UNIVERSITY OF CALIFORNIA SAN DIEGO

Constrained Network Optimization: Algorithms, and Applications in Frequency Regulation

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Priyank Srivastava

Committee in charge:

Professor Jorge Cortés, Chair
Professor Maurício de Oliveira
Professor Jan Kleissl
Professor Melvin Leok
Professor Behrouz Touri

2021

The dissertation of Priyank Srivastava is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

DEDICATION

To my family and prabhu Shri Ram.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor Prof. Jorge Cortés, for his constant support during the past five years. To be honest, Jorge, I do not think writing a few sentences here does proper justice to how awesome of an advisor you are! Your dedication towards your students is commendable. Your passion towards research and mentoring helped me grow in more ways than I imagined at the beginning of this journey. I would always cherish the time we spent together in our meetings. Thanks for your patience while educating me and turning my crappy drafts to eloquent papers. Apart from the research, I am also thankful for your support on other miscellaneous stuff. In short, thank you so much for everything. I sincerely hope that we keep working together in the future.

Thanks to my dissertation committee professors: Prof. Maurício de Oliveira, Prof. Jan Kleissl, Prof. Melvin Leok, and Prof. Behrouz Touri for taking time out of their schedule and providing me with suggestions. Special thanks to Jan for making the collaboration on the ARPA-e and DERConnect projects so pleasant.

My sincere thanks to all the professors at UCSD whose teachings helped in shaping the dissertation implicitly. Thanks to my professors at Indian Institute of Technology Delhi for their encouragement and support during my Ph.D. applications.

Over the years, I have been fortunate enough to work with some amazing people. I thank Prof. Sonia Martínez for all the fruitful discussions we had. Thanks to Prof. Patricia Hidalgo-Gonzalez for all the technical and non-technical discussions. Special thanks to Dr. Guido Cavraro for being so proactive and always bringing new viewpoints to the discussion, and Dr. Chin-Yao Chang for helping me in the early stages of my Ph.D. Thanks to Manasa and Dr. Hamed Valizadeh

Haghi for their help on the ARPA-e and DERConnect projects.

I would also like to thank current and former members of our research group: Aamodh, Parth, Prasad, Miguel, Dan, Dimitris, Zhichao, Eduardo, Tor, Erfan, Yifu, Ahmed, Shenyu, Aaron, Scott, Masih, and Vishaal, for the all the technical discussions and fun conversations. Special thanks to my former colleague Prof. Ashish Cherukuri, who has guided me on more than one occasion, and my friend Pio Ong, who is up for discussions at any time of the day. Pio, we will definitely write a paper together!

Thanks to my friends Ankit Dubey and Debojyoti Biswas for their help during my initial days in the US. Special thanks to my best friend Krishan Kant Bhalla for always being there for me, and my friend Chetan Balaji Nauduri who has helped me selflessly in many instances.

Finally, I would like to thank my parents for their wholehearted support and patience. Thanks to my brother Piyoosh for providing me the family support over the last couple of years, and my partner Madhurima for her unconditional love and always prioritizing my achievements over hers.

The research of the thesis was supported generously by the ARPA-e NODES program, Cooperative Agreement DE-AR0000695, NSF Awards ECCS-1917177 and ECCS-1947050, and National Renewable Energy Laboratory (NREL) under Contract DE-AC36-08GO28308.

Chapter 3, in part, is a reprint of the material [SC22] as it appears in 'Solving linear equations with separable problem data over directed networks' by P. Srivastava and J. Cortés, in the IEEE Control Systems Letters, 2022, as well as [SC21b] where it appears as 'Network optimization via smooth exact penalty functions enabled by distributed gradient computation' by P. Srivastava and J. Cortés in the IEEE Transactions on Control of Network Systems, 2021. The dissertation author was the primary investigator and author of these papers.

Chapter 4, in part, is a reprint of the material [SC21b] as it appears in 'Network optimization via smooth exact penalty functions enabled by distributed gradient computation' by P. Srivastava and J. Cortés, in the IEEE Transactions on Control of Network Systems, 2021, as well as [SC18] where it appears as 'Distributed algorithm via continuously differentiable exact penalty method for network optimization' by P. Srivastava and J. Cortés in the proceedings of the 2018 IEEE Conference on Decision and Control. The dissertation author was the primary investigator and author of these papers.

Chapter 5, in full, is a reprint of the material [SC21a] as it appears in 'Nesterov acceleration for equality-constrained convex optimization via continuously differentiable penalty functions' by P. Srivastava and J. Cortés, in the IEEE Control Systems Letters, 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 6, in full, has been submitted for publication of the material [SCC21a] as it may appear as 'Agent-supervisor coordination for decentralized event-triggered coordination' by P. Srivastava, Guido Cavraro, and J. Cortés, in the IEEE Control Systems Letters, 2021. The dissertation author was the primary investigator and author of this paper.

Chapter 7, in part, is a reprint of the material [SCC21b] conditionally accepted for publication as 'Enabling DER participation in frequency regulation markets' by P. Srivastava, Chin-Yao Chang, and J. Cortés, in the IEEE Transactions on Control Systems Technology, 2021, as well as [SCC18] where it appears as 'Participation of microgrids in frequency regulation markets' by P. Srivastava, Chin-Yao Chang, and J. Cortés in the American Control Conference, 2018. The dissertation author was the primary investigator and author of these papers.

Chapter 8, in full, is currently being prepared for submission for publication of the material as 'Frequency regulation via simultaneously stabilizing data-driven controller' by P. Srivastava, P.

Hidalgo-Gonzalez, and J. Cortés. The dissertation author was the primary investigator and author of this paper.

VITA

2012	Bachelor of Technology in Electrical Engineering, National Institute of Technology, Kurukshetra
2016	Master of Technology in Electrical Engineering (Control & Automation), Indian Institute of Technology Delhi
2017	Master of Science in Engineering Science (Mechanical Engineering), University of California San Diego
2021	Doctor of Philosophy in Engineering Science (Mechanical Engineering), University of California San Diego

PUBLICATIONS

Journal publications:

[1] P. Srivastava, G. Cavraro, and J. Cortés. Agent-supervisor coordination for decentralized event-triggered optimization. *IEEE Control Systems Letters*, 2021. Submitted

[2] P. Srivastava, C.-Y. Chang, and J. Cortés. Enabling DER participation in frequency regulation markets. *IEEE Transactions on Control Systems Technology*, 2021. Conditionally accepted

[3] P. Srivastava and J. Cortés. Solving linear equations with separable problem data over directed networks. *IEEE Control Systems Letters*, 6:596–601, 2022

[4] T. Anderson, M. Muralidharan, P. Srivastava, H. Valizadeh Haghi, J. Cortés, J. Kleissl, S. Martínez, and B. Washom. Frequency regulation with heterogeneous energy resources: A realization using distributed control. *IEEE Transactions on Smart Grid*, 12(5):4126–4136, 2021

[5] P. Srivastava and J. Cortés. Network optimization via smooth exact penalty functions enabled by distributed gradient computation. *IEEE Transactions on Control of Network Systems*, 2021. To appear

[6] P. Srivastava and J. Cortés. Nesterov acceleration for equality-constrained convex optimization via continuously differentiable penalty functions. *IEEE Control Systems Letters*, 5(2):415–420, 2021

Conference proceedings:

[7] P. Srivastava, G. Cavraro, and J. Cortés. Agent-supervisor coordination for decentralized event-triggered optimization. In *American Control Conference*, Atlanta, Georgia, June 2021. Submitted

[8] P. Srivastava and J. Cortés. Solving linear equations with separable problem data over directed networks. In *IEEE Conf. on Decision and Control*, Austin, TX, December 2021. To appear

[9] P. Srivastava and J. Cortés. Nesterov acceleration for equality-constrained convex optimization via continuously differentiable penalty functions. In *IEEE Conf. on Decision and Control*, Jeju Island, South Korea, December 2020

[10] P. Srivastava and J. Cortés. Distributed algorithm via continuously differentiable exact penalty method for network optimization. In *IEEE Conf. on Decision and Control*, pages 975–980, Miami Beach, FL, December 2018

[11] P. Srivastava, C.-Y. Chang, and J. Cortés. Participation of microgrids in frequency regulation markets. In *American Control Conference*, pages 3834–3839, Milwaukee, WI, May 2018

[12] P. Srivastava, S. Singh, and S. Janardhanan. Linear functional observers for unforced multi-output nonlinear systems. In *IFAC International Conference on Advances in Control and Optimization of Dynamical Systems*, volume 51, pages 708–712, Hyderabad, India, February 2018

[13] S. Singh, P. Srivastava, and S. Janardhanan. Adaptive higher order sliding mode control for nonlinear uncertain systems. In *IFAC International Conference on Advances in Control and Optimization of Dynamical Systems*, volume 51, pages 341–346, Hyderabad, India, February 2018

ABSTRACT OF THE DISSERTATION

Constrained Network Optimization: Algorithms, and Applications in Frequency Regulation

by

Priyank Srivastava

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California San Diego, 2021

Professor Jorge Cortés, Chair

Network optimization problems arise naturally as a way of encoding the coordination task entrusted to multi-agent systems deployed in many areas of engineering, including power, communication, transportation, and swarm robotics. The large-scale nature of these systems coupled with the intrinsic modularity in their structure due to the technological advances in communication, embedded computing, and parallel processing requires a shift from the traditional paradigm of centralized decision-making to a distributed one. This transition, which is essential to harness the true capabilities of modern cyberphysical systems, raises a number of noteworthy challenges as well as opportunities, and has sparked the development of solutions that scale with the number of agents,

provide plug-and-play capabilities, and are resilient against single points of failure. Motivated by these considerations, this thesis is a contribution to the growing body of work that deals with the synthesis and analysis of provably correct algorithmic solutions to structured network problems.

Specifically, the thesis is divided into two parts. The first part focuses on synthesizing algorithmic solutions for application-agnostic large-scale network problems. We consider constrained optimization problems where the global objective function is the aggregate of local objectives of the participating agents; the collective goal of the agents and the underlying interaction pattern among them define the constraints. Using continuously differentiable exact penalty functions and globally projected dynamical systems, we then propose privacy-preserving, scalable, accelerated and anytime algorithms to solve these optimization problems. The second part is application-oriented and deals with constrained optimization problems in the context of power systems. In particular, we focus on the utilization of distributed energy resources for frequency regulation in the modern grid. We design distributed time-invariant controllers stabilizing the time-varying power dynamics for primary frequency control, and develop meaningful abstractions for groups of distributed energy resources to participate in the secondary frequency control market.

Chapter 1

Introduction

A careful glance around makes one realize that we are surrounded by plentiful complex systems consisting of various interconnected subsystems or modules (referred to as *agents* in the following). For example, think about thousands of planets and billions of stars in the galaxy, billion of neurons in the brain, and billions of interacting individuals in the society. Owing to the recent technological advancements in digital systems, communication, and sensing, this modular structure which is inherent in natural, biological, and social systems, is now also at the heart of many engineered systems. Notable examples include power systems, transportation, communication, and swarm robotics. Each of these complex systems has an underlying intricate pattern encoding the interactions between its components, referred to as *network* in the following. Any global task entrusted to these systems requires proper coordination between different agents of the network. This naturally gives rise to constrained optimization problems where the objective function is the aggregate of the local objectives of all the agents, and the constraints capture their collective goal, the network model and the physical limitations of the agents. Due to the complex nature of these cyber-physical systems, often times, these optimization problems are not standalone and their solutions

serve as input to other layers in the control design.

Harnessing the true capabilities of network systems require us to shift from the traditional paradigm of centralized decision-making to a decentralized/distributed one. As Richard Feynman said about 40 years ago, if a problem is big enough and a lot of calculation needs to be done, parallel computation can speed things up enormously and this goes beyond scientific problems. One of the first implementations of this scheme was done in a machine known as the Cosmic Cube at Caltech, where there were a large number of computers working on different parts of a given problem and transferring information to each other as needed. Recent developments in embedded computing and communication have further sparked the development of distributed algorithmic solutions. In addition to the original motivation of increasing speeds with parallel processing, distributed solutions also have several features particularly advantageous to network systems. In particular, since the exchange of data is much sparser and segmented, privacy of the participation agents is naturally preserved, and devices dropping in and out of the system can be naturally compensated thereby providing plug-and-play capabilities. Moreover, due to the absence of a central processor, distributed algorithms are inherently robust to failures and malicious attacks. However, coming up with distributed solutions to a global problem often requires more than just modifying a centralized solution. Depending on the coupling between the agents and the problem at hand, in certain cases, it might even be necessary to start all over again. This is the reason that adoption of distributed algorithm often faces resistance from the system operators and industrial partners, necessitating the development of provably correct algorithmic solutions providing desired performance guarantees.

This thesis is specifically divided into two parts. The first part (Chapters 3-6) focuses on synthesizing algorithmic solutions for large-scale network problems. We consider constrained optimization problems without focusing precisely on any single application. In a sense, this

part is application-agnostic. The algorithm design approach employed in the thesis follows the works [Bro91, HM94], and uses a dynamical systems approach to characterize the convergence and robustness properties. The second part (Chapters 7-8) deals with constrained optimization problems appearing in power systems. In particular, we focus on the frequency regulation in the modern grid with increasing penetration of distributed energy resources (DERs).

1.1 Literature Review

1.1.1 Algorithms for Network Problems

The breadth of applications of network optimization [BT97a, RN05, WL09] has motivated a growing body of work that builds on consensus-based approaches to produce rich algorithmic designs with asymptotic convergence guarantees, see [Ned15, YYW+19] for comprehensive surveys. In this class of problems, each agent in the network maintains, communicates, and updates an estimate of the complete solution vector, whose dimension is independent of the network size. In contrast to this setting, in many applications the structure of the optimization problem lends itself to having instead each agent optimize over and communicate its own local variable. Considered collectively, these variables give rise to the solution vector. Distributed algorithms to address this setting fall under Lagrangian-based approaches that rely on primal-dual updates, e.g., [FP10, RC15, MZL17, BPC+11, XHC+17, AYS20] or unconstrained reformulations that employ non-smooth penalty functions [XB06, JJ09, CC15]. Our approach here is based on the exact reformulation of the original problem using continuously differentiable penalty functions [GP79, DG89, Luc92, Di 94]. The work [DG89] establishes, under appropriate regularity

conditions on the feasibility set, the complete equivalence between the solutions of the original constrained and the reformulated unconstrained optimization problems. The work [Luc92] proposes a continuously differentiable exact penalty function that relaxes some of the assumptions of [DG89]. Notably, the works on continuously differentiable exact penalty functions use centralized optimization algorithms because the computations involved in the definition of the unconstrained penalty function are of a centralized nature.

As we argue in Chapter 4, linear equations with separable data arise naturally when considering the distributed solution of exact penalty optimization problems, but the importance of solving linear algebraic equations is paramount anyway. Justifying their ubiquity, there is a vast and expanding literature to solve them in a distributed fashion, cf. [BT97a, MLM15, AMMH16] and references therein. Different algorithmic solutions exist depending on the assumptions about the information available to the individual agents. Most of the works consider the information structure where each agent knows some rows of the coefficient matrix and the constant vector. In those cases, the collective problem has a solution if and only if the individual equations are solvable. However in many applications, the structure of the problem is such that each agent has a full coefficient matrix and constant vector of its own. This setting appears frequently in distributed sensor fusion, where sensors are spatially distributed and they seek to build a global state estimate (e.g., about the location of a source or the position of a target) from local measurements, cf. [SOSM05a, XBL05]. The work [SOSM05a] relies on the positive definiteness of the individual matrices to compute the updates and prove stability. [XBL05] uses element-wise average consensus for the coefficient matrix as well as the constant vector, which does not scale with either the problem dimension or the network size, and is not desirable from a privacy standpoint. [LT18] also exploits the positive definite property of the individual matrices and requires the agents to know the state as well as the matri-

ces of the neighbors. [WM18a] proposes a distributed algorithm without any positive definiteness condition, but agents are allowed to converge to different solutions. Another important point to note is that all the aforementioned works rely on the communication graph being *undirected*. For directed networks, we use dynamic average consensus [SOSM05b, KSC+19] to estimate certain non-distributed terms in a gradient-based algorithm for the reformulated optimization problem. We also draw inspiration from [TG15, TG20] on distributed optimization to extend our treatment to deal with unbalanced networks.

Another well known issue in the context of constrained optimization is that of convergence speed. For unconstrained optimization, the accelerated gradient descent method proposed by Nesterov [Nes83] uses only first-order information combined with momentum terms [SBC16, SDSJ19] to achieve an optimal convergence rate, thereby avoiding the complexity associated with the second-order Newton method [BV04]. For simple set constraints where the projection of any point can be computed in closed form a generalization of Nesterov's method, for constrained convex optimization described in [Nes04]. However for general constraints, incorporating the momentum term saddle-point or primal-dual dynamics, see e.g., [Kos56, AHU58, CGC17, CMLC18] is not clear. When the saddle function is strongly convex-strongly concave, the primal-dual dynamics converges exponentially fast, see e.g., [CL12]. Recent work [CN19, QL19, DJ19, DH19] has explored the partial relaxation of the strong convexity requirement while retaining the exponential convergence rate. Another method with improved rate of convergence for constrained problems is accelerated mirror descent [KBB15] which, however necessitates the choice of an appropriate mirror map depending on the geometry of the problem and requires that each update solves a constrained optimization problem (which might be challenging itself). Some works [GR12, AO17, DKJ17] have sought to generalize Newton's method for equality constrained problems, designing second-order updates

that require the inversion of the Hessian matrix of the augmented Lagrangian. We employ the continuously differentiable penalty functions to reformulate the constrained convex optimization problem and identify sufficient conditions under which the unconstrained problem is also convex.

In addition to the peer-to-peer information exchange structure for networked systems considered above, an alternative architecture has agents send and receive information from a central entity or network supervisor. Such architecture is well suited for scenarios where the objective function of the optimization problem is the combination of a component separable amongst the agents and another one coupling all the agents' decision variables. Notable examples of applications exhibiting this structure include virtual power plants [SMT11, CBCZ20], where a central entity works as an aggregator enabling the participation of a cluster of distributed energy sources in the energy market; HVAC systems in intelligent buildings [KB13, WM18b], where a central processor manages the main air supply and helps coordinate a group of thermal zones equipped with thermostats and controllers regulating heated or cooled air input; multi-agent systems with access to the cloud [HNE17], which provides superior processing and storing capabilities; and sensor-actuator networks with a central computation node [MT11, ZSJ09] responsible for computing the control signal with measurements from distributed sensors. In these scenarios, there is a need to structure the interaction between individual agents and the network supervisor to design solutions that scale up. Event-triggered control, see e.g. [Tab07, HJT12, HFO$^+$17] and references therein, offers a framework to prescribe, in a principled way, when to use the available resources efficiently while still guaranteeing a desired quality of service in performing the intended task. A number of works [KCM15a, NGC19] have explored the use of event-triggered approaches for achieving network coordination tasks. Of particular relevance to our work are [MT11, BNP16, ALDJ15], which consider similar network architectures, and [OC21], which mixes continuous updates, computed

with the locally available information, with aperiodic updates, computed with the externally provided information. An issue in the context of event-triggered control with distributed triggering is the emergence of Zeno behavior. This is addressed in [MT11, BNP16] by using *time regularization*, i.e., preventing by design any update before certain dwell time has elapsed. In general, time regularization requires an offline computation with global information. A final body of work we build on for this part is that of continuous projected dynamical systems for optimization, cf. [FBM+94, XW00, Gao03].

1.1.2 Frequency Regulation from DERs

In power systems, any mismatch between electricity generation and consumption results in the deviation of frequency from its nominal value [Kun94]. As the frequency starts deviating, some generators respond automatically with control input proportional to the frequency deviation. This is known as primary control. After primary control starts actuating, secondary control mechanisms restore the frequency to its nominal value [EMK11]. With the increasing penetration of the DERs, the inertia of the grid is decreasing and time-varying [UBA14]. This can provoke an increase in the variation of frequency under abrupt changes in generation and demand. In order to model power dynamics taking into account the variability of inertia, it is necessary to utilize a new modeling framework for frequency dynamics that captures the time-varying nature of the dynamical system. The work in [HGCD+18] proposes to model power dynamics as a switched affine hybrid system. The work [HGHACT19a] proposes a learning-based frequency controller utilizing the local DERs for these time-varying dynamics under the assumption of all the nodes having the same inertia, and [HGHACT19b] extends this work by enhancing sparsity in the learned controller design. Al-

though the individual modes are stabilized in these works by using virtual inertia, stability of the switched system is not guaranteed.

In addition to primary control, DERs can also be effective for secondary control and provide power to the external grid. However, DERs are limited in size and might not meet the minimum size criteria specified by system operators to participate in the frequency regulation market. The recent Order No. 2222 [Fed20] by the U.S. Federal Energy Regulatory Commission (FERC) enables aggregators of DERs to participate in the energy markets and requires all Regional Transmission Organizations (RTOs) to revise their tariffs to establish DERs as a category of market participant. Order No. 755 [Fed11] requires RTOs to compensate energy resources based on the actual frequency regulation provided. The payment to resources comprises of two parts, the capacity and performance payments. The capacity payment compensates resources for their provision of regulation capacity. The performance payment reflects the accuracy of the tracking of the allocated regulation signal. The work [KM14] describes how different RTOs across the United States have implemented FERC Order 755 for participation of resources in frequency regulation market. In the literature on power networks and smart grid, some works have considered the possibility of obtaining frequency regulation services from collections of homogeneous loads such as electric vehicles (EVs) and thermostatically controlled loads (TCLs), cf. [MKC13, CPP, SHPV14]. The work [HDGP16] presents a method to model flexible loads as a virtual battery for providing frequency regulation. [RSR13] proposes the use of aggregators to integrate heterogeneous loads such as heat pumps, supermarket refrigerators and batteries present in industrial buildings to provide frequency regulation. The works [BPP16, BKP+18] describe the challenges that need to be overcome for providing frequency regulation by DERs for some European countries. The work [DGS+18] provides a framework to emulate virtual power plants (VPPs) via aggregations of DERs and provide regulation services tak-

ing into account the power flow constraints. [BAP+13] provides a dispatch strategy for an aggregate of ON/OFF devices to provide frequency regulation. In [HDGP17, CMC17, XUDGS17], work has been done in the context of microgrids to design mechanisms for optimally allocating a given signal among the DERs within the microgrid. [GAB18] proposes a distributed algorithm to minimize the aggregated cost while satisfying the local constraints and collective demand constraint at the aggregator. However, the aforementioned works assume that the allocated signal from the RTO is available to the aggregator. [MKBD16] applies machine learning to forecast the power capacity of VPPs. The work [WAT+16] provides a framework for optimal bidding and dispatch of multiple VPPs. [ZMS17] proposes the use of renewable energy aggregators to utilize small-scale distributed generators for frequency regulation services via forecasting the available power from individual resources. The work [CMK18] also uses forecasting to estimate the aggregate production from a wind and solar power-based VPP, and then uses the estimation to determine the optimal volume of reserves that can be provided to the system operator. A distributed algorithm for coordinating multiple aggregators to provide frequency regulation, without any consideration of cost, is proposed in [HCG+17]. Here, we focus on (i) participation of microgrids in frequency regulation markets operated by the RTO through the identification of appropriate bids and (ii) the coordination among RTO and aggregators to efficiently dis-aggregate the regulation signal amongst the aggregators. The actual tracking performance within the microgrid would depend on the physical condition of the resources. We have provided some results for this in [AMS+21] on experiments carried out on the University of California, San Diego (UCSD) microgrid. Our ensuing discussion pertains specifically to microgrid participation in frequency regulation markets. We assume that, if the microgrids also exchange energy with the bulk grid at slower time scales, e.g., for the day-ahead market, cf. [ZD16, ZD17], those commitments are known to the aggregators and taken into account

in their baseline generation profiles at the time of participation in the frequency regulation market.

1.2 Statement of Contributions

The main focus of the thesis is to develop algorithmic solutions for large-scale constrained optimization problems. Our contributions are structured in two blocks. We start with application-agnostic network problems, and propose privacy-preserving, scalable, accelerated and anytime algorithms to solve them via continuously differentiable exact penalty functions and globally projected dynamical systems. Next, we focus on the problem of frequency regulation with the increasing penetration of DERs, and develop meaningful abstractions of certain bidding quantities required by groups of DERs (referred to as *microgrids*) to participate in the secondary frequency control market and propose distributed algorithms to optimally disaggregate the frequency regulation signal amongst the microgrids. We also design decentralized time-invariant controllers stabilizing the switching power dynamics for primary frequency control. We describe our contributions for both the parts below.

1.2.1 Algorithms for Network Problems

We start in Chapter 3, with linear algebraic equations where the coefficient matrices and constant vector for the overall problem are given, respectively, by the summation of the individual agents' coefficient matrices and constant vectors. Our starting point is providing two exact reformulations of this problem based on whether the communication graph is undirected or directed. For the undirected case, propose an initialization-free distributed algorithm that converges to a solution of the linear equation. We establish the exponential convergence and characterize the input-to-state

stability properties of this algorithm. For the directed case, we first propose a centralized algorithm which works for weight-balanced networks and serves as a reference for the design of distributed algorithms. Using dynamic average consensus, we then propose a distributed algorithm that does not require the agent matrices to be positive definite, works for time-varying weight-balanced networks and is guaranteed to converge to a solution of the original problem exponentially fast. Building on the insights gained in establishing these results, we propose a distributed algorithm that is not limited to weight-balanced networks and is also guaranteed to converge to a solution of the linear equation exponentially fast.

Chapter 4 builds on the results of Chapter 3 and continuously differentiable exact penalty functions. We consider nonlinear programming problem with a separable objective function and locally coupled constraints and reformulate it as unconstrained optimization of a continuously differentiable exact penalty function. Motivated by enabling the computation of the gradient of this function by the network agents, we show that the calculation of certain non-distributed terms in the gradient can be formulated as solving appropriately defined systems of linear algebraic equations defined by separable data. Our next contribution is the design of the distributed algorithm that solves the original constrained optimization problem. This algorithm is based on following gradient descent of the penalty function while estimating the actual value of the gradient with the distributed strategy that solves systems of linear algebraic equations. We establish the convergence of the resulting interconnection and illustrate its performance in simulation, comparing it with alternative approaches. We end by noting that, since the proposed approach relies on the distributed computation of the gradient, the methodology can also be used for accelerated distributed optimization using Nesterov's method, something which we also illustrate in simulation.

Motivated by the promising simulation results of Nesterov's acceleration on the penalty

function, in Chapter 5, we seek to obtain theoretical guarantees for the same. We consider equality constrained convex optimization problems. We show via a counterexample that the unconstrained penalty function might not be convex for any value of the penalty parameter even if the original problem is convex. This motivates our study of sufficient conditions on the objective and constraint functions of the original problem for the unconstrained penalty function to be convex. Our results are based on analyzing the positive semi-definiteness of the Hessian of the penalty function. We provide explicit bounds below which, for any value of the penalty parameter, the penalty function is either convex or strongly convex on the domain, resp. Since the optimizers of the unconstrained convex penalty function are the same as the optimizers of the original problem, we deduce that the proposed Nesterov implementation solves the original constrained problem with an accelerated convergence rate starting from an arbitrary initial condition. Finally, we establish that Nesterov's algorithm applied to the penalty function renders the feasible set forward invariant. This, coupled with the fact that the penalty terms vanish on the feasible set, ensures that the accelerated convergence rate is also achieved from any feasible initialization.

In Chapter 6, we consider network optimization problems where the objective function is the sum of a component given by the summation of local costs and a coupling component whose evaluation requires knowledge of all the agents' decision variables. Individual agents rely on the network supervisor to obtain information about the coupling component and to solve the optimization problem. Our contributions are structured in two blocks, corresponding to unconstrained and constrained problems. For unconstrained systems, we build on the gradient descent dynamics; for constrained systems with separable constraints, we build on globally projected dynamical systems. For both cases, we design novel event-triggered agent-supervisor coordination algorithms where agents continuously employ their local information and resort to opportunistic interactions with

the supervisor for information about the coupling cost. The criterion for triggering employed by each agent depends only on locally available information, which makes the proposed approach privacy preserving. We show the monotonic decrease of the objective function and establish the existence of a minimum inter-event time, thus ruling out Zeno behavior (without the need for any time regularization) and ensuring asymptotic convergence to the optimizer. In the constrained case, we also show that the feasible set is positively invariant, thus guaranteeing anytime feasibility.

1.2.2 Frequency Regulation from DERs

In Chapter 7, we propose a hierarchical framework for the participation of aggregations of DERs in the frequency regulation market. We start by briefly reviewing the current practice of frequency regulation from individual resources, consisting of three stages: (i) market clearance, (ii) disaggregation of the regulation signal and (iii) real-time tracking of the regulation signal. Our first contribution is the identification of the limitations of current practice and the challenges that need to be overcome for integration of microgrids. Our second contribution is the identification of abstractions for the capacity, cost of generation, and ramp rates of a microgrid as a combination of the individual energy resources that compose it, along with a formal description of its convexity and monotonicity properties. Using chance constraints, we extend these abstractions to the case when the loads inside the microgrid do not remain constant for the regulation period. Equipped with these abstractions, a microgrid can submit bids to participate in the market clearance stage. Our third contribution is the reformulation of the RTO-DERP coordination problem to optimally disaggregate regulation signal amongst the microgrids and accompanying design of an algorithmic solution. Our proposed reformulation ensures feasibility. The proposed algorithm is distributed over directed

graphs with only one aggregator needing to know the required regulation, and is guaranteed to asymptotic converge to the desired optimizers. We conclude with simulation results based on the proposed abstractions of capacities, cost, and ramp rate and the RTO-DERP coordination algorithm on a reduced-order model of the University of California, San Diego (UCSD) microgrid.

Chapter 8 deals with the problem of primary control. Our starting point is the reformulation of the optimal learning-based controller problem such that its solution guarantees the stability of the switched affine system, thereby avoiding the non-optimality from adding virtual inertia a posteriori. We then establish the existence of an invariant controller stabilizing the switched system. Building on this, we then prove the existence of a decentralized common controller that stabilizes the switched system and each node just needs to know the value of its phase and frequency to implement it. Our results are not limited to networks where all the nodes have the same inertia and work for heterogeneous networks.

1.3 Organization

This thesis is organized as follows. Chapter 2 introduces the notations and basic concepts on graph theory, convex analysis, probability theory, dynamic average consensus, constrained optimization, and event-triggered control. Chapter 3 proposes distributed algorithms to solve linear equations with separable data. Based on continuously differentiable exact penalty functions introduced in Chapter 2, we propose a novel distributed algorithm for constrained nonlinear optimization in Chapter 4, and a framework to employ Nesterov's accelerated method for equality-constrained convex programs in Chapter 5. In Chapter 6, we propose decentralized resource-aware coordination schemes for network optimization problems which combine locally evaluable costs with network-

wide coupling components. In Chapter 7, we propose a framework enabling DERs to participate in the frequency regulation market. Chapter 8 focuses on the problem of primary control as the inertia of the network changes. Finally, Chapter 9 summarizes the thesis and points towards future research directions.

Chapter 2

Preliminaries

Here, we introduce our notational conventions and review basic concepts on graph theory, convex analysis, probability theory, dynamic average consensus, constrained optimization, and event-triggered control. Beyond the knowledge of these concepts, the reader can defer parsing through the specific technical details below.

2.1 Notation

Let $\mathbb{C}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, $\mathbb{Z}$, and $\mathbb{N}$ denote the set of complex numbers, real numbers, non-negative real numbers, positive real numbers, positive integers, and natural numbers, respectively. We use $|x|$ to denote the absolute value of a scalar x, $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$ denote respectively, the space of n dimensional vectors and m by n matrices with real entries. $\mathbf{1}$, $\mathbf{0}$, and I denote the vector or matrix of all ones, all zeros, and the identity matrix of appropriate dimension, respectively. Unless otherwise stated, we denote vectors and matrices by lowercase and uppercase letters, respectively. Given a vector $x \in \mathbb{R}^n$, x_i refers to its ith component. For a matrix $A \in \mathbb{R}^{m\times n}$, A_{ij} denotes its

entry in the ith row and jth column, A_i its ith row, $A^\top$ its transpose, A^{-1} its inverse (if it exists) and null(A) its null space. $A \otimes B$ denotes the Kronecker product between two matrices A and B. Unless otherwise stated, $\mathbf{x} \in \mathbb{R}^{mn}$ denotes the concatenated vector obtained after stacking the vectors $\{x_i\}_{i=1}^n \in \mathbb{R}^m$. For a symmetric matrix A, $A \succ \mathbf{0}$ and $A \succeq \mathbf{0}$ imply that a matrix A is positive definite and semidefinite, respectively, and $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ denote its maximum and minimum eigenvalue, respectively. Regardless of the multiplicity of eigenvalue 0, $\lambda_2(A)$ denotes the minimum non-zero eigenvalue of a positive semidefinite matrix A. For two vectors $x, y \in \mathbb{R}^n$, $[x; y]$ denotes the concatenated vector containing the entries of x and y, in that order, and $x > y$ means that the inequality holds elementwise. e_i^n denotes the $n-$dimensional unit vector in direction i. Finally, $V^\perp$ denotes the orthogonal complement of the vector space V. $\text{diag}(v) \in \mathbb{R}^{n \times n}$ denotes the diagonal matrix with the elements of $v \in \mathbb{R}^n$ in its diagonal. Similarly, for a group of square matrices $\{A_i\}_{i \in \{1,\dots,n\}} \in \mathbb{R}^{m \times m}$, $\text{diag}(A_i) \in \mathbb{R}^{mn \times mn}$ denotes the block-diagonal matrix with each of the matrices A_i arranged along the principal diagonal. We use $\dim(W)$ to denote the dimension of vector space W. $[x]^+$ denotes $\max\{x, 0\}$ and $[x]_a^+$ is defined as

$$[x]_a^+ = \begin{cases} [x]^+ & \text{if } a > 0, \\ 0 & \text{if } a \leq 0. \end{cases}$$

If x is a vector, these functions are applied elementwise.

We let $\mathcal{X}^o$ and $\overline{\mathcal{X}}$ denote the interior and closure of $\mathcal{X}$, resp. $|\mathcal{X}|$ denotes the cardinality of a set $\mathcal{X}$. For 2 sets $\mathcal{X}$ and $\mathcal{Y}$, $\mathcal{X} \cup \mathcal{Y}$ denotes their union and $\mathcal{X} \subseteq \mathcal{Y}$ denotes that $\mathcal{X}$ is a subset of $\mathcal{Y}$.

For a real-valued function $f : \mathbb{R}^n \to \mathbb{R}$, we let ∇f denote its gradient. When we take the partial derivative with respect to a specific argument x, we employ the notation $\nabla_x f$. A continuous

function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathcal{K}_\infty$ if it is strictly increasing, $\alpha(0) = 0$, and $\alpha(x) \to \infty$ as $x \to \infty$. Given a vector field $\psi : \mathbb{R}^n \to \mathbb{R}^n$ and a continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$, $\mathcal{L}_\psi V$ denotes the Lie derivative of V along the flow induced by ψ.

2.2 Graph Theory

We present the basic notions of graph theory following [BCM09, GR01]. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathsf{A})$ be a weighted directed graph (or digraph), with $\mathcal{V}$ as the set of vertices (or nodes) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ as the set of edges: $(v_i, v_j) \in \mathcal{E}$ iff there is an edge from node v_i to node v_j. A graph is *undirected* if $(v_i, v_j) \in \mathcal{E}$ iff $(v_j, v_i) \in \mathcal{E}$. With $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$, the *adjacency matrix* $\mathsf{A} \in \mathbb{R}^{n \times n}$ of $\mathcal{G}$ is such that $\mathsf{A}_{ij} > 0$ if $(v_i, v_j) \in \mathcal{E}$ and $\mathsf{A}_{ij} = 0$, otherwise. A directed path is an ordered sequence of vertices such that any pair of consecutive vertices is an edge. A loop is a path in which the first and last vertices are same and none of the other vertices is repeated. A graph is strongly connected if there is a path between any two distinct vertices. A tree is a graph whose underlying undirected graph does not have any loops and is connected. A digraph is strongly connected if there is a directed path between any two distinct vertices. The out- and in-degree of a node are, resp., the number of outgoing edges from and incoming edges to it. The weighted out-degree and weighted in-degree of a node v_i are $d^{\text{out}}(v_i) = \sum_{j=1}^n \mathsf{A}_{ij}$ and $d^{\text{in}}(v_i) = \sum_{j=1}^n \mathsf{A}_{ji}$, resp. The *out-degree matrix* $\mathsf{D}^{\text{out}} \in \mathbb{R}^{n \times n}$ and *in-degree matrix* $\mathsf{D}^{\text{in}} \in \mathbb{R}^{n \times n}$ are diagonal matrices defined as $\mathsf{D}^{\text{out}}_{ii} = d^{\text{out}}(v_i)$ and $\mathsf{D}^{\text{in}}_{ii} = d^{\text{in}}(v_i)$, resp. A graph is weight-balanced if $\mathsf{D}^{\text{out}} = \mathsf{D}^{\text{in}}$. The *Laplacian* $\mathsf{L} \in \mathbb{R}^{n \times n}$ is $\mathsf{L} = \mathsf{D}^{\text{in}} - \mathsf{A}$. All eigenvalues of L have nonnegative real parts, 0 is simple with left eigenvector $\mathbf{1}$ iff $\mathcal{G}$ is strongly connected, and $\mathsf{L}\mathbf{1} = \mathbf{0}$ iff $\mathcal{G}$ is weight-balanced iff $\mathsf{L} + \mathsf{L}^\top$ is positive semidefinite. If $\mathcal{G}$ is strongly connected, it follows from [LZ15, Lemma 3] that there exists a positive right eigenvector $\bar{v} \in \mathbb{R}^n$

associated to 0. The *incidence matrix* $\mathsf{M} \in \mathbb{R}^{n\times m}$ is defined such that $\mathsf{M}_{ij} = 1$ if the edge e_j leaves vertex v_i, -1 if it enters vertex v_i, and 0 otherwise. Note that every column of M has only two non-zero entries and $\mathbf{1}^\top \mathsf{M} = \mathbf{0}$. The *fundamental loop matrix* $\mathsf{N} \in \mathbb{R}^{m\times(m-n+1)}$ of a graph has N_{ij} as 1 (-1, respectively) if the edge $e_i$has the same (opposite, respectively) orientation as loop l_j, and $\mathsf{N}_{ij} = 0$ if edge e_i is not part of loop l_j. We use $\mathsf{P}_{\text{ref}} \in \mathbb{R}^{(n-1)\times m}$ to denote the *path matrix* of a tree with reference vertex ref: the ijth entry of the path matrix is +1/-1 if edge e_j is in the directed path from v_i to ref and has the same/opposite orientation as this path, and is 0 otherwise.

2.3 Convex Analysis

Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a convex set. A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* on $\mathcal{C}$

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y),$$

for all $x, y \in \mathcal{C}$ and $\lambda \in [0, 1]$. Convex functions have the property of having the same local and global minimizers. A continuously differentiable $f : \mathbb{R}^n \to \mathbb{R}$ is convex on $\mathcal{C}$ *iff*

$$f(y) \geq f(x) + (y-x)^\top \nabla f(x),$$

for all $x, y \in \mathcal{C}$. A twice differentiable function is convex *iff* its Hessian is positive semi-definite. A twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is *strongly convex* on $\mathcal{C}$ with parameter $c \in \mathbb{R}_{>0}$ *iff* $\nabla^2 f(x) \geq cI$ for all $x \in \mathcal{C}$. For a convex set Ω, $\Pi_\Omega(y)$ denotes the projection of a point $y \in \mathbb{R}^n$

on Ω, i.e.,

$$\Pi_{\Omega}(y) = \text{argmin}_{x\in\Omega}\|x - y\|.$$

2.4 Dynamic Average Consensus

Consider a group of $n \in \mathbb{Z}_{>1}$ agents communicating over a strongly connected weight-balanced digraph $\mathcal{G}$ with Laplacian L. Each agent $i \in \{1, \dots, n\}$ has a state $x_i \in \mathbb{R}$ and an input $z_i \in \mathbb{R}$. The dynamic average consensus algorithm aims at making all the agents track the average $\frac{1}{n}\sum_{i=1}^{n} z_i$ asymptotically. Here we first present the algorithm following [SOSM05b], where it was introduced for undirected graphs. Consider

$$\dot{\mathbf{x}} = -\mathsf{L}\,\mathbf{x} + \dot{\mathbf{z}}. \tag{2.1}$$

If $\sum_{i=1}^{n} x_i(0) = \sum_{i=1}^{n} z_i(0)$ and the input $\mathbf{z}$ is bounded, then $x_i(t) \to \frac{1}{n}\sum_{i=1}^{n} z_i(t)$ as $t \to \infty$ for $i \in \{1, \dots, n\}$. Algorithm (2.1) requires the knowledge of the initial conditions. To overcome this, we next present the algorithm following [KCM15b] and given by

$$\dot{\mathbf{x}} = \dot{\mathbf{z}} - \nu(\mathbf{x} - \mathbf{z}) - \beta\,\mathsf{L}\,\mathbf{x} - \mathbf{v}, \tag{2.2a}$$

$$\dot{\mathbf{v}} = \nu\beta\,\mathsf{L}\,\mathbf{x}, \tag{2.2b}$$

where $\nu, \beta > 0$ are the design parameters. If the algorithm is initialized with $\sum_{i=1}^{n} v(0) = 0$, then the steady-state error between the state x_i of each agent $i \in \{1, \dots, n\}$ and the average signal $\frac{1}{n}\sum_{i=1}^{n} z_i$ is bounded, and goes to zero if $\dot{\mathbf{z}} \to \mathbf{0}$.

2.5 Probability Theory

Given an event E, we let E^c denote its complement and $\Pr(E)$ its probability. $\mathbb{E}(w)$ denotes the expected value of a random variable w. Given a normally distributed random variable $\zeta \sim \mathcal{N}(\mu, \sigma)$ with mean μ and variance σ, the probability $\Pr(\zeta \leq x)$ of ζ being less than or equal to x is denoted by $\Phi(x)$.

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{(\zeta-\mu)^2}{2\sigma}} d\zeta.$$

For $x \geq 0$, the error function erf, defined as $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\zeta^2} du$, denotes the probability of a normal random variable with mean 0 and variance 1/2 being in the interval $[-x, x]$. For a normal random variable with mean 0 and variance 1/2, the functions Φ and erf are related by

$$\Phi(x) = \frac{1}{2}\Big(1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\Big). \tag{2.3}$$

2.6 Constrained Optimization

Here, we introduce basic concepts of constrained optimization following [Ber99]. Consider the following nonlinear optimization problem

$$\begin{aligned} \min_{x \in D} \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0,\ h(x) = 0, \end{aligned} \tag{2.4}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^p$ are twice continuously differentiable functions with $p \leq n$ and $\mathcal{D} \subset \mathbb{R}^n$ is a compact set which is regular (i.e., $\mathcal{D} = \overline{\mathcal{D}^o}$). The feasible set of (2.4) is

$$\mathcal{F} = \{x \mid x \in \mathcal{D}, g(x) \leq 0, h(x) = 0\}.$$

Based on the index sets for the inequality constraints

$$I_0(x) = \{j \mid g_j(x) = 0\},$$

$$I_+(x) = \{j \mid g_j(x) \geq 0\},$$

we define the following regularity conditions:

(a) The *linear independence constraint qualification* (LICQ) holds at $x \in \mathbb{R}^n$ if $\{\nabla g_j(x)\}_{j \in I_0(x)} \cup \{\nabla h_k\}_{k \in \{1,\ldots,p\}}$ are linearly independent;

(b) The *extended Mangasarian-Fromovitz constraint qualification* (EMFCQ) holds at $x \in \mathbb{R}^n$ if $\{\nabla h_k\}_{k \in \{1,\ldots,p\}}$ are linearly independent and there exists $z \in \mathbb{R}^n$ with

$$\nabla g_j(x)^\top z < 0, \quad \forall j \in I_+(x), \tag{2.5a}$$

$$\nabla h_k(x)^\top z = 0, \quad \forall k \in \{1, \ldots, p\}. \tag{2.5b}$$

The Lagrangian function $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ associated with (2.4) is given by

$$L(x, \lambda, \mu) = f(x) + \lambda^\top g(x) + \mu^\top h(x),$$

where $\lambda \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^p$ are the Lagrange multipliers (also called dual variables) associated with the inequality and equality constraints, resp. A Karush-Kuhn-Tucker (KKT) point for (2.4) is a triplet $(\bar{x}, \bar{\lambda}, \bar{\mu})$ such that

$$\begin{aligned} \nabla_x L(\bar{x}, \bar{\lambda}, \bar{\mu}) &= 0, \\ \bar{\lambda}^\top g(\bar{x}) = 0, \quad \bar{\lambda} \geq 0, \quad g(\bar{x}) &\leq 0, \\ h(\bar{x}) &= 0. \end{aligned}$$

Under any of the regularity conditions above, the KKT conditions are necessary for a point to be locally optimal.

