control theory is concerned with implementing the control and estimation policies in a centralized fashion [1]. However, in large networks of dynamical systems, centralized processing is often computationally expensive or even practically infeasible. This motivates the development of distributed control strategies that require limited information exchange between the individual subsystems. Analysis and design of distributed averaging protocols have been an active research area recently; e.g., see [2–11].

Moreover, distributed computing over networks is of fundamental importance in network science [2]. Consensus problem has received significant attention because it encounters in a number of applications ranging from animal group behavior [12, 13], to social networks [14, 15], to power systems [16–18], to distributed processing networks [19, 20], to spreading processes [21, 22], and to coordination of autonomous vehicles [23–26], see Fig. 1.1 for some examples. An inherent challenge in all of these problems is that it is desired for all nodes to reach an agreement or to achieve synchronization by only exchanging relative information with limited number of nodes. The restriction on the absence of the absolute measurements imposes structural constraints for the analysis and design.

Furthermore, in many of the real-world applications, the dynamics of the underlying system are unknown, the dynamical systems are large-scale, and only input-output measurements are available. This necessitates the development of theory and techniques that utilize distributed computing architectures to cope with large problem sizes and

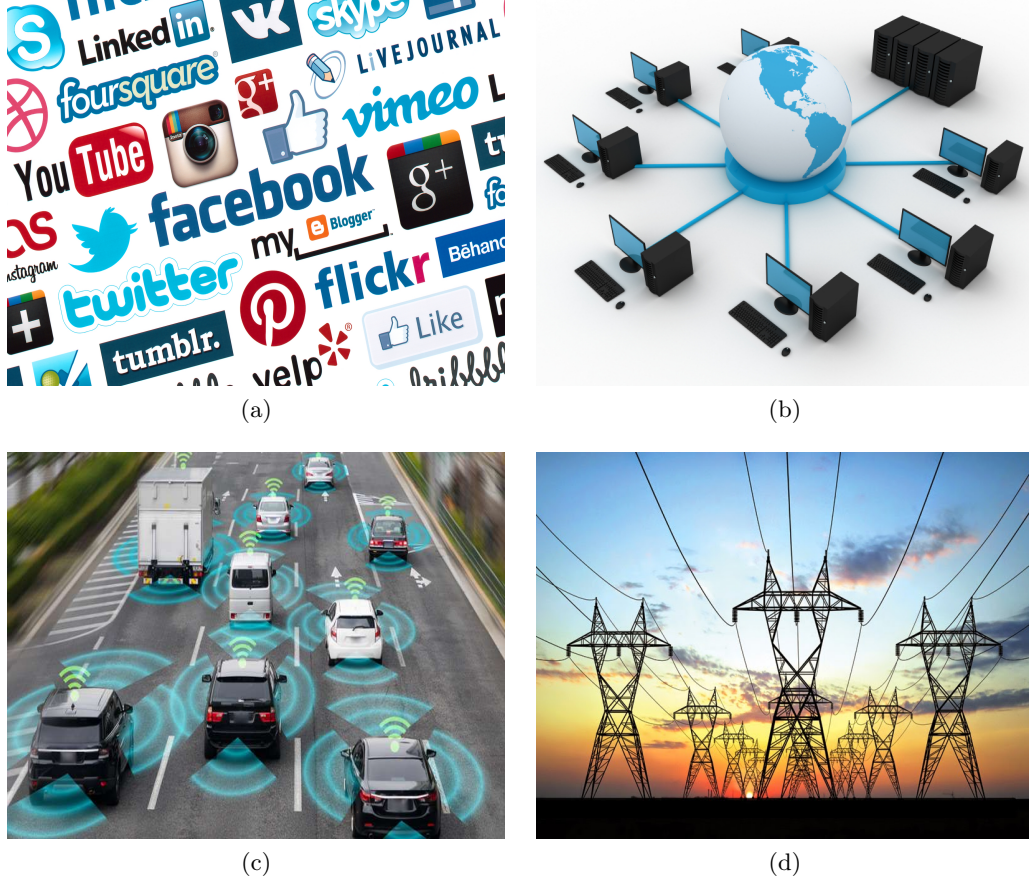


Figure 1.1: (a) Social network. (b) Computer network. (c) Vehicular formation. (d) Power network.

do not require the knowledge of underlying dynamics of the system. Moreover, in some applications, having access to even the input-output measurements is not feasible and only second-order data is available. Recovering network topology from partially available statistical signatures is a challenging problem. Due to experimental or numerical limitations, it is often the case that only noisy partial network statistics are known. Thus, the objective is to account for all observed correlations that may be available and to develop efficient identification algorithms that are well-suited for big data problems [27, 28].

Moreover, analysis of optimization algorithms from the system theoretic point of

view has received significant recent attention [29–31]. In these references, the optimization algorithm is interpreted as a feedback interconnection in which the states converge to the optimal solution of the optimization problem. Many modern large-scale and distributed optimization problems can be cast into a form in which the objective function is a sum of a smooth term and a nonsmooth regularizer. Such problems can be solved via a proximal gradient method which generalizes standard gradient descent to a non-smooth setup. We leverage the tools from control theory to study global convergence of proximal gradient flow algorithms in continuous time. We also provide a distributed implementation of the gradient flow dynamics based on the proximal augmented Lagrangian and prove global exponential stability for strongly convex problems.

In this dissertation, we combine tools and ideas from control theory, distributed optimization, reinforcement learning, and compressive sensing to develop distributed estimation and control strategies that require limited information exchange between the individual subsystems. The remainder of this introductory chapter is organized as follows. In Section 1.1, we overview different topics of the dissertation and we provide a brief discussion on each of them. The outline of the dissertation is provided in Section 1.2. Finally, we provide a summary of our main results and contributions in Section 1.3.

**1.1 Main topics of the dissertation** In this section, we discuss the main topics of the dissertation.

**1.1.1 Distributed design and control of consensus networks** Reaching agreement using relative information exchange in a decentralized fashion has received lots of attention. In large networks, centralized implementation of control policies imposes heavy communication and computation burden on individual nodes. This motivates the development of distributed control strategies that require limited information exchange between the nodes in order to reach consensus or achieve synchronization [5, 7–9].

Optimal design of the edge weights for networks with pre-specified topology has received significant attention. In [3], the design of the fastest averaging protocol for

undirected networks was cast as a semidefinite program (SDP). Two customized algorithms, based on primal barrier interior-point (IP) and subgradient methods, were developed and the advantages of optimal weight selection over commonly used heuristics were demonstrated. Similar SDP characterization, for networks with state-dependent graph Laplacians, was provided in [4]. The allocation of symmetric edge weights that minimize the mean-square deviation from average for networks with additive stochastic disturbances was solved in [5]. A related problem, aimed at minimizing the total effective resistance of resistive networks, was addressed in [7]. In [8], the edge Laplacian was used to provide graph-theoretic characterization of the  $\mathcal{H}_2$  and  $\mathcal{H}_\infty$  symmetric agreement protocols.

Network coherence quantifies the ability of distributed estimation and control strategies to guard against exogenous disturbances [6, 10]. The coherence is determined by the sum of reciprocals of the non-zero eigenvalues of the graph Laplacian and its scaling properties cannot be predicted by algebraic connectivity of the network. In [10], performance limitations of spatially-localized consensus protocols on regular lattices were examined. It was shown that the fundamental limitations for large-scale networks are dictated by the network topology rather than by the optimal selection of the edge weights. Moreover, epidemic spread in networks is strongly influenced by their topology [21, 22, 32]. Thus, optimal topology design represents an important challenge. It is precisely this problem, for undirected consensus networks, that we address here.

More specifically, we study an optimal control problem aimed at achieving a desired tradeoff between the network performance and communication requirements in the distributed controller. Our goal is to add a certain number of edges to a given undirected network in order to optimally enhance the closed-loop performance. One of our key contributions is the formulation of topology design as an optimal control problem that admits convex characterization and is amenable to the development of efficient optimization algorithms. In our formulation, the plant network can contain disconnected components and optimal topology of the controller network is an integral part of the design. In general, this problem is NP-hard [33] and it amounts to an intractable combinatorial search. Several references have examined convex relaxations or greedy algorithms to design topology that optimizes algebraic connectivity [34] or

network coherence [9, 35–37].

In this dissertation, we tap on recent developments regarding sparse representations in conjunction with regularization penalties on the level of communication in a distributed controller. Similar approaches have been used for solving the problem of optimal sensors/actuators selections in large-scale dynamical systems [38–40]. This allows us to formulate convex optimization problems that exploit the underlying structure and are amenable to the development of efficient optimization algorithms. To avoid combinatorial complexity, we approach optimal topology design using a sparsity-promoting optimal control framework introduced in [41, 42]. Performance is captured by the  $\mathcal{H}_2$  norm of the closed-loop network and  $\ell_1$ -regularization is introduced to promote controller sparsity. While this problem is in general nonconvex [42], for undirected networks we show that it admits a convex characterization with a non-differentiable objective function and a positive definite constraint. This problem can be transformed into an SDP and, for small size networks, the optimal solution can be computed using standard IP method solvers, e.g., SeDuMi [43] and SDPT3 [44].

Moreover, we study the problem of performance enhancement in stochastically-forced directed consensus networks by adding edges to an existing topology. Significant amount of research has been devoted to the study of the consensus problem in networks, where both the plant and the controller graphs are undirected. We consider the problem of adding edges to a weakly connected directed consensus network in order to improve performance. In particular, we are interested in designing sparse communication graphs that strike a balance between the variance amplification of the closed-loop system and the number of communication links. In general, this is a combinatorial search problem and is non-convex.

**1.1.2 Network estimation and inference** Identifying network topology and learning graphical models from partially available statistical signatures are topics of immense interest in areas ranging from machine learning to statistics to neuroscience [45–52]. Moreover, structured covariance completion problem has received significant attention in the recent years [53–55]. In these works, the unknown elements of a partially known covariance matrix are estimated such that they are consistent with the dynamics of the underlying system. Studying the human brain as a complex network has received

significant attention recently [56–58]. The brain functional connectivity can be measured by computing the correlation between time-series functional magnetic resonance imaging (fMRI) data. The functional connectivity structure between different regions can be revealed by utilizing different thresholding techniques [59, 60]. In general, this is a challenging problem because it is often the case that only noisy partial network statistics are known. The goal is to develop an efficient algorithm for recovering the underlying topology of a network utilizing the limited sample data.

Recovering the underlying network topology using sample covariance matrix of the node values under structural constraints has been studied in [27, 61, 62]. Moreover, a rich body of literature has been devoted to the problems of designing network topology to improve performance [33, 34, 36, 37, 63]. Several algorithms can be employed to identify the underlying network structure from limited statistical data. In [64], the authors show inability of standard graphical-LASSO to identify network topology. It was demonstrated that this popular algorithm fails to recover the underlying topology even when the abundance of data is available.

Over the last decade, a rich body of literature has been devoted to the problems of designing network topology to improve performance [9, 34–37, 63, 65, 66] and identifying an unknown network topology from available data [45–51, 67]. Moreover, the problem of sparsifying a network with dense interconnection structure in order to optimize a specific performance measure has been studied in [68, 69]. In [70], it was demonstrated that individual starlings within a large flock interact only with a limited number of immediate neighbors. The authors have shown that the networks with six or seven nearest neighbors provide an optimal trade-off between flock cohesion and effort of individual birds. These observations were made by examining how robustness of the group depends on the network structure. The proposed measure of robustness captures the network coherence [10] and it is well-suited for developing tractable optimization framework.

We develop a convex optimization framework for identifying sparse interaction topology using sample covariance matrix of the state of the network. Our framework utilizes an  $\ell_1$ -regularized Gaussian maximum likelihood estimator. We also develop a method based on growing a Chow-Liu tree that is well-suited for identifying the underlying

structure of large-scale systems. We apply this technique to resting-state functional MRI (fMRI) data as well as synthetic datasets to illustrate the effectiveness of the proposed approach.

### 1.1.3 Control theoretic approach for analysis of optimization algorithms

Structured optimal control and inverse problems, that arise when trying to identify and control dynamical representations of systems evolving in real-time, typically lead to optimization of objective functions that consist of a sum of a smooth term and a nonsmooth regularizer. Such problems are of increasing importance in control [41, 42, 71] and it is thus necessary to develop efficient algorithms for distributed and embedded nonsmooth composite optimization [72–75]. The lack of differentiability in the objective function precludes the use of standard descent methods from smooth optimization. Proximal gradient method [76] generalizes gradient descent to nonsmooth context and provides a powerful tool for solving problems in which the nonsmooth term is separable over the optimization variable.

Examining optimization algorithms as continuous-time dynamical systems has been an active topic since the seminal work of Arrow, Hurwicz, and Uzawa [77]. This viewpoint can provide important insight into performance of optimization algorithms and, during the last decade, it has been advanced and extended to a broad class of problems including convergence analysis of primal-dual [73, 78–82] and accelerated [83–88] first-order methods. This should be compared and contrasted to existing techniques which employ subgradient methods and discontinuous projected dynamics [89–91]. Furthermore, establishing the connection between theory of ordinary differential equations (ODEs) and numerical optimization algorithms has been a topic of many studies, including [92, 93]; for recent efforts, see [84, 94].

Most algorithms can be viewed as a feedback interconnection of linear dynamical systems with nonlinearities that possess certain structural properties. This system-theoretic interpretation was exploited in [31] and further advanced in a number of recent papers [81, 95–101]. The key idea is to exploit structural features of linear and nonlinear terms and utilize theory and techniques from stability analysis of nonlinear dynamical systems to study properties of optimization algorithms. This approach provides new

methods for studying not only convergence rate but also robustness of optimization routines [102–105] and can lead to new classes of algorithms that strike a desired tradeoff between the speed and robustness.

We study a class of nonsmooth composite convex optimization problems in which the objective is a sum of a differentiable function and a possibly nondifferentiable regularizer. These problems emerge in compressive sensing, machine learning, and control. For example, structured feedback design can be cast as a nonsmooth composite optimization problem [42, 71, 106]. Standard descent methods cannot be used in the presence of nondifferentiable component. Proximal gradient algorithms [76, 107] offer viable alternatives to the generic descent methods for solving nonsmooth problems. Another effective strategy is to transform the associated augmented Lagrangian into the continuously differentiable proximal augmented Lagrangian [108] in which the former is restricted to the manifold that corresponds to the explicit minimization over the variable in the nonsmooth term.

**1.1.4 Distributed in-network optimization** Distributed algorithms are critically important for solving large-scale optimization problems. The decentralized consensus problem in multi-agent networks [72, 73, 109, 110] arises in many applications. Moreover, distributed control techniques are critically important in the design of large-scale systems. In these systems, conventional control strategies that rely on centralized computation and implementation are often prohibitively expensive. For example, finding the optimal controller requires computation of the solution to the algebraic Riccati equations which is often infeasible because of high computational requirements. This necessitates the development of theory and techniques that utilize distributed computing architectures to cope with large problem sizes.

We study a class of nonsmooth composite optimization problems in which the convex objective function is a sum of differentiable and nondifferentiable functions. Among other applications, these problems emerge in machine learning, compressive sensing, and control. Recently, regularization has been used as a promising tool for enhancing utility of standard optimal control techniques. Generic descent methods cannot be used in the nonsmooth composite optimization problems due to the presence of a



nondifferentiable component in the objective function. Moreover, these standard methods are not well-suited for distributed implementation. An alternative approach is to separate the smooth and nonsmooth parts of the objective function and use the alternating direction method of multipliers (ADMM). In [111], we exploit separability of the objective function and utilize an ADMM-based consensus algorithm to solve the regularized optimal control problem in a distributed manner over multiple processors. Even though the optimal control problem is in general non-convex, recent results can be utilized to show convergence to a local minimum [112]. However, in an update step of the ADMM algorithm, all the processors halt to compute the weighted average (the gathering step) [113].

By introducing auxiliary variables in nondifferentiable terms, we provide an equivalent consensus-based characterization that is convenient for distributed implementation. The Moreau envelope associated with the nonsmooth part of the objective function is used to bring the optimization problem into a continuously differentiable form that serves as a basis for the development of a primal-descent dual-ascent gradient flow method. This algorithm exploits separability of the objective function and is well-suited for in-network optimization. We prove global asymptotic stability of the proposed algorithm. Moreover, we study primal-descent dual-ascent gradient flow dynamics based on the proximal augmented Lagrangian [108] that can be used as an effective algorithm to solve smooth optimization problems with a separable objective function. We provide a distributed implementation and prove global exponential stability in the presence of strong convexity.

**1.1.5 Data-driven and model-free distributed control** One of the major challenges in analysis and design of large-scale dynamical systems is the development of fast and scalable methods. Such systems involve large-scale interconnections of components, have rapidly-evolving structure and limitations on communication/processing power, and require real-time distributed control actions. These requirements make control strategies that rely on centralized information processing infeasible and motivate new classes of optimal control problems. In these, standard performance metrics are augmented with typically nonsmooth regularizers to promote desired structural features (e.g., low communication requirements) in the optimal controller [41, 42, 71, 114, 115].

Moreover, in many applications, the dynamics of the plant are unknown and only limited input-output measurements are available. Designing optimal controllers for these systems is challenging even in the absence of structural constraints. The LQR problem in the model-free setup has been recently studied in [116, 117].

Modern control applications impose additional requirements on controller design that cannot be addressed using standard optimal control tools. These requirements may arise from limited communication and computation resources or the size of the problem. The standard optimal control techniques typically induce an all-to-all communication requirements in the controller which is infeasible in large-scale setting. Recently, regularization has emerged as a promising tool for enhancing utility of standard optimal control techniques. In this approach, commonly used performance measures (e.g.,  $\mathcal{H}_2$  or  $\mathcal{H}_\infty$ ) are augmented with regularization functions that are supposed to promote some desired structural features in the distributed controller, e.g., sparsity. Such an approach has received significant level of attention in recent years [41, 42, 71, 106, 115, 118, 119], but computing optimal solutions in large-scale problems still remains a challenge. Distributed computing techniques have been commonly used to cope with large problem sizes; for example, stability and synthesis of cooperative distributed model predictive controllers for linear systems have been recently studied in [120].

Herein, we address complexity through a combination of linear-algebraic techniques and computational methods adapted from both machine learning and reinforcement learning. We study the problem of designing optimal structured feedback gains for large-scale systems with unknown dynamics in a distributed manner. We quantify performance using the  $\mathcal{H}_2$  norm and introduce regularization functions to promote desired structural properties in the controller. The key challenge is to evaluate the objective function and its gradient without solving the large-scale Lyapunov equations. By exploiting the square-additive property of the  $\mathcal{H}_2$  norm, we propose customized proximal and ADMM-based algorithms that are convenient for distributed implementation.

**1.2 Dissertation structure** This dissertation consists of four parts. Each part focuses on a specific topic and includes individual chapters that studies relevant subjects. In each chapter, we provide a brief introduction, background and motivation, problem formulation, algorithm and design steps, computational experiments, and

conclusion.

**Part I.** Traditional control theory is concerned with implementing the control policies in a centralized fashion. However, due to high communication and computation costs, this is not practically feasible for large-scale networks of dynamical systems. Leveraging tools from compressive sensing and distributed optimization, we have developed a theoretical and computational framework for designing distributed controllers for large-scale systems [63, 66, 106, 119]. In these, standard performance metrics are augmented with typically nonsmooth regularizers to promote desired structural features (e.g., low communication requirements) in the optimal controller. These algorithms scale gracefully with problem size and are well-suited for distributed implementation over multiple processors.

**Part II.** Identifying network topology and learning graphical models from partially available statistical signatures are topics of immense interest in areas ranging from machine learning to statistics to neuroscience. A motivating application in neuroscience is reconstructing the pattern of causal interactions between distinct units of the brain using time series data of neural activities. For undirected consensus networks, the topology identification problem can be imposed as a sparse inverse covariance estimation with additional structural constraints. We have developed an algorithm to identify simple models of large consensus networks from a known sample covariance matrix [27]. Moreover, we have developed an inference algorithm based on growing a Chow-Liu tree which is well-suited for identifying the underlying structure of large-scale dynamical systems [28].

**Part III.** Analysis of optimization algorithms from the system theoretic point of view has received significant recent attention [29–31]. In these references, the optimization algorithm is interpreted as a feedback interconnection in which the states converge to the optimal solution of the optimization problem. Moreover, centralized computation over large-scale systems are often impractical or very expensive. In this part, we propose distributed computational algorithm based on alternating direction method of multiplier and proximal augmented Lagrangian for solving non-smooth composite optimization problems over networks and provide global convergence analysis using control

theoretic techniques [98, 99, 121].

**Part IV.** Distributed feedback design and complexity constrained control are examples of problems posed within the domain of structured optimal feedback synthesis. The optimal feedback gain is typically a non-convex function of system primitives. However, in recent years, algorithms have been proposed to obtain locally optimal solutions. In applications to large-scale distributed control, the major obstacle is computational complexity. We address complexity through a combination of linear-algebraic techniques and computational methods adapted from both machine learning and reinforcement learning. Moreover, in many applications, the dynamics of the plant are unknown and only limited input-output measurements are available. Designing optimal controllers for these systems is challenging even in the absence of structural constraints. We have shown that for general classes of optimal control problems, the objective function and its gradient can be computed from data [111]. By employing incremental and stochastic gradient and ADMM based algorithms, we have developed a data-driven framework for the structured optimal control problem that does not require knowledge of the system parameters and avoids the need to solve large-scale matrical equations [122].

**1.3 Contributions of the dissertation** In this section, the structure of the dissertation is provided along with the main contributions of each part.

## Part I

**Topology design for stochastically forced undirected consensus networks.**

We have examined the problem of optimal topology design of the corresponding edge weights for undirected consensus networks. Our approach uses convex optimization to balance performance of stochastically-forced networks with the number of edges in the distributed controller. For  $\ell_1$ -regularized minimum variance optimal control problem, we have derived a Lagrange dual and exploited structure of the optimality conditions for undirected networks to develop customized algorithms that are well-suited for large problems. These are based on the proximal gradient and the proximal Newton methods. The proximal gradient algorithm is a first-order method that updates the controller graph Laplacian via the use of the soft-thresholding operator. In the proximal Newton

method, sequential quadratic approximation of the smooth part of the objective function is employed and the Newton direction is computed using cyclic coordinate descent over the set of active variables. Examples are provided to demonstrate utility of our algorithms. We have shown that proximal algorithms can solve the problems with millions of edges in the controller graph in several minutes, on a PC. Furthermore, we have specialized our algorithm to the problem of growing connected resistive networks. In this, the plant graph is connected and there are no joint edges between the plant and the controller graphs. We have exploited structure of such networks and demonstrated how additional edges can be systematically added in a computationally efficient manner.

#### **Edge addition in directed consensus networks.**

We consider the  $\ell_1$  regularized version of optimal control problem for adding edges to directed consensus networks in order to reach consensus and optimally enhance performance. Although the given plant network is not necessarily balanced, in order to reach agreement, we restrict the closed-loop graph Laplacian to be balanced. The performance is measured by the  $\mathcal{H}_2$  norm from the disturbance to the output of the closed-loop network. In general, this problem is a combinatorial search problem. We use sparsity promoting optimal control framework and introduce weighted  $\ell_1$  regularization as a proxy for promoting sparsity of the controller. By exploiting structure of the problem, we develop an algorithm based on ADMM.

## **Part II**

#### **Topology identification of undirected consensus networks via sparse inverse covariance estimation.**

We have developed a method for identifying the topology of an undirected consensus network using available statistical data. In order to promote network sparsity, we introduce a convex optimization framework aimed at finding the solution to the  $\ell_1$ -regularized maximum likelihood problem. This problem is closely related to the problem of sparse inverse covariance estimation that has received significant attention in the literature. In our setup, additional structure arises from the requirement that data is generated by an undirected consensus network. By exploiting the structure of the problem, we develop an efficient algorithm based on the sequential quadratic approximation method

in which the search direction is determined using coordinate descent with active set strategy. Several examples have been provided to illustrate utility of the method and efficiency of the customized algorithm.

#### **Topology identification via growing a Chow-Liu tree network.**

We have studied the problem of sparse topology identification of an undirected consensus network with leaders using second-order statistical data. The goal is to identify a sparse interaction topology using sample covariance matrix of the network state. We have introduced two algorithms based on regularized Gaussian maximum likelihood and growing a Chow-Liu tree. In the first algorithm, we propose a structured graphical-LASSO algorithm that uses the weighted  $\ell_1$  regularizer as a proxy for inducing sparse network topology. The other method is based on growing a Chow-Liu tree that is well-suited for identifying the underlying structure of large-scale networks. Several examples have been provided to demonstrate the performance of our framework.

### **Part III**

#### **Global exponential stability via integral quadratic constraints for proximal gradient flow and Douglas-Rachford splitting dynamics.**

We study a class of nonsmooth optimization problems in which it is desired to minimize the sum of a continuously differentiable function with a Lipschitz continuous gradient and a nondifferentiable function. For strongly convex problems, we employ the theory of integral quadratic constraints to prove global exponential stability of proximal gradient flow and Douglas-Rachford splitting dynamics. We also propose a generalization of the Polyak-Lojasiewicz condition to nonsmooth problems and demonstrate the global exponential convergence of the forward-backward envelope for the proximal gradient flow algorithm even in the absence of strong convexity.

#### **Distributed in-network optimization.**

We have studied a class of convex nonsmooth composite optimization problems in which the objective function is a combination of differentiable and nondifferentiable functions. By exploiting the structure of the problem, we have provided an equivalent consensus-based characterization and have developed an algorithm based on primal-descent dual-ascent gradient flow method. This algorithm exploits the separability of

the objective function and is well-suited for distributed implementation. Convexity of the smooth part of the objective function is utilized to prove global asymptotic stability of our algorithm. Finally, by exploiting the structure of the  $\mathcal{H}_2$  norm, we have employed this algorithm to design a sparse controller network that improves the performance of the closed-loop system in a large-scale undirected consensus network in a distributed manner. An example is provided to demonstrate the utility of the developed approach. We are currently working on implementing this algorithm in C++ and will use it to solve structured optimal control problems for large-scale systems in a distributed manner.

## Part IV

### **Data-driven proximal algorithms for the design of structured optimal feedback gains.**

We have considered the problem of designing optimal structured feedback controllers for large-scale systems. We have shown that for general classes of optimal control problems, the objective function and its gradient can be computed from data. Transformations borrowed from the theory of reinforcement learning are adapted to obtain simulation-based algorithms for computing the structured optimal  $H_2$  feedback gain. Customized proximal algorithms based on gradient descent and incremental gradient are tested in computational experiments and their relative merits are discussed. Moreover, by exploiting the structure of the  $\mathcal{H}_2$  norm, we have shown that this problem can be separated into  $N$  subproblems and that it can be solved efficiently using distributed optimization. We have utilized an ADMM-based consensus algorithm to design a sparse controller network that improves the performance of the closed-loop system. By splitting the problem into  $N$  separate subproblems over  $N$  different cores, we have implemented this algorithm in C++. Our parallel implementation can be used to solve structured optimal control problems for large-scale systems, e.g., power networks [16,18]. Computational experiments are provided to demonstrate the utility of the developed approach.

## Part I

# Distributed design and control of networks



## Chapter 2

# Topology design for stochastically forced undirected consensus networks

In this chapter, we study an optimal control problem aimed at adding a certain number of edges to an undirected network, with a known graph Laplacian, in order to optimally enhance closed-loop performance. The performance is quantified by the steady-state variance amplification of the network with additive stochastic disturbances. To promote controller sparsity, we introduce  $\ell_1$ -regularization into the optimal  $\mathcal{H}_2$  formulation and cast the design problem as a semidefinite program. We derive a Lagrange dual, provide interpretation of dual variables, and exploit structure of the optimality conditions for undirected networks to develop customized proximal gradient and Newton algorithms that are well-suited for large problems. We illustrate that our algorithms can solve the problems with more than million edges in the controller graph in a few minutes, on a PC. We also exploit structure of connected resistive networks to demonstrate how additional edges can be systematically added in order to minimize the  $\mathcal{H}_2$  norm of the closed-loop system.

**2.1 Introduction** Herein, we study an optimal control problem aimed at achieving a desired tradeoff between the network performance and communication requirements in the distributed controller. Our goal is to add a certain number of edges to

a given undirected network in order to optimally enhance the closed-loop performance. One of our key contributions is the formulation of topology design as an optimal control problem that admits convex characterization and is amenable to the development of efficient optimization algorithms. In our formulation, the plant network can contain disconnected components and optimal topology of the controller network is an integral part of the design. In general, this problem is NP-hard [33] and it amounts to an intractable combinatorial search. Several references have examined convex relaxations or greedy algorithms to design topology that optimizes algebraic connectivity [34] or network coherence [9, 35–37].

To enable design of large networks, we pay particular attention to the computational aspects of the edge-addition problem. We derive a Lagrange dual of the optimal control problem, provide interpretation of dual variables, and develop efficient proximal algorithms. Furthermore, building on preliminary work [63], we specialize our algorithms to the problem of growing connected resistive networks described in [7, 34]. In this, the plant graph is connected and inequality constraints amount to non-negativity of controller edge weights. This allows us to simplify optimality conditions and further improve computational efficiency of our customized algorithms.

Proximal gradient algorithms [76] and their accelerated variants [107] have recently found use in distributed optimization, statistics, machine learning, image and signal processing. They can be interpreted as generalization of standard gradient projection to problems with non-smooth and extended real-value objective functions. When the proximal operator is easy to evaluate, these algorithms are simple yet extremely efficient.

For networks that can contain disconnected components and non-positive edge weights, we show that the proximal gradient algorithm iteratively updates the controller graph Laplacian via convenient use of the soft-thresholding operator. This extends the Iterative Shrinkage Thresholding Algorithm (ISTA) to optimal topology design of undirected networks. In contrast to the  $\ell_1$ -regularized least-squares, however, the step-size has to be selected to guarantee positivity of the second smallest eigenvalue of the closed-loop graph Laplacian. We combine the Barzilai-Borwein (BB) step-size initialization with backtracking to achieve this goal and enhance the rate of convergence. The biggest computational challenge comes from evaluation of the objective function and its gradient. We exploit problem structure to speed up computations and save memory. Finally, for

the problem of growing connected resistive networks, the proximal algorithm simplifies to gradient projection which additionally improves the efficiency.

We also develop a customized algorithm based on the proximal Newton method. In contrast to the proximal gradient, this method sequentially employs the second-order Taylor series approximation of the smooth part of the objective function; e.g., see [123]. We use cyclic coordinate descent over the set of active variables to efficiently compute the Newton direction by consecutive minimization with respect to individual coordinates. Similar approach has been recently utilized in a number of applications, including sparse inverse covariance estimation in graphical models [124].

Both of our customized proximal algorithms significantly outperform a primal-dual IP method developed in [63]. It is worth noting that the latter is significantly faster than the general-purpose solvers. While the customized IP algorithm of [63] with a simple diagonal preconditioner can solve the problems with hundreds of thousands of edges in the controller graph in several hours, on a PC, the customized algorithms based on proximal gradient and Newton methods can solve the problems with millions of edges in several minutes. Furthermore, they are considerably faster than the greedy algorithm with efficient rank-one updates developed in [37].

Our presentation is organized as follows. In Section 2.2, we formulate the problem of optimal topology design for undirected networks subject to additive stochastic disturbances. In Section 2.3, we derive a Lagrange dual of the sparsity-promoting optimal control problem, provide interpretation of dual variables, and construct dual feasible variables from the primal ones. In Section 2.4, we develop customized algorithms based on the proximal gradient and Newton methods. In Section 2.5, we achieve additional speedup by specializing our algorithms to the problem of growing connected resistive networks. In Section 2.6, we use computational experiments to design optimal topology of a controller graph for benchmark problems and demonstrate efficiency of our algorithms. In Section 2.7, we provide a brief overview of the chapter.

**2.2 Problem formulation** We consider undirected consensus networks with  $n$  nodes

$$\dot{\psi} = -L_p \psi + u + d \quad (2.1)$$

where  $d$  and  $u$  are the exogenous disturbance and the control input, respectively,  $\psi$  is the state of the network, and  $L_p$  is a symmetric  $n \times n$  matrix that represents graph Laplacian of the open-loop system, i.e., plant. Such networks arise in applications ranging from load balancing to power systems to opinion formation to control of multi-agent systems. The goal is to improve performance of a consensus algorithm in the presence of stochastic disturbances by adding a certain number of edges (from a given set of candidate edges). We formulate this problem as a feedback design problem with

$$u = -L_x \psi$$

where the symmetric feedback-gain matrix  $L_x$  is required to have the Laplacian structure. This implies that each node in (2.1) forms control action using a weighted sum of the differences between its own state and the states of other nodes and that information is processed in a symmetric fashion. Since a nonzero  $ij$ th element of  $L_x$  corresponds to an edge between the nodes  $i$  and  $j$ , the communication structure in the controller graph is determined by the sparsity pattern of the matrix  $L_x$ .

Upon closing the loop we obtain

$$\dot{\psi} = -(L_p + L_x)\psi + d. \quad (2.2a)$$

For a given  $L_p$ , our objective is to design the topology for  $L_x$  and the corresponding edge weights  $x$  in order to achieve the desired tradeoff between controller sparsity and network performance. The performance is quantified by the steady-state variance amplification of the stochastically-forced network, from the white-in-time input  $d$  to the performance output  $\zeta$ ,

$$\zeta := \begin{bmatrix} Q^{1/2} \\ 0 \end{bmatrix} \psi + \begin{bmatrix} 0 \\ R^{1/2} \end{bmatrix} u = \begin{bmatrix} Q^{1/2} \\ -R^{1/2}L_x \end{bmatrix} \psi \quad (2.2b)$$

which penalizes deviation from consensus and control effort. Here,  $Q = Q^T \succeq 0$  and  $R = R^T \succ 0$  are the state and control weights in the standard quadratic performance index.

The interesting features of this problem come from structural restrictions on the Laplacian matrices  $L_p$  and  $L_x$ . Both of them are symmetric and are restricted to

having an eigenvalue at zero with the corresponding eigenvector of all ones,

$$L_p \mathbf{1} = 0, \quad L_x \mathbf{1} = 0. \quad (2.3)$$

Since each node uses relative information exchange with its neighbors to update its state, in the presence of white noise, the average mode  $\bar{\psi}(t) := (1/n) \mathbf{1}^T \psi(t)$  experiences a random walk and its variance increases linearly with time. To make the average mode unobservable from the performance output  $\zeta$ , the matrix  $Q$  is also restricted to having an eigenvalue at zero associated with the vector of all ones,  $Q \mathbf{1} = 0$ . Furthermore, to guarantee observability of the remaining eigenvalues of  $L_p$ , we consider state weights that are positive definite on the orthogonal complement of the subspace spanned by the vector of all ones,  $Q + (1/n) \mathbf{1} \mathbf{1}^T \succ 0$ ; e.g.,  $Q = I - (1/n) \mathbf{1} \mathbf{1}^T$  penalizes mean-square deviation from the network average.

In what follows, we express  $L_x$  as

$$L_x := \sum_{l=1}^m x_l \xi_l \xi_l^T = E \text{diag}(x) E^T \quad (2.4)$$

where  $E$  is the incidence matrix of the controller graph  $L_x$ ,  $m$  is the number of edges in  $L_x$ , and  $\text{diag}(x)$  is a diagonal matrix containing the vector of the edge weights  $x \in \mathbb{R}^m$ . The matrix  $E$  is given and it determines the set of candidate edges in controller network. This set can contain all possible edges in the network or it can only include edges that are not in the plant network. Many other options are possible as long as the union of the sets of edges in the plant and controller networks yields a connected graph. We note that the size of the set of candidate edges in controller network influences computational complexity of our algorithms.

It is desired to select a subset of edges in order to balance the closed-loop performance with the number of added edges. Vectors  $\xi_l \in \mathbb{R}^n$  determine the columns of  $E$  and they signify the connection with weight  $x_l$  between nodes  $i$  and  $j$ : the  $i$ th and  $j$ th entries of  $\xi_l$  are 1 and  $-1$  and all other entries are equal to 0. Thus,  $L_x$  given by (2.4) satisfies structural requirements on the controller graph Laplacian in (2.3) by construction.

To achieve consensus in the absence of disturbances, the closed-loop network has to be connected [2]. Equivalently, the second smallest eigenvalue of the closed-loop

graph Laplacian,  $L := L_p + L_x$ , has to be positive, i.e.,  $L$  has to be positive definite on  $\mathbb{1}^\perp$ . This amounts to positive definiteness of the “strengthened” graph Laplacian of the closed-loop network

$$G := L_p + L_x + (1/n) \mathbb{1}\mathbb{1}^T = G_p + E \text{diag}(x) E^T \succ 0 \quad (2.5a)$$

where

$$G_p := L_p + (1/n) \mathbb{1}\mathbb{1}^T. \quad (2.5b)$$

Structural restrictions (2.3) on the Laplacian matrices introduce an additional constraint on the matrix  $G$ ,

$$G \mathbb{1} = \mathbb{1}. \quad (2.5c)$$

**2.2.1 Design of optimal sparse topology** Let  $d$  be a white stochastic disturbance with zero-mean and unit variance,

$$\mathbf{E}(d(t)) = 0, \quad \mathbf{E}(d(t_1) d^T(t_2)) = I \delta(t_1 - t_2)$$

where  $\mathbf{E}$  is the expectation operator. The square of the  $\mathcal{H}_2$  norm of the transfer function from  $d$  to  $\zeta$ ,

$$\|H\|_2^2 = \lim_{t \rightarrow \infty} \mathbf{E}(\psi^T(t) (Q + L_x R L_x) \psi(t))$$

quantifies the steady-state variance amplification of closed-loop system (2.2). As noted earlier, the network average  $\bar{\psi}(t)$  corresponds to the zero eigenvalue of the graph Laplacian and it is not observable from the performance output  $\zeta$ . Thus, the  $\mathcal{H}_2$  norm is equivalently given by

$$\begin{aligned} \|H\|_2^2 &= \lim_{t \rightarrow \infty} \mathbf{E}(\tilde{\psi}^T(t) (Q + L_x R L_x) \tilde{\psi}(t)) \\ &= \text{trace}(P(Q + L_x R L_x)) = \langle P, Q + L_x R L_x \rangle \end{aligned}$$

where  $\tilde{\psi}(t)$  is the vector of deviations of the states of individual nodes from  $\bar{\psi}(t)$ ,

$$\tilde{\psi}(t) := \psi(t) - \mathbb{1} \bar{\psi}(t) = (I - (1/n) \mathbb{1}\mathbb{1}^T) \psi(t)$$

and  $P$  is the steady-state covariance matrix of  $\tilde{\psi}$ ,

$$P := \lim_{t \rightarrow \infty} \mathbf{E} \left( \tilde{\psi}(t) \tilde{\psi}^T(t) \right).$$

The above measure of the amplification of stochastic disturbances is determined by  $\|H\|_2^2 = (1/2)J(x)$ , where

$$J(x) := \left\langle (G_p + E \operatorname{diag}(x) E^T)^{-1}, Q + L_x R L_x \right\rangle. \quad (2.6)$$

It can be shown that  $J$  can be expressed as

$$J(x) = \left\langle (G_p + E \operatorname{diag}(x) E^T)^{-1}, Q_p \right\rangle + \operatorname{diag}(E^T R E)^T x - \langle R, L_p \rangle - 1 \quad (2.7)$$

with

$$Q_p := Q + (1/n) \mathbf{1}\mathbf{1}^T + L_p R L_p.$$

Note that the last two terms in (2.7) do not depend on the optimization variable  $x$  and that the term  $L_p R L_p$  in  $Q_p$  has an interesting interpretation: it determines a state-weight that guarantees inverse optimality (in LQR sense) of  $u = -L_p \psi$  for a system with no coupling between the nodes,  $\dot{\psi} = u + d$ .

We formulate the design of a controller graph that provides an optimal tradeoff between the  $\mathcal{H}_2$  performance of the closed-loop network and the controller sparsity as

$$\begin{aligned} & \underset{x}{\text{minimize}} && J(x) + \gamma \|x\|_1 \\ & \text{subject to} && G_p + E \operatorname{diag}(x) E^T \succ 0 \end{aligned} \quad (\text{SP})$$

where  $J(x)$  and  $G_p$  are given by (2.7) and (2.5b), respectively. The  $\ell_1$  norm of  $x$ ,  $\|x\|_1 := \sum_{l=1}^m |x_l|$ , is introduced as a convex proxy for promoting sparsity. In (SP), the vector of the edge weights  $x \in \mathbb{R}^m$  is the optimization variable; the problem data are the positive regularization parameter  $\gamma$ , the state and control weights  $Q$  and  $R$ , the plant graph Laplacian  $L_p$ , and the incidence matrix of the controller graph  $E$ .

The sparsity-promoting optimal control problem (SP) is a constrained optimization problem with a convex non-differentiable objective function [35] and a positive definite inequality constraint. This implies convexity of (SP). Positive definiteness of the

strengthened graph Laplacian  $G$  guarantees stability of the closed-loop network (2.2a) on the subspace  $\mathbb{1}^\perp$ , and thereby consensus in the absence of disturbances [2].

The consensus can be achieved even if some edge weights are negative [3, 5]. By expressing  $x$  as a difference between two non-negative vectors  $x_+$  and  $x_-$ , (SP) can be written as

$$\begin{aligned} & \underset{x_+, x_-}{\text{minimize}} \quad \left\langle (G_p + E \operatorname{diag}(x_+ - x_-) E^T)^{-1}, Q_p \right\rangle + \\ & \quad (\gamma \mathbb{1} + c)^T x_+ + (\gamma \mathbb{1} - c)^T x_- \\ & \text{subject to} \quad G_p + E \operatorname{diag}(x_+ - x_-) E^T \succ 0 \\ & \quad x_+ \geq 0, \quad x_- \geq 0 \end{aligned} \tag{2.8}$$

where  $c := \operatorname{diag}(E^T R E)$ . By utilizing the Schur complement, (2.8) can be cast to an SDP, and solved via standard IP method algorithms for small size networks.

