Augmented Distributed Optimization for Networked Systems

Jinming XU

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2016

Statement of Originality

I hereby certify that the intellectual content of this thesis is the product of my original research work and has not been submitted for a higher degree to any other University or Institution.

....

Date Jinming XU

Acknowledgements

I wish to express my greatest gratitude to my advisor, Professor Yeng Chai Soh, for his professional guidance, endless patience and encouragement throughout my PhD study. I am sincerely grateful for the opportunity he gave and the independence he provided. His solid and wide knowledge as well as his insights to problems always impress me a lot during our discussions. I also wish to give my deepest gratitude to Professor Lihua Xie, for his unconditional help in supervising us in the last year of my PhD study. His attitude towards research and strong knowledge in control theory no doubt set a good example of what a true researcher should be.

My special gratitude goes to Professor Wonham not only for his bringing me to the area of control but also for his kind understanding of my situation and encouragement when I had a tough time in Zhejiang University and later in NTU.

I really enjoy the collaboration with Dr. Shanying Zhu who always challenges me with critical concepts, which greatly deepen my understanding to key aspects of problems. I also enjoy the discussions with Professor Hong Yiguang, Ye Maojiao and Guo Fanghong, which make me feel not alone in doing my research.

I would also like to thank Dr. Shuai Liu for his generous help before I came to NTU. It is my great privilege to work with so many talented fellow students in the Machine Learning and Internet of Things Labs, Zhao Shizheng, Sheldon, Lan Yuan, Zong Weiwei, Zhou Hongming, Zhao Wei, Ren Ye, Gao Kaizhou, Zhang Le, Cheng Zhenghua, Zhou zhi, Bai Zuo, Zhao Rui, Zhang Yong, Zhu Qingchang, Ding Jie, Xing Lantao, Cui Wei and many others, who made my stay in the lab really like being at home. I also appreciate the time that I have spent with the friends, Weng Renliang, Qi Yongbin, Wu Kai, Zhou Dexiang, Wang Shuai, Wang Xuehe, and many others, in body building, hiking, swimming and traveling, which made my stay in NTU so joyful and memorable. I am especially thankful for all the brothers and sisters who brought me a wonderful and peaceful church life.

Last but not least, I must also thank my beloved parents and sister, without whose unconditional love, support and understanding this work would not have been possible. I also like to thank my new family member–my little cute nephew–for bringing me so much happiness in the end of my PhD study.

Jinming Xu
Dec. 2015

"If I had one hour to save the world, I would spend 55 minutes defining the problem and only five minutes finding the solution."

—Einstein, Albert

To my dear family

Abstract

The dissertation develops several new schemes and algorithms for solving distributed optimization problems in large-scale networked systems in which a group of agents are to collaboratively seek the global optimum through peer-to-peer communication networks. The problem arises in various application domains, such as coordinated control, resource allocation and sensor fusion. Common features to these areas are that the system in question typically has a large number of agents involved without any centralized coordinator and that resources, such as sensing, communication and computation, are usually scattered throughout the network. As a result, the agents have to coordinate their behaviors with each other through only local information exchange to achieve a desired network (system) objective.

For coordinated control of large-scale networked systems, we propose a novel distributed simultaneous perturbation approach (D-SPA) to solve the distributed optimization problem based on simultaneous perturbation techniques as well as consensus strategies. The proposed method is model-free and requires little knowledge on the coupling structure of the problem to be optimized. Using singular perturbation and averaging theory, we show that the proposed scheme will converge to the neighborhood of the Pareto optimum of the problem so long as the energy of perturbation signals is sufficiently small. To illustrate its effectiveness, we apply the proposed approach to a simulated offshore wind farm for energy maximization and make a comprehensive comparison with the existing state-of-the-art technique.