2.6.1 Continuously differentiable exact penalty functions

With exact penalty functions, the basic idea is to replace the constrained optimization problem (2.4) by an equivalent unconstrained problem. Here, we introduce continuously differentiable exact penalty functions following [GP79, DG89]. The key observation is that one can interpret a KKT tuple as establishing a relationship between a primal solution $\bar{x}$ and the dual variables $(\bar{\lambda}, \bar{\mu})$. In turn, the following result introduces multiplier functions that extend this relationship to any $x \in \mathbb{R}^n$.

Proposition 2.6.1. ***(Multiplier functions and their derivatives [DG89]).*** *Assume that LICQ is*

satisfied at all $x \in \mathcal{D}$. Let $G(x) = diag(g(x))$ and, for $\gamma \neq 0$, define $N : \mathbb{R}^n \to \mathbb{R}^{(m+p)\times(m+p)}$ by

$$N(x) = \begin{bmatrix} \nabla g(x)^\top \nabla g(x) + \gamma^2 G^2(x) & \nabla g(x)^\top \nabla h(x) \\ \nabla h(x)^\top \nabla g(x) & \nabla h(x)^\top \nabla h(x) \end{bmatrix}. \tag{2.6}$$

Then $N(x)$ is a positive definite matrix for any $x \in \mathsf{D}$. Given the functions $x \mapsto (\lambda(x), \mu(x))$ defined by

$$\begin{bmatrix} \lambda(x) \\ \mu(x) \end{bmatrix} = -N^{-1}(x) \begin{bmatrix} \nabla g(x)^\top \\ \nabla h(x)^\top \end{bmatrix} \nabla f(x), \tag{2.7}$$

one has that

(a) if $(\bar{x}, \bar{\lambda}, \bar{\mu})$ is a KKT triple for problem (2.4)*, then $\lambda(\bar{x}) = \bar{\lambda}$ and $\mu(\bar{x}) = \bar{\mu}$;*

(b) both functions are continuously differentiable and their Jacobian matrices are given by

$$\begin{bmatrix} \nabla \lambda(x)^\top \\ \nabla \mu(x)^\top \end{bmatrix} = -N^{-1}(x) \begin{bmatrix} R(x) \\ S(x) \end{bmatrix}, \tag{2.8}$$

where

$$\begin{aligned} R(x) =& \nabla g(x)^\top \nabla_x^2 L(x, \lambda(x), \mu(x)) + \sum_{j=1}^{m} e_j^m \nabla_x L(x, \lambda(x), \mu(x))^\top \nabla^2 g_j(x) && \text{(2.9a)} \\ &+ 2\gamma^2 \Lambda(x) G(x) \nabla g(x)^\top \\ S(x) =& \nabla h(x)^\top \nabla_x^2 L(x, \lambda(x), \mu(x)) + \sum_{k=1}^{p} e_k^p \nabla_x L(x, \lambda(x), \mu(x))^\top \nabla^2 h_k(x) && \text{(2.9b)} \end{aligned}$$

where we use the shorthand notation

$$\nabla_x L(x, \lambda(x), \mu(x)) = [\nabla_x L(x, \lambda, \mu)]_{\substack{\lambda=\lambda(x),\\ \mu=\mu(x)}},$$

$$\nabla_x^2 L(x, \lambda(x), \mu(x)) = [\nabla_x^2 L(x, \lambda, \mu)]_{\substack{\lambda=\lambda(x),\\ \mu=\mu(x)}},$$

$\Lambda(x) = diag(\lambda(x))$*, and* e_j^m *and* e_k^p *denote, resp., the* j*th and* k*th column of the* $m \times m$ *and* $p \times p$ *identity matrix.*

The multiplier functions in Proposition 2.6.1 can be used to replace the multiplier vectors in the augmented Lagrangian of [Roc74] to define the continuously differentiable exact penalty function. Given $\epsilon > 0$ and $j \in \{1, \ldots, m\}$, define

$$y_j^\epsilon(x) = \Big(- \min \Big[0, g_j(x) + \frac{\epsilon}{2}\lambda_j(x)\Big]\Big)^{1/2},$$

and let $Y^\epsilon(x) = \text{diag}(y^\epsilon(x))$. Consider the continuously differentiable function $f^\epsilon : \mathbb{R}^n \to \mathbb{R}$,

$$\begin{aligned} f^\epsilon(x) = f(x) &+ \lambda(x)^\top (g(x) + Y^\epsilon(x) y^\epsilon(x)) + \mu(x)^\top h(x) \\ &+ \frac{1}{\epsilon}\|g(x) + Y^\epsilon(x) y^\epsilon(x)\|^2 + \frac{1}{\epsilon}\|h(x)\|^2. \end{aligned} \tag{2.10}$$

The following result characterizes the extent to which f^ϵ is an exact penalty function.

Proposition 2.6.2. ***(Continuously differentiable exact penalty function [DG89]).*** *Assume LICQ is satisfied at all* $x \in \mathcal{D}$ *and consider the unconstrained problem*

$$\min_{x \in \mathcal{D}^o} f^\epsilon(x). \tag{2.11}$$

Then, the following holds:

(a) there exists $\bar{\epsilon}$ *such that the set of global minimizers of* (2.4) *and* (2.11) *are equal for all* $\epsilon \in (0, \bar{\epsilon}]$*;*

(b) if $(\bar{x}, \bar{\lambda}, \bar{\mu})$ *is a KKT point for problem* (2.4)*, then* $\nabla f^{\epsilon}(\bar{x}) = 0$ *for all* $\epsilon > 0$*;*

(c) under the additional assumption that EMFCQ holds on $\mathcal{D}$*, there exists* $\bar{\epsilon}$ *such that for all* $\epsilon \in (0, \bar{\epsilon}]$*,* $\nabla f^{\epsilon}(\bar{x}) = 0$ *implies that* $(\bar{x}, \lambda(\bar{x}), \mu(\bar{x}))$ *is a KKT point for problem* (2.4)*.*

2.6.2 Globally Projected Dynamical Systems

Here we review the basic concepts on the stability of continuous projected dynamical systems and their application in constrained optimization following [XW00, Gao03]. Problem (2.4) can also be solved using

$$\dot{x} = \Pi_{\mathcal{F}}(x - \lambda \nabla h(x)) - x, \tag{2.12}$$

where $\lambda > 0$ is a design parameter. Unlike the commonly employed projected gradient dynamics, cf. [HM94], which is discontinuous at the boundary of Ω, the dynamics (2.12) is continuous due to the gradual application of the projection operator throughout the constraint set. The following result characterizes its convergence properties.

Theorem 2.6.3. ***(Forward invariance of the feasible set and convergence to an optimizer [XW00, Gao03]).*** *Assume that* ∇h *is locally Lipschitz continuous on an open set containing* Ω*. Then*

(i) the solution of (2.12) *approaches the set* Ω *exponentially fast. Moreover, if* $x(0) \in \Omega$*, then* $x(t) \in \Omega$ *for all* $t > 0$*;*

(ii) for all $\lambda > 0$, the dynamics (2.12) *is stable, and for any initial condition $x(0) \in \Omega$, the trajectory of* (2.12) *converges to a solution of* (2.4).

2.7 Event-Triggered Control

Here we present the basics of event-triggered control following [Tab07, HJT12]. Consider

$$\dot{x} = f(x, u), \tag{2.13}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^p$ denote the system state and input, respectively. Assume there exists a control

$$u = k(x), \tag{2.14}$$

such that the closed-loop dynamics,

$$\dot{x} = f(x, k(x + e)), \tag{2.15}$$

abbreviated ψ_{cl}, is input-to-state stable (ISS) with respect to the error $e \in \mathbb{R}^n$. Formally, assume there exists a Lyapunov function V such that its Lie derivative along (2.15) satisfies

$$\mathcal{L}_{\psi_{\text{cl}}} V \leq -\alpha(\|x\|) + \gamma(\|e\|),$$

where α and γ are class $\mathcal{K}_\infty$ functions. The implementation of the closed-loop system (2.13)-(2.14) requires continuous updates of the actuator, which is not realizable in practice. Instead, event-triggered control seeks to prescribe opportunistic updates of the actuator that ensure the convergence properties of the closed-loop system are retained. This leads to a sample-and-hold implementation of the controller of the form

$$u(t) = k(x(t_k)) \quad t \in [t_k, t_{k+1}), \tag{2.16}$$

where $\{t_k\}_{k=0}^{\infty}$ are the *triggering times* when the control input is updated. In order to ensure the stability of the nonlinear system under (2.16) and to prescribe the triggering times, we look at the evolution of the Lyapunov function V. Define the error variable as $e = x - x^k$, where we use the shorthand notation $x^k = x(t_k)$. With $\sigma \in (0, 1)$, if the error satisfies $\gamma(\|e\|) \leq \sigma\alpha(\|x\|)$, then the Lie derivative of the Lyapunov function during the time $[t_k, t_{k+1})$ is upper bounded as

$$\mathcal{L}_{\psi_{\text{cl}}} V \leq (1 - \sigma)\alpha(\|x\|).$$

Note that at $t = t_k$, the error satisfies $e = 0$. Hence, setting $t_0 = 0$, we ensure $\mathcal{L}_{\psi_{\text{cl}}} V < 0$ defining the triggering times as

$$t_{k+1} = \min\{t > t_k \mid \gamma(\|e\|) \leq \sigma\alpha(\|x\|)\}. \tag{2.17}$$

Although (2.17) guarantees the stability of the closed-loop system, it could result in an infinite number of triggering times in a finite time interval (a.k.a. *Zeno behavior*). Hence, for implementation in practice and to conclude asymptotic stability, it is necessary to have a uniform lower bound $\tau > 0$

of the inter-event times, i.e., $t_{k+1} - t_k \geq \tau$ for all k. We refer to τ as the minimum inter-event time (*MIET*). The existence of a MIET is guaranteed with the control law (2.16) and the triggering condition (2.17) if the dynamics (2.13) is linear, and also for certain nonlinear systems under suitable assumptions, cf. [Tab07].

Part I

Application-Agnostic Distributed Algorithms for Network Problems

Chapter 3

Distributed Algorithms for Linear Equations

This chapter deals with linear algebraic equations where the global coefficient matrix and constant vector are given respectively, by the summation of the (not necessarily invertible) coefficient matrices and constant vectors of the individual agents. We start with developing exact unconstrained optimization reformulations for the original problem. Based on whether the communication graph is undirected or directed, our approach is structured into two blocks. We first present an exponentially fast input-to-state stable 2-hop distributed algorithm for the undirected case. For the directed case, we first provide a gradient-based, centralized algorithm which serves as a reference for the ensuing design of distributed algorithms. We then propose two sets of exponentially stable continuous-time distributed algorithms based on estimating non-distributed terms in the centralized algorithm using dynamic average consensus. The first algorithm works for time-varying weight-balanced directed networks, and the second algorithm works for general directed networks for which the communication graphs might not be balanced.

3.1 Problem Formulation

Consider a group of n agents interacting over a graph that seek to solve in a distributed way the linear algebraic equation

$$\underbrace{\left(\sum_{i=1}^{n} A_i\right)}_{A} x = \underbrace{\left(\sum_{i=1}^{n} b_i\right)}_{b}, \tag{3.1}$$

where $x \in \mathbb{R}^m$ is the unknown solution vector, and $A_i \in \mathbb{R}^{m\times m}$ and $b_i \in \mathbb{R}^m$ are the coefficient matrix and constant vector corresponding to agent $i \in \{1, \dots, n\}$. We assume that (3.1) has at least one solution. The formulation (3.1) includes, as a particular case, scenarios where each agent i knows only some rows of the coefficient matrix A and constant vector b. Our approach consists of first formulating (3.1) as a system involving n unknown solution vectors, one per agent, and then reformulating it as a convex optimization problem.

We start by endowing each agent with its own version $x_i \in \mathbb{R}^m$ of x. Then (3.1) can be equivalently written as

$$\sum_{i=1}^{n} A_i x_i = \sum_{i=1}^{n} b_i \tag{3.2a}$$

$$x_i = x_j \quad \forall i, j. \tag{3.2b}$$

Equation (3.2b) ensures that $x_i = x$ for all the agents. Clearly the set of equations (3.2) and the original problem (3.1) are equivalent.

Next, we provide equivalent convex optimization reformulations for (3.2) and synthesize distributed algorithms, depending on whether the communication graph is undirected or directed.

3.2 Distributed Algorithm Over Undirected Networks

Here, we design and prove the convergence and robustness properties of a distributed algorithm to solve (3.1) over undirected graphs. We begin by noting that although (3.2b) is distributed, (3.2a) is not. To address this, we introduce a new variable $y_i \in \mathbb{R}^m$ per agent $i \in \{1, \dots, n\}$. Let $\mathbf{y} = [y_1; \dots; y_n] \in \mathbb{R}^{mn}$ and consider the following set of equations

$$\underbrace{\begin{pmatrix} \mathbf{A} & -\mathbf{L} \\ \mathbf{L} & \mathbf{0} \end{pmatrix}}_{\mathbf{P}} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \underbrace{\begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix}}_{\mathbf{q}}, \tag{3.3}$$

where $\mathbf{A} \in \mathbb{R}^{mn \times mn}$ denotes the block-diagonal matrix obtained after putting the matrices $\{A_i\}_{i=1}^n$ along the principal diagonal, $\mathbf{b} = [b_1; \dots; b_n] \in \mathbb{R}^{mn}$, and $\mathbf{L} = \mathsf{L} \otimes I$. Note that the set of equations (3.3) is distributed. The following result characterizes the equivalence between (3.3) and (3.1).

Proposition 3.2.1. ***(Equivalence between*** (3.3) ***and*** (3.1)***).*** *The solutions of* (3.3) *are of the form* $(\mathbf{1} \otimes x; \bar{\mathbf{y}} + \mathbf{1} \otimes y)$*, where* $x \in \mathbb{R}^m$ *solves* (3.1)*,* $\mathbf{A}(\mathbf{1} \otimes x) - \mathbf{b} = \mathbf{L}\bar{\mathbf{y}}$*, and* $y \in \mathbb{R}^m$*.*

Proof. Note that (3.3) can be rewritten as

$$\begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_n \end{bmatrix} \mathbf{x} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} + \mathbf{L}\mathbf{y} \tag{3.4a}$$

$$\mathbf{L}\mathbf{x} = \mathbf{0}. \tag{3.4b}$$

Equation (3.4b) implies that $\mathbf{x} = \mathbf{1} \otimes x$, with $x \in \mathbb{R}^m$. Then, from (3.4a), we have for each

$i \in \{1, \dots, n\}$,

$$A_i x = b_i + (\mathsf{L}_i \otimes I)\mathbf{y},$$

where L_i denotes the ith row of the Laplacian L. Summing over all agents, we obtain

$$\Big(\sum_{i=1}^{n} A_i \Big) x = \sum_{i=1}^{n} b_i + \sum_{i=1}^{n} (\mathsf{L}_i \otimes I)\mathbf{y}.$$

Since $\mathbf{1}^\top \mathsf{L} = 0$, the last summand vanishes, which yields (3.1). The expression for $\mathbf{y}$ again follows directly from the fact that $\mathbf{1}^\top \mathsf{L} = 0$. □

Our next goal is to synthesize a distributed algorithm to solve (3.3). Our algorithm design is based on formulating this equation as an unconstrained optimization problem. Let $\mathbf{z} = (\mathbf{x}; \mathbf{y})$ and consider the quadratic function $V_1 : \mathbb{R}^{2mn} \to \mathbb{R}$

$$V_1(\mathbf{z}) = \frac{1}{2}(\mathbf{P}\mathbf{z} - \mathbf{q})^\top (\mathbf{P}\mathbf{z} - \mathbf{q}). \tag{3.5}$$

Note that V_1 vanishes over the solution set of $\mathbf{P}\mathbf{z} = \mathbf{q}$ and takes positive values otherwise. The problem of solving (3.3) can be reformulated as

$$\min_{\mathbf{z}} V_1(\mathbf{z}).$$

The gradient descent dynamics of V_1 is given by

$$\dot{\mathbf{z}} = -\mathbf{P}^\top (\mathbf{P}\mathbf{z} - \mathbf{q}).$$

When convenient, we refer to this dynamics as ϕ_{grad}. In expanded form, it takes the form

$$\dot{\mathbf{x}} = -\mathbf{A}^\top[\mathbf{A}\mathbf{x} - \mathbf{L}\mathbf{y} - \mathbf{b}] - \mathbf{L}^2\mathbf{x} \tag{3.6a}$$

$$\dot{\mathbf{y}} = \mathbf{L}[\mathbf{A}\mathbf{x} - \mathbf{L}\mathbf{y} - \mathbf{b}]. \tag{3.6b}$$

From (3.6), each agent $i \in \{1, \dots, n\}$ has the dynamics

$$\dot{x}_i = -A_i^\top\Big(A_i x_i - b_i - \sum_{j\in\mathcal{N}_i}(y_i - y_j)\Big) - \sum_{j\in\mathcal{N}_i}(x_i^{\mathsf{L}} - x_j^{\mathsf{L}})$$

$$\dot{y}_i = \sum_{j\in\mathcal{N}_i}\Big(A_i x_i - b_i - (A_j x_j - b_j)\Big) - \sum_{j\in\mathcal{N}_i}(y_i^{\mathsf{L}} - y_j^{\mathsf{L}}),$$

where $x_k^{\mathsf{L}} = \sum_{j\in\mathcal{N}_k}(x_k - x_j)$ and $y_k^{\mathsf{L}} = \sum_{j\in\mathcal{N}_k}(y_k - y_j)$. This algorithm is 2-hop distributed, meaning that to execute it, each agent $i \in \{1, \dots, n\}$ needs to know its state $(x_i; y_i)$ and the state of its 2-hop neighbors. The next result characterizes its convergence properties.

Proposition 3.2.2. ***(Exponential convergence of*** (3.6) ***to solution of linear system).*** *The dynamics* (3.6) *converges to a solution of* (3.3) *exponentially with a rate proportional to* $\lambda_2(\mathbf{P}^\top\mathbf{P})$.

Proof. Let $\mathbf{w} \in \text{null}(\mathbf{P})$ and note

$$\mathbf{w}^\top\dot{\mathbf{z}} = -\mathbf{w}^\top\mathbf{P}^\top(\mathbf{P}\mathbf{z} - \mathbf{q}) = 0.$$

This means that the dynamics of $\mathbf{z}$ is orthogonal to null($\mathbf{P}$). Let us decompose $\mathbf{z}(t)$ as $\mathbf{z}(t) = \mathbf{z}_\parallel(t) + \mathbf{z}_\perp(t)$. Here, $\mathbf{z}_\parallel(t)$ is the component of $\mathbf{z}(t)$ in null($\mathbf{P}$) and $\mathbf{z}_\perp(t)$ is the component orthogonal to it. From the above discussion, we have that $\mathbf{z}_\parallel(t) = \mathbf{z}_\parallel(0)$ under the dynamics (3.6). Since this component does not change, consider the particular solution $\mathbf{z}^*$ of (3.3) that satisfies $\mathbf{z}_\parallel^* = \mathbf{z}_\parallel(0)$.

Note that $\mathbf{z}^*$ defined in this way is unique. Now, consider the Lyapunov function $V_2 : \mathbb{R}^{2mn} \to \mathbb{R}$

$$V_2(\mathbf{z}) = \frac{1}{2}(\mathbf{z} - \mathbf{z}^*)^\top(\mathbf{z} - \mathbf{z}^*). \tag{3.7}$$

The derivative of V_2 along the dynamics (3.6) is given by

$$\begin{aligned}
\mathcal{L}_{\phi_{\text{grad}}} V_2(\mathbf{z}) &= (\mathbf{z} - \mathbf{z}^*)^\top \dot{\mathbf{z}} \\
&= -(\mathbf{z} - \mathbf{z}^*)^\top \mathbf{P}^\top(\mathbf{P}\mathbf{z} - \mathbf{q}) \\
&= -(\mathbf{z} - \mathbf{z}^*)^\top \mathbf{P}^\top \mathbf{P}(\mathbf{z} - \mathbf{z}^*) \leq -2\lambda_2(\mathbf{P}^\top\mathbf{P})V_2(\mathbf{z}).
\end{aligned}$$

The last inequality follows from the fact that the evolution of $\mathbf{z}$ is orthogonal to the nullspace of $\mathbf{P}$. This proves that, starting from $\mathbf{z}(0)$, the dynamics converges to the solution $\mathbf{z}^*$ of (3.3) exponentially fast with a rate determined by the minimum non-zero eigenvalue of $\mathbf{P}^\top\mathbf{P}$. □

Remark 3.2.3. ***(Comparison with the literature).*** *The algorithmic design procedure we employ here is similar to the one used in [WM18a], which leads to an algorithm that also does not require the positive definiteness of the individual matrices. Interestingly, the convergence analysis in [WM18a] uses the linearity of the dynamics and La Salle's invariance principle to conclude exponential stability, although it does not guarantee that the agents converge to the same solution. By contrast, the Lyapunov-based technical analysis presented here, based on exploiting the orthogonality of the dynamics to the nullspace of the reformulated system matrix, allows us to lower bound the exponential convergence rate and formally characterize the robustness properties of the algorithm against disturbances. Both properties are key for the application later in Chapter 4 to distributed gradient computation via characterizing the stability of the interconnected system.* •

Next, we examine the robustness to disturbances of the dynamics (3.6). This is motivated by the observation that, in practical scenarios, one may face errors in the execution due to imperfect knowledge of the problem data, imperfect information about the state of other agents, or other external disturbances. Formally, we consider

$$\dot{\mathbf{z}} = \phi_{\text{grad}}(\mathbf{z}) + d(t) = -\mathbf{P}^\top(\mathbf{P}\mathbf{z} - \mathbf{q}) + d(t), \tag{3.8}$$

where $d(t)$ denotes the disturbance.

Proposition 3.2.4. ***(Robustness of*** (3.8) ***against disturbances).*** *The dynamics* (3.8) *is input-to-state stable (ISS) with respect to the set of equilibria of* (3.6)*.*

Proof. The disturbance $d(t)$ in (3.8) can be decomposed as $d(t) = d_{\|}(t) + d_{\perp}(t)$. Due to the presence of $d_{\|}(t) \in \text{null}(\mathbf{P})$, the component of $\mathbf{z}(t)$ in $\text{null}(\mathbf{P})$ does not remain constant any more. In fact, along (3.8), we have $\mathbf{w}^\top \dot{\mathbf{z}} = \mathbf{w}^\top d_{\|}$ for all $\mathbf{w} \in \text{null}(\mathbf{P})$, and therefore we deduce that $\dot{\mathbf{z}}_{\|}(t) = d_{\|}(t)$. Consider then the equilibrium trajectory $t \mapsto z^*(t)$, where $\mathbf{z}^*(t)$ is uniquely determined by the equations $\mathbf{P}\mathbf{z}^*(t) = \mathbf{q}$ and $\mathbf{z}^*_{\|}(t) = \mathbf{z}_{\|}(t)$. Let V_2 be the same function as in (3.7), but now with the time-varying $\mathbf{z}^*(t)$. The Lie derivative of V_2 is now given by

$$\begin{aligned}
\mathcal{L}_{\phi_{\text{grad}}+d} V_2 &= (\mathbf{z} - \mathbf{z}^*)^\top (\dot{\mathbf{z}} - \dot{\mathbf{z}}^*) \\
&= (\mathbf{z} - \mathbf{z}^*)^\top (-\mathbf{P}^\top(\mathbf{P}\mathbf{z} - \mathbf{q}) + d - d_{\|}) \\
&= (\mathbf{z} - \mathbf{z}^*)^\top (-\mathbf{P}^\top(\mathbf{P}\mathbf{z} - \mathbf{q}) + d_{\perp}) \\
&\leq -\lambda_2(\mathbf{P}^\top\mathbf{P})\|\mathbf{z} - \mathbf{z}^*\|^2 + \|\mathbf{z} - \mathbf{z}^*\|\|d_{\perp}\| \\
&\leq -\lambda_2(\mathbf{P}^\top\mathbf{P})\|\mathbf{z} - \mathbf{z}^*\|^2 + \|\mathbf{z} - \mathbf{z}^*\|\|d\|.
\end{aligned}$$

Choose $\theta \in (0, 1)$. Then the above inequality can be decomposed as

$$\mathcal{L}_{\phi_{\text{grad}}+d} V_2 \leq -\lambda_2(\mathbf{P}^\top\mathbf{P})(1-\theta)\|\mathbf{z}-\mathbf{z}^*\|^2 - \lambda_2(\mathbf{P}^\top\mathbf{P})\theta\|\mathbf{z}-\mathbf{z}^*\|^2 + \|\mathbf{z}-\mathbf{z}^*\|\|d\|.$$

Hence, $L_{\phi_{\text{grad}}+d} V_2 \leq -\lambda_2(\mathbf{P}^\top\mathbf{P})(1-\theta)\|\mathbf{z}-\mathbf{z}^*\|^2$ if $\|\mathbf{z}-\mathbf{z}^*\| \geq \dfrac{\|d\|}{\lambda_2(\mathbf{P}'\mathbf{P})\theta}$. From [Kha02, Theorem 4.19], this means that the system is input-to-state stable with respect to the set of equilibria with gain $\gamma(r) = \dfrac{r}{\lambda_2(\mathbf{P}^\top\mathbf{P})\theta}$. □

Proposition 3.2.4 implies that the trajectories of (3.8) asymptotically converge to a neighborhood of the set of equilibria of (3.6) (with the size of the neighborhood scaling up with the size of the disturbance). All equilibria correspond to solutions of (3.1).

3.3 Distributed Algorithms Over Directed Networks

We present distributed algorithms to solve (3.1) over directed networks. We start with weight-balanced networks and then extend our treatment to unbalanced networks. As in Section 3.2, our approach here is based on reformulation (3.2) as a convex optimization problem. Consider the quadratic function $f : \mathbb{R}^{mn} \to \mathbb{R}$

$$f(\mathbf{x}) = \Big(\sum_{i=1}^{n}(A_i x_i - b_i)\Big)^\top \Big(\sum_{i=1}^{n}(A_i x_i - b_i)\Big),$$

which is convex and attains its minimum over the solution set of (3.2a). For convenience, we use $f(\mathbf{x}) = (\mathbf{A}\mathbf{x}-\mathbf{b})^\top \mathbb{1}\,\mathbb{1}^\top(\mathbf{A}\mathbf{x}-\mathbf{b})$, where $\mathbb{1} = \mathbf{1} \otimes I$. If $\mathcal{G}$ is strongly connected, the solutions of (3.2)

are the same as the optimizers of

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{L}^\top \mathbf{x} = \mathbf{0}. \end{aligned} \tag{3.9}$$

3.3.1 Centralized Algorithm Over Weight-Balanced Networks

First, we introduce a centralized algorithm using the fact that the objective function vanishes at the optimizers of (3.9). Let

$$\min_{\mathbf{x}} \quad \frac{1}{2}\alpha \mathbf{x}^\top (\mathbf{L} + \mathbf{L}^\top)\mathbf{x} + \beta f(\mathbf{x}), \tag{3.10}$$

where $\alpha, \beta > 0$. Clearly, (3.9) and (3.10) have the same set of solutions if $\mathcal{G}$ is strongly connected and weight-balanced. Since problem (3.10) is unconstrained, one can use gradient descent to find its optimizers. However, the gradient $-\alpha(\mathbf{L} + \mathbf{L}^\top)\mathbf{x} - \beta \mathbf{A}^\top \mathbb{1}\mathbb{1}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})$ of the objective function in (3.10) involves terms with $\mathbf{L}^\top$, whose computation would require information from in-neighbors. Instead, we consider the following gradient-based dynamics

$$\dot{\mathbf{x}} = -\alpha \mathbf{L}\mathbf{x} - \beta \mathbf{A}^\top \mathbb{1}\mathbb{1}^\top(\mathbf{A}\mathbf{x} - \mathbf{b}). \tag{3.11}$$

Whenever convenient, we refer to (3.11) as ψ_{grad}. Note that the first term in the dynamics (3.11) is distributed, meaning that each agent can implement it with information from its out-neighbors. The second term, however, requires collective information from all the agents because of the summation across the network. Nevertheless, this algorithm serves as the basis for our distributed algorithm design in the next section.

The next result formally characterizes the equivalence between the equilibria of (3.11) and

the solutions of (3.1).

Lemma 3.3.1. ***(Equivalence between*** (3.11) ***and*** (3.1)***).*** *Let $\mathcal{G}$ be a strongly connected and weight-balanced digraph. Then for all $\alpha, \beta \in \mathbb{R}_{>0}$, $\mathbf{x}^*$ is an equilibrium of* (3.11) *if and only if $\mathbf{x}^* = \mathbf{1} \otimes x^*$, where $x^* \in \mathbb{R}^m$ solves* (3.1)*.*

Proof. The implication from right to left is immediate. To prove the implication in the other direction, let $\bar{x} \in \mathbb{R}^m$ be a solution of (3.1) and consider $\bar{\mathbf{x}} = \mathbf{1} \otimes \bar{x}$. Since $\mathbf{x}^*$ and $\bar{\mathbf{x}}$ are equilibria of (3.11),

$$\alpha \mathbf{L}(\mathbf{x}^* - \bar{\mathbf{x}}) + \beta \mathbf{A}^\top \mathbb{1}\mathbb{1}^\top \mathbf{A}(\mathbf{x}^* - \bar{\mathbf{x}}) = \mathbf{0}. \tag{3.12}$$

Let $\mathbf{Q}_{11} = \frac{1}{2}\alpha(\mathbf{L} + \mathbf{L}^\top) + \beta \mathbf{A}^\top \mathbb{1}\mathbb{1}^\top \mathbf{A}$. Then (3.12) implies

$$(\mathbf{x}^* - \bar{\mathbf{x}})^\top \mathbf{Q}_{11}(\mathbf{x}^* - \bar{\mathbf{x}}) = 0.$$

Since $\mathcal{G}$ is weight-balanced, $(\mathbf{L} + \mathbf{L}^\top) \geq \mathbf{0}$. This along with the fact that $\mathbf{A}^\top \mathbb{1}\mathbb{1}^\top \mathbf{A} \geq \mathbf{0}$ implies $\mathbf{L}^\top(\mathbf{x}^* - \bar{\mathbf{x}}) = \mathbf{0}$ and $\mathbb{1}^\top \mathbf{A}(\mathbf{x}^* - \bar{\mathbf{x}}) = \mathbf{0}$. Therefore, $\mathbf{x}^* = \mathbf{1} \otimes x^*$, for some $x^* \in \mathbb{R}^m$ which satisfies $Ax^* = A\bar{x} = b$, as claimed. □

The next result characterizes the convergence of (3.11).

Proposition 3.3.2. ***(Exponential stability of*** (3.11)***).*** *Let $\mathcal{G}$ be a strongly connected and weight-balanced digraph. Then for all $\alpha, \beta \in \mathbb{R}_{>0}$, any trajectory of* (3.11) *converges exponentially to a point of the form $\mathbf{x}^* = \mathbf{1} \otimes x^*$, where $x^* \in \mathbb{R}^m$ solves* (3.1)*.*

Proof. Consider a vector $\mathbf{w} \in \mathbb{R}^{mn}$ in the null space of $\mathbf{Q}_{11}$. Using the same line of arguments as

in the proof of Lemma 3.3.1, this implies that $\mathbf{L}^\top \mathbf{w} = \mathbf{0}$ and $\mathbb{1}^\top \mathbf{A}\mathbf{w} = \mathbf{0}$. Therefore, along (3.11),

$$\dot{\mathbf{x}}^\top \mathbf{w} = -(\alpha \mathbf{x}^\top \mathbf{L}^\top + \beta(\mathbf{A}\mathbf{x} - \mathbf{b})^\top \mathbb{1}\mathbb{1}^\top \mathbf{A})\mathbf{w} = 0.$$

This means that the dynamics (3.11) are orthogonal to the null space of $\mathbf{Q}_{11}$ and hence the component of $\mathbf{x}$ in the null space of $\mathbf{Q}_{11}$, say $\mathbf{x}_\parallel$, remains constant. Given the initial condition $\mathbf{x}(0)$, consider the particular equilibrium $\mathbf{x}^*$ of (3.11) satisfying $\mathbf{x}^*_\parallel = \mathbf{x}(0)_\parallel$. Since different equilibria differ only in their null space component, $\mathbf{x}^*$ defined this way is unique. Consider the Lyapunov function candidate $V : \mathbb{R}^{mn} \to \mathbb{R}$

$$V(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^\top (\mathbf{x} - \mathbf{x}^*).$$

The Lie derivative of V along the dynamics (3.11) is given by

$$\begin{aligned}\mathcal{L}_{\psi_{\text{grad}}} V &= -(\mathbf{x} - \mathbf{x}^*)^\top (\alpha \mathbf{L}\mathbf{x} + \beta \mathbf{A}^\top \mathbb{1}\mathbb{1}^\top (\mathbf{A}\mathbf{x} - \mathbf{b})) \\ &= -(\mathbf{x} - \mathbf{x}^*)^\top \mathbf{Q}_{11}(\mathbf{x} - \mathbf{x}^*) \leq -2\lambda_2(\mathbf{Q}_{11})V.\end{aligned}$$

The last inequality follows from applying the Courant-Fischer theorem [HJ85, Theorem 4.2.11] together with the fact that $(\mathbf{x} - \mathbf{x}^*)^\top \mathbf{w} = 0$ as $\mathbf{x}_\parallel$ is constant. Using the monotonicity theorem [HJ85, Corollary 4.3.3], we further have

$$\mathcal{L}_{\psi_{\text{grad}}} V \leq -2 \min\left\{ \frac{1}{2}\alpha \lambda_2(\mathsf{L} + \mathsf{L}^\top), \beta \lambda_2(\mathbf{A}^\top \mathbb{1}\mathbb{1}^\top \mathbf{A}) \right\} V.$$

Hence, the dynamics (3.11) is exponentially stable with a rate depending on $\alpha, \beta, \mathsf{L}$ and $\{A_i\}_{i=1}^n$. □

3.3.2 Distributed Algorithm Over Weight-Balanced Networks

We present a distributed algorithm to find a solution of (3.1), which is based on the centralized algorithm (3.11) and involves employing dynamic average consensus (cf. Section 2.4) to estimate the aggregate $\mathbb{1}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})$. Formally,

$$\dot{\mathbf{x}} = -\alpha \mathbf{L}\mathbf{x} - n\beta \mathbf{A}^\top \mathbf{y}, \tag{3.13a}$$

$$\dot{\mathbf{y}} = -\alpha \mathbf{A}\mathbf{L}\mathbf{x} - n\beta \mathbf{A}\mathbf{A}^\top \mathbf{y} - \gamma \mathbf{L}\mathbf{y}, \tag{3.13b}$$

with design parameter $\gamma > 0$. Here, each agent $i \in \{1, \dots, n\}$ updates $y_i \in \mathbb{R}^m$ which estimates the average mismatch $\frac{1}{n}\mathbb{1}^\top(\mathbf{A}\mathbf{x} - \mathbf{b})$. The dynamics (3.13) is distributed as each agent just needs to know its state and that of its out-neighbors. Whenever convenient, we refer to it as ψ_{gdac}. The following result characterizes the equilibria of (3.13) and shows that the total deviation from the average mismatch is conserved.

Lemma 3.3.3. ***(Equilibria of*** (3.13) ***and invariance of total deviation).*** *Let $\mathcal{G}$ be a strongly connected and weight-balanced digraph. Then, if* $(\mathbf{x}^*, \mathbf{0})$ *is an equilibrium of* (3.13) *then* $\mathbf{x}^* = \mathbf{1} \otimes x^*$*, where* $x^* \in \mathbb{R}^m$*. Moreover, for all* $\alpha, \beta, \gamma \in \mathbb{R}_{>0}$*,* $\mathbb{1}^\top(\mathbf{y} - \mathbf{A}\mathbf{x})$ *remains constant along the evolution of* (3.13)*.*

Proof. Let $(\mathbf{x}^*, \mathbf{0})$ be an equilibrium of (3.13). From (3.13a), it follows that $\mathbf{L}\mathbf{x} = \mathbf{0}$, and hence $\mathbf{x}^* = \mathbf{1} \otimes x^*$ for some $x^* \in \mathbb{R}^m$, establishing the first statement. Now, consider the derivative $\mathbb{1}^\top(\dot{\mathbf{y}} - \mathbf{A}\dot{\mathbf{x}}) = -\gamma\, \mathbb{1}^\top \mathbf{L}\mathbf{y} = \mathbf{0}$. Hence, $\mathbb{1}^\top(\mathbf{y} - \mathbf{A}\mathbf{x})$ is conserved along the evolution of (3.13). □

Remark 3.3.4. ***(Distributed initialization of the*** ψ_{gdac} ***algorithm).*** *From Lemma 3.3.3, we observe that in order for a trajectory of* (3.13) *to converge to an equilibrium of the form* $(\mathbf{x}^*, \mathbf{y}^*) = (\mathbf{1} \otimes x^*, \mathbf{0})$*,*

where $x^* \in \mathbb{R}^m$ solves (3.1)*, its initial condition must satisfy*

$$\mathbb{1}^\top \mathbf{y}(0) = \mathbb{1}^\top(\mathbf{A}\mathbf{x}(0) - \mathbf{b}).$$

This could be implemented in a distributed way if each agent $i \in \{1, \dots, n\}$ chooses its initial states satisfying $y_i(0) = A_i x_i(0) - b_i$. One trivial selection, for example, is $\mathbf{x}(0) = \mathbf{0}$ and $\mathbf{y}(0) = -\mathbf{b}$. •

The next result characterizes the convergence of (3.13).

Theorem 3.3.5. ***(Exponential stability of*** (3.13) ***over balanced networks).*** *Let $\mathcal{G}$ be a strongly connected and weight-balanced digraph and assume* $\operatorname{null}(A) \subseteq \operatorname{null}(A_i)$*, for all $i \in \{1, \dots, n\}$. Let $\alpha, \beta \in \mathbb{R}_{>0}$ and define*

$$\bar{\gamma} = \max\left\{ \frac{2}{\lambda_2(\mathsf{L} + \mathsf{L}^\top)} \lambda_{\max}\left(\frac{\mathbf{Q}_{12}^\top \mathbf{Q}_{12}}{\lambda_2(\mathbf{Q}_{11})} - n\beta \mathbf{A}\mathbf{A}^\top \right), 0 \right\},$$

where $\mathbf{Q}_{11} = \frac{1}{2}\alpha(\mathbf{L} + \mathbf{L}^\top) + \beta \mathbf{A}^\top \mathbb{1}\,\mathbb{1}^\top \mathbf{A}$ and $\mathbf{Q}_{12} = \frac{1}{2}(n\beta \mathbf{A}^\top + \alpha \mathbf{L}^\top \mathbf{A}^\top + \beta \mathbf{A}^\top \mathbb{1}\,\mathbb{1}^\top \mathbf{A}\mathbf{A}^\top)$. Then, for all $\gamma \in (\bar{\gamma}, \infty)$, any trajectory of (3.13) *with initial condition satisfying $\mathbb{1}^\top \mathbf{y}(0) = \mathbb{1}^\top(\mathbf{A}\mathbf{x}(0) - \mathbf{b})$ converges exponentially to $(\mathbf{x}^*, \mathbf{0})$, where $\mathbf{x}^* = \mathbf{1} \otimes x^*$ and $x^* \in \mathbb{R}^m$ solves* (3.1)*.*

Proof. Define the error variable

$$\mathbf{e} = \mathbf{y} - \frac{1}{n}\mathbb{1}\,\mathbb{1}^\top(\mathbf{A}\mathbf{x} - \mathbf{b}), \tag{3.14}$$

measuring the difference between the agents' estimates and the actual value of average mismatch.

Note that

$$\dot{\mathbf{e}} = -\alpha\mathbf{\Pi}\mathbf{A}\mathbf{L}\mathbf{x} - n\beta\mathbf{\Pi}\mathbf{A}\mathbf{A}^\top\mathbf{y} - \gamma\mathbf{L}\mathbf{y},$$

where $\mathbf{\Pi} = I - \frac{1}{n}\mathbb{1}\mathbb{1}^\top$. Rewriting (3.13) in terms of $\mathbf{x}$ and $\mathbf{e}$,

$$\dot{\mathbf{x}} = -\alpha\mathbf{L}\mathbf{x} - \beta\mathbf{A}^\top\mathbb{1}\mathbb{1}^\top(\mathbf{A}\mathbf{x} - \mathbf{b}) - n\beta\mathbf{A}^\top\mathbf{e}, \tag{3.15a}$$

$$\dot{\mathbf{e}} = -\alpha\mathbf{\Pi}\mathbf{A}\mathbf{L}\mathbf{x} - \beta\mathbf{\Pi}\mathbf{A}\mathbf{A}^\top\mathbb{1}\mathbb{1}^\top(\mathbf{A}\mathbf{x} - \mathbf{b}) - n\beta\mathbf{\Pi}\mathbf{A}\mathbf{A}^\top\mathbf{e} - \gamma\mathbf{L}\mathbf{e}. \tag{3.15b}$$

From the proof of Proposition 3.3.2, we know that if $\mathbf{w} \in \mathbb{R}^{mn}$ is in the null space of $\mathbf{Q}_{11}$, then $\mathbf{L}^\top\mathbf{w} = \mathbf{0}$ and $\mathbb{1}^\top\mathbf{A}\mathbf{w} = \mathbf{0}$. Therefore, $\mathbf{w} = \mathbf{1} \otimes w$, where $w \in \mathbb{R}^m$ belongs to $w \in \text{null}(A)$. By hypothesis, $A_i w = \mathbf{0}$ for all $i \in \{1, \dots, n\}$. Therefore, from (3.15a), $\dot{\mathbf{x}}^\top\mathbf{w} = 0$, and the $\mathbf{x}$ component of the equilibrium $(\mathbf{x}^*, \mathbf{y}^*)$ of (3.13) satisfies $\mathbf{x}^*_{\|} = \mathbf{x}(0)_{\|}$ and is unique. With the initialization of the statement, it follows from Lemma 3.3.3 that $\mathbf{y}^* = \mathbf{1} \otimes \frac{1}{n}\mathbb{1}^\top(\mathbf{A}\mathbf{x}^* - \mathbf{b})$. Substituting this value of $\mathbf{y}^*$ in (3.13a) and following the proof of Lemma 3.3.1, one can establish that the corresponding equilibrium is of the form $(\mathbf{1} \otimes x^*, \mathbf{0})$, where $x^* \in \mathbb{R}^m$ is a solution of (3.1).