### Reweighted $\ell_1$ norm

An alternative proxy for promoting sparsity is given by the weighted  $\ell_1$  norm [125],  $\|w \circ x\|_1 := \sum_{l=1}^m w_l |x_l|$  where  $\circ$  denotes elementwise product. The vector of non-negative weights  $w \in \mathbb{R}^m$  can be selected to provide better approximation of non-convex cardinality function than the  $\ell_1$  norm. An effective heuristic for weight selection is given by the iterative reweighted algorithm [125], with  $w_l$  inversely proportional to the magnitude of  $x_l$  in the previous iteration,

$$w_l^+ = 1/(|x_l| + \varepsilon). \tag{2.9}$$

This puts larger emphasis on smaller optimization variables, where a small positive parameter  $\varepsilon$  ensures that  $w_l^+$  is well-defined. If the weighted  $\ell_1$  norm is used in (SP), the vector of all ones  $\mathbb{1}$  should be replaced by the vector  $w$  in (2.8).

**2.2.2 Structured optimal control problem: debiasing step** After the structure of the controller graph Laplacian  $L_x$  has been designed, we fix the structure of  $L_x$  and optimize the corresponding edge weights. This “polishing” or “debiasing” step is used to improve the performance relative to the solution of the regularized optimal control problem (SP); see [126, Section 6.3.2] for additional information. The structured



optimal control problem is obtained by eliminating the columns from the incidence matrix  $E$  that correspond to zero elements in the vector of the optimal edge weights  $x^*$  resulting from (SP). This yields a new incidence matrix  $\hat{E}$  and leads to

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad \left\langle \left( G_p + \hat{E} \text{diag}(x) \hat{E}^T \right)^{-1}, Q_p \right\rangle + \text{diag} \left( \hat{E}^T R \hat{E} \right)^T x \\ & \text{subject to} \quad G_p + \hat{E} \text{diag}(x) \hat{E}^T \succ 0. \end{aligned}$$

Alternatively, this optimization problem is obtained by setting  $\gamma = 0$  in (SP) and by replacing the incidence matrix  $E$  with  $\hat{E}$ . The solution provides the optimal vector of the edge weights  $x$  for the controller graph Laplacian with the desired structure.

**2.2.3 Gradient and Hessian of  $J(x)$**  We next summarize the first- and second-order derivatives of the objective function  $J$ , given by (2.7), with respect to the vector of the edge weights  $x$ . The second-order Taylor series approximation of  $J(x)$  around  $\bar{x} \in \mathbb{R}^m$  is given by

$$J(\bar{x} + \tilde{x}) \approx J(\bar{x}) + \nabla J(\bar{x})^T \tilde{x} + \frac{1}{2} \tilde{x}^T \nabla^2 J(\bar{x}) \tilde{x}.$$

For related developments we refer the reader to [7].

**Proposition 1.** *The gradient and the Hessian of  $J$  at  $\bar{x} \in \mathbb{R}^m$  are determined by*

$$\begin{aligned} \nabla J(\bar{x}) &= -\text{diag} \left( E^T (Y(\bar{x}) - R) E \right) \\ \nabla^2 J(\bar{x}) &= H_1(\bar{x}) \circ H_2(\bar{x}) \end{aligned}$$

where

$$\begin{aligned} Y(\bar{x}) &:= (G_p + E D_{\bar{x}} E^T)^{-1} Q_p (G_p + E D_{\bar{x}} E^T)^{-1} \\ H_1(\bar{x}) &:= E^T Y(\bar{x}) E \\ H_2(\bar{x}) &:= E^T (G_p + E D_{\bar{x}} E^T)^{-1} E \\ D_{\bar{x}} &:= \text{diag}(\bar{x}). \end{aligned}$$

**2.3 Dual problem** Herein, we study the Lagrange dual of the sparsity-promoting optimal control problem (2.8), provide interpretation of dual variables, and construct dual feasible variables from primal feasible variables. Since minimization of

the Lagrangian associated with (2.8) does not lead to an explicit expression for the dual function, we introduce an auxiliary variable  $G$  and find the dual of

$$\begin{aligned}
& \underset{G, x_{\pm}}{\text{minimize}} \quad \langle G^{-1}, Q_p \rangle + (\gamma \mathbf{1} + c)^T x_+ + (\gamma \mathbf{1} - c)^T x_- \\
& \text{subject to} \quad G - G_p - E \text{diag}(x_+ - x_-) E^T = 0 \\
& \quad \quad \quad G \succ 0, \quad x_+ \geq 0, \quad x_- \geq 0.
\end{aligned} \tag{P}$$

In (P),  $G$  represents the “strengthened” graph Laplacian of the closed-loop network and the equality constraint comes from (2.5a). As we show next, the Lagrange dual of the primal optimization problem (P) admits an explicit characterization.

**Proposition 2.** *The Lagrange dual of the primal optimization problem (P) is given by*

$$\begin{aligned}
& \underset{Y}{\text{maximize}} \quad 2 \text{trace} \left( (Q_p^{1/2} Y Q_p^{1/2})^{1/2} \right) - \langle Y, G_p \rangle \\
& \text{subject to} \quad \|\text{diag}(E^T(Y - R)E)\|_{\infty} \leq \gamma \\
& \quad \quad \quad Y \succ 0, \quad Y \mathbf{1} = \mathbf{1}
\end{aligned} \tag{D}$$

where  $Y = Y^T \in \mathbb{R}^{n \times n}$  is the dual variable associated with the equality constraint in (P). The duality gap is

$$\eta = y_+^T x_+ + y_-^T x_- = \mathbf{1}^T (y_+ \circ x_+ + y_- \circ x_-) \tag{2.10}$$

where

$$y_+ = \gamma \mathbf{1} - \text{diag}(E^T(Y - R)E) \geq 0 \tag{2.11a}$$

$$y_- = \gamma \mathbf{1} + \text{diag}(E^T(Y - R)E) \geq 0. \tag{2.11b}$$

are the Lagrange multipliers associated with elementwise inequality constraints in (P).

*Proof.* The Lagrangian of (P) is given by

$$\begin{aligned}
\mathcal{L} = & \langle G^{-1}, Q_p \rangle + \langle Y, G \rangle - \langle Y, G_p \rangle + \\
& (\gamma \mathbf{1} - \text{diag}(E^T(Y - R)E) - y_+)^T x_+ + \\
& (\gamma \mathbf{1} + \text{diag}(E^T(Y - R)E) - y_-)^T x_-.
\end{aligned} \tag{2.12}$$

Note that no Lagrange multiplier is assigned to the positive definite constraint on  $G$  in  $\mathcal{L}$ . Instead, we determine conditions on  $Y$  and  $y_{\pm}$  that guarantee  $G \succ 0$ .

Minimizing  $\mathcal{L}$  with respect to  $G$  yields

$$G^{-1} Q_p G^{-1} = Y \quad (2.13a)$$

or, equivalently,

$$G = Q_p^{1/2} \left( Q_p^{1/2} Y Q_p^{1/2} \right)^{-1/2} Q_p^{1/2}. \quad (2.13b)$$

Positive definiteness of  $G$  and  $Q_p$  implies  $Y \succ 0$ . Furthermore, since  $Q_p \mathbf{1} = \mathbf{1}$ , from (2.5c) and (2.13a) we have

$$Y \mathbf{1} = \mathbf{1}.$$

Similarly, minimization with respect to  $x_+$  and  $x_-$  leads to (2.11a) and (2.11a). Thus, non-negativity of  $y_+$  and  $y_-$  amounts to

$$-\gamma \mathbf{1} \leq \text{diag} \left( E^T (Y - R) E \right) \leq \gamma \mathbf{1}$$

or, equivalently,

$$\| \text{diag} \left( E^T (Y - R) E \right) \|_{\infty} \leq \gamma.$$

Substitution of (2.13) and (2.11) into (2.12) eliminates  $y_+$  and  $y_-$  from the dual problem. We can thus represent the dual function,  $\inf_{G, x_{\pm}} \mathcal{L}(G, x_{\pm}; Y, y_{\pm})$ , as

$$2 \text{trace} \left( (Q_p^{1/2} Y Q_p^{1/2})^{1/2} \right) - \langle Y, G_p \rangle$$

which allows us to bring the dual of (P) to (D).  $\square$

Any dual feasible  $Y$  can be used to obtain a lower bound on the optimal value of the primal problem (P). Furthermore, the difference between the objective functions of the primal (evaluated at the primal feasible  $(G, x_{\pm})$ ) and dual (evaluated at the dual feasible  $Y$ ) problems yields expression (2.10) for the duality gap  $\eta$ , where  $y_+$  and  $y_-$  are given by (2.11a) and (2.11b). The duality gap can be used to estimate distance to optimality.

Strong duality follows from Slater's theorem [126], i.e., convexity of the primal problem (P) and strict feasibility of the constraints in (P). This implies that at optimality, the duality gap  $\eta$  for the primal problem (P) and the dual problem (D) is zero. Furthermore, if  $(G^*, x_\pm^*)$  are optimal points of (P), then  $Y^* = (G^*)^{-1} Q_p (G^*)^{-1}$  is the optimal point of (D). Similarly, if  $Y^*$  is the optimal point of (D),

$$G^* = Q_p^{1/2} \left( Q_p^{1/2} Y^* Q_p^{1/2} \right)^{-1/2} Q_p^{1/2}$$

is the optimal point of (P). The optimal vector of the edge weights  $x^*$  is determined by the non-zero off-diagonal elements of the controller graph Laplacian,  $L_x^* = G^* - G_p$ .

### Interpretation of dual variables

For electrical networks, the dual variables have appealing interpretations. Let  $\iota \in \mathbb{R}^n$  be a random current injected into the resistor network satisfying

$$\mathbf{1}^T \iota = 0, \quad \mathbf{E}(\iota) = 0, \quad \mathbf{E}(\iota \iota^T) = Q + L_p R L_p.$$

The vector of voltages  $\vartheta \in \mathbb{R}^m$  across the edges of the network is then given by  $\vartheta = E^T G^{-1} \iota$ . Furthermore, since

$$\mathbf{E}(\vartheta \vartheta^T) = E^T G^{-1} \mathbf{E}(\iota \iota^T) G^{-1} E = E^T Y E,$$

the dual variable  $Y$  is related to the covariance matrix of voltages across the edges. Moreover, (2.11) implies that  $y_+$  and  $y_-$  quantify the deviations between variances of edge voltages from their respective upper and lower bounds.

**Remark 1.** For a primal feasible  $x$ ,  $Y$  resulting from (2.13a) with  $G$  given by (2.5a) may not be dual feasible. Let

$$\hat{Y} := \beta Y + \frac{1 - \beta}{n} \mathbf{1} \mathbf{1}^T \tag{2.14a}$$

and let the control weight be  $R = r I$  with  $r > 0$ . If

$$\beta \leq \frac{\gamma + 2r}{\|\text{diag}(E^T(Y - R)E)\|_\infty + 2r} \tag{2.14b}$$

then  $\hat{Y}$  satisfies the inequality constraint in (D) and it is thus dual feasible.

**2.4 Customized algorithms** We next exploit the structure of the sparsity-promoting optimal control problem (SP) and develop customized algorithms based on the proximal gradient and Newton methods. The proximal gradient algorithm is a first-order method that uses a simple quadratic approximation of  $J$  in (SP). This yields an explicit update of the vector of the edge weights via application of the soft-thresholding operator. In the proximal Newton method a sequential quadratic approximation of the smooth part of the objective function in (SP) is used and the search direction is efficiently computed via cyclic coordinate descent over the set of active variables.

**2.4.1 Proximal gradient method** We next use the proximal gradient method to solve (SP). A simple quadratic approximation of  $J(x)$  around the current iterate  $x^k$ ,

$$J(x) \approx J(x^k) + \nabla J(x^k)^T (x - x^k) + \frac{1}{2\alpha_k} \|x - x^k\|_2^2$$

is substituted to (SP) to obtain

$$x^{k+1} = \underset{x}{\operatorname{argmin}} g(x) + \frac{1}{2\alpha_k} \|x - (x^k - \alpha_k \nabla J(x^k))\|_2^2.$$

Here,  $\alpha_k$  is the step-size and the update is determined by the proximal operator of the function  $\alpha_k g$ ,

$$x^{k+1} = \mathbf{prox}_{\alpha_k g} \left( x^k - \alpha_k \nabla J(x^k) \right).$$

In particular, for  $g(x) = \gamma \|x\|_1$ , we have

$$x^{k+1} = \mathcal{S}_{\gamma\alpha_k} \left( x^k - \alpha_k \nabla J(x^k) \right)$$

where  $\mathcal{S}_\kappa(y) = \operatorname{sign}(y) \max(|y| - \kappa, 0)$  is the soft-thresholding function.

The proximal gradient algorithm converges with rate  $O(1/k)$  if  $\alpha_k < 1/L$ , where  $L$  is the Lipschitz constant of  $\nabla J$  [76, 107]. It can be shown that  $\nabla J$  is Lipschitz continuous but, since it is challenging to explicitly determine  $L$ , we adjust  $\alpha_k$  via backtracking. To provide a better estimate of  $L$ , we initialize  $\alpha_k$  using the Barzilai-Borwein (BB) method which provides an effective heuristic for approximating the Hessian of the function  $J$

via the scaled version of the identity [127],  $(1/\alpha_k)I$ . At the  $k$ th iteration, the initial BB step-size  $\alpha_{k,0}$ ,

$$\alpha_{k,0} := \frac{\|x^k - x^{k-1}\|_2^2}{(x^{k-1} - x^k)^T (\nabla J(x^{k-1}) - \nabla J(x^k))} \quad (2.15)$$

is adjusted via backtracking until the inequality constraint in (SP) is satisfied and

$$J(x^{k+1}) \leq J(x^k) + \nabla J(x^k)^T (x^{k+1} - x^k) + \frac{1}{2\alpha_k} \|x^{k+1} - x^k\|_2^2.$$

Since  $J$  is continuously differentiable with Lipschitz continuous gradient, this inequality holds for any  $\alpha_k < 1/L$  and the algorithm converges sub-linearly [107]. This condition guarantees that objective function decreases at every iteration. Our numerical experiments in Section 2.6 suggest that BB step-size initialization significantly enhances the rate of convergence.

**Remark 2.** *The biggest computational challenge comes from evaluation of the objective function and its gradient. Since the inverse of the strengthened graph Laplacian  $G$  has to be computed, with direct computations these evaluations take  $O(n^3)$  and  $O(nm^2)$  flops, respectively. However, by exploiting the problem structure,  $\nabla J$  can be computed more efficiently. The main cost arises in the computation of  $\text{diag}(E^T Y E)$ . We instead compute it using  $\text{sum}(E^T \circ (Y E))$  which takes  $O(n^2 m)$  operations. Here,  $\text{sum}(A)$  is a vector which contains summation of each row of the matrix  $A$  in its entries. For networks with  $m \gg n$  this leads to significant speed up. Moreover, in contrast to direct computation, we do not need to store the  $m \times m$  matrix  $E^T Y E$ . Only formation of the columns is required which offers memory saving.*

**2.4.2 Proximal Newton method** In contrast to the proximal gradient algorithm, the proximal Newton method benefits from second-order Taylor series expansion of the smooth part of the objective function in (SP). Herein, we employ cyclic coordinate descent over the set of active variables to efficiently compute the Newton direction.

By approximating the smooth part of the objective function  $J$  in (SP) with the second-order Taylor series expansion around the current iterate  $\bar{x}$ ,

$$J(\bar{x} + \tilde{x}) \approx J(\bar{x}) + \nabla J(\bar{x})^T \tilde{x} + \frac{1}{2} \tilde{x}^T \nabla^2 J(\bar{x}) \tilde{x}$$

the problem (SP) becomes

$$\begin{aligned} & \underset{\tilde{x}}{\text{minimize}} \quad \nabla J(\bar{x})^T \tilde{x} + \frac{1}{2} \tilde{x}^T \nabla^2 J(\bar{x}) \tilde{x} + \gamma \|\bar{x} + \tilde{x}\|_1 \\ & \text{subject to} \quad G_p + E \text{diag}(\bar{x} + \tilde{x}) E^T \succ 0. \end{aligned} \quad (2.16)$$

Let  $\tilde{x}$  denote the current iterate approximating the Newton direction. By perturbing  $\tilde{x}$  in the direction of the  $i$ th standard basis vector  $\mathbf{e}_i$  in  $\mathbb{R}^m$ , the objective function in (2.16) becomes

$$\nabla J(\bar{x})^T (\tilde{x} + \delta_i \mathbf{e}_i) + \frac{1}{2} (\tilde{x} + \delta_i \mathbf{e}_i)^T \nabla^2 J(\bar{x}) (\tilde{x} + \delta_i \mathbf{e}_i) + \gamma |\bar{x}_i + \tilde{x}_i + \delta_i|.$$

Elimination of constant terms allows us to bring (2.16) into

$$\underset{\delta_i}{\text{minimize}} \quad \frac{1}{2} a_i \delta_i^2 + b_i \delta_i + \gamma |c_i + \delta_i| \quad (2.17)$$

where the optimization variable is the scalar  $\delta_i$  and  $(a_i, b_i, c_i, \bar{x}_i, \tilde{x}_i)$  are the problem data with

$$\begin{aligned} a_i &:= \mathbf{e}_i^T \nabla^2 J(\bar{x}) \mathbf{e}_i \\ b_i &:= (\nabla^2 J(\bar{x}) \mathbf{e}_i)^T \tilde{x} + \mathbf{e}_i^T \nabla J(\bar{x}) \\ c_i &:= \bar{x}_i + \tilde{x}_i. \end{aligned}$$

The explicit solution to (2.17) is given by

$$\delta_i = -c_i + \mathcal{S}_{\gamma/a_i}(c_i - b_i/a_i).$$

After the Newton direction  $\tilde{x}$  has been computed, we determine the step-size  $\alpha$  via backtracking. This guarantees positive definiteness of the strengthened graph Laplacian and sufficient decrease of the objective function. We use generalization of Armijo rule [128] to find an appropriate step-size  $\alpha$  such that  $G_p + E \text{diag}(\bar{x} + \alpha \tilde{x}) E^T$  is positive definite matrix and

$$J(\bar{x} + \alpha \tilde{x}) + \gamma \|\bar{x} + \alpha \tilde{x}\|_1 \leq J(\bar{x}) + \gamma \|\bar{x}\|_1 + \alpha \sigma (\nabla J(\bar{x})^T \tilde{x} + \gamma \|\bar{x} + \tilde{x}\|_1 - \gamma \|\bar{x}\|_1).$$

**Remark 3.** The parameter  $a_i$  in (2.17) is determined by the  $i$ th diagonal element of the

Hessian  $\nabla^2 J(\bar{x})$ . On the other hand, the  $i$ th column of  $\nabla^2 J(\bar{x})$  and the  $i$ th element of the gradient vector  $\nabla J(\bar{x})$  enter into the expression for  $b_i$ . All of these can be obtained directly from  $\nabla^2 J(\bar{x})$  and  $\nabla J(\bar{x})$  and forming them does not require any multiplication. Computation of a single vector inner product between the  $i$ th column of the Hessian and  $\tilde{x}$  is required in  $b_i$ , which typically takes  $O(m)$  operations. To avoid direct multiplication, in each iteration after finding  $\delta_i$ , we update the vector  $\nabla^2 J(\bar{x})^T \tilde{x}$  using the correction term  $\delta_i(E^T Y E_i) \circ ((G^{-1} E_i)^T E)^T$  and take its  $i$ th element to form  $b_i$ . Here,  $E_i$  is the  $i$ th column of the incidence matrix of the controller graph. This also avoids the need to store the Hessian of  $J$ , which is an  $m \times m$  matrix, thereby leading to a significant memory saving.

**Remark 4.** Active set strategy is an effective means for determining the directions that do not need to be updated in the coordinate descent algorithm. At each outer iteration, we classify the variable as either active or inactive based on the values of  $\bar{x}_i$  and the  $i$ th component of the gradient vector  $\nabla J(\bar{x})$ . For  $g(x) = \gamma \|x\|_1$ , the  $i$ th search direction is inactive if

$$\bar{x}_i = 0 \quad \text{and} \quad |e_i^T \nabla J(\bar{x})| < \gamma - \epsilon$$

and it is active otherwise. Here,  $\epsilon > 0$  is a small number (e.g.,  $\epsilon = 0.0001\gamma$ ). The Newton direction is then obtained by solving the optimization problem over the set of active variables. This significantly improves algorithmic efficiency for large values of the regularization parameter  $\gamma$ .

### Convergence analysis

In (SP),  $J(x)$  is smooth for  $G_p + E \text{diag}(x) E^T \succ 0$  and the non-smooth part is given by the  $\ell_1$  norm of  $x$ . The objective function of the form  $J(x) + g(x)$  was studied in [124], where  $J$  is smooth over the positive definite cone and  $g$  is a separable non-differentiable function. Theorem 16 from [124] thus implies super-linear (i.e., quadratic) convergence rate of the quadratic approximation method for (SP).



Table 2.1: Comparison of our customized proximal algorithms

Algorithm	proximal gradient	proximal Newton
Order	1st	2nd
Search direction	explicit update	coordinate descent
Speed-up strategy	BB step-size initialization	active set strategy
Memory	no storage of $m \times m$ matrices	no storage of $m \times m$ matrices
Most expensive part	$O(n^2m)$	$O(m^2)$
Convergence rate	linear	super-linear (quadratic)

Table 2.2: Algorithmic properties of the primal dual IP method.

Algorithm	primal-dual IP method
Order	2nd
Search direction	PCG
Speed-up strategy	PCG with preconditioner
Memory	no storage of $m \times m$ matrices
Most expensive part	$O(m^3)$
Convergence rate	super-linear

### Stopping criteria

The norms of the primal and dual residuals  $r_p$  and  $r_d^\pm$  as well as the duality gap  $\eta$  are used as stopping criteria. In contrast to the stopping criteria available in the literature, this choice enables fair comparison of the algorithms. We use (2.14) to construct a dual feasible  $\hat{Y}$  and obtain  $y_\pm$  from (2.11), (2.10) to compute the duality gap  $\eta$ , and

$$\begin{aligned}
r_p(x, x_\pm) &:= x - x_+ + x_- \\
r_d^+(x, y_+) &:= \gamma \mathbb{1} - \text{diag} \left( E^T (\hat{Y} - R) E \right) - y_+ \\
r_d^-(x, y_-) &:= \gamma \mathbb{1} + \text{diag} \left( E^T (\hat{Y} - R) E \right) - y_-.
\end{aligned}$$

to determine the primal and dual residuals.

### Comparison of algorithms

Tables 2.1 and 2.2 compare and contrast features of our customized proximal algorithms and the algorithm based on the primal-dual IP method developed in [63].

**2.5 Growing connected resistive networks** The problem of optimal topology design for stochastically-forced networks has many interesting variations. An important class is given by resistive networks in which all edge weights are non-negative,  $x \geq 0$ . Here, we study the problem of growing connected resistive networks; e.g., see [34]. In this, the plant graph is connected and there are no joint edges between the plant and the controller graphs. Our objective is to enhance the closed-loop performance by adding a small number of edges. As we show below, inequality constraints in this case amount to non-negativity of controller edge weights. This simplifies optimality conditions and enables further improvement of the computational efficiency of our customized algorithms.

The restriction on connected plant graphs implies positive definiteness of the strengthened graph Laplacian of the plant,  $G_p = L_p + (1/n) \mathbf{1}\mathbf{1}^T \succ 0$ . Thus,  $G_p + E \operatorname{diag}(x) E^T$  is always positive definite for connected resistive networks and (SP) simplifies to

$$\underset{x}{\text{minimize}} \quad f(x) + g(x) \tag{2.18}$$

where

$$f(x) := J(x) + \gamma \mathbf{1}^T x$$

and  $g(x)$  is the indicator function for the non-negative orthant,

$$g(x) := I_+(x) = \begin{cases} 0, & x \geq 0 \\ +\infty, & \text{otherwise.} \end{cases}$$

As in Section 2.3, in order to determine the Lagrange dual of the optimization problem (2.18), we introduce an additional optimization variable  $G$  and rewrite (2.18) as

$$\begin{aligned} & \underset{G, x}{\text{minimize}} \quad \langle G^{-1}, Q_p \rangle + (\gamma \mathbf{1} + \operatorname{diag}(E^T R E))^T x \\ & \text{subject to} \quad G - G_p - E \operatorname{diag}(x) E^T = 0 \\ & \quad \quad \quad x \geq 0. \end{aligned} \tag{P1}$$

**Proposition 3.** *The Lagrange dual of the primal optimization problem (P1) is given*

by

$$\begin{aligned}
& \underset{Y}{\text{maximize}} && 2 \operatorname{trace} \left( (Q_p^{1/2} Y Q_p^{1/2})^{1/2} \right) - \langle Y, G_p \rangle \\
& \text{subject to} && \operatorname{diag} (E^T (Y - R) E) \leq \gamma \mathbf{1} \\
& && Y \succ 0, \quad Y \mathbf{1} = \mathbf{1}
\end{aligned} \tag{D1}$$

where  $Y$  is the dual variable associated with the equality constraint in (P1). The duality gap is

$$\eta = y^T x = \mathbf{1}^T (y \circ x) \tag{2.19}$$

where

$$y := \gamma \mathbf{1} - \operatorname{diag} (E^T (Y - R) E) \geq 0 \tag{2.20}$$

represents the dual variable associated with the non-negativity constraint on the vector of the edge weights  $x$ .

**Remark 5.** For connected resistive networks with the control weight  $R = r I$ ,  $\hat{Y}$  given by (2.14a) is dual feasible if

$$\beta \leq \frac{\gamma + 2r}{\max (\operatorname{diag} (E^T (Y - R) E)) + 2r}. \tag{2.21}$$

**2.5.1 Proximal gradient method** Using a simple quadratic approximation of the smooth part of the objective function  $f$  around the current iterate  $x^k$

$$f(x) \approx f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2\alpha_k} \|x - x^k\|_2^2$$

the optimal solution of (2.18) is determined by the proximal operator of the function  $g(x) = I_+(x)$ ,

$$x^{k+1} = \left( x^k - \alpha_k \nabla f(x^k) \right)_+$$

where  $(\cdot)_+$  is the projection on the non-negative orthant. Thus, the action of the proximal operator is given by the projected gradient.

As in Section 2.4.1, we initialize  $\alpha_k$  using the BB heuristics but we skip the backtracking step here and employ a non-monotone BB scheme [129, 130]. The effectiveness

of this strategy has been established on quadratic problems [127, 129], but its convergence in general is hard to prove. In Section 2.6, we demonstrate efficiency of this approach.

**2.5.2 Proximal Newton method** We next adjust the customized algorithm based on proximal Newton method for growing connected resistive networks. We approximate the smooth part of the objective function  $f$  in (2.18) using the second-order Taylor series expansion around the current iterate  $\bar{x}$ ,

$$f(\bar{x} + \tilde{x}) \approx f(\bar{x}) + \nabla f(\bar{x})^T \tilde{x} + \frac{1}{2} \tilde{x}^T \nabla^2 f(\bar{x}) \tilde{x}$$

and rewrite (2.18) as

$$\begin{aligned} & \underset{\tilde{x}}{\text{minimize}} \quad \nabla f(\bar{x})^T \tilde{x} + \frac{1}{2} \tilde{x}^T \nabla^2 f(\bar{x}) \tilde{x} \\ & \text{subject to} \quad \bar{x} + \tilde{x} \geq 0. \end{aligned} \tag{2.22}$$

By perturbing  $\tilde{x}$  in the direction of the  $i$ th standard basis vector  $\mathbf{e}_i$  in  $\mathbb{R}^m$ ,  $\tilde{x} + \delta_i \mathbf{e}_i$ , the objective function in (2.22) becomes

$$\nabla f(\bar{x})^T (\tilde{x} + \delta_i \mathbf{e}_i) + \frac{1}{2} (\tilde{x} + \delta_i \mathbf{e}_i)^T \nabla^2 f(\bar{x}) (\tilde{x} + \delta_i \mathbf{e}_i).$$

Elimination of constant terms allows us to bring (2.22) into

$$\begin{aligned} & \underset{\delta_i}{\text{minimize}} \quad \frac{1}{2} a_i \delta_i^2 + b_i \delta_i \\ & \text{subject to} \quad \bar{x}_i + \tilde{x}_i + \delta_i \geq 0. \end{aligned} \tag{2.23}$$

The optimization variable is the scalar  $\delta_i$  and  $a_i$ ,  $b_i$ ,  $\bar{x}_i$ , and  $\tilde{x}_i$  are the problem data with

$$\begin{aligned} a_i &:= \mathbf{e}_i^T \nabla^2 f(\bar{x}) \mathbf{e}_i \\ b_i &:= (\nabla^2 f(\bar{x}) \mathbf{e}_i)^T \tilde{x} + \mathbf{e}_i^T \nabla f(\bar{x}) \end{aligned}$$

The explicit solution to (2.23) is given by

$$\delta_i = \begin{cases} -b_i/a_i, & \bar{x}_i + \tilde{x}_i - b_i/a_i \geq 0 \\ -(\bar{x}_i + \tilde{x}_i), & \text{otherwise.} \end{cases}$$

After the Newton direction  $\tilde{x}$  has been computed, we determine the step-size  $\alpha$  via backtracking. This guarantees positivity of the updated vector of the edge weights,  $\bar{x} + \alpha\tilde{x}$ , and sufficient decrease of the objective function,  $f(\bar{x} + \alpha\tilde{x}) \leq f(\bar{x}) + \alpha \sigma \nabla f(\bar{x})^T \tilde{x}$ .

**Remark 6.** *As in Section 2.4.2, we use an active set strategy to identify the directions that do not need to be updated in the coordinate descent algorithm. For  $g(x) = I_+(x)$ , the  $i$ th search direction is inactive if  $\{\bar{x}_i = 0 \text{ and } e_i^T \nabla f(\bar{x}) \geq 0\}$  and it is active otherwise.*

### Stopping criteria

The norm of the dual residual,  $r_d$ , and the duality gap,  $\eta$ , are used as stopping criteria. The dual variable  $y$  is obtained from (2.20) where  $\hat{Y}$  is given by (2.14a) and  $\beta$  satisfies (2.21). At each iteration,  $\eta$  is evaluated using (2.19) and the dual residual is determined by

$$r_d(x, y) := \gamma \mathbb{1} - \text{diag}(E^T(Y(x) - R)E) - y.$$

**2.6 Computational experiments** We next provide examples and evaluate performance of our customized algorithms. Algorithm proxBB represents proximal gradient method with BB step-size initialization and proxN identifies proximal Newton method in which the search direction is found via coordinate descent. Performance is compared with the PCG-based primal-dual IP method of [63] and the greedy algorithm of [37]. We have implemented all algorithms in MATLAB and executed tests on a 3.4 GHz Core(TM) i7-3770 Intel(R) machine with 16GB RAM.

In all examples, we set  $R = I$  and choose the state weight that penalizes the mean-square deviation from the network average,  $Q = I - (1/n)\mathbb{1}\mathbb{1}^T$ . The absolute value of the dual residual,  $r_d$ , and the duality gap,  $\eta$ , are used as stopping criteria. We set the tolerances for  $r_d$  and  $\eta$  to  $10^{-3}$  and  $10^{-4}$ , respectively. Finally, for connected plant networks

$$\gamma_{\max} := \|\text{diag}(E^T G_p^{-1} Q G_p^{-1} E)\|_{\infty}$$

identifies the value of the regularization parameter  $\gamma$  for which all edge weights in the controller graph are equal to zero.

Table 2.3: Comparison of algorithms (solve times in seconds/number of iterations) for the problem of growing connected resistive Erdős-Rényi networks with different number of nodes  $n$ , edge probability  $1.05 \log(n)/n$ , and  $\gamma = 0.8 \gamma_{\max}$ .

number of nodes	$n = 300$	$n = 700$	$n = 1000$
number of edges	$m = 43986$	$m = 242249$	$m = 495879$
IP (PCG)	16.499/8	394.256/13	1014.282/13
proxBB	1.279/11	15.353/11	55.944/13
proxN	1.078/4	11.992/4	34.759/4

Table 2.4: Table 2.3 continued

number of nodes	$n = 1300$	$n = 1500$
number of edges	$m = 839487$	$m = 1118541$
IP (PCG)	15948.164/13	179352.208/14
proxBB	157.305/16	239.567/16
proxN	82.488/4	124.307/4

Additional information about our computational experiments, along with MATLAB source codes, can be found at:

[www.ece.umn.edu/~mihailo/software/graphsp/](http://www.ece.umn.edu/~mihailo/software/graphsp/)

**2.6.1 Performance comparison** In what follows, the incidence matrix of the controller graph is selected to satisfy the following requirements: (i) in the absence of the sparsity-promoting term, the closed-loop network is given by a complete graph; and (ii) there are no joint edges between the plant and the controller graphs.

We first solve the problem (P1) for growing connected resistive Erdős-Rényi networks with different number of nodes. The generator of the plant dynamics is given by an undirected unweighted graph with edge probability  $1.05 \log(n)/n$ . Tables 2.3 and 2.4 compare our customized algorithms in terms of speed and the number of iterations. Even for small networks, proximal methods are significantly faster than the IP method and proxN takes smaller number of iterations and converges quicker than proxBB. For a larger network (with 1500 nodes and 1118541 edges in the controller graph), it takes about 50 hours for the PCG-based IP method to solve the problem. In contrast, proxN and proxBB converge in about 2 and 4 minutes, respectively.

Figure 2.1 compares our proximal gradient algorithm with the fast greedy algorithm

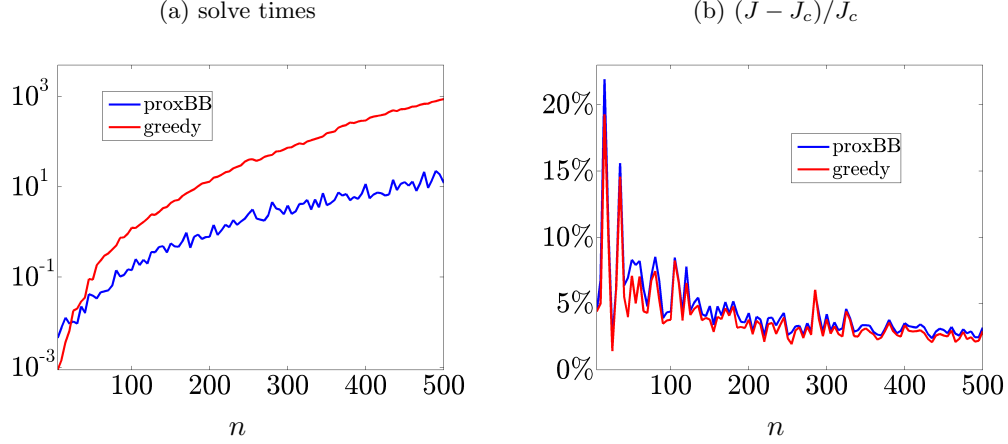


Figure 2.1: (a) Solve times (in seconds); and (b) performance degradation (in percents) of proximal gradient and greedy algorithms relative to the optimal centralized controller.

of [37]. We solve problem (P1) for Erdős-Rényi networks with different number of nodes ( $n = 5$  to 500) and  $\gamma = 0.4\gamma_{\max}$ . After proxBB identifies the edges in the controller graph, we use the greedy method to select the same number of edges. Finally, we polish the identified edge weights for both methods. Figure 2.1a shows the solve times (in seconds) versus the number of nodes. As the number of nodes increases the proximal algorithm significantly outperforms the fast greedy method. Relative to the optimal centralized controller, both methods yield similar performance degradation of the closed-loop network; see Fig. 2.1b.

**2.6.2 Large-scale Facebook network** To evaluate effectiveness of our algorithms on large networks, we solve the problem of growing a network of friendships. In such social networks, nodes denote people and edges denote friendships. There is an edge between two nodes if two people are friends. The network is obtained by examining social network of 10 users (the so-called ego nodes); all other nodes are friends to at least one of these ego nodes [131]. The resulting network is undirected and unweighted with 4039 nodes and 88234 edges; the data is available at <http://snap.stanford.edu/data/>. Our objective is to improve performance by adding a small number of extra edges. We assume that people can only form friendships with friends of their friends. This restricts the number of potential edges in the controller graph to 1358067.

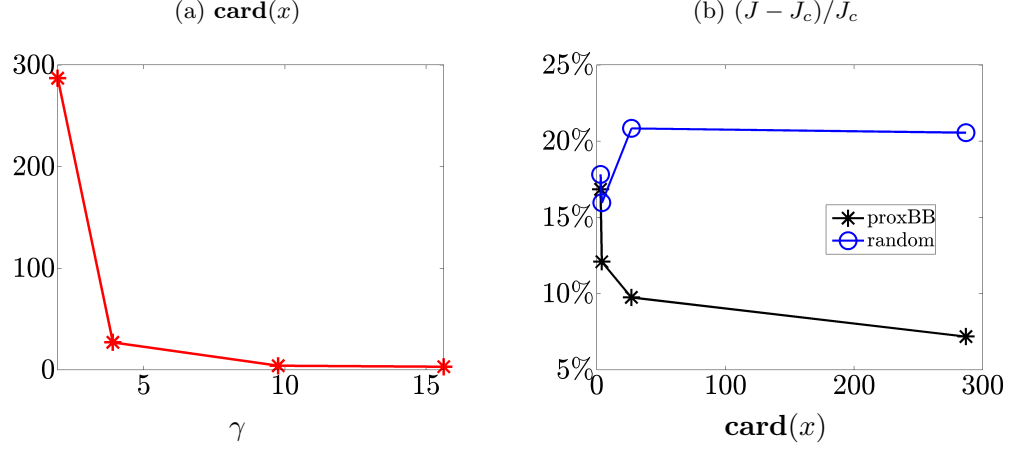


Figure 2.2: (a) Sparsity level; and (b) optimal tradeoff curves resulting from the application of proximal gradient algorithm and a heuristic strategy for the Facebook network.

To avoid memory issues, we have implemented our algorithms in C++. For  $\gamma = c \gamma_{\max}$  with  $c = \{0.1, 0.2, 0.5, 0.8\}$  and  $\gamma_{\max} = 19.525$ , the proximal gradient algorithm computes the solution in about 10, 2.6, 0.87, and 0.43 hours, respectively. After designing the topology of the controller graph, we optimize the resulting edge weights via polishing.

Figure 2.2a shows that the number of nonzero elements in the vector  $x$  decreases as  $\gamma$  increases and Fig. 2.2b illustrates that the  $\mathcal{H}_2$  performance deteriorates as the number of nonzero elements in  $x$  decreases. In particular, for  $\gamma = 0.8 \gamma_{\max}$ , the identified sparse controller has only 3 nonzero elements (it uses only 0.0002% of the potential edges). Relative to the optimal centralized controller, this controller degrades performance by 16.842%,  $(J - J_c)/J_c = 16.842\%$ .

In all of our experiments, the added links with the largest edge weights connect either the ego nodes to each other or three non-ego nodes to the ego nodes. Thus, our method recognizes significance of the ego nodes and identifies non-ego nodes that play an important role in improving performance.

We compare performance of the identified controller to a heuristic strategy that is described next. The controller graph contains 16 potential edges between ego nodes. If the number of edges identified by our method is smaller than 16, we randomly select the desired number of edges between ego nodes. Otherwise, we connect all ego nodes and



select the remaining edges in the controller graph randomly. We then use polishing to find the optimal edge weights. The performance of resulting random controller graphs are averaged over 10 trials and the performance loss relative to the optimal centralized controller is displayed in Fig. 2.2b. We see that our algorithm always performs better than the heuristic strategy. On the other hand, the heuristic strategy outperforms the strategy that adds edges randomly (without paying attention to ego nodes). Unlike our method, the heuristic strategy does not necessarily improve the performance by increasing the number of added edges. In fact, the performance deteriorates as the number of edges in the controller graph increases from 4 to 27; see Fig. 2.2b.

**2.6.3 Random disconnected network** The plant graph (blue lines) in Fig. 2.3 contains 50 randomly distributed nodes in a region of  $10 \times 10$  units. Two nodes are neighbors if their Euclidean distance is not greater than 2 units. We examine the problem of adding edges to a plant graph which is not connected and solve the sparsity-promoting optimal control problem (SP) for controller graph with  $m = 1094$  potential edges. This is done for 200 logarithmically-spaced values of  $\gamma \in [10^{-3}, 2.5]$  using the path-following iterative reweighted algorithm as a proxy for inducing sparsity [125]. As indicated by (2.9), we set the weights to be inversely proportional to the magnitude of the solution  $x$  to (SP) at the previous value of  $\gamma$ . We choose  $\varepsilon = 10^{-3}$  in (2.9) and initialize weights for  $\gamma = 10^{-3}$  using the solution to (SP) with  $\gamma = 0$  (i.e., the optimal centralized vector of the edge weights). Topology design is followed by the polishing step that computes the optimal edge weights; see Section 2.2.2.

As illustrated in Fig. 2.3, larger values of  $\gamma$  yield sparser controller graphs (red lines). In contrast to all other examples, the plant graph is not connected and the optimal solution is obtained using the algorithms of Section 2.4. Note that greedy method [37] cannot be used here. Since the plant graph has three disconnected subgraphs, at least two edges in the controller are needed to make the closed-loop network connected.