On the other hand, for coordinated estimation in large-scale statistical signal processing, most existing distributed algorithms usually require a perfect synchronization mechanism and decaying stepsize for achieving the exact optimum, restricting it from being asynchronously implemented and resulting in slower convergence rates. In addition, the assumption of the boundedness of (sub)gradient is often made for convergence analysis, which is quite restrictive in unconstrained problems. To overcome these issues, we propose two augmented distributed algorithms both of which involve an extra step of consensus in each iteration. Specifically, a general

efficient distributed algorithm, termed Distributed Forward-Backward Bregman Splitting (D-FBBS), is proposed to simultaneously solve the primal problem as well as its dual based on the Bregman method and operator splitting. The proposed algorithm allows agents to communicate asynchronously and thus lends itself to stochastic networks. This algorithm belongs to the family of general proximal point algorithms and is shown to have close connections with some existing well-known algorithms for fixed networks but generally different from them in handling stochastic networks. To further tackle the asynchronous issues in computation, we propose a new augmented distributed gradient method (AugDGM) based on the existing well-known distributed gradient method. Both algorithms are able to converge to the exact optimum even with constant stepsize over stochastic networks without the assumption of boundedness of (sub)gradient. With proper assumptions, we establish a non-ergodic convergence rate of $o(1/k)$ in terms of fixed point residual for fixed networks and an ergodic convergence rate of $O(1/k)$ for stochastic networks respectively for the D-FBBS algorithm. For the asynchronous version of AugDGM (AsynDGM), we obtain an ergodic convergence rate of $O(1/\sqrt{k})$ in terms of the objective error for strongly convex functions with Lipschitz gradients over both fixed and stochastic networks. Some examples of sensor fusion problems are provided to illustrate the effectiveness of the proposed algorithms.

Contents

List of Figures

Symbols and Acronyms

Symbols

$\mathcal{R}^n$	the n-dimensional Euclidean space
$\mathcal{H}$	the Euclidean space
$\|\cdot\|$	the 2-norm of a vector or matrix in Euclidean space
$\|\cdot\|_G$	the induced norm of a vector in G-space
$\|\cdot\|_E$	the induced norm of a vector or matrix in probabilistic space
$\odot$	the Hadamard (component-wise) product
$\otimes$	the Kronecker product
$\langle\cdot,\cdot\rangle$	the inner product of two vectors
$\circ$	the composition of functions
$\mathcal{C}$	the average space, i.e., $span\{\mathbf{1}\}$
$\mathcal{C}^\perp$	the disagreement space, i.e., $span^\perp\{\mathbf{1}\}$
$\Pi_\parallel$	the projection matrix to the average space $\mathcal{C}$
$\Pi_\perp$	the projection matrix to the disagreement space $\mathcal{C}^\perp$
∇f	the gradient vector
$\bar{x}$	the vector with the average of all components of x as each entry
$\tilde{x}$	the disagreement vector defined as $x - \bar{x}$
$\mathbf{1}$	all-ones column vector with proper dimension
$O(\cdot)$	order of magnitude or ergodic convergence rate (running average)
$o(\cdot)$	non-ergodic convergence rate
$\{\cdot\}_{\geq 0}$	the non-negative sequence
$\mathcal{R}_{\geq 0}$	the set of non-negative reals
$\Gamma(\mathcal{H})$	the class of proper lower semi-continuous convex functions
$\mathcal{C}^k$	the function with continuous partial derivatives up to k orders

L	the Laplacian matrix or Lipschitz Constant
W	the weight matrix for communication
$\mathcal{G}$	the communication graph
$\mathcal{N}_i$	the index set of the neighbors of agent i

Acronyms

DOP	Distributed Optimization Problem
EDOP	Equivalent Distributed Optimization Problem
SDOP	Stochastic Distributed Optimization Problem
OEP	Optimal Exchange Problem
OCP	Optimal Consensus Problem
DOCP	Dynamic Optimal Consensus Problem
AugDGM	Augmented Distributed Gradient Methods
AsynDGM	Asynchronous Distributed Gradient Methods
D-ESC	Distributed Extremum Seeking Control
D-SPA	Distributed Simultaneous Perturbation Approach
D-FBBS	Distributed Forward-Backward Bregman Splitting
ADMM	Alternating Direction Method of Multipliers
DSM	Distributed (Sub)gradient Method
GAS	Globally Asymptotically Stable
UGAS	Uniformly Globally Asymptotically Stable
SPAS	Semi-globally Practically Asymptotically Stable
USPAS	Uniformly Semi-globally Practically Asymptotically Stable
HoS	Heterogeneity of Stepsize
FPR	Fixed Point Residual
OBE	Objective Error
i.i.d.	independent and identically distributed
a.s.	almost sure convergence of a random sequence