Consider the Lyapunov function candidate $V_2 : \mathbb{R}^{2mn} \to \mathbb{R}$

$$V_2(\mathbf{x}, \mathbf{e}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^*)^\top(\mathbf{x} - \mathbf{x}^*) + \frac{1}{2}\mathbf{e}^\top\mathbf{e}.$$

The Lie derivative of V_2 along (3.15) is given by

$$\begin{aligned}\mathcal{L}_{\psi_{\text{gdac}}} V_2 =& -(\mathbf{x}-\mathbf{x}^*)^\top(\alpha\mathbf{L}\mathbf{x}+\beta\mathbf{A}^\top\mathbb{1}\mathbb{1}^\top(\mathbf{A}\mathbf{x}-\mathbf{b}))-n\beta(\mathbf{x}-\mathbf{x}^*)^\top\mathbf{A}^\top\mathbf{e}-\mathbf{e}^\top\mathbf{\Pi}\mathbf{A}(\alpha\mathbf{L}\mathbf{x}+n\beta\mathbf{A}^\top\mathbf{e})\\ &-\mathbf{e}^\top(\beta\mathbf{\Pi}\mathbf{A}\mathbf{A}^\top\mathbb{1}\mathbb{1}^\top(\mathbf{A}\mathbf{x}-\mathbf{b})+\gamma\mathbf{L}\mathbf{e})\\ =& -\begin{bmatrix}\mathbf{x}-\mathbf{x}^*\\ \mathbf{e}\end{bmatrix}^\top\begin{bmatrix}\mathbf{Q}_{11} & \mathbf{Q}_{12}\\ \mathbf{Q}_{12}^\top & \mathbf{Q}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{x}-\mathbf{x}^*\\ \mathbf{e}\end{bmatrix},\end{aligned}$$

where $\mathbf{Q}_{22}=\frac{1}{2}\gamma(\mathbf{L}+\mathbf{L}^\top)+n\beta\mathbf{A}\mathbf{A}^\top$ and we have used the fact that due to the mentioned initialization, $\mathbb{1}^\top\mathbf{e}=\mathbf{0}$ from Lemma 3.3.3. Since $\mathbf{x}_\parallel$ is constant, $(\mathbf{x}-\mathbf{x}^*)^\top\mathbf{w}=0$ and from the Courant-Fischer theorem [HJ85, Theorem 4.2.11],

$$-(\mathbf{x}-\mathbf{x}^*)^\top\mathbf{Q}_{11}(\mathbf{x}-\mathbf{x}^*)\leq-\lambda_2(\mathbf{Q}_{11})(\mathbf{x}-\mathbf{x}^*)^\top(\mathbf{x}-\mathbf{x}^*).$$

Also, since $\mathbb{1}^\top\mathbf{e}=\mathbf{0}$ and $\mathcal{G}$ is weight-balanced, it again follows from the Courant-Fischer theorem that

$$-\mathbf{e}^\top\mathbf{Q}_{22}\mathbf{e}\leq-\frac{1}{2}\gamma\lambda_2(\mathsf{L}+\mathsf{L}^\top)\mathbf{e}^\top\mathbf{e}-n\beta\mathbf{e}^\top\mathbf{A}\mathbf{A}^\top\mathbf{e}.$$

Therefore, we can upper bound the Lie derivative as

$$\mathcal{L}_{\psi_{\text{gdac}}} V_2\leq-\begin{bmatrix}\mathbf{x}-\mathbf{x}^*\\ \mathbf{e}\end{bmatrix}^\top\underbrace{\begin{bmatrix}\lambda_2(\mathbf{Q}_{11})I & \mathbf{Q}_{12}\\ \mathbf{Q}_{12}^\top & \bar{\mathbf{Q}}_{22}\end{bmatrix}}_{\bar{\mathbf{Q}}}\begin{bmatrix}\mathbf{x}-\mathbf{x}^*\\ \mathbf{e}\end{bmatrix},$$

where $\bar{\mathbf{Q}}_{22} = \frac{1}{2}\gamma\lambda_2(\mathsf{L}+\mathsf{L}^\top)I + n\beta\mathbf{A}\mathbf{A}^\top$. Next, we examine the positive definiteness of $\bar{\mathbf{Q}}$. Using the Schur complement [BV09], $\bar{\mathbf{Q}} \succ \mathbf{0}$ iff

$$\frac{1}{2}\gamma\lambda_2(\mathsf{L}+\mathsf{L}^\top)I + n\beta\mathbf{A}\mathbf{A}^\top - \frac{1}{\lambda_2(\mathbf{Q}_{11})}\mathbf{Q}_{12}^\top\mathbf{Q}_{12} \succ \mathbf{0}.$$

Hence, $\bar{\mathbf{Q}} \succ \mathbf{0}$ if $\gamma > \bar{\gamma}$, and $\mathcal{L}_{\psi_{\text{gdac}}} V_2 \leq -2\lambda_{\min}(\bar{\mathbf{Q}})V_2$. □

The null space condition in Theorem 3.3.5 makes sure that $\mathbf{x}_\parallel^*$ remains invariant along the evolution of (3.13) and all the agents approach the solution of (3.1) closest to $\mathbf{x}(0)$. This condition is automatically satisfied if the matrix A is full rank, or in other words, equation (3.1) has a unique solution. We believe (and simulations also suggest) that if this condition is not satisfied, the $\mathbf{x}$ component of the dynamics still converges to a solution of (3.1).

Remark 3.3.6. ***(Lower bound on γ).*** *The lower bound $\bar{\gamma}$ in Theorem 3.3.5 is conservative in general. In fact, the algorithm may converge even if this condition is not satisfied, something that we have observed in simulation. Note also that although α and β are free parameters, they should still be carefully chosen as $\bar{\gamma}$ depends on them.* •

The result above can be extended to time-varying networks. In case $\mathcal{G}(t)$ is time-varying, the algorithm in (3.13) reads as

$$\dot{\mathbf{x}} = -\alpha\mathbf{L}(t)\mathbf{x} - n\beta\mathbf{A}^\top\mathbf{y}, \tag{3.16a}$$

$$\dot{\mathbf{y}} = -\alpha\mathbf{A}\mathbf{L}(t)\mathbf{x} - n\beta\mathbf{A}\mathbf{A}^\top\mathbf{y} - \gamma\mathbf{L}(t)\mathbf{y}. \tag{3.16b}$$

The next result formally characterizes the convergence of (3.16). Its proof is similar to that of Theorem 3.3.5 and hence omitted.

Theorem 3.3.7. ***(Exponential stability of*** (3.16) ***over time-varying balanced networks).*** *Let* $\{\mathcal{G}(t)\}_{t=0}^{\infty}$ *be a sequence of strongly connected and weight-balanced digraphs with uniformly bounded edge weights (i.e., there exists* $a \in (0, \infty)$ *such that* $\mathsf{A}_{ij}(t) < a$ *for all* (i, j) *and* $t \geq 0$*), and assume* $\text{null}(A) \subseteq \text{null}(A_i)$*, for all* $i \in \{1, \dots, n\}$*. Let* $\alpha, \beta \in \mathbb{R}_{>0}$ *and define* $\bar{\gamma}(t)$ *as*

$$\max\left\{\frac{2}{\lambda_2(\mathsf{L}(t)+\mathsf{L}(t)^\top)}\lambda_{\max}\left(\frac{\mathbf{Q}_{12}(t)^\top\mathbf{Q}_{12}(t)}{\lambda_2(\mathbf{Q}_{11}(t))}-n\beta\mathbf{A}\mathbf{A}^\top\right),0\right\},$$

where $\mathbf{Q}_{11}(t)=\frac{1}{2}\alpha(\mathbf{L}(t)+\mathbf{L}(t)^\top)+\beta\mathbf{A}^\top\mathbb{1}\mathbb{1}^\top\mathbf{A}$ *and* $\mathbf{Q}_{12}(t)=\frac{1}{2}(n\beta\mathbf{A}^\top+\alpha\mathbf{L}(t)^\top\mathbf{A}^\top+\beta\mathbf{A}^\top\mathbb{1}\mathbb{1}^\top\mathbf{A}\mathbf{A}^\top)$*. Then for all* $\gamma \in (\hat{\gamma}, \infty)$*, where* $\hat{\gamma} = \sup_{t\geq 0}\bar{\gamma}(t)$*, any trajectory of* (3.16) *with initial conditions* $\mathbb{1}^\top\mathbf{y}(0) = \mathbb{1}^\top(\mathbf{A}\mathbf{x}(0)-\mathbf{b})$ *converges exponentially to* $(\mathbf{x}^*, \mathbf{0})$*, where* $\mathbf{x}^* = \mathbf{1}\otimes x^*$ *and* $x^* \in \mathbb{R}^m$ *solves* (3.1)*.*

3.3.3 Distributed Algorithms Over Unbalanced Networks

In this section, we extend our approach to solve problem (3.1) over graphs that are not necessarily balanced. In those scenarios, since $\mathsf{L}\mathbf{1} \neq \mathbf{0}$, the one-to-one correspondence between the desired equilibria of (3.11) or (3.13) and the solutions of (3.1) does not hold anymore. To overcome this, we propose

$$\dot{\mathbf{x}} = -\alpha\mathbf{L}\bar{\mathbf{V}}\mathbf{x} - n\beta\mathbf{A}^\top\mathbf{y}, \tag{3.17a}$$

$$\dot{\mathbf{y}} = -\alpha\mathbf{A}\mathbf{L}\bar{\mathbf{V}}\mathbf{x} - n\beta\mathbf{A}\mathbf{A}^\top\mathbf{y} - \gamma\mathbf{L}\bar{\mathbf{V}}\mathbf{y}, \tag{3.17b}$$

where $\bar{\mathbf{V}} = \text{diag}(\bar{\mathbf{v}})$, $\bar{\mathbf{v}} = \mathbf{1}\otimes\bar{v}$, and $\bar{v}$ is a positive right eigenvector with eigenvalue 0 of L. Exponential stability of (3.17) can be established by interpreting $\mathsf{L}\cdot\text{diag}(\bar{v})$ as the Laplacian of a

weight-balanced graph and then following the same steps as in the proof of Theorem 3.3.5, but we omit it here for reasons of space. Although (3.17) is distributed, it assumes that agents have a priori knowledge of the corresponding entries of $\bar{v}$ which might be limiting in practice. To deal with this limitation, we propose an algorithm that does not require such knowledge by augmenting (3.17) with an additional dynamics converging to $\bar{\mathbf{v}}$,

$$\dot{\mathbf{x}} = -\alpha \mathbf{L}\mathbf{V}\mathbf{x} - n\beta \mathbf{A}^\top \mathbf{y}, \tag{3.18a}$$

$$\dot{\mathbf{y}} = -\alpha \mathbf{A}\mathbf{L}\mathbf{V}\mathbf{x} - n\beta \mathbf{A}\mathbf{A}^\top \mathbf{y} - \gamma \mathbf{L}\mathbf{V}\mathbf{y}, \tag{3.18b}$$

$$\dot{\mathbf{v}} = -\mathbf{L}\mathbf{v}, \tag{3.18c}$$

where $\mathbf{V} = \operatorname{diag}(\mathbf{v})$. Whenever convenient, we refer to dynamics (3.18) as ψ_{dist}. Note that, unlike all the dynamics discussed so far, ψ_{dist} is nonlinear.

Remark 3.3.8. ***(Distributed nature of*** (3.18)***).*** *The dynamics* (3.18) *is out-distributed, but requires each agent $i \in \{1, \dots, n\}$ to have knowledge of its in-degree because $\mathsf{L} = \mathsf{D}^{\text{in}} - \mathsf{A}$ and the graph is not weight-balanced. If we use instead the out-Laplacian $\mathsf{L} = \mathsf{D}^{\text{out}} - \mathsf{A}$, then one could still define an equivalent algorithm for* (3.17) *with $\mathbf{L}\bar{\mathbf{V}}$ replaced by $\bar{\mathbf{V}}\mathbf{L}$, but* (3.18c) *would look like $\dot{\mathbf{v}} = -\mathbf{L}^\top \mathbf{v}$, which would require state information from in-neighbors too.* •

The next result characterizes the convergence of (3.18).

Theorem 3.3.9. ***(Exponential stability of*** (3.18) ***over unbalanced networks).*** *Let $\mathcal{G}$ be a strongly connected digraph and assume* $\operatorname{null}(A) \subseteq \operatorname{null}(A_i)$*, for all $i \in \{1, \dots, n\}$. Let $\alpha, \beta \in \mathbb{R}_{>0}$ and define*

$$\bar{\gamma} = \max \left\{ \frac{2}{\lambda_2(\mathsf{L}\bar{V} + \bar{V}\mathsf{L}^\top)} \lambda_{\max} \left(\frac{\mathbf{Q}_{12}^\top \mathbf{Q}_{12}}{\lambda_2(\mathbf{Q}_{11})} - n\beta \mathbf{A}\mathbf{A}^\top \right), 0 \right\},$$

where $\mathbf{Q}_{11} = \frac{1}{2}(\alpha\mathbf{L}\bar{\mathbf{V}} + \bar{\mathbf{V}}\mathbf{L}^\top) + \beta\mathbf{A}^\top \mathbb{1}\mathbb{1}^\top\mathbf{A}$, $\mathbf{Q}_{12} = \frac{1}{2}(n\beta\mathbf{A}^\top + \alpha\bar{\mathbf{V}}\mathbf{L}^\top\mathbf{A}^\top + \beta\mathbf{A}^\top\mathbb{1}\mathbb{1}^\top\mathbf{A}\mathbf{A}^\top)$, $\bar{v}$ *is the positive eigenvector with eigenvalue* 0 *of* L *satisfying* $\mathbf{1}^\top\bar{v} = 1$, *and* $\bar{V} = diag(\bar{v})$. *Then, for all* $\gamma \in (\bar{\gamma}, \infty)$, *any trajectory of* (3.18) *with initial condition satisfying* $\mathbb{1}^\top\mathbf{y}(0) = \mathbb{1}^\top(\mathbf{A}\mathbf{x}(0) - \mathbf{b})$ *and* $\mathbf{v}(0) = \frac{1}{n}\mathbf{1}$, *converges exponentially to* $(\mathbf{x}^*, \mathbf{0}, \bar{\mathbf{v}})$, *where* $\mathbf{x}^* = \mathbf{1} \otimes x^*$ *and* $x^* \in \mathbb{R}^m$ *solves* (3.1), *and* $\bar{\mathbf{v}} = \mathbf{1} \otimes \bar{v}$.

Proof. From [TG20, Proposition 2.2], we have that $\mathbf{v}(t) > \mathbf{0}$ for all $t \geq 0$. Also, since $\mathbf{1}^\top\mathsf{L} = \mathbf{0}$, $\mathbf{1}^\top\mathbf{v}$ is conserved along the evolution of (3.18c). Hence $\mathbf{v}(t) \to \bar{\mathbf{v}}$ exponentially fast with a rate determined by the non-zero eigenvalue of L with the smallest real part. Let us interpret the dynamics (3.18a)-(3.18b) as the dynamics (3.17) with some disturbance $\mathbf{d}(t)$ defined by

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}^{\mathbf{x}} \\ \mathbf{d}^{\mathbf{y}} \end{bmatrix} = \begin{bmatrix} -\alpha\mathbf{L}(\mathbf{V} - \bar{\mathbf{V}})\mathbf{x} \\ -\alpha\mathbf{A}\mathbf{L}(\mathbf{V} - \bar{\mathbf{V}})\mathbf{x} - \gamma\mathbf{L}(\mathbf{V} - \bar{\mathbf{V}})\mathbf{y} \end{bmatrix},$$

which goes to $\mathbf{0}$ as $t \to \infty$. Consider a vector $\mathbf{w} \in \text{null}(\mathbf{Q}_{11})$. Then as in the proof of Theorem 3.3.5, $\mathbf{w} = \mathbf{1} \otimes w$, where $w \in \text{null}(A)$ and by hypothesis, $A_i w = \mathbf{0}$ for all $i \in \{1, \dots, n\}$. Since $\mathbb{1}^\top\mathbf{L} = \mathbf{0}$, therefore, $\mathbf{w}^\top\mathbf{d}^{\mathbf{x}} = 0$ and we still have $\mathbf{w}^\top\dot{\mathbf{x}} = 0$, and the $\mathbf{x}$ component of the equilibrium $(\mathbf{x}^*, \mathbf{y}^*, \bar{\mathbf{v}})$ of (3.18) satisfies $\mathbf{x}^*_{\|} = \mathbf{x}(0)_{\|}$ and is unique. With the initialization of the statement and following the same steps as in the proof of Lemma 3.3.3, one can establish that $\mathbf{y}^* = \mathbf{1} \otimes \frac{1}{n}\mathbb{1}^\top(\mathbf{A}\mathbf{x}^* - \mathbf{b})$. Substituting this value of $\mathbf{y}^*$ in (3.18a) and following the proof of Lemma 3.3.1, one can establish that the corresponding equilibrium is of the form $(\mathbf{1}\otimes x^*, \mathbf{0}, \bar{\mathbf{v}})$, where $x^* \in \mathbb{R}^m$ is a solution of (3.1). Consider now the Lyapunov function candidate $V_3 : \mathbb{R}^{3mn} \to \mathbb{R}$

$$V_3(\mathbf{x}, \mathbf{e}, \mathbf{v}) = V_2(\mathbf{x}, \mathbf{e}) + \frac{\delta}{2}(\mathbf{v} - \bar{\mathbf{v}})^\top\mathbf{P}(\mathbf{v} - \bar{\mathbf{v}}),$$

where $\delta > 0$, $\mathbf{P} = \bar{\mathbf{V}}^{-1}$, $\mathbf{e}$ is defined as in (3.14), and V_2 is the same function as in the proof of Theorem 3.3.5. The Lie derivative of V_3 along (3.18) is given by

$$\mathcal{L}_{\psi_{\text{dist}}} V_3 = -\begin{bmatrix} \mathbf{x}-\mathbf{x}^* \\ \mathbf{e} \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{12}^\top & \mathbf{Q}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}-\mathbf{x}^* \\ \mathbf{e} \end{bmatrix} + (\mathbf{x}-\mathbf{x}^*)^\top \mathbf{d}^{\mathbf{x}}$$
$$+ \mathbf{e}^\top \mathbf{d}^{\mathbf{e}} - \delta(\mathbf{v}-\bar{\mathbf{v}})^\top (\mathbf{L}^\top \mathbf{P} + \mathbf{P}\mathbf{L})(\mathbf{v}-\bar{\mathbf{v}}),$$

where $\mathbf{d}^{\mathbf{e}} = -\alpha \mathbf{\Pi}\mathbf{A}\mathbf{L}(\mathbf{V}-\bar{\mathbf{V}})\mathbf{x} - \gamma \mathbf{L}(\mathbf{V}-\bar{\mathbf{V}})\mathbf{e}$, and $\mathbf{Q}_{22} = \frac{1}{2}\gamma(\mathbf{L}\bar{\mathbf{V}} + \bar{\mathbf{V}}\mathbf{L}^\top) + n\beta\mathbf{A}\mathbf{A}^\top$. Interestingly, $\mathsf{L}\,\bar{V}$ can be interpreted as the Laplacian of a weight-balanced graph and as a result, $\mathbf{L}\bar{\mathbf{V}} + \bar{\mathbf{V}}\mathbf{L}^\top \succeq \mathbf{0}$ implying that $\mathbf{L}^\top\mathbf{P} + \mathbf{P}\mathbf{L} \succeq \mathbf{0}$. Once again, following Lemma 3.3.3, one can establish that with the initialization of the statement, $\mathbb{1}^\top \mathbf{e} = \mathbf{0}$ and therefore using the Courant-Fischer theorem [HJ85, Theorem 4.2.11] together with the fact that $(\mathbf{x}-\mathbf{x}^*)^\top \mathbf{w} = 0$ due to invariance of $\mathbf{x}_{\|}$, we can upper bound the Lie derivative as

$$\mathcal{L}_{\psi_{\text{dist}}} V_3 \leq -\begin{bmatrix} \mathbf{x}-\mathbf{x}^* \\ \mathbf{e} \end{bmatrix}^\top \underbrace{\begin{bmatrix} \lambda_2(\mathbf{Q}_{11}) I & \mathbf{Q}_{12} \\ \mathbf{Q}_{12}^\top & \bar{\mathbf{Q}}_{22} \end{bmatrix}}_{\bar{\mathbf{Q}}} \begin{bmatrix} \mathbf{x}-\mathbf{x}^* \\ \mathbf{e} \end{bmatrix}$$
$$+ \alpha\|\mathbf{x}-\mathbf{x}^*\|\|\mathbf{L}\|\|\mathbf{v}-\bar{\mathbf{v}}\|(\|\mathbf{x}-\mathbf{x}^*\| + \|\mathbf{x}^*\|)$$
$$+ \alpha\|\mathbf{e}\|\|\mathbf{\Pi}\mathbf{A}\mathbf{L}\|\|\mathbf{v}-\bar{\mathbf{v}}\|(\|\mathbf{x}-\mathbf{x}^*\| + \|\mathbf{x}^*\|) + \gamma\|\mathbf{e}\|\|\mathbf{L}\|\|\mathbf{v}-\bar{\mathbf{v}}\|\|\mathbf{e}\|$$
$$- \delta\lambda_2(\mathbf{L}^\top\mathbf{P} + \mathbf{P}\mathbf{L})\|\mathbf{v}-\bar{\mathbf{v}}\|^2,$$

where $\bar{\mathbf{Q}}_{22} = \frac{1}{2}\gamma\lambda_2(\mathsf{L}\,\bar{V} + \bar{V}\,\mathsf{L}^\top)I + n\beta\mathbf{A}\mathbf{A}^\top$. Define $\mathbf{z} = [\|\mathbf{x}-\mathbf{x}^*\|; \|\mathbf{e}\|; \|\mathbf{v}-\bar{\mathbf{v}}\|]$. If $\gamma > \bar{\gamma}$, then

$\bar{\mathbf{Q}} \succ \mathbf{0}$ and from the Courant-Fischer theorem, we have

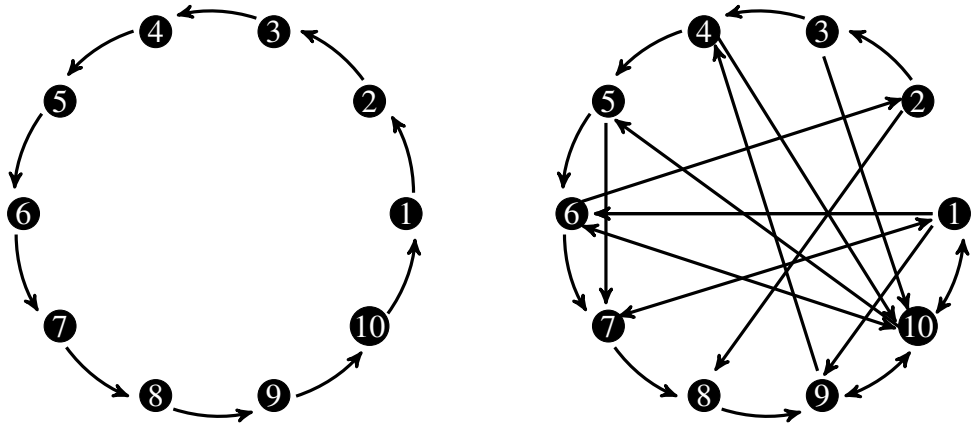
$$\mathcal{L}_{\psi_{\text{dist}}} V_3 \leq -\mathbf{z}^\top \underbrace{\begin{bmatrix} \lambda_{\min}(\bar{\mathbf{Q}}) & 0 & \hat{\mathbf{Q}}_{13}(\mathbf{z}) \\ 0 & \lambda_{\min}(\bar{\mathbf{Q}}) & \hat{\mathbf{Q}}_{23}(\mathbf{z}) \\ \hat{\mathbf{Q}}_{13}(\mathbf{z}) & \hat{\mathbf{Q}}_{23}(\mathbf{z}) & \delta\lambda_2(\mathbf{L}^\top\mathbf{P} + \mathbf{P}\mathbf{L}) \end{bmatrix}}_{\hat{\mathbf{Q}}(\mathbf{z})} \mathbf{z},$$

where $\hat{\mathbf{Q}}_{23}(\mathbf{z}) = -\frac{1}{2}\alpha\|\mathbf{\Pi}\mathbf{A}\mathbf{L}\|(\mathbf{z} + \|\mathbf{x}^*\|) - \frac{1}{2}\gamma\|\mathbf{L}\|\mathbf{z}$ and $\hat{\mathbf{Q}}_{13}(\mathbf{z}) = -\frac{1}{2}\alpha\|\mathbf{L}\|(\mathbf{z} + \|\mathbf{x}^*\|)$. Using the Schur complement, one can verify that for a given value of $\mathbf{z}$, $\hat{\mathbf{Q}}(\mathbf{z}) \succ \mathbf{0}$ iff $\delta > \bar{\delta}(\mathbf{z}) = \dfrac{1}{\lambda_{\min}(\bar{\mathbf{Q}})\lambda_2(\mathbf{L}^\top\mathbf{P} + \mathbf{P}\mathbf{L})}(\hat{\mathbf{Q}}_{13}(\mathbf{z})^2 + \hat{\mathbf{Q}}_{23}(\mathbf{z})^2)$. Hence, if $\delta > \bar{\delta}(\mathbf{z}(0))$, then $\mathcal{L}_{\psi_{\text{dist}}} V_3 \leq -\lambda_{\min}(\hat{\mathbf{Q}}(\mathbf{z}(0)))\mathbf{z}^\top\mathbf{z}$. This along with the fact that $\frac{1}{2}\min\{1, \delta\lambda_{\min}(\mathbf{P})\}\|\mathbf{z}\|^2 \leq V_3 \leq \frac{1}{2}\max\{1, \delta\lambda_{\max}(\mathbf{P})\}\|\mathbf{z}\|^2$, implies that V_3 satisfies the hypotheses of [Kha02, Theorem 4.10] for exponential stability. □

The exponential convergence of algorithms (3.11) and (3.13) for weight-balanced graphs, and (3.17) for unbalanced graphs follows from their linear nature. For algorithm (3.18), exponential convergence could be attributed to the fact that the dynamics (3.18c) converge exponentially and hence, after some time, (3.18a)-(3.18b) and (3.17) are essentially the same.

3.4 Simulations

We consider 10 agents communicating over the digraphs shown in Fig. 3.1, seeking to solve problem (3.1) with $\{A_i\}_{i=1}^{10} \in \mathbb{R}^{5\times5}$ and $\{b_i\}_{i=1}^{10} \in \mathbb{R}^5$. Since the proposed dynamics are in continuous time, we use a first-order Euler discretization with stepsize 2.5×10^{-3} for the MATLAB implementation. The values of α, β and γ are 2, 10^{-1} and 20, respectively. The edge weights for



(a) $\mathcal{G}_1$ (b) $\mathcal{G}_2$

Figure 3.1: Communication topologies among the agents. The edge weights are adjusted to make the graphs either weight-balanced or unbalanced, as needed.

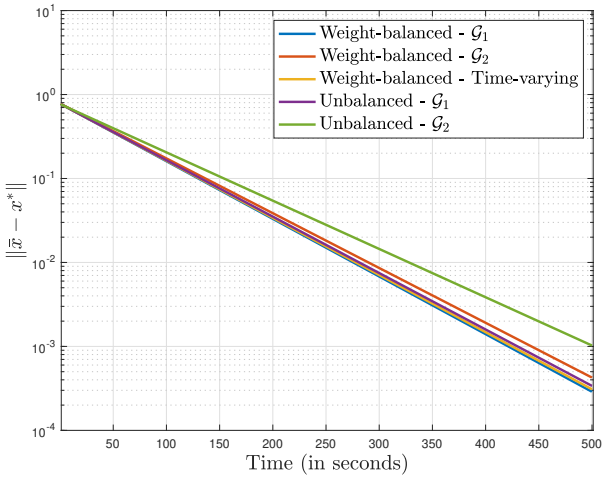


Figure 3.2: Evolution of the error between the actual solution and the average state using the proposed algorithms (3.13), (3.16) and (3.17) over the graphs shown in Fig. 3.1. Straight lines correspond to exponential convergence.

various cases are adjusted to make the graphs weight-balanced and unbalanced, as needed. For the time-varying case, at every iteration, the communication graph is switched randomly between $\mathcal{G}_1$ and $\mathcal{G}_2$. In Fig. 3.2, we plot the evolution of the error between the actual solution of (3.1) and the average state $\bar{x} = \frac{1}{n}\mathbb{1}^\top \mathbf{x}$ using (3.13), (3.16) and (3.18). The initial conditions for all the algorithms are chosen according to Remark 3.3.4 as $\mathbf{x}(0) = \mathbf{0}$ and $\mathbf{y}(0) = -\mathbf{b}$. Even though $\mathcal{G}_2$ (with 4.6 as

the minimum of the real parts of non-zero eigenvalues of L and $\lambda_2(L + L^\top) = 7.6$, for the weight-balanced case) is more connected than $\mathcal{G}_1$ (with 1.9 as the minimum of the real parts of non-zero eigenvalues of L and $\lambda_2(L + L^\top) = 3.8$, for the weight-balanced case), convergence is slower. The error in the time-varying case is lower and upper bounded by the error for $\mathcal{G}_1$ and $\mathcal{G}_2$, resp.

Acknowledgements

This chapter, in part, is a reprint of the material [SC22] as it appears in 'Solving linear equations with separable problem data over directed networks' by P. Srivastava and J. Cortés, in the IEEE Control Systems Letters, 2022, as well as [SC21b] where it appears as 'Network optimization via smooth exact penalty functions enabled by distributed gradient computation' by P. Srivastava and J. Cortés in the IEEE Transactions on Control of Network Systems, 2021. The dissertation author was the primary investigator and author of these papers. This work was partially supported by the ARPA-e NODES program, Cooperative Agreement DE-AR0000695 and NSF Award ECCS-1917177.

Chapter 4

Network Optimization via Smooth Penalty Functions

This chapter proposes a distributed algorithm for a network of agents to solve an optimization problem with separable objective function and locally coupled constraints. Our strategy is based on reformulating the original constrained problem as the unconstrained optimization of a smooth (continuously differentiable) exact penalty function. Computing the gradient of this penalty function in a distributed way is challenging even under the separability assumptions on the original optimization problem. Our technical approach shows that the distributed computation problem for the gradient can be formulated as a system of linear algebraic equations defined by separable problem data which can be solved using the algorithms in Chapter 3. We employ this strategy to compute the gradient of the penalty function at the current network state. Our distributed algorithmic solver for the original constrained optimization problem interconnects this estimation with the prescription of having the agents follow the resulting direction.

4.1 Problem Statement

We consider separable network optimization problems where the overall objective function is the aggregate of individual objective functions, one per agent, and the constraints are locally expressible. Formally, consider a group of $n \in \mathbb{N}$ agents whose interaction is modeled by an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each agent $i \in \mathcal{V}$ is responsible for a decision variable $x_i \in \mathbb{R}$. Agent i is equipped with a twice continuously differentiable function $f_i : \mathbb{R} \to \mathbb{R}$. The optimization problem takes the form

$$\begin{aligned} \min_{x \in D} \quad & f(x) = \sum_{i=1}^{n} f_i(x_i) \\ \text{s.t.} \quad & g(x) \leq 0, \ h(x) = 0, \end{aligned} \tag{4.1}$$

with twice continuously differentiable vector-valued functions $g : \mathbb{R}^n \to \mathbb{R}^m$, $h : \mathbb{R}^n \to \mathbb{R}^p$, and $p \leq n$. Each component $\{g_j : \mathbb{R}^n \to \mathbb{R}\}_{j=1}^{m}$ and $\{h_k : \mathbb{R}^n \to \mathbb{R}\}_{k=1}^{p}$ of the constraint functions is locally expressible. Such kind of coupled constraints arise in numerous applications, such as power [DZG13], communication [KMT98], and transportation [BSC15] networks, to name only a few. By locally expressible, we mean that, for each constraint, e.g., g_j, there exists an agent, which we term corresponding agent, such that the function g_j depends on the state of the corresponding agent and its 1-hop neighbors' state. We assume that all the agents involved in a constraint know the functional form of the constraint and its derivatives. According to this definition, different constraints might have different corresponding agents. Under this structure, agents require up to 2-hop communication to evaluate any constraint in which they are involved (1-hop communication in the case of the corresponding agent, 2-hop communication in the case of the other agents involved in the constraint).

Our aim is to develop a smooth distributed algorithm to find an optimizer of the constrained problem (4.1). Our solution strategy employs a continuously differentiable exact penalty function, cf. Section 2.4, to reformulate the problem as an unconstrained optimization one. We then face the task of implementing its gradient dynamics in a distributed way. To do so, we show that the problem of distributed calculation of Lagrange multiplier functions and other necessary terms in the gradient of the penalty function can be formulated as a linear algebraic equation with separable data (cf. Section 4.2) that can be solved in a distributed manner (cf. Chapter 3). Based on this, we propose a distributed algorithmic solution based on smooth gradient descent to solve (4.1). But before we go there, let us characterize the extent to which the gradient descent dynamics of the penalty function satisfies the constraints while finding the optimizers of the original constrained optimization problem.

Proposition 4.1.1. ***(Constraint satisfaction under gradient dynamics of penalty function).*** *Given the optimization problem* (2.4)*, assume LICQ is satisfied at all $x \in \mathcal{D}$. Consider the gradient dynamics $\dot{x} = -\nabla f^{\epsilon}(x)$ of the penalty function f^{ϵ} in* (2.10)*. Then, if at any time t_0, $x(t_0) \in \mathcal{F}$, we have*

(i) (Equality constraints): $x(t) \in \mathcal{F}$, for all $t \geq t_0$ and all $\epsilon > 0$ if the problem (2.4) *has just equality constraints;*

(ii) (Scalar inequality constraint): there exists $\bar{\epsilon} > 0$ such that $x(t) \in \mathcal{F}$, for all $t \geq t_0$ and all $\epsilon \in (0, \bar{\epsilon}]$ if the problem (2.4) *has only one inequality constraint;*

(iii) (General constraints): in general, there is no guarantee that the evolution of the gradient dynamics stays feasible when the problem (2.4) *has more than one constraint if one of them is an inequality.*

Proof. To prove the result, we examine the Lie derivative of the constraint functions along the dynamics. We consider the different cases below:

(Equality constraints): Given the constraint function h, consider the Lie derivative over the set $\mathcal{F}$,

$$\begin{aligned} L_{-\nabla f^\epsilon} h(x) &= -\nabla h(x)^\top \left(\nabla f(x) + \nabla h(x)\mu(x) + \nabla \mu(x) h(x) + \frac{2}{\epsilon}\nabla h(x) h(x)\right) \\ &= -\nabla h(x)^\top (\nabla f(x) + \nabla h(x)\mu(x)) \end{aligned}$$

where we have used the fact that $h(x) = 0$ for $x \in \mathcal{F}$. Substituting the value of $\mu(x)$ from (2.7),

$$L_{-\nabla f^\epsilon} h(x) = -\nabla h(x)^\top (\nabla f(x) - \nabla h(x) N(x)^{-1} \nabla h(x)^\top \nabla f(x)) = 0.$$

This means that the constraint function remains constant along the gradient dynamics over $\mathcal{F}$. Hence, $x(t) \in \mathcal{F}$ for all $t \geq t_0$ regardless of the value of ϵ.

(Scalar inequality constraint): With only one inequality constraint defined by a scalar-valued function g, we have $x \in \mathcal{F}$ iff $g(x) \leq 0$. To determine the invariance of the feasibility set, we only need to look at points where $g(x) = 0$. In this case, the Lie derivative is

$$L_{-\nabla f^\epsilon} g(x) = -\nabla g(x)^\top \left(\nabla \lambda(x) + \frac{2}{\epsilon}\nabla g(x)\right) y^{\epsilon^2}(x),$$

where we have already used the fact that $g(x) = 0$ and the definition of $\lambda(x)$ from (2.7). Due to LICQ assumption, $\nabla g(x)^\top \nabla g(x) > 0$, and $y^{\epsilon^2}(x) \geq 0$. Since $\nabla \lambda$ is continuous, it is bounded over the compact set $\mathcal{D}$. Hence, there exists $\bar{\epsilon}$ such that for all $\epsilon \in (0, \bar{\epsilon}]$, $L_{-\nabla f^\epsilon} g(x) \leq 0$ for all x such

that $g(x) = 0$. This means that $x(t) \in \mathcal{F}$ for all $t \geq t_0$.

(General constraints): Here we provide a counterexample for the case with multiple inequality constraints (a similar one can be constructed for the case of both equality and inequality constraints). Consider now a vector-valued function g. The expression of the Lie derivative evaluated at x such that $g(x) = 0$ is

$$L_{-\nabla f^\epsilon} g = -\nabla g(x)^\top \Big(\nabla \lambda(x) + \frac{2}{\epsilon} \nabla g(x) \Big) Y^\epsilon(x) y^\epsilon(x).$$

The LICQ assumption implies that $\nabla g(x)^\top \nabla g(x)$ is positive definite. However, in general, this is not sufficient to ensure that the trajectory of the gradient dynamics starting from x will remain in $\mathcal{F}$. To see this, consider the following example.

$$\begin{aligned} \min_x \quad & (x_1 - 1)^2 + (x_2 + 1)^2 \\ \text{s.t.} \quad & x_1 - 6x_2 \leq 0 \\ & -x_1 + x_2 \leq 0 \end{aligned}$$

Take $x = (0; 0)$, where $g(x) = 0$. After some calculations, it can be verified that $\lambda(x) = (0; -2)$ and $Y^\epsilon(x) y^\epsilon(x) = (0; \epsilon)$. As a result, $\nabla g(x)^\top \nabla g(x) Y^\epsilon(x) y^\epsilon(x) = (-7\epsilon; 2\epsilon)$ and $L_{-\nabla f^\epsilon} g = (14; 2\epsilon - 4)$. The first component of $L_{-\nabla f^\epsilon} g$ is independent of ϵ. This means that no matter what value of ϵ we choose, $L_{-\nabla f^\epsilon} g \not\leq 0$ when $g(x) = 0$. Hence, the feasible set is not invariant. □

Remark 4.1.2. ***(Alternative approaches).*** *To solve problem* (4.1) *in a distributed way, we can instead construct the Lagrangian and then use primal-dual (also known as saddle-point) dynamics [AHU58, FP10, CGC17]. This dynamics uses gradient descent in the primal variable and gra-*

dient ascent in the dual variable. For the problem structure described above, these dynamics is distributed (requiring up to 2-hop communication). However, the dynamics is in general slow, exhibits oscillations in the distance from the feasible set, and there is no guarantee of satisfying the constraints during the evolution, even if the initial state is feasible. Also, it is not clear how to apply accelerated methods, cf. [Nes83] to the primal-dual approach. Another approach to solve (4.1) *in an (up to 2-hop) distributed way consists of reformulating the problem as an unconstrained optimization [XB06, JJ09, CC15] by adding to the original objective function non-differentiable penalty terms replacing the constraints [Ber82] and employing subgradient-based methods. However, these methods are difficult to implement, often lead to chattering, and the study of their convergence properties requires tools from nonsmooth analysis. Yet another approach is the alternating direction method of multipliers [BPC+11], which requires using some additional reformulation techniques [MXAP13] to make it distributed and convergence to an optimizer is only guaranteed when the optimization problem is convex. Although it enjoys fast convergence, each agent needs to solve a local optimization problem at every iteration to update its state, which might be computationally inefficient depending on the form of the constraint and the objective functions.* •

4.2 Distributed Computation of the Gradient of Penalty Function

We pursue next our strategy to solve the constrained optimization problem (4.1) in a distributed fashion by using the gradient dynamics of the continuously differentiable exact penalty function (2.10). In this section, we first identify the challenges associated with the distributed com-

putation of ∇f^ϵ and then employ the algorithmic tools and results of Section 3.2 to address them.

The gradient of $f^\epsilon(x)$ with respect to x_i is given by

$$\begin{aligned}\nabla_{x_i} f^\epsilon(x) =& \nabla_{x_i} f_i(x_i) + \sum_{j=1}^{m} \lambda_j(x) \nabla_{x_i} g_j(x) + \sum_{k=1}^{p} \mu_k(x) \nabla_{x_i} h_k(x) + \sum_{k=1}^{p} h_k(x) \nabla_{x_i} \mu_k(x) \\ &+ \sum_{j=1}^{m} \left(g_j(x) + y_j^{\epsilon 2}(x)\right) \nabla_{x_i} \lambda_j(x) + \frac{2}{\epsilon} \sum_{j=1}^{m} \left(g_j(x) + y_j^{\epsilon 2}(x)\right) \nabla_{x_i} g_j(x) \\ &+ \frac{2}{\epsilon} \sum_{k=1}^{p} h_k(x) \nabla_{x_i} h_k(x).\end{aligned} \tag{4.2}$$

In this expression, and with the assumptions made in Section 4.1, if agent i knew $(\lambda(x), \mu(x))$, then it could compute all the terms locally except for

$$\rho_i(x) \equiv \sum_{j=1}^{m} \left(g_j(x) + y_j^{\epsilon 2}(x)\right) \nabla_{x_i} \lambda_j(x) + \sum_{k=1}^{p} h_k(x) \nabla_{x_i} \mu_k(x).$$

The rest of this section is devoted to show how to deal with these two issues. First, we show how we can formulate and solve the problem of calculating $(\lambda(x), \mu(x))$ in a distributed way. After that, we show how agent i can calculate $\rho_i(x)$ with only local information and communication.

4.2.1 Distributed computation of multiplier functions

Given $x \in \mathbb{R}^n$, $(\lambda(x), \mu(x))$ are defined by the linear algebraic equation (2.7). Note that this equation can also be written as

$$N(x) \begin{bmatrix} \lambda(x) \\ \mu(x) \end{bmatrix} = - \begin{bmatrix} \nabla g(x)^\top \\ \nabla h(x)^\top \end{bmatrix} \nabla f(x). \tag{4.3}$$

The next result proves that we can actually decompose the matrix $N(x)$ and the righthand side of (4.3) as the summation of locally computable matrices. This makes the equation have the same structure as (3.1), and hence we can use the distributed algorithms in Chapter 3 to solve it.