Figure 2.4 shows that the number of nonzero elements in the vector of the edge weights  $x$  decreases and that the closed-loop performance deteriorates as  $\gamma$  increases. In particular, Fig. 2.4c illustrates the optimal tradeoff curve between the  $\mathcal{H}_2$  performance loss (relative to the optimal centralized controller) and the sparsity of the vector  $x$ . For  $\gamma = 2.5$ , only four edges are added. Relative to the optimal centralized vector of the controller edge weights  $x_c$ , the identified sparse controller in this case uses only 0.37% of

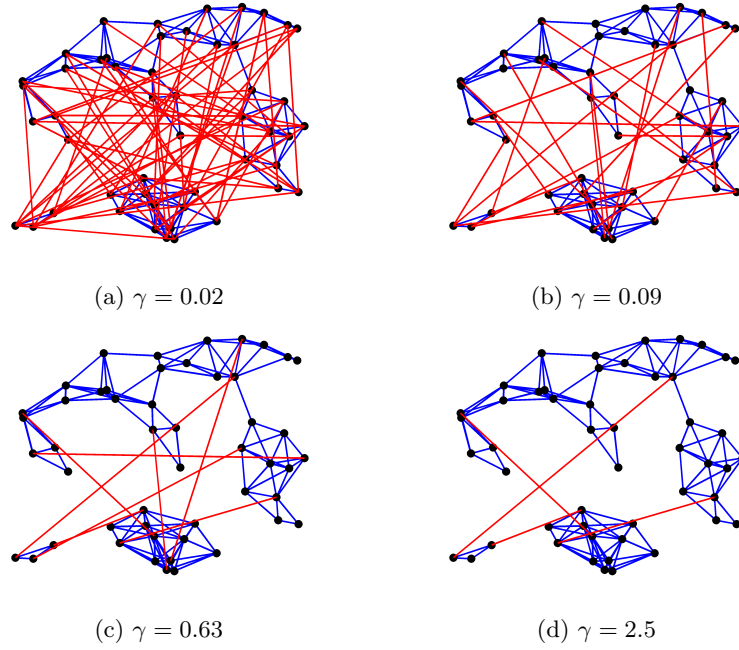


Figure 2.3: Topologies of the plant (blue lines) and controller graphs (red lines) for an unweighted random network with three disconnected subgraphs.

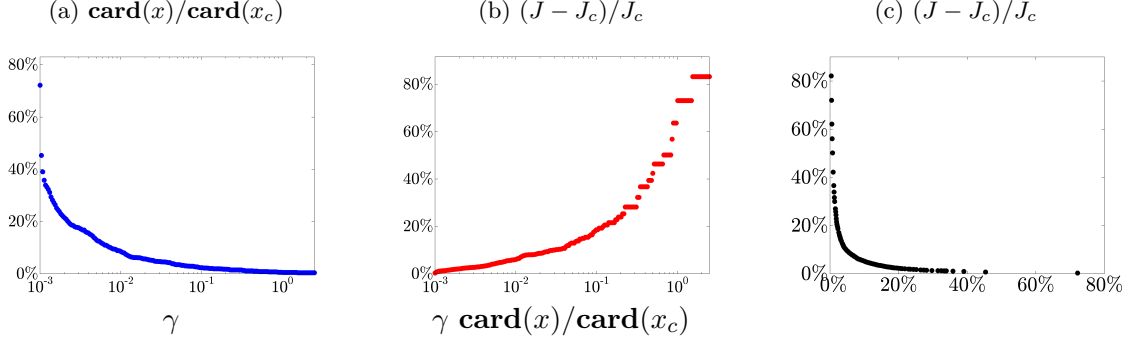


Figure 2.4: (a) Sparsity level; (b) performance degradation; and (c) the optimal tradeoff curve between the performance degradation and the sparsity level of optimal sparse  $x$  compared to the optimal centralized vector of the edge weights  $x_c$ . The results are obtained for unweighted random disconnected plant network with topology shown in Fig. 2.3.

the edges, and achieves a performance loss of 82.13%, i.e.,  $\text{card}(x)/\text{card}(x_c) = 0.37\%$  and  $(J - J_c)/J_c = 82.13\%$ . Here,  $x_c$  is the solution to (SP) with  $\gamma = 0$  and the pattern of non-zero elements of  $x$  is obtained by solving (SP) with  $\gamma = 2.5$  via the path-following iterative reweighted algorithm.

**2.6.4 Path and ring networks** For path networks, our computational experiments show that for a large enough value of the sparsity-promoting parameter  $\gamma$  a single edge, which generates the longest cycle, is added; see Fig. 2.5, top row. This is in agreement with [9] where it was proved that the longest cycle is most beneficial for improving the  $\mathcal{H}_2$  performance of tree networks. Similar observations are made for the spatially-invariant ring network with nearest neighbor interactions. For large values of  $\gamma$ , each node establishes a link to the node that is farthest away in the network; see Fig. 2.5, bottom row. This is in agreement with recent theoretical developments [132] where perturbation analysis was used to identify optimal weak links in edge-transitive consensus networks. Thus, for these regular networks and large enough values of the regularization parameter, our approach indeed provides the globally optimal solution to the original non-convex cardinality minimization problem.

**2.7 Concluding remarks** We have examined the problem of optimal topology design of the corresponding edge weights for undirected consensus networks. Our

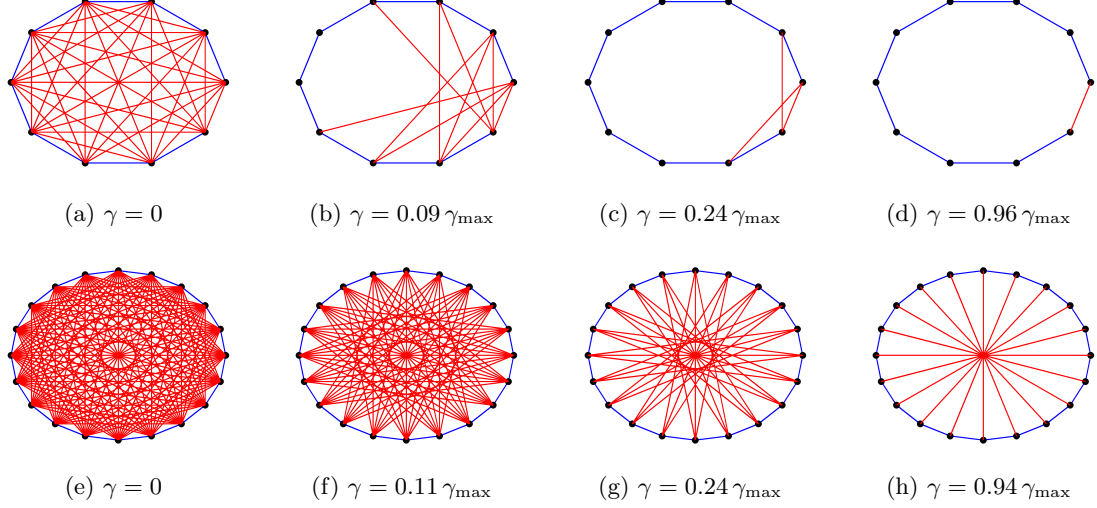


Figure 2.5: The problems of growing unweighted path (top row) and ring (bottom row) networks. Blue lines identify edges in the plant graph, and red lines identify edges in the controller graph.

approach uses convex optimization to balance performance of stochastically-forced networks with the number of edges in the distributed controller. For  $\ell_1$ -regularized minimum variance optimal control problem, we have derived a Lagrange dual and exploited structure of the optimality conditions for undirected networks to develop customized algorithms that are well-suited for large problems. These are based on the proximal gradient and the proximal Newton methods. The proximal gradient algorithm is a first-order method that updates the controller graph Laplacian via the use of the soft-thresholding operator. In the proximal Newton method, sequential quadratic approximation of the smooth part of the objective function is employed and the Newton direction is computed using cyclic coordinate descent over the set of active variables. Examples are provided to demonstrate utility of our algorithms. We have shown that proximal algorithms can solve the problems with millions of edges in the controller graph in several minutes, on a PC. Furthermore, we have specialized our algorithm to the problem of growing connected resistive networks. In this, the plant graph is connected and there are no joint edges between the plant and the controller graphs. We have exploited structure of such networks and demonstrated how additional edges can be systematically added in

a computationally efficient manner.

## Chapter 3

# Edge addition in directed consensus networks

We study the problem of performance enhancement in stochastically-forced directed consensus networks by adding edges to an existing topology. We formulate the problem as a feedback control design, and represent the links as the elements of the controller graph Laplacian matrix. The topology design of the controller network can be cast as an  $\ell_1$  regularized version of the  $\mathcal{H}_2$  optimal control problem. The goal is to optimize the performance of the network by selecting a controller graph with low communication requirements. To deal with the structural constraints that arise from the absence of absolute measurements, we introduce a coordinate transformation to eliminate the average mode and assure convergence of all states to the average of the initial node values. By exploiting structure of the optimization problem, we develop a customized algorithm based on the alternating direction method of multipliers to design a sparse controller network that improves the performance of the closed-loop system.

**3.1 Introduction** In this chapter, we consider the problem of adding edges to a weakly connected directed consensus network in order to improve performance. In particular, we are interested in designing sparse communication graphs that strike a balance between the variance amplification of the closed-loop system and the number of communication links. In general, this is a combinatorial search problem and is non-convex. In undirected networks, convex relaxations or greedy algorithms have been

introduced in order to optimize algebraic connectivity of the network [34, 65] or network coherence [35, 133, 134] by adding edges from a given set of edges.

In the absence of disturbances, a strongly connected balanced network converges to the average of the initial node values [2]. However, in the presence of additive stochastic disturbances, the network average experiences a random walk. Thus, the control objective is to minimize mean square deviation from average. To cope with structural constraints that arise from the absence of absolute information exchange, we introduce a coordinate transformation to eliminate the average mode. While it is desired to promote sparsity of controller graph in the physical domain, the  $\mathcal{H}_2$  optimal control problem is solved in the transformed set of coordinates where the average mode is eliminated.

We formulate the edge addition problem for directed networks with an objective of optimizing the closed-loop coherence. We consider the scenario in which the plant graph is unbalanced. Structural requirements that the closed-loop graph Laplacian is weakly connected and balanced make the optimal control problem challenging. We approach this problem using sparsity-promoting optimal control framework [41, 42, 71]. In our formulation, performance is captured by the  $\mathcal{H}_2$  norm of the closed-loop network and  $\ell_1$  regularization is introduced as a proxy for inducing sparsity in the controller graph [35, 133]. By exploiting the structure of the problem, we develop a customized algorithm based on alternating direction method of multipliers (ADMM) [113].

The rest of the chapter is structured as follows. In Section 3.2, we provide necessary background on graph theory for directed networks, define consensus problems on graphs, and introduce an appropriate change of coordinates. In Section 3.3, we formulate the optimal control problem. In Section 3.4, we develop a customized algorithm based on ADMM. In Section 3.5, we use our algorithm for sparse edge addition to directed consensus networks. Finally, in Section 3.6, we provide concluding remarks and highlight future directions.

**3.2 Motivation and background** In this section, after providing the necessary background, we describe the dynamics of consensus networks, inherent structural constraints, and the challenges in the related design problems. We then introduce a change of coordinates that enables us to overcome these restrictions.

**3.2.1 Directed graphs** Herein, we provide a brief overview of the background material; for additional information, see [2].

*Weighted digraph:* a weighted directed graph is denoted by  $\mathcal{D} = (V, E, w)$  where  $V$  and  $E$  are the sets of nodes and edges and  $w$  is the vector of edge weights. The edges are directed and a value  $w_{ij}$  is the edge weight between nodes  $v_i$  and  $v_j$ . If the ordered pair  $(v_i, v_j) \in E$  then  $v_i$  is the tail of the edge and  $v_j$  is its head.

*Strongly connected digraph:* a digraph is strongly connected if there is a directed path between every pair of nodes.

*Weakly connected digraph:* a digraph is weakly connected if it is a connected graph when the directions are omitted.

*Adjacency matrix:* the  $ij$ th element is given by

$$A_{ij} = \begin{cases} w_{ij}, & (v_j, v_i) \in E \\ 0, & \text{otherwise.} \end{cases}$$

*Degree matrix* is a diagonal matrix with  $D_{ii} = d_{\text{in}}(v_i)$  where  $d_{\text{in}}(v_i)$  is the weighted in-degree of node  $v_i$ ,

$$d_{\text{in}}(v_i) = \sum_j w_{ij}.$$

We sum over  $j$ 's for  $(v_j, v_i) \in E$ .

*Laplacian matrix* is a matrix  $L = D - A$ , where  $L \mathbf{1} = 0$  by definition.

*Balanced digraph:* a digraph is balanced if the weighted in-degree and the weighted out-degree for each node are equal.

**3.2.2 Feedback design in consensus networks** We consider a consensus dynamics

$$\dot{x} = -L_p x + u + d$$

where  $d$  and  $u$  are the disturbance and control inputs,  $x$  is the vector of the states, and  $L_p \in \mathbb{R}^{n \times n}$  is the graph Laplacian of the plant network. We assume that the plant network is weakly connected and formulate the edge addition problem as a feedback design problem with

$$u = -L_x x$$



where  $L_x \in \mathbb{R}^{n \times n}$  is the weighted directed Laplacian matrix of the controller. This matrix represents the locations, directions, and edge weights. Note that each nonzero element in  $L_x$  can either indicate addition of an edge or re-tuning of an existing edge gain.

The closed-loop system is given by

$$\dot{x} = -(L_p + L_x)x + d. \quad (3.1a)$$

Our goal is to optimally design the feedback gain matrix  $L_x$  in order to achieve the desired tradeoff between the controller sparsity and network performance. The performance is quantified by the steady-state variance amplification of the stochastically-forced network, from the white-in-time input  $d$  to the performance output  $z$  that quantifies deviation from consensus and control effort,

$$z = \begin{bmatrix} Q^{1/2} \\ -R^{1/2}L_x \end{bmatrix} x. \quad (3.1b)$$

Here, the matrices  $Q = Q^T \succeq 0$  and  $R \succ 0$  are the state and control weights, respectively, and  $\succ$  ( $\succeq$ ) signifies positive definiteness (semi-definiteness) of a matrix.

In consensus networks, each node updates its state using the relative information exchange with its neighbors. In the presence of white noise, the average mode  $\bar{x} = (1/n)x^T \mathbf{1}$  experiences a random walk and variance increased linearly with time. A key property of the Laplacian matrix of the controller is  $L_x \mathbf{1} = 0$ . Since our primary control objective is to achieve consensus, only differences between node values are penalized in the performance output. Therefore, the state weight matrix  $Q$  has a zero eigenvalue with the corresponding eigenvector of all ones, i.e.,  $Q \mathbf{1} = 0$ . Furthermore, we choose  $Q$  to be positive definite on the orthogonal complement of the subspace  $\text{span}(\mathbf{1})$ ,

$$Q + (1/n) \mathbf{1} \mathbf{1}^T \succ 0.$$

The following lemma summarizes the well-known conditions for achieving consensus in the absence of disturbances in directed networks [2].

**Lemma 4.** *The agreement protocol over a digraph reaches the average consensus,*

$(1/n)\mathbf{1}\mathbf{1}^T x(0)$ , for every initial condition if and only if it is weakly connected and balanced.

Thus, for a weakly connected  $L_p$  which is not necessarily balanced, it is required that the closed-loop graph Laplacian,  $L_p + L_x$  be balanced, which amounts to

$$\mathbf{1}^T (L_p + L_x) = 0. \quad (3.2)$$

The problem of designing a controller graph that provides a desired tradeoff between performance index  $J$  of the network and the sparsity of the controller  $L_x$  can be formulated as

$$\begin{aligned} & \underset{L_x}{\text{minimize}} \quad J(L_x) + \gamma \|W \circ L_x\|_1 \\ & \text{subject to} \quad \mathbf{1}^T (L_x + L_p) = 0 \\ & \quad \quad \quad L_x \mathbf{1} = 0. \end{aligned} \quad (3.3)$$

The positive scalar  $\gamma$  is the sparsity-promoting parameter that characterizes a trade-off between network performance and sparsity of the controller and  $\circ$  denotes elementwise multiplication. The first condition guarantees asymptotic consensus to the initial network average value in the absence of disturbances and the second condition secures the row stochastic property of the controller graph Laplacian. We guarantee that the closed-loop graph Laplacian remains weakly connected via proper step-size selection in the algorithm.

In what follows, we quantify the network performance using the  $\mathcal{H}_2$  norm. The weighted  $\ell_1$  norm [125] of  $L_x$  is a convex approximation of the cardinality function. In each iteration, each element of the weight matrix  $W(i, j)$  is chosen to be inversely proportional to the magnitude of  $L_x(i, j)$  in the previous iteration and can be determined as

$$W(i, j) = 1/(|L_x(i, j)| + \varepsilon). \quad (3.4)$$

This puts larger emphasis on smaller optimization variables, where a small positive parameter  $\varepsilon$  ensures that  $W(i, j)$  is well-defined.

**3.2.3 Elimination of the average mode** Since both  $Q$  and  $L_x$  have a zero eigenvalue associated with the vector of all ones, the average mode  $\bar{x}$  is not observable

from the output  $z$ . In order to eliminate the average mode, we introduce the following change of coordinates

$$\begin{bmatrix} \psi \\ \bar{x} \end{bmatrix} = \begin{bmatrix} U^T \\ \mathbf{1}^T/n \end{bmatrix} x$$

where  $U \in \mathbb{R}^{n \times (n-1)}$  is a full-rank matrix and its columns span the orthogonal complement of  $\mathbf{1}$ . Using the properties of the matrix  $U$ ,

$$U^T U = I, \quad U U^T = I - (1/n) \mathbf{1} \mathbf{1}^T, \quad U^T \mathbf{1} = 0,$$

we have

$$\begin{bmatrix} \dot{\psi} \\ \dot{\bar{x}} \end{bmatrix} = \begin{bmatrix} -U^T (L_p + L_x) U & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \bar{x} \end{bmatrix} + \begin{bmatrix} U^T \\ (1/n) \mathbf{1}^T \end{bmatrix} d.$$

In the new coordinates, the performance output is given by

$$z = \begin{bmatrix} Q^{1/2} U & 0 \\ -R^{1/2} L_x U & 0 \end{bmatrix} \begin{bmatrix} \psi \\ \bar{x} \end{bmatrix}.$$

The network average  $\bar{x} = (1/n) \mathbf{1}^T d$  experiences a random walk and the vector  $\psi$  quantifies the deviation from average of the states. Since our control objective is to minimize the deviation from average in the nodes, the average mode is not of interest. Therefore, the minimal representation of the system containing only the state  $\psi$  is given by

$$\begin{aligned} \dot{\psi} &= -U^T (L_p + L_x) U \psi + U^T d \\ z &= \begin{bmatrix} Q^{1/2} U \\ -R^{1/2} L_x U \end{bmatrix} \psi. \end{aligned} \tag{3.5}$$

In order to guarantee that the closed-loop graph Laplacian  $(L_p + L_x)$  is balanced, we introduce the following change of variables

$$L_x = U F U^T - (1/n) \mathbf{1} \mathbf{1}^T L_p \Leftrightarrow F = U^T L_x U. \tag{3.6}$$

Note that the main contribution that differentiates this work from previous results [133] is that we are considering a weakly-connected directed unbalanced network, and we require the closed-loop network to be balanced. The problem of adding edges to a directed network has not been addressed in [133]. Moreover, the presented framework cannot be used to solve the current constrained problem due to an additional bias term. In our framework, a new change of variables is introduced to accommodate structural constraints.

Using the properties of the matrix  $U$ , the constraints (3.2) and  $L_x \mathbf{1} = 0$  are automatically satisfied. The equation (3.6) demonstrates how we can move back and forth between two variables  $F$  and  $L_x$ .

Substituting  $L_x$  given by (3.6) into (3.5), we can write the state-space representation of the closed-loop system as

$$\begin{aligned} \dot{\psi} &= -(U^T L_p U + F) \psi + U^T d \\ z &= \begin{bmatrix} Q^{1/2} U \\ -R^{1/2}(U F - (1/n) \mathbf{1} \mathbf{1}^T L_p U) \end{bmatrix} \psi. \end{aligned} \quad (3.7)$$

Next, we formulate the optimal control problem and propose a framework to design sparse controller  $L_x$ .

**3.3 Topology design for directed networks** In this section, we approach the problem of topology design as a regularized optimal control problem.

**3.3.1 Sparsity-promoting optimal control problem** The  $\mathcal{H}_2$  norm of the transfer function from  $d$  to  $z$ ,

$$\|H\|_2^2 = J(F) = \text{trace}(P(F))$$

quantifies the variance amplification of the closed-loop network (3.7). Here,  $P(F)$  is the closed-loop observability Gramian which is the solution of the Lyapunov equation,

$$(U^T L_p U + F)^T P + P (U^T L_p U + F) = C^T C \quad (3.8)$$

where

$$C = \begin{bmatrix} Q^{1/2} U \\ -R^{1/2}(U F - (1/n) \mathbf{1}\mathbf{1}^T L_p U) \end{bmatrix}.$$

The control design problem (3.3) takes the following form

$$\begin{aligned} & \underset{F, L_x}{\text{minimize}} && J(F) + \gamma \|W \circ L_x\|_1 \\ & \text{subject to} && U F U^T - L_x - (1/n) \mathbf{1}\mathbf{1}^T L_p = 0. \end{aligned} \tag{3.9}$$

In (3.9), the Laplacian matrix  $L_x$  and the matrix  $F$  are the optimization variables; the problem data is given by plant graph Laplacian  $L_p$ , state and control weights  $Q$  and  $R$ , and positive regularization parameter  $\gamma$ . The matrix  $W$  is the weight matrix that imposes a penalty on the magnitude of the elements in  $L_x$ . With the problem formulation (3.9), we are able to minimize the  $\mathcal{H}_2$  norm of the network  $J(F)$  in the transformed coordinates where the average mode is eliminated, while promoting sparsity of  $L_x$  in the physical domain. For  $\gamma = 0$ , the solution to (3.9) is typically given by a matrix  $L_x$  with all non-zero components. As the regularization parameter increases, the number of non-zero elements in the controller graph decreases.

**3.3.2 Structured optimal control problem** After the structure of the controller graph Laplacian  $L_x$  has been designed, we fix the sparsity pattern  $\mathcal{S}$  and then solve the following problem

$$\begin{aligned} & \underset{F}{\text{minimize}} && J(F) \\ & \text{subject to} && U F U^T - (1/n) \mathbf{1}\mathbf{1}^T L_p \in \mathcal{S} \end{aligned} \tag{3.10}$$

whose solution provides the optimal controller graph Laplacian with the desired structure. This optimization problem is obtained by setting  $\gamma = 0$  in (3.9) and adding the sparsity pattern  $\mathcal{S}$ . This “polishing” or “debiasing” step is used to improve the performance relative to the solution of the sparsity-promoting optimal control problem (3.9).

**3.4 An ADMM-based algorithm** We next exploit the structure of the constrained optimization problem (3.9) and develop a customized algorithm based on

ADMM. The augmented Lagrangian associated with (3.9) is given by

$$\begin{aligned} \mathcal{L}_\rho(F, L_x; \Lambda) = & J(F) + \gamma \|W \circ L_x\|_1 + \langle \Lambda, U F U^T - L_x - (1/n) \mathbf{1}\mathbf{1}^T L_p \rangle + \\ & \frac{\rho}{2} \|U F U^T - L_x - (1/n) \mathbf{1}\mathbf{1}^T L_p\|_F^2 \end{aligned} \quad (3.11)$$

where the matrix  $\Lambda$  is the Lagrange multiplier,  $\rho$  is a positive scalar, and  $\|\cdot\|_F$  is the Frobenius norm. The ADMM algorithm consists of the following steps at each iteration

$$\begin{aligned} F^{k+1} &= \underset{F}{\operatorname{argmin}} \mathcal{L}_\rho(F, L_x^k; \Lambda^k) \\ L_x^{k+1} &= \underset{L_x}{\operatorname{argmin}} \mathcal{L}_\rho(F^{k+1}, L_x; \Lambda^k) \\ \Lambda^{k+1} &= \Lambda^k + \rho (U F^{k+1} U^T - L_x^{k+1} - (1/n) \mathbf{1}\mathbf{1}^T L_p). \end{aligned}$$

The algorithm terminates when  $\|L_x^{k+1} - L_x^k\|_F \leq \epsilon_1$  and  $\|U F^{k+1} U^T - L_x^{k+1} - (1/n) \mathbf{1}\mathbf{1}^T L_p\| \leq \epsilon_2$ , where  $\epsilon_1$  and  $\epsilon_2$  are desired tolerances.

Note that, the smooth part  $J(F)$  and the non-smooth part  $\|W \circ L_x\|_1$  are now operating in different coordinates; therefore, descent algorithms can be utilized in the  $F$ -minimization step. Moreover, the  $\ell_1$  norm is a separable function with respect to each element of  $L_x$ . Thus, we can determine the solution to the  $L_x$ -minimization step analytically. In the Lagrange multiplier update step, we use the step-size equal to  $\rho$  in order to guarantee the dual feasibility [113].

**3.4.1  $F$ -minimization step** We bring the  $F$ -minimization step to the following form by using the properties of the matrices  $L_p$  and  $U$ ,

$$F^{k+1} = \underset{F}{\operatorname{argmin}} J(F) + \frac{\rho}{2} \|F - S^k\|_F^2$$

where

$$S^k = U^T (L_x^k + (1/n) \mathbf{1}\mathbf{1}^T L_p - (1/\rho) \Lambda^k) U = U^T (L_x^k - (1/\rho) \Lambda^k) U.$$

We employ the Anderson-Moore method to solve this problem [135]. This algorithm converges faster compared to the gradient method and its implementation is simpler compared to Newton method [42].

We next summarize the first- and second-order derivatives of the objective function  $J$ . The second order approximation of the smooth part of objective function  $J$  around  $\bar{F}$  is given by

$$J(\bar{F} + \tilde{F}) \approx J(\bar{F}) + \left\langle \nabla_F J(\bar{F}), \tilde{F} \right\rangle + \frac{1}{2} \left\langle \nabla_F^2 J(\bar{F}, \tilde{F}), \tilde{F} \right\rangle.$$

For related developments we refer the reader to [135].

**Proposition 5.** *The gradient and the Hessian of  $J$  at  $\bar{F}$  are determined by*

$$\begin{aligned} \nabla J(\bar{F}) &= 2(\bar{R}F - (1/n)U^T R \mathbf{1}\mathbf{1}^T L_p U - P)L \\ \nabla^2 J(\bar{F}, \tilde{F}) &= 2\left((\bar{R}\tilde{F} - \tilde{P})L + (\bar{R}F - (1/n)U^T R \mathbf{1}\mathbf{1}^T L_p U - P)\tilde{L}\right) \end{aligned} \quad (3.12)$$

where  $\bar{R} = U^T R U$  and the matrix  $P$  is given by (3.8) and is the observability Gramian. The matrix  $L$  is the controllability Gramian and is determined by

$$(U^T L_p U + F)L + L(U^T L_p U + F)^T = U^T U \quad (3.13)$$

where  $U^T U = I$  is identity. The matrices  $\tilde{P}$  and  $\tilde{L}$  are the solutions to the following Lyapunov equations

$$\begin{aligned} (U^T L_p U + F)\tilde{L} + \tilde{L}(U^T L_p U + F) &= -\tilde{F}L - L\tilde{F} \\ (U^T L_p U + F)^T \tilde{P} + \tilde{P}(U^T L_p U + F) &= \\ &\tilde{F}^T(\bar{R}F - P - (1/n)U^T R \mathbf{1}\mathbf{1}^T L_p U) + \\ &(F^T \bar{R} - P - (1/n)U^T L_p^T \mathbf{1}\mathbf{1}^T R U)\tilde{F}. \end{aligned}$$

By setting  $\nabla_F \mathcal{L}_\rho = \nabla_F J + \rho(F - S^k) = 0$ , we obtain

$$2(\bar{R}F - (1/n)U^T R \mathbf{1}\mathbf{1}^T L_p U - P)L + \rho(F - S^k) = 0. \quad (3.14)$$

The necessary conditions for the optimality of  $\mathcal{L}_\rho(F, L_x; \Lambda)$  are given by (3.8), (3.13) and (3.14).

The Anderson-Moore method solves the  $F$ -minimization step iteratively. In each

iteration, the algorithm starts with a stabilizing  $F$  and solves two Lyapunov equations and one Sylvester equation. It first solves the Lyapunov equations (3.8), (3.13) with a fixed  $F$  to obtain controllability and observability Gramians  $L$  and  $P$ , respectively. Then it solves the Sylvester equation (3.14) for  $F$  with fixed  $L$  and  $P$ . Then we use Newton's method to find the descent direction between two consecutive steps by utilizing (3.12). We next employ a line search strategy to determine an appropriate step-size in order to guarantee convergence to a stationary point and the closed-loop stability.

**3.4.2  $L_x$ -minimization step** Using the expression for the augmented Lagrangian (3.11), we can write the  $L_x$ -minimization step as

$$L_x^{k+1} = \underset{L_x}{\operatorname{argmin}} \quad \gamma \|L_x\|_1 + \frac{\rho}{2} \|L_x - T^k\|_F^2$$

where

$$T^k = U F^{k+1} U^T - (1/n) \mathbf{1}\mathbf{1}^T L_p + (1/\rho) \Lambda^k.$$

The solution is given by soft-thresholding

$$L_k(i, j)^{k+1} = \begin{cases} (1 - \frac{v}{|T^k(i, j)|}) T^k(i, j), & |T^k(i, j)| > v \\ 0, & \text{otherwise} \end{cases}$$

where  $v = (\gamma/\rho) W(i, j)$ .

Convergence analysis of ADMM for convex problems can be found in [113]. For non-convex problems, the quadratic term in the augmented Lagrangian locally convexify the problem for a large  $\rho$ . A recent result on convergence analysis of ADMM for a family of non-convex problems can be found in [136]. Computational results also show that ADMM works well when  $\rho$  is sufficiently large [137, 138].

## 3.5 Computational experiments

**3.5.1 Synthetic example** In this section, we employ our customized algorithm based on ADMM to add certain number of edges to a given directed network. The plant network is a randomly generated graph with  $n$  nodes and with edge weight  $l_i \in (0, 1)$



for the  $i$ th edge which is obtained randomly.

We set  $R = I$  and choose the state weight that penalizes the mean-square deviation from the network average,

$$Q = I - (1/n) \mathbf{1}\mathbf{1}^T.$$

We solve the sparsity-promoting optimal control problem (3.9) for controller graph for 100 logarithmically-spaced values of  $\gamma \in [0.001, 3]$  using the path-following iterative reweighted algorithm as a proxy for inducing sparsity [125]. We set the weights to be inversely proportional to the magnitude of the solution  $L_x$  to (3.9) at the previous value of  $\gamma$ . We choose  $\varepsilon = 10^{-3}$  in (3.4) and initialize weights for  $\gamma = 0.01$  using the solution to (3.9) with LQR state-feedback matrix. Topology design is followed by a polishing step that computes the optimal weights of identified edges; see Section I-A.

**3.5.2 Performance improvement** The randomly generated plant graph in Fig. 3.1a is a directed graph with  $n = 20$  nodes. The plant graph is weakly connected but unbalanced. Thus, it will not converge to the initial nodes average value. In order to reach consensus and improve the performance, adding edges to the plant network is required. Figure 3.1 illustrates that by increasing  $\gamma$ , the controller graph becomes sparser. The number of added edges to the network is equal to the number of nonzero off-diagonal elements in the controller. Specifically, for  $\gamma = 0.001$ , the number of nonzero elements in the controller graph is 84, among which 74 edges are added to the original network, and the other 10 nonzero elements represent the diagonal elements. By increasing  $\gamma$  to 0.0788, there are 33 nonzero elements in the controller graph, and only 25 edges are added to the plant network. It is noteworthy that the Laplacian matrix of the controller graph can not be a zero matrix, because the plant network is an unbalance graph and a nonzero Laplacian matrix of the controller is needed to make the closed-loop graph Laplacian balanced.

Figure 3.2 shows that the closed-loop performance deteriorates and the number of nonzero elements in the controller graph Laplacian  $L_x$  decreases as  $\gamma$  increases. As shown in Fig. 3.2b, relative to the optimal LQR controller,  $L_c$ , the  $\mathcal{H}_2$  performance loss decreases as the sparsity of the controller graph Laplacian matrix  $L_x$  increases. In particular, for  $\gamma = 3$ , there are only 24 nonzero elements in the controller graph. This is equivalent to have only 16 added edges. The identified sparse controller in

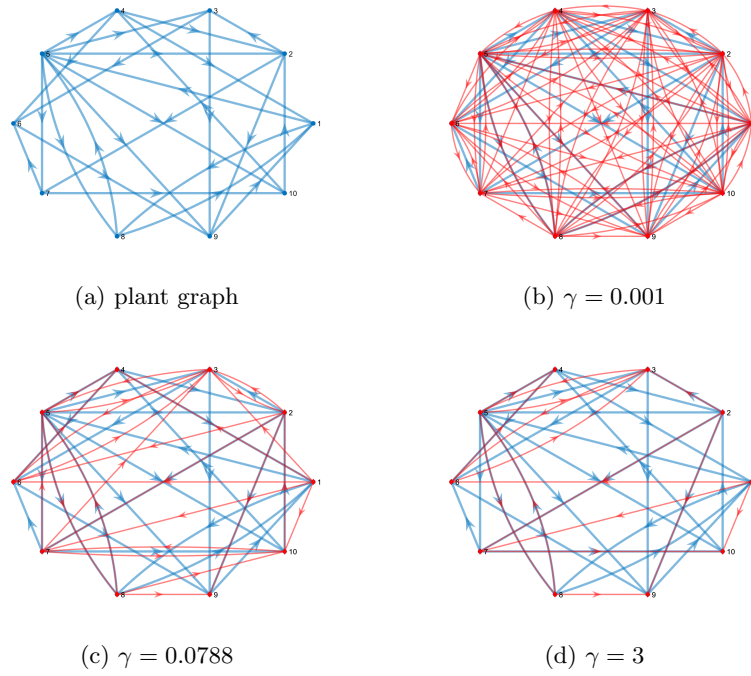


Figure 3.1: Topologies of the plant (blue lines) and controller graphs (red lines) for a randomly generated weighted network.

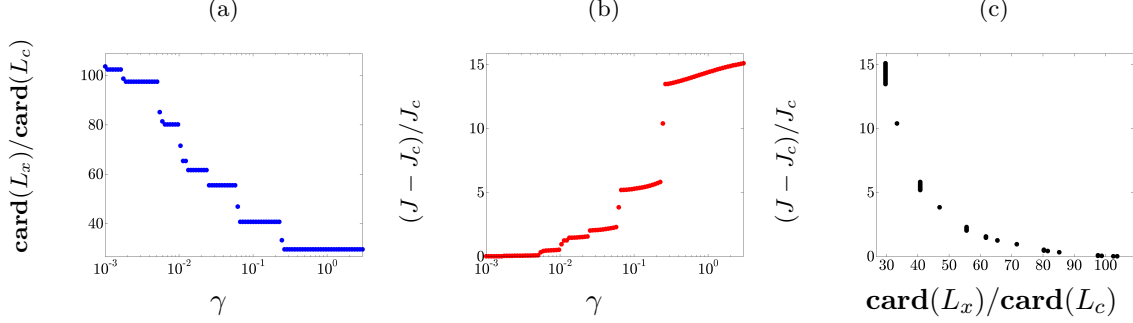


Figure 3.2: (a) Sparsity level; (b) performance degradation; and (c) the optimal tradeoff between the performance degradation and the sparsity level of the optimal sparse  $L_x$  compared to the optimal centralized controller  $L_c$ . The results are obtained for the weighted random plant network with topology shown in Fig. 3.1a.

this case uses only 0.057% of the edges, relative to the optimal LQR controller, i.e.,  $\text{card}(L_x)/\text{card}(L_c) = 29.629\%$  and achieves a performance loss of 15.118%, i.e.,  $(J - J_c)/J_c = 15.118\%$ .

**3.6 Concluding remarks** We consider the  $\ell_1$  regularized version of optimal control problem for adding edges to directed consensus networks in order to reach consensus and optimally enhance performance. Although the given plant network is not necessarily balanced, in order to reach agreement, we restrict the closed-loop graph Laplacian to be balanced. The performance is measured by the  $\mathcal{H}_2$  norm from the disturbance to the output of the closed-loop network. In general, this problem is a combinatorial search problem. We use sparsity promoting optimal control framework and introduce weighted  $\ell_1$  regularization as a proxy for promoting sparsity of the controller. By exploiting structure of the problem, we develop an algorithm based on ADMM. An example is provided to demonstrate the utility of the developed algorithm.

## Part II

# Network estimation and inference

## Chapter 4

# Topology identification of undirected consensus networks via sparse inverse covariance estimation

We study the problem of identifying sparse interaction topology using sample covariance matrix of the state of the network. Specifically, we assume that the statistics are generated by a stochastically-forced undirected first-order consensus network with unknown topology. We propose a method for identifying the topology using a regularized Gaussian maximum likelihood framework where the  $\ell_1$  regularizer is introduced as a means for inducing sparse network topology. The proposed algorithm employs a sequential quadratic approximation in which the Newton's direction is obtained using coordinate descent method. We provide several examples to demonstrate good practical performance of the method.

**4.1 Introduction** In this chapter, we develop a convex optimization framework for identifying sparse interaction topology using sample covariance matrix of the state of the network. Our framework utilizes an  $\ell_1$ -regularized Gaussian maximum likelihood estimator. Because of strong theoretical guarantees, this approach has been

commonly used for recovering sparse inverse covariance matrices [139–143]. We utilize the structure of undirected networks to develop an efficient second-order method based on a sequential quadratic approximation. As in [124, 144], we compute the Newton’s direction using coordinate descent method [145–147] that employs active set strategy. The main point of departure is the formulation of a convex optimization problem that respects particular structure of undirected consensus networks. We also use a reweighted  $\ell_1$ -heuristics as an effective means for approximating non-convex cardinality function [125], thereby improving performance relative to standard  $\ell_1$  regularization.

Our presentation is organized as follows. In Section 4.2, we formulate the problem of topology identification using sparse inverse covariance matrix estimation. In Section 4.3, we develop a customized second-order algorithm to solve the  $\ell_1$ -regularized Gaussian maximum likelihood estimation problem. In Section 4.4, we use computational experiments to illustrate features of the method. In Section 4.5, we conclude with a brief summary.

**4.2 Problem formulation** In this section, we provide background material on stochastically forced undirected first-order consensus networks and formulate the problem of topology identification using a sample covariance matrix. The inverse of a given sample covariance matrix can be estimated using Gaussian maximum likelihood estimator. For undirected consensus networks, we show that the estimated matrix is related to the graph Laplacian of the underlying network.

Our objective is to identify the underlying graph structure of a stochastically forced undirected consensus network with a known number of nodes by sampling its second-order statistics. In what follows, we relate the problem of topology identification for consensus networks to the inverse covariance matrix estimation problem.

**4.2.1 Undirected consensus networks** We consider an undirected consensus network

$$\dot{\psi} = -L_x \psi + d, \quad (4.1)$$

where  $\psi \in \mathbb{R}^n$  represents the state of  $n$  nodes,  $d$  is the disturbance input, and the symmetric  $n \times n$  matrix  $L_x$  represents the graph Laplacian. The matrix  $L_x$  is restricted to have an eigenvalue at zero corresponding to an eigenvector of all ones,  $L_x \mathbf{1} = 0$ . This

requirement is implicitly satisfied by expressing  $L_x$  in terms of the incidence matrix  $E$

$$L_x := \sum_{l=1}^m x_l \xi_l \xi_l^T = E \operatorname{diag}(x) E^T,$$

where  $\operatorname{diag}(x)$  is a diagonal matrix containing the vector of edge weights  $x \in \mathbb{R}^m$ . Each column  $\xi_l = \mathbf{e}_i - \mathbf{e}_j$ , where  $\mathbf{e}_i \in \mathbb{R}^n$  is the  $i$ th basis vector, represents an edge connecting nodes  $i$  and  $j$ . The  $m$  columns of  $E$  specify the edges that may be used to construct the consensus network. For a complete graph, there are  $m = n(n-1)/2$  potential edges.

In order to achieve consensus in the absence of disturbances, it is required that the closed-loop graph Laplacian,  $L_x$ , be positive definite on the orthogonal complement of the vector of all ones,  $\mathbf{1}^\perp$  [2]. This amounts to positive definiteness of the “strengthened” graph Laplacian of the closed-loop network

$$X := L_x + (1/n) \mathbf{1} \mathbf{1}^T \succ 0. \quad (4.2)$$

Clearly, since  $L_x \mathbf{1} = 0$ ,  $X \mathbf{1} = \mathbf{1}$ .

Consensus networks attempt to compute the network average; thus, it is of interest to study the deviations from average

$$\tilde{\psi}(t) := \psi(t) - \mathbf{1} \bar{\psi}(t) = (I - (1/n) \mathbf{1} \mathbf{1}^T) \psi(t),$$

where  $\bar{\psi}(t) := (1/n) \mathbf{1}^T \psi(t)$  is the network average and corresponds to the zero eigenvalue of  $L_x$ . From (4.1), it follows that the dynamics of the deviation from average are,

$$\dot{\tilde{\psi}} = -L_x \tilde{\psi} + (I - (1/n) \mathbf{1} \mathbf{1}^T) d.$$

The steady-state covariance of  $\tilde{\psi}$ ,

$$P := \lim_{t \rightarrow \infty} \mathbf{E} \left( \tilde{\psi}(t) \tilde{\psi}^T(t) \right),$$

is given by the solution to the algebraic Lyapunov equation

$$L_x P + P L_x = I - (1/n) \mathbf{1} \mathbf{1}^T.$$

For connected networks, the unique solution is given by

$$\begin{aligned} P &= \frac{1}{2} L_x^\dagger = \frac{1}{2} \left( (L_x + (1/n) \mathbf{1}\mathbf{1}^T)^{-1} - (1/n) \mathbf{1}\mathbf{1}^T \right) \\ &= \frac{1}{2} (X^{-1} - (1/n) \mathbf{1}\mathbf{1}^T), \end{aligned} \quad (4.3)$$

where  $(\cdot)^\dagger$  is the pseudo-inverse of a matrix. Thus, the inverse of the steady-state covariance matrix of the deviation from network average is determined by the strengthened graph Laplacian of the consensus network  $X$ .