Chapter 1

Introduction

1.1 Scope and Overview

Networked systems are becoming prevalent nowadays due to the rapid development and deployment of control, communication and computation technologies in varied applications [1]. Examples include power grids [2], multi-robot systems [3] and sensor networks [4], just to name a few. A common feature of this kind of systems is that they typically consist of a large number of subsystems (agents) without any center involved[1], and the constituent components, such as actuators, sensors and controllers, are usually spatially scattered and connected over communication networks (cf. Figure 1.1). As a result, conventional centralized schemes will be no longer valid and, instead, each subsystem has to operate locally for the sake of scalability and robustness, leading to distributed approaches. In distributed schemes, agents acquire data locally, take actions independently and exchange information constantly with each other in order to serve certain network goals.

Distributed optimization has recently received renewed attention from the control and machine learning communities due to its wide applications in resource allocation [5], sensor fusion [6] and distributed learning [7]. Many coordinated control and estimation problems, such as distributed model predictive control (D-MPC) [8] and source localization [4], can be casted as distributed optimization problems (DOP). The desire of distributed optimization mainly comes from the requirements of faster and scalable algorithms by allocating relatively light subproblems to agents with

[1]It is impractical to have a central coordinator taking charge of the whole large-scale system.

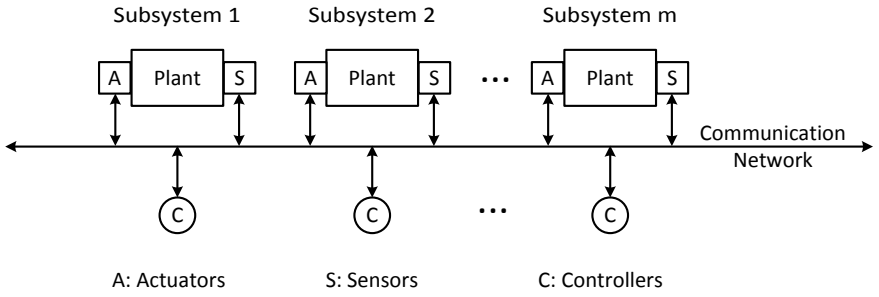


FIGURE 1.1: An illustration of a typical large-scale networked system.

limited computational power and communication resources, as well as the privacy concern, i.e., the objective being locally known to the associated agent. Besides privacy and speed, there is a more fundamental concern: the distributed scheme or algorithm should be able to operate correctly in time-varying and asynchronous scenarios for robustness concern (see Figure 1.2 for an overview of the practical concerns and issues of distributed optimization). The ultimate goal of distributed optimization is to have all agents coordinated in a distributed manner to achieve certain system objective while still taking into account their local interests. In large-scale dynamic systems, such as wind farm systems, one of the key problems is to coordinate the operating point of each subsystem (e.g., wind turbine) in order to reach an overall target, e.g., maximizing the overall power generation. Also, in large-scale signal processing, sensors need to work in collaboration with others for completing an overall task, such as reconstructing a temperature field or localizing a source which cannot be done by any sensor alone.

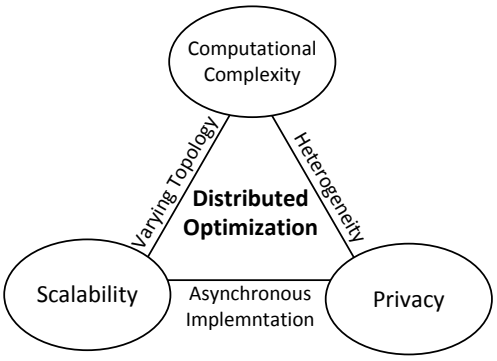


FIGURE 1.2: The main practical concerns and issues of distributed optimization.