Proposition 4.2.1. ***(Equivalence between*** (4.3) ***and*** (3.1)***).*** *For each $x \in \mathbb{R}^n$, calculating $(\lambda(x), \mu(x))$ can be cast as solving a linear algebraic equation with separable problem data.*

Proof. For convenience, for each $i \in \{1, \dots, n\}$, we define $\mathrm{g}_i(x) = (\mathrm{g}_{i1}(x), \dots, \mathrm{g}_{im}(x)) \in \mathbb{R}^m$, where

$$\mathrm{g}_{ij}(x) = \begin{cases} \dfrac{g_j(x)}{n_j} & \text{if } i \text{ is involved in constraint } g_j, \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

and n_j is the total number of agents involved in constraint $j \in \{1, \dots, m\}$. From this definition, we have that $\sum_{i=1}^{n} (\mathrm{diag}(\gamma \mathrm{g}_i(x)))^2 = \gamma^2 G^2(x)$. Using this fact, we define $N_i(x)$, for each agent $i \in \{1, \dots, n\}$ as

$$N_i(x) = \begin{bmatrix} \nabla_{x_i} g(x)^\top \\ \nabla_{x_i} h(x)^\top \end{bmatrix} \begin{bmatrix} \nabla_{x_i} g(x) \nabla_{x_i} h(x) \end{bmatrix} + \begin{bmatrix} (\mathrm{diag}(\gamma \mathrm{g}_i(x)))^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

From the definition (2.6) of $N(x)$, note that

$$N(x) = \sum_{i=1}^{n} N_i(x). \tag{4.5}$$

The righthand side of (4.3) could be decomposed as

$$\begin{bmatrix} \nabla g(x)^\top \\ \nabla h(x)^\top \end{bmatrix} \nabla f(x) = \sum_{i=1}^{n} \begin{bmatrix} \nabla_{x_i} g(x)^\top \nabla_{x_i} f_i(x_i) \\ \nabla_{x_i} h(x)^\top \nabla_{x_i} f_i(x_i) \end{bmatrix}. \tag{4.6}$$

Hence, (4.3) is equivalent to (3.1), completing the proof. □

From the definition of the matrices $\{N_i\}_{i=1}^n$ in the proof of Proposition 4.2.1, one can deduce that, individually, these matrices might not be positive definite in general. This highlights the importance of the algorithms in Chapter 3 to solve equations with separable problem data when the coefficient matrices are not necessarily positive definite. Combining Proposition 4.2.1 with the discussion of Chapter 3, we deduce that each agent can compute $(\lambda(x), \mu(x))$ in a distributed way.

4.2.2 Distributed computation of the gradient

Here, we describe how agent $i \in \{1, \dots, n\}$ can calculate $\rho_i(x)$ locally, completing the distributed computation of $\nabla_{x_i} f^\epsilon(x)$.

Proposition 4.2.2. ***(Local computation of*** $\rho_i(x)$***).*** *For each $x \in \mathbb{R}^n$, agent $i \in \{1, \dots, n\}$ can calculate $\rho_i(x)$ locally via communication with its 2-hop neighbors.*

Proof. In compact form, $\rho_i(x)$ can be written as

$$\rho_i(x) = \begin{bmatrix} g(x) + Y^\epsilon(x) y^\epsilon(x) \\ h(x) \end{bmatrix}^\top \begin{bmatrix} \nabla_{x_i} \lambda(x) \\ \nabla_{x_i} \mu(x) \end{bmatrix}.$$

This means that $\rho_i(x)$ is given by the ith column of $[g(x) + y^\epsilon(x); h(x)]^\top [\nabla\lambda(x)^\top; \nabla\mu(x)^\top]$. From (2.8), this is equivalent to saying that $\rho_i(x)$ is given by $-\varrho(x)^\top (r_i(x); s_i(x))$, where $\varrho(x) =$

$N^{-1}(x)[g(x)+Y^{\epsilon}(x)y^{\epsilon}(x); h(x)]$ whose transpose is $[g(x)+Y^{\epsilon}(x)y^{\epsilon}(x); h(x)]^{\top}N^{-1}(x)$ (since $N(x)$ is symmetric) and $(r_i(x); s_i(x))$ denotes the ith column of $[R(x); S(x)]$. Based on this, we divide the distributed computation of $\rho_i(x)$ in two parts:

(a) First we show how all agents can compute $\varrho(x)$ using a 2-hop distributed algorithm;

(b) Next we show that each agent $i \in \{1, \ldots, n\}$ can calculate $r_i(x)$ and $s_i(x)$ locally via communication with its 2-hop neighbors.

For (a), consider the following equation in ϱ

$$N(x)\varrho(x) = \begin{bmatrix} g(x) + Y^{\epsilon}(x)y^{\epsilon}(x) \\ h(x) \end{bmatrix}. \tag{4.7}$$

We can decompose the righthand side of (4.7) as

$$\begin{bmatrix} g(x) + Y^{\epsilon}(x)y^{\epsilon}(x) \\ h(x) \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} \mathrm{g}_i(x) + \mathrm{y}_i^2(x) \\ \mathrm{h}_i(x) \end{bmatrix}, \tag{4.8}$$

where $\mathrm{g}_i(x)$ is defined in (4.4), and $\mathrm{y}_i^2(x)$ and $\mathrm{h}_i(x)$ are defined similarly. From (4.5) and (4.8), equation (4.7) has the structure described in (3.1) and hence can be solved in a distributed manner by the algorithm of Section 3.2.

Next we look at the decomposition of $[R(x); S(x)]$ for (b). We describe here only the decomposition for $R(x)$ (the decomposition for $S(x)$ is similar). From (2.9a), $R(x)$ in expanded form

could be written as

$$
\begin{aligned}
R(x) =& \nabla g(x)^\top \Big(\nabla^2 f(x) + \sum_{j=1}^{m} \lambda_j(x) \nabla^2 g_j(x) + \sum_{k=1}^{p} \mu_k \nabla^2 h_k(x) \Big) \\
&+ \sum_{j=1}^{m} e_j^m (\nabla f(x)^\top + \lambda^\top \nabla g(x)^\top + \mu^\top \nabla h(x)^\top) \nabla^2 g_j(x) \\
&+ 2\gamma^2 \Lambda(x) G(x) \nabla g(x)^\top,
\end{aligned}
$$

which clearly corresponds to a sum of matrices. Here, we look at the first column of these matrices one by one and show that $r_1(x)$ can be calculated by agent 1 with information from its 2-hop neighbors (following the same reasoning justifies that each $r_i(x)$ can be calculated by agent $i \in \{1, \dots, n\}$). The first column of the first matrix is given by $\nabla g(x)^\top \nabla^2_{x_1 x} f(x)$. To calculate it, in addition to $\nabla^2_{x_1} f_1(x)$, agent 1 only needs to know the partial derivative of the constraints in which it is involved (which are available to it by assumption, cf. Section 4.1). The first column corresponding to the next two matrices is given by $\nabla g(x)^\top \Big(\sum_{j=1}^{m} (\nabla^2_{x_1 x} g_j(x)) \lambda_j(x) + \sum_{k=1}^{p} (\nabla^2_{x_1 x} h_p(x)) \mu_k(x) \Big)$. For these, agent 1 only needs information about the partial first and second derivatives of the constraints in which it is involved, in addition to the values of the multiplier functions. The first column corresponding to the next three matrices is $\sum_{j=1}^{m} e_j^m (\nabla f(x)^\top + \lambda^\top \nabla g(x)^\top + \mu^\top \nabla h(x)^\top) \nabla^2_{x x_1} g_j(x)$. The calculation of the first term is straightforward. Rewriting the second term as $\sum_{j=1}^{m} e_j^m \lambda^\top \nabla g(x)^\top \nabla^2_{x x_1} g_j(x)$ and knowing the structure of $\nabla g(x)^\top \nabla^2_{x x_1} g_j(x)$ from the discussion above, we can say that can be calculated by agent 1 (a similar observation applies to the third term). Regarding the last matrix, the first column is $2\gamma^2 [\lambda_1 g_1 \nabla_{x_1} g_1; \dots; \lambda_m g_m \nabla_{x_1} g_m]$. Clearly, agent 1 only needs to know the values and partial derivatives of the constraints in which it is involved for calculating this, concluding the proof. □

Based on Propositions 4.2.1 and 4.2.2, for a given $x \in \mathbb{R}^n$, we can compute asymptotically the values of $\lambda(x)$, $\mu(x)$ and $\varrho(x)$, and in turn, the gradient of the penalty function in a distributed way. For its use later, we denote by $\mathbf{P}_{\text{est}}(x)$ the corresponding matrix defined as in (3.3), which now depends on x due to the x-dependence of N_i and b_i (and hence $\mathbf{A}$ and $\mathbf{b}$) in equations (4.3) and (4.7).

Remark 4.2.3. ***(Robustness in the calculation of gradient).*** *From Proposition 3.2.4, the distributed calculation of the gradient of the exact penalty function is robust to bounded disturbances due to errors in the problem data (e.g., errors in the value of the constraint functions or the gradients of the objective and constraint functions), packet drops, or communication noise. Furthermore, since the matrix $N(x) = \sum_i^n N_i(x)$ is positive definite (and hence invertible) from Proposition 2.6.1, it follows that all equilibria have the same unique primary variable, whereas the auxiliary ones may take multiple values according to Proposition 3.2.1. This means that, for a given $x \in \mathbb{R}^n$, the primary variables converge uniquely to $\lambda(x)$, $\mu(x)$ and $\varrho(x)$ under each of the algorithms.* •

4.3 Distributed Optimization via Interconnected Dynamics

In this section, we finally put all the elements developed so far together to propose a distributed algorithm to solve (4.1). The basic idea is to implement the gradient dynamics of the exact penalty function. However, the algorithmic solutions resulting from Section 4.2 only asymptotically compute the gradient of the exact penalty function at a given state. This state, in turn, changes by the action of the gradient descent dynamics. The proposed distributed algorithm is then the result of the interconnection of these two complementary dynamics.

Formally, the gradient descent dynamics of f^ϵ which serves as reference for our algorithm

design takes the form

$$\dot{x} = -\nabla f^{e}(x). \tag{4.9}$$

For convenience, define $\chi : \mathbb{R}^n \to \mathbb{R}^{2(m+p)}$ by $\chi(x) = (\lambda(x), \mu(x), \varrho(x))$ and rewrite (4.9) as $\dot{x} = \psi_{\text{grad}}(x, \chi(x))$ for an appropriate function ψ_{grad} defined by examining the expression in (4.2) for $i \in \{1, \dots, n\}$ (note that, given the assumptions on the problem functions, for each $x \in \mathbb{R}^n$, the function ψ_{grad} is locally Lipschitz in its argument χ, and from Proposition 2.6.1 and equation (4.7), $x \mapsto \chi(x)$ is continuously differentiable). The variable χ corresponds to those terms appearing in the gradient that are not immediately computable with local information. However, with the distributed algorithms described in Section 4.2, the network agents can asymptotically compute $\chi(x)$ in a distributed fashion. Let $\Upsilon \in \mathbb{R}^{4(m+p)}$ denote the augmented variable containing the estimates of $\chi(x)$ and the associated auxiliary variables, available to the network agents via

$$\dot{\Upsilon} = \psi_{\text{est}}(x, \Upsilon), \tag{4.10a}$$

where $\psi_{\text{est}}(x, \Upsilon)$ denotes the algorithms of the form (3.6) described in Section 4.2. Let $\hat{\chi} = \mathcal{P}_{\chi}\Upsilon$ denote the projection of Υ onto the χ space, i.e., corresponding to the set of primary variables. From Proposition 3.2.2, we note that, for fixed $x \in \mathbb{R}^n$, $\hat{\chi} \to \chi(x)$ exponentially fast. Hence, with the information available to the agents, instead of (4.9), the network implements

$$\dot{x} = \psi_{\text{grad}}(x, \mathcal{P}_{\chi}\Upsilon). \tag{4.10b}$$

Our proposed algorithm is the interconnected dynamical system (4.10). When convenient, we re-

fer to it as ψ_{interc}. Note that this algorithm is 2-hop distributed. Moreover, for each equilibrium $(x_{\text{eq}}, \Upsilon_{\text{eq}})$, of (4.10), its x-component x_{eq} is an equilibrium of (4.9) (which is also a KKT point of problem (4.1) if EMFCQ is satisfied, cf. Proposition 2.6.2). We characterize the convergence properties of the algorithm (4.10) next.

Theorem 4.3.1. ***(Asymptotic convergence of distributed algorithm to solution of optimization problem).*** *Assume LICQ is satisfied at each $x \in \mathcal{D}$. For each x, let $L_\chi(x)$ be the Lipschitz constant of $\chi \mapsto \psi_{\text{grad}}(x, \chi)$. Then the equilibria of the interconnected dynamics* (4.10) *are asymptotically stable if there exists $\alpha > 0$ such that*

$$\max_{x \in D} \frac{\eta_\alpha(x)}{\lambda_2(\mathbf{P}_{est}(x)^\top \mathbf{P}_{est}(x))} < 1, \tag{4.11}$$

where $\eta_\alpha(x) = \frac{1}{4\alpha}(\alpha L_\chi(x) + \|\nabla_x \chi(x)\|)^2 + L_\chi(x)\|\nabla_x \chi(x)\|$.

Proof. We start by noting that (4.11) is well defined since, from the definitions of $R(x)$ and $S(x)$ in (2.9) and the expression of the gradient in (4.2), we deduce that $L_\chi(x)$ is continuous in x, and, moreover, since $\mathcal{D}$ is compact, L_χ and $\|\nabla_x \chi\|$ are bounded over $\mathcal{D}$. Consider now the Lyapunov function candidate for the interconnected system as

$$V_{\text{c}}(x, \Upsilon) = \alpha f^{\epsilon}(x) + V_2(x, \Upsilon), \tag{4.12}$$

where V_2 is defined as in (3.7), but due to the dependence of $\mathbf{z}^*$ on x from equations (4.3) and (4.7),

is now a function of x too. The derivative of V_{c} along the dynamics (4.10) is

$$\begin{aligned}\mathcal{L}_{\psi_{\text{interc}}} V_{\mathrm{c}}(x,\Upsilon) &= (\alpha\nabla f^{\epsilon}(x) + \nabla_x V_2)^{\top}\, \psi_{\text{grad}}(x, \mathcal{P}_{\chi}\Upsilon) + \nabla_{\Upsilon} V_2^{\top} \psi_{\text{est}}(x,\Upsilon) \\ &\leq -(\alpha\nabla f^{\epsilon}(x) + \nabla_x \chi(\hat{\chi} - \chi(x)))^{\top}(\nabla f^{\epsilon}(x) \\ &\quad - \psi_{\text{grad}}(x,\hat{\chi}) + \psi_{\text{grad}}(x,\chi(x))) - \lambda_2(x)\|\hat{\chi} - \chi(x)\|^2,\end{aligned}$$

where we have added and subtracted $\nabla f^{\epsilon}(x) = \psi_{\text{grad}}(x,\chi(x))$ to $\psi_{\text{grad}}(x,\mathcal{P}_{\chi}\Upsilon)$ and used the shorthand notation $\lambda_2(x) \equiv \lambda_2(\mathbf{P}_{\text{est}}^{\top}(x)\mathbf{P}_{\text{est}}(x))$. Hence, we have

$$\mathcal{L}_{\psi_{\text{interc}}} V c_{\mathrm{c}}(x,\Upsilon) \leq -\begin{bmatrix} \|\nabla f^{\epsilon}(x)\| \\ \|\hat{\chi} - \chi(x)\| \end{bmatrix}^{\top} A(x) \begin{bmatrix} \|\nabla f^{\epsilon}(x)\| \\ \|\hat{\chi} - \chi(x)\| \end{bmatrix},$$

with

$$A(x) = \begin{bmatrix} \alpha & -\frac{1}{2}(\alpha L_{\chi}(x) + \|\nabla_x \chi\|) \\ -\frac{1}{2}(\alpha L_{\chi}(x) + \|\nabla_x \chi\|) & \lambda_2(x) - L_{\chi}(x)\|\nabla_x \chi\|) \end{bmatrix}.$$

Next, we examine the positive-definiteness nature of the 2×2-matrix $A(x)$. Since $\alpha > 0$, note that $A(x) \succ 0$ if the determinant is positive. For $x \in \mathcal{D}$, the latter holds if and only if α is such that

$$\eta_{\alpha}(x)/\lambda_2(x) < 1.$$

Hence, under (4.11), this inequality holds over $\mathcal{D}$, and consequently $\mathcal{L}_{\psi_{\text{interc}}} V_{\mathrm{c}}(x,\Upsilon) < 0$ over $\mathcal{D}\times \mathbb{R}^{4(m+p)}$. □

The condition (4.11) in Theorem 4.3.1 can be interpreted as requiring the estimation dynam-

ics (4.10a) to be fast enough to ensure the error in the gradient computation remains manageable, resulting in the convergence of the interconnected system. In general, however, (4.11) might not be satisfied. To address this, and inspired by this interpretation, we propose to execute the estimation dynamics on a tunable timescale, substituting (4.10a) by

$$\tau \dot{\Upsilon} = \psi_{\text{est}}(x, \Upsilon). \tag{4.13}$$

Here, $\tau > 0$ is a design parameter capturing the timescale at which the estimation dynamics is now executed. Resorting to singular perturbation theory, cf. [Kha02, Vel97], one could show that $x(t) \to x_{\text{grad}}(t)$ as $\tau \to 0$, where x_{grad} denotes the trajectory of the gradient descent dynamics (4.9). However, for the proposed approach to be practical, it is desirable to have a strictly positive value of the timescale below which convergence is guaranteed. The following result shows that such critical value exists.

Proposition 4.3.2. ***(Asymptotic convergence of distributed algorithm via accelerated estimation dynamics).*** *Assume LICQ is satisfied at each $x \in \mathcal{D}$ and let*

$$\tau_* = \frac{\lambda_{\min}(\mathbf{P}_{est}^\top \mathbf{P}_{est})}{2\bar{L}_\chi \|\nabla_x \bar{\chi}\|} > 0,$$

where $\lambda_{\min}(\mathbf{P}_{est}^\top \mathbf{P}_{est})$ denotes the minimum of $\lambda_2(\mathbf{P}_{est}^\top(x)\mathbf{P}_{est}(x))$, and $\bar{L}_\chi$ and $\|\nabla_x \bar{\chi}\|$ denote the maximum of L_χ and $\|\nabla_x \chi\|$ resp., over $\mathcal{D}$. Then, for any $\tau \in [0, \tau_*)$, the equilibria of the interconnected dynamics (4.10b) *and* (4.13) *are asymptotically stable.*

Proof. Let $\alpha > 0$ and consider the Lyapunov function candidate (4.12). Define

$$A_\tau(x) = \begin{bmatrix} \alpha & -\frac{1}{2}(\alpha L_\chi(x) + \|\nabla_x \chi\|) \\ -\frac{1}{2}(\alpha L_\chi(x) + \|\nabla_x \chi\|) & \tau^{-1}\lambda_2(x) - L_\chi(x)\|\nabla_x \chi\|) \end{bmatrix}.$$

Following the same line of argument as in the proof of Theorem 4.3.1, we arrive at

$$L_{\psi_{\text{interc}}} V c_{\text{c}}(x, \Upsilon) \leq -\begin{bmatrix} \|\nabla f^e(x)\| \\ \|\hat{\chi} - \chi(x)\| \end{bmatrix}^\top A_\tau(x) \begin{bmatrix} \|\nabla f^e(x)\| \\ \|\hat{\chi} - \chi(x)\| \end{bmatrix}$$

and the condition $\tau < \lambda_2(x)/\eta_\alpha(x)$ to ensure $L_{\psi_{\text{interc}}} V c_{\text{c}}(x, \Upsilon) < 0$. Using the bounds for L_χ and $\|\nabla_x \chi\|$, we upper bound η_α over D as

$$\eta_\alpha(x) \leq \bar{\eta}_\alpha = \frac{1}{4\alpha}(\alpha \bar{L}_\chi + \|\nabla_x \bar{\chi}\|)^2 + \bar{L}_\chi \|\nabla_x \bar{\chi}\|.$$

Consequently, it is enough to have $\tau < \lambda_2(x)/\bar{\eta}_\alpha$ for all $x \in \mathsf{D}$. To establish the maximum admissible value of τ, we can select the value of α minimizing $\bar{\eta}_\alpha$. Since $\bar{\eta}_\alpha$ is strictly convex in $\alpha \in [0, \infty)$, this is given by the solution of

$$\frac{d}{d\alpha}\left(\frac{1}{\alpha}(\alpha \bar{L}_\chi + \|\nabla_x \bar{\chi}\|)^2\right) = 0.$$

After some algebraic manipulations, one can verify that $\alpha^* = \|\nabla_x \bar{\chi}\|/\bar{L}_\chi$. Substituting this value in the expression of $\bar{\eta}_\alpha$ and taking the minimum over all $x \in \mathsf{D}$ yields the definition of τ_*. □

Note that the conditions identified in Theorem 4.3.1 and Proposition 4.3.2 to ensure convergence are based on upper bounding the terms appearing in the Lie derivative of the Lyapunov

function candidate using 2-norms and, as such, are conservative in general. In fact, the algorithm may converge even if these conditions are not satisfied, something that we have observed in simulation.

Remark 4.3.3. ***(Constraint satisfaction with the distributed dynamics).*** *The centralized gradient descent on which we build our approach enjoys the constraint satisfaction properties stated in Proposition 4.1.1. This means, using a singular perturbation argument [Kha02, Vel97], that the distributed gradient descent approach proposed here has the same guarantees as $\tau \to 0$. Although for a fixed $\tau > 0$, we do not have a formal guarantee that the state remains feasible, we have observed this to be the case in simulations, even under general constraints. We believe this is due to the error-correcting terms in the original penalty function, which penalize deviations from the feasible set. This anytime nature is especially important in applications where the optimization problem is not stand alone and its solution serves as an input to other layer in the control design (for example as a power/thermal set point, cf. [RJ14, RK12]), where the algorithm should yield a feasible solution if terminated in finite time.* •

4.4 Simulations

Here, we illustrate the effectiveness of the proposed distributed dynamics (4.10). Our optimization problem is inspired by [KMT98]: we consider 50 agents connected in a circle forming a ring topology and seeking to solve

$$\begin{aligned} \max_{x \in D} \quad & \sum_{i=1}^{50} f_i(x_i) \\ \text{s.t.} \quad & Ax \leq C. \end{aligned}$$

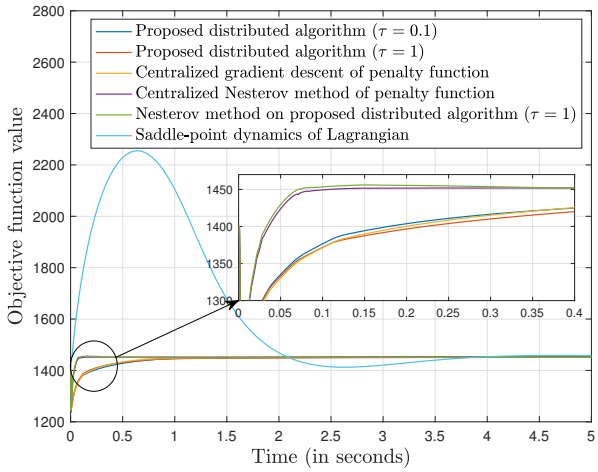
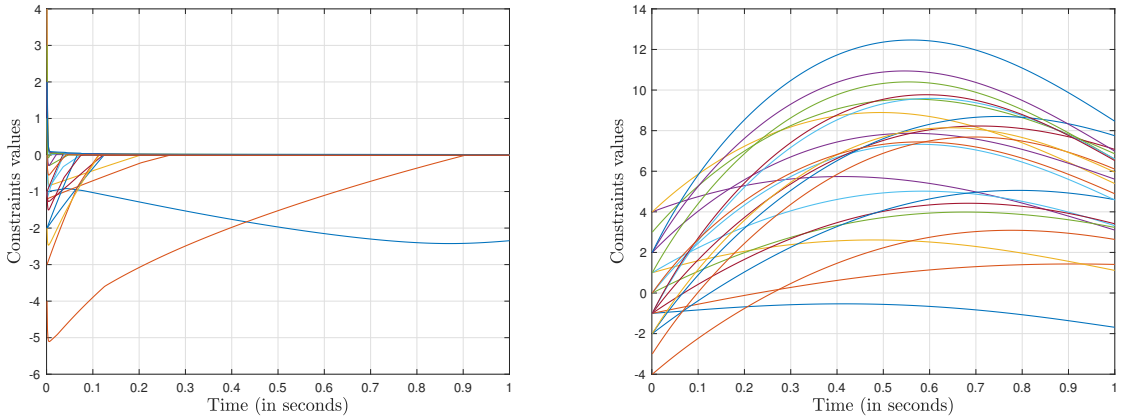


Figure 4.1: Evolution of the objective function value under the proposed distributed dynamics with $\tau = 10^{-1}$ and 1, resp., the centralized gradient descent, the centralized and the distributed Nesterov's accelerated gradient method of the penalty function, and the saddle-point dynamics of the Lagrangian.

Here, $f_i(x_i) = i \log x_i$ for $i \in \{1, \dots, 50\}$. The sparse matrix $A \in \mathbb{R}^{23\times 50}$ is such that each of the 23 constraints it defines involves a different corresponding agent and its 1-hop neighbors. We take $\mathcal{D} = \{x \in \mathbb{R}^n \mid 10^{-1} \leq \|x\|_\infty \leq 10\}$. Throughout the simulations, we consider the exact penalty function (2.10) with $\epsilon = 10^{-2}$ and $\gamma = 1$. Since the dynamics are in continuous time, we use a first-order Euler discretization for the MATLAB implementation with stepsize 10^{-3}. We compare the performance of the proposed distributed algorithm with values $\tau = 1$ and $\tau = 10^{-1}$, resp., against the centralized gradient descent (4.9), the saddle-point dynamics [CGC17] of the Lagrangian, and the centralized and the distributed Nesterov's accelerated gradient method [Nes83] of the penalty function. To implement the latter, we use $\tau = 1$ and replace (4.10b) with Nesterov's acceleration step. We use the same initial condition for all the algorithms. Figure 4.1 shows the evolution of the objective function under each algorithm. One can observe that the proposed distributed algorithm performs much better than the saddle-point dynamics. As expected, centralized Nesterov's accelerated gradient method performs the best, followed by the distributed Nesterov method obtained



(a) Proposed dynamics

(b) Saddle-point dynamics

Figure 4.2: Evolution of the constraints.

by applying the acceleration to our proposed distributed algorithm. The output of the distributed algorithm for both values of τ is also close to that of the centralized gradient descent. Figure 4.2 show the evolution of the value of $Ax - C$ for the proposed distributed algorithm with $\tau = 1$ and the saddle-point dynamics. Even though Proposition 4.1.1 states that, for the centralized gradient descent counterpart, there is no guarantee of staying inside the feasible set for general constraints, Figure 4.2 shows that the distributed algorithm satisfies the constraints much better during the evolution than the saddle-point dynamics.

In the next simulation we illustrate the robustness of the proposed dynamics. For this, we add a disturbance to the dynamics (4.10) using random vectors at each iteration as follows. For (4.10a), we add $d = \beta\|u(x, \Upsilon)\| \times$ (unit-norm random vector), where we use the MATLAB function `rand` to generate random numbers between 0 and 1. Similarly, for (4.10b), we add $d = \beta\|w(x, \mathcal{P}_\chi \Upsilon)\| \times$(unit-norm random vector). For the scaling constant β, which also equals the ratio of the norm of the total disturbance to the norm of the unperturbed dynamics, we use gradually increasing values between 0.1 to 0.5. For each value of β, we plot the evolution of the objective

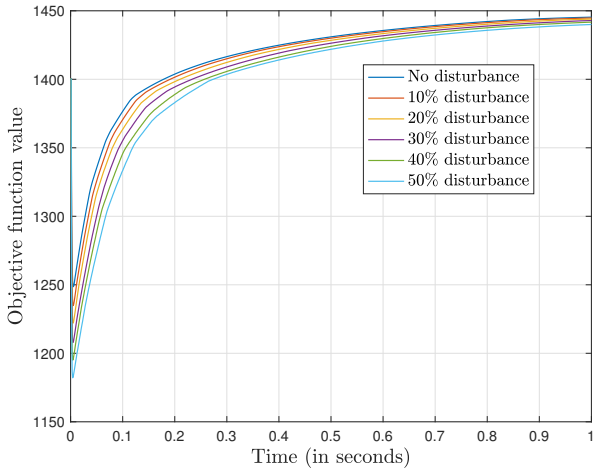


Figure 4.3: Evolution of the objective function value under the proposed distributed dynamics in the presence of disturbances. The amount of disturbance in percentage denotes the ratio of the norm of the disturbance to the norm of the unperturbed dynamics.

function with $\tau = 1$ in Figure 4.3. The plot shows the graceful degradation of the performance as the ratio of the norm of disturbance to the norm of unperturbed dynamics increases, demonstrating the effectiveness of the proposed dynamics against disturbances.

Acknowledgements

This chapter, in part, is a reprint of the material [SC21b] as it appears in 'Network optimization via smooth exact penalty functions enabled by distributed gradient computation' by P. Srivastava and J. Cortés, in the IEEE Transactions on Control of Network Systems, 2021, as well as [SC18] where it appears as 'Distributed algorithm via continuously differentiable exact penalty method for network optimization' by P. Srivastava and J. Cortés in the proceedings of the 2018 IEEE Conference on Decision and Control. The dissertation author was the primary investigator and author of these papers. This work was supported by the ARPA-e NODES program, Cooperative Agreement DE-AR0000695 and NSF Award ECCS-1917177.

Chapter 5

Nesterov Acceleration for Equality-Constrained Convex Optimization

This chapter proposes a framework to use Nesterov's accelerated method for constrained convex optimization problems. Our approach consists of first reformulating the original problem as an unconstrained optimization problem using a continuously differentiable exact penalty function. This reformulation is based on replacing the Lagrange multipliers in the augmented Lagrangian of the original problem by Lagrange multiplier functions. The expressions of these Lagrange multiplier functions, which depend upon the gradients of the objective function and the constraints, can make the unconstrained penalty function non-convex in general even if the original problem is convex. We establish sufficient conditions on the objective function and the constraints of the original problem under which the unconstrained penalty function is convex. This enables us to use Nesterov's accelerated gradient method for unconstrained convex optimization and achieve a guaranteed rate of convergence which is better than the state-of-the-art first-order algorithms for constrained convex optimization.

5.1 Problem Statement

Consider the following convex optimization problem

$$\begin{aligned} \min_{x \in D} \quad & f(x) \\ \text{s.t.} \quad & Ax - b = 0, \end{aligned} \tag{5.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable convex function and D is a convex set. Here $A \in \mathbb{R}^{p \times n}$ and $b \in \mathbb{R}^p$ with $p < n$. Without loss of generality, we assume A has full row rank (implying that LICQ holds at all $x \in \mathbb{R}^n$).

Our aim is to design a Nesterov-like fast method to solve (5.1). We do this by reformulating the problem as an unconstrained optimization using continuously differentiable penalty function methods, cf. Section 2.6.1. Then, we employ the Nesterov's accelerated gradient method to design

$$x_{k+1} = y_k - \alpha \nabla f^\epsilon(y_k), \tag{5.2a}$$

$$a_{k+1} = \frac{1 + \sqrt{4a_k^2 + 1}}{2}, \tag{5.2b}$$

$$y_{k+1} = x_{k+1} + \frac{a_k - 1}{a_{k+1}}(x_{k+1} - x_k), \tag{5.2c}$$

where $\alpha \in \mathbb{R}_{>0}$ is the stepsize. If f^ϵ is convex with Lipschitz gradient L and the algorithm is initialized at an arbitrary initial condition x_0 with $y_0 = x_0$ and $a_0 = 1$, then according to [Nes83, Theorem 1],

$$f^\epsilon(x_k) - f^\epsilon(x^*) \leq \frac{C}{(k+1)^2}, \tag{5.3a}$$

where $x^* \in \mathbb{R}^n$ is a global minimizer of f^ϵ and $C \in \mathbb{R}_{\geq 0}$ is a constant dependant upon the initial condition and L. If f^ϵ is strongly convex with parameter s, and (5.2b) and (5.2c) are replaced by

$$y_{k+1} = x_{k+1} + \frac{\sqrt{L} - \sqrt{s}}{\sqrt{L} + \sqrt{s}}(x_{k+1} - x_k), \tag{5.2d}$$

then one has from [Nes18, Theorem 2.2.1]

$$f^\epsilon(x_k) - f^\epsilon(x^*) \leq C_s \exp\left(-k\sqrt{\frac{s}{L}}\right), \tag{5.3b}$$

where $C_s \in \mathbb{R}_{\geq 0}$ is a constant dependant upon the initial condition, s, and L. The key technical point for this approach to be successful is to ensure that the penalty function f^ϵ is (strongly) convex. Section 5.2 below shows that this is indeed the case for suitable values of the penalty parameter under appropriate assumptions on the objective and constraint functions of the original problem (5.1).

Remark 5.1.1. ***(Distributed algorithm implementation).*** *We note here that the algorithm* (5.2) *is amenable to distributed implementation if the objective function is separable and the constraints are locally coupled. In fact, Chapter 4 shows how, in this case, the computation of the gradient of the penalty function in* (5.2a) *can be implemented in a distributed way. Based on this observation, one could use the framework proposed here for fast optimization of convex problems in a distributed way. To obtain fast convergence, one could also use second-order augmented Lagrangian methods, e.g., [AO17, DKJ17], but their distributed implementation faces the challenge of computing the inverse of the Hessian of the augmented Lagrangian to update the primal and dual variables. Even if the Hessian is sparse for separable objective functions and local constraints, its inverse in general is not.* •

5.2 Convexity of the Penalty Function

We start by showing that the continuously differentiable exact penalty function f^ϵ defined in (2.10) might not be convex even if the original problem (5.1) is convex. For the convex problem (5.1), the penalty function takes the form

$$f^\epsilon(x) = f(x) - ([AA^\top]^{-1}A\nabla f(x))^\top(Ax-b) + \frac{1}{\epsilon}\|Ax-b\|^2. \tag{5.4}$$

A look at this expression makes it seem like a sufficiently small choice of ϵ might make f^ϵ convex for all $x \in \mathcal{D}$. The following shows that this is always not the case.

Example 5.2.1. ***(Non-convex penalty function).*** *Consider*

$$\begin{aligned} \min_{x\in\mathcal{D}} \quad & x_1^4 + x_2^4 \\ s.t. \quad & x_1 + x_2 = 0. \end{aligned}$$

The optimizer is $(0,0)$*. The penalty function takes the form*

$$f^\epsilon(x) = x_1^4 + x_2^4 + \mu(x)^\top(x_1+x_2) + \frac{1}{\epsilon}(x_1+x_2)^2,$$

where $\mu(x) = -(2x_1^3 + 2x_2^3)$*. The Hessian of this function is*

$$\nabla^2 f^\epsilon(x) = \begin{bmatrix} -12x_1^2 - 12x_1x_2 + \frac{2}{\epsilon} & -6x_1^2 - 6x_2^2 + \frac{2}{\epsilon} \\ -6x_1^2 - 6x_2^2 + \frac{2}{\epsilon} & -12x_2^2 - 12x_1x_2 + \frac{2}{\epsilon} \end{bmatrix}.$$

If $x_1 = 0$*, then the determinant of* $\nabla^2 f^\epsilon(x)$ *evaluates to* $-36x_2^4$*, which is independent of* ϵ*. Hence,*

f^ϵ cannot be made convex over any set containing the vertical axis. •

Example 5.2.1 shows that the penalty function cannot always be convexified by adjusting the value of ϵ. Intuitively, the reason for this fact is that the term susceptible to be scaled in the expression (5.4) which depends on the parameter ϵ is not strongly convex. This implies that there are certain subspaces where non-convexity arising from the term that involve the Lagrange multiplier function cannot be countered. In turn, these subspaces are defined by the kernel of the Hessian of the last term in the expression (5.4) of the penalty function.

These observations motivate our study of conditions on the objective function and the constraints that guarantee that the penalty function is convex. In our discussion, we start by providing sufficient conditions for the convexity of the penalty function over $\mathcal{D}$.

5.2.1 Sufficient Conditions for Convexity over the Domain

Here we provide conditions for the convexity of the penalty function f^ϵ by establishing the positive semi definiteness of its Hessian. Throughout the section, we assume f is three times differentiable. Note that the gradient and the Hessian of f^ϵ are given, resp., by

$$\nabla f^\epsilon(x) = \nabla f(x) - \nabla^2 f(x) A^\top [AA^\top]^{-1}(Ax - b) - A^\top [AA^\top]^{-1} A \nabla f(x) + \frac{2}{\epsilon} A^\top (Ax - b). \quad (5.5a)$$

$$\nabla^2 f^\epsilon(x) = \nabla^2 f(x) - W(x) - \nabla^2 f(x) A^\top [AA^\top]^{-1} A - A^\top [AA^\top]^{-1} A \nabla^2 f(x) + \frac{2}{\epsilon} A^\top A, \quad (5.5b)$$

where we use the short-hand notation

$$W(x) = \sum_{i=1}^{n} \nabla_{x_i} \nabla^2 f(x) A^\top [AA^\top]^{-1}(Ax - b) e_i^{n\top}. \quad (5.6)$$

The following result provides sufficient conditions under which the penalty function (5.4) is convex on $\mathcal{D}$.

Theorem 5.2.2. ***(Convexity of the penalty function).*** *For the optimization problem* (5.1)*, assume* $\nabla^2 f(x) - W(x) \succ 0$ *for all* $x \in \mathcal{D}$ *and let*

$$\bar{\epsilon} = \min_{x \in D} \frac{2\lambda_{\min}(AA^\top)\lambda_{\min}(\nabla^2 f(x) - W(x))}{\lambda^2_{\max}(\nabla^2 f(x)) + R(x)\lambda_{\min}(\nabla^2 f(x) - W(x))},$$

where $R(x) = 2\lambda_{\max}(\nabla^2 f(x)) - \lambda_{\min}(\nabla^2 f(x) - W(x))$*. Then* f^ϵ *is convex on* $\mathcal{D}$ *for all* $\epsilon \in (0, \bar{\epsilon}]$ *and consequently the convergence guarantee* (5.3a) *holds.*

Proof. For an arbitrary $x \in \mathcal{D}$, we are interested in determining the conditions under which $\nabla^2 f^\epsilon(x) \succeq 0$, or in other words, $v^\top \nabla^2 f^\epsilon(x) v \geq 0$ for all $v \in \mathbb{R}^n$. From (5.5b),

$$\begin{aligned} v^\top \nabla^2 f^\epsilon(x) v = &\frac{2}{\epsilon} v^\top A^\top A v + v^\top (\nabla^2 f(x) - W(x)) v \\ &- 2v^\top (\nabla^2 f(x) A^\top [AA^\top]^{-1} A) v. \end{aligned} \tag{5.7}$$

Let us decompose v as $v = v_\parallel + v_\perp$, where $v_\parallel$ is the component of v in the nullspace of A and $v_\perp$ is the component orthogonal to it. Then (5.7) becomes

$$\begin{aligned} v^\top \nabla^2 f^\epsilon(x) v = &\frac{2}{\epsilon} v_{\perp\top} A^\top A v_\perp + v^\top (\nabla^2 f(x) - W(x)) v \\ &- 2v_\parallel^\top \nabla^2 f(x) A^\top [AA^\top]^{-1} A v_\perp \\ &- 2v_\perp^\top \nabla^2 f(x) A^\top [AA^\top]^{-1} A v_\perp. \end{aligned}$$

Since $A^\top (AA^\top)^{-1} A v_\perp = v_\perp$, cf. [CM09, Theorem 1.1.1], the above expression reduces to

$$\begin{aligned} v^\top \nabla^2 f^\epsilon(x) v =& \frac{2}{\epsilon} v_\perp^\top A^\top A v_\perp + v^\top (\nabla^2 f(x) - W(x)) v - 2 v_\parallel^\top \nabla^2 f(x) v_\perp - 2 v_\perp^\top \nabla^2 f(x) v_\perp \\ \geq& \left(\frac{2}{\epsilon} \lambda_2(A^\top A) - 2\lambda_{\max}(\nabla^2 f(x)) \right) \|v_\perp\|^2 + \lambda_{\min}(\nabla^2 f(x) - W(x))(\|v_\perp\|^2 + \|v_\parallel\|^2) \\ & - 2\lambda_{\max}(\nabla^2 f(x)) \|v_\perp\| \|v^\parallel\| \\ =& \begin{bmatrix} \|v_\perp\| \\ \|v_\parallel\| \end{bmatrix}^\top \underbrace{\begin{bmatrix} S(x) & -\lambda_{\max}(\nabla^2 f(x)) \\ -\lambda_{\max}(\nabla^2 f(x)) & \lambda_{\min}(\nabla^2 f(x) - W(x)) \end{bmatrix}}_{P(x)} \begin{bmatrix} \|v_\perp\| \\ \|v_\parallel\| \end{bmatrix}, \end{aligned}$$

where $S(x) = \frac{2}{\epsilon} \lambda_{\min}(AA^\top) - R(x)$. Therefore, we deduce that $\nabla^2 f^\epsilon(x) \succeq 0$ if ϵ is such that $P(x) \succeq 0$. Being a 2×2-matrix, the latter holds if $S(x)$ and determinant of $P(x)$ are non-negative. The determinant is non-negative if and only if

$$\epsilon \leq \frac{2\lambda_{\min}(AA^\top) \lambda_{\min}(\nabla^2 f(x) - W(x))}{\lambda_{\max}^2(\nabla^2 f(x)) + R(x) \lambda_{\min}(\nabla^2 f(x) - W(x))}.$$

The above value of ϵ also ensures that $S(x) > 0$. Taking the minimum over all $x \in \mathcal{D}$ completes the proof. □

Remark 5.2.3. ***(Differentiability of the objective function).*** *Note that the implementation of* (5.2) *requires the objective function f to be twice continuously differentiable, while the definition of W in* (5.6) *involves the third-order partial derivatives of f. We believe that an extension of Theorem 5.2.2 could be pursued in case the objective function is only twice differentiable using tools from nonsmooth analysis, e.g., [Cla83], but we do not pursue it here.* •

The next result provides sufficient conditions under which the penalty function is strongly

convex on $\mathcal{D}$.

Corollary 5.2.4. ***(Strong convexity of the penalty function).*** *For the optimization problem* (5.1)*, assume* $\nabla^2 f(x) - W(x) \succeq cI$ *for all* $x \in \mathcal{D}$ *and let*

$$\bar{\epsilon}_s = \min_{x \in \mathcal{D}} \frac{2\lambda_{\min}(AA^\top)(c-s)}{\lambda^2_{\max}(\nabla^2 f(x)) + 2(c-s)\lambda_{\max}(\nabla^2 f(x)) - (c-s)^2}.$$

Then f^ϵ *is strongly convex on* $\mathcal{D}$ *with parameter* $s \in (0, c)$ *for all* $\epsilon \in (0, \bar{\epsilon}_s]$ *and the convergence guarantee* (5.3b) *holds.*

Proof. Let us decompose $\nabla^2 f(x) - W(x)$ as $\nabla^2 f(x) - W(x) = B(x) + sI$. Since $\nabla^2 f(x) - W(x) \succeq cI$, it follows that $B(x) \succeq (c-s)I$. Establishing that the penalty function is strongly convex with parameter s is equivalent to establishing that, for all $x \in \mathcal{D}$, $v^\top(\nabla^2 f^\epsilon(x) - sI)v \geq 0$ for all $v \in \mathbb{R}^n$. Following the same steps as in the proof of Theorem 5.2.2, one can verify that this is true if, for all $x \in \mathcal{D}$, ϵ is less than or equal to

$$\frac{2\lambda_{\min}(AA^\top)\lambda_{\min}(B(x))}{\lambda^2_{\max}(\nabla^2 f(x)) + 2\lambda_{\min}(B(x))\lambda_{\max}(\nabla^2 f(x)) - \lambda^2_{\min}(B(x))}.$$

Replacing $\lambda_{\min}(B(x))$ by $c - s$, it follows that the penalty function is strongly convex with parameter s if $\epsilon \leq \bar{\epsilon}_s$. □

It is easy to verify that Example 5.2.1 does not satisfy the sufficient condition identified in Theorem 5.2.2. This condition can be interpreted as requiring the original objective function to be sufficiently convex to handle the non-convexity arising from the penalty for being infeasible. Finding the value of $\bar{\epsilon}$ still remains a difficult problem as computing $\lambda_{\min}(\nabla^2 f(x) - W(x))$ for all

$x \in \mathcal{D}$ is not straightforward. The next result simplifies the conditions of Theorem 5.2.2 for linear and quadratic programming problems.