**4.2.2 Topology identification** A sparse precision matrix can be obtained as the solution to the regularized maximum log-likelihood problem [141],

$$\begin{aligned} &\underset{X}{\text{minimize}} \quad -\log \det(X) + \text{trace}(S X) + \gamma \|X\|_1 \\ &\text{subject to} \quad X \succ 0, \end{aligned} \quad (4.4)$$

where  $S$  is the sample covariance matrix,  $\gamma$  is a positive regularization parameter, and  $\|X\|_1 := \sum |X_{ij}|$  is the  $\ell_1$  norm of the matrix  $X$ . The  $\ell_1$  norm is introduced as a means for inducing sparsity in the inverse covariance matrix where a zero element implies conditional independence. This problem has received significant attention in recent years [124, 139, 141, 144, 148–151].

In this work, we establish a relation between inverse covariance matrix estimation and the problem of topology identification of an undirected consensus network. We are interested in identifying a sparse topology that yields close approximation of a given sample covariance matrix. This is achieved by solving the following problem,

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad J(x) + \gamma \sum_{l=1}^m |x_l| \\ &\text{subject to} \quad E \text{diag}(x) E^T + (1/n) \mathbf{1}\mathbf{1}^T \succ 0, \end{aligned} \quad (\text{NI})$$

where

$$J(x) = -\log \det(E \text{diag}(x) E^T + (1/n) \mathbf{1}\mathbf{1}^T) + \text{trace}(S E \text{diag}(x) E^T).$$



Relative to [124, 139, 144, 148, 149], our optimization problem has additional structure induced by the dynamics of undirected consensus networks.

The network identification problem (NI) is a convex but non-smooth problem where the optimization variable is the vector of the edge weights  $x \in \mathbb{R}^m$  and the problem data is the sample covariance matrix  $S$  and the incidence matrix. The incidence matrix is selected to contain all possible edges. The  $\ell_1$  norm of  $x$  is a convex relaxation of the cardinality function and it is introduced to promote sparsity. The positive parameter  $\gamma$  specifies the emphasis on sparsity versus matching the sample covariance matrix  $S$ . For  $\gamma = 0$ , the solution to (NI) is typically given by a vector  $x$  with all non-zero elements. The positive definite constraint comes from (4.2) and guarantees a connected closed-loop network and thus asymptotic consensus in the absence of disturbances.

**Remark 7.** *We also use this framework to address the network sparsification problem where it is of interest to find a sparse network that generates close approximation of the covariance matrix of a given dense network. We choose  $E$  to be equal to the incidence matrix of the primary network.*

### 4.3 Customized algorithm based on sequential quadratic approximation

We next exploit the structure of the optimization problem (NI) to develop an efficient customized algorithm. Our algorithm is based on sequential quadratic approximation of the smooth part  $J$  of the objective function in (NI). This method benefits from exploiting second-order information about  $J$  and from computing the Newton direction using cyclic coordinate descent [145–147] over the set of active variables. We find a step-size that ensures the descent direction via backtracking line search. Furthermore, by restricting our computations to active search directions, computational cost is significantly reduced. A similar approach has been recently utilized in a number of applications, including sparse inverse covariance estimation in graphical models [124, 144, 152]. In this work, we have additional structural constraints and use reweighted heuristics in order to achieve sparsity. We use an alternative proxy for promoting sparsity which is given by the weighted  $\ell_1$  norm [125]. In particular, we solve problem (NI) for different values of  $\gamma$  using a path-following iterative reweighted algorithm; see Section II (A) in [?]. The topology identification then is followed by a

polishing step to debias the identified edge weights.

**4.3.1 Structured problem: debiasing step** In addition to promoting sparsity of the identified edge weights, the  $\ell_1$  norm penalizes the magnitude of the nonzero edge weights. In order to gauge the performance of the estimated network topology, once we identify a set of sparse topology via (NI), we solve the structured “polishing” or “debiasing” problem to optimize  $J$  over the set of identified edges. To do this, we form a new incidence matrix  $\hat{E}$  which contains only those edges identified as nonzero in the solution to (NI) and form the problem,

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & -\log \det \left( \hat{E} \text{diag}(x) \hat{E}^T + (1/n) \mathbf{1}\mathbf{1}^T \right) + \text{trace} \left( S \hat{E} \text{diag}(x) \hat{E}^T \right) \\ \text{subject to} \quad & \hat{E} \text{diag}(x) \hat{E}^T + (1/n) \mathbf{1}\mathbf{1}^T \succ 0. \end{aligned}$$

whose solution provides the optimal estimated graph Laplacian with the desired structure.

**4.3.2 Gradient and Hessian of  $J(x)$**  We next derive the gradient and Hessian of  $J$  which can be used to form a second-order Taylor series approximation of  $J(x)$  around  $x^k$ ,

$$J(x^k + \tilde{x}) \approx J(x^k) + \nabla J(x^k)^T \tilde{x} + \frac{1}{2} \tilde{x}^T \nabla^2 J(x^k) \tilde{x}. \quad (4.5)$$

**Proposition 6.** *The gradient and the Hessian of  $J$  at  $x^k$  are*

$$\begin{aligned} \nabla J(x^k) &= \text{diag} \left( E^T (S - X^{-1}(x^k)) E \right) \\ \nabla^2 J(x^k) &= M(x^k) \circ M(x^k) \end{aligned} \quad (4.6)$$

where  $\circ$  denotes the elementwise (Hadamard) product and

$$\begin{aligned} X^{-1}(x^k) &:= \left( E D_{x^k} E^T + (1/n) \mathbf{1}\mathbf{1}^T \right)^{-1}, \\ M(x^k) &:= E^T X^{-1}(x^k) E. \end{aligned}$$

*Proof.* Utilizing the second order expansion of the log-determinant function we have

$$J(x^k + \tilde{x}) - J(x^k) \approx \text{trace} \left( E^T (S - X^{-1}(x^k)) E D_{\tilde{x}} \right) + \frac{1}{2} \text{trace} \left( D_{\tilde{x}} E^T X^{-1}(x^k) E D_{\tilde{x}} E^T X^{-1}(x^k) E \right).$$

The expressions in (4.6) can be established using a sequence of straightforward algebraic manipulations in conjunction with the use of the commutativity invariance of the trace function and the following properties for a matrix  $T$ , a vector  $\alpha$ , and a diagonal matrix  $D_\alpha$ ,

$$\begin{aligned} \text{trace} (T D_\alpha) &= \alpha^T \text{diag} (T) \\ \text{trace} (D_\alpha T D_\alpha T) &= \alpha^T (T \circ T) \alpha. \end{aligned}$$

□

**4.3.3 Algorithm** Our algorithm is based on building the second-order Taylor series expansion of the smooth part of the objective function  $J$  in (NI) around the current iterate  $x^k$ . Approximation  $J$  in (NI) with (4.5),

$$\begin{aligned} \underset{\tilde{x}}{\text{minimize}} \quad & \nabla J(x^k)^T \tilde{x} + \frac{1}{2} \tilde{x}^T \nabla^2 J(x^k) \tilde{x} + \gamma \|x^k + \tilde{x}\|_1 \\ \text{subject to} \quad & E \text{diag} (x^k + \tilde{x}) E^T + (1/n) \mathbf{1}\mathbf{1}^T \succ 0. \end{aligned} \tag{4.7}$$

We use the coordinate descent algorithm to determine the Newton direction. Let  $\tilde{x}$  denote the current iterate approximating the Newton direction. By perturbing  $\tilde{x}$  in the direction of the  $i$ th standard basis vector  $e_i \in \mathbb{R}^m$ ,  $\tilde{x} + \mu_i e_i$ , the objective function in (4.7) becomes

$$\nabla J(x^k)^T (\tilde{x} + \mu_i e_i) + \frac{1}{2} (\tilde{x} + \mu_i e_i)^T \nabla^2 J(x^k) (\tilde{x} + \mu_i e_i) + \gamma |x_i^k + \tilde{x}_i + \mu_i|.$$

Elimination of constant terms allows us to express (4.7) as

$$\underset{\mu_i}{\text{minimize}} \quad \frac{1}{2} a_i \mu_i^2 + b_i \mu_i + \gamma |c_i + \mu_i| \tag{4.8}$$

where  $(a_i, b_i, c_i, x_i^k, \tilde{x}_i)$  are the problem data,

$$\begin{aligned} a_i &:= \mathbf{e}_i^T \nabla^2 J(x^k) \mathbf{e}_i \\ b_i &:= (\nabla^2 J(x^k) \mathbf{e}_i)^T \tilde{x} + \mathbf{e}_i^T \nabla J(x^k) \\ c_i &:= x_i^k + \tilde{x}_i. \end{aligned}$$

The explicit solution to (4.8) is given by

$$\mu_i = -c_i + \mathcal{S}_{\gamma/a_i}(c_i - b_i/a_i),$$

where  $\mathcal{S}_\kappa(y) = \text{sign}(y) \max(|y| - \kappa, 0)$ , is the soft-thresholding function.

After the Newton direction  $\tilde{x}$  has been computed, we determine the step-size  $\alpha$  via backtracking. This guarantees positive definiteness of the strengthened graph Laplacian and sufficient decrease of the objective function. We use Armijo rule to find an appropriate step-size such that  $E \text{diag}(x^k + \alpha \tilde{x}) E^T + (1/n) \mathbf{1}\mathbf{1}^T$  is positive definite matrix and

$$\begin{aligned} J(x^k + \alpha \tilde{x}) + \gamma \|x^k + \alpha \tilde{x}\|_1 &\leq J(x^k) + \gamma \|x^k\|_1 + \\ &\alpha \sigma \left( \nabla J(x^k)^T \tilde{x} + \gamma \|x^k + \alpha \tilde{x}\|_1 - \gamma \|x^k\|_1 \right). \end{aligned}$$

There are two computational aspects in our work which lead to suitability of this algorithm for large-scale networks.

- **Active set strategy**

We propose an active set prediction strategy as an efficient method to solve the problem (NI) for large values of  $\gamma$ . It is an effective means for determining which directions need to be updated in the coordinate descent algorithm. The classification of a variable as either active or inactive is based on the values of  $x_i^k$  and the  $i$ th component of the gradient vector  $\nabla J(x^k)$ . Specifically, the  $i$ th search direction is only inactive if

$$x_i^k = 0 \quad \text{and} \quad |\mathbf{e}_i^T \nabla J(x^k)| < \gamma - \epsilon$$

where  $\epsilon > 0$  is a small number (e.g.,  $\epsilon = 0.0001\gamma$ ). At each outer iteration, the Newton search direction is obtained by solving the optimization problem over the

set of active variables. The size of active sets is small for large values of the regularization parameter  $\gamma$ .

- **Memory saving**

Computation of  $b_i$  in (4.8) requires a single vector inner product between the  $i$ th column of the Hessian and  $\tilde{x}$ , which typically takes  $O(m)$  operations. To avoid direct multiplication, in each iteration after finding  $\mu_i$ , we update the vector  $\nabla^2 J(x^k)^T \tilde{x}$  using the correction term  $\mu_i(E^T X^{-1} \xi_i) \circ ((X^{-1} \xi_i)^T E)^T$  and take its  $i$ th element to form  $b_i$ . Here,  $\xi_i$  is the  $i$ th column of the incidence matrix of the controller graph. This also avoids the need to store the Hessian of  $J$ , which is an  $m \times m$  matrix, thereby leading to a significant memory saving. Moreover, the  $i$ th column of  $\nabla^2 J(x^k)$  and the  $i$ th element of the gradient vector  $\nabla J(x^k)$  enter into the expression for  $b_i$ . On the other hand,  $a_i$  is determined by the  $i$ th diagonal element of the Hessian matrix  $\nabla^2 J(x^k)$ . All of these can be obtained directly from  $\nabla^2 J(x^k)$  and  $\nabla J(x^k)$  which are formed in each outer iteration without any multiplication.

Our problem is closely related to the problem in [124]. The objective function there has the form  $f(x) = J(x) + g(x)$ , where  $J(x)$  is smooth over the positive definite cone, and  $g(x)$  is a separable non-differentiable function. In our problem formulation,  $J(x)$  is smooth for  $E \text{diag}(x) E^T + (1/n) \mathbf{1}\mathbf{1}^T \succ 0$  while the non-smooth part is the  $\ell_1$  norm which is separable. Thus, convergence can be established using similar arguments. According to [124, Theorems 1,2], the quadratic approximation method converges to the unique global optimum of (NI) and at super-linear rate.

The optimality condition for any  $x^*$  that satisfies  $E \text{diag}(x^*) E^T + (1/n) \mathbf{1}\mathbf{1}^T \succ 0$  is given by

$$\nabla J(x^*) + \gamma \partial \|x^*\|_1 \in 0,$$

where  $\partial \|x^*\|_1$  is the subgradient of the  $\ell_1$  norm. This means that for any  $i$

$$\nabla_i J(x^*) \in \begin{cases} -\gamma, & x_i > 0; \\ \gamma, & x_i < 0; \\ [-\gamma, \gamma], & x_i = 0. \end{cases}$$

The stopping criterion is to check the norm of  $\nabla J(x)$  and the sign of  $x$  to make sure that  $x$  is the optimal solution.

**4.4 Computational experiments** We next illustrate the performance of our customized algorithm. We have implemented our algorithm in Matlab, and all tests were done on 3.4 GHz Core(TM) i7-3770 Intel(R) machine with 16GB RAM.

The problem (NI) is solved for different values of  $\gamma$  using the path-following iterative reweighted algorithm [125] with  $\varepsilon = 10^{-5}$ . The initial weights are computed using the solution to (NI) with  $\gamma = 0$  (i.e., the optimal centralized vector of the edge weights). We then adjust  $\varepsilon = 0.001\|x\|_2$  at each iteration. Topology identification is followed by the polishing step described in Section 4.3.1. In the figures, we use black dots to represent nodes, blue lines to identify the original graph, and red lines to denote the edges in the estimated sparse network. In all examples, we set the tolerance for the stopping criterion to  $10^{-4}$ .

**4.4.1 Network identification** We solve the problem of identification of a sparse network using sample covariance matrix for 500 logarithmically-spaced values of  $\gamma \in [0.1, 1000]$ . The sample covariance matrix  $S$  is obtained by sampling the nodes of the stochastically-forced undirected unweighted network whose topology is shown in Fig. 4.1a. To generate samples, we conducted 20 simulations of system (4.1) forced with zero-mean unit variance band-limited white noise  $d$ . The sample covariance matrix is averaged over all simulations and asymptotically converges to the steady-state covariance matrix. The incidence matrix  $E$  in (NI) contains all possible edges.

Empirically, we observe that after about 5 seconds the sample covariance matrix converges to the steady-state covariance. First, we sample the states after 3 seconds, so the sample covariance matrix we compute is different than the true steady-state covariance matrix. For (NI) solved with this problem data, Figures 4.3 and 4.1c illustrate the topology of the identified networks for the minimum and maximum values of  $\gamma$ . In Fig. 4.1d, the blue line shows an edge in the original network that has not been recovered by the algorithm for the largest value of  $\gamma$ . The red lines in this figure show two extra edges in the estimated network for the smallest value of  $\gamma$  which were not present in the original graph. Next, we solve the problem (NI) using a sample

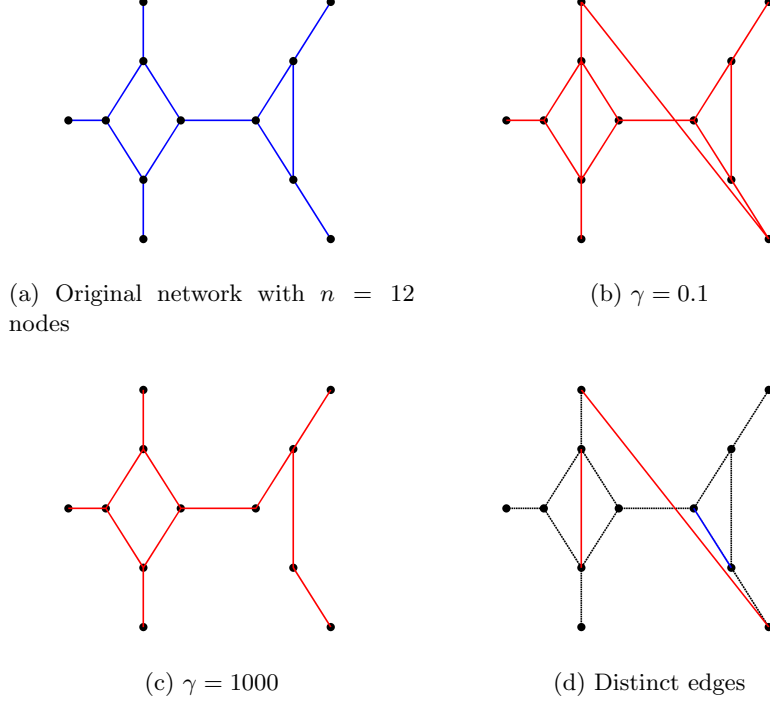


Figure 4.1: The problem of identification of sparse networks using sample covariance matrix for a network with  $n = 12$  nodes.

covariance matrix generated by sampling the node states after 15 seconds. In this case, the sample covariance matrix is closer to the steady-state covariance matrix than the previous experiment. As shown in Fig. 4.2a, the identified network is exactly the same as the original network for  $\gamma = 0.1$ . If  $\gamma$  is further increased, a network sparser than the original is identified; see Fig. 4.2b.

For  $\gamma = 0$ , the relative error between the covariance matrix of the estimated network and the sample covariance matrix  $S$  is given by

$$\frac{\| (E \operatorname{diag}(x_c) E^T + (1/n) \mathbf{1} \mathbf{1}^T)^{-1} - S \|_F}{\|S\|_F} = 0.004\%,$$

where  $\|\cdot\|_F$  is the Frobenius norm and  $x_c$  is the solution to (NI) with  $\gamma = 0$ . As  $\gamma$  increases, the number of nonzero elements in the vector of the edge weights  $x$  decreases and the state covariance matrix gets farther away from the given sample covariance

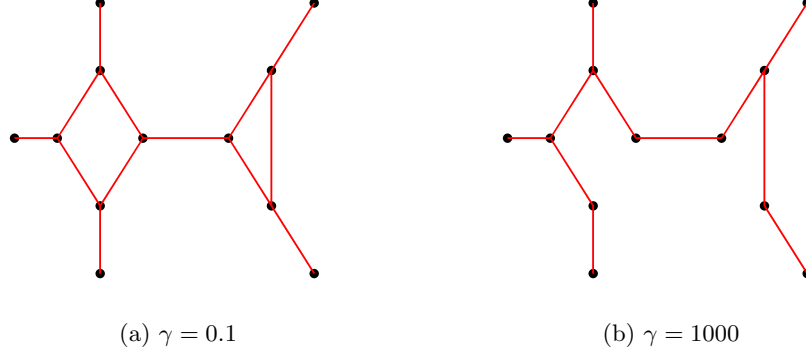


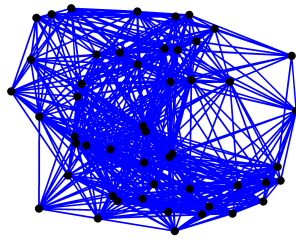
Figure 4.2: The problem of identification of sparse networks using sample covariance matrix for a network with  $n = 12$  nodes.

matrix. In particular, in the first experiment for  $\gamma = 1000$ , only twelve edges are chosen. Relative to the centralized network with the vector of the edge weights  $x_c$ , the identified sparse network in this case uses only 18.182% of the edges, i.e.,  $\text{card}(x)/\text{card}(x_c) = 18.182\%$  and achieves a relative error of 53.841%,  $(\|X^{-1} - S\|_F)/\|S\|_F = 53.841\%$ , with  $X = (E \text{diag}(x) E^T + (1/n) \mathbf{1} \mathbf{1}^T)$ . In the second experiment, the identified sparse network has 16.666% of the potential edges and achieves a relative error of 51.067%.

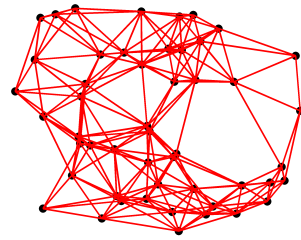
**4.4.2 Network sparsification** We next use (NI) to find a sparse representation of a dense consensus network. Inspired by [153], we generate a network by randomly distributing nodes in a  $10 \times 10$  box. A pair of nodes can communicate if the Euclidean distance between them,  $d(i, j)$ , is not greater than 5 units and the edge connecting them has weight  $\exp(-d(i, j))$ . The incidence matrix of the identified graph is selected to be equal to the incidence matrix of the given graph; i.e., the sparse network's edges are a subset of the original network's. Figure 4.3a shows a graph with 50 nodes. We use the reweighted  $\ell_1$  regularized Gaussian maximum likelihood estimation framework, for 200 logarithmically-spaced values of  $\gamma \in [0.01, 100]$  following by the polishing step. The sparse graph topologies identified for different values of  $\gamma$  are shown in figures 4.3b, 4.3c, and 4.3d. As  $\gamma$  increases, the identified graph becomes sparser.

For  $\gamma = 0.01$ , 222 edges are chosen to be in the sparse estimated network which is only 34.851% of the 637 potential edges. The network with these selected edges achieves

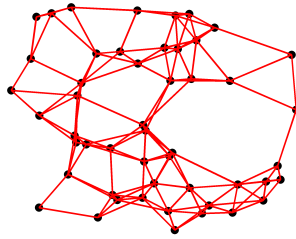




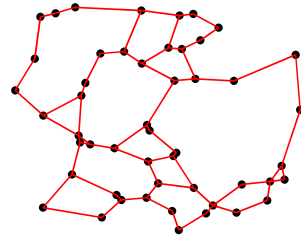
(a) Original network with  $n = 50$  nodes



(b)  $\gamma = 0.01$



(c)  $\gamma = 0.102$



(d)  $\gamma = 1.138$

Figure 4.3: The problem of sparsification of a network with  $n = 50$  nodes using sample covariance matrix.

a relative error of 29.965%,

$$\frac{\|X^{-1} - S\|_F}{\|S\|_F} = 29.965\%.$$

For the largest value of the sparsity-promoting parameter,  $\gamma = 1.138$ , only 64 edges are present (10.047% of the potential edges) in the estimated graph that gets a relative error of 207.493%.

To provide a point of comparison, we compare the performance of our algorithm to a simple truncation scheme. In this scheme, the edge with the smallest weights that does not disconnect the network is iteratively removed until the network has the desired sparsity. After identifying the topology in this way, the polishing step optimizes the edge weights of the selected set of edges.

Figure 4.4 shows the relative errors of our algorithm (in red dashed lines) and the truncation algorithm (in blue solid lines) on a log scale against the number of removed edges. As the number of edges in the estimated graph decreases, the relative error of both algorithms increases. The relative error of network topologies identified by our algorithm is much smaller than the error of those identified by the truncation algorithm, and thus our customized algorithm outperforms the truncation method. In particular, when 573 edges are removed, the relative errors for our customized algorithm and the truncation algorithm are 2.075 and 8.977, respectively.

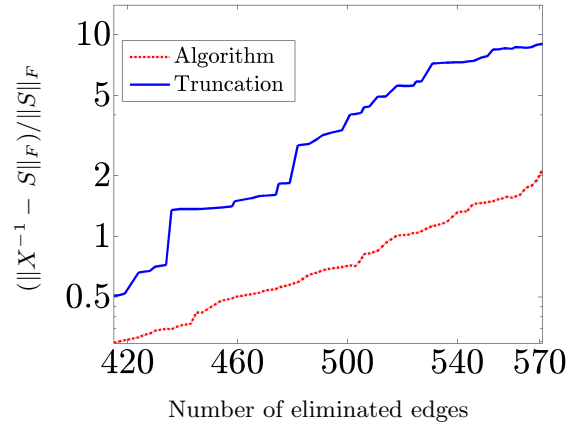


Figure 4.4: Relative error in log-scale for the sparsification problem of a network with  $n = 50$  nodes using sample covariance matrix.

**4.5 Concluding Remarks** We have developed a method for identifying the topology of an undirected consensus network using available statistical data. In order to promote network sparsity, we introduce a convex optimization framework aimed at finding the solution to the  $\ell_1$ -regularized maximum likelihood problem. This problem is closely related to the problem of sparse inverse covariance estimation that has received significant attention in the literature. In our setup, additional structure arises from the requirement that data is generated by an undirected consensus network. By exploiting the structure of the problem, we develop an efficient algorithm based on the sequential quadratic approximation method in which the search direction is determined using coordinate descent with active set strategy. Several examples have been provided to illustrate utility of the method and efficiency of the customized algorithm.

## Chapter 5

# Topology identification via growing a Chow-Liu tree network

We study the problem of sparse interaction topology identification using sample covariance matrix of the states of the network. We postulate that the statistics are generated by a stochastically-forced undirected consensus network with unknown topology in which some of the nodes may have access to their own states. We first propose a method for topology identification using a regularized Gaussian maximum likelihood framework where the  $\ell_1$  regularizer is introduced as a means for inducing sparse network topology. We also develop a method based on growing a Chow-Liu tree that is well-suited for identifying the underlying structure of large-scale systems. We apply this technique to resting-state functional MRI (fMRI) data as well as synthetic datasets to illustrate the effectiveness of the proposed approach.

**5.1 Introduction** In this chapter, we develop a convex optimization algorithm for identifying sparse interaction topology using the sample covariance matrix of the states of the network. First, we utilize an  $\ell_1$ -regularized Gaussian maximum likelihood estimator that has been commonly used for recovering sparse inverse covariance matrices [139–141]. We show that the performance of graphical-LASSO can improve significantly by imposing additional structure on the problem and using reweighting schemes [125]. In particular, our framework overcomes challenges that standard graphical-LASSO faced and performs well for the case study in [64]. Moreover, inspired

by [64], we combine the Chow-Liu algorithm [154] with the techniques for growing networks developed in [106] to identify the underlying structure of an undirected consensus network. Constructing the Chow-Liu tree from statistical data does not require any matrix inversion; thereby, it is well-suited for large-scale problems. Furthermore, we have developed efficient algorithms [106] for growing connected resistive consensus networks. Herein, we demonstrate that combining these two algorithms yields an efficient method for recovering the network topology in large-scale systems.

Our presentation is organized as follows. In Section 5.2, we discuss the properties of consensus networks and formulate the problem of topology identification using sparse inverse covariance matrix estimation. We also briefly comment on the customized second-order algorithm based on the proximal Newton method to solve the  $\ell_1$ -regularized Gaussian maximum likelihood estimation problem. In Section 5.3, we develop an algorithm for growing a Chow-Liu tree graph in order to identify the network that yields close approximation of a given sample covariance matrix. In Section 5.4, we use computational experiments to illustrate features of our method. In particular, we employ our algorithm to identify the underlying functional network of the human brain using FMRI data. Finally, in Section 5.5, we conclude with a brief summary.

## 5.2 Topology identification via structured graphical-LASSO

The problem of topology identification using a sample covariance matrix for stochastically forced undirected consensus networks has been studied in Chapter 4. In consensus networks, each node updates its own state using localized information exchange with the neighbors. Two nodes are neighbors if an edge connects them together. Herein, we consider a network that leader nodes are equipped with absolute information about their own states. In Chapter 4, we showed that the underlying topology of an undirected consensus network can be identified using Gaussian maximum likelihood estimator. In this chapter, we formulate the topology identification problem for undirected consensus networks with leader nodes and provide two algorithms to solve this problem.

Consider an undirected network with  $n$  nodes governed by

$$\dot{\psi}_i = \sum_{j \in \mathcal{N}_i} x_{ij} (\psi_j - \psi_i) - z_i \psi_i + w_i,$$

where each node  $i$  updates its own state using relative information exchange with its neighbors in the set  $\mathcal{N}_i$ . Moreover, certain nodes, the so-called leaders, have access to their own states. Here,  $z_i$  is the weight that the  $i$ th node sets for the absolute measurement,  $x_{ij}$  is the edge weight, and  $w_i$  is an exogenous stochastic disturbance. The  $i$ th node is a leader if  $z_i \neq 0$  and it is a follower if  $z_i = 0$ . By concatenating all the states in a vector  $\psi \in \mathbb{R}^n$ , the consensus dynamics can be written as

$$\dot{\psi} = -(L_x + D_z)\psi + w \quad (5.1)$$

Here,  $L_x \in \mathbb{R}^{n \times n}$  is the graph Laplacian of the consensus network and  $D_z \in \mathbb{R}^{n \times n}$  is a diagonal matrix with the  $i$ th diagonal entry  $z_i$ . The incidence matrix  $E$  of the underlying graph represents the edges in the network. The  $l$ th column of this matrix is given by

$$\xi_l = e_i - e_j,$$

that demonstrates the  $l$ th edge between the nodes  $i$  and  $j$ . Here,  $e_i \in \mathbb{R}^n$  is the  $i$ th basis vector. By using the incidence matrix, the Laplacian matrix  $L_x$  can be written as

$$L_x := \sum_{l=1}^m x_l \xi_l \xi_l^T = E \text{diag}(x) E^T$$

where  $\text{diag}(x)$  is a diagonal matrix with the edge weights  $x \in \mathbb{R}^m$  in its diagonal.

Given that the covariance of the disturbance is a multiple of the identity matrix  $I$ , the steady-state covariance of  $\psi$ ,

$$\Sigma := \lim_{t \rightarrow \infty} \mathbf{E}(\psi \psi^T),$$

can be computed as the solution to the associated algebraic Lyapunov equation

$$(L_x + D_z)\Sigma + \Sigma(L_x + D_z) = 2I.$$

Thus, the steady state covariance can be explicitly computed as

$$\Sigma = (L_x + D_z)^{-1}. \quad (5.2)$$

The inverse of the steady-state covariance matrix of the states of the system can be determined by the structure of the underlying graph that connects the  $n$  nodes. Thus, by using a sampled second-order statistics and estimating the inverse covariance matrix, the underlying topology of an undirected consensus network with leaders can be identified. The problem of sparse covariance estimation has received significant attention recently [124, 144, 150, 151]. Relative to these works, our optimization problem has additional structure coming from the dynamics of undirected consensus networks. Moreover, compared to previous chapter, we consider consensus networks with leaders and introduce a new algorithm that is convenient for solving large-scale problems.

We first generalize the proposed algorithm based on the structured graphical-LASSO in Chapter 4 to solve the problem of topology identification in undirected consensus networks with leaders. It is well-known that the estimation of the inverse covariance matrix  $X$  can be obtained as the solution to the regularized maximum log-likelihood problem [141],

$$\begin{aligned} & \underset{X}{\text{minimize}} \quad -\log \det(X) + \text{trace}(SX) + \gamma \|F \circ X\|_1 \\ & \text{subject to} \quad X \succ 0, \end{aligned} \tag{5.3}$$

where  $S$  is the sample covariance matrix,  $\gamma$  is a positive regularization parameter,  $F$  is the weight matrix, and  $\|F \circ X\|_1 := \sum F_{ij} |X_{ij}|$  is the weighted  $\ell_1$  norm of the matrix  $X$ . By substituting the expression (5.2) for the inverse covariance matrix in (5.3) and using the incidence matrix  $E$ , the topology of a network that generates close approximation of a given sample covariance matrix can be identified by solving the following problem,

$$\begin{aligned} & \underset{x, z}{\text{minimize}} \quad J(x, z) + \gamma_1 \sum_{l=1}^m f_l |x_l| + \gamma_2 \sum_{k=1}^N g_k |z_k| \\ & \text{subject to} \quad E \text{diag}(x) E^T + \text{diag}(z) \succ 0, \end{aligned} \tag{NI}$$

where

$$J(x, z) = -\log \det(E \text{diag}(x) E^T + \text{diag}(z)) + \text{trace}(S(E \text{diag}(x) E^T + \text{diag}(z))).$$

Moreover,  $f \in \mathbb{R}^m$  and  $g \in \mathbb{R}^N$  are the vectors of non-negative weights and  $(\gamma_1, \gamma_2)$

are the positive regularization parameters. An effective heuristic for weight selection is given by the iterative reweighted algorithm where the weights  $f$  and  $g$  are inversely proportional to the magnitudes of  $x$  and  $z$  in the previous iteration [125]. Problem (NI) is a convex but non-smooth optimization problem where the optimization variables are the vector of the edge weights  $x \in \mathbb{R}^m$  and the vector of leaders weights  $z \in \mathbb{R}^N$ . Relative to our prior work in the previous chapter, our optimization problem has additional structure induced by presence of the leader nodes.

The algorithm based on the sequential quadratic approximation of the smooth part of the objective function in Section 4.3 can be utilized for solving (NI) with minor changes. The difference is that the optimization variable size has increased from  $m$  to  $m + N$ . This method benefits from exploiting second-order information of the smooth part of the objective function and from computing the Newton direction using cyclic coordinate descent [145] over the set of active variables. We solve the problem (NI) for different values of  $(\gamma_1, \gamma_2)$  using a path-following iterative reweighted algorithm. The topology identification then is followed by a polishing step to debias the identified edge weights. We next propose an algorithm to solve the topology identification problem of large-scale networks.

**5.3 Growing a Chow-Liu tree network** In this section, we discuss an alternative algorithm for identifying the underlying network topology which is well-suited for large-scale systems. In order to find the underlying network structure using statistical data, the Chow-Liu tree algorithm [154] can be utilized. This method does not require any matrix inversion; thereby, suitable for large-scale usage. However, as discussed in [64], it causes false positives and negatives when using it for identifying the topology of disconnected networks or networks with cycles, respectively. Herein, we propose a framework in order to combine the Chow-Liu tree and the reweighted graphical-LASSO algorithms for identifying the structure of connected networks with cycles.

We consider the same consensus network in (5.1) and we assume that the sample covariance matrix  $S$  is given. In order to use the Chow-Liu algorithm, the mutual information matrix  $M$  should be constructed from the sample covariance matrix. Assuming



Gaussian distribution for the noise  $w$ , the mutual information is given by

$$M_{ij} = \frac{1}{2} \log \left( \frac{S_{ii} S_{jj}}{S_{ii} S_{jj} - S_{ij}^2} \right),$$

where  $S_{ij}$  is the  $ij$ th element of the matrix  $S$ . We only use the  $n(n-1)/2$  off-diagonal elements of this symmetric matrix to construct the Chow-Liu tree. A spanning tree of a graph with  $n$  nodes has  $(n-1)$  edges. To build the Chow-Liu tree, we sort the elements of the mutual information matrix and choose the biggest  $(n-1)$  of them that do not create cycles [154].

After finding the underlying tree network that generates close approximation of the sample covariance matrix, our goal is to add a certain number of edges to the tree graph in order to enhance the closed-loop performance [106]. The performance is measured by the proximity of the second-order statistical data generated by the network to the given sample covariance matrix.

Consider an undirected consensus tree network,

$$\dot{\psi} = -L_t \psi + u + w, \quad (5.4)$$

where  $w$  and  $u$  are the exogenous disturbance and the control input, respectively and  $L_t$  is the graph Laplacian of the tree network that is obtained using the Chow-Liu algorithm. The goal is to improve the performance of this system by growing the tree network. We approach this problem as a feedback design problem with

$$u = -(L_x + D_z) \psi, \quad (5.5)$$

where  $D_z$  is a diagonal matrix with the  $i$ th diagonal entry  $z_i$  and the symmetric feedback-gain matrix  $L_x$  is required to have the Laplacian structure. Since a nonzero  $ij$ th element of  $L_x$  corresponds to an edge between the nodes  $i$  and  $j$ , the communication structure in the controller graph is determined by the sparsity pattern of the matrix  $L_x$ . Moreover, the  $i$ th node is a leader if  $z_i$  is nonzero. By substituting the control input  $u$  from (5.5) in (5.4)

$$\dot{\psi} = -(L_t + L_x + D_z) \psi + w. \quad (5.6)$$

For a computed  $L_t$  from the Chow-Liu algorithm, our objective is to design the topology  $L_x$  and to identify the leader nodes in the network in order to achieve the desired tradeoff between the controller sparsity and the network performance. The performance is quantified by the proximity of the steady-state covariance matrix of the closed-loop system to the sample covariance matrix.

Next, we are going to establish a relation between the closed-loop graph Laplacian and the inverse covariance matrix of the network. The steady-state covariance of  $\psi$  is given by

$$\Sigma = (L_t + L_x + D_z)^{-1}, \quad (5.7)$$

where  $L_x = E \text{diag}(x) E^T$ . Thus, the problem of identifying the sparse topology of a network, i.e., finding  $L_x$  and  $D_z$ , that generates close approximation of a given sample covariance matrix is equivalent to sparse inverse covariance estimation problem. This can be achieved by solving a similar regularized maximum log-likelihood problem to (5.3) with one main difference. The inverse covariance matrix  $X$  is the summation of the Laplacian matrices of the tree plant network  $L_t$  and the controller network  $L_x + D_z$ . Thus, the problem of growing a tree network in order to match the available statistical data can be formulated as

$$\begin{aligned} \underset{x, z}{\text{minimize}} \quad & J(x, z) + \gamma_1 \sum_{l=1}^m f_l |x_l| + \gamma_2 \sum_{k=1}^N g_k |z_k| \\ \text{subject to} \quad & L_t + E \text{diag}(x) E^T + \text{diag}(z) \succ 0, \end{aligned} \quad (5.8)$$

where

$$\begin{aligned} J(x, z) = & -\log \det (L_t + E \text{diag}(x) E^T + \text{diag}(z)) + \\ & \text{trace} (S (E \text{diag}(x) E^T + \text{diag}(z))). \end{aligned}$$

In the case of resistive networks (i.e., all the edge weights are nonnegative), since the plant network is given by a tree graph, the closed-loop network is connected; thereby, the optimization problem simplifies to

$$\begin{aligned} \underset{x, z}{\text{minimize}} \quad & J(x, z) + \gamma_1 \sum_{l=1}^m f_l x_l + \gamma_2 \sum_{k=1}^N g_k z_k \\ \text{subject to} \quad & x \geq 0, \quad z \geq 0. \end{aligned} \quad (5.9)$$

In this scenario, the topology identification problem turn into the problem of growing a tree network and the positive definiteness constraint simplifies to nonnegativity constraints of the vectors  $x$  and  $z$ . Thus, several optimization algorithms can be employed to solve this problem efficiently for large-scale networks.

**5.4 Computational experiments** We next illustrate the performance of our methods by employing them to identify the topology of an RC network with 10 nodes and a tree structure shown in Fig. 5.1. The voltages of the nodes form the states of the system. In this network, node 5 is grounded with  $z_5 = 4$  and all other edge weights are one which implies that node 5 is a leader. This example is borrowed from [64].

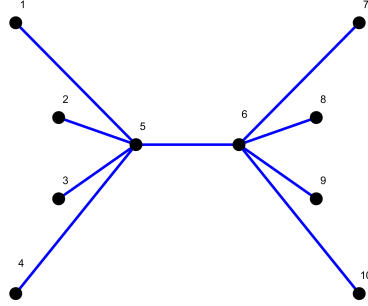


Figure 5.1: The RC tree network with  $n = 10$  nodes.

**5.4.1 Structured graphical-LASSO algorithm** Assume that infinite samples of the nodes' voltages are available; thereby, the sample covariance matrix  $S$  is equal to the exact solution of the Lyapunov equation. Moreover, we set  $\gamma_1 = \gamma_2 = \gamma$ . In this case, the structured graphical-LASSO algorithm in Section 5.2 can completely recover the underlying network topology for different values of  $\gamma$ . This example has been previously studied in [64]. They show that when the sample covariance matrix is precise, the graphical-LASSO algorithm results in 5 false positives and negatives. However, by adding a structural constraint to the problem and using a reweighting scheme, we showed that the same algorithm can recover the network topology with zero error.

Next, we utilize our method to solve the problem by using a sample covariance

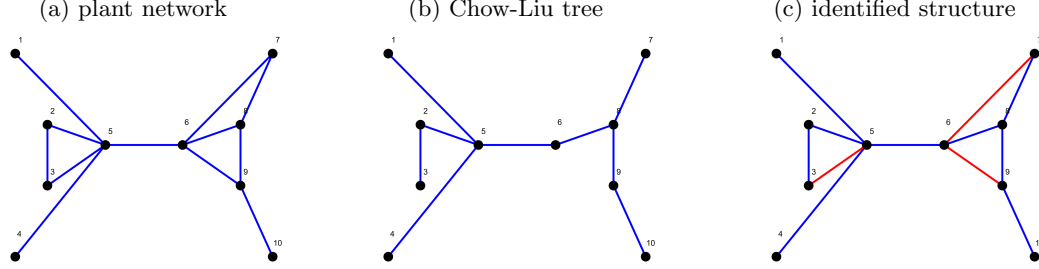


Figure 5.2: (a) Plant network; (b) constructed Chow-Liu tree network; and (c) the identified network from growing the Chow-Liu tree.

matrix which is not very close to the actual covariance matrix and is constructed from only 80 samples. The algorithm is again able to recover the network topology and to identify the leader for different values of  $\gamma$ .

It is worth to note that the performance of this method deteriorates if we replace the reweighted  $\ell_1$  norm scheme with the  $\ell_1$  norm. In particular, by eliminating the reweighted  $\ell_1$  norm, we observed the effect of grounding one of the nodes with high capacitance. Although the network topology will be identified for some values of  $\gamma$ , by increasing  $\gamma$  (to very large value), the algorithm chooses the optimal edges in the same way as [64]. In particular, it ignores the connections between nodes 1 to 5 because of their low variances. In the next example, we illustrate the effectiveness of growing a Chow-Liu tree by using it on a synthetic dataset.