On the other hand, in real large-scale networked systems, subsystems are coupled into each other in a very complex way and the comprehensive model of high-fidelity is too complex to be obtained. In addition, the communication network associated with the system is usually vulnerable to errors and is subject to random failures, leading to time-varying networks. Moreover, agents operate independently and may run out of pace with others due to the lack of global coordinator. One of the key challenges will be on how one can design a proper distributed scheme or algorithm that not only allows agents to operate locally and independently with each other but also be able to obtain the overall goal even under varying (stochastic) networks and asynchronous implementation[2], and being immune to the varying topology of the system. To this end, we study two basic distributed optimization problems involved in large-scale dynamic systems and sensor networks, corresponding to coordinated control and estimation over networks respectively. In particular, we develop a distributed scheme for the optimization of the steady-state performance of large-scale dynamic systems. The proposed scheme is immune to the varying coupling structure as well as the heterogeneity of the system to be controlled, i.e., using different feedback gains for different subsystems. For distributed estimation problems, we propose two basic distributed algorithms that not only allow for using constant stepsizes (thus being adaptive to varying environments) but also, most importantly, be able to seek the exact global optimal value of estimation even under stochastic networks and asynchronous implementation.

1.2 Major Contributions

Our main contributions can be stated as follows:

- *Coordinated control over networks*: For coordinated control of large-scale networked dynamic systems, we propose a novel approach for distributed optimization of the steady-state performance of the system based on consensus theory. The proposed scheme has favorable features of scalability and robustness in the sense that each subsystem takes action locally and only needs to communicate little information with its immediate neighbors for coordination without any center involved. We generalize the simultaneous perturbation

[2]See Section 6.4.1 for more details on asynchronous implementation.

technique to orthogonal perturbation technique where we employ orthogonal signals to perturb the system and extract the gradient information for subsequent optimization. In contrast to most related works, the stability of the system we obtained is semi-global and thus allows for more applications. Indeed, the proposed scheme is expected to be applicable to many existing large-scale dynamic systems, such as wind farm systems.

- *Coordinated estimation over fixed networks*: For distributed estimation problems in large-scale signal processing, we propose two basic distributed algorithms, namely AugDGM and D-FBBS. AugDGM can be regarded as an augmented version of the existing well-known distributed (sub)gradient method while D-FBBS can be thought of as node-based distributed alternating direction method of multipliers. The latter is shown to not only solve the primal problem but also the dual problem which is a typical problem in resource allocation. Both algorithms are able to seek the exact optimum even with constant stepsizes. For convergence performance, we establish an ergodic convergence rate of $O(1/\sqrt{k})$ in terms of the objective error for AugDGM employing a homogeneous (same) stepsize for coercive and convex functions with Lipschitz gradients while a non-ergodic convergence rate of $o(1/k)$ in terms of the fixed point residual for D-FBBS for general convex functions. Last but not least, to the best of our knowledge, we are the first to introduce the Bregman splitting method to solve distributed optimization problems which, in fact, provide a framework that allows us to design various distributed algorithms for different specific problems.

- *Coordinated estimation over stochastic networks*: The above proposed algorithms can be applied to stochastic networks even under asynchronous settings. In particular, with the assumption of strong convexity of the cost function, we establish an ergodic convergence rate of $O(1/k)$ in terms of the fixed point residual for D-FBBS over stochastic networks. Moreover, we show that, even under asynchronous implementation, AsynDGM can still achieve an ergodic convergence rate of $O(1/\sqrt{k})$ in terms of the objective error for strongly convex functions with Lipschitz gradients, which is the known best rate under the same setting as this work. Further, both algorithms are still able to seek the exact optimum almost surely even with constant stepsizes, yielding adaptive capability to varying environments. In these regards, we

have made significant improvements to distributed optimization, especially in dealing with stochastic networks and asynchronous implementation.