Corollary 5.2.5. ***(Sufficient conditions for problems with linear and quadratic objective functions).***

(i) If the objective function in problem (5.1) *is linear, then the penalty function is convex on* $\mathbb{R}^n$ *for all values of* ϵ*;*

(ii) If the objective function in problem (5.1) *is quadratic with Hessian* $Q \succ 0$*, then the penalty function is convex on* $\mathbb{R}^n$ *for all* $\epsilon \in (0, \bar{\epsilon}]$*, where*

$$\bar{\epsilon} = \frac{2\lambda_{\min}(AA^\top)\lambda_{\min}(Q)}{\lambda^2_{\max}(Q) + 2\lambda_{\min}(Q)\lambda_{\max}(Q) - \lambda^2_{\min}(Q)}.$$

In either case, the convergence guarantee (5.3a) *holds.*

Proof. We present our arguments for each case separately. For case (i), we have $\nabla^2 f(x) = 0$. Hence,

$$\nabla^2 f^\epsilon(x) = \frac{2}{\epsilon} A^\top A,$$

which means that $\nabla^2 f^\epsilon(x) \geq 0$ for all $x \in \mathbb{R}^n$. For case (ii),

$$f(x) = \frac{1}{2} x^\top Q x + h^\top x,$$

where $Q \in \mathbb{R}^{n\times n}$ and $h \in \mathbb{R}^n$. The expression for the Hessian of f^ϵ becomes

$$\nabla^2 f^\epsilon(x) = Q + \frac{2}{\epsilon} A^\top A - QA^\top [AA^\top]^{-1} A - A^\top [AA^\top]^{-1} AQ.$$

Clearly $W(x) = 0$ for all $x \in \mathbb{R}^n$, and the result follows from Theorem 5.2.2. □

Following Corollary 5.2.4, one can also state similar conditions for the penalty function to be strongly convex in the case of quadratic programs, but we omit them here for space reasons. From Corollary 5.2.5, ensuring that the penalty function convex is easier when the objective function is quadratic. This follows from the fact that $W(x)$, which depends on the third order derivatives of the objection function, vanishes. Hence, in the quadratic case, the condition in Theorem 5.2.2 requiring the Hessian of the objective function to be greater than $W(x)$ for all $x \in \mathcal{D}$ is automatically satisfied. In what follows we provide a very simple approach for general objective functions.

5.2.2 Convexity over Feasible Set Coupled with Invariance

Here we present a simplified version of the proposed approach, which is based on the fact that inside the feasible set the values of the penalty and the objective functions is the same. To build on this observation, we start by characterizing the extent to which the constraints are satisfied under the Nesterov's algorithm.

Lemma 5.2.6. ***(Forward invariance of the feasible set under Nesterov's algorithm applied to the penalty function).*** *Consider the Nesterov's accelerated gradient algorithm* (5.2) *applied to the penalty function* (5.4) *for an arbitrary* $\epsilon \geq 0$*. If the algorithm is initialized at* $y_0 = x_0$*, with* x_0 *belonging to the feasible set* $\mathcal{F}$*, then* $\{x_k\}_{k=0}^\infty$, $\{y_k\}_{k=0}^\infty \in \mathcal{F}$.

Proof. We need to prove that $Ax_k = b$ and $Ay_k = b$ for all $k \geq 0$ if $Ax_0 = Ay_0 = b$. We use the technique of mathematical induction to prove this. Since this clearly holds for $k = 0$, we next prove that if $Ax_k = Ay_k = b$, then $Ax_{k+1} = Ay_{k+1} = b$. From (5.2a) and (5.5a), we have

$$\begin{aligned} Ax_{k+1} =& Ay_k - \alpha A \nabla f^\epsilon(y_k) \\ =& Ay_k - \alpha A(\nabla f(y_k) - \nabla^2 f(y_k) A^\top [AA^\top]^{-1}(Ay_k - b) \\ & - A^\top [AA^\top]^{-1} A \nabla f(y_k) + \frac{2}{\epsilon} A^\top (Ay_k - b)). \end{aligned}$$

Substituting $Ay_k = b$, the above expression evaluates to b independent of $\epsilon \geq 0$. Then from (5.2c), one has $Ay_{k+1} = b$. Since the argument above is independent of the values of a_k for all $k \in \mathbb{N}$, it holds for the strongly convex case (5.2d) as well, thus completing the proof by induction. □

As a consequence of this result, if the trajectory starts in the feasible set $\mathcal{F}$, then it remains in it forever. This observation allows us to ensure the convergence rate guarantee for any convex objective function.

Corollary 5.2.7. ***(Accelerated convergence with feasible initialization).*** *For the optimization problem* (5.1) *and arbitrary* $\epsilon \geq 0$*, the algorithm* (5.2) *initialized in* $\mathcal{F}$ *enjoys the guarantee* (5.3) *on convergence to the optimal value.*

Proof. Note that $f^\epsilon(x) = f(x)$ whenever $Ax = b$, and hence by definition, is automatically (strongly) convex on $\mathcal{F}$ regardless of the value of ϵ. The convergence guarantee follows from this fact together with Lemma 5.2.6. □

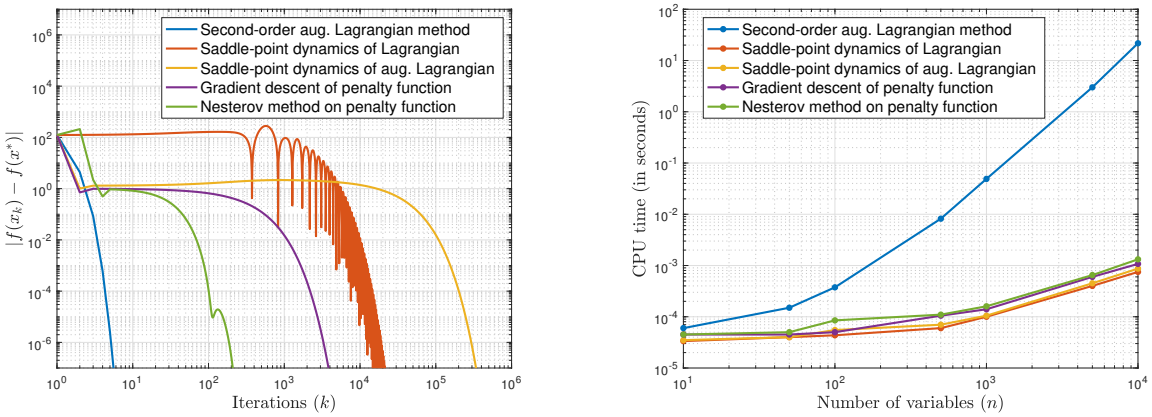
Remark 5.2.8. ***(Robustness of the proposed approach).*** *Given any* $x_0 \in \mathbb{R}^n$*, one can find a feasible initial point* $x_0 - A^\top [AA^\top]^{-1}(Ax_0 - b)$ *by projecting* x_0 *onto the feasible set* $\mathcal{F}$*, and then*

implement Nesterov's accelerated method with the projected gradient as $(I - A^\top[AA^\top]^{-1}A)\nabla f(x)$*. In fact, this projected gradient method coincides with the approach proposed here when evaluated over* $\mathcal{F}$*. The advantage of our approach resides in the incorporation of error-correcting terms incorporating the value of* $Ax - b$*, cf.* (5.5a)*, that penalize any deviation from the feasible set and hence provide additional robustness in the face of disturbances. By contrast, the projected gradient approach requires either an error-free execution or else, if error is present, the trajectory may leave and remain outside the feasible set unless repeated projections of the updated state are taken. The inherent robustness property of the approach proposed here is especially important in the context of distributed implementations, cf. Remark 5.1.1, where agents need to collectively estimate (and hence only implement approximations of)* $A^\top[AA^\top]^{-1}A\nabla f(x)$ *and taking the projection in a centralized fashion is not possible. The approach proposed here can also be extended to problems with convex inequality constraints, cf. [DG89], whereas computing the projection in closed form is not possible for general convex constraints.* •

5.3 Simulations

In this section, we show the effectiveness of the proposed approach through numerical simulations. We consider

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \sum_{i=1}^{n} \frac{1}{2}\beta_i x_i^2 + \gamma_i \exp(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i = 100, \end{aligned}$$



(a) Error evolution for $n = 50$

(b) CPU time per iteration

Figure 5.1: Performance comparison of the proposed algorithm with the second-order augmented Lagrangian method, the saddle-point dynamics applied to the Lagrangian and the augmented Lagrangian, respectively, and the gradient descent of the penalty function.

where $\beta_i, \gamma_i \in \mathbb{R}_{>0}$. We evaluate different scenarios with values of n as 10, 50, 100, 500, 1000, 5000 and 10000. We take $\mathcal{D} = \{x \in \mathbb{R}^n \mid \|x\|_\infty \leq 5, \sum_{i=1}^{n} x_i - 100 \leq 50\}$. By Corollary 5.2.4, for $n = 50$, the penalty function is strongly convex on $\mathcal{D}$ with parameter $s = 0.01$ for all $\epsilon \in (0, \bar{\epsilon}_s]$, where $\bar{\epsilon}_s = 0.3603$. In our simulations, we use $\epsilon = 10^{-1}$ and $\alpha = 10^{-3}$, resp. Figure 5.1 compares the performance of the proposed method with the second-order augmented Lagrangian method [AO17], the saddle-point dynamics [CMLC18, QL19] applied to the Lagrangian and the augmented Lagrangian, resp., and the gradient descent applied to the penalty function. Figure 5.1(a) shows the evolution of the error between the objective function and its optimal value for $n = 50$. For the same level of accuracy, the number of iterations taken by the second-order augmented Lagrangian method is smaller by an order of magnitude compared to the proposed method. However, one should note that the second-order augmented Lagrangian method involves the inversion of Hessian, which becomes increasingly expensive as the number of variables increases (see also Remark 5.1.1). To illustrate this, Figure 5.1(b) shows the computation time per iteration of the algorithms in Matlab

version 2018a running on a Macbook Pro with 2GHz i5 processor and 8 GB ram. The time taken by the first-order algorithms is about the same, and is smaller by several orders of magnitude (depending on the number of variables) than the second-order augmented Lagrangian method. When both aspects (number of iterations and computation time per iteration) are considered together, the proposed approach outperforms the other methods, especially if the problem dimension is large.

Acknowledgements

This chapter, in full, is a reprint of the material [SC21a] as it appears in 'Nesterov acceleration for equality-constrained convex optimization via continuously differentiable penalty functions' by P. Srivastava and J. Cortés, in the IEEE Control Systems Letters, 2021. The dissertation author was the primary investigator and author of this paper. This work was partially supported by the ARPA-e NODES program, Cooperative Agreement DE-AR0000695 and NSF Award ECCS-1917177.

Chapter 6

Decentralized Event-Triggered Optimization via Agent-Supervisor Coordination

This chapter proposes decentralized resource-aware coordination schemes for solving network optimization problems defined by objective functions which combine locally evaluable costs with network-wide coupling components. These methods are well suited for a group of supervised agents trying to solve an optimization problem under mild coordination requirements. Each agent has information on its local cost and coordinates with the network supervisor for information about the coupling term of the cost. The proposed approach is feedback-based and asynchronous by design, guarantees anytime feasibility, and ensures the asymptotic convergence of the network state to the desired optimizer.

6.1 Problem Formulation

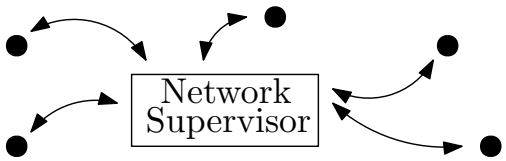


Figure 6.1: Communication infrastructure considered. Black dots represent the agents and the edges represent the communication links.

We consider network optimization problems where the objective function is the aggregate of individual objective functions, one per agent, and an additional function which couples all agents' states, which is non-separable in general. Formally, consider a network of $n \in \mathbb{Z}$ agents and a supervisor, cf. Figure 6.1, collectively seeking to solve

$$\min_{x \in \mathcal{X}} \quad \underbrace{\sum_{i=1}^{n} f_i(x_i)}_{f(x)} + g(x), \tag{6.1}$$

where, for $i \in \{1, \dots, n\}$, $f_i : \mathbb{R} \to \mathbb{R}$ is the local cost function of agent i, $g : \mathbb{R}^n \to \mathbb{R}$ is a function coupling all agents' states, $\mathcal{X} = \prod_{i=1}^{n} \mathcal{X}_i$ is the constraint set, and $\mathcal{X}_i$ is agent i's constraint set. Agent $i \in \{1, \dots, n\}$ has knowledge of its local state $x_i \in \mathbb{R}$, constraint set $\mathcal{X}_i$ and cost f_i, and relies on the network supervisor to obtain information pertaining the coupling cost g. For simplicity of exposition, we assume that (6.1) has a unique solution x^*, albeit the results of the paper can be extended with minor modifications to the case of multiple optimizers. We make the following assumptions on the cost functions and the constraints.

Assumption 6.1.1. ***(Convexity and Lipschitz gradients).*** *The functions $\{f_i\}_{i=1}^n$ and g are convex; the functions $\{f_i\}_{i=1}^n$ are twice continuously differentiable, and g is continuously differentiable with*

locally Lipschitz gradient; and the sets $\{\mathcal{X}_i\}_{i=1}^n$ *are compact and convex.*

Our goal is to design a decentralized algorithm that allows the agents to collectively solve (6.1). We want the algorithm to be anytime, meaning that if the network state starts feasible, it remains so during the algorithm's execution. This anytime nature is desirable in applications where the optimization problem is not stand-alone and its solution serves as an input to another layer in the control design. In such cases, the algorithm should yield a feasible solution even if terminated in finite time. Note that, without the presence of the coupling function g, (6.1) could be solved easily by having each agent i solve a local optimization problem with the function f_i over the constraint set $\mathcal{X}_i$. Instead, the presence of g couples the agents' decisions. Since information about g is not available at all times, we seek to endow the individual agents with a criterion that allows them to determine when to query the supervisor in an opportunistic fashion – this is what corresponds to the event-triggered component of the algorithmic solution. Based on the application at hand and the supervisor's capabilities, the coordination between the supervisor and the agents could either be sensing-based or computation-based, as defined next:

(i) *Sensing-based:* each agent $i \in \{1, \dots, n\}$ can evaluate $\nabla_{x_i} g$ with its local information and the one broadcast from the supervisor when an agent asks for an update. This is because the supervisor has access from its own measurements to enough knowledge about g. In this scheme, the states of all agents remain private (the virtual power plant case in [CBCZ20] is an example falling in this case);

(ii) *Computation-based:* the supervisor knows the functional form of the cost. Whenever an agent asks for an update, the supervisor gathers the state of all the agents, evaluates ∇g, and broadcasts it to the agents (scheduling links and channels for transmission of information in

wireless networks is an example scenario for this case, see e.g., [ZSJ09]).

The forthcoming design and the ensuing analysis can be applied to both scenarios.

6.2 Event-Triggered Coordination for Unconstrained Problems

Here, an event-triggered decentralized algorithm to solve (6.1) when $\mathcal{X} = \mathbb{R}^n$ is provided. Consider the standard gradient-descent dynamics

$$\dot{x} = -\lambda(\nabla f(x) + \nabla g(x)), \tag{6.2}$$

where $\lambda > 0$ is a design parameter. For agent $i \in \{1, \dots, n\}$, this takes the form

$$\dot{x}_i = -\lambda(\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x)). \tag{6.3}$$

From an implementation viewpoint, the first term in (6.3) can be evaluated locally by each agent. However, the second term in (6.3) requires knowledge of the partial gradient of the coupling cost with respect to the agent's state, which entails continuous communication with the supervisor. We avoid this by designing an event-triggered scheme that has the supervisor broadcast the information needed to compute $\nabla g(x)$ in an opportunistic fashion. Introducing the shorthand notation $x^k = x(t_k)$, consider the dynamics

$$\dot{x} = -\lambda(\nabla f(x) + \nabla g(x^k)) \quad t_k \leq t < t_{k+1}. \tag{6.4}$$

To implement (6.4), the network supervisor needs to broadcast the information required to compute $\nabla g(x)$ only at some specified times $\{t_k\}_{k=0}^{\infty}$. Here, $\nabla g(x^k)$ is the equivalent of the input in the standard event-triggered control, cf. Section 2.7. Whenever convenient, we refer to the dynamics (6.4) as ψ_{ev}. The next result identifies a decentralized condition on the triggering times $\{t_k\}_{k=0}^{\infty}$ that ensures that the dynamics (6.4) is stable. By decentralized, we mean that each agent $i \in \{1, \dots, n\}$ can identify the triggering criterion locally without knowing the states of the other agents or the coupling function.

Proposition 6.2.1. ***(Decentralized trigger).*** *Let x^k be the state when the trigger was last implemented, with $x^k \neq x^*$, $\sigma \in (0, 1)$, and*

$$\mathcal{F}_k = \{x \in \mathbb{R}^n \mid f(x) + g(x) \leq f(x^k) + g(x^k)\}.$$

Then for all $\lambda > 0$, the dynamics (6.4) *is stable and the value of the objective function $f + g$ is non-increasing if the triggering times are updated according to*

$$t_{k+1} = \min_{i \in \{1,\dots,n\}} \min\{t > t_k \mid L_g |x_i - x_i^k| = \sigma |\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k)| \neq 0\}, \tag{6.5}$$

where L_g is the Lipschitz constant of ∇g over $\mathcal{F}_0$.

Proof. Consider the Lyapunov function $V : \mathbb{R}^n \to \mathbb{R}$

$$V(x) = f(x) + g(x) - f(x^*) - g(x^*), \tag{6.6}$$

whose Lie derivative is

$$\mathcal{L}_{\psi_{\text{ev}}} V(x) = -\lambda(z+e)^\top z \leq -\lambda\|z\|^2 \left(1 - \frac{\|e\|}{\|z\|}\right),$$

where $z = [z_1 \ \dots \ z_n]^\top$, $z_i = \nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k)$, and $e = \nabla g(x) - \nabla g(x^k)$. At $t = t_k$, we have $e = 0$; then the error starts increasing as x^k becomes obsolete. However, $\mathcal{L}_{\psi_{\text{ev}}} V \leq 0$ if we ensure that $\|e\| \leq \sigma\|z\|$. The direct evaluation of the latter condition requires complete information about the network state. However, note that $\|e\| \leq L_g^k \|x - x^k\|$, where L_g^k is the Lipschitz constant of ∇g over $\mathcal{F}_k$. Hence, we can guarantee the stability of (6.4) if

$$L_g^k \|x - x^k\| \leq \sigma\|z\|.$$

The triggering rule (6.5) ensures that the latter inequality is satisfied noting that the set $\mathcal{F}_k$ is forward invariant, $\mathcal{F}_{k+1} \subseteq \mathcal{F}_k$, and hence $L_g \geq L_g^k$ for all k. □

From Proposition 6.2.1, it is clear that if the agents have knowledge of (an upper bound of) L_g, they can check (6.5) locally and request the supervisor for an update accordingly. Although (6.5) guarantees that the dynamics (6.4) is stable, we still need to establish the convergence to x^* and whether the proposed event-triggered scheme is Zeno-free. We prove both facts in the next result.

Proposition 6.2.2. ***(Non-Zeno behavior and convergence to the optimizer).*** *With the notation of Proposition 6.2.1, if the triggering times are updated as* (6.5)*, then for all $\lambda > 0$, the MIET is lower bounded by $\tau = \frac{1}{\lambda H} \log(\sigma\lambda H / L_g + 1) > 0$, where $H = \max_{i\in\{1,\dots,n\}} \max_{x\in\mathcal{F}_0} \nabla^2_{x_i} f_i(x_i)$. Moreover, any trajectory of* (6.4) *converges asymptotically to x^*.*

Proof. If $x^k = x^*$, then the result is immediate. Assume then that $x^k \neq x^*$. Let $\mathcal{I} = \{i \mid z_i(t_k) \neq 0\}$.

Since for all $i \in \{1, \dots, n\}$, $\dot{z}_i = -\lambda \nabla^2_{x_i} f_i(x_i) z_i$, we deduce that $z_i = 0$ for all $t \in [t_k, t_{k+1})$ if $i \notin \mathcal{I}$. For $i \in \mathcal{I}$, we examine the evolution of $|x_i - x_i^k|/|z_i|$,

$$\begin{aligned}\frac{d}{dt}\frac{|x_i - x_i^k|}{|z_i|} &= \frac{(x_i - x_i^k) z_i}{\sqrt{(x_i - x_i^k)^2}\sqrt{z_i^2}} - \frac{z_i \dot{z}_i \sqrt{(x_i - x_i^k)^2}}{|z_i|^3} \\ &\leq 1 + \frac{|\dot{z}_i|}{|z_i|}\frac{|x_i - x_i^k|}{|z_i|} \leq 1 + \lambda H \frac{|x_i - x_i^k|}{|z_i|}. \end{aligned} \tag{6.7}$$

Now consider the differential equation $\dot{y} = 1 + \lambda H y$, $t_k \leq t < t_{k+1}$, with initial condition $y(t_k) = 0$, whose closed-form solution is given by

$$y = \frac{1}{\lambda H}\left(e^{\lambda H (t - t_k)} - 1\right), \quad t_k \leq t < t_{k+1}.$$

By the Comparison Principle, cf. [Kha02, Lemma 3.4], we have

$$\frac{|x_i - x_i^k|}{|z_i|} \leq \frac{1}{\lambda H}\left(e^{\lambda H (t - t_k)} - 1\right), \quad t_k \leq t < t_{k+1}.$$

Equating the right-hand side of the above inequality with σ / L_g implies that the inter-event time is lower bounded by τ provided $z_i \neq 0$ for all $t \in [t_k, t_{k+1})$ and each $i \in \mathcal{I}$. We reason by contradiction to prove this. Since the ratio $|x_i - x_i^k|/|z_i|$ is bounded, $z_i = 0$ only if $x_i - x_i^k = 0$. Let $\bar{t} = \min\{t > t_k \mid x_i - x_i^k = 0\}$. Since $x_i - x_i^k = 0$ and $z_i \neq 0$ at $t = t_k$, this means that the sign of z_i has to change before $\bar{t}$, and from the continuity of the dynamics, there exists $\hat{t} < \bar{t}$ such $z_i(\hat{t}) = 0$, which contradicts $z_i \neq 0$ for all $t \in [t_k, \bar{t})$. To prove the attractivity part, note that from Proposition 6.2.1, $\mathcal{L}_{\psi_{\text{ev}}} V \leq 0$, and hence, $\mathcal{L}_{\psi_{\text{ev}}} V(x) < 0$ for all $x \neq x^*$ as $z_i \neq 0$ for all $i \in \mathcal{I}$ and all $t \in [t_k, t_{k+1})$. □

Remark 6.2.3. ***(Differentiability of the local objective functions).*** *Note that ruling out Zeno be-*

havior in Proposition 6.2.2 relies on the functions $\{f\}_{i=1}^n$ *being twice continuously differentiable, while the dynamics* (6.4) *and the triggering condition* (6.5) *involve just first-order derivatives. We believe, although we do not pursue it here, that Proposition 6.2.2 can be extended for the case when the separable component of the objective function is just continuously differentiable, using tools from nonsmooth analysis, e.g., [Cla83, Cor08].* •

Remark 6.2.4. ***(Self-triggered implementation).*** *In the absence of errors in the solution of the differential equations by the individual agents, the criterion* (6.5) *can also be implemented in a self-triggered fashion. In fact, we can write it as*

$$t_{k+1}^i = \min\{t > t_k \mid L_g|x_i - x_i^k| = \sigma|\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k)| \neq 0\},$$
$$t_{k+1} = \min_{i\in\{1,\dots,n\}} t_{k+1}^i.$$

This means that, with the information provided at time t_k*, each agent* $i \in \{1, \dots, n\}$ *can compute* t_{k+1}^i *by solving its differential equation, and convey it to the supervisor, which can then schedule the next triggering event at* t_{k+1}. •

6.3 Event-Triggered Coordination for Constrained Problems

To deal with constrained problems, we build on the continuous projected dynamics, cf. Section 2.6.2, which takes the form

$$\dot{x} = \Pi_{\mathcal{X}}(x - \lambda(\nabla f(x) + \nabla g(x))) - x, \tag{6.8}$$

where $\lambda > 0$. Its event-triggered counterpart is

$$\dot{x} = \Pi_{\mathcal{X}}(x - \lambda(\nabla f(x) + \nabla g(x^k))) - x, \tag{6.9}$$

for $t \in [t_k, t_{k+1})$. Whenever convenient, we refer to the dynamics (6.9) as ψ_{evco}. The following result identifies a decentralized condition on the triggering times $\{t_k\}_{k=0}^{\infty}$ that ensures the stability of (6.9).

Proposition 6.3.1. ***(Decentralized trigger for constrained problems).*** *Let x^k be the state when the trigger was last implemented, with $x_k \neq x^*$, and $\sigma \in (0, 1)$. If $x(0) \in \mathcal{X}$, then for all $\lambda > 0$, $x(t) \in \mathcal{X}$ for all $t > 0$, the dynamics* (6.9) *is stable and the value of the objective function $f + g$ is non-increasing if the triggering times are updated as*

$$t_{k+1} = \min_{i \in \{1,\dots,n\}} \min\{t > t_k \mid \lambda \bar{L}_g |x_i - x_i^k| = \sigma |\Pi_{\mathcal{X}_i}(x_i - \lambda(\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k))) - x_i| \neq 0\}, \tag{6.10}$$

where $\bar{L}_g$ is the Lipschitz constant of ∇g over $\mathcal{X}$.

Proof. We start by noting that from Theorem 2.6.3, for $t \in [t_k, t_{k+1})$, positive invariance of the feasible set $\mathcal{X}$ under (6.9) can be established by taking $h(x) \equiv f(x) + \nabla g(x^k)^\top x$. To prove stability, consider again the Lyapunov function candidate V defined in (6.6), whose Lie derivative is now given by

$$\mathcal{L}_{\psi_{\text{evco}}} V(x) = (\nabla f(x) + \nabla g(x^k) + e)^\top \bar{z},$$

where $\bar{z} = [\bar{z}_1 \ \dots \ \bar{z}_n]^\top$, $\bar{z}_i = \Pi_{\mathcal{X}_i}(x_i - \lambda(\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k))) - x_i$, and $e = \nabla g(x) - \nabla g(x^k)$. It is well known, cf. [Gao03], that for a convex set Ω

$$(u - \Pi_\Omega(u))^\top (\Pi_\Omega(u) - v) \geq 0,$$

for all $v \in \Omega$ and all $u \in \mathbb{R}^n$. With $\Omega = \mathcal{X}$, $v = x$, and $u = x - \lambda(\nabla f(x) + \nabla g(x^k))$, this implies that

$$(\lambda \nabla f(x) + \lambda \nabla g(x^k) + \bar{z})^\top \bar{z} \leq 0.$$

Using this, we upper bound the Lie derivative as

$$\mathcal{L}_{\psi_{\text{evco}}} V(x) \leq -\frac{1}{\lambda} \bar{z}^\top \bar{z} + e^\top \bar{z} \leq -\| \bar{z} \|^2 \left(\frac{1}{\lambda} - \frac{\|e\|}{\| \bar{z} \|} \right).$$

This expression is analogous to that of $\mathcal{L}_{\psi_{\text{ev}}} V$ in the proof of Proposition 6.2.1, and a similar argument concludes the proof. □

As in the unconstrained case, without excluding Zeno behavior, Proposition 6.3.1 is not enough to conclude the asymptotic convergence to x^*.

Proposition 6.3.2. ***(Non-Zeno behavior and convergence to the optimizer in the constrained case).*** *With the notation of Proposition 6.3.1, if the triggering times are updated as* (6.10)*, then for all $\lambda < \bar{\lambda} = 1/\bar{H}$, the MIET is lower bounded by $\bar{\tau} = \log(\sigma/\lambda \bar{L}_g + 1) > 0$, where $\bar{H} = \max_{i \in \{1,\dots,n\}} \max_{x_i \in \mathcal{X}_i} \nabla^2_{x_i} f_i(x_i)$. Moreover, any trajectory of* (6.9) *with $x(0) \in \mathcal{X}$ converges asymptotically to x^*.*

Proof. Since $\{\mathcal{X}_i\}_{i=1}^n$ are convex, without loss of generality let $\mathcal{X}_i = \{x_i \in \mathbb{R} \mid \underline{x}_i \leq x_i \leq \overline{x}_i\}$ for all i. For each agent $i \in \{1, \dots, n\}$, define $u_i : \mathcal{X}_i \to \mathbb{R}$ as $u_i(x_i) = x_i - \lambda(\nabla_{x_i} f_i(x_i) + \nabla_{x_i} g(x^k))$. The derivative of u_i w.r.t x_i is given by

$$\frac{du_i}{dx_i} = 1 - \lambda \nabla^2_{x_i} f_i(x_i).$$

For a given $i \in \{1, \dots, n\}$, the sign of du_i/dx_i at any $x_i \in \mathcal{X}_i$ depends on the value of λ and $\nabla^2_{x_i} f_i(x_i)$. If $\lambda < \bar{\lambda}$, then $du_i/dx_i > 0$ for all $x_i \in \mathcal{X}_i$. This means that if there is a point $\hat{x}_i \in \mathcal{X}_i$ such that $\Pi_{\mathcal{X}_i}(u_i(\hat{x}_i)) = \overline{x}_i$, then $\Pi_{\mathcal{X}_i}(u_i(x_i)) = \overline{x}_i$ for all $x_i > \hat{x}_i$. Similarly, if there is a point $\tilde{x}_i \in \mathcal{X}_i$ such that $\Pi_{\mathcal{X}_i}(u_i(\tilde{x}_i)) = \underline{x}_i$, then $\Pi_{\mathcal{X}_i}(u_i(x_i)) = \underline{x}_i$ for all $x_i < \tilde{x}_i$. Therefore, $\dot{\bar{z}}_i$ can be represented as a set-valued map, cf. [Cor08],

$$|\dot{\bar{z}}_i| = \begin{cases} |\lambda \nabla^2_{x_i} f_i(x_i)\, \bar{z}_i| & \tilde{x}_i < x_i < \hat{x}_i, \\ [|\lambda \nabla^2_{x_i} f_i(x_i)\, \bar{z}_i|, |\bar{z}_i|] & x_i = \tilde{x}_i, \hat{x}_i, \\ |\bar{z}_i| & \underline{x}_i \leq x_i < \tilde{x}_i,\ \hat{x}_i < x_i \leq \overline{x}_i. \end{cases}$$

Since $\lambda < \bar{\lambda}$, we have an expression similar to (6.7) for $\frac{d}{dt}|x_i - x_i^k|/|\bar{z}_i|$ for all i with $\bar{z}_i(t_k) \neq 0$, with λH replaced by 1. The remainder of the argument follows analogously to the proof of Proposition 6.2.2. □

The upper bound on λ in Proposition 6.3.2 is conservative in general. In fact, the dynamics (6.9) with the triggering rule (6.10) may be Zeno-free even if this condition is not satisfied, something that we have observed in simulation.

6.4 Simulations

Here we illustrate our coordination approach in a power distribution scenario, where n generators managed by a distribution system operator that acts as the network supervisor seek to minimize the total power generation cost. The active power output of generator $i = \{1, \dots, n\}$ is x_i and the power that flows from the external grid to the distribution grid through the substation, labeled as node 0, is denoted as x_0. Assuming the power line losses are negligible, x_0 can be approximated by, see e.g., [CBCZ20],

$$x_0(x) \simeq -\sum_{i=1}^{n} x_i + c, \tag{6.11}$$

where c depends on the grid load. We choose polynomial costs for power injection at each node $i = \{0, 1, \dots n\}$. Each function f_i is known only by generator $i \in \{1, \dots, n\}$, and the network supervisor has access only to f_0. By substituting (6.11) into $f_0(x_0)$, the problem of minimizing the total power cost is equivalent to solving (6.1) with $g(x) = f_0(-\sum_{i=1}^{n} x_i + c)$. From the chain rule,

$$\nabla_{x_i} g(x) = -\nabla_{x_0} f_0(x_0),$$

for all $i \in \{1, \dots, n\}$. Since the supervisor can measure x_0, whenever there is an update request, it can evaluate ∇g directly and broadcast it to the generators. This corresponds to the sensing-based scenario (cf. Section 6.1).

We test the algorithms resulting from the triggering criteria (6.5) and (6.10) on a single-phase equivalent of the IEEE 37-bus test feeder, reported in Figure 6.2. The network has five generators. The load buses are a mixture of constant-current, constant-impedance, and constant-power loads [Ker01]. The initial active and reactive power demands are 2 MW and 1 MVAR,

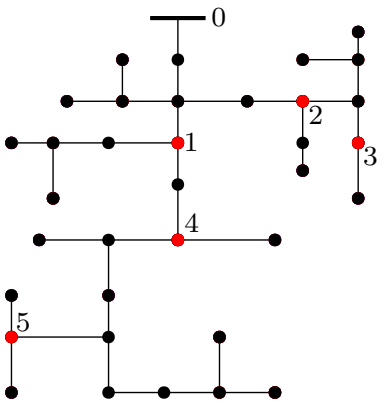


Figure 6.2: The IEEE 37-bus test feeder, where node 0 represents the supervisor, red nodes represent the micro-generators and black nodes represent the loads; edges represent the electrical connection between the nodes.

respectively. At $t = 40$ seconds, the active power demand increases by 1 MW. The algorithms are simulated using the nonlinear exact AC power flow solver MATPOWER [ZMSG11]. Since the proposed dynamics evolve in continuous time between the triggering instances, we use a first-order Euler discretization for the MATLAB implementation with stepsize 10^{-2}. The values of λ and σ are taken as 0.2 and 0.9, respectively.

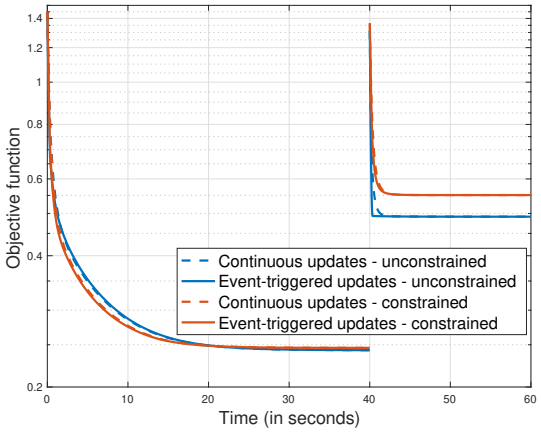
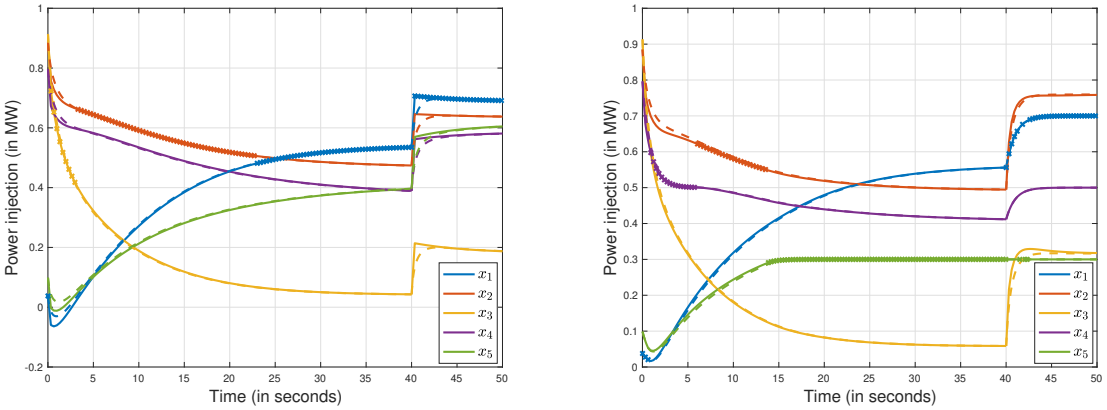


Figure 6.3: Evolution of the objective function for the unconstrained and the constrained cases using the proposed event-triggered mechanisms.

We provide two sets of simulations based on whether the generation capacities of the micro-



(a) Unconstrained (b) Constrained

Figure 6.4: State evolution using the proposed event-triggered coordination algorithms. $\times$ markers denote the triggering instances for the corresponding agent.

generators are unconstrained or constrained. For the constrained case, for each $i \in \{1, \dots, 5\}$, $\mathcal{X}_i = \{x_i \mid 0 \leq x_i \leq \overline{x}_i\}$, where $\{\overline{x}_i\}_{i=1}^5$ are taken as 0.7 MW, 1 MW, 0.8 MW, 0.5 MW and 0.3 MW, respectively. Figure 6.3 reports the objective function evolution for the proposed event-triggered dynamics (6.4) and (6.9) to their standard counterparts (6.2) and (6.8), respectively. The load increase at $t = 40$ seconds changes the optimal configuration and causes the jump in the objective function. Figure 6.4(a) and Figure 6.4(b) show the evolution of the injected active powers of all the agents using the proposed event-triggered dynamics (solid lines) and the continuous dynamics for the unconstrained and constrained case, respectively. Note that depending on the operating region, different agents request for the information update non-uniformly. Lastly, for a fixed load, we simulate 10 different initial conditions for 60 seconds and observe an average MIET of 3.3×10^{-1} seconds for the unconstrained and 2.0×10^{-1} seconds for the constrained case, with respective standard deviations of 6.7×10^{-3} seconds and 23.1×10^{-3} seconds, and 177 (unconstrained) and 190 (constrained) average updates per resource-aware execution.

Acknowledgements

This chapter, in full, has been submitted for publication of the material [SCC21a] as it may appear as 'Agent-supervisor coordination for decentralized event-triggered coordination' by P. Srivastava, Guido Cavraro, and J. Cortés, in the IEEE Control Systems Letters, 2021. The dissertation author was the primary investigator and author of this paper. This work was supported by the National Renewable Energy Laboratory (NREL) under Contract DE-AC36-08GO28308.

Part II

Applications of Constrained Optimization in Frequency Regulation from DERs

Chapter 7

Participation of DERs in Frequency Regulation Markets

In this chapter, we propose a framework for collections of distributed energy resources (DERs), combined to form microgrids and controlled by aggregators, to participate in frequency regulation markets. Our approach covers both the identification of bids for the market clearing stage and the mechanisms for the real-time allocation of the regulation signal. The proposed framework, cf. Figure 7.1, is hierarchical, consisting of a top layer and a bottom layer. The top layer consists of the aggregators communicating in a distributed fashion to optimally disaggregate the regulation signal requested by the system operator. The bottom layer consists of the DERs inside each microgrid whose power levels are adjusted so that the tie line power matches the output of the corresponding aggregator in the top layer. The coordination at the top layer requires the knowledge of cost functions, ramp rates and capacity bounds of the aggregators. We develop meaningful abstractions for these quantities respecting the power flow constraints and taking into account the load uncertainties, and propose a provably correct distributed algorithm for optimal disaggregation of

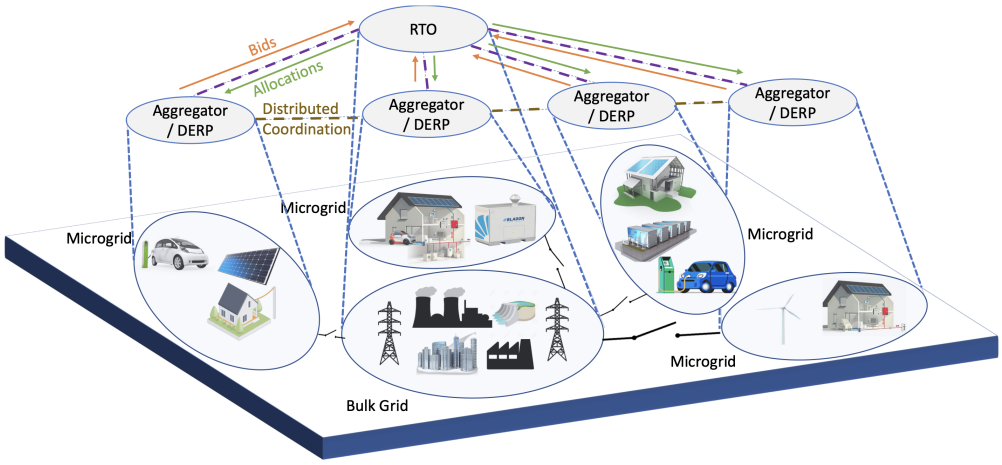


Figure 7.1: Power system framework considered. The Regional Transmission Organization (RTO) monitors the bulk grid and coordinates with the aggregators, which communicate with each other, and control the resources inside their respective microgrids.

regulation signal amongst the microgrids.

7.1 Frequency Regulation with Microgrids

We are interested in coordinating power aggregators to collectively provide frequency regulation. An aggregator is a virtual entity that aggregates the actions of a group of distributed energy resources to act as a single whole. In this paper, we identify an aggregator with a microgrid, but in general it may correspond to other entities (such as, for instance, a collection of microgrids). We consider microgrids with fast responding DERs (e.g., photovoltaics (PVs), electric vehicles, batteries and small generators) as they operate on time scales that match those needed for frequency regulation.

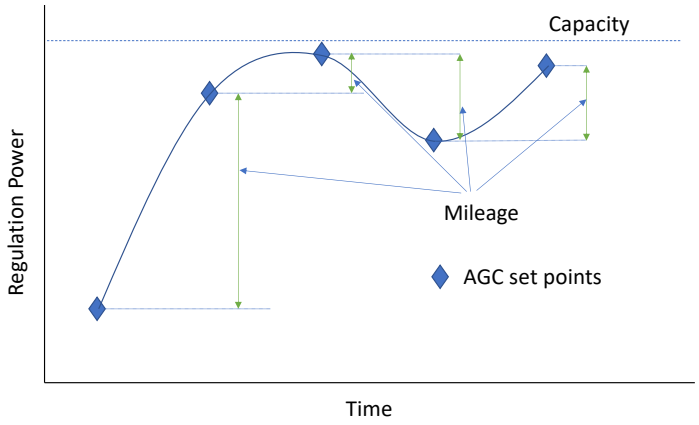


Figure 7.2: Illustration of the computation of capacity and mileage.

7.1.1 Review of Current Practice

The frequency regulation market is operated by an RTO to regulate the system frequency at its nominal value. To achieve this, the RTO coordinates the response of participating energy resources in a centralized fashion to assign the regulation signal and restore the power balance of the grid. Different RTOs follow slightly different procedures for the frequency regulation markets. The procedure followed by CAISO has the following stages, see e.g., [KM14, Cal12]:

[CP1]: Market clearance. All participating resources submit their capacity bids, capacity price bids, and mileage price bids to the RTO. Capacity bids are the maximum amount of regulation (up or down) that the resource can provide. Capacity price bids are the unit price of providing these regulations. Mileage is the sum of the absolute change in AGC set points, which corresponds to the summation of the vertical lines in Figure 7.2. The mileage price bid is the cost for unit change in regulation. Typically, expected mileages are calculated from historical data and resources do not submit mileage bids. Using the bids submitted by the resources, the RTO solves an optimization problem to minimize the expected cost and uses its solution to clear the market with uniform prices for capacity and mileage across the resources. The RTO then sends each resource its capacity and

mileage allocation. This off-line process happens only once per regulation event.

[CP2]: Allocation of regulation signal to each resource. The RTO sends the regulation set points to each of the procured energy resources every 2-4 seconds for the entire regulation period, which is usually 10-15 minutes. The regulation set points are computed from the AGC signal in real time in proportion to the procured mileage of each resource. In case the assigned capacity of a resource is violated, the overshoot power is redistributed to the other resources in proportion to their assigned mileages.