**5.4.2 Topology identification via growing a Chow-Liu tree** In this section, the second method is utilized to identify the underlying structure of a network with cycles. The original plant network is shown in Fig. 5.2a. We first assume that the sample covariance matrix  $S$  is equal to the exact solution of the Lyapunov equation. We form the mutual information matrix and construct the Chow-Liu tree accordingly which is shown in Fig. 5.2b. We next grow this tree network in order to enhance the performance of the closed-loop system. In particular, we solve the problem (5.8) to find the leader nodes and the Laplacian matrix of the controller graph  $L_x$ . In this case, our algorithm can completely recover the underlying network topology for different values of  $\gamma$ . Next, we employ this algorithm to identify the topology of a larger network with

real data to evaluate its performance.

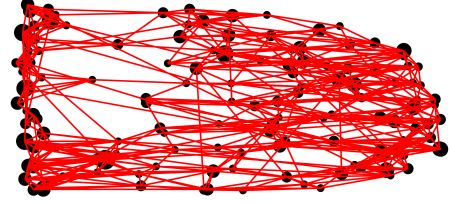
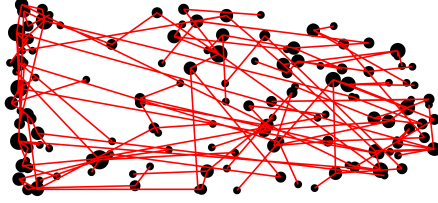
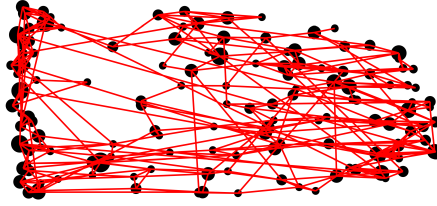
**5.4.3 FMRI dataset** The FMRI technique detects the activity of a region in the brain by measuring the blood flow to that region. Since the blood flow increases in an active part of the brain, the functioning regions can be identified by monitoring the blood flow. The functional connectivity structure between different regions can be revealed by utilizing different thresholding techniques [59,60]. The results indicate that different regions of the brain that are not anatomically connected act closely together and are functionally linked. Moreover, the previous studies have shown that the human brain has small-world network properties [60].

In this section, we employ the second algorithm based on growing a Chow-Liu tree to identify the underlying functional network of the human brain. The sample covariance matrix is computed using the resting-state FMRI data of 20 healthy patients [60]. In the resting-state FMRI, the patients are asked to close their eyes and try not to think. The studies have shown that even in the rest state, the brain is highly active and different regions of the brain are communicating with each other [155]. We collect 134 samples from 140 cortical brain regions (nodes) in the right hemisphere. The sample correlation matrix for each patient is a  $140 \times 140$  matrix and can be computed using the time series data. The sample covariance matrices are not invertible since the number of samples is smaller than the number of the nodes in the network. Thus, we use our proposed algorithm to estimate the inverse covariance matrix and to identify the underlying network structure of the human brain.

First, we form the mutual information matrix and construct the Chow-Liu tree Fig 5.3a. Next, we grow the obtained tree network to identify the remained edges and improve the performance of the closed-loop system. We set  $\gamma_1 = \gamma_2 = \gamma$ . The identified networks for a randomly chosen patient are shown in Fig 5.3. In particular, as the sparsity promoting parameter  $\gamma$  increases, the identified network gets sparser.

This example has been previously studied in [156]. Their results show that the nodes that are located in the lower left corner of the graphs are highly connected to their neighboring nodes. They compare this pattern of connectivity with the false positives created by their algorithm in a synthetic RC circuit and conclude that the high number of edges in that area is false positives created by the same phenomenon

(a) Chow-Liu tree graph with 139 edges

(b)  $\gamma = 0.1$ (c)  $\gamma = 1$ Figure 5.3: The identified networks for different values of  $\gamma$ .

in the circuit example. However, by adding a structural constraint to the problem and using a reweighting scheme, we showed that the underlying network can be recovered without high connectivity in the lower left corner. Moreover, the general shape of the identified network is consistent with the results reported in [60]. Furthermore, the small-world properties such as high clustering and high efficiency coefficients can be seen in the identified networks.

To conclude, it seems that using both an additional structural constraint and the reweighted  $\ell_1$  norm scheme can improve the performance of the graphical-LASSO algorithm significantly. Unlike the Chow-Liu algorithm that can be employed to construct tree networks only, our algorithm is more general and overcomes the challenges associated with the conventional algorithms proposed in [156].

**5.5 Concluding remarks** We have studied the problem of sparse topology identification of an undirected consensus network with leaders using second-order statistical data. The goal is to identify a sparse interaction topology using sample covariance matrix of the network state. We have introduced two algorithms based on regularized Gaussian maximum likelihood and growing a Chow-Liu tree. In the first algorithm, we propose a structured graphical-LASSO algorithm that uses the weighted  $\ell_1$  regularizer as a proxy for inducing sparse network topology. The other method is based on growing a Chow-Liu tree that is well-suited for identifying the underlying structure of large-scale networks. Several examples have been provided to demonstrate the performance of our framework.

## Part III

# Control theoretic approach for analysis of optimization algorithms



## Chapter 6

# Proximal gradient flow and Douglas-Rachford splitting dynamics: global exponential stability via integral quadratic constraints

Many large-scale and distributed optimization problems can be brought into a composite form in which the objective function is given by the sum of a smooth term and a nonsmooth regularizer. Such problems can be solved via a proximal gradient method and its variants, thereby generalizing gradient descent to a nonsmooth setup. In this chapter, we view proximal algorithms as dynamical systems and leverage techniques from control theory to study their global properties. In particular, for problems with strongly convex objective functions, we utilize the theory of integral quadratic constraints to prove global exponential stability of the differential equations that govern the evolution of proximal gradient and Douglas-Rachford splitting flows. In our analysis, we use the fact that these algorithms can be interpreted as variable-metric gradient methods on the forward-backward and the Douglas-Rachford envelopes and exploit structural properties of the nonlinear terms that arise from the gradient of the smooth part of the objective

function and the proximal operator associated with the nonsmooth regularizer. We also demonstrate that these envelopes can be obtained from the augmented Lagrangian associated with the original nonsmooth problem and establish conditions for global exponential convergence even in the absence of strong convexity.

**6.1 Introduction** Analysis of optimization algorithms from the system theoretic point of view has received significant recent attention [29–31]. In these references, the optimization algorithm is interpreted as a feedback interconnection in which the states converge to the optimal solution of the optimization problem.

In this chapter, we utilize techniques from control theory to establish global properties of proximal gradient flow and Douglas-Rachford (DR) splitting dynamics. These algorithms provide an effective tool for solving nonsmooth convex optimization problems in which the objective function is given by a sum of a differentiable term and a nondifferentiable regularizer. We exploit the fact that the proximal gradient flow can be interpreted as a variable-metric gradient method on forward-backward (FB) envelope [157–159] and show that this envelope can be obtained by restricting the proximal augmented Lagrangian to the manifold in which the dual variable is given by the negative gradient of the smooth part of the objective function. When this smooth part is strongly convex with a Lipschitz continuous gradient, we prove global exponential stability by utilizing the theory of IQCs [160]. We also generalize the Polyak-Lojasiewicz (PL) [161] condition to nonsmooth problems and show global exponential convergence of the FB envelope even in the absence of strong convexity. Finally, since the DR splitting algorithm [162] can be interpreted as a variable-metric gradient method on DR envelope [157], we utilize similar approach to establish global stability properties of the resulting continuous-time dynamics for strongly convex problems.

This chapter is structured as follows. In Section 6.2, we formulate the nonsmooth composite optimization problem and provide background material. In Section 6.3, we establish the global exponential stability of the proximal gradient flow dynamics for a problem with strongly convex objective function. Moreover, by exploiting the problem structure, we demonstrate the global exponential convergence of the forward-backward envelope even in the absence of strong convexity. In Section 6.4, we introduce a continuous-time gradient flow dynamics based on the celebrated Douglas-Rachford

splitting algorithm and utilize the theory of IQCs to prove global exponential stability for strongly convex problems. We offer concluding remarks in Section 6.5.

**6.2 Problem formulation and background** We consider a composite optimization problem,

$$\underset{x}{\text{minimize}} \quad f(x) + g(Tx) \quad (6.1)$$

where  $x \in \mathbb{R}^n$  is the optimization variable,  $T \in \mathbb{R}^{m \times n}$  is a given matrix,  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  is a continuously differentiable function with a Lipschitz continuous gradient, and  $g: \mathbb{R}^m \rightarrow \mathbb{R}$  is a nondifferentiable convex function. Such optimization problems arise in a number of different applications and depending on the structure of the functions  $f$  and  $g$ , different first- and second-order algorithms can be employed to solve them. We are interested in studying global convergence properties of methods based on proximal gradient flow algorithms. In what follows, we provide background material that we utilize in the rest of the chapter.

**6.2.1 Proximal operators and the associated envelopes** The proximal operator of a proper, lower semicontinuous, and convex function  $g$  is defined as

$$\mathbf{prox}_{\mu g}(v) := \underset{z}{\operatorname{argmin}} \left( g(z) + \frac{1}{2\mu} \|z - v\|_2^2 \right).$$

The value function of this optimization problem determines the associated Moreau envelope,

$$M_{\mu g}(v) := g(\mathbf{prox}_{\mu g}(v)) + \frac{1}{2\mu} \|\mathbf{prox}_{\mu g}(v) - v\|_2^2$$

which is a continuously differentiable function even when  $g$  is not [76],

$$\nabla M_{\mu g}(v) = \frac{1}{\mu} (v - \mathbf{prox}_{\mu g}(v)).$$

Combining the last two expressions yields,

$$M_{\mu g}(v) = g(\mathbf{prox}_{\mu g}(v)) + \frac{\mu}{2} \|\nabla M_{\mu g}(v)\|_2^2.$$

The Moreau envelope of  $g$  can be used to introduce the forward-backward (FB)

envelope [157–159] of the composite function

$$F(x) := f(x) + g(x).$$

The FB envelope is determined by the value function of the problem

$$\underset{v}{\text{minimize}} \quad J(x, v) \tag{6.2a}$$

where  $J$  approximates  $F$  via a simple quadratic expansion of the function  $f$  around  $x$ ,

$$\begin{aligned} J(x, v) &:= f(x) + \langle \nabla f(x), v - x \rangle + \frac{1}{2\mu} \|v - x\|_2^2 + g(v) \\ &= g(v) + \frac{1}{2\mu} \|v - (x - \mu \nabla f(x))\|_2^2 + f(x) - \frac{\mu}{2} \|\nabla f(x)\|_2^2. \end{aligned} \tag{6.2b}$$

The optimal solution of (6.2) is determined by

$$v_\mu^*(x) = \mathbf{prox}_{\mu g}(x - \mu \nabla f(x))$$

and it can be used to obtain the FB envelope of the function  $F$ ,

$$\begin{aligned} F_\mu(x) &:= J(x, v_\mu^*(x)) = J(x, \mathbf{prox}_{\mu g}(x - \mu \nabla f(x))) \\ &= f(x) - \mu \langle \nabla f(x), G_\mu(x) \rangle + \frac{\mu}{2} \|G_\mu(x)\|_2^2 + g(\mathbf{prox}_{\mu g}(x - \mu \nabla f(x))) \end{aligned} \tag{6.3}$$

where  $G_\mu$  is the generalized gradient map,

$$G_\mu(x) := \frac{1}{\mu} (x - \mathbf{prox}_{\mu g}(x - \mu \nabla f(x))). \tag{6.4}$$

Alternatively, the FB envelope  $F_\mu$  can be also expressed as

$$F_\mu(x) = f(x) + M_{\mu g}(x - \mu \nabla f(x)) - \frac{\mu}{2} \|\nabla f(x)\|_2^2. \tag{6.5}$$

Moreover, when  $f$  is twice continuously differentiable,  $F_\mu$  is continuously differentiable and its gradient is determined by [157],

$$\nabla F_\mu(x) = (I - \mu \nabla^2 f(x)) G_\mu(x). \tag{6.6}$$

The Douglas-Rachford (DR) envelope is another useful object that is obtained by evaluating the FB envelope at  $\mathbf{prox}_{\mu f}(x)$  [163],

$$F_{\mu}^{DR}(x) := F_{\mu}(\mathbf{prox}_{\mu f}(x)). \quad (6.7)$$

Alternatively, the DR envelope can be expressed as

$$F_{\mu}^{DR}(x) = M_{\mu g}(x - 2\mu \nabla M_{\mu f}(x)) + M_{\mu f}(x) - \mu \|\nabla M_{\mu f}(x)\|_2^2. \quad (6.8)$$

From the definition of the proximal operator of the continuously differentiable function  $f$ , we have

$$\mu \nabla f(\mathbf{prox}_{\mu f}(x)) + \mathbf{prox}_{\mu f}(x) - x = 0 \quad (6.9)$$

and, thus,

$$\nabla M_{\mu f}(x) = \nabla f(\mathbf{prox}_{\mu f}(x)). \quad (6.10)$$

Equality (6.7) follows from substituting the expression for  $\nabla M_{\mu f}(x)$  into (6.8), using equation (6.9), and leveraging the properties of the Moreau envelope,

$$\begin{aligned} F_{\mu}^{DR}(x) &= M_{\mu g}(x - 2\mu \nabla f(\mathbf{prox}_{\mu f}(x)) + f(\mathbf{prox}_{\mu f}(x)) + \\ &\quad \frac{1}{2\mu} \|\mathbf{prox}_{\mu f}(x) - x\|_2^2 - \mu \|\nabla f(\mathbf{prox}_{\mu f}(x))\|_2^2 \\ &= M_{\mu g}(\mathbf{prox}_{\mu f}(x) - \mu \nabla f(\mathbf{prox}_{\mu f}(x)) + f(\mathbf{prox}_{\mu f}(x)) - \frac{\mu}{2} \|\nabla f(\mathbf{prox}_{\mu f}(x))\|_2^2 \\ &= F_{\mu}(\mathbf{prox}_{\mu f}(x)). \end{aligned}$$

If  $f$  is twice continuously differentiable with  $\nabla^2 f(x) \preceq L_f I$  for all  $x$ , the DR envelope is continuously differentiable and its gradient is given by [163]

$$\nabla F_{\mu}^{DR}(x) = \frac{1}{\mu} (2\nabla \mathbf{prox}_{\mu f}(x) - I) G_{\mu}^{DR}(x) \quad (6.11)$$

where

$$\nabla \mathbf{prox}_{\mu f}(x) = (I + \mu \nabla^2 f(\mathbf{prox}_{\mu f}(x)))^{-1}$$

and

$$G_{\mu}^{DR}(x) := \mathbf{prox}_{\mu f}(x) - \mathbf{prox}_{\mu g}(2\mathbf{prox}_{\mu f}(x) - x). \quad (6.12)$$

**6.2.2 Proximal augmented Lagrangian** By introducing an auxiliary optimization variable  $z$ , problem (6.1) can be rewritten as follows,

$$\begin{aligned} & \underset{x, z}{\text{minimize}} && f(x) + g(z) \\ & \text{subject to} && Tx - z = 0. \end{aligned} \tag{6.13}$$

The augmented Lagrangian associated with constrained optimization problem (6.13) is given by,

$$\mathcal{L}_\mu(x, z; y) := f(x) + g(z) + \langle y, Tx - z \rangle + \frac{1}{2\mu} \|Tx - z\|_2^2$$

and the completion of squares yields,

$$\mathcal{L}_\mu = f(x) + g(z) + \frac{1}{2\mu} \|z - (Tx + \mu y)\|_2^2 - \frac{\mu}{2} \|y\|_2^2$$

where  $y$  is the Lagrange multiplier and  $\mu$  is a positive parameter. The minimizer of  $\mathcal{L}_\mu$  with respect to  $z$  is

$$z^*(x, y) = \mathbf{prox}_{\mu g}(Tx + \mu y)$$

and the evaluation of  $\mathcal{L}_\mu(x, z; y)$  along the manifold resulting from the explicit minimization over  $z$  yields the proximal augmented Lagrangian [81],

$$\begin{aligned} \mathcal{L}_\mu(x; y) &:= \mathcal{L}_\mu(x, z^*(x, y); y) \\ &= f(x) + M_{\mu g}(Tx + \mu y) - \frac{\mu}{2} \|y\|_2^2. \end{aligned} \tag{6.14}$$

This function is continuously differentiable with respect to both  $x$  and  $y$  and it can be used as a foundation for the development of different first- and second-order primal-dual methods for nonsmooth composite optimization [81, 108]. It is noteworthy that, for  $T = I$ , forward-backward envelope  $F_\mu(x)$  is obtained by restricting the proximal augmented Lagrangian  $\mathcal{L}_\mu(x; y)$  along the manifold  $y^*(x) = -\nabla f(x)$  resulting from KKT optimality conditions,

$$\begin{aligned} F_\mu(x) &:= \mathcal{L}_\mu(x; y^*(x)) = \mathcal{L}_\mu(x; y = -\nabla f(x)) \\ &= f(x) + M_{\mu g}(x - \mu \nabla f(x)) - \frac{\mu}{2} \|\nabla f(x)\|_2^2. \end{aligned}$$

Moreover, relation (6.7) between the two envelopes allows us to interpret the DR envelope as the proximal augmented Lagrangian evaluated at  $\mathbf{prox}_{\mu f}(x)$  and restricted along the manifold  $y^*(\mathbf{prox}_{\mu f}(x)) = -\nabla f(\mathbf{prox}_{\mu f}(x)) = -\nabla M_{\mu f}(x)$ ,

$$\begin{aligned} F_{\mu}^{DR}(x) &:= \mathcal{L}_{\mu}(\mathbf{prox}_{\mu f}(x); y^*(\mathbf{prox}_{\mu f}(x))) \\ &= M_{\mu f}(x) + M_{\mu g}(x - 2\mu \nabla M_{\mu f}(x)) - \mu \|\nabla M_{\mu f}(x)\|_2^2. \end{aligned}$$

**6.2.3 Strong convexity and Lipschitz continuity** The function  $f$  is strongly convex with parameter  $m_f$  if for any  $x$  and  $\hat{x}$ ,

$$f(\hat{x}) \geq f(x) + \langle \nabla f(x), \hat{x} - x \rangle + \frac{m_f}{2} \|\hat{x} - x\|_2^2$$

and equivalently,

$$\|\nabla f(x) - \nabla f(\hat{x})\|_2 \geq m_f \|x - \hat{x}\|_2. \quad (6.15)$$

The gradient of a continuously-differentiable function  $f$  is Lipschitz continuous with parameter  $L_f$  if for any  $x$  and  $\hat{x}$ ,

$$f(\hat{x}) \leq f(x) + \langle \nabla f(x), \hat{x} - x \rangle + \frac{L_f}{2} \|\hat{x} - x\|_2^2$$

and equivalently,

$$\|\nabla f(x) - \nabla f(\hat{x})\|_2 \leq L_f \|x - \hat{x}\|_2. \quad (6.16)$$

Moreover, if an  $m_f$ -strongly convex function  $f$  has an  $L_f$ -Lipschitz continuous gradient, the following inequality holds for any  $x$  and  $\hat{x}$ ,

$$\langle \nabla f(x) - \nabla f(\hat{x}), x - \hat{x} \rangle \geq \frac{m_f L_f}{m_f + L_f} \|x - \hat{x}\|_2^2 + \frac{1}{m_f + L_f} \|\nabla f(x) - \nabla f(\hat{x})\|_2^2. \quad (6.17)$$

Furthermore, the subgradient  $\partial g$  of a nondifferentiable function  $g$  is defined as the set of points  $z \in \partial g(x)$  that for any  $x$  and  $\hat{x}$  satisfy,

$$g(\hat{x}) \geq g(x) + z^T(\hat{x} - x). \quad (6.18)$$

**6.2.4 Proximal Polyak-Lojasiewicz inequality** The Polyak-Lojasiewicz (PL) condition can be used to establish linear convergence of standard gradient descent

method in the absence of strong convexity (or even convexity) [164]. For an unconstrained optimization problem with a non-empty solution set,

$$\underset{x}{\text{minimize}} \quad f(x)$$

where  $f$  is a twice differentiable function with a Lipschitz continuous gradient, the PL condition is given by

$$\|\nabla f(x)\|_2^2 \geq \gamma(f(x) - f^*)$$

where  $\gamma > 0$  and  $f^*$  is the optimal value of the function  $f$ .

We next provide the generalization of the PL condition for nonsmooth composite optimization problems (6.13) with  $T = I$ . For this class of problems, the proximal PL inequality holds for  $\mu \in (0, 1/L_f)$  if there exist  $\gamma > 0$  such that

$$\|G_\mu(x)\|_2^2 \geq \gamma(F_\mu(x) - F_\mu^*). \quad (6.19)$$

Here,  $L_f$  is the Lipschitz constant of  $\nabla f$ ,  $F_\mu$  is the FB envelope, and  $G_\mu$  is the generalized gradient map. In Appendix A, we show that (6.19) is equivalent to the condition provided in [164].

**6.3 Exponential stability of proximal algorithms** In this section, we briefly discuss the Arrow-Hurwicz-Uzawa gradient flow dynamics that can be used to solve (6.13) by computing the saddle points of the proximal augmented Lagrangian [81]. We then show that the proximal gradient method in continuous time can be obtained from the proximal augmented Lagrangian method by restricting the dual variable along the manifold  $y = -\nabla f(x)$ . Finally, we discuss global stability properties of proximal algorithms both in the presence and in the absence of strong convexity.

Continuous differentiability of the proximal augmented Lagrangian (6.14) can be utilized to compute its saddle points via the Arrow-Hurwicz-Uzawa gradient flow dynamic,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -(\nabla f(x) + T^T \nabla M_{\mu g}(Tx + \mu y)) \\ \mu(\nabla M_{\mu g}(Tx + \mu y) - y) \end{bmatrix}. \quad (6.20)$$

As shown in [81], these primal-descent dual-ascent gradient flow dynamics are globally exponentially stable for convex problems in which the matrix  $TT^T$  is invertible and the



smooth part of the objective function  $f$  is strongly convex.

For convex problems with  $T = I$  in (6.1),

$$\underset{x}{\text{minimize}} \quad f(x) + g(x) \quad (6.21)$$

the proximal gradient method,

$$x^{k+1} = \mathbf{prox}_{\alpha_k g}(x^k - \alpha_k \nabla f(x^k)) \quad (6.22)$$

with the stepsize  $\alpha_k \in (0, 1/L_f]$  can be used to solve (6.21), where  $L_f$  is the Lipschitz constant of  $\nabla f$ . In [157], it was demonstrated that (6.22) can be interpreted as a variable-metric gradient method on FB envelope,

$$\begin{aligned} x^{k+1} &= x^k - \alpha_k (I - \alpha_k \nabla^2 f(x))^{-1} \nabla F_{\alpha_k}(x^k) \\ &= x^k - \alpha_k G_{\alpha_k}(x^k). \end{aligned}$$

This interpretation can be utilized to compute the optimal solution to (6.21) using the continuous-time proximal gradient flow dynamics

$$\begin{aligned} \dot{x} &= -G_\mu(x) \\ &= -(\nabla f(x) + \nabla M_{\mu g}(x - \mu \nabla f(x))) \\ &= -\frac{1}{\mu} (x - \mathbf{prox}_{\mu g}(x - \mu \nabla f(x))). \end{aligned} \quad (6.23)$$

**Remark 8.** *Proximal gradient flow dynamics (6.23) are different from the subgradient flow dynamics associated with nonsmooth composite optimization problem (6.21). We note that proximal gradient algorithm (6.22) can be obtained via explicit forward Euler discretization of (6.23) with the stepsize  $\mu = \alpha_k$ . This should be compared and contrasted to a standard interpretation [76] in which (6.22) results from implicit backward Euler discretization of the subgradient flow dynamics associated with (6.21). We also note that (6.23) can be obtained by substituting  $-\nabla f(x)$  for the dual variable  $y$  in the  $x$ -update step of primal-descent dual-ascent gradient flow dynamics (6.20) with  $T = I$ .*

We next examine global stability properties of proximal gradient flow dynamics (6.23), first for strongly convex problems and then for the problems in which only the PL condition holds.

**6.3.1 Strongly convex problems** Herein, we utilize the theory of integral quadratic constraints to establish global asymptotic stability of proximal gradient flow dynamics (6.23) under the following assumption.

**Assumption 1.** *Let the differentiable part  $f$  of the objective function in (6.21) be strongly convex with parameter  $m_f$ , let  $\nabla f$  be Lipschitz continuous with parameter  $L_f$ , and let the regularization function  $g$  be proper, lower semicontinuous, and convex.*

Proximal gradient flow dynamics (6.23) can be expressed as a feedback interconnection of an LTI system

$$\begin{aligned}\dot{x} &= Ax + Bu \\ \xi &= Cx\end{aligned}\tag{6.24a}$$

with a nonlinear term,

$$u(x) := \mathbf{prox}_{\mu g}(x - \mu \nabla f(x)).\tag{6.24b}$$

Here,

$$A = -\frac{1}{\mu}I, \quad B = \frac{1}{\mu}I, \quad C = I\tag{6.24c}$$

and the corresponding transfer function  $H(s)$ ,  $\xi(s) = H(s)u(s)$ , is

$$H(s) = C(sI - A)^{-1}B = \frac{1}{\mu s + 1}I.\tag{6.24d}$$

The following lemma combines firm nonexpansiveness of  $\mathbf{prox}_{\mu g}$ , strong convexity of  $f$ , and Lipschitz continuity of  $\nabla f$  to characterize nonlinear map (6.24b) by establishing a quadratic inequality that  $u(x) = \mathbf{prox}_{\mu g}(x - \mu \nabla f(x))$  satisfies.

**Lemma 7.** *Let Assumption 1 hold. Then, for any  $x \in \mathbb{R}^n$ ,  $\hat{x} \in \mathbb{R}^n$ ,  $u := \mathbf{prox}_{\mu g}(x - \mu \nabla f(x))$ , and  $\hat{u} := \mathbf{prox}_{\mu g}(\hat{x} - \mu \nabla f(\hat{x}))$ , the pointwise quadratic inequality*

$$\begin{bmatrix} x - \hat{x} \\ u - \hat{u} \end{bmatrix}^T \underbrace{\begin{bmatrix} \eta^2 I & 0 \\ 0 & -I \end{bmatrix}}_{\Pi} \begin{bmatrix} x - \hat{x} \\ u - \hat{u} \end{bmatrix} \geq 0\tag{6.25}$$

holds, where

$$\eta = \max\{|1 - \mu m_f|, |1 - \mu L_f|\}.\tag{6.26}$$

*Proof.* See Appendix A.1.  $\square$

**Lemma 8.** *The nonlinear function  $u(x) := \mathbf{prox}_{\mu g}(x - \mu \nabla f(x))$  is a contraction for  $\mu \in (0, 2/L_f)$ .*

*Proof.* From (A.8), it follows that the nonlinear function  $u(x)$  is a contraction for  $\eta < 1$ . From (6.26), we see that  $\eta < 1$  if and only if,

$$-1 < 1 - \mu L_f < 1 \quad \text{and} \quad -1 < 1 - \mu m_f < 1.$$

Since  $m_f \leq L_f$ , these conditions hold for  $\mu \in (0, 2/L_f)$ .  $\square$

We next employ the KYP lemma in the frequency domain [165]

$$\begin{bmatrix} H_\rho(j\omega) \\ I \end{bmatrix}^* \Pi \begin{bmatrix} H_\rho(j\omega) \\ I \end{bmatrix} \prec 0, \quad \forall \omega \in \mathbb{R} \quad (6.27)$$

where  $\Pi$  is given by (6.25),  $\omega$  is the temporal frequency, and

$$H_\rho(j\omega) := C(j\omega I - (A + \rho I))^{-1} B = \frac{1}{j\mu\omega + 1 - \mu\rho} I$$

to establish global exponential stability of (6.24). Note that stability of  $H_\rho$  requires  $\mu < 1/\rho$ .

**Theorem 9.** *Let Assumption 1 hold and let  $\mu \in (0, 2/L_f)$ . Then, proximal gradient flow dynamics (6.24) are globally exponentially stable with the rate  $\rho < (1 - \eta)/\mu$ , i.e.,*

$$\|x(t) - x^*\|_2 \leq c e^{-\rho t} \|x(0) - x^*\|_2, \quad \forall t \geq 0$$

where  $c$  is a positive constant and  $\eta$  is given by (6.26).

*Proof.* Substituting  $\Pi$  given by (6.25) into (6.27) implies that the condition (6.27) holds for all  $\omega \in \mathbb{R}$  if

$$I - \eta^2 H_\rho^*(j\omega) H_\rho(j\omega) \succ 0.$$

This inequality can be equivalently written as

$$(\mu^2 \omega^2 + (1 - \mu\rho)^2 - \eta^2) I \succ 0 \quad (6.28)$$

which holds for every  $\omega \in \mathbb{R}$  if

$$(1 - \mu\rho + \eta)(1 - \mu\rho - \eta) > 0.$$

Since stability of  $H_\rho$  requires  $1 - \mu\rho > 0$ , the first term is positive and (6.28) is satisfied for all  $\omega$  if  $1 - \mu\rho - \eta > 0$ . Finally, since  $\eta < 1$  for  $\mu \in (0, 2/L_f)$ , dynamics (6.24) are globally exponentially stable with the rate  $\rho < (1 - \eta)/\mu$ .  $\square$

**Remark 9.** For  $\mu = 2/(L_f + m_f)$ , the second term on the right-hand-side in (A.7) disappears and the parameter  $\eta$  is given by

$$\eta = \frac{L_f - m_f}{L_f + m_f} = \frac{\kappa - 1}{\kappa + 1}$$

where  $\kappa := L_f/m_f$  is the condition number associated with the function  $f$ . In this case, the convergence rate is upper bounded by the strong convexity module  $m_f$ , i.e.,  $\rho < m_f$ .

**Remark 10.** In Appendix B, by restricting our attention to in-network optimization, we provide a distributed implementation based on the proximal augmented Lagrangian. Furthermore, by introducing an appropriate change of coordinates, we utilize the theory of IQCs to prove global exponential stability under the strong convexity assumption.

**6.3.2 Proximal Polyak-Lojasiewicz condition** Next, we consider the problems in which the function  $f$  is not strongly convex but the function  $F := f + g$  satisfies the proximal PL condition (6.19).

**Assumption 2.** Let the regularization function  $g$  in (6.13) be proper, lower semicontinuous, and convex, let  $f$  be twice continuously differentiable with  $\nabla^2 f(x) \preceq L_f I$ , and let the generalized gradient map satisfy the proximal PL condition,

$$\|G_\mu(x)\|_2^2 \geq \gamma(F_\mu(x) - F_\mu^*)$$

where  $\mu \in (0, 1/L_f)$ ,  $\gamma > 0$ , and  $F_\mu^*$  is the optimal value of the FB envelope  $F_\mu$ .

**Remark 11.** We recall that the proximal gradient algorithm can be interpreted as a variable-metric gradient method on FB envelope and that (6.23) can be equivalently written as

$$\dot{x} = -(I - \mu\nabla^2 f(x))^{-1} \nabla F_\mu(x).$$

Under Assumption 2, the matrix  $I - \mu \nabla^2 f(x)$  is invertible and the functions  $F$  and  $F_\mu$  have the same minimizers and the same optimal values [157],

$$\operatorname{argmin}_x F(x) = \operatorname{argmin}_x F_\mu(x), \quad F^\star = F_\mu^\star.$$

This motivates study of the convergence properties of (6.23) in terms of the FB envelope.

**Theorem 10.** *Let Assumption 2 hold. Then the forward-backward envelope associated with proximal gradient flow dynamics (6.23) converge exponentially to  $F_\mu^\star = F^\star$  with the rate  $\rho = \gamma(1 - \mu L_f)$ , i.e.,*

$$F_\mu(x(t)) - F_\mu^\star \leq e^{-\rho t} (F_\mu(x(0)) - F_\mu^\star), \quad \forall t \geq 0.$$

*Proof.* We introduce a Lyapunov function candidate,

$$V(x) = F_\mu(x) - F_\mu^\star$$

where  $F_\mu$  is the FB envelope. The derivative of  $V$  along the solutions of (6.23) is given by

$$\begin{aligned} \dot{V}(x) &= \langle \nabla F_\mu(x), \dot{x} \rangle \\ &= - \langle \nabla F_\mu(x), (I - \mu \nabla^2 f(x))^{-1} \nabla F_\mu(x) \rangle \\ &= - \langle G_\mu(x), (I - \mu \nabla^2 f(x)) G_\mu(x) \rangle \end{aligned}$$

where the last expression follows from (6.6). Since the gradient of  $f$  is  $L_f$ -Lipschitz continuous, i.e.,  $\nabla^2 f(x) \preceq L_f I$  for all  $x \in \mathbb{R}^n$ , Assumption 2 implies

$$-(I - \mu \nabla^2 f(x)) \preceq -(1 - \mu L_f) I$$

and, thus,

$$\begin{aligned} \dot{V}(x) &\leq -(1 - \mu L_f) \|G_\mu(x)\|_2^2 \\ &\leq -\gamma(1 - \mu L_f) (F_\mu(x) - F_\mu^\star) \end{aligned} \tag{6.29}$$

is non-positive for  $\mu \in (0, 1/L_f)$ . Moreover, combining the last inequality with the definition of  $V$  yields,

$$\dot{V} \leq -\gamma(1 - \mu L_f) V$$

which implies

$$F_\mu(x(t)) - F_\mu^\star \leq e^{-\gamma(1-\mu L_f)t} (F_\mu(x(0)) - F_\mu^\star).$$

□

**Remark 12.** *When the proximal PL condition is satisfied,  $F_\mu(x(t)) - F_\mu^\star$  converges exponentially but, in the absence of strong convexity, the exponential convergence rate cannot be established for  $\|x(t) - x^\star\|_2$ . Thus, in the absence of strong convexity, although the objective function converges exponentially fast, the solution to (6.23) does not enjoy this convergence rate. To the best of our knowledge, the convergence rate of  $x(t)$  to the set of optimal values  $x^\star$  is not known in this case.*

## 6.4 Global exponential stability of Douglas-Rachford splitting dynamics

We next introduce a continuous-time gradient flow dynamics based on the well-known Douglas-Rachford splitting algorithm [162] and establish global exponential stability for strongly convex  $f$ .

**6.4.1 Non-smooth composite optimization problem** The optimality condition for non-smooth composite optimization problem (6.21) is given by

$$0 \in \nabla f(x) + \partial g(x).$$

Multiplication by  $\mu$  and addition/subtraction of  $x$  yields,

$$0 \in [I + \mu \nabla f](x) + \mu \partial g(x) - x.$$

Since the proximal operator associated with the function  $\mu f$  is determined by the resolvent operator of  $\mu \nabla f$ , we have

$$x = (I + \mu \nabla f)^{-1}(x - \mu \partial g(x)) = \mathbf{prox}_{\mu f}(x - \mu \partial g(x)).$$

Introducing a new variable  $z := x - \mu \partial g(x)$  allows us to bring the optimality condition into the following form

$$x = \mathbf{prox}_{\mu f}(z)$$

or, equivalently,

$$\mu \partial g(x) = \mathbf{prox}_{\mu f}(z) - z.$$

Now, adding  $x$  to both sides of this equation yields

$$\begin{aligned} [I + \mu \partial g](x) &= x + \mathbf{prox}_{\mu f}(z) - z \\ &= 2\mathbf{prox}_{\mu f}(z) - z \end{aligned}$$

which leads to,

$$x^* = \mathbf{prox}_{\mu g}(2\mathbf{prox}_{\mu f}(z^*) - z^*) = \mathbf{prox}_{\mu f}(z^*). \quad (6.30a)$$

Furthermore, the reflected proximal operators [166] of the functions  $f$  and  $g$ ,

$$R_{\mu f}(z) := [2\mathbf{prox}_{\mu f} - I](z), \quad R_{\mu g} := [2\mathbf{prox}_{\mu g} - I](z)$$

can be used to write optimality condition (6.30a) as

$$[R_{\mu g}R_{\mu f}](z^*) = z^*. \quad (6.30b)$$

This follows from (6.30a) and

$$[R_{\mu g}R_{\mu f}](z) = z + 2(\mathbf{prox}_{\mu g}(2\mathbf{prox}_{\mu f}(z) - z) - \mathbf{prox}_{\mu f}(z)).$$

Building on the optimality conditions, the DR splitting algorithm consists of the following iterative steps,

$$\begin{aligned} x^k &= \mathbf{prox}_{\mu f}(z^k) \\ y^k &= \mathbf{prox}_{\mu g}(2x^k - z^k) \\ z^{k+1} &= z^k + 2\alpha(y^k - x^k). \end{aligned} \quad (6.31)$$

Under standard convexity assumptions [167], the DR splitting algorithm converges for  $\alpha \in (0, 1)$ . Combining all the steps in (6.31) yields the first-order recurrence,

$$\begin{aligned} z^{k+1} &= z^k + 2\alpha(\mathbf{prox}_{\mu g}(2x^k - z^k) - x^k) \\ &= [(1 - \alpha)I + \alpha R_{\mu g}R_{\mu f}](z^k) \end{aligned} \quad (6.32)$$

where  $z^k$  converges to the fixed point of the operator  $R_{\mu f}R_{\mu g}$  and  $x^k$  converges to the

optimal solution of (6.21).

Optimality conditions (6.30) can be used to obtain the continuous-time gradient flow dynamics to compute  $z^*$ ,

$$\begin{aligned}\dot{z} &= -z + [R_{\mu g} R_{\mu f}](z) \\ &= \mathbf{prox}_{\mu g}(2 \mathbf{prox}_{\mu f}(z) - z) - \mathbf{prox}_{\mu f}(z) \\ &= -G_{\mu}^{DR}(z)\end{aligned}\tag{6.33}$$

where  $G_{\mu}^{DR}(z)$  is given by (6.12). We note that the discrete-time system (6.32) results from the explicit forward Euler discretization of (6.33) with the stepsize  $\alpha$ .

**Remark 13.** *Using the definition of  $\nabla F_{\mu}^{DR}(x)$  in (6.11), the continuous-time system (6.33) can be written as*

$$\dot{z} = -\mu (2 \nabla \mathbf{prox}_{\mu f}(z) - I)^{-1} \nabla F_{\mu}^{DR}(z)$$

where the inverse is well-defined for  $\mu \in (0, 1/L_f)$ . Thus, the DR splitting algorithm can be interpreted as a variable-metric gradient method on the DR envelope  $F_{\mu}^{DR}$  [157].

The continuous-time dynamics (6.33) can be also seen as a feedback interconnection of an LTI system

$$\begin{aligned}\dot{z} &= A z + B u \\ \xi &= C z\end{aligned}\tag{6.34a}$$

with a nonlinear term,

$$u(z) := [R_{\mu g} R_{\mu f}](z).\tag{6.34b}$$

Herein, the matrices in LTI representation (6.34a) are given by

$$A = -I, B = I, C = I\tag{6.34c}$$

and the corresponding transfer function is

$$H(s) = C(sI - A)^{-1}B = \frac{1}{s + 1} I.\tag{6.34d}$$



We first characterize properties of nonlinearity  $u$  in (6.34b) and then, similar to the previous section, employ the KYP lemma to establish the conditions for global exponential stability of (6.34).

**Lemma 11.** *Let Assumption 1 hold and let  $\mu \in (0, 2/L_f)$ . Then, the operator  $R_{\mu f}$  is  $\sigma$ -contractive,*

$$\|R_{\mu f}(x) - R_{\mu f}(y)\|_2 \leq \sigma \|x - y\|_2$$

where  $\sigma$  is given by

$$\sigma = \max\{|1 - \mu m_f|, |1 - \mu L_f|\} < 1. \quad (6.35)$$

*Proof.* Given  $z_x := \mathbf{prox}_{\mu f}(x)$  and  $z_y := \mathbf{prox}_{\mu f}(y)$ ,  $x$  and  $y$  can be computed as follows

$$x = z_x + \mu \nabla f(z_x), \quad y = z_y + \mu \nabla f(z_y).$$

Thus,

$$\begin{aligned} \|R_{\mu f}(x) - R_{\mu f}(y)\|^2 &= \|2(z_x - z_y) - (x - y)\|^2 \\ &= \|(z_x - z_y) - \mu(\nabla f(z_x) - \nabla f(z_y))\|^2 \\ &= \|z_x - z_y\|^2 + \|\mu(\nabla f(z_x) - \nabla f(z_y))\|^2 - \\ &\quad 2\mu \langle \nabla f(z_x) - \nabla f(z_y), z_x - z_y \rangle \\ &\leq \max\{(1 - \mu L_f)^2, (1 - \mu m_f)^2\} \|z_x - z_y\|^2 \\ &\leq \sigma \|x - y\|^2. \end{aligned}$$

where the firm non-expansiveness of the proximal operator is used in the last step. Moreover, according to Lemma 8, for  $\mu \in (0, 2/L_f)$  we have  $\sigma < 1$ , which completes the proof.  $\square$

**Lemma 12.** *Let Assumption 1 hold and let  $\mu \in (0, 2/L_f)$ . Then, the operator  $R_{\mu g}$  is firmly non-expansive.*

*Proof.*

$$\begin{aligned}
\|R_{\mu g}(x) - R_{\mu g}(y)\|_2^2 &= 4 \|\mathbf{prox}_{\mu f}(x) - \mathbf{prox}_{\mu f}(y)\|_2^2 + \|x - y\|_2^2 - \\
&\quad 4 \langle x - y, \mathbf{prox}_{\mu f}(x) - \mathbf{prox}_{\mu f}(y) \rangle \\
&\leq \|x - y\|_2^2.
\end{aligned}$$

□

**Remark 14.** Since  $R_{\mu g}$  is firmly non-expansive and  $R_{\mu f}$  is  $\sigma$ -contractive, the composite operator  $R_{\mu g}R_{\mu f}$  is also  $\sigma$ -contractive. Moreover, since the operator  $R_{\mu f}$  and nonlinearity  $u$  in (6.24b) have the same contraction parameters, the quadratic inequality that describes nonlinearity (6.24b) can be also utilized to characterize the composite operator  $R_{\mu g}R_{\mu f}$ .