For the coordinated estimation part, the main differences of the proposed algorithms over the existing ones is summarized in terms of assumption, formulation, convergence rate and applicability to stochastic networks as follows:

Algorithm	**Consensus-based**†				**Dual-Decomposition-based**		
	DSM/DualAve*	ATC	NRC	**AsynDGM**	**D-FBBS**/P-EXTRA	D-ADMM	AsynADMM
Assumption (Convex)	Bounded (Sub)gradient	Bounded Hessien		**Coercive** $f \in \mathcal{C}^1$	f**: proper, closed, convex,** i.e., $f \in \Gamma(\mathcal{H})$	Convex	
Formulation	**Node-based**					Edge-based	
Convergence Rate	Objective Error (OBE)				Fixed Point Residual (FPR)		
	$O(\frac{log(k)}{\sqrt{k}})$‡	N.A.	$O(p^k)$	$\mathbf{O}(\frac{\mathbf{1}}{\sqrt{\mathbf{k}}})$	$\mathbf{o}(\frac{\mathbf{1}}{\mathbf{k}})$	N.A.	
Stochastic Network	Yes				Yes, but f to be strongly convex	No	
Remark	∗: DualAve stands for Dual Averaging Method [9]. †: Fast Distributed Gradient Methods require multiple cycles of running consensus at each iteration [10]. ‡: The rate is improved to $O(log(k)/k)$ when f is strongly convex [11] or Nestrov method is employed [10].						

TABLE 1.1: A comprehensive comparison of existing algorithms

1.3 Outline of the Thesis

Chapter 1 introduces the scope of this thesis and provides an overview of the general problem we consider in terms of the practical concerns as well as issues involved, which essentially motivates this research work. In this chapter, we also state our main contributions and briefly outline the thesis.

Chapter 2 reviews the existing literature from two basic categories: coordinated control over networks (corresponding to large-scale dynamic systems) and coordinated estimation over networks (corresponding to large-scale sensor networks). In particular, we provide a comprehensive review of the existing schemes and algorithms that are employed to solve the distributed optimization problem as well as the main drawbacks of these approaches, which is followed by the statement of what we have achieved towards overcoming these drawbacks.

Chapter 3 is devoted to the general distributed optimization problem as well as the philosophy behind it. In particular, we provide some insights to the problem by re-examining the popular (dynamic) average consensus protocol from the perspective of distributed optimization as well as the fundamental concept of coordination. These insights are then consolidated into the guiding principles for

designing distributed schemes and algorithms, which turns out to be very useful in understanding the schemes as well as algorithms proposed in this thesis.

Chapter 4 deals with the distributed optimization problem involved in networked large-scale dynamic systems. In particular, we present a distributed simultaneous perturbation approach for solving the problem by employing simultaneous perturbation techniques as well as consensus strategies. The stability analysis of the specific (basic and high-order) schemes is carried out based on singular perturbation and averaging theory. In this chapter, we also apply the proposed scheme to coordinated control of a simulated wind farm system and make a comprehensive comparison with the state-of-the-art technique to verify their effectiveness.

Chapter 5 is dedicated to the distributed estimation problem in large-scale sensor networks with fixed topology and synchronous implementation. In particular, under the setting of fixed topology, we develop two basic distributed algorithms for solving the estimation problem. We establish the connections of the proposed algorithms to some well-known existing algorithms and show that they outperform the existing algorithms in terms of convergence speed as well as accuracy.

Chapter 6 extends the algorithms proposed in the previous chapter to stochastic networks and asynchronous implementation. In particular, we show that the proposed algorithms, with some extra conditions on the cost function, can be still guaranteed to converge to the exact optimum even over stochastic networks while being asynchronous implemented. In this chapter, we have also established the specific convergence rates for both proposed algorithms, which are the best known rates under the same setting as this research work.

Chapter 7 summarizes the thesis and envisions the future work.