[CP3]: Real-time tracking of regulation signal. Once the regulation set points have been assigned, the resources need to track them in real time.

Payment to the resources comprises of two components, capacity payment and mileage payment. The capacity payment is done based on the assigned capacity in [CP1] while the mileage payment is done based on the actual mileage provided which reflects the performance of the resources while tracking the assigned signal in [CP3].

Limitations of Current Practice

The centralized way of assigning the set points to the resources in [CP2] relies on the fixed number of resources with fixed generation capacities procured in [CP1], which are available for the entire regulation period. This is problematic in the context of aggregators, as they are subject to variabilities and uncertainties associated with the DERs inside them. Even if the DERs inside the microgrid participating stay during the regulation period, the users inside the microgrid can change their power consumption, which in turn leads to changes in the effective regulation capacity. Furthermore, in current practice, there is no direct consideration of the operational costs of the resources, which may result in suboptimal power allocation. Instead, we argue that the assignment

of the regulation signal should be done, at each time step, in a way that optimizes the aggregate cost functions of the resources and takes into account their (possibly dynamic) operational limits. We refer to this approach as the RTO-DERP coordination problem. This idea has also been pointed out in the past by CAISO for traditional energy resources, cf. [BHH12]. The lack of robustness and the information sharing requirements of centralized schemes motivate the investigation of distributed schemes to solve the RTO-DERP coordination problem.

Challenges for Frequency Regulation from Microgrids

Here we describe the challenges specific to microgrid participation in frequency regulation markets. First, note that solving the RTO-DERP coordination problem with microgrids requires the identification, or rather the abstraction, of aggregate cost functions and regulation capacity bounds based on the cost functions and flexibilities of their DERs. Second, the determination of capacity bids requires taking into account the uncertainties associated with the microgrids. There is a need to calculate bids for each regulation interval, as they might need to considerably change from one interval to the next. Even within a regulation interval itself, the power level of the uncontrollable nodes might vary significantly. Third, the mileage bids should be determined by taking into account the dependency of ramp rates on the composition and participation of the individual DERs. The current method of calculating expected mileages in [CP1] makes sense for conventional resources as their ramp rates are fixed and historical data provides reliable accuracy. In the case of microgrids, individual resources keep changing and as a result, ramp rates do not remain constant over time. Also, the performance of participating resources for one regulation period to another might be substantially different.

7.1.2 Problem Statement

Consider N microgrids, each controlled by an aggregator. To enable microgrid participation in the frequency regulation market, we focus on [CP1] and [CP2]. Based on the proposed framework in Figure 7.1 and the discussion in Section 7.1.1, our goal is to equip the aggregators with abstracted bids to enable their participation in the market and design a distributed optimization algorithm to solve the RTO-DERP coordination problem. We formalize the following problems.

[P1]: Meaningful abstractions for the microgrid. To enable the submission of bids in [CP1], each aggregator needs to quantify the maximum up/down regulation capacity that the microgrid can provide, the unit cost of providing such regulation, and the ramp rate at which the microgrid can change its power level. Our first goal is therefore to provide meaningful abstractions for these elements, capturing the aggregate behavior of the composing DERs, and specifically cost functions and ramp rate functions of the microgrids for [P2] below, a problem we tackle in Section 7.2.

[P2]: RTO-DERP distributed coordination. The RTO-DERP coordination problem for computing the set points for each resource advocated for [CP2] consists of an economic dispatch problem with ramp rate constraints *at every instant of the regulation interval*. Formally, for x_r regulation at a given time instant, we have

$$\begin{aligned} \min_{x} \quad & f(x) = \sum_{\mathrm{i}=1}^{N} f_{\mathrm{i}}(x_{\mathrm{i}}) \\ \text{s.t.} \quad & \sum_{\mathrm{i}=1}^{N} x_{\mathrm{i}} = x_r \\ & \underline{x_{\mathrm{i}}} \leq x_{\mathrm{i}} \leq \overline{x_{\mathrm{i}}} \quad \forall\, \mathrm{i} \\ & |x_{\mathrm{i}} - x_{\mathrm{i}}^{-}| \leq R_{\mathrm{i}}(x_{\mathrm{i}}^{-}) \quad \forall\, \mathrm{i}, \end{aligned} \tag{7.1}$$

where $x \in \mathbb{R}^N$ is the vector of regulation power from the microgrids, $f_{\mathrm{i}}(x_{\mathrm{i}})$ is the cost of x_{i} regulation for microgrid i, $\underline{x_{\mathrm{i}}}$ and $\overline{x_{\mathrm{i}}}$ are the lower and upper bounds of regulation for microgrid i which are bounded by the solutions of [P1] and determined by [CP1] for a specific regulation period, x_{i}^{-} is the regulation that the microgrid i was providing at the previous instant, and $R_{\mathrm{i}}(x_{\mathrm{i}})$ is the ramp rate of the microgrid when it is providing regulation x_{i}. Because of the ramp constraints present in (7.1), this problem might not be always feasible (since mileage requirements set by the RTO while clearing the market in [CP1] capture the average mileage required, and not the extreme cases). In such cases, we want to minimize the error between the procured regulation and the required one. We tackle these in Section 7.3.

7.2 Microgrid Abstractions

Consider a microgrid with $n \in \mathbb{Z}_{>1}$ buses, described by $\mathcal{G}_m = (\mathcal{V}, \mathcal{E}, \mathsf{A})$. Without loss of generality, we assume that the first bus is connected to the bulk grid through a tie line. We partition the remaining set of buses as $\mathcal{V}_g \cup \mathcal{V}_l$, where $\mathcal{V}_g$ is the set of the generators and controllable loads, referred to as controllable nodes in the following, and $\mathcal{V}_l$ denotes the set of the fixed loads and devices outside the aggregator's authority, referred collectively as the uncontrollable nodes. Let $n = |\mathcal{V}|$, $n_g = |\mathcal{V}_g|$, $n_l = |\mathcal{V}_l|$ and $m = |\mathcal{E}|$. Following [FZD18], we assume that the lines connecting various buses inside the microgrid are lossless and inductive. In case the electrical lines inside the microgrid are lossy with sufficiently uniform resistance to reactance ratios, they could still be represented via a lossless model obtainable through a linear transformation [DSPB16]. Since the voltage dynamics governed by the voltage droop controllers operate at much faster scale than the secondary frequency regulation [LD14], we assume the voltage magnitude of every bus

to be approximately 1 p.u. Further, we assume that the network and inverter filter dynamics are fast enough so that we can model them as power injections with no dynamics [AAC+17, ZH12]. We adopt the convention that the value of the power injection is negative if the device consumes power and vice versa. The power level of each controllable node $p \in \mathcal{V}_g$ is denoted by g_p, with g_p^0 denoting the baseline generation/consumption. The power level of each uncontrollable node $q \in \mathcal{V}_l$ is denoted by l_q. We denote the incoming power through the tie line by P and its baseline value by P^0. When the microgrid provides frequency regulation, the value of the tie line power P is

$$P = P^0 + x,$$

where x is the allocated AGC signal. Note that since we model P as the incoming power from bulk grid, x would be negative when the microgrid is providing up regulation. Following [ZDG20], we assume that $\mathcal{G}_m$ is a graph with non-overlapping loops, meaning that there is no common edge between any loops. This assumption helps linearize the power flow equations inside the microgrid, which are given by

$$[P; g; -l] = \mathsf{M}\,\omega \tag{7.2a}$$

$$|\omega| \leq \overline{\omega}, \tag{7.2b}$$

where $g \in \mathbb{R}^{n_g}$ and $l \in \mathbb{R}^{n_l}$ are the vectors of controllable and uncontrollable nodes, resp., $\mathsf{M} \in \mathbb{R}^{n \times m}$ is the incidence matrix of the graph, $\omega \in \mathbb{R}^m$ is the vector of line flows and $\overline{\omega} \in \mathbb{R}^m$ is the vector of maximum permissible flows. Since the columns of the fundamental loop matrix form a

basis for the null space of the incidence matrix, cf. [SR61, Theorem 4-6], we write (7.2) as

$$|\,\mathsf{M}^{+}[\mathbf{1}^{\top}l - \mathbf{1}^{\top}g; g; -l] + \mathsf{N}\,\gamma| \leq \overline{\omega}, \tag{7.3}$$

where M^{+} denotes the Moore-Penrose pseudoinverse of M, $\mathsf{N} \in \mathbb{R}^{m\times(m-n+1)}$ is the fundamental loop matrix of $\mathcal{G}_m$, and $\gamma \in \mathbb{R}^{m-n+1}$.

7.2.1 Capacity Bounds

The microgrid needs to solve an optimization problem to find the maximum up (or down) regulation that it can provide. For up regulation, the power consumption of the microgrid is less than the baseline power. Since the latter is constant for the regulation period, computing the capacity is equivalent to minimizing P while satisfying the power flow constraints. If the power level of uncontrollable nodes is constant for the entire regulation period, then the problem reads as

$$\begin{aligned} \min_{g,\omega} \quad & P \\ \text{s.t.} \quad & [P; g; -l] = \mathsf{M}\,\omega \\ & \underline{g} \leq g \leq \overline{g}, \quad |\omega| \leq \overline{\omega}, \end{aligned} \tag{7.4}$$

where $\underline{g}$ and $\overline{g}$ are the vectors of minimum and maximum possible power levels of controllable nodes, respectively. If $\underline{P}$ denotes the solution of (7.4), then the maximum up regulation is $\overline{x} = \underline{P} - P^0$. The maximum down regulation $\underline{x}$ can be obtained solving a similar maximization problem.

The formulation (7.4) assumes the power level of the uncontrollable nodes remains constant, and therefore does not take into account the varying nature of the loads. In practice, this

makes sense for a specific regulation instant, and would rarely be the case for the whole regulation period. Instead, a more robust way of calculating the capacity bounds that the aggregator can use in bidding for the whole regulation period is to account for worst-case scenarios, i.e., taking the expected maximum value for the uncontrollable nodes while computing the maximum up regulation. Although robust to variations in the uncontrollable nodes' powers, this way of computing capacity bounds might be too conservative and, in fact, might prohibit the microgrid from participating in the regulation market at all. As an alternative, we propose a reformulation of problem (7.4) based on chance constraints. Using (7.3), we rewrite the optimization problem (7.4) as

$$\begin{aligned} \min_{g,\gamma,t} \quad & t \\ \text{s.t.} \quad & t \geq \mathbf{1}^\top l - \mathbf{1}^\top g \\ & |\,\mathsf{M}^+[\mathbf{1}^\top l - \mathbf{1}^\top g; g; -l] + \mathsf{N}\gamma| \leq \overline{\omega} \\ & \underline{g} \leq g \leq \overline{g}. \end{aligned} \tag{7.5}$$

Assume that a probability distribution describing the power levels of uncontrollable nodes at any instant of the regulation period is available. To account for load variability, we instead consider the following chance-constrained optimization

$$\begin{aligned} \min_{g,\gamma,t} \quad & t \\ \text{s.t.} \quad & \Pr(t \geq \mathbf{1}^\top l - \mathbf{1}^\top g) \geq 1 - \epsilon' \\ & \Pr(|\,\mathsf{M}^+[\mathbf{1}^\top l - \mathbf{1}^\top g; g; -l] + \mathsf{N}\gamma|_j \leq \overline{\omega}_j) \geq 1 - \epsilon \quad \forall j \\ & \underline{g} \leq g \leq \overline{g}. \end{aligned} \tag{7.6}$$

where $\epsilon', \epsilon \in [0, 1]$. In this formulation, each flow constraint can be violated, with a probability no

more than ϵ.

Since the regulation period lasts for only a short period of time (10-15 minutes), the variation in the loads would not be significant and it is reasonable to assume it could be approximately characterized by a normal distribution. The next result, whose proof is in the Appendix, shows that the chance-constrained optimization (7.6) can be solved via a deterministic linear program if the loads are normally distributed.

Lemma 7.2.1. ***(Capacity bounds for variable loads via deterministic optimization).*** *Assume the loads are distributed normally with mean $\hat{l}$ and variance V_l. Then, the solution of the deterministic linear program*

$$\begin{aligned} \min_{g,\gamma,t} \quad & t \\ \text{s.t.} \quad & \mathbf{1}^\top \hat{l} - \mathbf{1}^\top g - t \leq \sqrt{2}\,\mathrm{erf}^{-1}(2\epsilon' - 1)(\mathbf{1}^\top V_l \mathbf{1})^{1/2} \\ & |(\mathsf{M}_1 \mathbf{1}^\top - \mathsf{M}_3)\hat{l} + (\mathsf{M}_2 - \mathsf{M}_1 \mathbf{1}^\top)g + \mathsf{N}\gamma| \leq \overline{\omega}^l \\ & \underline{g} \leq g \leq \overline{g}, \end{aligned} \tag{7.7}$$

where $\mathsf{M}^+ = [\mathsf{M}_1 \quad \mathsf{M}_2 \quad \mathsf{M}_3]$ *with* $\mathsf{M}_1 \in \mathbb{R}^m$, $\mathsf{M}_2 \in \mathbb{R}^{m \times n_g}$ *and* $\mathsf{M}_3 \in \mathbb{R}^{m \times n_l}$, $\overline{\omega}^l = \overline{\omega} + K$ *and* $K_j = \sqrt{2}\,\mathrm{erf}^{-1}(\epsilon - 1)\left((\mathsf{M}_{1_j} \mathbf{1}^\top - \mathsf{M}_{3_j}) V_l (\mathsf{M}_{1_j} \mathbf{1}^\top - \mathsf{M}_{3_j})^\top\right)^{1/2}$ *is a solution of problem* (7.6).

Proof. With the notation of the statement, (7.3) can be written as

$$|\,\mathsf{M}_1(\mathbf{1}^\top l - \mathbf{1}^\top g) + \mathsf{M}_2\, g - \mathsf{M}_3\, l + \mathsf{N}\gamma| \leq \overline{\omega}.$$

Without loss of generality, let us for now consider only the following constraint in (7.6)

$$\Pr(|\zeta_j| - \overline{\omega}_j \leq 0) \geq 1 - \epsilon. \tag{7.8}$$

where $\zeta_j = (\mathsf{M}_{1j}\,\mathbf{1}^\top - \mathsf{M}_{3j})l + (\mathsf{M}_{2j} - \mathsf{M}_{1j}\,\mathbf{1}^\top)g + \mathsf{N}_j\,\gamma$. Let $\xi_j^+ = \{\zeta_j \in \mathbb{R} \mid \zeta_j - \overline{\omega}_j \leq 0\}$ and $\xi_j^- = \{\zeta_j \in \mathbb{R} \mid -\zeta_j - \overline{\omega}_j \leq 0\}$. Then (7.8) is equivalent to

$$\Pr(\xi_j^+ \cap \xi_j^-) \geq 1 - \epsilon, \tag{7.9}$$

We can further rewrite (7.9) as

$$\Pr(\xi_j^+ \cap \xi_j^-)^c \leq \epsilon \Rightarrow \Pr(\xi_j^{+c} \cup \xi_j^{-c}) \leq \epsilon. \tag{7.10}$$

We next break (7.10) down into single chance constraints. Using the fact that ξ_j^{+c} and ξ_j^{-c} are mutually exclusive, $\Pr(\xi_j^{+c} \cup \xi_j^{-c}) = \Pr(\xi_j^{+c}) + \Pr(\xi_j^{-c})$. Therefore, (7.10) is equivalent to

$$\Pr(\xi_j^{+c}) \leq \epsilon/2, \text{ and } \Pr(\xi_j^{-c}) \leq \epsilon/2. \tag{7.11}$$

If $l \sim \mathcal{N}(\hat{l}, V_l)$, then $\zeta_j \sim \mathcal{N}(\hat{\zeta}_j, \sigma_j^2)$ where

$$\hat{\zeta}_j = (\mathsf{M}_{1j}\,\mathbf{1}^\top - \mathsf{M}_{3j})\hat{l} + (\mathsf{M}_{2j} - \mathsf{M}_{1j}\,\mathbf{1}^\top)g + \mathsf{N}_j\,\gamma,$$
$$\sigma_j^2 = (\mathsf{M}_{1j}\,\mathbf{1}^\top - \mathsf{M}_{3j})V_l(\mathsf{M}_{1j}\,\mathbf{1}^\top - \mathsf{M}_{3j})^\top.$$

Defining $w = (\zeta_j - \hat{\zeta}_j)/\sigma_j$, we have $w \sim \mathcal{N}(0, 1)$.

$$\Pr(\xi_j^+) = \Pr\left(w \leq \frac{\overline{\omega}_j - \hat{\zeta}_j}{\sigma_j}\right) = \Phi\left(\overline{\omega}_j - \frac{\hat{\zeta}_j}{\sigma_j}\right). \tag{7.12}$$

Using equations (7.12) and (2.3), we have from (7.11) for $\Pr(\xi_j^+)$

$$\begin{aligned}
&\frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(\frac{\overline{\omega}_j - \hat{\zeta}_j}{\sqrt{2}\sigma_j}\right) \geq 1 - \epsilon/2, \\
&\Rightarrow \operatorname{erf}\left(\frac{\overline{\omega}_j - \hat{\zeta}_j}{\sqrt{2}\sigma_j}\right) \geq 1 - \epsilon, \\
&\Rightarrow \hat{\zeta}_j \leq \sqrt{2}\sigma_j \operatorname{erf}^{-1}(\epsilon - 1) + \overline{\omega}_j.
\end{aligned}$$

A similar inequality could be obtained from (7.11) for $\Pr(\xi_j^-)$. As a result, (7.11) could be rewritten as

$$|\hat{\zeta}_j| \leq \sqrt{2}\sigma_j \operatorname{erf}^{-1}(\epsilon - 1) + \overline{\omega}_j,$$

where we have used the fact that erf^{-1} is an odd function. The righthand side of the above constraint is a constant dependent on ϵ and the left hand side depends on the decision variables g and γ.

The same technique could be applied to the remaining set of constraints, including the first one. If we apply this to all the chance constraints in (7.6), then problem (7.6) could be solved by solving the deterministic linear program (7.7). □

Remark 7.2.2. ***(Beyond normally distributed loads).*** *The assumption of loads being normally distributed helps us to convert the original chance-constrained problem* (7.6) *to an equivalent deterministic problem* (7.7)*. It is reasonable to argue that this assumption might be violated in practice. In those cases, one needs to extend the result in Lemma 7.2.1 to identify a computationally efficient way of solving* (7.6)*. An alternative is to use the results in [NS06] to find an approximate solution*

of (7.6) *via solving*

$$
\begin{aligned}
\min_{g,\gamma,t} \quad & t \\
s.t. \quad & \inf_{s>0}[s\mathbb{E}(\phi_0(s^{-1}(-\mathbf{1}^\top g + \mathbf{1}^\top l - t))) - s\epsilon'] \le 0 \\
& \inf_{s>0}[s\mathbb{E}(\phi_j(s^{-1}(|\,\mathsf{M}^+[\mathbf{1}^\top l - \mathbf{1}^\top g; g; l] + \mathsf{N}\gamma|_j - \overline{\omega}_j))) - s\epsilon] \le 0 \quad \forall j \\
& \underline{g} \le g \le \overline{g},
\end{aligned}
$$

where $\{\phi_{j'}\}_{j'=0}^m : \mathbb{R} \to \mathbb{R}_{\ge 0}$ *are non-decreasing and convex functions satisfying* $\phi(u) > \phi(0) = 1$ *for all* $u > 0$. *Note that this approximation is conservative and yields a sub-optimal solution of* (7.6). *The degree of conservativeness depends on the choice of functions* $\{\phi_{j'}\}_{j'=0}^m$. •

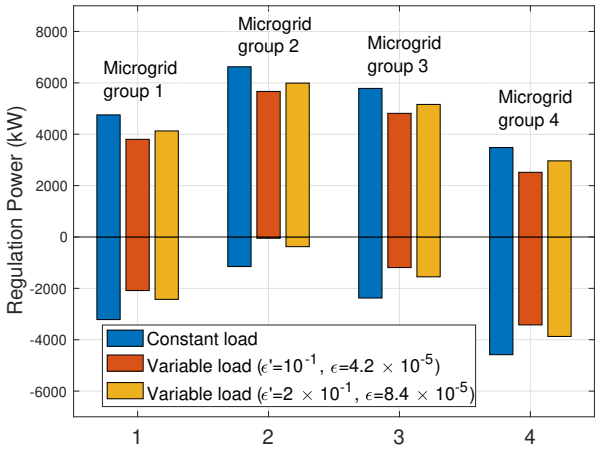


Figure 7.3: Regulation capacities for different instantiations of the reduced-order UCSD microgrid.

We use Lemma 7.2.1 to compute in Figure 7.3 the maximum up and down regulation for several microgrids modeled after the reduced-order UCSD microgrid described later in Section 7.4. The microgrids are divided into 4 groups, each with a different value of baseline generation and mean load for the UCSD model. Within each group, we consider 3 different scenarios, one with constant load and the other two with correlated varying loads (generated using normal distributions

characterized by random variance matrices with entries in the range [0,1000]) and different confident values ($\epsilon', \epsilon = 10^{-1}, 4.2\times10^{-5}$ and $2\times10^{-1}, 8.4\times10^{-5}$, respectively). One can see in Figure 7.3 that the capacity bounds increase with ϵ', ϵ, which is in agreement with the fact that larger values of these correspond to lower probability of satisfying the constraints.

Note that the probabilistic capacity bounds identified above and obtained after solving (7.7) are good only for the bidding in [CP1]. The actual regulation bounds at a given regulation instant still depend on the load at that instant.

7.2.2 Ramp Rate Function

In the following we discuss how to compute the ramp up rate for the microgrid (the discussion for ramp down rate is analogous). If there were no constraints on the power flows, then the ramp rate of the microgrid would be the summation of ramp rates of all the controllable nodes. However, the presence of flow constraints may prevent every controllable node from ramping at its full capacity and as such, the ramp rate is a function that depends on the operating point of the controllable nodes. Let $\mathcal{F}_g = \{g \in \mathbb{R}^{n_g} \mid \exists\ \omega \in \mathbb{R}^m \text{ satisfying (7.2)}\}$ denote the set of feasible operating points for controllable nodes. If the power levels of the uncontrollable nodes are constant, then the ramp up rate, $\mathcal{R} : \mathcal{F}_g \to \mathbb{R}_{\geq 0}$, is formally given by

$$\begin{aligned} \max_{\Delta g, \Delta\omega} \quad & \mathbf{1}^\top \Delta g \\ \text{s.t.} \quad & [P - \mathbf{1}^\top \Delta g; g + \Delta g; -l] = \mathsf{M}(\omega + \Delta\omega) \\ & \Delta g \leq r \\ & |\omega + \Delta\omega| \leq \overline{\omega}, \end{aligned} \tag{7.13}$$

where $r \in \mathbb{R}^{n_g}$ is the vector whose component r_p is the nominal ramping capacity of the controllable node p, and $\omega + \Delta\omega$ is the vector of line flows corresponding to the operating point $g + \Delta g$.

If the power levels of the uncontrollable nodes are variable, we use chance-constraints as in the case of capacity bounds and the ramp up rate $\mathcal{R}$ is given by

$$\begin{aligned} \max_{\Delta g, \gamma} \quad & \mathbf{1}^\top \Delta g \\ \text{s.t.} \quad & \Pr(|\, \mathsf{M}^+[\mathbf{1}^\top l - \mathbf{1}^\top (g + \Delta g); g + \Delta g; -l] + \mathsf{N}\gamma|_j \leq \overline{\omega}_j) \geq 1 - \epsilon \quad \forall j \\ & \Delta g \leq r. \end{aligned} \tag{7.14}$$

The following result, whose proof is similar to that of Lemma 7.2.1 and omitted to avoid repetition, converts the chance-constrained optimization (7.14) into a deterministic linear program if the loads are normally distributed.

Lemma 7.2.3. ***(Ramp rate for variable loads via deterministic optimization).*** *Assume the loads are distributed normally with mean $\hat{l}$ and variance V_l. Then, the solution of the deterministic linear program*

$$\begin{aligned} \max_{\Delta g, \gamma} \quad & \mathbf{1}^\top \Delta g \\ \text{s.t.} \quad & |(\mathsf{M}_1 \mathbf{1}^\top - \mathsf{M}_3)\hat{l} + (\mathsf{M}_2 - \mathsf{M}_1 \mathbf{1}^\top)(g + \Delta g) + \mathsf{N}\gamma| \leq \overline{\omega}^l \\ & \Delta g \leq r, \end{aligned} \tag{7.15}$$

where M_1, M_2, M_3 and $\overline{\omega}^l$ are as defined in Lemma 7.2.1, is a solution for problem (7.14).

The next results describes of the feasible region in terms of the power levels of the controllable nodes for a tree network.

Lemma 7.2.4. ***(Simplified power flow constraints for tree network).*** *Let $\mathcal{G}_m$ be a tree and $\mathsf{P}_{\text{ref}} \in$*

$\mathbb{R}^{(n-1)\times(n-1)}$ *denote its path matrix with first vertex as reference* ref. *Then the constraints*

$$[P - \mathbf{1}^\top \Delta g; g + \Delta g; -l] = \mathsf{M}(\omega + \Delta\omega), \tag{7.16a}$$

$$|\omega + \Delta\omega| \leq \overline{\omega}, \tag{7.16b}$$

in (7.13) *could be equivalently written as*

$$\mathsf{P}_1^\top \Delta g \leq \overline{\omega} + \mathsf{P}_2^\top l - \mathsf{P}_1^\top g, \tag{7.17}$$

where $[\mathsf{P}_1^\top \quad \mathsf{P}_2^\top] = |\,\mathsf{P}_{\text{ref}}^\top\,|$, *with* $\mathsf{P}_1 \in \mathbb{R}^{n_g\times(n-1)}$ *and* $\mathsf{P}_2 \in \mathbb{R}^{n_l\times(n-1)}$, *and* $|\,\mathsf{P}_{\text{ref}}^\top\,|$ *denotes the non-negative matrix whose elements are given by the absolute values of the corresponding elements of* $\mathsf{P}_{\text{ref}}^\top$.

Proof. Let $\mathsf{M}_{\text{ref}} \in \mathbb{R}^{(n-1)\times(n-1)}$ denote the matrix obtained after removing the row corresponding to vertex ref from M. According to [Res63], we have

$$\mathsf{M}_{\text{ref}}^{-1} = \mathsf{P}_{\text{ref}}^\top. \tag{7.18}$$

With first vertex as ref, equation (7.2a) could be rewritten as

$$\begin{bmatrix} g + \Delta g \\ -l \end{bmatrix} = \mathsf{M}_{\text{ref}}(\omega + \Delta\omega), \tag{7.19}$$

where we have used the fact that $\text{rank}(\mathsf{M}) = \text{rank}(\mathsf{M}_{\text{ref}}) = n - 1$, cf. [SR61, Corollary 4-4]. Us-

ing (7.19) and (7.18), constraint (7.16) is equivalent to

$$-\overline{\omega} \leq \mathsf{P}_{\text{ref}}^{\top} \begin{bmatrix} g + \Delta g \\ -l \end{bmatrix} \leq \overline{\omega}.$$

Due to the structure of P_{ref}, cf. Section 2.2, all the non-zero entries for any row of $\mathsf{P}_{\text{ref}}^{\top}$ are either 1 or -1. Since we are characterizing the ramp up rate and are only concerned with what happens to the feasible region with the increase in some component(s) of g, the active constraint for the lines for which the non-zero entries are 1 would be

$$\mathsf{P}_{\text{ref}}^{\top} \begin{bmatrix} g + \Delta g \\ -l \end{bmatrix} \leq \overline{\omega}, \tag{7.20a}$$

and for the lines for which the non-zero entries are -1 would be

$$-\mathsf{P}_{\text{ref}}^{\top} \begin{bmatrix} g + \Delta g \\ -l \end{bmatrix} \leq \overline{\omega}. \tag{7.20b}$$

(7.20) is equivalent to (7.17), completing the proof. □

The next result states the properties of the ramp rate function (7.13) for a tree network. For the ramp rate function with normally distributed loads defined in (7.14), one can obtain a similar result following Lemma 7.2.3 (with $\overline{\omega}$ replaced by $\overline{\omega}^{l}$).

Proposition 7.2.5. ***(Ramp rate of tree network).*** *Let $\mathcal{G}_m$ be a tree and H denote the hyperrectangle describing the region of operation of the controllable nodes, where opposite faces correspond to the minimum and maximum possible power level of a controllable node. Then the ramp rate $\mathcal{R}$ is*

piecewise affine on H, i.e., for some $s > 0$, H admits a decomposition

$$H = V_1 \cup V_2 \cup \ldots \cup V_s,$$

where $\{V_\alpha\}_{\alpha=1}^{s}$ are polyhedra, and $\mathcal{R}$ is affine on each V_α.

Proof. Let us start by denoting the region where

$$\mathsf{P}_1^\top r \leq \overline{\omega} + \mathsf{P}_2^\top l - \mathsf{P}_1^\top g,$$

by V_1. Boundaries of V_1 are $(n-1)$ hyperplanes given by

$$\mathsf{P}_1^\top r = \overline{\omega} + \mathsf{P}_2^\top l - \mathsf{P}_1^\top g.$$

Some of these hyperplanes could even be outside H. But in general, all these $(n-1)$ hyperplanes could be the faces of V_1. It is clear that in V_1, none of the flow constraints is active and $\mathcal{R}(g) = \mathbf{1}^\top r$.

Outside V_1, we have

$$\overline{\omega}_j + \mathsf{P}_{2j}^\top l - \mathsf{P}_{1j}^\top g < \mathsf{P}_{1j}^\top r \tag{7.21}$$

for at least one $j \in \{1, \ldots, n-1\}$. First we consider the region where (7.21) holds for only one such j, denoted as j'. Then either

$$\overline{\omega}_{j'} + \mathsf{P}_{2j'}^\top l - \mathsf{P}_{1j'}^\top g > 0, \text{ or } \overline{\omega}'_j + \mathsf{P}_{2j'}^\top l - \mathsf{P}_{1j'}^\top g = 0.$$

In the former case, we are in the polyhedron whose two faces are given by

$$\overline{\omega}_{j'} + \mathsf{P}_{2j'}^{\top} l - \mathsf{P}_{1j'}^{\top} g = \mathsf{P}_{1j'}^{\top} r, \text{ and } \overline{\omega}_{j'} + \mathsf{P}_{2j'}^{\top} l - \mathsf{P}_{1j'}^{\top} g = 0.$$

Let us call one of these polyhedron V_2. In V_2, $\mathcal{R}(g) = \mathbf{1}^{\top}\Delta g$, where Δg satisfies

$$\overline{\omega}_{j'} + \mathsf{P}_{2j'}^{\top} l - \mathsf{P}_{1j'}^{\top} g = \mathsf{P}_{1j'}^{\top} \Delta g.$$

For $\mathbf{1}^{\top}\Delta g$ to be maximum, the controllable nodes for which the corresponding entries are zero in $\mathsf{P}_{1j'}$, we will have $\Delta g_p = r_p$. As some component(s) of g for which the corresponding entry in $\mathsf{P}_{1j'} = 1$ increases, some components of Δg with corresponding entry 1, decrease to balance it. Hence, $\mathcal{R}(g) = \mathbf{1}^{\top} r - \mathsf{P}_{1j'}^{\top} g$. Now considering the latter case when

$$\overline{\omega}'_{j} + \mathsf{P}_{2j'}^{\top} l - \mathsf{P}_{1j'}^{\top} g = 0.$$

On this hyperplane, $\mathcal{R}(g)$ becomes constant again as the controllable nodes for which the corresponding entries are zero in $\mathsf{P}_{1j'}$ have $\Delta g_p = r_p$ and other entries of Δg have to be zero. Hence, $\mathcal{R}(g) = (\mathbf{1} - \mathsf{P}_{1j'})^{\top} r$. Note that different polyhedrons similar to V_2 might exist with different j'.

Now we consider the regions where (7.21) holds for multiple $j \in \{1, \ldots, n-1\}$. Let us denote by V_3 the polyhedron, whose few faces are given by

$$\overline{\omega}_{j} + \mathsf{P}_{2j}^{\top} l - \mathsf{P}_{1j}^{\top} g = \mathsf{P}_{1j}^{\top} r,$$

for all j satisfying (7.21). Inside V_3, $\mathcal{R}(g) = \mathbf{1}^{\top}\Delta g$, where Δg is given by the simultaneous solution

of

$$\mathsf{P}_{1j}^{\top}\,\Delta g \leq \overline{\omega}_j + \mathsf{P}_{2j}^{\top} - \mathsf{P}_{1j}^{\top}\,g,$$

for all the corresponding j and $\mathbf{1}^{\top}\Delta g$ is maximum. At least, one of these inequalities would hold with equality. Similar to V_2, we notice that if we increase some component(s) of g in V_3 with corresponding entry in any of P_{ij} as 1, $\mathcal{R}(g)$ decreases linearly. While increasing some component of g, a point would be reached where

$$\overline{\omega}_j + \mathsf{P}_{2j}^{\top}\,l - \mathsf{P}_{1j}^{\top}\,g = 0, \tag{7.22}$$

for some j and that would be another face of V_3. On this hyperplane, $\Delta g_p = 0$ for the controllable nodes for which the corresponding entry of $\mathsf{P}_{1j} = 1$ in (7.22). Note that $\mathcal{R}(g)$ is still linear as V_3 but with a different slope.

In general, depending on the parameters of the microgrid at hand, there would be several polyhedrons where (7.21) holds for different j. But the characterization of ramping capacity would be similar to V_3 in all these. Since the ramp rate is either affine or constant in all the polyhedra, it is affine. □

Remark 7.2.6. ***(Ramp rate for networks with non-overlapping loops).*** *If the network is not a tree, then the flows corresponding to a power injection vector are not unique. Nevertheless, the ramp rate for networks with non-overlapping loops is a non-increasing function of g, as the feasible region of* (7.13) *can only shrink with increase in some component(s) of g.* •

Given a regulation power x, we note that there may be more than one feasible operating

point for the microgrid that produces it. As a result, the ramp rate as a function of regulation power is not uniquely defined. We address this by defining $R : [\overline{x}, \underline{x}] \to \mathbb{R}_{\geq 0}$, as

$$R(x) = \max_{g^*} \mathcal{R}(g^*),$$

where g^* denotes a minimizer of the cost of producing the regulation x while respecting the power flow and capacity constraints. We take the maximum, since the optimizer g^* might not be unique. If the cost functions for all the controllable nodes are convex, each g^* is a decreasing function with respect to x, which means that at least one component of g^* would decrease as x increases (using the convention that up regulation is negative). Using this fact, we conclude that R as a function of x is non-decreasing, with maximum possible value as $\mathbf{1}^\top r$. Figure 7.4 provides the ramp rate functions of the four groups of microgrids displayed in Figure 7.3 in the constant load case.

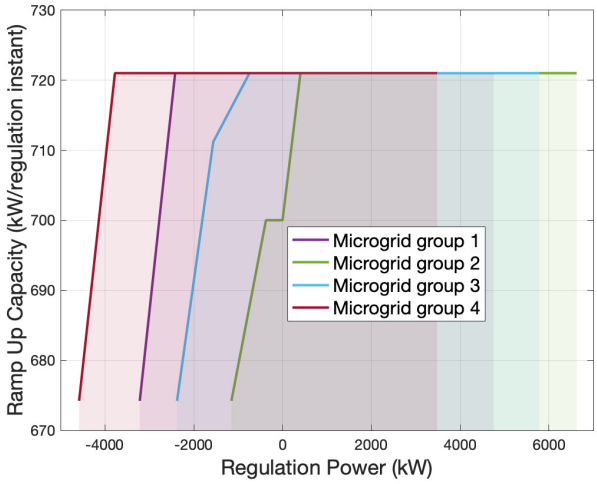


Figure 7.4: Ramp rate functions for different instantiations of the reduced-order UCSD microgrid with constant loads. The shaded regions represent the range of regulation power that the corresponding microgrid can provide.

In Remark 7.2.7, we discuss the conditions under which the minimum ramp rate of the

microgrid is always non-zero.

Remark 7.2.7. ***(Non-zero minimum ramp rate).*** *It is natural to argue that the microgrid could have a zero minimum ramp rate. Here, we discuss conditions under which the minimum ramp rate of the microgrid is non-zero. Let* $\mathcal{E}' = \{e_j \in \mathcal{E} \mid \omega_j < \overline{\omega}_j\}$ *be the set of all the lines which have not reached their flow limits when providing the maximum up regulation. Next, consider the graph* $\mathcal{G}'_m = (\mathcal{V}, \mathcal{E}')$ *and let* $\mathcal{V}'_g = \{v_i \in \mathcal{V}_g \mid \exists$ *a path from* i *to* 1 *in* $\mathcal{G}'_m\}$*, i.e., the set of controllable nodes which are connected to the tie line. If* $\mathcal{V}'_g \neq \phi$*, then the minimum ramp rate is always non-zero. The intuitive explanation is that, when the microgrid is providing the maximum up regulation, the condition specifies that there should be a path from some controllable node to the node connected to the tie line with every line in that path operating away from its flow limits.* •

7.2.3 Cost Function

Each aggregator needs to calculate the cost of providing a given amount of regulation by capturing the effect of operating the controllable nodes away from their baseline operating points. For an operating point g, the total cost for the aggregator is given by

$$h(g) = \sum_{p \in \mathcal{V}_g} h_p(g_p), \tag{7.23}$$

where $h_p : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is the cost of operating node p away from its baseline level g_p^0. One representative example of such a function is $h_p(g_p) = (g_p - g_p^0)^2$. The total regulation that the aggregator provides is the combination of individual regulations of controllable nodes. Therefore, for a specified regulation level x, one would ideally choose the value of g that minimizes the total cost given by (7.23) respecting the power flow constraints in (7.2) and the minimum and maximum capacity

constraints on each controllable node. Formally, $f : [\overline{x}, \underline{x}] : \mathbb{R} \to \mathbb{R}_{\geq 0}$, is given by

$$f(x) = \begin{cases} \min\limits_{g,\omega} & h(g) \\ \text{s.t.} & \underline{g} \leq g \leq \overline{g} \\ & [P^0 + x; g; -l] = \mathsf{M}\,\omega \\ & |\omega| \leq \overline{\omega}. \end{cases} \tag{7.24}$$

However, a cost function defined like this does not take into account the previous operating point of the microgrid and assumes that it can transition between the optimal points corresponding to different regulation powers arbitrarily fast. In practice, however, since the regulation set points change every 2-4 seconds, ramp rates might limit the change from optimal point at one time instant to the next. This suggests that the cost of providing certain amount of regulation at one instant also depends on the value of the regulation power at the previous instant. Hence, we define the cost $\mathbf{f} : [\overline{x}, \underline{x}] \times [\overline{x}, \underline{x}] \to \mathbb{R}_{\geq 0}$, of providing regulation power x, if providing regulation power x^- at the previous instant, as

$$\begin{aligned} \min_{g,\Delta g,\omega,\Delta\omega} \quad & h(g + \Delta g) \\ \text{s.t.} \quad & \underline{g} \leq g + \Delta g \leq \overline{g} \\ & \Delta g \leq r \\ & [P^0 + x; g + \Delta g; -l] = \mathsf{M}(\omega + \Delta\omega) \\ & |\omega + \Delta\omega| \leq \overline{\omega} \\ & \underline{g} \leq g \leq \overline{g}, \quad |\omega| \leq \overline{\omega} \\ & [P^0 + x^-; g; -l] = \mathsf{M}\,\omega. \end{aligned} \tag{7.25}$$

Here, $(g + \Delta g, \omega + \Delta\omega)$ and (g, ω) are the vectors of the power levels of controllable nodes and line flows when the microgrid provides regulation power x and x^-, respectively. The constraints also enforce the capacity limits for the individual controllable nodes and the flow limit constraints for both values of regulation power, and the ramp constraints in transitioning from x^- to x. The reason to include the power flow constraints at x^- in (7.25) is to enable the aggregator to pre-compute the cost function independently of the regulation power it might be asked to provide. Otherwise, if the cost is computed at every regulation instant, g and ω providing x^- would be known, and the optimization variables would only be Δg, and $\Delta\omega$. As such, $\mathbf{f}(x, x^-)$ is a lower bound on the actual cost since (g, ω) are also decision variables and are selected optimally to move to the next operating point.

The following result identifies a condition that simplifies the computation of the cost function $\mathbf{f}(x, x^-)$ defined in (7.25).

Lemma 7.2.8. ***(Simplified formulation and convexity of cost function).*** *Given regulation powers x^- and x, if $|x - x^-| \leq R(x^-)$, then $\mathbf{f}(x, x^-) = f(x)$. If h is (strictly) convex, then f is (strictly) convex.*

Proof. If the difference between two regulation powers, i.e., $|x - x^-|$ is greater than the ramp rate at x^-, then the microgrid might not be able to provide the regulation power at all. On the other hand, if the difference is less than the ramp rate, then it is clear that the microgrid would be able to provide the required regulation power optimally. So, in the latter case, the cost of providing regulation power x or the solution of (7.25) is equivalent to the optimization in (7.24).

Next, we provide a proof for the convexity of f if h is convex. Let $C(x) = C_0 \cap C_1(x)$,

where C_0 denotes the capacity constraints for g and

$$C_1(x) = \{g \mid [P^0 + x; g; -l] = \mathsf{M}\,\omega \text{ and } |\omega| \leq \overline{\omega}\}.$$

Then, we have $f(x) = \min_{g \in C(x)} h(g)$. Let $x_1, x_2 \in [\overline{x}, \underline{x}]$, where $\overline{x}$ and $\underline{x}$ are respectively, the maximum up and down regulation identified in Section 7.2.1. Then $f(x_1) = \min_{g \in C(x_1)} h(g)$, which means that for all $\delta > 0$, there exists $g_1 \in C(x_1)$ such that $f(x_1) + \delta \geq h(g_1)$. Similarly, there exists $g_2 \in C(x_2)$ such that $f(x_2) + \delta \geq h(g_2)$. Since $g_1 \in C(x_1)$ and $g_2 \in C(x_2)$, therefore $\lambda g_1 + (1 - \lambda) g_2 \in C(\lambda x_1 + (1 - \lambda) x_2)$, where $\lambda \in [0, 1]$. Hence,

$$\begin{aligned} f(\lambda x_1 + (1 - \lambda) x_2) &= \min_{g \in C(\lambda x_1 + (1-\lambda) x_2)} h(g) \\ &\leq h(\lambda g_1 + (1 - \lambda) g_2) \leq \lambda h(g_1) + (1 - \lambda) h(g_2) \\ &\leq \lambda f(x_1) + (1 - \lambda) f(x_2) + \delta, \end{aligned}$$

where the second inequality would be strict in case of strict convexity. Since δ is arbitrary, f is (strictly) convex. □

Figure 7.5 provides the cost functions (7.24) of the four groups of microgrids displayed in Figure 7.3 in the constant load case.