**Theorem 13.** Let Assumption 1 hold. Then, the DR splitting dynamics (6.34) are globally exponentially stable, i.e., there is  $c > 0$  and  $\rho \in (0, 1 - \sigma)$  such that,

$$\|z(t) - z^*\| \leq c e^{-\rho t} \|z(0) - z^*\|, \quad \forall t \geq 0.$$

*Proof.* Note that although the nonlinearities in systems (6.20) and (6.34) are different, they have the same quadratic characterizations. Moreover, the dynamics are the same except for a multiplicative coefficient  $1/\mu$ . Thus, the KYP lemma implies the global exponential stability of (6.34) if there exists  $\rho \in (0, 1)$  such that,

$$\sigma^2 H_\rho(j\omega)^* H_\rho(j\omega) - I \prec 0, \quad \forall \omega \in \mathbb{R}, \quad (6.36)$$

where

$$H_\rho(j\omega) = C(j\omega I - (A + \rho I))^{-1} B = \frac{1}{j\omega + 1 - \rho} I.$$

The inequality (6.36) is satisfied if,

$$\sigma^2 - (1 - \rho)^2 - \omega^2 < 0, \quad \forall \omega \in \mathbb{R}$$

which proves  $\rho < 1 - \sigma$ .

□

**6.4.2 Douglas-Rachford splitting on the dual problem** The DR splitting algorithm cannot be used to directly solve a problem with a more general linear equality constraint,

$$\begin{aligned} & \underset{x, z}{\text{minimize}} && f(x) + g(z) \\ & \text{subject to} && Tx + Sz = r \end{aligned} \tag{6.37}$$

where  $T \in \mathbb{R}^{m \times n}$ ,  $S \in \mathbb{R}^{m \times n}$ , and  $r \in \mathbb{R}^m$  are problem data. However, it can be utilized to solve the dual problem,

$$\underset{\zeta}{\text{minimize}} \quad f_1(\zeta) + g_1(\zeta) \tag{6.38}$$

where

$$\begin{aligned} f_1(\zeta) &= f^*(-T^T \zeta) + r^T \zeta \\ g_1(\zeta) &= g^*(-S^T \zeta) \end{aligned}$$

and  $h^*(\zeta) := \sup_x (\zeta^T x - h(x))$  is the conjugate of the function  $h$ . It is a standard fact [167, 168] that solving the dual problem (6.38) via the DR splitting algorithm is equivalent to using ADMM for the original problem (6.37). Next, we introduce a gradient flow dynamics based on the DR splitting algorithm for solving (6.38) and demonstrate global exponentially stability under the following assumption.

**Assumption 3.** *Let the differentiable part  $f$  of the objective function in (6.37) be strongly convex with parameter  $m_f$ , let  $\nabla f$  be Lipschitz continuous with parameter  $L_f$ , let the function  $g$  be proper, lower semicontinuous, and convex, and let the matrix  $T$  be full row rank.*

The continuous-time DR splitting algorithm

$$\dot{\zeta} = \mathbf{prox}_{\mu g_1}(2 \mathbf{prox}_{\mu f_1}(\zeta) - \zeta) - \mathbf{prox}_{\mu f_1}(\zeta) \tag{6.39}$$

can be used to compute the optimal solution  $\zeta^*$  to (6.38). It is well-known that the conjugate functions are convex [126]. Since (6.39) is identical to (6.33), if  $f_1$  satisfies the conditions in Assumption 1 global exponential stability of (6.39) follows from Theorem 13. For a full row rank  $T$ ,  $f_1$  is strongly convex and  $\nabla f_1$  is Lipschitz continuous

with parameters [166, Proposition 4],

$$L_{f_1} = \|T^T\|^2/m_f, \quad m_{f_1} = \theta^2/L_f$$

where  $\theta$  is a positive parameter that always exists and satisfies  $\|T^T\nu\| \geq \theta\|\nu\|$  for all  $\nu \in \mathbb{R}^m$ . Thus, both  $f_1$  and  $g_1$  satisfy Assumption 3 and global exponential stability of (6.39) follows from Theorem 13.

**6.5 Concluding remarks** We study a class of nonsmooth optimization problems in which it is desired to minimize the sum of a continuously differentiable function with a Lipschitz continuous gradient and a nondifferentiable function. For strongly convex problems, we employ the theory of integral quadratic constraints to prove global exponential stability of proximal gradient flow and Douglas-Rachford splitting dynamics. We also propose a generalization of the Polyak-Lojasiewicz condition to nonsmooth problems and demonstrate the global exponential convergence of the forward-backward envelope for the proximal gradient flow algorithm even in the absence of strong convexity.

## Chapter 7

# Distributed proximal augmented Lagrangian method for nonsmooth composite optimization

We study a class of nonsmooth composite optimization problems in which the convex objective function is given by a sum of differentiable and nondifferentiable terms. By introducing auxiliary variables in nondifferentiable terms, we provide an equivalent consensus-based characterization that is convenient for distributed implementation. The Moreau envelope associated with the nonsmooth part of the objective function is used to bring the optimization problem into a continuously differentiable form that serves as a basis for the development of a primal-descent dual-ascent gradient flow method. This algorithm exploits separability of the objective function and is well-suited for in-network optimization. We prove global asymptotic stability of the proposed algorithm and solve the problem of growing undirected consensus networks in a distributed manner to demonstrate its effectiveness.

**7.1 Introduction** We study a class of nonsmooth composite optimization problems in which the convex objective function is a sum of differentiable and nondifferentiable functions. Among other applications, these problems emerge in machine

learning, compressive sensing, and control. Recently, regularization has been used as a promising tool for enhancing utility of standard optimal control techniques. In this approach, commonly used performance measures are augmented with regularization functions that are supposed to promote some desired structural features in the distributed controller, e.g., sparsity. Such an approach has received significant attention in recent years [41, 42, 71, 106, 115, 118, 119], but computing optimal solutions in large-scale problems still remains a challenge.

Generic descent methods cannot be used in the nonsmooth composite optimization problems due to the presence of a nondifferentiable component in the objective function. Moreover, these standard methods are not well-suited for distributed implementation. An alternative approach is to separate the smooth and nonsmooth parts of the objective function and use the alternating direction method of multipliers (ADMM). In [111], we exploit separability of the objective function and utilize an ADMM-based consensus algorithm to solve the regularized optimal control problem in a distributed manner over multiple processors. Even though the optimal control problem is in general non-convex, recent results can be utilized to show convergence to a local minimum [112]. However, in an update step of the ADMM algorithm, all the processors halt to compute the weighted average (the gathering step) [113].

Herein, we build on recent work [108] in which the structure of proximal operators associated with nonsmooth regularizers was exploited to bring the augmented Lagrangian into a continuously differentiable form. Such an approach is suitable for developing an algorithm based on primal-descent dual-ascent gradient method. We use the Arrow-Hurwicz-Uzawa gradient flow dynamics [77] and propose an algorithm that can be implemented in a fully distributed manner over multiple processors. This increases the computational efficiency and reduces the overall computation time. By exploiting convexity of the smooth part of the objective function, we show asymptotic convergence of our algorithm.

The point of departure of our work from [108] is that we study a more general form of consensus optimization problems in which the optimization variable is a matrix

and develop a fully distributed algorithm. Furthermore, while most existing primal-dual techniques for nonsmooth distributed optimization employ subgradient flow methods [73, 79, 169], our approach yields a gradient flow dynamics with a continuous right-hand side even for nonsmooth problems.

The rest of the chapter is structured as follows. In Section 7.2, we formulate the nonsmooth composite optimization problem, discuss a motivating example, and provide background on proximal operators and the consensus-based ADMM algorithm. In Section 7.3, by exploiting the structure of proximal operators, we introduce the proximal augmented Lagrangian. In Section 7.4, we use the Arrow-Hurwicz-Uzawa method to develop the gradient flow dynamics that are well-suited for distributed computations and prove global asymptotic stability. In Section 7.5, we discuss distributed implementation, in Section 7.6, we provide examples, and, in Section 7.7, we offer concluding remarks.

**7.2 Problem formulation** We consider a composite convex optimization problem,

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^n f_i(x) + g(x) \quad (7.1)$$

where  $x \in \mathbb{R}^m$  is the optimization variable, the functions  $f_i$  are continuously differentiable, and the function  $g$  is possibly nondifferentiable. This problem can be brought into a standard consensus form by introducing  $n$  local variables  $x_i$  and a global variable  $z$ ,

$$\underset{x_i, z}{\text{minimize}} \quad \sum_{i=1}^n f_i(x_i) + g(z) \quad (7.2)$$

$$\text{subject to} \quad x_i - z = 0, \quad i = 1, \dots, n.$$

Even though this reformulation increases the number of optimization variables, it facilitates distributed computations by bringing the objective function into a separable form. Clearly, the solutions to (7.1) and (7.2) coincide but, in contrast to (7.1), optimization problem (7.2) is convenient for distributed implementation. Solving (7.2) on a single processor is not necessarily more computationally efficient than solving the original problem via a centralized algorithm. However, optimization problem (7.2) can be split into  $n$  separate subproblems over  $n$  different processors. In such a setup, each

processor solves an optimization problem that involves a local objective function  $f_i$  of a single local variable  $x_i$ . This is advantageous for large-scale systems where centralized implementation is prohibitively complex and cannot be afforded.

**7.2.1 Motivating application** Problem (7.1) arises in feedback design when a performance metric, e.g., the  $\mathcal{H}_2$  norm, is augmented with a regularization function to promote structural features in the optimal controller. Herein, we discuss the problem of growing undirected consensus networks and show that the objective function is separable; thereby, it completely fits into the framework (7.2).

We consider the controlled undirected network,

$$\begin{aligned}\dot{\psi} &= -L_p \psi + d + u \\ \zeta &= \begin{bmatrix} Q^{1/2} \\ 0 \end{bmatrix} \psi + \begin{bmatrix} 0 \\ R^{1/2} \end{bmatrix} u\end{aligned}$$

where  $d$  and  $u$  are the disturbance and control inputs,  $\psi$  is the state, and  $\zeta$  is the performance output. The dynamic matrix  $L_p$  is the Laplacian matrix of the plant network and symmetric matrices  $Q \succeq 0$  and  $R \succ 0$  specify the state and control weights in the performance output. For memoryless control laws,

$$u = -L_x \psi$$

where  $L_x$  is the Laplacian matrix of the controller graph, the closed-loop system is given by

$$\begin{aligned}\dot{\psi} &= -(L_p + L_x) \psi + d \\ \zeta &= \begin{bmatrix} Q^{1/2} \\ -R^{1/2} L_x \end{bmatrix} \psi.\end{aligned}\tag{7.3}$$

In the absence of exogenous disturbances, the network converges to the average of the initial node values  $\bar{\psi} = (1/n) \sum_i \psi_i(0)$  if and only if it is connected [2]. Let  $Q := I - (1/n) \mathbf{1}\mathbf{1}^T$  penalizes the deviation of individual node values from average. The objective is to minimize the mean square deviation from the network average by adding a few additional edges, specified by the graph Laplacian  $L_x$  of a controller network. If



$E$  is the incidence matrix of the controller graph,  $L_x$  can be written as

$$L_x = E \operatorname{diag}(x) E^T$$

where  $\operatorname{diag}(x)$  is a diagonal matrix containing the optimization variable  $x \in \mathbb{R}^m$  (i.e., the vector of the edge weights in the controller graph). Regularization terms may be used to promote sparsity of the controller network or to impose some additional constraints on the edge weights. The matrix  $L_x$  that optimizes the closed-loop performance and has certain structural properties can be obtained by solving the regularized optimal control problem

$$\underset{x}{\text{minimize}} \quad f(x) + g(x). \quad (7.4)$$

Here,  $f$  is the function that quantifies the closed-loop performance, i.e. the  $\mathcal{H}_2$  norm, and  $g$  is the regularization function that is introduced to promote certain structural properties of  $L_x$ . For example, when it is desired to design  $L_x$  with a specified pattern of zero elements,  $g$  is an indicator function of the set that characterizes this pattern [114]. When it is desired to promote sparsity of  $L_x$ , the  $\ell_1$  norm  $g(x) = \gamma \sum_i |x_i|$  can be used as a sparsity-enhancing regularizer, where  $\gamma$  is the positive parameter that characterizes emphasis on sparsity [42].

Next, we exploit the square-additive property of the  $\mathcal{H}_2$  norm to provide an equivalent representation that is convenient for large-scale and distributed optimization. As shown in [106], up to an additive constant, the square of the  $\mathcal{H}_2$  norm (from  $d$  to  $\zeta$ ) is determined by

$$f(x) = \operatorname{trace} \left( (E \operatorname{diag}(x) E^T + L_p)^\dagger (I + L_p R L_p) \right) + \operatorname{diag} (E^T R E)^T x$$

where the pseudo-inverse of the closed-loop graph Laplacian is given by

$$(E \operatorname{diag}(x) E^T + L_p)^\dagger = (E \operatorname{diag}(x) E^T + (1/n) \mathbf{1} \mathbf{1}^T + L_p)^{-1}.$$

It is easy to show that  $f(x)$  can be written as

$$f(x) = \sum_{i=1}^n f_i(x)$$

where

$$f_i(x) = \xi_i^T (E \operatorname{diag}(x) E^T + (1/n) \mathbf{1}\mathbf{1}^T + L_p)^{-1} \xi_i + (1/n) \operatorname{diag}(E^T R E)^T x. \quad (7.5)$$

Here,  $\xi_i = (I + L_p R L_p)^{1/2} e_i$  is the  $i$ th column of the square root of the matrix  $(I + L_p R L_p)$ . Moreover, it can be shown that the gradient of  $f_i(x)$  is given by

$$\nabla f_i(x) = (1/n) \operatorname{diag}(E^T R E) - \nu_i(x) \circ \nu_i(x) \quad (7.6)$$

where  $\circ$  is the elementwise multiplication and

$$\nu_i(x) = E^T (E \operatorname{diag}(x) E^T + (1/n) \mathbf{1}\mathbf{1}^T + L_p)^{-1} \xi_i. \quad (7.7)$$

In what follows, we provide essential background on the proximal operators that we utilize for the latter developments.

## 7.2.2 Background

### Proximal operators

The proximal operator of the function  $g$  is given by

$$\mathbf{prox}_{\mu g}(v) := \operatorname{argmin}_z g(z) + \frac{1}{2\mu} \|z - v\|^2,$$

and the Moreau envelope determines the corresponding value function,

$$M_{\mu g}(v) := g(\mathbf{prox}_{\mu g}(v)) + \frac{1}{2\mu} \|\mathbf{prox}_{\mu g}(v) - v\|_2^2. \quad (7.8)$$

Irrespective of differentiability of  $g$ , Moreau envelope is a continuously differentiable function and its gradient is given by [76],

$$\nabla M_{\mu g}(v) = \frac{1}{\mu} (v - \mathbf{prox}_{\mu g}(v)). \quad (7.9)$$

The above defined functions play essential role in our subsequent developments.

### Alternating Direction Method of Multipliers (ADMM)

We next demonstrate that a standard consensus algorithm based on the ADMM [113] can be used to solve the problem (7.2). This algorithm is well-suited for distributed implementation in which each processor solves an optimization problem. More details can be found in [111].

The augmented Lagrangian associated with (7.2) is given by

$$\mathcal{L}_{\mu_i}(x_i, z; \lambda) := g(z) + \sum_{i=1}^n (f_i(x_i) + \langle \lambda_i, x_i - z \rangle + \frac{1}{2\mu_i} \|x_i - z\|_2^2) \quad (7.10)$$

where  $\lambda_i$ 's are the Lagrange multipliers and  $\mu_i$ 's are positive parameters. The distributed ADMM algorithm consists of the following iterative steps,

$$\begin{aligned} x_i^{k+1} &= \underset{x_i}{\operatorname{argmin}} f_i(x_i) + \frac{1}{2\mu_i} \|x_i - u_i^k\|_2^2 \\ z^{k+1} &= \underset{z}{\operatorname{argmin}} g(z) + \sum_{i=1}^n \frac{1}{2\mu_i} \|z - v_i^k\|_2^2 \\ \lambda_i^{k+1} &= \lambda_i^k + \frac{1}{\mu_i} (x_i^{k+1} - z^{k+1}) \end{aligned}$$

where

$$\begin{aligned} u_i^k &:= z^k - \mu_i \lambda_i^k \\ v_i^k &:= x_i^{k+1} + \mu_i \lambda_i^k. \end{aligned}$$

The  $x_i$ -minimization step can be done via distributed computation by spreading subproblems to  $n$  different processors. On the other hand, the update of  $z$  amounts to the evaluation of the proximal operator of the function  $g$ ,

$$z^{k+1} = \mathbf{prox}_{\hat{\mu}g} \left( \hat{\mu} \sum_{i=1}^n \frac{1}{\mu_i} v_i^k \right).$$

where  $\hat{\mu} := (\sum_i 1/\mu_i)^{-1}$ . Thus, the update of  $z$  requires gathering each  $x_i^{k+1}$  and the associated Lagrange multipliers  $\lambda_i^k$  in order to form  $v_i^k$ .

The above presented consensus algorithm is standard (e.g., see [113]). We have previously used this algorithm for distributed design of structured feedback gains in [111].

Recently, convergence of this algorithm was established even for problems with non-convex objective functions  $f_i$  [112].

**7.3 Proximal augmented Lagrangian** The interconnection graph between the nodes in the consensus-based formulation in (7.2) is given by a star graph. Each node has access to the internal node of the star graph with the state  $z$ . This topology yields the  $z$ -update in ADMM that requires gathering the states of all subsystems. Recently, an algorithm based on the proximal augmented Lagrangian for solving (7.2) was developed in [108]. To avoid the above described computational requirement in the  $z$ -update of the ADMM algorithm, we propose a primal-dual algorithm based on the proximal augmented Lagrangian that can be implemented in a fully distributed manner.

Problem (7.2) can be equivalently written as,

$$\begin{aligned} & \underset{x_i, z}{\text{minimize}} && \sum_{i=1}^n f_i(x_i) + g(z) \\ & \text{subject to} && x_i - x_j = 0, \quad (i, j) \in \mathcal{I} \\ & && x_k - z = 0, \end{aligned} \tag{7.11}$$

where  $\mathcal{I}$  is the set of indices between 1 and  $n$  such that any index appears in one pair of the set at least once. This set characterizes structure of the information exchange network between the agents. The interaction topology is given by a connected graph. Moreover, the index  $k \in \{1, \dots, n\}$  can be chosen arbitrarily. In what follows, we study one particular instance of problem (7.11).

Without loss of generality, we assume that  $k = n$  and that the underlying communication network between different nodes in (7.11) is given by a path graph. By introducing the optimization variable  $X$ ,

$$X := \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{m \times n}$$

the column vector  $e_n$ ,

$$e_n = \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}^T \in \mathbb{R}^n$$

and the matrix  $T$ ,

$$T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{bmatrix} \in \mathbb{R}^{n \times (n-1)}$$

we can rewrite (7.11) as,

$$\begin{aligned} & \underset{X, z}{\text{minimize}} && f(X) + g(z) \\ & \text{subject to} && XT = 0, \\ & && X e_n - z = 0. \end{aligned} \tag{7.12}$$

In (7.12), the matrix  $T$  is the incidence matrix of an undirected path network that connects  $n$  nodes. We note that any connected network can be used to build the information exchange structure between the nodes. For an arbitrary connected graph with  $m$  edges, the matrix  $T \in \mathbb{R}^{n \times m}$  has to satisfy the following properties,

$$T^T \mathbf{1} = 0, \quad TT^T = L,$$

where  $\mathbf{1}$  is the vector of all ones and  $L \in \mathbb{R}^{n \times n}$  is the Laplacian matrix of the underlying graph. This is exactly the problem (6.37) that we discussed in the previous Chapter 6. In Sec. 6.4.2, we have introduced a gradient flow dynamics based on the DR splitting algorithm that can be used to solve the dual form of the problem (7.12) with a guaranteed global exponential convergence under certain assumptions 3. Herein, we provide an algorithm based on proximal augmented Lagrangian to solve the primal problem that is well-suited for distributed implementation and we show global asymptotic convergence.

The augmented Lagrangian associated with (7.12) is given by

$$\begin{aligned} \mathcal{L}_{\mu_i}(X, z; \lambda, Y) = & f(X) + g(z) + \langle \lambda, X e_n - z \rangle + \langle Y, XT \rangle + \frac{1}{2\mu_1} \|X e_n - z\|_2^2 + \\ & \frac{1}{2\mu_2} \|XT\|_F^2 \end{aligned} \tag{7.13}$$

where  $\lambda \in \mathbb{R}^m$  and  $Y \in \mathbb{R}^{m \times (n-1)}$  are the Lagrange multipliers and  $\mu_1$  and  $\mu_2$  are positive parameters. The proximal augmented Lagrangian is obtained by evaluating the

augmented Lagrangian on the manifold that results from the explicit minimization of  $\mathcal{L}_{\mu_i}$  with respect to  $z$  [108]. This yields a function that is once but not twice continuously differentiable with respect to both the primal variable  $X$  and the dual variables  $\lambda$  and  $Y$ .

The proximal augmented Lagrangian associated with (7.12) is given by

$$\begin{aligned} \mathcal{L}_{\mu_i}(X; \lambda, Y) &= \mathcal{L}_{\mu_i}(X, z_{\mu_1}^*(X; \lambda); \lambda, Y) = \\ f(X) + M_{\mu_1 g}(X \mathbf{e}_n + \mu_1 \lambda) - \frac{\mu_1}{2} \|\lambda\|_2^2 + \langle Y, X T \rangle + \frac{1}{2\mu_2} \|X T\|_F^2, \end{aligned} \quad (7.14)$$

where  $M_{\mu_1 g}$  is the Moreau envelope of the function  $g(z)$  and  $z_{\mu_1}^*(X; \lambda)$  is given by

$$z_{\mu_1}^*(X; \lambda) = \mathbf{prox}_{\mu_1 g}(X \mathbf{e}_n + \mu_1 \lambda)$$

and by substituting  $z^*$  in the augmented Lagrangian (7.13), the proximal augmented Lagrangian can be written as (7.14).

**7.4 Arrow-Hurwicz-Uzawa gradient flow** The proximal augmented Lagrangian is a continuously differentiable function because the Moreau envelope is continuously differentiable. This facilitates the use of the Arrow-Hurwicz-Uzawa algorithm which is a primal-descent dual-ascent gradient flow method. In this algorithm, the primal variable  $X$  and the dual variables  $\lambda$  and  $Y$  are updated simultaneously. The gradient flow dynamics are given by

$$\begin{aligned} \dot{X} &= -\nabla_X \mathcal{L}_{\mu}(X; \lambda, Y) \\ \dot{\lambda} &= +\nabla_{\lambda} \mathcal{L}_{\mu}(X; \lambda, Y) \\ \dot{Y} &= +\nabla_Y \mathcal{L}_{\mu}(X; \lambda, Y). \end{aligned}$$

By taking the derivatives, the updates can be written as

$$\begin{aligned} \dot{X} &= -(\nabla f(X) + \nabla M_{\mu_1 g}(X \mathbf{e}_n + \mu_1 \lambda) \mathbf{e}_n^T + \frac{1}{\mu_2} X T T^T + Y T^T) \\ \dot{\lambda} &= \mu_1 \nabla M_{\mu_1 g}(X \mathbf{e}_n + \mu_1 \lambda) - \mu_1 \lambda \\ \dot{Y} &= X T. \end{aligned} \quad (7.15)$$

It is worth to note that if  $f(x)$  is separable, i.e.  $f(x) = \sum_i f_i(x_i)$ , the gradient  $\nabla f(X)$  is an  $m \times n$  matrix and can be written as

$$\nabla f(X) = \begin{bmatrix} \nabla f_1(x_1) & \dots & \nabla f_n(x_n) \end{bmatrix}.$$

**Asymptotic convergence** Our subsequent developments are based on the following assumption

**Assumption 4.** *The function  $f$  is continuously differentiable and convex, and the function  $g$  is proper, lower semicontinuous, and convex.*

We show that under Assumption 4, dynamics (7.15) are globally asymptotically stable and converge to  $(X^*, \lambda^*, Y^*)$  where each of the columns of  $X^*$  is the optimal solution to (7.1). The optimal primal and dual points  $(X^*, \lambda^*, Y^*)$  satisfy the following first order optimality conditions

$$\nabla f(X^*) + \lambda^* e_n^T + Y^* T^T = 0 \quad (7.16a)$$

$$X^* e_n - z^* = 0 \quad (7.16b)$$

$$\partial g(z^*) - \lambda^* \ni 0 \quad (7.16c)$$

$$X^* T = 0 \quad (7.16d)$$

where  $\partial g$  is the subgradient of  $g$ .

**Proposition 14.** *Let Assumption 4 hold. Then, gradient flow dynamics (7.15) are globally asymptotically stable, i.e., they converge globally to the optimal primal and dual points  $(X^*, \lambda^*, Y^*)$  of (7.12).*

*Proof.* We introduce a change of variables

$$\tilde{X} := X - X^*, \quad \tilde{\lambda} := \lambda - \lambda^*, \quad \tilde{Y} := Y - Y^*$$

and a Lyapunov function

$$V(\tilde{X}, \tilde{\lambda}, \tilde{Y}) = \frac{1}{2} \langle \tilde{X}, \tilde{X} \rangle + \frac{1}{2} \langle \tilde{\lambda}, \tilde{\lambda} \rangle + \frac{1}{2} \langle \tilde{Y}, \tilde{Y} \rangle$$

where  $(\tilde{X}, \tilde{\lambda}, \tilde{Y})$  satisfy

$$\begin{aligned}\dot{\tilde{X}} &= \nabla f(X^*) - \nabla f(X) - \frac{1}{\mu_1} \tilde{m} \mathbf{e}_n^T - \frac{1}{\mu_2} \tilde{X} T T^T - \tilde{Y} T^T, \\ \dot{\tilde{\lambda}} &= -\mu_1 \tilde{\lambda} + \tilde{m}, \quad \dot{\tilde{Y}} = \tilde{X} T,\end{aligned}\tag{7.17}$$

with

$$\tilde{m} := \mu_1 (\nabla M_{\mu_1 g}(X \mathbf{e}_n + \mu_1 \lambda) - \nabla M_{\mu_1 g}(X^* \mathbf{e}_n + \mu_1 \lambda^*)).$$

Based on [108, Lemma 2], we can write

$$P(\tilde{X} \mathbf{e}_n + \mu_1 \tilde{\lambda}) = \mathbf{prox}_{\mu_1 g}(X \mathbf{e}_n + \mu_1 \lambda) - \mathbf{prox}_{\mu_1 g}(X^* \mathbf{e}_n + \mu_1 \lambda^*),\tag{7.18}$$

where  $I$  is the identity matrix and  $P$  is a positive semidefinite matrix such that  $P \preceq I$ . Thus, from (7.9) we have,

$$\tilde{m} = (I - P)(\tilde{X} \mathbf{e}_n + \mu_1 \tilde{\lambda}).\tag{7.19}$$

The derivative of the Lyapunov function candidate along the solutions of (7.17) is determined by

$$\begin{aligned}\dot{V} &= \langle \tilde{X}, \dot{\tilde{X}} \rangle + \langle \tilde{\lambda}, \dot{\tilde{\lambda}} \rangle + \langle \tilde{Y}, \dot{\tilde{Y}} \rangle = -\langle \tilde{X}, \nabla f(X) - \nabla f(X^*) \rangle - \mu_1 \langle \tilde{\lambda}, P \tilde{\lambda} \rangle - \\ &\quad \frac{1}{\mu_1} \langle (I - P) \tilde{X} \mathbf{e}_n, \tilde{X} \mathbf{e}_n \rangle - \frac{1}{\mu_2} \langle \tilde{X} T T^T, \tilde{X} \rangle.\end{aligned}$$

Since  $f$  is convex, the first term is nonpositive. Moreover, by utilizing  $0 \preceq P \preceq I$ , it follows that  $\dot{V} \leq 0$ . We next invoke LaSalle's invariance principle [170] to establish global asymptotic stability.

The points  $(\tilde{X}, \tilde{\lambda}, \tilde{Y})$  in the set

$$S = \{(\tilde{X}, \tilde{\lambda}, \tilde{Y}) \mid \dot{V}(\tilde{X}, \tilde{\lambda}, \tilde{Y}) = 0\},$$



satisfy

$$\nabla f(X^* + \tilde{X}) = \nabla f(X^*), \quad (7.20a)$$

$$\tilde{\lambda} \in \ker(P), \quad (7.20b)$$

$$\tilde{X} e_n \in \ker(I - P), \quad (7.20c)$$

$$\tilde{X}^T \in \text{span}\{[\mathbf{1} \cdots \mathbf{1}]\}, \quad (7.20d)$$

where  $\ker(A)$  denotes the null space of the matrix  $A$  and  $A \in \text{span}\{[\mathbf{1} \cdots \mathbf{1}]\}$  signifies that each column of the matrix  $A$  is given by a scalar multiple of the vector of all ones,  $\mathbf{1}$ . From (7.19), for the points in this set, we have  $\tilde{m} = \mu_1 \tilde{\lambda}$ . In order to identify the largest invariant set in  $S$ , we evaluate dynamics (7.17) under constraints (7.20) to obtain

$$\dot{\tilde{X}} = -\tilde{\lambda} e_n^T - \tilde{Y} T^T, \quad \dot{\tilde{\lambda}} = 0, \quad \dot{\tilde{Y}} = 0. \quad (7.21)$$

Thus, the invariant set is characterized by  $\tilde{\lambda} e_n^T + \tilde{Y} T^T = 0$  for constant  $\tilde{\lambda}$  and  $\tilde{Y}$ . To complete the proof, we need to show that the largest invariant set in  $S$  yields

$$(X, \lambda, Y) = (X^*, \lambda^*, Y^*) + (\tilde{X}, \tilde{\lambda}, \tilde{Y}),$$

that satisfy optimality conditions (7.16).

Points  $(X, \lambda, Y)$  satisfy optimality condition (7.16a) if

$$\nabla f(X) + (\tilde{\lambda} + \lambda^*) e_n^T + (\tilde{Y} + Y^*) T^T = 0.$$

For any  $(\tilde{X}, \tilde{\lambda}, \tilde{Y})$  in the invariant set, we can use (7.20a) to replace  $\nabla f(X)$  with  $\nabla f(X^*)$ . Furthermore, since  $\tilde{\lambda} e_n^T + \tilde{Y} T^T = 0$ , the resulting  $(X, \lambda, Y)$  satisfy (7.16a). Moreover, the substitution of  $P\tilde{\lambda} = 0$  and  $(I - P)\tilde{X} e_n = 0$  to (7.18) yields

$$X e_n - X^* e_n = \mathbf{prox}_{\mu_1 g}(X e_n + \mu_1 \lambda) - \mathbf{prox}_{\mu_1 g}(X^* e_n + \mu_1 \lambda^*).$$

The optimality condition (7.16b) leads to

$$X e_n = \mathbf{prox}_{\mu_1 g}(X e_n + \mu_1 \lambda) = z$$

which implies that the pair  $(X, z)$  satisfies (7.16b). We next show that the optimality condition (7.16c) holds for any  $(z, \lambda)$  in this set. Taking sub-differential of the proximal operator of the function  $g$  in (7.8) yields

$$\partial g(z) + \frac{1}{\mu_1}(z - v) \ni 0$$

where  $v$  is an arbitrary vector. Choosing  $v = Xe_n + \mu_1\lambda$  and utilizing the fact that  $Xe_n = z$  yields the third optimality condition (7.16c). Furthermore,  $\tilde{X}T = 0$  yields  $XT = 0$ . Thus,  $X$  satisfies (7.16d) and the dynamics (7.15) converges asymptotically to the optimal points  $(X^*, \lambda^*, Y^*)$ .  $\square$

**7.5 Distributed implementation** In this section, we exploit the structure of the problem and show that the gradient flow dynamics (7.15) is well-suited for distributed implementation. In this case, the underlying interconnection network is given by a path graph. We first discuss how the gradient flow of the primal variable  $X$  can be implemented in a distributed manner and then show that the dual variables can be also updated in a distributed fashion.

**7.5.1 Primal update** The vector  $x_k$  denotes the  $k$ th column of the matrix  $X$ . Each of the columns from  $k = 2, \dots, (n-1)$  can be updated in a distributed manner as follows

$$\dot{x}_k = -\nabla f_k(x_k) - \frac{1}{\mu_2}(2x_k - x_{k-1} - x_{k+1}) - y_k + y_{k-1}, \quad (7.22a)$$

where the vector  $y_k$  is the  $k$ th columns of the matrix  $Y$ . Thus, each agent only uses its neighbors' states and the corresponding dual variables to update its own state. The updates for the first and last column of  $X$  which are  $x_1$  and  $x_n$  are different than the other updates and can be written as follows

$$\dot{x}_1 = -\nabla f_1(x_1) - \frac{x_1 - x_2}{\mu_2} - y_1 \quad (7.22b)$$

$$\dot{x}_n = -\nabla f_n(x_n) - \nabla M_{\mu_1 g}(x_n + \mu_1 \lambda) - \frac{1}{\mu_2}(x_n - x_{n-1}) + y_{n-1}. \quad (7.22c)$$

Similarly, we can see only local information exchange and access to local dual variable is required for these two updates.

**7.5.2 Dual updates** The dual variable  $\lambda$  is a column vector and its update can be done by using the following column update

$$\dot{\lambda} = \mu_1 \nabla M_{\mu_1 g}(x_n + \mu_1 \lambda) - \mu_1 \lambda. \quad (7.22d)$$

Thus, we only need the state of the  $n$ th agent to update its value. The second dual variable  $Y$  is an  $m \times (n-1)$  matrix and the  $k$ th column of it,  $y_k$ , can be updated using only the states of  $x_k$  and  $x_{k+1}$  agents for  $k = 1, \dots, (n-1)$ ,

$$\dot{y}_k = x_k - x_{k+1}. \quad (7.22e)$$

**7.6 Computational experiments** In this section, we employ our algorithm for growing connected resistive Erdős-Rényi networks with edge probability  $1.05 \log(n)/n$  for different number of nodes using multiple cores. We choose the control weight matrix  $R = I$  and a state weight matrix that penalizes the mean-square deviation from the network average,  $Q = I - (1/n) \mathbf{1}\mathbf{1}^T$ . Moreover, the incidence matrix of the controller is such that there are no joint edges between the plant and the controller graphs. This algorithm is implemented in a distributed fashion by splitting the problem into  $N$  separate subproblems over  $N$  different cores. We have provided a parallel implementation in MATLAB and have executed tests on a machine featuring an Intel Core i7-3770 with 16GB of RAM to measure the performance of the algorithm.

The solve times are averaged over 10 trials and the speedup relative to a single core is displayed in Fig 7.1. It demonstrates that the algorithm is scalable. In particular, multi-core execution outperforms running just on a single core. Moreover, the speed-up is even higher for larger networks since overheads of parallel execution are less and more time is spent on actual parallel computation.

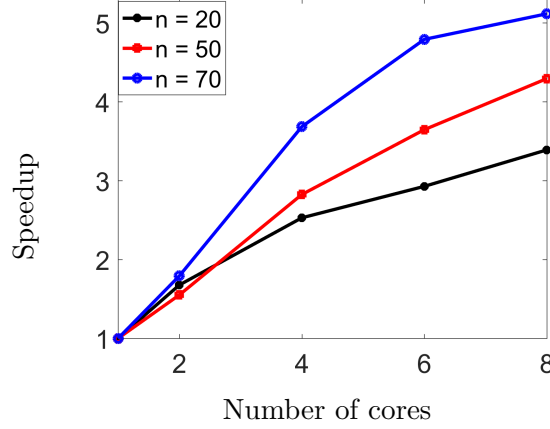


Figure 7.1: Speedup ratio versus the number of the cores used for growing connected resistive networks with  $n$  nodes.

**7.7 Concluding remarks** We have studied a class of convex nonsmooth composite optimization problems in which the objective function is a combination of differentiable and nondifferentiable functions. By exploiting the structure of the problem, we have provided an equivalent consensus-based characterization and have developed an algorithm based on primal-descent dual-ascent gradient flow method. This algorithm exploits the separability of the objective function and is well-suited for distributed implementation. Convexity of the smooth part of the objective function is utilized to prove global asymptotic stability of our algorithm. Finally, by exploiting the structure of the  $\mathcal{H}_2$  norm, we have employed this algorithm to design a sparse controller network that improves the performance of the closed-loop system in a large-scale undirected consensus network in a distributed manner. An example is provided to demonstrate the utility of the developed approach. We are currently working on implementing this algorithm in C++ and will use it to solve structured optimal control problems for large-scale systems in a distributed manner.

## Part IV

# Data-driven and model-free distributed control

## Chapter 8

# Data-driven proximal algorithms for the design of structured optimal feedback gains

Distributed feedback design and complexity constrained control are examples of problems posed within the domain of structured optimal feedback synthesis. The optimal feedback gain is typically a non-convex function of system primitives. However, in recent years, algorithms have been proposed to obtain locally optimal solutions. In applications to large-scale distributed control, the major obstacle is computational complexity. This chapter addresses complexity through a combination of linear-algebraic techniques and computational methods adapted from both machine learning and reinforcement learning. It is shown that for general classes of optimal control problems, the objective function and its gradient can be computed from data. Transformations borrowed from the theory of reinforcement learning are adapted to obtain simulation-based algorithms for computing the structured optimal  $H_2$  feedback gain. Customized algorithms based on proximal gradient descent, incremental proximal gradient, an ADMM are tested in computational experiments and their relative merits are discussed.

**8.1 Introduction** In this chapter, by exploiting the square-additive property of the  $\mathcal{H}_2$  norm, we provide efficient algorithms to solve the regularized  $\mathcal{H}_2$  optimal control problem. We demonstrate that the objective function and its gradients can

be evaluated without solving large-scale Lyapunov equations. In particular, our approach is inspired by the framework developed in [171, 172] where an iterative method for computing the structured linear quadratic regulator is proposed. Herein, we address complexity through a combination of linear-algebraic techniques and computational methods adapted from both machine learning and reinforcement learning. We approach the structured optimal control problem via a data-driven framework that does not require knowledge of the system parameters and avoids the need to solve large-scale matricial equations. For the structured optimal  $H_2$  state-feedback problem, we show that the objective function and its gradient can be computed from data and develop customized proximal algorithms based on gradient descent and incremental gradient method.

Moreover, we exploit separability of the objective function and utilize an ADMM-based consensus algorithm to solve the regularized optimal control problem in a distributed manner over multiple processors. Even though the optimal control problem is in general non-convex, recent results can be utilized to show convergence to a local minimum [112]. The ADMM-based consensus algorithm that we use is standard (e.g., see [113]) but, to the best of our knowledge, it has not been previously used for distributed design of structured feedback gains. Even though the optimal control problem is in general non-convex, recent results can be utilized to show convergence to a local minimum [112]. The ADMM-based consensus algorithm that we use is standard (e.g., see [113]) but, to the best of our knowledge, it has not been previously used for distributed design of structured feedback gains.

Our presentation is organized as follows. In Section 8.2, we describe the regularized structured optimal control problem. In Section 8.3, a square-additive property of the  $H_2$  norm is used to obtain a decomposition of the objective function and its gradient and duality arguments are utilized to transform these into quantities that can be estimated from experiments or simulations. In Section 8.4, we illustrate how structure of symmetric systems and undirected consensus networks can be exploited to simplify computations. In Section 8.5, proximal algorithms of tractable complexity for solving the structured optimal control problem are described and computational experiments are provided. In Section 8.6, we propose an ADMM-based consensus algorithm to solve the regularized optimal control problem in a distributed manner. Finally, in Section 8.7,

the chapter is concluded with remarks and a summary of outstanding challenges.

**8.2 Problem formulation** Consider the LTI control system in state-space form

$$\begin{aligned}\dot{\psi} &= A\psi + Bu \\ \zeta &= \begin{bmatrix} Q^{1/2} \\ 0 \end{bmatrix} \psi + \begin{bmatrix} 0 \\ R^{1/2} \end{bmatrix} u\end{aligned}\tag{8.1}$$

where  $\psi(t) \in \mathbb{R}^n$  is the state,  $u(t) \in \mathbb{R}^q$  is the control input, and  $\zeta(t) \in \mathbb{R}^{n+q}$  is the performance output. The matrices are all of compatible dimensions, and the standard assumptions are imposed:  $Q \succeq 0$ ,  $R \succ 0$ ,  $(A, B)$  is stabilizable and  $(A, Q^{1/2})$  is detectable.

The control input is defined by state-feedback,

$$u(t) = -K\psi(t)$$

with gain matrix  $K \in \mathbb{R}^{q \times n}$ . The closed-loop system is thus

$$\begin{aligned}\dot{\psi} &= A_{\text{cl}}\psi \\ \zeta &= \begin{bmatrix} Q^{1/2} \\ -R^{1/2}K \end{bmatrix} \psi\end{aligned}\tag{8.2}$$

with  $A_{\text{cl}} := A - BK$ .