Chapter 2

Literature Review

2.1 Coordinated Control over Networks

Coordinated control of large-scale networked dynamic systems has been receiving renewed interests nowadays with emphasis on local communication among subsystems and local control of individual subsystems [1]. At the heart of this kind of control is distributed optimal control which not only stabilizes the overall system in a distributed fashion but also, most importantly, optimizes its *transient* as well as *steady-state* performance. One of the most well-established techniques is the distributed model predictive control (D-MPC) [12–15]. This control method, however, mainly focuses on real-time optimization of the *transient* performance of the system in a distributed way with known setpoint. In contrast to this method, there is also a resurgence of interest in extremum seeking control (ESC) which attempts to optimize the *steady-state* performance of the system in real-time without knowing the analytical form of performance so long as its value can be measured [16, 17]. This technique has been employed for seeking the Nash equilibrium in a multi-player game [18, 19], which is usually a sub-optimal solution [20]. It is well known that game theory, as a modeling technique dealing with the optimization problem of multiple decision makers, is closely related to distributed optimal control [20]. In particular, Waslander *et al.* [21] made an attempt to solve the decentralized optimization problem by utilizing the Nash Bargaining method. Semsar-Kazerooni and Khorasani [22] considered the multiple LQR problem from the viewpoint of game theory and attempted to obtain the Pareto-efficient solution.

Although many approaches are available for optimizing the *steady-state* performance of the system, gradient-like methods are much more robust and suitable in dealing with large-scale problems. In fact, gradient-based methods are widely employed in the existing literature to solve large-scale optimization problems in a distributed way. In particular, Tsitsiklis *et al.* [23] first studied the distributed gradient-like optimization algorithms in which a bunch of processors perform computations and exchange messages intending to minimize a common cost function. In the context of distributed computation, consensus theory lends itself to distributed implementation of algorithms as it allows agents to obtain global results by taking local actions and communicating limited information with its neighbors [24–26]. In line with these works, Nedic and Ozdaglar [27] applied consensus theory to multi-agent optimization problems where each agent only knows the cost of itself and aims to minimize the sum of the cost of all agents through cooperation with others, resulting in a distributed (sub)gradient method (DSM). Two drawbacks of these kinds of methods are that the communication cost will increase with the dimension of the problem to be optimized and the gradient should be computable exactly for optimization. On the other hand, dual decomposition has also been widely used to solve large-scale optimization problems which are separable in the dual domain [28, 29]. Rather than directly dealing with the primal problem, this method solves the dual counterpart which can be further divided into several small sub-problems that are relatively light to solve. Examples include formation control [3], multi-agent optimization [28], network utility maximization [30] and resource allocation [5]. This technique, however, requires the cost function to be separable for efficient gradient calculation and needs to consider the specific structure of the problem, restricting its application in dynamic networks.

In large-scale dynamic optimization problems, gradients may not be immediately available and we have to resort to some gradient approximation approaches. Instead of using the true gradient, one can solve the optimization problem by using the pseudo-gradient estimated from probing the system using perturbation techniques. One promising approach is the abovementioned ESC technique which has been widely employed to optimize systems without knowing their specific reference-to-output equilibrium map [16]. Here, we are particularly interested in the distributed implementation of ESC, which we term D-ESC. Existing applications of D-ESC include mobile sensor networks [18] and non-cooperative game [19]. Since they only considered non-cooperative games, their results are of Nash Equilibrium [20], which

is a sub-optimal solution. In order to obtain the global optimum (Pareto-efficient solution), using dual decomposition, the authors developed a preliminary version of D-ESC scheme which takes into account the global constraints but the proposed scheme needs to explicitly consider the physical interaction topology among agents which is not practical especially in dynamically changing environment [31]. To deal with time-varying networks, Kvaternik and Pavel [32] incorporated consensus protocols into extremum seeking algorithms. However, there is no explicit explanation on how to obtain the gradient information using certain probing technique which is crucial for implementation[1]. Although ESC has a long history [17], the rigorous proof of its stability of the general form is given only recently in [33] for local results and [34] for non-local results using singular perturbation and averaging analysis. It is claimed that their stability results can be extended without much effort to multi-variable extremum seeking control as done in [35].