Note that the cost function (7.24) assumes the load to be constant, but since the aggregator is not required to submit its cost functions in [CP1], there is no need to pre-compute this using probabilistic techniques. Instead, the cost function at a given regulation instant could be computed online using the load at that instant. The time taken to compute the cost function at a given instant would depend upon the type of solver used, but is usually small (e.g., less than a second with built

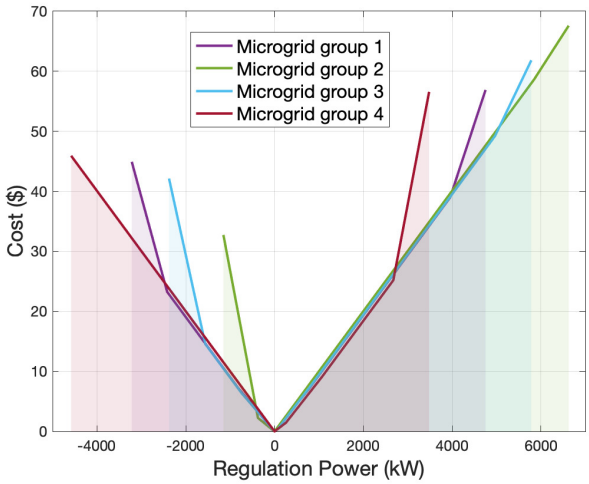


Figure 7.5: Abstracted cost functions for different instantiations of the reduced-order UCSD microgrid with constant loads. The shaded regions represent the range of regulation power that the corresponding microgrid can provide.

in MATLAB solver `fmincon`). In addition, since the regulation period lasts for 10-15 minutes, the variation in load would be limited, thereby requiring the recomputation of the cost function sparingly.

7.2.4 Bids for Participation in Market Clearance

Based on the abstractions in Sections 7.2.1-7.2.3, here we specify the bid information used by each aggregator to participate in [CP1]. Without loss of generality, we specify the bid quantities for up regulation market. Let $g^{\text{up}} \in \mathbb{R}^{n_g}$ denote the component in g of the solution of (7.4).

Table 7.1: Bidding quantities for up regulation market

Bid Quantity	Value
Capacity	$\lvert\overline{x} = P^* - P^0\rvert$
Mileage	$k\mathcal{R}(g^{\text{up}})$
Capacity price	$h(g^{\text{up}})/\lvert\overline{x}\rvert$

Table 7.1 specifies the proposed values for the bidding quantities. Here $k > 0$ is a constant depending on the duration of the regulation period and update frequency of the AGC setpoints. The

suggested bids are conservative, meaning that the aggregator would be able to provide whatever it promises, and there is no strategy to maximize profit. It might seem from Table 7.1 that there is no need to compute beforehand the whole ramp rate function $\mathcal{R}$ in Section 7.2.2. However, a risk taking aggregator might use a higher value of mileage bid based on the shape of $\mathcal{R}$. It is also interesting to note that, from the convexity of cost function in Lemma 7.2.8 and the capacity price bid in Table 7.1, the aggregator would never be at loss regardless of the regulation power being provided.

7.3 RTO-DERP Coordination Problem

Here we describe our algorithmic solution for the RTO-DERP coordination problem [P2] to disaggregate the regulation signal. Equipped with the microgrids' capacities and cost and ramp rate functions identified in Section 7.2, the aggregators, communicating over a graph $\mathcal{G}$, seek to solve, at each instant of the regulation period, the optimization problem (7.1). However, as we have noted before, this problem might not always be feasible due to the presence of ramp constraints. This means that in principle, at each regulation instant, one would need to solve (7.1) if it is feasible or minimize the difference between the required regulation and the procured regulation if it is infeasible. Such dichotomy also raises the issue of the necessary information available to the aggregators to determine which one of the two cases to address at each regulation instant and as such, distributed algorithms designed for solving economic dispatch problem that assume feasibility, see e.g., [CY18, CC16b, KH12] and references therein, are not directly applicable.

Instead, we propose to reformulate the optimization problem in a way that lends itself to the identification of solutions that minimize the error between the procured regulation and the required

regulation whenever (7.1) is not feasible. Without loss of generality, throughout this section we assume the required regulation power to be positive. We start by defining the problem

$$\begin{aligned} \min_{x} \quad & f^{\mu}(x) = f(x) + \mu[\Delta x]^{+} \\ \text{s.t.} \quad & \underline{x_{\mathrm{i}}} \leq x_{\mathrm{i}} \leq \overline{x_{\mathrm{i}}} \quad \forall\, \mathrm{i} \\ & |x_{\mathrm{i}} - x_{\mathrm{i}}^{-}| \leq R_{\mathrm{i}}(x_{\mathrm{i}}^{-}) \quad \forall\, \mathrm{i}, \end{aligned} \tag{7.26}$$

where $\mu > 0$ is a penalty parameter and $\Delta x = x_r - \mathbf{1}^{\top} x$. The following result, whose proof is given in the Appendix, characterizes the equivalence between problems (7.26) and (7.1).

Lemma 7.3.1. ***(Equivalence between*** (7.1) ***and*** (7.26)***).*** *Optimization* (7.26) *is always feasible and there exists $\hat{\mu} < \infty$ such that for all $\mu \in [\hat{\mu}, \infty)$,* (7.1) *and* (7.26) *have the same solution set if* (7.1) *is feasible.*

Proof. We begin by noting that $x_{\mathrm{i}} = x_{\mathrm{i}}^{-}$ for each i satisfies both set of constraints in (7.26), since x^{-} is the set of regulations provided by the aggregators at the previous instant. Hence, (7.26) is always feasible. To prove the equivalence between the two problems, as our first step, we rewrite (7.1) as

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & x_r \leq \mathbf{1}^{\top} x, \\ & \underline{x_{\mathrm{i}}} \leq x_{\mathrm{i}} \leq \overline{x_{\mathrm{i}}} \quad \forall\, \mathrm{i}, \\ & |x_{\mathrm{i}} - x_{\mathrm{i}}^{-}| \leq R_{\mathrm{i}}(x_{\mathrm{i}}^{-}) \quad \forall\, \mathrm{i}\,. \end{aligned} \tag{7.27}$$

Note that the equality constraint in (7.1) is replaced by the inequality constraint in (7.27). If feasible, both problems have the same set of solutions. Problem (7.27) can still be infeasible. Let $\mathcal{F}$

denote its feasible set. Since $\mathcal{F}$ is compact, the solution set of (7.27) is also compact. Also, since the constraints in (7.27) are affine, the refined Slater condition is satisfied. According to [Ber75, Proposition 1], if (7.27) is convex, has a non-empty and compact solution set and satisfies the refined Slater condition, then (7.27) and (7.26) have exactly the same solution set if $\mu > \|\lambda\|_\infty$, for some Lagrange multiplier λ of (7.27), as claimed. □

Remark 7.3.2. ***(Establishing the threshold value $\hat{\mu}$ without the knowledge of dual optimizers).*** *The threshold value $\hat{\mu}$ in Lemma 7.3.1 depends on the optimal values of the dual variables, which is not known beforehand. Interestingly, the explicit knowledge of the Lagrange multipliers to obtain a lower bound on the value of μ can be avoided. In fact, according to [CC15, Proposition 5.2], we have*

$$\hat{\mu} \geq 2 \max_{x \in \mathcal{F}} \|\nabla f(x)\|_\infty. \qquad \bullet$$

Given Lemma 7.3.1, we focus on solving problem (7.26) in a distributed way. To handle the local constraints, we again reformulate (7.26) using exact penalty function as

$$\min_x \; f^p(x) = \underbrace{f(x) + \mu_2 \sum_{i=1}^{N} \left([\overline{b_i}]^+ + [\underline{b_i}]^+ \right)}_{f^{\mu_2}(x)} + \mu [\Delta x]^+, \tag{7.28}$$

$$\text{where } \overline{b_i} = x_i - \min\{\overline{x_i}, x_i^- + R_i(x_i^-)\},$$

$$\text{and } \underline{b_i} = \max\{\underline{x_i}, x_i^- - R_i(x_i^-)\} - x_i,$$

are the box constraints taking care of the capacity and ramp rate for aggregator $i \in \{1, \dots, N\}$ and $\mu_2 > 0$ is again a penalty parameter. Once again, similar to Lemma 7.3.1, there exist finite values

of μ_2 for which the reformulation (7.28) is exact.

Since problem (7.28) is unconstrained, consider the dynamics

$$\dot{x} \in -\partial f^p(x), \tag{7.29}$$

where $\partial f^p : \mathbb{R}^N \rightrightarrows \mathbb{R}^N$ denotes the generalized gradient of f^p. For each agent $\mathrm{i} \in \{1, \dots, N\}$, we have

$$[\partial f^p(x)]_{\mathrm{i}} = \begin{cases} \nabla f_{\mathrm{i}}(x_{\mathrm{i}}) - [\mu]^+_{\Delta x} - [\mu_2]^+_{\overline{b}_i} + [\mu_2]^+_{\underline{b}_i}, & \Delta x, \overline{b}_{\mathrm{i}}, \underline{b}_{\mathrm{i}} \neq 0, \\ \nabla f_{\mathrm{i}}(x_{\mathrm{i}}) - [0, \mu] - [\mu_2]^+_{\overline{b}_i} + [\mu_2]^+_{\underline{b}_i}, & \Delta x = 0, \overline{b}_{\mathrm{i}}, \underline{b}_{\mathrm{i}} \neq 0, \\ \nabla f_{\mathrm{i}}(x_{\mathrm{i}}) - [\mu]^+_{\Delta x} - [0, \mu_2] + [\mu_2]^+_{\underline{b}_i}, & \Delta x, \underline{b}_{\mathrm{i}} \neq 0, \overline{b}_{\mathrm{i}} = 0, \\ \nabla f_{\mathrm{i}}(x_{\mathrm{i}}) - [\mu]^+_{\Delta x} - [\mu_2]^+_{\overline{b}_{\mathrm{i}}} + [0, \mu_2], & \Delta x, \overline{b}_{\mathrm{i}} \neq 0, \underline{b}_{\mathrm{i}} = 0, \\ \nabla f_{\mathrm{i}}(x_{\mathrm{i}}) - [0, \mu] - [0, \mu_2] + [\mu_2]^+_{\underline{b}_i}, & \Delta x, \overline{b}_{\mathrm{i}} = 0, \underline{b}_{\mathrm{i}} \neq 0, \\ \nabla f_{\mathrm{i}}(x_{\mathrm{i}}) - [0, \mu] - [\mu_2]^+_{\overline{b}_i} + [0, \mu_2], & \Delta x, \underline{b}_{\mathrm{i}} = 0, \overline{b}_{\mathrm{i}} \neq 0. \end{cases}$$

The equilibria of the dynamics (7.29) satisfy $\mathbf{0} \in \partial f^p(x)$. Asymptotic convergence of (7.29) to the optimizers of (7.28) could be easily established using tools from non-smooth analysis, cf. [Cor08, Proposition 14]. However, the implementation of (7.29) requires every aggregator to have knowledge of the total regulation at all times. To handle this, we use dynamic average consensus, cf. Chapter 2, to estimate the average of the difference between the required regulation and procured regulation from all the microgrids. Since $\frac{1}{N}\Delta x$ and Δx have the same signs, we modify (7.29) and introduce a new algorithm by enabling each aggregator to estimate the average mismatch using

dynamic average consensus as follows

$$\dot{x} \in -\partial f^{\mu_2}(x) + [\mu]_z^+, \tag{7.30a}$$

$$\dot{z} \in -\nu z - \beta \mathsf{L}\, z - \upsilon + \nu(x_r e - x) + \partial f^{\mu_2}(x) - [\mu]_z^+, \tag{7.30b}$$

$$\dot{\upsilon} = \nu\beta \mathsf{L}\, z, \tag{7.30c}$$

where $z, \upsilon \in \mathbb{R}^N$, z_{i} is the ith aggregator's estimate of $\frac{1}{N}\Delta x$, $[\mu]_z^+ \in \mathbb{R}^N$ with its ith element as $[\mu]_{z_{\mathrm{i}}}^+$, $\mathsf{L} \in \mathbb{R}^{N\times N}$ is the Laplacian matrix of $\mathcal{G}$, and e is the unit vector with only one entry as one and all others as zero. Note immediately that the algorithm (7.30) is distributed over the communication graph, meaning that each aggregator $\mathrm{i} \in \{1, \dots, N\}$ needs to know just its state and the state of its neighbors to implement it, and only one aggregator needs to know the required regulation. We refer to (7.30) as "gradient descent + dynamic average consensus" algorithm, abbreviated as ψ_{gdac}. The equilibria for x are the points satisfying $\mathbf{0} \in -\partial f^{\mu_2}(x) + [\mathbf{0}, \mu\mathbf{1}]$. The next result, whose proof is given in the Appendix, characterizes the convergence properties of the ψ_{gdac} algorithm.

Theorem 7.3.3. ***(Asymptotic convergence of the distributed dynamics to the optimizers).*** *Let $\mathcal{G}$ be strongly connected and weight-balanced, and the initial conditions satisfy $\mathbf{1}^\top \upsilon(0) = 0$ and $\mathbf{1}^\top z(0) - \Delta x(0) = 0$, then there exists $\bar{\mu} < \infty$ such that the dynamics ψ_{gdac} find the optimizers of* (7.28) *for all $\mu \in [\bar{\mu}, \infty)$.*

Proof. For simplicity of exposition, we ignore the box constraints and write (7.30) as

$$\dot{x} = -\nabla f(x) + [\mu]_z^+, \tag{7.31a}$$

$$\dot{z} = -\nu z - \beta \llcorner z - \upsilon + \nu(x_r e - x) + \nabla f(x) - [\mu]_z^+, \tag{7.31b}$$

$$\dot{\upsilon} = \nu\beta \llcorner z, \tag{7.31c}$$

First, consider the function $V_2 : \mathbb{R}^{2N} \to \mathbb{R}_{\geq 0}$, $V_2(x, z) = \mathbf{1}^\top z - \Delta x$. The Lie derivative $\mathcal{L}_{\psi_{\text{gdac}}} V_2 : \mathbb{R}^{2N} \rightrightarrows \mathbb{R}$ is then given by

$$\mathcal{L}_{\psi_{\text{gdac}}} V_2 = \mathbf{1}^\top \dot{z} + \mathbf{1}^\top \dot{x} = -\nu \mathbf{1}^\top (z - (x_r e - x)) = -\nu V_2,$$

where we have used the fact that $\mathbf{1}^\top \upsilon = 0$ due to the initial condition $\mathbf{1}^\top \upsilon(0) = 0$ and dynamics (7.31c). The above equation implies that the summation of all the entries of z converges to the mismatch between the required regulation and procured regulation exponentially with rate ν. Hence $\mathbf{1}^\top z - \Delta x \equiv 0$ with the stated initialization.

Next consider the change of coordinates $(x, z, \upsilon) \mapsto (x, z, \eta)$, with $\eta = \nu(z - (x_r e - x)) + \upsilon$. The dynamics for z and η are then given by

$$\dot{z} = -\beta \llcorner z - \eta + \nabla f(x) - [\mu]_z^+,$$

$$\dot{\eta} = -\nu\eta.$$

Consider the Lyapunov function candidate $V : \mathbb{R}^{3N} \to \mathbb{R}_{\geq 0}$,

$$V(x, z, \eta) = f^{\mu}(x) + \mu \sum_{i=1}^{N} [z_i]^+ + \frac{1}{2}\|\eta\|^2,$$

whose generalized gradient $\partial V : \mathbb{R}^{3N} \rightrightarrows \mathbb{R}^{3N}$ is given by

$$\partial V(x, z, \eta) = \begin{cases} \{\nabla f(x) - [\mu \mathbf{1}]^+_{\Delta x}, [\mu]^+_z, \eta\}, & \Delta x \neq 0, z \neq \mathbf{0}, \\ \{\nabla f(x) - [\mathbf{0}, \mu \mathbf{1}], [\mu]^+_z, \eta\}, & \Delta x = 0, z \neq \mathbf{0}, \\ \{\nabla f(x) - [\mu \mathbf{1}]^+_{\Delta x}, [\mathbf{0}, \mu \mathbf{1}], \eta\}, & \Delta x \neq 0, z = \mathbf{0}, \\ \{\nabla f(x) - [\mathbf{0}, \mu \mathbf{1}], [\mathbf{0}, \mu \mathbf{1}], \eta\}, & \Delta x = 0, z = \mathbf{0}. \end{cases}$$

Following [Cor08], set-valued Lie derivative $\mathcal{L}_{\psi_{\text{gdac}}} V : \mathbb{R}^{3N} \rightrightarrows \mathbb{R}$ can then be computed as

$$\mathcal{L}_{\psi_{\text{gdac}}} V(x, z, \eta) = \begin{cases} (\nabla f - [\mu \mathbf{1}]^+_{\Delta x})^\top (-\nabla f + [\mu]^+_z) + ([\mu]^+_z)^\top (-\beta \llcorner z - \eta + \nabla f - [\mu]^+_z) \\ -\nu \|\eta\|^2, \qquad \Delta x \neq 0, z \neq \mathbf{0}, \\ \phi, \qquad \text{otherwise.} \end{cases}$$

We now analyze various cases of $\Delta x \neq 0, z \neq \mathbf{0}$ in the following

Case 1: $\Delta x < 0$ and $z < \mathbf{0}$.

$$\mathcal{L}_{\psi_{\text{gdac}}} V = -\|\nabla f\|^2 - \nu \|\eta\|^2.$$

Case 2: $\Delta x > 0$ and $z > \mathbf{0}$.

$$\mathcal{L}_{\psi_{\text{gdac}}} V = -\|\nabla f\|^2 + 3\mu \nabla f^\top \mathbf{1} - 2N\mu^2 - \mu\eta^\top \mathbf{1} - \nu\|\eta\|^2.$$

Case 3: $\Delta x > 0$ and $z \not> \mathbf{0}$.

$$\mathcal{L}_{\psi_{\text{gdac}}} V = -\|\nabla f\|^2 - 2N_p\mu^2 + \nabla f^\top(\mu\mathbf{1} + 2[\mu]_z^+) - \beta([\mu]_z^+)^\top \llcorner z - \eta^\top [\mu]_z^+ - \nu\|\eta\|^2,$$

where N_p is the number of positive elements of z.

Case 4: $\Delta x < 0$ and $z \not< \mathbf{0}$.

$$\mathcal{L}_{\psi_{\text{gdac}}} V = -\|\nabla f\|^2 + 2\nabla f^\top [\mu]_z^+ - \beta([\mu]_z^+)^\top \llcorner z - \eta^\top [\mu]_z^+ - N_p\mu^2 - \nu\|\eta\|^2.$$

We do not need to consider the case when $\Delta x > 0$ and $z < \mathbf{0}$ since $\mathbf{1}^\top z - \Delta x \equiv 0$ due to the discussion above. Out of the 4 cases, $\mathcal{L}_{\psi_{\text{gdac}}} V < 0$ for Case 1. For the remaining cases, since f is globally proper and $\|\nabla f\|$ is bounded over any compact set, $\mathcal{L}_{\psi_{\text{gdac}}} V < 0$ if the value of μ is taken large enough for the worst-case scenario ($N_p = 1$). Since $\max \phi = -\infty$, $\max \mathcal{L}_{\psi_{\text{gdac}}} V < 0$ except at the equilibrium. This along with the fact that V is locally Lipschitz and regular implies that V satisfies the hypothesis of [Cor08, Theorem 1]. Hence, the dynamics ψ_{gdac} converge to the optimal solution asymptotically. □

Remark 7.3.4. ***(Initialization of the distributed algorithm).*** *For the dynamics ψ_{gdac} to converge to the optimizers, Theorem 7.3.3 specifies requirements on the initial conditions. The requirement $\mathbf{1}^\top v(0) = 0$ could be implemented trivially by selecting $v(0) = \mathbf{0}$. For the implementation of*

$\mathbf{1}^\top z(0) - \Delta x(0) = 0$, *the aggregators can simply choose* $z(0) = \mathbf{0}$ *and* $x_i(0) = 0$ *for all* i, *except for the aggregator having knowledge of the required regulation for which* $x_i(0) = x_r$. •

7.4 Simulations

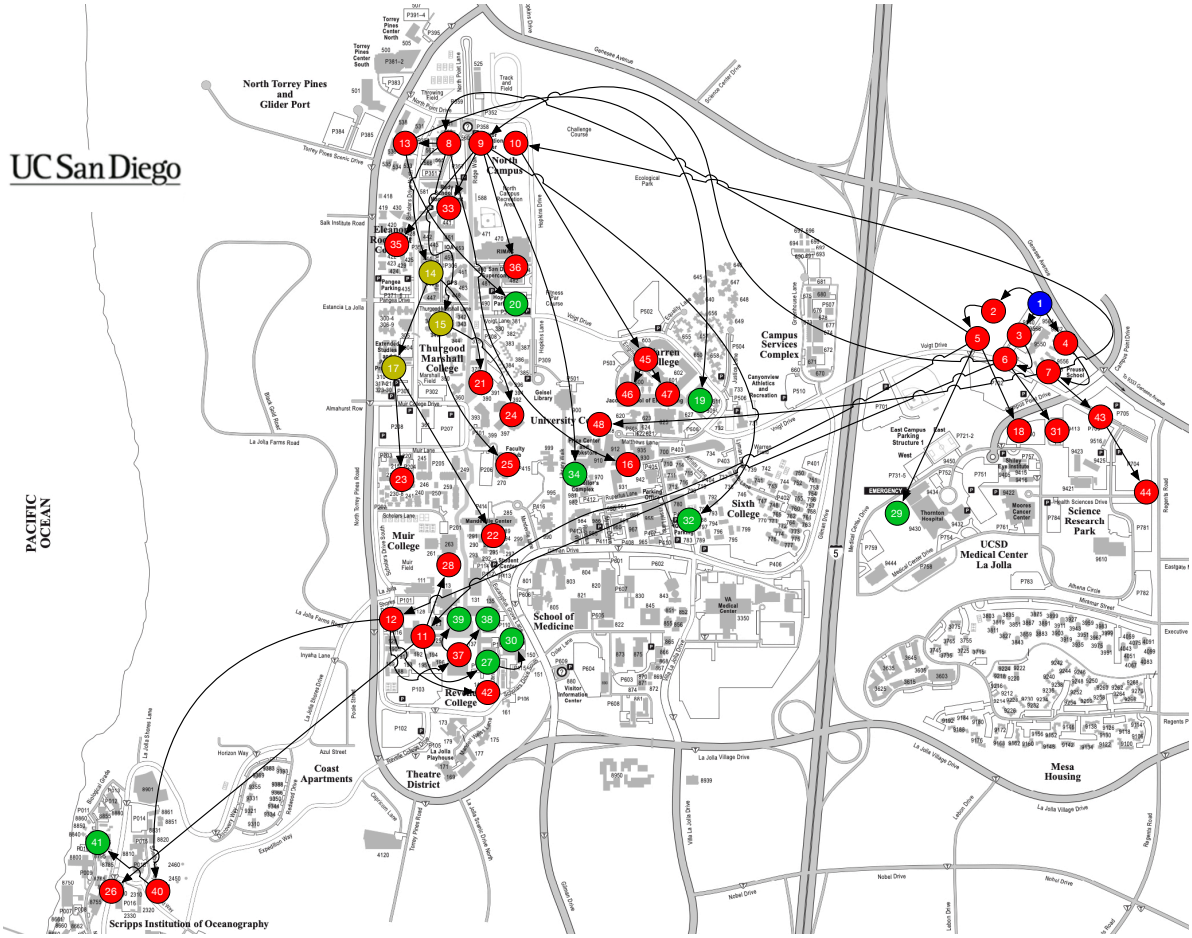


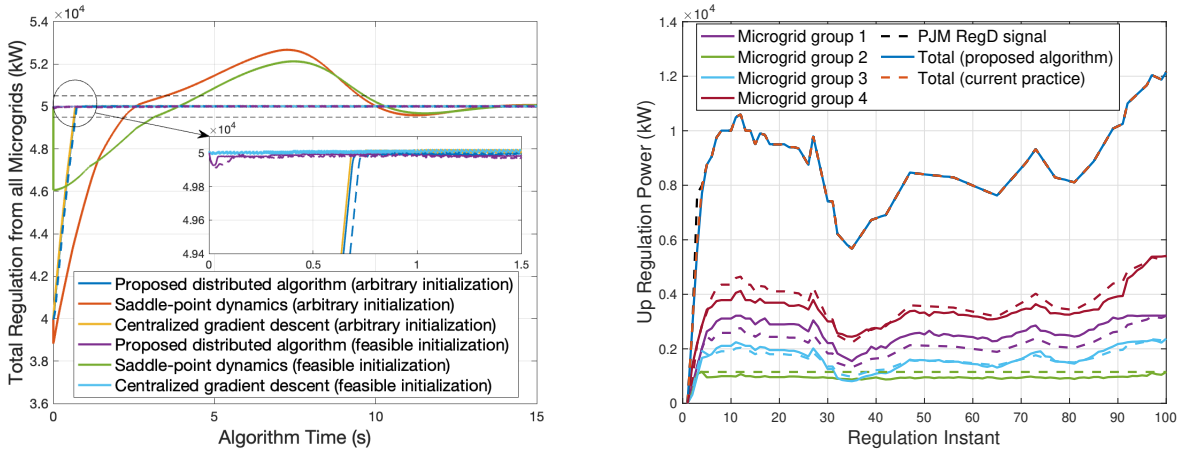
Figure 7.6: Reduced-order model of the UCSD microgrid where blue node is connected to the tie line, green nodes represent the generators, dark yellow the electric vehicle stations and red the building loads.

We provide here our simulation results based on the abstractions of capacities, cost, and ramp rate developed in Section 7.2 and the RTO-DERP coordination algorithm (7.30) in Section 7.3. For the purpose of simulations, we consider a reduced-order model of the University

of California, San Diego (UCSD) microgrid developed using the distributor feeder reduction algorithm in [PDRK18] and provided by the research group of Prof. Jan Kleissl. Compared to the full-order model of the UCSD microgrid [WDW+13] which is a radial, balanced network with 1289 buses (3869 nodes), the reduced-order model is a balanced tree network with 48 buses. The buses in the reduced-order model are obtained by retaining the key buses in the full order model which are the buses where the building loads aggregate or which have generators. Since the UCSD reduced-order model is balanced, we consider only one phase in our simulations. The model consists of 10 generators (2 gas turbines, 1 steam turbine, and 7 solar PV systems) and 37 loads (34 building loads and 3 electric vehicle stations). We show the location of the buses on the geographical map of the campus in Figure 7.6. For our simulation, we take the UCSD microgrid as a template, and we instantiate it using different baseline scenarios to construct 12 different microgrids, divided into 4 groups. Each group has its own baseline values of generation and mean load. The 3 different scenarios within a group consist of (a) constant load, (b) variable load with failure probabilities $\epsilon' = 10^{-1}$, $\epsilon = 4.2 \times 10^{-5}$ and (c) variable load with failure probabilities $\epsilon' = 2 \times 10^{-1}$, $\epsilon = 8.4 \times 10^{-5}$. The abstracted regulation capacities and ramp rate functions of different microgrid groups are shown in Figures 7.3 and 7.4, resp. For cost functions, we consider quadratics for all the resources. The abstracted cost functions for different groups are shown in Figure 7.5.

We demonstrate the performance of the distributed algorithm (7.30) in two sets of simulations. To implement the continuous-time algorithm, we use a first-order Euler discretization with step size of 0.001 to show its practical feasibility. The values of μ, μ_2, β and ν are taken to be 1000, 1100, 400 and 400, respectively. In the first simulation, cf. Figure 7.7(a), we consider one regulation instant and first show the evolution of the proposed algorithm (7.30) for required down regulation of 50000 kW, and compare it, for the same communication topology (undirected ring

with few additional edges), against the (2-hop distributed) saddle-point dynamics [CGC17] of the augmented Lagrangian for the equivalent reformulated problem as per [CC16a] and against the centralized generalized gradient descent dynamics (7.29). As can be seen from the plots, the algorithm time required by the proposed distributed algorithm to reach the 1% band of the required regulation power is much less compared to the saddle-point dynamics, and is only slightly greater than the time taken by the centralized algorithm. The time required does increase when the communication topology is changed to a directed ring –which is the worst possible topology for strongly connected graphs, but still remains less than a second, implying that the number of iterations is less than 1000.



(a) (b)

Figure 7.7: Performance of the proposed RTO-DERP distributed coordination algorithm. (a) shows the state evolution where black dashed lines represent 1% band of the required regulation power. (b) compares the proposed approach with the algorithm followed currently.

For the second simulation, we consider the dynamic regulation test signal (RegD), available on the Pennsylvania-New Jersey-Maryland Interconnection (PJM) website [PJM]. Since the RegD signal on the PJM website is normalized and could be scaled as long as the problem remains feasible, we scale it by a factor of 50000 and then use our abstractions and clear the market according to [CP1]. Once the market is cleared, we use our algorithm to track the scaled RegD signal and

compare it using the current algorithm of disaggregating the regulation signal described in [CP2]. For the sake of clarity, we show only the first 100 instants of the regulation period, and instead of contributions from each of the 12 microgrids, show the total contributions from the 4 groups. As we can see from Figure 7.7(b), when it is not possible to provide the required amount due to limits on ramp rates, both the proposed algorithm and the current algorithm try to provide as much regulation power as possible, and the tracking performance for both the algorithms is similar. But, if we compare the cost, the proposed algorithm with a cost of $8818 outperforms the current algorithm with a cost of $9728. This difference in cost comes from very different power contributions from the microgrids for the two algorithms. The proposed algorithm allocates the regulation signal to the microgrids based on their abstracted cost functions (cf. Figure 7.5), whereas current practice does not take them into account. It can be noticed in Figure 7.7(b) that, under current practice, if not capped by the cleared capacities, the power allocations for different microgrid groups have the same ratios for every regulation instant. For example, the shape of the regulation power curves for microgrid groups 1, 3 and 4 are similar and only differ in terms of scaling (by factors depending on the ratio of their procured mileages).

Acknowledgements

This chapter, in part, is a reprint of the material [SCC21b] conditionally accepted for publication as 'Enabling DER participation in frequency regulation markets' by P. Srivastava, Chin-Yao Chang, and J. Cortés, in the IEEE Transactions on Control Systems Technology, 2021, as well as [SCC18] where it appears as 'Participation of microgrids in frequency regulation markets' by P. Srivastava, Chin-Yao Chang, and J. Cortés in the American Control Conference, 2018. The disser-

tation author was the primary investigator and author of these papers. This work was supported by the ARPA-e NODES program, Cooperative Agreement DE-AR0000695.

Chapter 8

Frequency Regulation via Simultaneously Stabilizing Data-Driven Controller

This chapter presents a framework to design a time invariant controller for the power network with switching inertia. We establish the fact that regardless of the possible operating modes, the simultaneous stabilization problem for the switched system is always feasible. Furthermore, we prove that there always exists a decentralized controller stabilizing the switched system whose implementation requires each node to know just its state.

8.1 Problem Formulation

Consider a power network with $n \in \mathbb{N}$ nodes, described by an undirected graph $\mathcal{G}$. Following [PBD17], we consider a DC approximation of the power system containing only the active

buses and described by the dynamic, for each $i \in \{1, \dots, n\}$,

$$m_i \ddot{\theta}_i + d_i \dot{\theta}_i = u_i - \sum_{\mathcal{N}_i} b_{ij}(\theta_i - \theta_j), \tag{8.1}$$

where u_i is the power input at node i and $b_{ij} \in \mathbb{R}_{\geq 0}$ is the susceptance between lines i and j. If node i is a synchronous generator, then $\theta_i \in \mathbb{R}$ denotes the rotor angle, $m_i \in \mathbb{R}_{>0}$ the rotational inertia of the generator i and $d_i \in \mathbb{R}_{>0}$ the primary speed droop control at node i. In case node i corresponds to a renewable or battery interfaced via a power electronics converter, then θ_i is the voltage phase angle, m_i is the power measurement time constant or the virtual inertia through a controlled device, and d_i is the droop control coefficient. The state-space representation of the swing equation (8.1) is given by

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & I \\ -M^{-1}\mathsf{L} & -M^{-1}D \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ M^{-1} \end{bmatrix} u, \tag{8.2}$$

where the state $x = (\theta; \omega) \in \mathbb{R}^{2n}$ corresponds to the stacked vector of angles and frequencies at each node, $M = \mathrm{diag}(m_i) \in \mathbb{R}^{n \times n}$ is the diagonal matrix with inertia coefficients, $D = \mathrm{diag}(d_i) \in \mathbb{R}^{n \times n}$ is the diagonal matrix with droop control coefficients, L is the Laplacian of $\mathcal{G}$ with edge weights of the adjacency matrix defined as $A_{ij} = b_{ij}$ for all $i, j \in \{1, \dots, n\}$.

The formulation above assumes that the inertia of the system remains constant and makes sense in the traditional paradigm of power system. However, due to the increasing penetration of renewables, the inertia of the network changes over time. Hence, it is reasonable to incorporate the time dependence in the inertia at each node. To do so, we follow the modeling framework introduced in [HGCD+18], and represent the swing equation via a switched-affine system instead

of the invariant system (8.2). Let $m \in \mathbb{N}$ be the number of modes in which the switched system might operate. The power dynamics are then given by

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & I \\ -M_{q(t)}^{-1} \mathsf{L} & -M_{q(t)}^{-1} D \end{bmatrix}}_{A_{q(t)}} \underbrace{\begin{bmatrix} \theta \\ \omega \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} \mathbf{0} \\ M_{q(t)}^{-1} \end{bmatrix}}_{B_{q(t)}} u. \tag{8.3}$$

Here, at time t, the system is in mode $q(t) \in \{1, \dots, m\}$ and $M_q(t)$ denotes the inertia of the network in mode $q(t)$. The inertia on the network at time t depends on the online generators and the connected power electronics converters at that time.

Our goal is to design an optimal controller that brings back the system to its steady-state configuration following any perturbation. Since the pairs $\{A_q, B_q\}_{q=1}^m$ are stabilizable, it is clear that this could be achieved with different controllers for each mode. However, since we might not have the knowledge at all times of the operating mode, our aim is to design a time-invariant controller of the form

$$u = Kx \tag{8.4}$$

stabilizing (8.3) and is optimizing the state deviation and control input required.

For an invariant linear system, this is easily achievable using the solution to the linear quadratic control (LQR) problem. But for the switched system, solving the LQR problem is not obvious. So instead, we employ a data-driven design strategy based on sampling some cases among the switching sequences. We solve a finite-horizon LQR problem for each of the switching sequence sampled and use the resulting trajectories to design an optimal controller mimicking the observed

behavior.

8.2 Data-Driven Controller Design

In this section, we carry out our approach to design a common stabilizing time-invariant controller using training data generated for system (8.3) for a variety of scenarios. We start by describing in Section 8.2.1 how the data is generated via a finite-horizon LQR formulation. Then we provide in Section 8.2.2 a least-square formulation to learn the controller while guaranteeing the stability of each mode $q \in \{1, \dots, m\}$. Since the stability of all the modes is not sufficient to guarantee the stability of the switched system, we generalize in Section 8.2.3 our treatment to the stabilization of the switched system via a common Lyapunov function.

8.2.1 Training Data from Optimal Input Trajectories

In order to generate the training data which would later be used to learn the controller gain K, we solve S scenarios of the finite-horizon LQR problem

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & \int_0^T \left(x(t)^\top Q x(t) + u(t)^\top R u(t) \right) dt \\ \text{s.t.} \quad & \dot{x}(t) = A_{q(t)} x(t) + B_{q(t)} u(t), \end{aligned} \tag{8.5}$$

where $Q \succeq \mathbf{0} \in \mathbb{R}^{2n \times 2n}$, $R \succ \mathbf{0} \in \mathbb{R}^{n \times n}$, $T > 0$ is the time-horizon, and $\mathbf{x}$ and $\mathbf{u}$ are the variables describing the optimal state and input trajectories, respectively. For generating different scenarios, different initial conditions $x(0)$ and switching sequences are used. The training dataset for scenario $k \in \{1, \dots, S\}$ is denoted by $(\mathbf{x}^k, \mathbf{u}^k)$.

8.2.2 Common Controller for Stabilization of All Modes

Here, we are interested in designing a time-invariant controller which guarantees stability for each mode $q \in \{1, \dots, m\}$. Let $\mathcal{H}$ denote the set of Hurwitz matrices. Then the controller design problem described above could be cast as an optimization of the form

$$\begin{aligned} \min_{K} \quad & \sum_{k=1}^{S} \int_{0}^{T} \|\mathbf{u}^k(t) - K\mathbf{x}^k(t)\|_2^2 \, dt \\ \text{s.t.} \quad & A_q + B_q K \in \mathcal{H} \quad \text{for } q \in \{1, \dots, m\}. \end{aligned} \tag{8.6}$$

In general, the simultaneous stabilization problem (8.6) is NP-hard for general system and input matrices, cf. [BT97b]. However, the matrices $\{A_q\}_{q=1}^m$ and $\{B_q\}_{q=1}^m$ in our setup are not totally arbitrary, and indeed have a well defined structure. Specifically, the only quantity that specifies the operating mode $q \in \{1, \dots, m\}$ is the inertia matrix M_q. Building on this insight, we prove next that the simultaneous stabilization problem is always feasible. Our proof is constructive and is based on identifying a controller that stabilizes all the modes.

Proposition 8.2.1. ***(Feasibility of the simultaneous stabilization problem).*** *Problem* (8.6) *is always feasible.*

Proof. Let $K = [K_1 \quad K_2]$, where $K_1, K_2 \in \mathbb{R}^{n \times n}$. Then from equation (8.3), the closed loop system matrix for mode $q \in \{1, \dots, m\}$ is given by

$$A_q + B_q K = \begin{bmatrix} \mathbf{0} & I \\ -M_q^{-1} \mathsf{L} & -M_q^{-1} D \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ M_q^{-1} \end{bmatrix} \begin{bmatrix} K_1 & K_2 \end{bmatrix}.$$

After some algebraic manipulations, we have

$$A_q + B_q K = \begin{bmatrix} \mathbf{0} & I \\ -M_q^{-1}(\mathsf{L} - K_1) & -M_q^{-1}(D - K_2) \end{bmatrix}. \tag{8.7}$$

Let us assume for now that in a given mode q, the inertia of all the nodes is same and is given by $m_q \in \mathbb{R}_{>0}$. Then we have $M_q = m_q I$. Choosing $K_1 = \mathsf{L} - I$ and $K_2 = D - I$, the closed loop system matrix (8.7) becomes

$$A_q + B_q K = \begin{bmatrix} \mathbf{0} & I \\ -1/m_q I & -1/m_q I \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1/m_q & -1/m_q \end{bmatrix}}_{S_q} \otimes I.$$

The eigenvalues of the 2×2 matrix S_q are negative for all $m_q > 0$. Hence, $A_q + B_q K \in \mathcal{H}$ for all q with the selected controller.

Next we relax the assumption of each node $i \in \{1, \ldots, n\}$ having the same inertia. Once again, choose $K_1 = \mathsf{L} - I$ and $K_2 = D - I$. The closed loop system matrix (8.7) now takes the form

$$A_q + B_q K = \begin{bmatrix} \mathbf{0} & I \\ -M_q^{-1} & -M_q^{-1} \end{bmatrix}$$

For each mode $q \in \{1, \ldots, m\}$, define

$$P_q = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & M_q \end{bmatrix},$$

and consider the Lyapunov function candidate $V_q : \mathbb{R}^{2n} \to \mathbb{R}$

$$V_q = x^\top P_q x.$$

The Lie derivative of V_q is given by

$$\mathcal{L}_f V = x^\top \big((A_q + B_q K)^\top P_q + P_q (A_q + B_q K)\big) x = x^\top \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -2I \end{bmatrix} x \leq 0.$$

This means that each mode $q \in \{1, \dots, m\}$ is stable and the result follows from the linear nature of the dynamics in all the modes. □

Note that although establishing the feasibility of the simultaneous stabilization problem (8.6) in case of all the nodes having the same inertia is a special case of all the nodes having a different inertia, it is still interesting to consider it separately as the eigenvalues of the closed loop system could be explicitly characterized in the former case.

Remark 8.2.2. ***(Distributed controller stabilizing all the modes).*** *The proof of Proposition 8.2.1 is constructive and relies on identifying a (not necessarily optimal) controller stabilizing all the modes. It is interesting to note that the identified controller is distributed over $\mathcal{G}$, meaning that to implement it, each node $i \in \{1, \dots, n\}$ needs to know just its angle and frequency, and the angle of the nodes to which it is electrically connected.* •

8.2.3 Common Controller for Stabilization of Switched System

The simultaneous stabilization problem in Section 8.2.2 although guarantees the stability of each mode individually, does not guarantee the stability of the switched system (8.3) in general, cf. [Bra98]. To overcome this, in this section, we are interested in designing a time-invariant controller that stabilizes all the modes with a common Lyapunov function. The controller design problem could then be formulated as

$$\begin{aligned} \min_{K,P} \quad & \sum_{k=1}^{S} \int_0^T \|\mathbf{u}^k(t) - K\mathbf{x}^k(t)\|_2^2 \, dt \\ \text{s.t.} \quad & (A_q + B_q K)^\top P + P(A_q + B_q K) \prec \mathbf{0} \quad \forall q \\ & P \succ \mathbf{0}. \end{aligned} \tag{8.8}$$

In formulation (8.8), we have assumed the Lyapunov function to be quadratic and given by $x^\top P x$. The first constraint in (8.8) ensures that the Lie derivative of the Lyapunov function along the evolution of (8.3) remains negative for each mode $q \in \{1, \dots, m\}$, thereby guaranteeing the stability of the switched system. As before, it is desirable to establish the feasibility of the problem (8.8). We do so in the following result.

Proposition 8.2.3. ***(Feasibility of the switched system simultaneous stabilization).*** *Problem* (8.8) *is always feasible.*

Proof. It is well known, cf. [DP00] that with $X = P^{-1}$ and $K = YX^{-1}$, the constraints in prob-

lem (8.8) could be equivalently written as

$$A_q X + X A_q^\top + B_q Y + Y^\top B_q^\top \prec \mathbf{0} \quad \forall q \tag{8.9a}$$

$$X \succ \mathbf{0}. \tag{8.9b}$$

Let $X = \begin{bmatrix} X_1 & X_2 \\ X_2^\top & X_3 \end{bmatrix}$ and $Y = \begin{bmatrix} Y_1 & Y_2 \end{bmatrix}$, where $X_1, X_2, X_3, Y_1, Y_2 \in \mathbb{R}^{n \times n}$. Then using the structure of $\{A_q\}_{q=1}^m$ and $\{B_q\}_{q=1}^m$, constraint (8.9a) could be rewritten as

$$\begin{bmatrix} \mathbf{0} & I \\ -M_q^{-1} \llcorner & -M_q^{-1} D \end{bmatrix} \begin{bmatrix} X_1 & X_2 \\ X_2^\top & X_3 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ M_q^{-1} \end{bmatrix} \begin{bmatrix} Y_1 & Y_2 \end{bmatrix} + \begin{bmatrix} X_1 & X_2 \\ X_2^\top & X_3 \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\llcorner M_q^{-1} \\ I & -D M_q^{-1} \end{bmatrix} + \begin{bmatrix} Y_1^\top \\ Y_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{0} & M_q^{-1} \end{bmatrix} \prec \mathbf{0} \quad \forall q.$$

After some algebraic manipulations, the above inequality could further be rewritten as

$$\begin{bmatrix} X_2^\top & X_3 \\ Z_q & -W_q + M_q^{-1} Y_2 \end{bmatrix} + \begin{bmatrix} X_2 & Z_q^\top \\ X_3^\top & -W_q^\top + Y_2^\top M_q^{-1} \end{bmatrix} \prec \mathbf{0} \forall q,$$

$$\text{where } Z_q = -M_q^{-1} \llcorner X_1 - M_q^{-1} D X_2^\top + M_q^{-1} Y_1 \tag{8.10a}$$

$$\text{and } W_q = M_q^{-1} \llcorner X_2 + M_q^{-1} D X_3. \tag{8.10b}$$

Hence, (8.9a) is satisfied if the matrix

$$\begin{bmatrix} -X_2 - X_2^\top & -X_3 - Z_q^\top \\ -X_3^\top - Z_q & W_q + W_q^\top - M_q^{-1}Y_2 - Y_2^\top M_q^{-1} \end{bmatrix},$$

is positive definite for all q. Using the Schur complement, cf. [BV04], the positive definiteness condition above and hence, (8.9a) is equivalent to

$$-X_2 - X_2^\top \succ \mathbf{0} \tag{8.11a}$$

$$W_q + W_q^\top - M_q^{-1}Y_2 - Y_2^\top M_q^{-1} \succ \mathbf{0} \quad \forall q, \tag{8.11b}$$

$$W_q + W_q^\top - M_q^{-1}Y_2 - Y_2^\top M_q^{-1} + (X_3^\top + Z_q)(X_2 + X_2^\top)^{-1}(X_3 + Z_q^\top) \succ \mathbf{0} \quad \forall q. \tag{8.11c}$$

Choose $X \succ \mathbf{0}$ satisfying (8.11a). Then since W_q and Z_q are independent of Y_2, therefore, for any given X_1, X_2, X_3, Y_1, there exists $V \prec \mathbf{0}$, independent of Y_2, such that for all $q \in \{1, \dots, m\}$

$$W_q + W_q^\top - V \succ \mathbf{0},$$

$$\text{and } (X_3^\top + Z_q)(X_2 + X_2^\top)^{-1}(X_3 + Z_q^\top) - V \succ \mathbf{0}.$$

Then for a common controller to exist, it suffices that there exist Y_2 such that

$$2V - M_q^{-1}Y_2 - Y_2^\top M_q^{-1} \succ \mathbf{0} \quad \forall q.$$

Let $\overline{M}$ denote the matrix obtained after taking the entry wise maximum of the inertia matrix at all

nodes. Then the above inequality is satisfied if Y_2 is chosen such that

$$Y_2 \prec V\overline{M}. \tag{8.13}$$

This completes the proof. □

Note that Proposition 8.2.3 could be considered as a special case of Proposition 8.2.1. However, both the results differ in their proof methodologies. The proof of Proposition 8.2.1 is constructive and depends on the identification of a controller, which is distributed over $\mathcal{G}$. On the other hand, although Proposition 8.2.3 guarantees the existence of an invariant controller stabilizing the switched system, there is no guarantee for the resulting controller to be distributed. Nevertheless, Proposition 8.2.3 serves as the basis for the design of a distributed controller in the next section.