Closed-loop performance is quantified by the square of the  $L_2$  norm of the impulse response for the closed-loop system, which is expressed using either of the following expressions:

$$\begin{aligned}f(K) &= \text{trace}(P) \\ &= \text{trace}((Q + K^T R K) L)\end{aligned}\tag{8.3}$$

where  $P \succ 0$  is the closed-loop observability Gramian,

$$A_{\text{cl}}^T P + P A_{\text{cl}} + Q + K^T R K = 0\tag{8.4}$$



and  $L \succ 0$  is the closed-loop controllability Gramian,

$$A_{\text{cl}} L + L A_{\text{cl}}^T + I = 0. \quad (8.5)$$

Our objective is to promote certain structural properties of the feedback gain matrix  $K$  by solving the regularized optimal control problem

$$\underset{K}{\text{minimize}} \quad f(K) + g(K) \quad (8.6)$$

where  $f$  is given by (8.3) and  $g$  is the regularization function. When it is desired to design  $K$  to belong to a set  $\mathcal{S}$  with a specified pattern of zero elements [114],  $g$  is an indicator function associated with the underlying sparsity constraints,

$$g(K) := \begin{cases} 0 & K \in \mathcal{S} \\ \infty & K \notin \mathcal{S}. \end{cases}$$

Similarly, the  $\ell_1$  norm and the nuclear norm are commonly used convex proxies for promoting sparsity, or for designing  $K$  with low rank. In this case,  $g(K) = \gamma \sum_{i,j} |K_{ij}|$  or  $g(K) = \gamma \|K\|_* = \gamma \sum_i \sigma_i(K)$ , where  $\gamma$  is a positive regularization parameter and  $\sigma_i$  is the  $i$ th singular value.

In general, the optimization problem (8.6) is not convex because  $f$  is a non-convex function of  $K$ . The proximal augmented Lagrangian method [108] can be used to compute a local minimum of (8.6); for details, see [173]. The minimization (8.6) remains non-convex even in the absence of the regularization function  $g$ . Of course, in this special case it reduces to the standard LQR problem, and the globally optimal solution is given by

$$K^* = R^{-1} B^T P$$

where  $P$  is the unique positive definite solution of the algebraic Riccati equation,

$$A^T P + P A + Q - P B R^{-1} B^T P = 0.$$

The recent work [117] established conditions for convergence of gradient descent methods

for the standard LQR problem to the global minimum, in spite of the lack of convexity.

The goal of this chapter is twofold: first is to obtain algorithms of tractable complexity for truly large-scale problems. A second goal is to solve the regularized, structured optimal control problem without knowledge of the underlying model. The approach to the second goal is a component of our approach to the first: a data-driven framework is proposed for computing the optimal feedback gain, with unknown matrices  $A$  and  $B$  in (8.2). This approach also avoids the need to solve large-scale Lyapunov equations.

**8.3 Computation from data** A square-additive property of the  $\mathcal{H}_2$  norm is used to obtain a decomposition of the objective function  $f$  and its gradient  $\nabla f$ . Duality arguments are then used to transform each term into a quantity that can be estimated from experiments or simulations. The resulting algorithm obtains estimates of the optimal gain without solving a Lyapunov equation, and without knowledge of the system parameters.

The gradient of  $f$  with respect to  $K$  is given by [135],

$$\nabla f(K) = 2(RK - B^T P)L. \quad (8.7)$$

In what follows, we use the three decompositions

$$L = \sum_{k=1}^n L^k, \quad f(K) = \sum_{k=1}^n f_k, \quad \nabla f(K) = \sum_{k=1}^n \nabla f_k$$

where  $L^k \succeq 0$  solves the Lyapunov equation

$$A_{\text{cl}} L^k + L^k A_{\text{cl}}^T + e_k e_k^T = 0, \quad (8.8)$$

$e_k$  is the  $k$ th unit vector in the canonical basis of  $\mathbb{R}^n$ , and

$$\begin{aligned} f_k(K) &= \text{trace}((Q + K^T R K) L^k) \\ \nabla f_k(K) &= 2(RK - B^T P)L^k. \end{aligned} \quad (8.9)$$

**8.3.1 Computation of  $P$  and  $L$  from data** The solution to (8.8) can be expressed

$$L^k = \int_0^\infty e^{A_{\text{cl}} t} e_k e_k^T e^{A_{\text{cl}}^T t} dt = \int_0^\infty \psi^k(t) (\psi^k(t))^T dt \quad (8.10)$$

where  $\psi^k(t) := e^{A_{\text{cl}} t} e_k$  is the solution to

$$\dot{\psi}^k = A_{\text{cl}} \psi^k, \quad \psi^k(0) = e_k, \quad k = 1, \dots, n. \quad (8.11)$$

This leads to the first component of the data-driven architecture: for a given feedback gain  $K$ , the term  $KL^k$  that appears both in  $f$  and  $\nabla f$  can be expressed as the integral

$$KL^k = \int_0^\infty u^k(t) (\psi^k(t))^T dt$$

where  $\psi^k(t)$  and  $u^k(t) = K\psi^k(t)$  are obtained from simulations or experiments of system (8.11).

Similarly, the observability Gramian is given by

$$P = \int_0^\infty e^{A_{\text{cl}}^T t} (Q + K^T R K) e^{A_{\text{cl}} t} dt$$

and its  $(i, j)$ th component is

$$P_{i,j} = \int_0^\infty (\varphi^i(t))^T (Q + K^T R K) \varphi^j(t) dt \quad (8.12)$$

where  $\varphi^i(t) := e^{A_{\text{cl}} t} e_i$  is the solution to

$$\dot{\varphi}^i = A_{\text{cl}} \varphi^i, \quad \varphi^i(0) = e_i, \quad i = 1, \dots, n.$$

Thus, using  $2n$  forward-in-time numerical simulations, the gradient of the objective function  $\nabla f(K)$  can be computed without solving a Lyapunov equation. Computation can be further reduced by avoiding multiplication of  $P$  and  $L^k$  in (8.9). A data-driven approach to compute the product  $PL^k$  directly is introduced in the next subsection.

**8.3.2 Computation of  $PL$  from data** As showed in [111,172], the product  $PL^k$  that appears in  $\nabla f_k(K)$  can be expressed as

$$X^k := PL^k = - \int_0^\infty \phi^k(t) (\psi^k(t))^T dt \quad (8.13)$$

where  $\phi^k$  is obtained from the solution of the adjoint system

$$\begin{aligned} \dot{\phi}^k &= -A_{\text{cl}}^T \phi^k + (Q + K^T R K) \psi^k \\ \phi^k(\infty) &= 0 \end{aligned} \quad (8.14)$$

and  $\psi^k$  is the solution to (8.11). Thus, numerical simulations of the primal and adjoint systems (8.11) and (8.14) along with numerical evaluations of the corresponding integrals can be used to compute  $L^k$ ,  $PL^k$ ,  $f_k$ , and  $\nabla f_k$ .

Next, we show that the matrix  $X^k$  in (8.13) can be computed without simulating the adjoint system (8.14). This is a critically important step in the model-free setup where only experimental or numerical data is available. Introduce the new variable

$$\eta^k := (Q + K^T R K) \psi^k$$

so that (8.14) becomes

$$\dot{\phi}^k = -A_{\text{cl}}^T \phi^k + \eta^k, \quad \phi^k(\infty) = 0 \quad (8.15)$$

which admits the solution,

$$\phi^k(t) = [H\eta^k](t) = - \int_t^\infty e^{-A_{\text{cl}}^T(t-\tau)} \eta^k(\tau) d\tau. \quad (8.16)$$

The linear operator  $H$  is introduced here to facilitate an adjoint transformation below. This operator and its adjoint  $H^*$  are defined on  $L_2([0, \infty)) \rightarrow \mathbb{R}^n$ . For any functions  $\xi, \chi \in L_2([0, \infty))$ , we have by definition  $\langle \chi, H\xi \rangle = \langle H^*\chi, \xi \rangle$ ; see, e.g., [174]. An explicit representation for  $\mu = H^*\gamma$  is obtained using elementary calculus:

$$\mu(t) = \int_0^t e^{A_{\text{cl}}(t-\tau)} \gamma(\tau) d\tau.$$

The  $(i, j)$ th element of the matrix  $X^k$  can be expressed

$$\begin{aligned} X_{i,j}^k &:= \mathbf{e}_i^T X_k \mathbf{e}_j = - \int_0^\infty \phi_i^k(t) \psi_j^k(t) dt \\ &= - \langle \psi_j^k, \phi_i^k \rangle \end{aligned} \quad (8.17)$$

where  $\phi_i^k(t) := \mathbf{e}_i^T \phi^k(t)$  and  $\psi_j^k(t) := \mathbf{e}_j^T \psi^k(t)$  are the  $i$ th and  $j$ th elements of the vectors  $\phi^k(t)$  and  $\psi^k(t)$ , respectively, and the inner product is in  $L_2([0, \infty))$ . By substituting the expression (8.16) for  $\phi^k$  in (8.17), we obtain

$$X_{i,j}^k = - \langle \psi_j^k, \mathbf{e}_i^T H \eta^k \rangle = - \langle H^* \mathbf{e}_i \psi_j^k, \eta^k \rangle =: \langle \xi_{i,j}^k, \eta^k \rangle. \quad (8.18)$$

Applying the expression for  $H^*$  then gives

$$\xi_{i,j}^k(t) := - \left[ H^* \mathbf{e}_i \mathbf{e}_j^T \psi^k \right] (t) = \int_0^t \mathbf{e}^{A_{\text{cl}}(t-\tau)} \mathbf{e}_i \mathbf{e}_j^T \psi^k(\tau) d\tau.$$

This implies that  $\xi_{i,j}^k$  can be obtained as the solution to the LTI system,

$$\dot{\xi}_{i,j}^k = A_{\text{cl}} \xi_{i,j}^k + \mathbf{e}_i \mathbf{e}_j^T \psi^k, \quad \xi_{i,j}^k(0) = 0. \quad (8.19)$$

Thus, to compute  $\nabla f(K)$  for a given gain  $K$ , the only systems that need to be simulated are the forward in time systems (8.11) and (8.19). The system (8.11) is unforced, and its solution determines the input to the forced system (8.19).

**Remark 15.** *Similar adjoint techniques are used in analysis of reinforcement learning. The proof that the TD(1) algorithm solves a minimum norm problem is based on related adjoint transformations [175, 176]. And, a similar adjoint transformation is a crucial step in a Q-learning algorithm for deterministic continuous-time systems [177].*

**Remark 16.** *By writing the Lyapunov equation for the controllability Gramian as (8.8), unlike the requirement of stochastic simulations in [172], only deterministic simulations are needed in the present framework.*

**Remark 17.** *Since the optimal unstructured gain  $K^*$  only depends on the closed-loop observability Gramian  $P$  (which can be obtained from the solution of the algebraic Riccati*

equation),  $L$  does not influence  $K^*$  and its computation can be avoided. To promote structure (e.g., via proximal algorithms) we need to compute both  $P$  and  $L$  to form  $\nabla f$ . In contrast, only computation of  $P$  is required to form natural policy gradient, which is defined as [178, 179],

$$\nabla h(K) = \nabla f(K) L^{-1} = 2(RK - B^T P). \quad (8.20)$$

**8.4 Systems with special structure** If the underlying system has additional structure, evaluation of both the objective function and the corresponding gradients can be further simplified. We next illustrate how structure of symmetric systems and undirected consensus networks can be exploited to simplify computations. In both cases, the closed-loop  $L_2$  norm has an explicit convex dependance on the optimization variable and there is no need to conduct simulations of the adjoint system and numerically evaluate the underlying integrals.

#### Symmetric systems

Let  $A$  and  $K$  in (8.2) be symmetric matrices and let  $B = I$ . For a stabilizing  $K$ , the  $L_2$  norm of the impulse response for the closed-loop system (8.2) is determined by

$$f(K) = \frac{1}{2} \text{trace}((K - A)^{-1}(Q + K R K)).$$

Convex dependence of  $f$  on  $K$  can be established via a straightforward use of the Schur complement. Furthermore, the design for symmetric systems provides a useful starting point for the design of non-symmetric systems [180].

Equivalently,  $f(K)$  can be written as

$$\begin{aligned} f(K) &= \frac{1}{2} \text{trace}((K - A)^{-1}(Q + A R A)) + \frac{1}{2} \text{trace}(R K) + \frac{1}{2} \text{trace}(R A) \\ &= \frac{1}{2} \sum_{i=1}^n f_i(K) \end{aligned}$$

where

$$f_i(K) = q_i^T(K - A)^{-1}q_i + a_i^T(K - A)^{-1}a_i + r_i^T(K + A)r_i.$$

The vectors indexed by  $i$  in this expression are given by

$$q_i = Q^{1/2} e_i, \quad r_i = R^{1/2} e_i, \quad a_i = A r_i$$

where  $e_i$  is the  $i$ th canonical basis vector in  $\mathbb{R}^n$ . Furthermore, the gradient of  $f_i$  with respect to  $K$  is given by

$$\nabla f_i(K) = r_i r_i^T - (K - A)^{-1} q_i q_i^T (K - A)^{-1} - (K - A)^{-1} a_i a_i^T (K - A)^{-1}$$

Thus, there is no need to compute and store the inverse of the matrix  $K - A$  in order to evaluate  $f_i(K)$  and  $\nabla f_i(K)$ ; only actions of  $(K - A)^{-1}$  on the vectors  $q_i$  and  $a_i$  is necessary. For example, the preconditioned conjugate gradients method can be used to compute them efficiently.

**Remark 18.** *The vector  $(K - A)^{-1}q_i$  represents the steady-state solution of a stable linear system*

$$\dot{\psi}_i = (A - K)\psi_i + q_i$$

*and it can be computed via numerical integration. This illustrates that the symmetric nature of the closed-loop system (8.2) allows us to avoid the need for numerical integration of the corresponding adjoint system.*

### Undirected consensus networks

For undirected consensus networks with  $n$  nodes, the matrices  $A$  and  $K \in \mathbb{R}^{n \times n}$  in (8.2) determine the graph Laplacians of the plant and controller networks, respectively. In the absence of exogenous disturbances, the network converges to the average of the initial node values  $\bar{\psi} = (1/n) \sum_i \psi_i(0)$  if and only if it is connected [2]. Let  $B = I$  and let  $Q := I - (1/n)\mathbf{1}\mathbf{1}^T$  penalize the deviation of individual node values from average. The objective is to minimize the mean square deviation from the network average by adding a few additional edges, specified by the graph Laplacian  $K$  of a controller network. If

$E$  is the incidence matrix of the controller graph,  $K$  can be written as

$$K(x) = E \operatorname{diag}(x) E^T$$

where  $\operatorname{diag}(x)$  is a diagonal matrix containing the optimization variable  $x \in \mathbb{R}^m$  (i.e., the vector of the edge weights in the controller graph). Regularization terms may be used to promote sparsity of the controller network or to impose some additional constraints on the edge weights.

As shown in [106], up to an additive constant, the  $L_2$  norm of the impulse response for the closed-loop system is determined by

$$f(x) = \operatorname{trace}((E \operatorname{diag}(x) E^T - A)^\dagger (I + A R A)) + \operatorname{diag}(E^T R E)^T x$$

where the pseudo-inverse of the closed-loop graph Laplacian is given by

$$(E \operatorname{diag}(x) E^T - A)^\dagger = (E \operatorname{diag}(x) E^T + (1/n) \mathbf{1}\mathbf{1}^T - A)^{-1}.$$

It is easy to show that  $f(x)$  can be written as

$$f(x) = \sum_{i=1}^n f_i(x)$$

where

$$f_i(x) = \xi_i^T (E \operatorname{diag}(x) E^T + (1/n) \mathbf{1}\mathbf{1}^T - A)^{-1} \xi_i + (1/n) \operatorname{diag}(E^T R E)^T x. \quad (8.21)$$

Here,  $\xi_i = (I + A R A)^{1/2} \mathbf{e}_i$  is the  $i$ th column of the square root of the matrix  $(I + A R A)$ . Moreover, it can be shown that the gradient of  $f_i(x)$  is given by

$$\nabla f_i(x) = (1/n) \operatorname{diag}(E^T R E) - \nu_i(x) \circ \nu_i(x) \quad (8.22)$$

where  $\circ$  is the elementwise multiplication and

$$\nu_i(x) = E^T (E \operatorname{diag}(x) E^T + (1/n) \mathbf{1}\mathbf{1}^T - A)^{-1} \xi_i. \quad (8.23)$$



As for symmetric systems, there is no need to compute and store the inverse of the matrix in (8.23) and the preconditioned conjugate gradients method can be used to compute the gradient efficiently.

We next provide a proposition on the Lipschitz continuity of the gradient of  $f_i$  for connected resistive networks which enables us to choose appropriate step-size for proximal gradient algorithms. In such networks, the plant graph is connected and all edge weights in both the plant and the controller graphs are non-negative.

**Proposition 15.** *The gradient of  $f_i$  for connected resistive networks is Lipschitz continuous with the Lipschitz constant*

$$L_i = \xi_i^T \hat{A}(0)^{-1} E E^T \hat{A}(0)^{-1} \xi_i \|E^T \hat{A}(0)^{-1} E\|_2 \quad (8.24)$$

where  $\hat{A}(x) = E \text{diag}(x) E^T + (1/n) \mathbb{1}\mathbb{1}^T - A$ .

*Proof.* For a convex and twice differentiable function  $f_i$ , the gradient  $\nabla f_i$  is Lipschitz continuous with Lipschitz constant  $L_i$ , if

$$\nabla^2 f_i(x) \preceq L_i I$$

or, equivalently,  $z^T \nabla^2 f_i(x) z \leq L_i \|z\|^2$ . It can be shown that the second order derivative of  $f_i$  is given by

$$\nabla^2 f_i(x) = (\nu_i \nu_i^T) \circ (E^T \hat{A}(x)^{-1} E).$$

Thus,  $z^T \nabla^2 f_i(x) z$  can be written as

$$\xi_i^T \hat{A}(x)^{-1} E D_z E^T \hat{A}(x)^{-1} E D_z E^T \hat{A}(x)^{-1} \xi_i$$

where  $D_z := \text{diag}(z)$ . For positive definite matrices  $S$  and  $T$  with  $S \preceq T$ , we have  $z^T S z \leq z^T T z$  and

$$z^T S z \leq \lambda_{\max}(S) \|z\|^2 \leq \sqrt{\lambda_{\max}(S^T S)} \|z\|^2 = \|S\|_2 \|z\|^2$$

for any vector  $z$ . Therefore,

$$D_z E^T \hat{A}(x)^{-1} E D_z \preceq \|E^T \hat{A}(0)^{-1} E\|_2 \|z\|^2.$$

As a result,

$$z^T \nabla^2 f_i(x) z \leq \nu_i(0)^T \nu_i(0) \|E^T \hat{A}(0)^{-1} E\|_2 \|z\|^2$$

where  $\nu_i(0) = E^T \hat{A}(0)^{-1} \xi_i$ . □

**8.5 Proximal algorithms** Algorithms of tractable complexity are introduced here for solving the regularized structured optimal control problem (8.6). A standard proximal gradient descent algorithm is considered as a starting point. It is argued however that computing the full gradient may be prohibitively expensive. Thus, we propose a more efficient algorithm based on incremental proximal gradient that is well-suited for the design of structured optimal feedback gains in large-scale systems.

**8.5.1 Proximal gradient descent** The proximal gradient algorithm is a first-order method that extends standard gradient descent to a class of non-smooth composite optimization problems. In our setup, it iteratively updates the feedback gain  $K$  via application of the proximal operator associated with the function  $g$  on the standard gradient descent update. The algorithm is initialized with a stabilizing  $K^0$  and the iterates are determined by,

$$K^{l+1} = \mathbf{prox}_{\alpha_l g} \left( K^l - \alpha_l \nabla f(K^l) \right)$$

where  $l$  is the iteration index and  $\alpha_l$  is the step-size. At each iteration, the step-size is selected via backtracking to guarantee sufficient descent of the objective function and stability of closed-loop system (8.2). To enhance practical performance, we use the Barzilai-Borwein (BB) step-size initialization [127].

For any matrix  $V$  and a positive scalar  $\alpha$ , the proximal operator of the function  $g$  is defined as [76],

$$\mathbf{prox}_{\alpha g}(V) := \underset{Z}{\operatorname{argmin}} \left( g(Z) + \frac{1}{2\alpha} \|Z - V\|_F^2 \right)$$

where  $\|\cdot\|_F$  is the Frobenius norm. In particular, for  $g(K) = \gamma \|K\|_1 = \gamma \sum_{i,j} |K_{ij}|$ , we have

$$K^{l+1} = \mathcal{S}_{\gamma\alpha_l}(K^l - \alpha_l \nabla f(K^l))$$

where  $\mathcal{S}_\kappa(V)$  is the soft-thresholding function that acts on the individual entries  $V_{ij}$  of the matrix  $V$  according to  $\mathcal{S}_\kappa(V_{ij}) = \text{sign}(V_{ij}) \max(|V_{ij}| - \kappa, 0)$ . In the absence of the regularization function  $g$  in (8.6), the proximal gradient algorithm simplifies to standard gradient descent with the updates  $K^{l+1} = K^l - \alpha_l \nabla f(K^l)$ .

Recall that estimates of  $\nabla f(K^l)$  require the matrices  $KL$  and  $PL$ . These can be computed or estimated from data using the techniques described in the previous section.

**Remark 19.** *The step-size  $\alpha_l$  is adjusted via backtracking to ensure stability of closed-loop system (8.2) at each iteration. It is initialized using the BB method which provides a heuristic for approximating the Hessian of the function  $f$  via the scaled version of the identity matrix [127],  $(1/\alpha_l)I$ . At the  $l$ th iteration, the initial BB step-size*

$$\alpha_{l,0} := \frac{\|K^l - K^{l-1}\|_2^2}{\text{trace}((K^{l-1} - K^l)^T (\nabla f(K^{l-1}) - \nabla f(K^l)))} \quad (8.25)$$

*is adjusted via backtracking until system (8.2) is stable and sufficient descent is achieved at each iteration, i.e.,*

$$\begin{aligned} f(K^{l+1}) &< f(K^l) + \text{trace}((K^{l+1} - K^l)^T \nabla f(K^l)) \\ &\quad + \frac{1}{2\alpha_l} \|K^{l+1} - K^l\|_2^2. \end{aligned} \quad (8.26)$$

*The algorithm stops when the generalized gradient map becomes smaller than the given tolerance,*

$$\left\| \frac{K^l - \mathbf{prox}_{\alpha_l g}(K^l - \alpha_l \nabla f(K^l))}{\alpha_l} \right\| < \epsilon.$$

**8.5.2 Incremental proximal gradient** Next, we exploit the separable structure obtained in Section 8.2 to speed up computations and save memory. This formulation allows to work with a single function  $f_k$  and its gradient  $\nabla f_k$  in the controller design, which reduces computational complexity and allows implementation at scale.

We start with an initial stabilizing gain  $K^0$ , and at each iteration we solve the

following optimization problem

$$\underset{K}{\text{minimize}} \quad f_k(K) + g(K). \quad (8.27)$$

The explicit solution to (8.27) is given by

$$K^{l+1} = \mathbf{prox}_{\alpha_l g} \left( K^l - \alpha_l \nabla f_k(K^l) \right).$$

Here,  $\mathbf{prox}$  is the proximal operator of the function  $g$ , and  $\alpha_l$  is a step-size. This is the incremental proximal gradient algorithm because, at each iteration, the optimization variable is updated in the negative direction of a single element of the gradient where the index  $k$  is selected in a random manner. At each iteration, we choose step-size to be  $O(1/l)$  and adjust it to guarantee stability of closed-loop system (8.2).

Incremental and stochastic gradient based algorithms have recently found widespread use in large-scale optimization and machine learning. High variance that results from estimating the full gradient  $\nabla f$  using samples of its entries can result in slow convergence. Moreover, to ensure convergence, the step-size has to decay to zero.

In the first few iterations, the objective function in the incremental proximal gradient decreases dramatically but it starts to oscillate after that. Thus, in large-scale optimization problems where having high accuracy may not be achievable, these methods are useful because of their fast initial convergence rate.

**8.6 An ADMM-based consensus algorithm** We next demonstrate that a standard consensus algorithm based on the Alternating Direction Method of Multipliers (ADMM) is well-suited for distributed design of optimal structured feedback gains. As shown in Section 8.3, the regularized optimal control problem can be written as

$$\underset{K}{\text{minimize}} \quad \sum_{i=1}^N f_i(K) + g(K). \quad (8.28)$$

This characterization is suitable for distributed implementation in which each processors solves an optimization problem. By introducing  $N$  local variables  $K_i$  and a global

variable  $K_0$ , problem (8.28) can be brought into a standard consensus form,

$$\begin{aligned} & \underset{K_i, K_0}{\text{minimize}} \quad \sum_{i=1}^N f_i(K_i) + g(K_0) \\ & \text{subject to} \quad K_i - K_0 = 0, \quad i = 1, \dots, N. \end{aligned} \quad (8.29)$$

This formulation increases the number of optimization variables but it brings the objective function into a separable form and facilitates distributed computations. When implemented on a single machine, solving the reformulated problem is not necessarily more computationally efficient than solving the original problem using centralized algorithms. However, this formulation allows to work with a local objective function  $f_i$  that depends on a single local variable  $K_i$ . This is advantageous for large-scale systems where centralized algorithms cannot be afforded (e.g., because of the computational cost associated with solving the large-scale Lyapunov equations).

The augmented Lagrangian associated with (8.29) is

$$\mathcal{L}(K_i, K_0; \Lambda) = g(K_0) + \sum_{i=1}^N (f_i(K_i) + \langle \Lambda_i, K_i - K_0 \rangle + \frac{\rho_i}{2} \|K_i - K_0\|_F^2) \quad (8.30)$$

where  $\Lambda_i$ 's are the Lagrange multipliers and  $\rho_i$ 's are positive parameters. The ADMM algorithm consists of the following iterative steps,

$$\begin{aligned} K_i^{k+1} &= \underset{K_i}{\text{argmin}} \quad f_i(K_i) + \frac{\rho_i}{2} \|K_i - U_i^k\|_F^2 \\ K_0^{k+1} &= \underset{K_0}{\text{argmin}} \quad g(K_0) + \sum_{i=1}^N \frac{\rho_i}{2} \|K_0 - V_i^k\|_F^2 \\ \Lambda_i^{k+1} &= \Lambda_i^k + \rho_i (K_i^{k+1} - K_0^{k+1}) \end{aligned}$$

where

$$\begin{aligned} U_i^k &:= K_0^k - (1/\rho_i) \Lambda_i^k \\ V_i^k &:= K_i^{k+1} + (1/\rho_i) \Lambda_i^k. \end{aligned}$$

The  $K_i$ -minimization step can be done via distributed computation by spreading subproblems to  $N$  different processors. On the other hand, the update of  $K_0$  amounts

to the evaluation of the proximal operator of the function  $g$ ,

$$K_0^{k+1} = \mathbf{prox}_{g/(N\bar{\rho})} \left( \frac{1}{N\bar{\rho}} \sum_{i=1}^N \rho_i V_i^k \right).$$

where  $\bar{\rho} := (1/N) \sum_i \rho_i$ . Thus, the update of  $K_0$  requires gathering each  $K_i^{k+1}$  and the associated Lagrange multipliers  $\Lambda_i^k$  in order to form  $V_i^k$ .

**Remark 20.** *The above presented consensus algorithm is standard (e.g., see [113]) but to the best of our knowledge it has not been previously used for optimal design of structured feedback gains via distributed optimization. Recently, convergence of this algorithm was established even for problems with non-convex objective functions  $f_i$  [112]. The authors of [112] also show that additional computational advantage can be gained by updating only a subset of  $K_i$ 's in each iteration and offer several alternative implementations to speed computations.*

**Remark 21.** *The update of each  $K_i$  amounts to the computation of the proximal operator associated with  $f_i$ ,*

$$K_i^{k+1} = \mathbf{prox}_{f_i/\rho_i} (U_i^k)$$

*However, while the proximal operators of commonly used regularization functions are easy to evaluate, the computation of  $\mathbf{prox}_{f_i/\rho_i}$  is significantly more involved. In [42], the proximal operator associated with the closed-loop  $\mathcal{H}_2$  norm  $f(K)$  was computed using the Anderson-Moore algorithm. This approach requires computation of the closed-loop controllability and observability gramians. In contrast, the developments of Section 8.3 facilitate computation of  $\mathbf{prox}_{f_i/\rho_i}$  via a proximal gradient algorithm that avoids the need for solving large-scale Lyapunov equations.*

**Remark 22.** *For the problem of growing connected resistive consensus networks [106], Algorithm 3 in [112] can be used to solve the  $K_i$ -minimization step in the ADMM-based algorithm explicitly. The key point of departure compared to the standard ADMM implementation is linearization in the  $K_i$ -minimization step of the function  $f_i$  with a Lipschitz continuous gradient around the current  $K_0$  iterate. This offers a significant speed-up and enables an explicit update of  $K_i$ .*

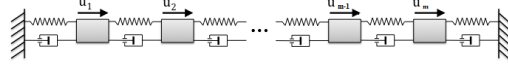


Figure 8.1: A mass-spring-damper system on a line.

**8.6.1 Computational experiments** We next provide examples to illustrate the performance of proximal algorithms. The acronym proxG represents the proximal gradient and proxIG is the incremental proximal gradient. We have implemented all algorithms in MATLAB. In all examples, we choose  $R = I$  and  $Q = I$ .

The algorithms were tested for a mass-spring-damper (MSD) model with  $N$  masses, illustrated in Figure 8.1. Our goal is to find the local minimum of the problem (8.6) with  $g(K) = \gamma \|K\|_1$ , where  $\gamma$  is the sparsity-promoting parameter and the  $\ell_1$  norm is a proxy for inducing sparsity in the feedback gain matrix. The MSD system with  $N$  masses has  $n = 2N$  states where the first  $N$  states denote the positions and the rest are velocities. We consider the case where all masses, spring, and damping constants are equal to one, which results in the state-space model with

$$A = \begin{bmatrix} 0 & I \\ -T & -T \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

where  $T$  is an  $N \times N$  tridiagonal symmetric Toeplitz matrix with 2 on the main diagonal and  $-1$  on the first upper and lower sub-diagonals, and  $I$  and  $0$  are  $N \times N$  identity and zero matrices.

Sparsity-promoting controllers were designed for this model with  $N = 10$  masses and compared with the ADMM-based algorithm [42] that does not utilize the iterative re-weighting scheme. The algorithms were initialized with the (unstructured) optimal feedback gain  $K^*$ . Figure 8.2 shows how the sparsity pattern of the controller changes as the value of the parameter  $\gamma$  is increased. It is worth to note that all algorithms give the same structure of  $K$  with very close feedback gain values. For  $\gamma = 0$ , the optimal feedback controller is given by a dense matrix and as  $\gamma$  increases the controller becomes sparser. It is diagonal when  $\gamma = 1$ , and for sufficiently large  $\gamma$  the gain is identically zero. This is a feasible solution because the open-loop system is stable.

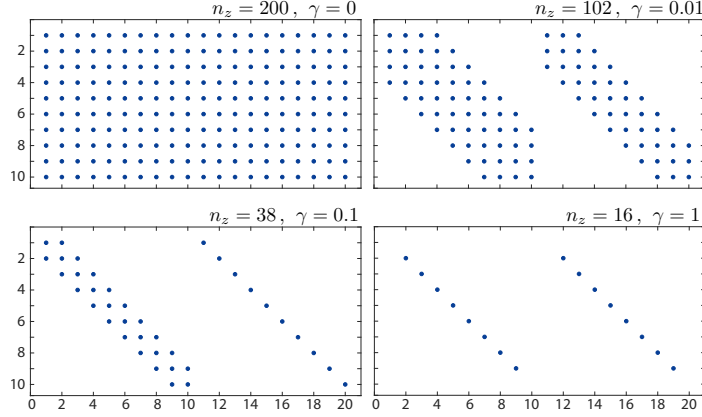


Figure 8.2: Structure of feedback gains resulting from the solution to (8.6) for different values of the regularization parameter  $\gamma$  for an MSD system with  $N = 10$  masses. The number of non-zero elements in  $K$  is denoted by  $n_z$ .

Next, we employ the ADMM-based consensus algorithm to design optimal structured feedback gains in a distributed fashion by splitting the problem into  $N$  separate subproblems over  $N$  different cores. We have provided a parallel implementation in C++ using pthreads library and executed tests on a machine featuring an Intel Core i7-3770 with 16GB of RAM to measure the performance of the algorithm. We use our algorithm for growing a connected resistive network with  $n = 20$  nodes. The plant graph is given by an Erdős-Rényi network with edge probability  $1.05 \log(n)/n$ . We choose the control weight matrix  $R = I$  and a state weight matrix that penalizes the mean-square deviation from the network average,  $Q = I - (1/n) \mathbf{1}\mathbf{1}^T$ . Moreover, the incidence matrix of the controller is such that there are no joint edges between the plant and the controller graphs. As discussed in Section 8.4, for consensus networks, the controller can be written as a function of the vector of the edge weights  $x$ . Thus, the  $\mathcal{H}_2$  norm of the closed-loop system is  $f(x)$  and the regularization function is given by  $g(x) = \gamma \|w \circ x\|_1$  where  $w$  is the vector of the weights. Since the plant network is resistive and connected, all the edge weights are nonnegative, thereby if the added edges have nonnegative edge weights, the closed-loop system is stable. We can write the smooth and nonsmooth parts of the objective function as  $f(x) + \gamma w^T x$  and  $g(x) = I_+(x)$  where  $I_+(x)$  is an indicator function for nonnegative orthant.



We solve the problem (8.28) to find the controller graph for 500 logarithmically-spaced values of  $\gamma \in [0.001, 0.3]$  using the path-following iterative reweighted algorithm as a proxy for inducing sparsity [125]. We set the weights to be inversely proportional to the magnitude of the solution  $x$  at the previous value of  $\gamma$  following by a polishing step that computes the optimal weights of identified edges; see [106].

As  $\gamma$  increases, the number of nonzero edges decreases and the closed-loop performance deteriorates. As shown in Fig. 8.3, relative to the optimal centralized vector of the edge weights,  $x_c$ , the  $\mathcal{H}_2$  loss decreases as the sparsity of the vector of the edge weights  $x$  increases. In particular, for  $\gamma = 0.3$ , there is only one nonzero element in the vector of the edge weights. The identified sparse controller in this case uses only 0.62% of the edges, relative to the optimal centralized controller, i.e.,  $\text{card}(x)/\text{card}(x_c) = 0.62\%$  and achieves a performance loss of 23.47%, i.e.,  $(J - J_c)/J_c = 23.47\%$ .

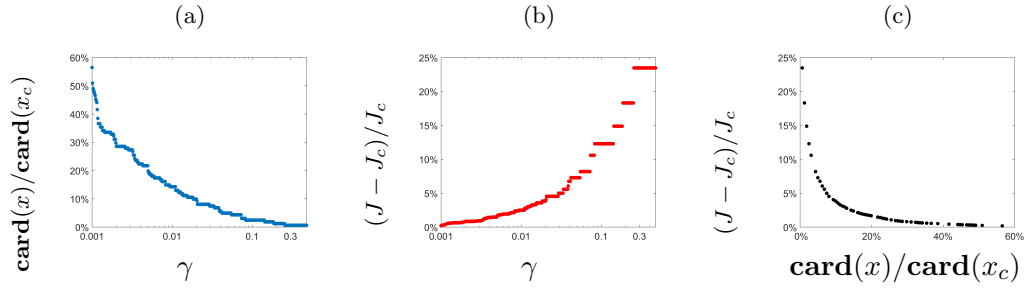


Figure 8.3: (a) Sparsity level; (b) performance degradation; and (c) the optimal trade-off between the performance degradation and the sparsity level of the optimal sparse  $x$  compared to the optimal centralized controller  $x_c$ . The results are obtained for a randomly generated Erdős-Rényi network with  $n = 20$  nodes.

Next, we employ our algorithm for growing connected resistive networks with different number of nodes using multiple cores. The solve times are averaged over 10 trials and the speedup relative to a single core is displayed in Fig 8.4. This figure demonstrates that the algorithm is scalable. In particular, multi-core execution outperforms running just on a single core. The speed-up is even higher for larger networks since overheads of parallel execution are smaller and more time is spent on actual parallel computation.

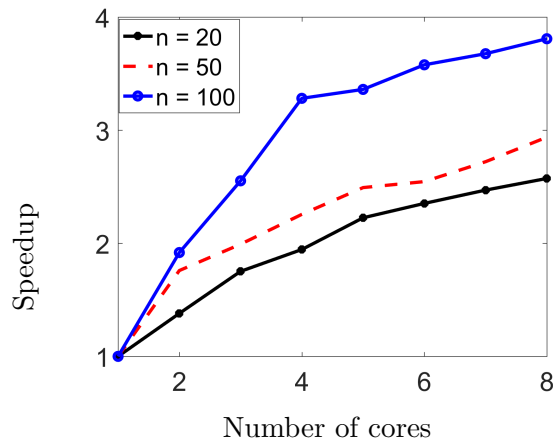


Figure 8.4: Speedup ratio versus the number of the cores used for growing connected resistive networks with  $n$  nodes.

**8.7 Concluding remarks** This chapter provides a step towards a comprehensive theory for distributed and structured optimal control of LTI systems. Even within the linear setting there are many open questions, including

- Development of simulation-based methods in the presence of disturbances and uncertainty;
- Construction of truly recursive algorithms, similar to those used in traditional reinforcement learning settings;
- Techniques to avoid local minima, and finding conditions that ensure non-existence of local minima for  $f$ ;
- Development of algorithms that combine proximal methods with natural policy gradient.

Extensions to nonlinear control will probably require more modest objectives since it is not an easy task to compute the performance of a given policy. However, both  $Q$ -learning and TD-learning are based on value function approximation. Consequently, as in the present work, these algorithms can be designed so that each value function approximation serves as a Lyapunov function for the current approximating policy.

This provides bounds on performance, as well as stability. Consequently, algorithms for computation of structured stabilizing policies with good performance are not out of reach even for nonlinear control systems.

## Chapter 9

# Conclusions

This dissertation focuses on structure identification and optimal control of large-scale networks of dynamical systems. To overcome the challenges in this area, we combine tools and ideas from control theory, distributed optimization, reinforcement learning, and compressive sensing to develop distributed estimation and control strategies that require limited information exchange between the individual subsystems. In the first and second parts, we have studied the problems of optimal distributed design and inference of large-scale networks of dynamical systems. We have developed theoretical and computational frameworks to solve these problems. In particular, we have designed and implemented customized optimization algorithms that are well-suited for solving the above-mentioned problems in large-scale dynamical systems. In the third chapter, we have studied the convergence properties of these algorithms from system-theoretic point of view. We have utilized control-theoretic tools to view different optimization algorithms as dynamical systems and study their convergence properties by using stability analysis techniques from control theory. Finally, in the last chapter, we have provided a data-driven framework for the structured optimal control problem that does not require knowledge of the system parameters and avoids the need to solve large-scale matrical equations.

### **Future directions**

The current research on design and control of networks can be extended to make them applicable to a larger class of systems such as unbalanced time-varying large-scale

networks. We have also developed an algorithm to recover the underlying undirected network topology of a complex system using partially available statistical data. We will extend this framework to identify the structure of directed networks which is a more challenging problem since the resulting optimization problem is non-convex. Also, we are utilizing optimization theoretical results to develop a new model with more complicated dynamics than first-order consensus. Furthermore, we are extending the work on distributed computation over networks to solve the non-smooth optimization problems with more than one regularizers. In this case, the dynamical system of the primal-dual updates is more complicated and convergence analysis is more challenging. Moreover, we will extend the data-driven framework to be truly recursive, similar to those used in traditional reinforcement learning settings and will combine the proximal methods with natural policy gradient. The framework can be generalized by developing simulation-based methods in the presence of disturbances and uncertainty.

# References

- [1] G. E. Dullerud and F. Paganini. *A course in robust control theory*. Springer, 2000.
- [2] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
- [3] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Syst. Control Lett.*, 53:65–78, 2004.
- [4] Y. Kim and M. Mesbahi. On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian. *IEEE Trans. Automat. Control*, 51(1):116–120, 2006.
- [5] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *J. Parallel Distrib. Comput.*, 67(1):33–46, 2007.
- [6] P. Barooah and J. P. Hespanha. Estimation on graphs from relative measurements: Distributed algorithms and fundamental limits. *IEEE Control Syst. Mag.*, 27(4):57–74, 2007.
- [7] A. Ghosh, S. Boyd, and A. Saberi. Minimizing effective resistance of a graph. *SIAM Rev.*, 50(1):37–66, 2008.
- [8] D. Zelazo and M. Mesbahi. Edge agreement: Graph-theoretic performance bounds and passivity analysis. *IEEE Trans. Automat. Control*, 56(3):544–555, 2011.
- [9] D. Zelazo, S. Schuler, and F. Allgöwer. Performance and design of cycles in consensus networks. *Syst. Control Lett.*, 62(1):85–96, 2013.

- [10] B. Bamieh, M. R. Jovanovic, P. Mitra, and S. Patterson. Coherence in large-scale networks: dimension dependent limitations of local feedback. *IEEE Trans. Automat. Control*, 57(9):2235–2249, September 2012. (2013 George S. Axelby Outstanding Paper Award).
- [11] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.
- [12] C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34, 1987.
- [13] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Physical Review Letters*, 75(6):1226, 1995.
- [14] M. H. DeGroot. Reaching a consensus. *J. Amer. Statist. Assoc.*, 69(345):118–121, 1974.
- [15] B. Golub and M. Jackson. Naive learning social networks and the wisdom of crowds. *American Economic Journal: Microeconomics*, 2(1):112–149, 2010.
- [16] F. Dorfler, M. R. Jovanovic, M. Chertkov, and F. Bullo. Sparsity-promoting optimal wide-area control of power networks. *IEEE Trans. Power Syst.*, 29(5):2281–2291, September 2014.
- [17] X. Wu, F. Dorfler, and M. R. Jovanovic. Decentralized optimal control of inter-area oscillations in bulk power systems. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pages 5532–5537, Osaka, Japan, 2015.
- [18] X. Wu, F. Dorfler, and M. R. Jovanovic. Input-output analysis and decentralized optimal control of inter-area oscillations in power systems. *IEEE Trans. Power Syst.*, 31(3):2434–2444, May 2016.
- [19] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *J. Parallel Distrib. Comput.*, 7(2):279–301, 1989.
- [20] J. E. Boillat. Load balancing and Poisson equation in a graph. *Concurrency: Practice and Experience*, 2(4):289–313, 1990.