In this thesis, we propose a new approach for optimizing the steady-state performance of the system in a distributed manner by resorting to simultaneous perturbation [36, 37] and consensus theory. In our approach, each agent is assumed to update only a subset of the components of the global vector, which is desirable in cases where only local action can be taken. The proposed scheme, termed distributed simultaneous perturbation approach (D-SPA), is model-free (derivative-free) and, different from most existing literature, only requires little knowledge regarding the dimension of the system as well as the underlying coupling structure of the problem[2]. It is also envisioned that the favorable properties of consensus algorithms are preserved, such as allowing for asynchronous implementation. We will show that the D-SPA scheme is able to obtain Pareto-optimum, which takes into account the interest of the adversary, in a distributed manner with a gap of the same order of the root mean square (RMS) amplitude of perturbation signals. In all, the D-SPA scheme is especially suitable for problems where we do not have much knowledge, e.g, wind farm system where the aerodynamic interactions among turbines are difficult to model. However, the drawback of gradient-free techniques is their slow convergence speed. Some extensions can be made to overcome this issue, e.g., Newton-based multi-variable extremum seeking control [38].

[1] Indeed, it is impossible to obtain the gradient information therein since the perturbation can not be made on the introduced auxiliary variables.

[2] Thus, the proposed scheme has the potential to adapt to slowly changing environment.

2.2 Coordinated Estimation over Networks

Existing distributed algorithms for large-scale coordinated estimation problems can be generally categorized into two main steams: (1) consensus-based approach, and (2) dual-decomposition-based approach. The former relies on the approximation of distributed algorithms to its centralized counterpart via consensus mechanism while the later incorporates the consensus requirement as a global consistency constraint, leading to a constrained optimization problem (cf. Section 3.2.2). In the following, we first review the existing algorithms that can be applied into fixed networks under synchronous implementation and then move on to discuss those that can be further applied into stochastic networks even under asynchronous implementation. The comprehensive comparison of the convergence performance of existing algorithms is addressed separately at the end of this section.

2.2.1 Fixed network and synchronous implementation

Besides (sub)gradient-based methods as mentioned above (cf. Section 2.1) [23, 27], there have been many extended versions presented in the existing literature. In particular, to speed up the convergence, Zanella and Varagnolo *et al.* [39] developed a distributed version of the Newton-Raphson algorithm by making use of the second-order derivative. Utilizing proximal functions, Duchi and Agarwal *et al.* [9] proposed a dual subgradient averaging method which shows better convergence results in terms of network scaling. The primal-dual approach has been widely used to account for (global) constraints imposed on the system [40–42]. In addition, cases with noisy observation of the gradient have been considered in [11, 43] and with directed graphs in [44]. However, most of the abovementioned consensus-based methods[3] require decaying stepsizes and the assumption of bounded (sub)gradient to achieve the exact optimum.

On the other hand, dual decomposition has been widely employed to solve large-scale optimization problems [28, 29]. It has been shown that distributed optimization problems can be transferred to an equivalent constrained optimization problem [45] for which we have a lot of existing solution techniques available. In particular, doing this allows us to transfer the problem as a saddle point problem

[3]Only the algorithm of Newton-Raphson consensus allows for using constant stepsize.

for which the traditional Arrow-Hurwitz-Uzawa method can be applied [46], especially the augmented Lagrangian method which permits better convergence performance [47]. However, introducing the augmented term results in the coupling issue among cost functions. To overcome this, a popular alternating direction method of multipliers (ADMM) is proposed which is shown to have very good convergence performance even for large-scale problems [48]. Distributed versions of ADMM have also been proposed for solving the distributed optimization problem [49–51]. However, this kind of technique, as mentioned earlier, depends heavily on the (coupling) structure of the problem. It is well known that the abovementioned methods are specific applications of the proximal point algorithm and operator splitting [52–54], which have been widely applied in signal processing [55] and image processing[56]. Moreover, due to the efficiency in solving optimization problems, Bregman-based proximal point algorithms, where the Euclidean distance is replaced with Bregman distance, have been successfully applied into image processing [57–59], min-max problems [60] and compressive sensing [61]. The dual-decomposition-based approach is generally able to achieve the exact optimum and obtain a better convergence rate. However, different from the consensus-based approach, its effectiveness relies heavily on the knowledge of the structure of the problem.