8.2.4 Common Controller for Stabilization of Switched System via Distributed Control

Here, we present a method to design a sparse controller to stabilize (8.3). We use the standard technique of penalizing the summation of absolute values of all the elements of K. The sparse controller design problem then could be reformulated as an optimization of the form

$$\begin{aligned} \min_{K,P} \quad & \sum_{k=1}^{S} \int_0^T \|\mathbf{u}^k(t) - K\mathbf{x}^k(t)\|_2^2 \, dt + \beta \sum_{i=1}^{n}\sum_{j=1}^{n} |K_{ij}| \\ \text{s.t.} \quad & (A_q + B_q K)^\top P + P(A_q + B_q K) \prec \mathbf{0} \quad \forall q \\ & P \succ \mathbf{0}, \end{aligned} \tag{8.14}$$

where $\beta > 0$ is the design parameter and specifies the importance of sparsity compared to the optimality [DJL14]. Since (8.8) is feasible, problem (8.14) is feasible too. However, in general there is no guarantee for the resulting controller to be distributed over $\mathcal{G}$. In the following result, we prove that a controller, distributed over any desired communication graph does exist. In fact, there exists a completely decentralized local controller that does not need communication even with the neighboring nodes. The proof methodology leverages from the freedom in choosing various parameters in the proof of Proposition 8.2.3.

Corollary 8.2.4. ***(Local controller stabilizing the switched system).*** *There exists a controller of the form $u = D_1\theta + D_2\omega$, where $D_1, D_2 \in \mathbb{R}^{n\times n}$ are diagonal matrices, satisfying the constraints in problem* (8.8)*.*

Proof. Following the proof of Proposition 8.2.3, we are interested in identifying X and Y satisfying (8.9). Let us choose $X_1 = I,\ X_2 = -I$, and $Y_1 = \mathbf{0}$. Then using the Schur complement, (8.9b) is satisfied iff

$$X_3 - I \succ \mathbf{0}. \tag{8.15}$$

Once again, following the same steps as in the proof of Proposition 8.2.3, Y_2 can be chosen such that (8.13) holds, where V is selected such that (8.12) holds. The controller K is then given by

$$K = \begin{bmatrix} \mathbf{0} & Y_2 \end{bmatrix} \begin{bmatrix} I & -I \\ -I & X_3 \end{bmatrix}^{-1}.$$

Using the formula for the inverse of a partitioned matrix [HJ85, Section 0.7.3], we have

$$K = \begin{bmatrix} \mathbf{0} & Y_2 \end{bmatrix} \begin{bmatrix} (I - X_3^{-1})^{-1} & (X_3 - I)^{-1} \\ (X_3 - I)^{-1} & (X_3 - I)^{-1} \end{bmatrix} = \begin{bmatrix} Y_2(X_3 - I)^{-1} & Y_2(X_3 - I)^{-1} \end{bmatrix}.$$

If Y_2 and X_3 are chosen as the diagonal matrices satisfying (8.13) and (8.15) respectively, then the resulting controller is guaranteed to stabilize the switched system and is local. □

8.3 Simulations

In this section, we demonstrate the effectiveness of the proposed approach via numerical experiments. We use the standard 12-bus 3-region network, shown in Figure 8.1, that has also been used in [HGCD+18, Kun94, PBD17]. We take $m = 10$ and assume that at a given time t,

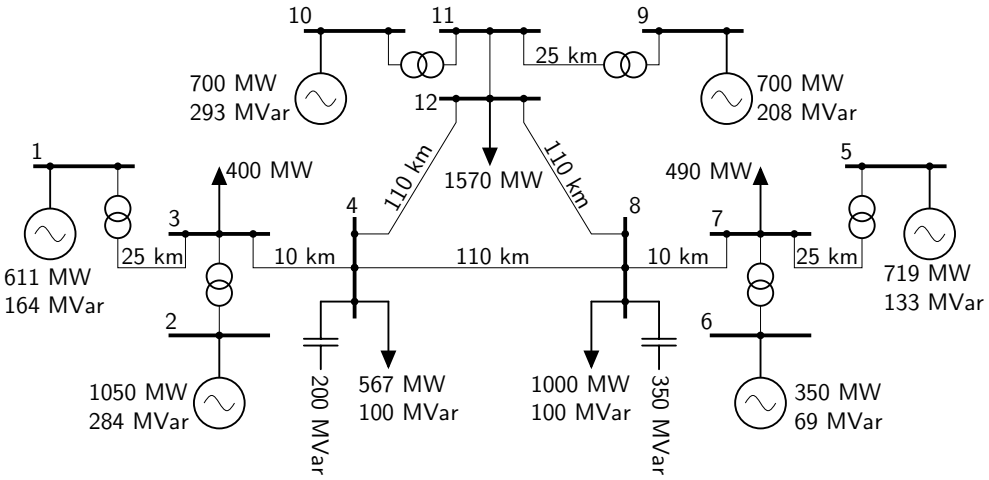


Figure 8.1: 12-bus 3-region network used in simulations.

the rotational inertia for each node $i \in \{1, \dots, n\}$ is same. Hence, each mode $q \in \{1, \dots, m\}$ of the hybrid system is given by one value of inertia. To generate the training data-set, we use

$Q = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & 10^5 I \end{bmatrix}$ and $R = 10I$. To implement (8.5), we use its discrete-time counterpart

$$\begin{aligned} \min_{\mathbf{x},\mathbf{u}} \quad & \sum_{t=0}^{T} x_t^\top Q x_t + u_t^\top R u_t \\ \text{s.t.} \quad & x_{t+1} = A^d_{q(t)} x_t + B^d_{q(t)} u_t, \end{aligned} \tag{8.16}$$

where $A^d_{q(t)}\mathbb{R}^{2n\times 2n}$ and $B^d_{q(t)} \in \mathbb{R}^{2n\times n}$ are respectively, the state and input matrices of the discretized system using a zero-order hold. We use a step-size of 0.01 seconds and simulate 50 scenarios of (8.16), each for 50 time steps, using `cvx` [GB14]. The initial conditions for all the scenarios are different and drawn from a normal distribution with 0 mean and 0.1 variance.

We design three sets of controllers. To design the first (optimal) controller, we solve (8.8) using the BMI algorithm in [TGMD12]. Since the algorithm requires a feasible initialization, we solve the feasibility problem associated with the LMI constraints (8.9) using `cvx`. To design the second controller, which is sparse and distributed over $\mathcal{G}$, we take the first controller as the initial point and solve (8.14) with $\beta = 100$, again using the algorithm in [TGMD12]. The third controller that we provide is local, and based on Corollary 8.2.4. Unfortunately, in our simulations, we observed that simply increasing the penalty parameter β in (8.14) is not enough to ensure that the resulting controller is local, and as such, the design of a local controller which is also optimal for the training data is left for future.

Since the local controller that we design is not optimal for the training data, we compare the dynamical response of just the first two controllers in our simulations. We assume that the system starts in mode 5, and can switch to any other mode every 0.01 seconds. The initial frequency devi-

ation at each node is 0.15 Hz. In Figure 8.2, we plot the frequency deviation at node 1 for different switching sequences. It is interesting to note that even though the optimal controller has an higher

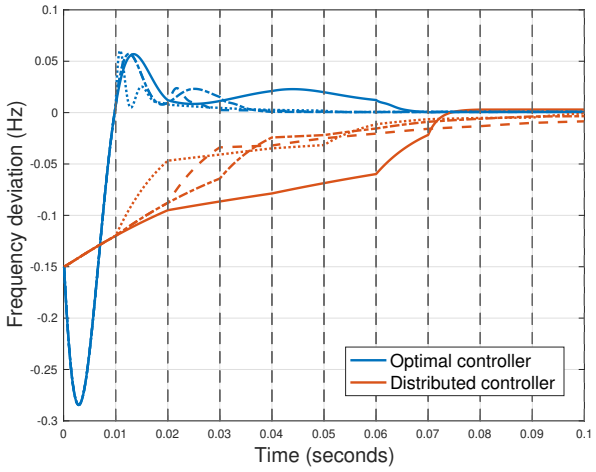


Figure 8.2: Frequency deviation at node 1 for different switching sequences using the optimal and distributed controllers. Dashed vertical lines represent the switching instances and different line styles correspond to different switching sequences.

overshoot for all the switching sequences, convergence is also faster. To further compare the controllers, in Table 8.1, we provide the total absolute value of the control input and the total absolute value of frequency deviation for 3 fixed inertia modes ($q = 1, 5, 10$). As expected, the optimal

Table 8.1: Performance metrics for the designed controllers under different inertia modes

Mode	Controller	$\int_0^1 \sum_{i=1}^{n} \lvert u_i(t)\rvert dt$	$\int_0^1 \sum_{i=1}^{n} \lvert \omega_i(t)\rvert dt$
1	Optimal	13.46	0.005
	Distributed	14.58	0.008
5	Optimal	165.83	0.033
	Distributed	268.96	0.128
10	Optimal	736.16	0.140
	Distributed	773.10	0.418

controller outperforms the distributed one at the expense of increased communication burden.

Acknowledgements

This chapter, in full, is currently being prepared for submission for publication of the material as 'Frequency regulation via simultaneously stabilizing data-driven controller' by P. Srivastava, P. Hidalgo-Gonzalez, and J. Cortés. The dissertation author was the primary investigator and author of this paper.

Chapter 9

Conclusions

In this thesis, we synthesized algorithmic solutions for large-scale constrained optimization problems and analyzed them from a controls point of view. Exploiting the structure of these network problems, we privacy-preserving, scalable, accelerated and anytime algorithms to solve them, and show how constrained optimization problems surface naturally when dealing with frequency regulation in the modern grid. In the following, we summarize the contributions of the thesis and point towards future research directions.

9.1 Summary

We started in Chapter 3 by presenting continuous-time algorithms to solve linear algebraic equations whose problem data is represented as the summation of the data of individual agents. The proposed algorithms do not require the individual agent matrices to be positive definite and are guaranteed to converge to a solution of the linear equation exponentially fast.

In Chapter 4, we have considered the problem of distributed optimization of a separable

function under locally coupled constraints by a group of agents. Our approach relies on the reformulation of the optimization problem via a continuously differentiable exact penalty function. To enable the distributed computation of the gradient of this function, we utilized the algorithms in Chapter 3, to solve linear algebraic equations defined by separable data. Building on this, we have introduced dynamics to asymptotically compute the gradient of the penalty function in a distributed fashion. Our algorithmic solution for optimization consists of implementing gradient descent and Nesterov's accelerated method with the running estimates provided by this dynamics.

Next, we formalized the accelerated convergence using the Nesterov's method for equality-constrained convex optimization in Chapter 5. We have provided sufficient conditions under which we can reformulate the original problem as the unconstrained optimization of a continuously differentiable (strongly) convex penalty function. Via simulations, we have shown that in terms of computation time required to reach the desired accuracy, the proposed method performs the best compared to other state-of-the-art methods.

Chapter 6 took a step towards the actual implementation of continuous-time algorithms in practice and designed decentralized event-triggered coordination mechanisms to solve network optimization problems whose objective function is a combination of a separable component among the agents and a non-separable coupling term. The proposed coordination mechanisms prescribe opportunistic requests for information from the agents to the network supervisor, are anytime, and guarantee asymptotic convergence to the desired optimizer.

Turning our attention to a specific problem, we considered the problem of providing frequency regulation services by aggregations of DERs in Chapter 7. We have described the limitations of current practice and identified the challenges to overcome them with DER aggregators modeled as microgrids. We have developed meaningful abstractions for the capacity, cost of gen-

eration, and ramp rates by taking into account the power flow equations inside the microgrid. We have employed these abstractions to design a provably correct distributed algorithm that solves the RTO-DERP coordination problem to optimally disaggregate the regulation signal when the problem is feasible and minimize the difference between the required regulation and procured regulation when it is infeasible.

Finally, we applied constrained optimization to design stable time-invariant controller for the time-varying power dynamics in Chapter 8. We showed that owing to the structure inherent in the power network dynamics, simultaneous stabilization of the switching power dynamics is always feasible, more so with a completely decentralized controller.

9.2 Future Directions

Based on the results of the thesis, here we outline some immediate short term extensions and also point towards some long term research plans.

9.2.1 Extensions

For linear equations with separable data, it would be interesting to extend our work to finding least-squares solutions when exact ones do not exist, and the communication graph is unbalanced and time-varying. Future work will also involve formally extending our approach of accelerating the gradient dynamics to problems with general convex constraints. We would also like to rigorously characterize the rate of convergence in its distributed implementation, and extend our treatment to problems involving global non-sparse coupling constraints.

Regarding the work on frequency regulation in the thesis, the power flow equations used for

the controller design and developing the abstractions are simplified and arguably simple. Future plan is to develop more realistic abstractions using exact power flow equations. In addition, it would also be interesting to consider dynamic models for the DERs and characterize the temporal dependence of the abstractions.

9.2.2 And Beyond...

The focus of this thesis has mainly been on designing continuous time optimization algorithms. Although one can argue that this way of synthesizing algorithms is more intuitive, from an implementation perspective, it suffers from the need of continuous communication and actuation, which is not realizable in practice. Of course, we can always use Runge-Kutta discretizations with small stepsizes, but this gives rise to periodic implementations at high frequencies which is clearly not efficient. This necessitates the design of asynchronous and opportunistic *resource-aware* coordination mechanisms which limit communication and actuation to only those instances when the system needs attention. Identifying proper trigger points for this event-based coordination requires analysis from a hybrid systems perspective. This becomes particularly interesting for network systems as different subsystems might have very different properties. And as such, translating the performance guarantees for the interconnected network system to locally identifiable criteria for individual subsystems (and vice versa) requires coming up with novel and adaptive equivalence mappings ensuring uniform utilization of all the resources.

Another important point to note is that the contributions of the thesis are based on the assumption that we have perfect knowledge about the problem at hand. Since the network models and the functions involved might be unknown, in practice, this is not always the case. This is es-

pecially true with the rapid advancements in different technologies driving the rise in number of independent modules joining into existing large-scale systems. The plan, therefore, is to develop feedback-based *model-free* solutions that do not require a priori knowledge of the system. Since extending this data-driven framework to general problems might be too ambitious, it is reasonable to start with solutions aimed at specific applications. To some degree, the model-free approaches also handle scenarios where the problem data is not static and varies over time. Considering the importance of model-free approaches together with the discussion in the former paragraph on building practically realizable systems, a natural line of future work is to devise data-driven event-triggered algorithms encompassing the merits of both.

Bibliography

[AAC+17] O. Ajala, M. Almeida, I. Celanovic, P. W. Sauer, and A. D. Domínguez-García. A hierarchy of models for microgrids with grid-feeding inverters. In *IREP Bulk Power System Dynamics and Control Symposium*, Espinho, Portugal, August 2017.

[AHU58] K. Arrow, L Hurwitz, and H. Uzawa. *Studies in Linear and Non-Linear Programming*. Stanford University Press, Stanford, CA, 1958.

[ALDJ15] A. Adaldo, D. Liuzza, D. V. Dimarogonas, and K. H. Johansson. Control of multi-agent systems with event-triggered cloud access. In *European Control Conference*, pages 954–961, Linz, Austria, 2015.

[AMMH16] B. D. O. Anderson, S. Mou, A. S. Morse, and U. Helmke. Decentralized gradient algorithm for solution of a linear equation. *Numerical Algebra, Control and Optimization*, 6(3):319–328, 2016.

[AMS+21] T. Anderson, M. Muralidharan, P. Srivastava, H. Valizadeh Haghi, J. Cortés, J. Kleissl, S. Martínez, and B. Washom. Frequency regulation with heterogeneous energy resources: A realization using distributed control. *IEEE Transactions on Smart Grid*, 12(5):4126–4136, 2021.

[AO17] P. Armand and R. Omheni. A globally and quadratically convergent primal-dual augmented Lagrangian algorithm for equality constrained optimization. *Optimization Methods and Software*, 32(1):1–21, 2017.

[AYS20] S. A. Alghunaim, K. Yuan, and A. H. Sayed. A proximal diffusion strategy for multi-agent optimization with sparse affine constraints. *IEEE Transactions on Automatic Control*, 2020. To appear.

[BAP+13] B. Biegel, P. Andersen, T. S. Pedersen, K. M. Nielsen, J. Stoustrup, and L. H. Hansen. Smart grid dispatch strategy for on/off demand-side devices. In *European Control Conference*, pages 2541–2548, Zürich, Switzerland, 2013.

[BCM09] F. Bullo, J. Cortés, and S. Martinez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009.

[Ber75] D. P. Bertsekas. Necessary and sufficient conditions for a penalty method to be exact. *Mathematical Programming*, 9(1):87–99, 1975.

[Ber82] D. P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, Belmont, MA, 1982.

[Ber99] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 2nd edition, 1999.

[BHH12] J. Bushnell, S. M. Harvey, and B. F. Hobbs. Opinion on pay-for-performance regulation. Technical report, Market Surveillance Committee, California ISO, March 9 2012.

[BKP+18] O. Borne, K. Korte, Y. Perez, M. Petit, and A. Purkus. Barriers to entry in frequency-regulation services markets: Review of the status quo and options for improvements. *Renewable and Sustainable Energy Reviews*, 81:605–614, 2018.

[BNP16] S. L. Bowman, C. Nowzari, and G. J. Pappas. Coordination of multi-agent systems via asynchronous cloud communication. In *IEEE Conf. on Decision and Control*, pages 2215–2220, Las Vegas, NV, 2016.

[BPC+11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[BPP16] O. Borne, M. Petit, and Y. Perez. Provision of frequency-regulation reserves by distributed energy resources: Best practices and barriers to entry. In *International Conference on the European Energy Market (EEM)*, pages 1–7, June 2016.

[Bra98] M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

[Bro91] R. W. Brockett. Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. *Linear Algebra and its Applications*, 146:79–91, 1991.

[BSC15] Q. Ba, K. Savla, and G. Como. Distributed optimal equilibrium selection for traffic flow over networks. In *IEEE Conf. on Decision and Control*, pages 6942–6947, Osaka, Japan, 2015.

[BT97a] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[BT97b] V. Blondel and J. N. Tsitsiklis. NP-hardness of some linear control design problems. *SIAM Journal on Control and Optimization*, 35(6):2118–2127, 1997.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[BV09] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2009.

[Cal12] California ISO. Pay for performance regulation: Draft final proposal, February 2012. Available at `https://www.caiso.com/Documents/Addendum-DraftFinalProposal-Pay_PerformanceRegulation.pdf`.

[CBCZ20] G. Cavraro, A. Bernstein, R. Carli, and S. Zampieri. Distributed minimization of the power generation cost in prosumer-based distribution networks. In *American Control Conference*, pages 2370–2375, Denver, CO, 2020.

[CC15] A. Cherukuri and J. Cortés. Distributed generator coordination for initialization and anytime optimization in economic dispatch. *IEEE Transactions on Control of Network Systems*, 2(3):226–237, 2015.

[CC16a] A. Cherukuri and J. Cortés. Distributed algorithms for convex network optimization under non-sparse equality constraints. In *Allerton Conf. on Communications, Control and Computing*, pages 452–459, Monticello, IL, September 2016.

[CC16b] A. Cherukuri and J. Cortés. Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment. *Automatica*, 74:183–193, 2016.

[CGC17] A. Cherukuri, B. Gharesifard, and J. Cortés. Saddle-point dynamics: conditions for asymptotic stability of saddle points. *SIAM Journal on Control and Optimization*, 55(1):486–511, 2017.

[CL12] J. Chen and V. K. N. Lau. Convergence analysis of saddle point problems in time varying wireless systems - control theoretical approach. *IEEE Transactions on Signal Processing*, 60(1):443–452, 2012.

[Cla83] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Canadian Mathematical Society Series of Monographs and Advanced Texts. Wiley, 1983.

[CM09] S. L. Campbell and C. D. Meyer. *Generalized Inverses of Linear Transformations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2009.

[CMC17] C.-Y. Chang, S. Martinez, and J. Cortés. Grid-connected microgrid participation in frequency-regulation markets via hierarchical coordination. In *IEEE Conf. on Decision and Control*, pages 3501–3506, Melbourne, Australia, December 2017.

[CMK18] S. Camal, A. Michiorri, and G. Kariniotakis. Optimal offer of automatic frequency restoration reserve from a combined PV/wind virtual power plant. *IEEE Transactions on Power Systems*, 33(6):6155–6170, 2018.

[CMLC18] A. Cherukuri, E. Mallada, S. H. Low, and J. Cortés. The role of convexity in saddle-point dynamics: Lyapunov function and robustness. *IEEE Transactions on Automatic Control*, 63(8):2449–2464, 2018.

[CN19] J. Cortés and S. K. Niederländer. Distributed coordination for nonsmooth convex optimization via saddle-point dynamics. *Journal of Nonlinear Science*, 29(4):1247–1272, 2019.

[Cor08] J. Cortés. Discontinuous dynamical systems – a tutorial on solutions, nonsmooth analysis, and stability. *IEEE Control Systems*, 28(3):36–73, 2008.

[CPP] P. Codani, M. Petit, and Y. Perez. Missing money for EVs: Economics impacts of TSO market designs. Available at `https://ssrn.com/abstract=2525290`.

[CY18] G. Chen and Q. Yang. An ADMM-based distributed algorithm for economic dispatch in islanded microgrids. *IEEE Transactions on Industrial Informatics*, 14(9):3892–3903, 2018.

[DG89] G. Di Pillo and L. Grippo. Exact penalty functions in constrained optimization. *SIAM Journal on Control and Optimization*, 27(6):1333–1360, 1989.

[DGS+18] E. Dall'Anese, S. Guggilam, A. Simonetto, Y. C. Chen, and S. V. Dhople. Optimal regulation of virtual power plants. *IEEE Transactions on Power Systems*, 33(2):1868–1881, 2018.

[DH19] S. S. Du and W. Hu. Linear convergence of the primal-dual gradient method for convex-concave saddle point problems without strong convexity. In *The 22nd International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 196–205, Naha, Okinawa, Japan, 2019.

[Di 94] G. Di Pillo. Exact penalty methods. In E. Spedicato, editor, *Algorithms for Continuous Optimization: The State of the Art*, pages 209–253. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[DJ19] D. Ding and M. R. Jovanović. Global exponential stability of primal-dual gradient flow dynamics based on the proximal augmented Lagrangian. In *American Control Conference*, pages 3414–3419, Philadelphia, PA, July 2019.

[DJL14] N. K. Dhingra, M.R. Jovanović, and Z.-Q. Luo. An ADMM algorithm for optimal sensor and actuator selection. In *IEEE Conf. on Decision and Control*, pages 4039–4044, 2014.

[DKJ17] N. K. Dhingra, S. Z. Khong, and M. R. Jovanović. A second order primal-dual method for nonsmooth convex composite optimization. *IEEE Transactions on Automatic Control*, 2017. Submitted. https://arxiv.org/abs/1709.01610.

[DP00] G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory*. Number 36 in Texts in Applied Mathematics. Springer, 2000.

[DSPB16] F. Dörfler, J. W. Simpson-Porco, and F. Bullo. Breaking the hierarchy: Distributed control & economic optimality in microgrids. *IEEE Transactions on Control of Network Systems*, 3(3):241–253, 2016.

[DZG13] E. Dall'Anese, H. Zhu, and G. B. Giannakis. Distributed optimal power flow for smart microgrids. *IEEE Transactions on Smart Grid*, 4(3):1464–1475, 2013.

[EMK11] E. Ela, M. Milligan, and B. Kirby. Operating reserves and variable generation. Technical report, National Renewable Energy Laboratory, Aug 2011.

[FBM+94] T. L. Friesz, D. H. Bernstein, N. J. Mehta, R. L. Tobin, and S. Ganjlizadeh. Day-to-day dynamic network disequilibria and idealized traveler information systems. *Operations Research*, 42(6):1120–1136, 1994.

[Fed11] Order No. 755: Frequency regulation compensation in the organized wholesale power markets, 2011. Available at `http://www.ferc.gov/whats-new/comm-meet/2011/102011/E-28.pdf`.

[Fed20] Order No. 2222: Participation of distributed energy resource aggregations in markets operated by regional transmission organizations and independent system operators, September 2020. Available at `https://www.ferc.gov/sites/default/files/2020-09/E-1_0.pdf`.

[FP10] D. Feijer and F. Paganini. Stability of primal-dual gradient dynamics and applications to network optimization. *Automatica*, 46:1974–1981, 2010.

[FZD18] D. Fooladivanda, M. Zholbaryssov, and A. D. Domínguez-García. Control of networked distributed energy resources in grid-connected AC microgrids. *IEEE Transactions on Control of Network Systems*, 5(4):1875–1886, 2018.

[GAB18] R. Ghaemi, M. Abbaszadeh, and P. Bonanni. Scalable optimal flexibility control of distributed loads in the power grid. In *American Control Conference*, pages 6646–6651, Milwaukee, WI, June 2018.

[Gao03] Xing-Bao Gao. Exponential stability of globally projected dynamic systems. *IEEE Transactions on Neural Networks*, 14(2):426–431, 2003.

[GB14] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1, March 2014. Available at `http://cvxr.com/cvx`.

[GP79] T. Glad and E. Polak. A multiplier method with automatic limitation of penalty growth. *Mathematical Programming*, 17(1):140–155, 1979.

[GR01] C. D. Godsil and G. F. Royle. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer, 2001.

[GR12] P. E. Gill and D. P. Robinson. A primal-dual augmented Lagrangian. *Computational Optimization and Applications*, 51(1):1–25, 2012.

[HCG+17] J. Hu, J. Cao, J. M. Guerrero, T. Yong, and J. Yu. Improving frequency stability based on distributed control of multiple load aggregators. *IEEE Transactions on Smart Grid*, 8(4):1553–1567, 2017.

[HDGP16] J. T. Hughes, A. D. Domínguez-García, and K. Poolla. Identification of virtual battery models for flexible loads. *IEEE Transactions on Power Systems*, 31(6):4660–4669, Nov 2016.

[HDGP17] J. T. Hughes, A. D. Domínguez-García, and K. Poolla. Coordinating heterogeneous distributed energy resources for provision of frequency regulation services. In *Hawaii International Conference on System Sciences*, pages 2983–2992, Big Island, HI, January 2017.

[HFO+17] L. Hetel, C. Fiter, H. Omran, A. Seuret, E. Fridman, J. P. Richard, and S. I. Niculescu. Recent developments on the stability of systems with aperiodic sampling: An overview. *Automatica*, 76:309–335, 2017.

[HGCD+18] P. Hidalgo-Gonzalez, D. S Callaway, R. Dobbe, R. Henriquez-Auba, and C. J. Tomlin. Frequency regulation in hybrid power dynamics with variable and low inertia due to renewable energy. In *IEEE Conf. on Decision and Control*, pages 1592–1597, Miami Beach, FL, 2018.

[HGHACT19a] P. Hidalgo-Gonzalez, R. Henriquez-Auba, D. S. Callaway, and C. J. Tomlin. Frequency regulation using data-driven controllers in power grids with variable inertia due to renewable energy. In *IEEE PES General Meeting*, pages 1–5, Atlanta, GA, August 2019.

[HGHACT19b] P. Hidalgo-Gonzalez, R. Henriquez-Auba, D. S. Callaway, and C. J. Tomlin. Frequency regulation using sparse learned controllers in power grids with variable inertia due to renewable energy. In *IEEE Conf. on Decision and Control*, pages 3253–3259, Nice, France, Dec 2019.

[HJ85] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[HJT12] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *IEEE Conf. on Decision and Control*, pages 3270–3285, Maui, HI, 2012.

[HM94] U. Helmke and J. B. Moore. *Optimization and Dynamical Systems*. Springer, 1994.

[HNE17] M. T. Hale, A. Nedić, and M. Egerstedt. Asynchronous multiagent primal-dual optimization. *IEEE Transactions on Automatic Control*, 62(9):4421–4435, 2017.

[JJ09] B. Johansson and M. Johansson. Distributed non-smooth resource allocation over a network. In *IEEE Conf. on Decision and Control*, pages 1678–1683, Shanghai, China, December 2009.

[KB13] S. Koehler and F. Borrelli. Building temperature distributed control via explicit MPC and "Trim and Respond" methods. In *European Control Conference*, pages 4334–4339, Zurich, Switzerland, 2013.

[KBB15] W. Krichene, A. Bayen, and P.L. Bartlett. Accelerated mirror descent in continuous and discrete time. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2845–2853. Curran Associates, Inc., 2015.

[KCM15a] S. S. Kia, J. Cortés, and S. Martinez. Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. *Automatica*, 55:254–264, 2015.

[KCM15b] S. S. Kia, J. Cortés, and S. Martinez. Dynamic average consensus under limited control authority and privacy requirements. *International Journal on Robust and Nonlinear Control*, 25(13):1941–1966, 2015.

[Ker01] W. H. Kersting. Radial distribution test feeders. In *IEEE Power Engineering Society Winter Meeting*, volume 2, pages 908–912, Columbus, OH, January 2001.

[KH12] S. Kar and G. Hug. Distributed robust economic dispatch in power systems: A consensus + innovations approach. In *IEEE Power and Energy Society General Meeting*, San Diego, CA, July 2012. Electronic proceedings.

[Kha02] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, 3 edition, 2002.

[KM14] M. Kintner-Meyer. Regulatory policy and markets for energy storage in North America. *Proceedings of the IEEE*, 102(7):1065–1072, 2014.

[KMT98] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, 1998.

[Kos56] T. Kose. Solutions of saddle value problems by differential equations. *Econometrica*, 24(1):59–70, 1956.

[KSC+19] S. S. Kia, B. Van Scoy, J. Cortés, R. A. Freeman, K. M. Lynch, and S. Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems*, 39(3):40–72, 2019.

[Kun94] P. Kundur. *Power System Stability and Control*. McGraw-Hill, 1994.

[LD14] L. Luo and S. V. Dhople. Spatiotemporal model reduction of inverter-based islanded microgrids. *IEEE Transactions on Energy Conversion*, 29(4):823–832, 2014.

[LT18] J. Lu and C. Y. Tang. A distributed algorithm for solving positive definite linear equations over networks with membership dynamics. *IEEE Transactions on Control of Network Systems*, 5(1):215–227, 2018.

[Luc92] S. Lucidi. New results on a continuously differentiable exact penalty function. *SIAM Journal on Optimization*, 2(4):558–574, 1992.

[LZ15] Z. Li and D. Zhisheng. *Cooperative Control of Multi-Agent Systems: A Consensus Region Approach*. CRC Press, 2015.

[MKBD16] P. MacDougall, A. M. Kosek, H. Bindner, and G. Deconinck. Applying machine learning techniques for forecasting flexibility of virtual power plants. In *IEEE Electrical Power and Energy Conference (EPEC)*, pages 1–6, Ottawa, ON, Canada, Oct 2016.

[MKC13] J. L Mathieu, S. Koch, and D. S Callaway. State estimation and control of electric loads to manage real-time energy imbalance. *IEEE Transactions on Power Systems*, 28(1):430–440, 2013.

[MLM15] S. Mou, J. Liu, and A. S. Morse. A distributed algorithm for solving a linear algebraic equation. *IEEE Transactions on Automatic Control*, 60(11):2863–2878, 2015.

[MT11] M. Mazo Jr. and P. Tabuada. Decentralized event-triggered control over wireless sensor/actuator networks. *IEEE Transactions on Automatic Control*, 56(10):2456–2461, 2011.

[MXAP13] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.

[MZL17] E. Mallada, C. Zhao, and S. H. Low. Optimal load-side control for frequency regulation in smart grids. *IEEE Transactions on Automatic Control*, 62(12):6294–6309, 2017.

[Ned15] A. Nedić. Distributed optimization. In J. Baillieul and T. Samad, editors, *Encyclopedia of Systems and Control*. Springer, New York, 2015.

[Nes83] Y. E. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[Nes04] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Springer, 2004.

[Nes18] Y. Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer International Publishing, 2nd edition, 2018.

[NGC19] C. Nowzari, E. Garcia, and J. Cortés. Event-triggered control and communication of networked systems for multi-agent consensus. *Automatica*, 105:1–27, 2019.

[NS06] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2006.

[OC21] P. Ong and J. Cortés. Opportunistic robot control for interactive multiobjective optimization under human performance limitations. *Automatica*, 123:109263, 2021.

[PBD17] B. K. Poolla, S. Bolognani, and F. Dorfler. Optimal placement of virtual inertia in power grids. *IEEE Transactions on Automatic Control*, 62(12):6209–6220, 2017.

[PDRK18] Z. K. Pecenak, V. R. Disfani, M. J. Reno, and J. Kleissl. Multiphase distribution feeder reduction. *IEEE Transactions on Power Systems*, 33(2):1320–1328, March 2018.

[PJM] PJM. Dynamic regulation test signal (RegD) signal. `http://www.pjm.com/~/media/markets-ops/ancillary/regd-test-wave.ashx`.

[QL19] G. Qu and N. Li. On the exponential stability of primal-dual gradient dynamics. *IEEE Control Systems Letters*, 3(1):43–48, 2019.

[RC15] D. Richert and J. Cortés. Robust distributed linear programming. *IEEE Transactions on Automatic Control*, 60(10):2567–2582, 2015.

[Res63] J. Resh. The inverse of a nonsingular submatrix of an incidence matrix. *IEEE Transactions on Circuit Theory*, 10(1):131–132, 1963.

[RJ14] J. Rivera and H. Jacobsen. A distributed anytime algorithm for network utility maximization with application to real-time EV charging control. In *IEEE Conf. on Decision and Control*, pages 947–952, Los Angeles, CA, December 2014.

[RK12] A. U. Raghunathan and S. Krishnamurthy. A distributed anytime algorithm for maximizing occupant comfort. In *American Control Conference*, pages 1059–1066, Montreal, Canada, June 2012.

[RN05] M. G. Rabbat and R. D. Nowak. Quantized incremental algorithms for distributed optimization. *IEEE Journal on Selected Areas in Communications*, 23(4):798–808, 2005.

[Roc74] R. T. Rockafellar. Augmented Lagrange multiplier functions and duality in non-convex programming. *SIAM Journal on Control*, 12(2):268–285, 1974.

[RSR13] S. Rahnama, J. Stoustrup, and H. Rasmussen. Integration of heterogeneous industrial consumers to provide regulating power to the smart grid. In *IEEE Conf. on Decision and Control*, pages 6268–6273, Florence, Italy, 2013.

[SBC16] W. Su, S. Boyd, and E. J. Candès. A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. *Journal of Machine Learning Research*, 17:1–43, 2016.

[SC18] P. Srivastava and J. Cortés. Distributed algorithm via continuously differentiable exact penalty method for network optimization. In *IEEE Conf. on Decision and Control*, pages 975–980, Miami Beach, FL, December 2018.

[SC21a] P. Srivastava and J. Cortés. Nesterov acceleration for equality-constrained convex optimization via continuously differentiable penalty functions. *IEEE Control Systems Letters*, 5(2):415–420, 2021.

[SC21b] P. Srivastava and J. Cortés. Network optimization via smooth exact penalty functions enabled by distributed gradient computation. *IEEE Transactions on Control of Network Systems*, 8(3):1430–1441, 2021.

[SC22] P. Srivastava and J. Cortés. Solving linear equations with separable problem data over directed networks. *IEEE Control Systems Letters*, 6:596–601, 2022.

[SCC18] P. Srivastava, C.-Y. Chang, and J. Cortés. Participation of microgrids in frequency regulation markets. In *American Control Conference*, pages 3834–3839, Milwaukee, WI, May 2018.

[SCC21a] P. Srivastava, G. Cavraro, and J. Cortés. Agent-supervisor coordination for decentralized event-triggered optimization. *IEEE Control Systems Letters*, 2021. Submitted.

[SCC21b] P. Srivastava, C.-Y. Chang, and J. Cortés. Enabling DER participation in frequency regulation markets. *IEEE Transactions on Control Systems Technology*, 2021. Conditionally accepted.

[SDSJ19] B. Shi, S. S. Du, W. Su, and M. I. Jordan. Acceleration via symplectic discretization of high-resolution differential equations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Conference on Neural Information Processing Systems*, pages 5744–5752. Curran Associates, Inc., 2019.

[SHPV14] B. M Sanandaji, H. Hao, K. Poolla, and T. L Vincent. Improved battery models of an aggregation of thermostatically controlled loads for frequency regulation. In *American Control Conference*, pages 38–45, Portland, OR, 2014.

[SMT11] H. Saboori, M. Mohammadi, and R. Taghe. Virtual power plant (VPP), definition, concept, components and types. In *Asia-Pacific Power and Energy Engineering Conference*, pages 1–4, Wuhan, China, 2011.

[SOSM05a] D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Distributed sensor fusion using dynamic consensus. In *IFAC World Congress*, Prague, CZ, July 2005. Electronic proceedings.

[SOSM05b] D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Dynamic consensus on mobile networks. In *IFAC World Congress*, Prague, Czech Republic, July 2005.

[SR61] S. Seshu and M. B. Reed. *Linear Graphs and Electrical Networks*. Addison-Wesley Publishing Company, 1961.

[Tab07] P. Tabuada. Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52(9):1680–1685, 2007.

[TG15] B. Touri and B. Gharesifard. Continuous-time distributed convex optimization on time-varying directed networks. In *IEEE Conf. on Decision and Control*, pages 724–729, Osaka, Japan, 2015.

[TG20] B. Touri and B. Gharesifard. A modified saddle-point dynamics for distributed convex optimization on general directed graphs. *IEEE Transactions on Automatic Control*, 65(7):3098–3103, 2020.

[TGMD12] Q. Tran Dinh, S. Gumussoy, W. Michiels, and M. Diehl. Combining convex–concave decompositions and linearization approaches for solving BMIs, with application to static output feedback. *IEEE Transactions on Automatic Control*, 57(6):1377–1390, 2012.

[UBA14] A. Ulbig, T. S. Borsche, and G. Andersson. Impact of low rotational inertia on power system stability and operation. *IFAC Proceedings Volumes*, 47(3):7290–7297, 2014.

[Vel97] V. Veliov. A generalization of the Tikhonov theorem for singularly perturbed differential inclusions. *Journal of Dynamical & Control Systems*, 3(3):291–319, 1997.

[WAT+16] Y. Wang, X. Ai, Z. Tan, L. Yan, and S. Liu. Interactive dispatch modes and bidding strategy of multiple virtual power plants based on demand response and game theory. *IEEE Transactions on Smart Grid*, 7(1):510–519, Jan 2016.

[WDW+13] B. Washom, J. Dilliot, D. Weil, J. Kleissl, N. Balac, W. Torre, and C. Richter. Ivory tower of power: Microgrid implementation at the University of California, San Diego. *IEEE Power and Energy Magazine*, 11(4):28–32, 2013.

[WL09] P. Wan and M. D. Lemmon. Event-triggered distributed optimization in sensor networks. In *Symposium on Information Processing of Sensor Networks*, pages 49–60, San Francisco, CA, 2009.

[WM18a] X. Wang and S. Mou. A distributed algorithm for achieving the conservation principle. In *American Control Conference*, pages 5863–5867, Milwaukee, WI, June 2018.

[WM18b] J. T. Wen and S. Mishra. *Intelligent Building Control Systems*. Advances in Industrial Control. Springer, 2018.

[XB06] L. Xiao and S. Boyd. Optimal scaling of a gradient method for distributed resource allocation. *Journal of Optimization Theory & Applications*, 129(3):469–488, 2006.

[XBL05] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Symposium on Information Processing of Sensor Networks*, pages 63–70, Los Angeles, CA, April 2005.

[XHC+17] Y. Xu, T. Han, K. Cai, Z. Lin, G. Yan, and M. Fu. A distributed algorithm for resource allocation over dynamic digraphs. *IEEE Transactions on Signal Processing*, 65(10):2600–2612, 2017.

[XUDGS17] H. Xu, S. C. Utomi, A. D. Domínguez-García, and P. W. Sauer. Coordination of distributed energy resources in lossy networks for providing frequency regulation. In *IREP Bulk Power System Dynamics and Control Symposium*, Espinho, Portugal, August 2017.

[XW00] Y. S. Xia and J. Wang. On the stability of globally projected dynamical systems. *Journal of Optimization Theory and Applications*, 106(1):129–150, 2000.

[YYW+19] Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H. Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019.

[ZD16] M. Zachar and P. Daoutidis. Nonlinear economic model predictive control for microgrid dispatch. *IFAC-PapersOnLine*, 49(18):778–783, 2016.

[ZD17] M. Zachar and P. Daoutidis. Microgrid/macrogrid energy exchange: A novel market structure and stochastic scheduling. *IEEE Transactions on Smart Grid*, 8(1):178–189, 2017.

[ZDG20] M. Zholbaryssov and A. D. Domínguez-García. Convex relaxations of the network flow problem under cycle constraints. *IEEE Transactions on Control of Network Systems*, 7(1):64–73, 2020.

[ZH12] Q.-C. Zhong and T. Hornik. *Control of power inverters in renewable energy and smart grid integration*, volume 97. John Wiley & Sons, 2012.

[ZMS17] S. Zhang, Y. Mishra, and M. Shahidehpour. Utilizing distributed energy resources to support frequency regulation services. *Applied Energy*, 206:1484–1494, 2017.

[ZMSG11] R. D. Zimmerman, C. E. Murillo-Sánchez, and D. Gan. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011.

[ZSJ09] H. Zhang, P. Soldati, and M. Johansson. Optimal link scheduling and channel assignment for convergecast in linear WirelessHART networks. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 1–8, Seoul, South Korea, 2009.