- [21] V. Preciado, M. Zargham, C. Enyioha, A. Jadbabaie, and G. Pappas. Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 7486–7491. IEEE, 2013.
- [22] E. Ramírez-Llanos and S. Martínez. A distributed nonlinear dynamics for virus spread control. *Automatica*, 2016. To appear.
- [23] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Automat. Control*, 48(6):988–1001, 2003.
- [24] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Automat. Control*, 49(9):1520–1533, 2004.
- [25] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Trans. Automat. Control*, 50(2):169–182, 2005.
- [26] R. Carli, F. Fagnani, A. Speranzon, and S. Zampieri. Communication constraints in the average consensus problem. *Automatica*, 44(3):671–684, 2007.
- [27] S. Hassan-Moghaddam, N. K. Dhingra, and M. R. Jovanovic. Topology identification of undirected consensus networks via sparse inverse covariance estimation. In *Proceedings of the 55th IEEE Conference on Decision and Control*, pages 4624–4629, Las Vegas, NV, 2016.
- [28] S. Hassan-Moghaddam and M. R. Jovanovic. Topology identification via growing a chow-liu tree network. In *Proceedings of the 57th IEEE Conference on Decision and Control*, pages 5421–5426, Miami, FL, 2018.
- [29] M. Fazlyab, A. Ribeiro, M. Morari, and V. M. Preciado. Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems. 2017. arXiv:1705.03615.



- [30] B. Hu and P. Seiler. Exponential decay rate conditions for uncertain linear systems using integral quadratic constraints. *IEEE Trans. Autom. Control*, 61(11):3631–3637, 2016.
- [31] L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. Optim.*, 26(1):57–95, 2016.
- [32] Y. Wan, S. Roy, and A. Saberi. Designing spatially heterogeneous strategies for control of virus spread. *IET Systems Biology*, 2(4):184–201, 2008.
- [33] M. Siami and N. Motee. Tractable approximation algorithms for the np-hard problem of growing linear consensus networks. In *Proceedings of the 2016 American Control Conference*, pages 6429–6434, 2016.
- [34] A. Ghosh and S. Boyd. Growing well-connected graphs. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6605–6611, 2006.
- [35] F. Lin, M. Fardad, and M. R. Jovanovic. Identification of sparse communication graphs in consensus networks. In *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing*, pages 85–89, Monticello, IL, 2012. (Invited paper).
- [36] M. Fardad, F. Lin, and M. R. Jovanovic. Design of optimal sparse interconnection graphs for synchronization of oscillator networks. *IEEE Trans. Automat. Control*, 59(9):2457–2462, September 2014.
- [37] T. H. Summers, I. Shames, J. Lygeros, and F. Dörfler. Topology design for optimal network coherence. In *Proceedings of the 2015 European Control Conference*, pages 575–580, 2015.
- [38] B. Polyak, M. Khlebnikov, and P. Shcherbakov. An LMI approach to structured sparse feedback design in linear control systems. In *Proceedings of the 2013 European Control Conference*, pages 833–838, 2013.
- [39] N. K. Dhingra, M. R. Jovanovic, and Z. Q. Luo. An admm algorithm for optimal sensor and actuator selection. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 4039–4044, Los Angeles, CA, 2014. (Invited paper).

- [40] A. Zare, H. Mohammadi, N. K. Dhingra, T. T. Georgiou, and M. R. Jovanovic. Proximal algorithms for large-scale statistical modeling and optimal sensor/actuator selection. *IEEE Trans. Automat. Control*, 2019. doi:10.1109/TAC.2019.2948268; also arXiv:1807.01739.
- [41] M. Fardad, F. Lin, and M. R. Jovanovic. Sparsity-promoting optimal control for a class of distributed systems. In *Proceedings of the 2011 American Control Conference*, pages 2050–2055, San Francisco, CA, 2011.
- [42] F. Lin, M. Fardad, and M. R. Jovanovic. Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Trans. Automat. Control*, 58(9):2426–2431, September 2013.
- [43] J. F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11(1-4):625–653, 1999.
- [44] Kim-Chuan Toh, M. J. Todd, and R. H. Tütüncü. SDPT3 – A Matlab software package for semidefinite programming, version 1.3. *Optim. Methods Softw.*, 11(1-4):545–581, 1999.
- [45] Pedro A Valdés-Sosa, Jose M Sánchez-Bornot, Agustín Lage-Castellanos, Mayrim Vega-Hernández, Jorge Bosch-Bayard, Lester Melie-García, and Erick Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Phil. Trans. R. Soc. B*, 360(1457):969–981, 2005.
- [46] Dongchuan Yu, Marco Righero, and Ljupco Kocarev. Estimating topology of networks. *Phys. Rev. Lett.*, 97(18):188701, 2006.
- [47] Michael M Zavlanos, A Agung Julius, Stephen P Boyd, and George J Pappas. Identification of stable genetic networks using convex programming. In *Proceedings of the 2008 American Control Conference*, pages 2755–2760, 2008.
- [48] Domenico Napoletani and Timothy D Sauer. Reconstructing the topology of sparsely connected dynamical networks. *Physical Review E*, 77(2):026103, 2008.

- [49] Mustafa Ayazoglu, Mario Sznaiar, and Necmiye Ozay. Blind identification of sparse dynamic networks and applications. In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 2944–2950, 2011.
- [50] Marzieh Nabi-Abdolyousefi, Maryam Fazel, and Mehran Mesbahi. A graph realization approach to network identification. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 4642–4647, 2012.
- [51] D. Materassi and M. V. Salapaka. On the problem of reconstructing an unknown topology via locality properties of the wiener filter. *IEEE Trans. Automat. Control*, 57(7):1765–1777, 2012.
- [52] D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *JMLR*, 20(3):197–243, 1995.
- [53] A. Zare, Y. Chen, M. R. Jovanovic, and T. T. Georgiou. Low-complexity modeling of partially available second-order statistics: theory and an efficient matrix completion algorithm. *IEEE Trans. Automat. Control*, 62(3):1368–1383, March 2017.
- [54] A. Zare, M. R. Jovanovic, and T. T. Georgiou. Colour of turbulence. *J. Fluid Mech.*, 812:636–680, February 2017.
- [55] A. Zare, T. T. Georgiou, and M. R. Jovanovic. Stochastic dynamical modeling of turbulent flows. *Annu. Rev. Control Robot. Auton. Syst.*, 3, May 2020. in press; doi:10.1146/annurev-control-053018-023843; also arXiv:1908.09487.
- [56] Mikail M. Rubinov and O. Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069, 2010.
- [57] Bharat Biswal, F Zerrin Yetkin, Victor M Haughton, and James S Hyde. Functional connectivity in the motor cortex of resting human brain using echo-planar mri. *Magn. Reson. Med.*, 34(4):537–541, 1995.
- [58] M. Lynall, D. S. Bassett, R. Kerwin, P. J. McKenna, M. Kitzbichler, U. Muller, and E. Bullmore. Functional connectivity and brain networks in schizophrenia. *J. Neurosci.*, 30(28):9477–9487, 2010.

- [59] D. Cordes, V. Haughton, J. D. Carew, K. Arfanakis, and K. Maravilla. Hierarchical clustering to measure connectivity in fmri resting-state data. *JMRI*, 20(4):305–317, 2002.
- [60] P. E. Vértes, A. F. Alexander-Bloch, N. Gogtay, J. N. Giedd, J. L. Rapoport, and E. T. Bullmore. Simple models of human brain functional networks. *Proceedings of the National Academy of Sciences*, 109(15):5868–5873, 2012.
- [61] H. E. Egilmez, E. Pavez, and A. Ortega. Graph learning from data under laplacian and structural constraints. *IEEE J. Sel. Topics Signal Process*, 2017.
- [62] H. J. van Waarde, P. Tesi, and M. K. Camlibel. Topology reconstruction of dynamical networks via constrained lyapunov equations. *arXiv preprint arXiv:1706.09709*, 2017.
- [63] S. Hassan-Moghaddam and M. R. Jovanovic. An interior point method for growing connected resistive networks. In *Proceedings of the 2015 American Control Conference*, pages 1223–1228, Chicago, IL, 2015.
- [64] S. Sojoudi and J. Doyle. On the efficacy of computational methods for learning graphical models. submitted.
- [65] R. Dai and M. Mesbahi. Optimal topology design for dynamic networks. In *50th IEEE Conference on Decision and Control*, pages 1280–1285, 2011.
- [66] S. Hassan-Moghaddam and M. R. Jovanovic. Customized algorithms for growing connected resistive networks. In *Proceedings of the 10th IFAC Symposium on Nonlinear Control Systems*, pages 986–991, Monterey, CA, 2016.
- [67] Donatello Materassi and Giacomo Innocenti. Topological identification in networks of dynamical systems. *IEEE Trans. Automat. Control*, 55(8):1860–1871, 2010.
- [68] N. K. Dhingra, F. Lin, M. Fardad, and M. R. Jovanovic. On identifying sparse representations of consensus networks. In *Preprints of the 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 305–310, Santa Barbara, CA, 2012.

- [69] Milad Siami and Nader Motee. Network sparsification with guaranteed systemic performance measures. In *Preprints of the 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, volume 48, pages 246–251, 2015.
- [70] George F Young, Luca Scardovi, Andrea Cavagna, Irene Giardina, and Naomi E Leonard. Starling flock networks manage uncertainty in consensus at low cost. *PLoS Comput Biol*, 9(1):e1002894, 2013.
- [71] M. R. Jovanovic and N. K. Dhingra. Controller architectures: tradeoffs between performance and structure. *Eur. J. Control*, 30:76–91, July 2016.
- [72] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multiagent optimization. *IEEE Trans. on Automat. Control*, 54(1):48–61, 2009.
- [73] Jing Wang and Nicola Elia. A control perspective for centralized and distributed convex optimization. In *Proceedings of the 50th IEEE Conference on Decision and Control and the 10th European Control Conference*, pages 3800–3805, 2011.
- [74] P. Latafat, L. Stella, and P. Patrinos. New primal-dual proximal algorithm for distributed optimization. In *Proceedings of the 55th IEEE Conference on Decision and Control*, pages 1959–1964, 2016.
- [75] P. Latafat, N. Freris, and P. Patrinos. A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization. *IEEE Trans. Automat. Control*, 2019. doi:10.1109/TAC.2019.2906924.
- [76] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.
- [77] K. J. Arrow, L. Hurwicz, H. Uzawa, and H. B. Chenery. Studies in linear and non-linear programming. 1958.
- [78] D. Feijer and F. Paganini. Stability of primal–dual gradient dynamics and applications to network optimization. *Automatica*, 46(12):1974–1981, 2010.
- [79] Ashish Cherukuri, Enrique Mallada, and Jorge Cortés. Asymptotic convergence of constrained primal–dual dynamics. *Syst. Control Lett.*, 87:10–15, 2016.

- [80] A. Cherukuri, E. Mallada, S. Low, and J. Cortes. The role of convexity on saddle-point dynamics: Lyapunov function and robustness. *IEEE Trans. Automat. Control*, 63(8):2449–2464, 2018.
- [81] N. K. Dhingra, S. Z. Khong, and M. R. Jovanovic. The proximal augmented lagrangian method for nonsmooth composite optimization. *IEEE Trans. Automat. Control*, 64(7):2861–2868, July 2019.
- [82] G. Qu and N. Li. On the exponential stability of primal-dual gradient dynamics. *IEEE Control Syst. Lett.*, 3(1):43–48, 2018.
- [83] W. Su, S. Boyd, and E. Candes. A differential equation for modeling Nesterov’s accelerated gradient method: Theory and insights. *J. Mach. Learn. Res.*, 17:1–43, 2016.
- [84] A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *Proc. Natl. Acad. Sci.*, 113(47):E7351–E7358, 2016.
- [85] G. França, D. Robinson, and R. Vidal. ADMM and accelerated ADMM as continuous dynamical systems. 2018. arXiv:1805.06579.
- [86] B. Shi, S. Du, M. I. Jordan, and W. Su. Understanding the acceleration phenomenon via high-resolution differential equations. 2018. arXiv:1810.08907.
- [87] M. Muehlebach and M. I. Jordan. A dynamical systems perspective on Nesterov acceleration. 2019. arXiv:1905.07436.
- [88] J. I. Poveda and N. Li. Inducing uniform asymptotic stability in time-varying accelerated optimization dynamics via hybrid regularization. 2019. arXiv:1905.12110.
- [89] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Automat. Control*, 54:48, 2009.
- [90] D. Feijer and F. Paganini. Stability of primal-dual gradient dynamics and applications to network optimization. *Automatica*, 46(12):1974–1981, 2010.

- [91] A. Cherukuri, E. Mallada, S. Low, and J. Cortés. The role of convexity in saddle-point dynamics: Lyapunov function and robustness. *IEEE Trans. Automat. Control*, 63(8):2449–2464, 2017.
- [92] A. Brown and M. Bartholomew-Biggs. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. *J. Optimiz. Theory App.*, 62(2):211–224, 1989.
- [93] J. Schropp and I. Singer. A dynamical systems approach to constrained minimization. *Numer. Func. Anal. Opt.*, 21(3-4):537–551, 2000.
- [94] J. Zhang, A. Mokhtari, S. Sra, and A. Jadbabaie. Direct Runge-Kutta discretization achieves acceleration. In *Advances in Neural Information Processing Systems*, pages 3900–3909, 2018.
- [95] B. Hu, P. Seiler, and A. Rantzer. A unified analysis of stochastic optimization methods using jump system theory and quadratic constraints. In *Proceedings of the 2017 Conference on Learning Theory*, pages 1157–1189, 2017.
- [96] B. Hu and L. Lessard. Dissipativity theory for Nesterov’s accelerated method. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1549–1557, 2017.
- [97] M. Fazlyab, A. Ribeiro, M. Morari, and V. M. Preciado. Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems. *SIAM J. Optim.*, 28(3):2654–2689, 2018.
- [98] S. Hassan-Moghaddam and M. R. Jovanovic. Distributed proximal augmented lagrangian method for nonsmooth composite optimization. In *Proceedings of the 2018 American Control Conference*, pages 2047–2052, Milwaukee, WI, 2018.
- [99] S. Hassan-Moghaddam and M. R. Jovanovic. On the exponential convergence rate of proximal gradient flow algorithms. In *Proceedings of the 57th IEEE Conference on Decision and Control*, pages 4246–4251, Miami, FL, 2018. (Invited paper).
- [100] D. Ding, B. Hu, N. K. Dhingra, and M. R. Jovanovic. An exponentially convergent primal-dual algorithm for nonsmooth composite minimization. In *Proceedings of*

- the 57th IEEE Conference on Decision and Control*, pages 4927–4932, Miami, FL, 2018.
- [101] J. Seidman, M. Fazlyab, V. Preciado, and G. Pappas. A control-theoretic approach to analysis and parameter selection of Douglas-Rachford splitting. 2019. arXiv:1903.11525.
  - [102] H. Mohammadi, M. Razaviyayn, and M. R. Jovanovic. Variance amplification of accelerated first-order algorithms for strongly convex quadratic optimization problems. In *Proceedings of the 57th IEEE Conference on Decision and Control*, pages 5753–5758, Miami, FL, 2018.
  - [103] H. Mohammadi, M. Razaviyayn, and M. R. Jovanovic. Performance of noisy nesterov’s accelerated method for strongly convex optimization problems. In *Proceedings of the 2019 American Control Conference*, pages 3426–3431, Philadelphia, PA, 2019.
  - [104] H. Mohammadi, M. Razaviyayn, and M. R. Jovanovic. Robustness of accelerated first-order algorithms for strongly convex optimization problems. *IEEE Trans. Automat. Control*, 2019. submitted; also arXiv:1905.11011.
  - [105] S. Michalowsky, C. Scherer, and C. Ebenbauer. Robust and structure exploiting optimization algorithms: An integral quadratic constraint approach. 2019. arXiv:1905.00279.
  - [106] S. Hassan-Moghaddam and M. R. Jovanovic. Topology design for stochastically-forced consensus networks. *IEEE Trans. Control Netw. Syst.*, 5(3):1075–1086, September 2018.
  - [107] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.
  - [108] N. K. Dhingra, S. Z. Khong, and M. R. Jovanovic. A second order primal-dual method for nonsmooth convex composite optimization. *IEEE Trans. Automat. Control*, 2017. conditionally accepted; also arXiv:1709.01610.



- [109] W. Shi, Q. Ling, G. Wu, and W. Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM J. Optim.*, 25(2):944–966, 2015.
- [110] B. Gharesifard and J. Cortés. Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Trans. Autom. Control*, 59(3):781–786, 2014.
- [111] S. Hassan-Moghaddam and M. R. Jovanovic. Distributed design of optimal structured feedback gains. In *Proceedings of the 56th IEEE Conference on Decision and Control*, pages 6586–6591, Melbourne, Australia, 2017.
- [112] M. Hong, Z. Q. Luo, and M. Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM J. Optimiz.*, 26(1):337–364, 2016.
- [113] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–124, 2011.
- [114] F. Lin, M. Fardad, and M. R. Jovanovic. Augmented lagrangian approach to design of structured optimal state feedback gains. *IEEE Trans. Automat. Control*, 56(12):2923–2929, December 2011.
- [115] N. Matni and V. Chandrasekaran. Regularization for design. *IEEE Trans. Automat. Control*, 61(12):3991–4006, 2016.
- [116] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu. On the sample complexity of the linear quadratic regulator. 2017. arXiv:1710.01688.
- [117] M. Fazel, R. Ge, S. M. Kakade, and M. Mesbahi. Global convergence of policy gradient methods for linearized control problems. 2018. arXiv:1801.05039.
- [118] S. Schuler, P. Li, J. Lam, and F. Allgöwer. Design of structured dynamic output-feedback controllers for interconnected systems. *International Journal of Control*, 2011. doi:10.1080/00207179.2011.634029.
- [119] S. Hassan-Moghaddam, X. Wu, and M. R. Jovanovic. Edge addition in directed consensus networks. In *Proceedings of the 2017 American Control Conference*, pages 5592–5597, Seattle, WA, 2017.

- [120] C. Conte, C. N. Jones, M. Morari, and M. N. Zeilinger. Distributed synthesis and stability of cooperative distributed model predictive control for linear systems. *Automatica*, 69:117–125, 2016.
- [121] S. Hassan-Moghaddam and M. R. Jovanovic. Proximal gradient flow and douglas-rachford splitting dynamics: global exponential stability via integral quadratic constraints. *Automatica*, 2019. submitted; also arXiv:1908.09043.
- [122] S. Hassan-Moghaddam, M. R. Jovanovic, and S. Meyn. Data-driven proximal algorithms for the design of structured optimal feedback gains. In *Proceedings of the 2019 American Control Conference*, pages 5846–5850, Philadelphia, PA, 2019.
- [123] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal Newton-type methods for minimizing composite functions. *SIAM J. Optim.*, 24(3):1420–1443, 2014.
- [124] Cho-Jui Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. QUIC: Quadratic approximation for sparse inverse covariance estimation. *J. Mach. Learn. Res.*, 15:2911–2947, 2014.
- [125] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted  $\ell_1$  minimization. *J. Fourier Anal. Appl.*, 14:877–905, 2008.
- [126] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [127] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, 1988.
- [128] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Math. Prog.*, 117(1-2):387–423, 2009.
- [129] Yu-Hong Dai and Roger Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numerische Mathematik*, 100(1):21–47, 2005.
- [130] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.*, 57(7):2479–2493, 2009.

- [131] Julian J McAuley and Jure Leskovec. Learning to discover social circles in ego networks. In *Adv. Neural Inf. Process. Syst.*, pages 539–547, 2012.
- [132] M. Fardad, X. Zhang, F. Lin, and M. R. Jovanovic. On the properties of optimal weak links in consensus networks. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 2124–2129, Los Angeles, CA, 2014. (Invited paper).
- [133] X. Wu and M. R. Jovanovic. Sparsity-promoting optimal control of consensus and synchronization networks. In *Proceedings of the 2014 American Control Conference*, pages 2948–2953, Portland, OR, 2014.
- [134] T.H. Summers, Iman Shames, J. Lygeros, and F. Dörfler. Topology Design for Optimal Network Coherence. In *Proceedings of the 14th European Control Conference*, July 2015.
- [135] T. Rautert and E. W. Sachs. Computational design of optimal output feedback controllers. *SIAM J. Optim.*, 7(3):837–852, 1997.
- [136] M. Hong, Z. Q. Luo, and M. Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM J OPTIMIZ*, 26(1):337–364, 2016.
- [137] M Fortin and R. Glowinski. *Augmented Lagrangian method: Application to the numerical solution of boundary-value problems*. North-Holland, 1983.
- [138] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*. SIAM, 1989.
- [139] Katya Scheinberg, Shiqian Ma, and Donald Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In *NIPS*, pages 2101–2109, 2010.
- [140] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- [141] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d’Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. Mach. Learn. Res.*, 9:485–516, 2008.

- [142] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.
- [143] J. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse gaussians. *arXiv preprint arXiv:1206.3249*, 2012.
- [144] Cho-Jui Hsieh, Inderjit S Dhillon, Pradeep K Ravikumar, and Mátyás A Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. In *Adv. Neural Inf. Process. Syst.*, pages 2330–2338, 2011.
- [145] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494, 2001.
- [146] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for  $\ell_1$  regularized loss minimization. *J. Mach. Learn. Res.*, 12:1865–1892, 2011.
- [147] Ankan Saha and Ambuj Tewari. On the non-asymptotic convergence of cyclic coordinate descent methods. *SIAM J. Optim.*, 23(1):576–601, 2013.
- [148] Benjamin Rolfs, Bala Rajaratnam, Dominique Guillot, Ian Wong, and Arian Maleki. Iterative thresholding algorithm for sparse inverse covariance estimation. In *NIPS*, pages 1574–1582, 2012.
- [149] Figen Oztoprak, Jorge Nocedal, Steven Rennie, and Peder A Olsen. Newton-like methods for sparse inverse covariance estimation. In *NIPS*, pages 755–763, 2012.
- [150] Onureena Banerjee, Laurent El Ghaoui, Alexandre d’Aspremont, and Georges Natsoulis. Convex optimization techniques for fitting sparse gaussian graphical models. In *Proceedings of the 23rd international conference on Machine learning*, pages 89–96, 2006.
- [151] Alexandre d’Aspremont, Onureena Banerjee, and Laurent El Ghaoui. First-order methods for sparse covariance selection. *SIAM J. Matrix Anal. Appl.*, 30(1):56–66, 2008.
- [152] Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, Pradeep K Ravikumar, and Russell Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In *NIPS*, pages 3165–3173, 2013.

- [153] N. Motee and A. Jadbabaie. Optimal control of spatially distributed systems. *IEEE Trans. Automat. Control*, 53(7):1616–1629, 2008.
- [154] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. 14(3):462–467, 1968.
- [155] M. P. van den Heuvel, C. J. Stam, R. S. Kahn, and H. R. H Pol. Efficiency of functional brain networks and intellectual performance. *J. Neurosci.*, 29(23):7619–7624, 2009.
- [156] S. Sojoudi and J. Doyle. Study of the brain functional network using synthetic data. In *Proceedings of the 52nd Annual Allerton Conference on Communication, Control, and Computing*, pages 350–357, 2014.
- [157] P. Patrinos, L. Stella, and A. Bemporad. Forward-backward truncated Newton methods for convex composite optimization. 2014. arXiv:1402.6655.
- [158] L. Stella, A. Themelis, and P. Patrinos. Forward–backward quasi-Newton methods for nonsmooth optimization problems. *Comput. Optim. Appl.*, 67(3):443–487, 2017.
- [159] A. Themelis, L. Stella, and P. Patrinos. Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone line-search algorithms. *SIAM J. Optim.*, 28(3):2274–2303, 2018.
- [160] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Trans. Autom. Control*, 42(6):819–830, 1997.
- [161] B. T. Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963.
- [162] J. Douglas and H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82(2):421–439, 1956.
- [163] A. Themelis and P. Patrinos. Douglas-Rachford splitting and ADMM for nonconvex optimization: tight convergence results. 2017. arXiv:1709.05747.

- [164] H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811, 2016.
- [165] Anders Rantzer. On the Kalman-Yakubovich-Popov lemma. *Systems & Control Letters*, 28(1):7 – 10, 1996.
- [166] P. Giselsson and S. Boyd. Linear convergence and metric selection for Douglas-Rachford splitting and ADMM. *IEEE Trans. Automat. Control*, 62(2):532–544, 2017.
- [167] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55(1-3):293–318, 1992.
- [168] D. Gabay. Applications of the method of multipliers to variational inequalities. In *Studies in Mathematics and its Applications*, volume 15, pages 299–331. 1983.
- [169] A. Cherukuri, B. Gharesifard, J., and Cortés. Saddle-point dynamics: conditions for asymptotic stability of saddle points. *SIAM J. Control Optim.*, 55(1):486–511, 2017.
- [170] Hassan K. Khalil. Nonlinear systems. *Prentice-Hall, New Jersey*, 2(5):5–1, 1996.
- [171] K. Mårtensson and A. Rantzer. Gradient methods for iterative distributed control synthesis. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 549–554, 2009.
- [172] K. Mårtensson and A. Rantzer. A scalable method for continuous-time distributed control synthesis. In *Proceedings of the 2012 American Control Conference*, pages 6308–6313, 2012.
- [173] N. K. Dhingra and M. R. Jovanovic. A method of multipliers algorithm for sparsity-promoting optimal control. In *Proceedings of the 2016 American Control Conference*, pages 1942–1947, Boston, MA, 2016. (Invited paper).

- [174] D. G. Luenberger. *Optimization by vector space methods*. John Wiley & Sons Inc., New York, 1969. Reprinted 1997.
- [175] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. 42(5):674–690, 1997.
- [176] S. P. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, 2007. Pre-publication edition available online.
- [177] P. G. Mehta and S. P. Meyn. Q-learning and Pontryagin’s minimum principle. In *Proceedings of the 48th IEEE Conference on Decision and Control*, pages 3598–3605, 2009.
- [178] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.
- [179] S. M. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems*, pages 1531–1538, 2002.
- [180] N. K. Dhingra and M. R. Jovanovic. Convex synthesis of symmetric modifications to linear systems. In *Proceedings of the 2015 American Control Conference*, pages 3583–3588, Chicago, IL, 2015.
- [181] D. P. Bertsekas. *Convex optimization algorithms*. Athena Scientific, 2015.
- [182] S. Kia, J. Cortés, and S. Martínez. Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, 55:254–264, 2015.





## Appendix A

### Proximal PL condition

The generalization of the PL condition to nonsmooth problems was introduced in [164] and is given by

$$\frac{1}{2} \mathcal{D}_g(x, L_f) \geq \kappa (F(x) - F^\star) \quad (\text{A.1})$$

where  $\kappa$  is a positive constant,  $L_f$  is the Lipschitz constant of  $\nabla f$ , and  $\mathcal{D}_g(x, \alpha)$  is given by

$$-2\alpha \min_y \left( \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2} \|y - x\|_2^2 + g(y) - g(x) \right). \quad (\text{A.2})$$

Herein, we show that if proximal PL condition (A.1) holds, there is a lower bound given by (6.19) on the norm of the generalized gradient map  $G_\mu(x)$ . For  $\mu \in (0, 1/L_f)$ ,  $D_g(x, 1/\mu) \geq D_g(x, L_f)$ , and, thus, inequality (A.1) also holds for  $D_g(x, 1/\mu)$ . Moreover, from the definition of  $D_g(x, \alpha)$  given by (A.2), it follows that

$$D_g(x, 1/\mu) = \frac{2}{\mu} (F(x) - F_\mu(x))$$

where  $F := f + g$  and  $F_\mu$  is the associated FB envelope. Substituting this expression for  $D_g(x, 1/\mu)$  to (A.1) yields,

$$\frac{1}{\mu} (F(x) - F_\mu(x)) \geq \kappa (F(x) - F^\star). \quad (\text{A.3})$$

From (6.5), the smooth part of the objective function  $f$  can be written as

$$\begin{aligned} f(x) &= F_\mu(x) - g(\mathbf{prox}_{\mu g}(x - \mu \nabla f(x))) + \\ &\quad \mu \langle \nabla f(x), G_\mu(x) \rangle - \frac{\mu}{2} \|G_\mu(x)\|_2^2 \end{aligned}$$

and substituting this expression for  $f$  to both sides of (A.3) yields

$$\begin{aligned} \frac{(\mu\kappa-1)}{2} \|G_\mu(x)\|_2^2 &\geq \kappa(F_\mu(x) - F^\star) + \frac{(\mu\kappa-1)}{\mu} g(x) - \\ &\quad \frac{(\mu\kappa-1)}{\mu} g(\mathbf{prox}_{\mu g}(x - \mu \nabla f(x))) + \\ &\quad (\mu\kappa - 1) \langle \nabla f(x), G_\mu(x) \rangle. \end{aligned} \tag{A.4}$$

Since

$$G_\mu(x) - \nabla f(x) \in \partial g(x)$$

the subgradient inequality (6.18) implies

$$\begin{aligned} 0 \leq \mu \|G_\mu(x)\|_2^2 &\leq g(x) - g(\mathbf{prox}_{\mu g}(x - \mu \nabla f(x))) + \\ &\quad \mu \langle \nabla f(x), G_\mu(x) \rangle. \end{aligned} \tag{A.5}$$

Thus, if  $\mu\kappa - 1 > 0$ , inequality (A.4) can be written as

$$\frac{(\mu\kappa-1)}{2} \|G_\mu(x)\|_2^2 \geq \kappa(F_\mu(x) - F^\star).$$

Moreover, if  $\mu\kappa - 1 < 0$ , let  $\alpha := -(\mu\kappa - 1) \geq 0$ . From (A.4) and (A.5), we have

$$\frac{\alpha}{2} \|G_\mu(x)\|_2^2 \geq \kappa(F_\mu(x) - F^\star).$$

Thus, in both cases the inequality holds for  $\alpha = |\mu\kappa - 1|$ . Furthermore, it can be shown that [157],

$$F^\star = F_\mu^\star \text{ and } \operatorname{argmin} F = \operatorname{argmin} F_\mu.$$

Thus,  $F^\star$  can be substituted by  $F_\mu^\star$  and we have

$$\|G_\mu(x)\|_2^2 \geq \gamma(F_\mu(x) - F_\mu^\star)$$

where  $\gamma := 2\kappa/\alpha$ .

**A.1 Pointwise quadratic inequality** Herein, we provide a pointwise quadratic inequality that characterizes the nonlinear map  $u(x) = \mathbf{prox}_{\mu g}(x - \mu \nabla f(x))$  in (6.24b). Since the proximal operator of  $g$  is firmly nonexpansive [76], it is Lipschitz continuous with parameter 1 and we have

$$\|u - \hat{u}\|_2^2 \leq \|(x - \mu \nabla f(x)) - (\hat{x} - \mu \nabla f(\hat{x}))\|_2^2. \quad (\text{A.6})$$

Expanding the right-hand-side of (A.6) yields,

$$\begin{aligned} \|u - \hat{u}\|_2^2 \leq & \|x - \hat{x}\|_2^2 + \mu^2 \|\nabla f(x) - \nabla f(\hat{x})\|_2^2 - \\ & 2\mu \langle x - \hat{x}, \nabla f(x) - \nabla f(\hat{x}) \rangle \end{aligned}$$

and utilizing inequality (6.17) for an  $m_f$ -strongly convex function  $f$  with an  $L_f$ -Lipschitz continuous gradient, the last inequality can be further simplified as,

$$\begin{aligned} \|u - \hat{u}\|_2^2 \leq & \left(1 - \frac{2\mu m_f L_f}{L_f + m_f}\right) \|x - \hat{x}\|_2^2 + \\ & \left(\mu^2 - \frac{2\mu}{L_f + m_f}\right) \|\nabla f(x) - \nabla f(\hat{x})\|_2^2. \end{aligned} \quad (\text{A.7})$$

Now, depending on the sign of  $\mu - 2/(L_f + m_f)$  either (6.15) or (6.16) can be used to upper bound the second term on the right-hand-side of (A.7) and obtain

$$\|u - \hat{u}\|_2^2 \leq \max \{(1 - \mu L_f)^2, (1 - \mu m_f)^2\} \|x - \hat{x}\|_2^2. \quad (\text{A.8})$$

Choosing  $\eta$  given by (6.26) completes the proof of Lemma 7. We note that this contraction property has been also exploited in [97, 181].

## Appendix B

# On the exponential convergence rate of proximal gradient flow algorithms

Herein, we describe a primal-descent dual-ascent gradient flow dynamics based on proximal augmented Lagrangian (6.20) can be used for distributed optimization. We provide a distributed implementation and prove global exponential stability in the presence of strong convexity.

**B.0.1 Distributed optimization** Let us consider the unconstrained optimization problem,

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^n f_i(x)$$

where  $x \in \mathbb{R}^n$  is the optimization variable and  $f := \sum_i f_i$  is a strongly convex objective function. It is desired to solve this problem over an undirected connected network with the incidence matrix  $E^T$  and the graph Laplacian  $L := E^T E$ . To accomplish this objective, we reformulate it as,

$$\underset{x}{\text{minimize}} \quad \sum_{i=1}^n f_i(x_i) + g(Ex) \tag{B.1a}$$

where  $x := [x_1 \cdots x_n]^T$  and  $g(Ex)$  is an indicator function associated with the equality constraint  $Ex = 0$ ,

$$g(Ex) = \begin{cases} 0, & Ex = 0, \\ \infty, & \text{otherwise.} \end{cases} \quad (\text{B.1b})$$

This constraint is introduced to ensure asymptotic agreement between the node values  $x_i(t) \in \mathbb{R}$ .

As demonstrated in [108], the primal-descent dual-ascent gradient flow dynamics based on the proximal augmented Lagrangian (6.20) can be used to solve this problem. The resulting gradient flow dynamics are given by,

$$\begin{aligned} \dot{x} &= -\nabla f(x) - \frac{1}{\mu} Lx - \tilde{y} \\ \dot{\tilde{y}} &= Lx, \end{aligned} \quad (\text{B.2})$$

where  $L = E^T E$  is the Laplacian matrix of the underlying communication graph between neighboring nodes and the vector  $\tilde{y} := E^T y$  belongs to the orthogonal complement of the vector of all ones. This setup is well-suited for distributed implementation in which each node only shares its state  $x_i$  with its neighbors and maintains the corresponding dual variable  $\tilde{y}_i$ . A Lyapunov-based argument was used in [182] to prove the exponential convergence of (B.2). Herein, we provide an alternative proof that utilizes the theory of IQCs to establish global exponential stability of (B.2) under the condition that  $\mathbf{1}^T \tilde{y}(0) = 0$ .

**Assumption 5.** *Let the differentiable part  $f := \sum_i f_i(x)$  of the objective function in (B.1) be strongly convex with parameter  $m_f$ , let  $\nabla f$  be Lipschitz continuous with parameter  $L_f$ , let the regularization function  $g$  be proper, lower semicontinuous, and convex, let  $E^T$  be incidence matrix of a connected undirected network, and let  $\mathbf{1}^T \tilde{y}(0) = 0$  in (B.2).*

From Assumption 5 it follows that the graph Laplacian  $L := E^T E$  is a positive semidefinite matrix with one zero eigenvalue. Thus, it can be decomposed as

$$L = V \Lambda V^T = \begin{bmatrix} U & \frac{1}{n} \mathbf{1} \end{bmatrix} \begin{bmatrix} \Lambda_0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U^T \\ \frac{1}{n} \mathbf{1}^T \end{bmatrix}$$

where the columns of  $V$  are the eigenvectors of  $L$ ,  $\Lambda_0$  is a diagonal matrix of the nonzero eigenvalues of  $L$ , and the matrix  $U$  satisfies,

$$U^T U = I, \quad U^T \mathbf{1} = 0, \quad U U^T = I - \frac{1}{n} \mathbf{1} \mathbf{1}^T. \quad (\text{B.3})$$

By introducing a change of variables

$$\begin{bmatrix} \psi \\ \bar{x} \end{bmatrix} = \begin{bmatrix} U^T x \\ \frac{1}{n} \mathbf{1}^T x \end{bmatrix}, \quad \begin{bmatrix} \phi \\ \bar{y} \end{bmatrix} = \begin{bmatrix} U^T \tilde{y} \\ \frac{1}{n} \mathbf{1}^T \tilde{y} \end{bmatrix}$$

with  $\psi \in \mathbb{R}^{n-1}$  and  $\phi \in \mathbb{R}^{n-1}$ , (B.2) can be written as

$$\begin{bmatrix} \dot{\psi} \\ \dot{\bar{x}} \\ \dot{\phi} \\ \dot{\bar{y}} \end{bmatrix} = \begin{bmatrix} -(\frac{1}{\mu} \Lambda_0 + m_f I) \psi - \phi - U^T u \\ -\frac{1}{n} \mathbf{1}^T u - m_f \bar{x} - \bar{y} \\ \Lambda_0 \psi \\ 0 \end{bmatrix} \quad (\text{B.4})$$

where  $u = \nabla f(x) - m_f x$ , and from the properties (B.3) of the matrix  $U$  we have,

$$x = \begin{bmatrix} U & \mathbf{1} \end{bmatrix} \begin{bmatrix} \psi \\ \bar{x} \end{bmatrix}. \quad (\text{B.5})$$

By choosing  $\bar{y}(0) = 0$ , we have  $\bar{y} \equiv 0$ . Thus, the  $\bar{y}$ -dynamics can be eliminated from (B.4), which yields

$$\begin{aligned} \dot{w} &= A w + B u \\ \xi &= C w \\ u &= \nabla f(\xi) - m_f \xi. \end{aligned} \quad (\text{B.6a})$$

Here,  $w := [\psi^T \ \bar{x} \ \phi^T]^T$ ,  $\xi := x$ ,

$$\begin{aligned} A &= \begin{bmatrix} -(\frac{1}{\mu}\Lambda_0 + m_f I) & 0 & -I \\ 0 & -m_f & 0 \\ \Lambda_0 & 0 & 0 \end{bmatrix} \\ B &= \begin{bmatrix} -U^T \\ -\frac{1}{n}\mathbf{1}^T \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} U & \mathbf{1} & 0 \end{bmatrix} \end{aligned} \quad (\text{B.6b})$$

and the corresponding transfer function is

$$H(s) = - \begin{bmatrix} U & \mathbf{1} \end{bmatrix} \begin{bmatrix} H_1(s) & 0 \\ 0 & \frac{1}{s + m_f} \end{bmatrix} \begin{bmatrix} U^T \\ \frac{1}{n}\mathbf{1}^T \end{bmatrix} \quad (\text{B.6c})$$

where

$$H_1(s) = \text{diag} \left( \frac{s}{s^2 + (\lambda_i/\mu + m_f)s + \lambda_i} \right).$$

Furthermore, under Assumption 5 with  $u = \nabla f(\xi) - m_f \xi$ , for any  $\xi$  and  $\hat{\xi} \in \mathbb{R}^n$ , we have [31],

$$\begin{bmatrix} \xi - \hat{\xi} \\ u - \hat{u} \end{bmatrix}^T \begin{bmatrix} 0 & (L_f - m_f)I \\ (L_f - m_f)I & -2I \end{bmatrix} \begin{bmatrix} \xi - \hat{\xi} \\ u - \hat{u} \end{bmatrix} \succeq 0.$$

We now employ the KYP lemma to establish global exponential stability of (B.6a).

**Theorem 16.** *Let Assumption 5 hold. Then proximal gradient flow dynamics (B.6a) are globally exponentially stable, i.e., there is  $\tau, 0 < \rho < m_f$  such that,*

$$\|w(t) - w^*\| \leq \tau e^{-\rho t} \|w(0) - w^*\|$$

.

*Proof.* The KYP lemma implies global exponential stability if

$$\begin{bmatrix} H_\rho(j\omega) \\ I \end{bmatrix}^* \Pi \begin{bmatrix} H_\rho(j\omega) \\ I \end{bmatrix} \prec 0, \quad \forall \omega \in \mathbb{R} \quad (\text{B.7})$$

where

$$\Pi = \begin{bmatrix} 0 & (L_f - m_f)I \\ (L_f - m_f)I & -2I \end{bmatrix}$$

$$H_\rho(j\omega) = H(j\omega - \rho).$$

It is easy to show that (B.7) holds for all  $\omega \in \mathbb{R}$  if

$$2I - (L_f - m_f)(H_\rho^*(j\omega) + H_\rho(j\omega)) \succ 0.$$

This condition yields a decoupled family of inequalities,

$$\omega^2 + (m_f - \rho)^2 + (L_f - m_f)(m_f - \rho) > 0 \quad (\text{B.8})$$

$$(\omega^2 - b_i(\rho))^2 + c_i(\rho)\omega^2 + d_i(\rho) > 0 \quad (\text{B.9})$$

which have to hold for all  $\omega \in \mathbb{R}$  and for  $i = 1, \dots, n-1$ . Condition (B.8) clearly holds if  $\rho \in (0, m_f)$ . On the other hand, checking (B.9) amounts to checking a decoupled family of quadratic inequalities in  $\omega^2$  where  $b_i(\rho)$ ,  $c_i(\rho)$ , and  $d_i(\rho)$  are parameters that depend on  $\mu$ ,  $L_f$ ,  $m_f$ ,  $\lambda_i$ , and  $\rho$ . At  $\rho = 0$ , these are given by

$$b_i(0) = \lambda_i/\mu, \quad c_i(0) = (\lambda_i/\mu + m_f)(\lambda_i/\mu + L_f), \quad d_i(0) = 0.$$

Positivity of  $b_i(0)$  and  $c_i(0)$  for each  $i$  and continuity of  $b_i(\rho)$ ,  $c_i(\rho)$ , and  $d_i(\rho)$  with respect to  $\rho$  imply the existence of  $\rho > 0$  that guarantees (B.9) for each  $\omega \in \mathbb{R}$  and each  $i = 1, \dots, n-1$ , which completes the proof.

□