In this thesis, we propose two basic distributed algorithms. The first algorithm, termed AugDGM, can be regarded as an augmented version of distributed gradient methods where we introduce an extra step for the consensus on the gradients of objective functions. This algorithm allows for using *uncoordinated* stepsizes for local optimization, and is guaranteed to converge to the exact optimum even with *constant* stepsizes. This is a distinctive feature of the algorithm, which is not observed in the existing distributed algorithms in [6, 9, 10, 27, 49, 50, 62, 63]. We drop the restrictive assumption of boundedness of (sub)gradients of objective functions as required by those in [10, 27, 44] and, instead, only assume the standard condition of Lipschitz continuity to the problem. It is also important to note that the algorithm, though, has a similar augmented form as the ones proposed in [10, 39, 45, 64], it differs from them in that the assumptions (cf. Assumptions 5.6, 5.7, 6.3) are different and, most importantly, as we will show later, it can be applied to stochastic networks even under asynchronous implementation.

The other distributed algorithm, termed D-FBBS, is proposed based on the Bregman method and operator splitting thus belonging to the big family of proximal

point algorithms. Indeed, the Bregman splitting techniques used to develop this algorithm provide a framework which allows us to design different efficient distributed algorithms for specific objective functions with certain properties (e.g., having Lipschitz gradients) by employing corresponding splitting schemes. This proposed algorithm can deal with general convex functional and has close connections with some well-known existing algorithms. In particular, Jakovetic *et al.* [65] studied the linear convergence rate of a class of distributed augmented Lagrangian (AL) algorithms for twice continuously differentiable cost functions with a bounded Hessian when there are sufficient inner iterations of consensus being carried out. In the recent work [66], Shi *et al.* proposed a similar algorithm termed EXTRA for cost functions having Lipschitz gradients[4]. They also extended the algorithm to general convex problems and composite convex problems, yielding P-EXTRA and PG-EXTRA, respectively [67]. The proposed D-FBBS algorithm, though is generally different in its nature, has close connections with them in the sense that, with proper parameter setting, these existing algorithms can be shown to be equivalent to our algorithm with corresponding splitting schemes (cf. Section 5.3.4 for the detailed analysis). Note that the Bregman splitting method has been used in developing the Bregman Operator Splitting (BOS) algorithm [59] which deals with general equality-constrained problems. However, we will show that our algorithm differs from BOS in that we deal with distributed optimization problems and, instead of using a self-defined strongly convex function, we use the objective function to induce the Bregman distance. Also, different from this algorithm, we consider an asymmetric saddle point problem (cf. Equation (5.22)), which, as we will see later, allows us to effectively deal with stochastic networks. It should be noted that the algorithm is node-based and thus, unlike the edge-based counterparts [50, 68], is more capable in terms of algorithm scaling and distributed computation.

2.2.2 Stochastic network and asynchronous implementation

In distributed algorithms, due to the absence of global clock, agents will operate according to their local clocks and the execution of the algorithm may be out of synchronism. In addition, communication networks are vulnerable and are subject to random failures due to inevitable errors. Asynchronous implementation

[4]When the cost function is also strongly convex, linear convergence rate can be achieved.

of algorithms is thus necessary and typically involves in communication as well as computation processes. However, there are only few works devoted to asynchronous issues of distributed optimization problems. In particular, to avoid the worst-case bounded communication assumption made in [27], Lobel and Ozdaglar [69] considered the same distributed gradient method for stochastic networks where communication links are subject to random failures[5]. An asynchronous broadcast-based algorithm is designed in [62] to deal with random link failures as well as uncoordinated update. Srivastava and Nedic [43] considered the extension of this algorithm to account for subgradient error and noisy communication links and established the almost sure convergence for uncoordinated diminishing stepsize and error bounds for constant stepsize for continuously differentiable and strongly convex functions. Note that the communication model considered therein is bi-directional and the algorithm requires the stepsize to follow certain predefined decaying rule and the Poisson rates of activation to be the same for all agents to ensure the ability to seek the exact optimum. Asynchronous computation of Newton-Raphson Consensus (NRC) has been investigated in [70] which requires the cost function to be continuous up to the second derivative. Asynchronous ADMM based on node-wise or edge-wise randomized iterations have also been proposed for asynchronous implementation [63, 65, 71]. However, their methods are mainly based on randomized iteration of algorithms which requires the weight matrix to be constant [65, 68] and thus should be deemed applicable only to fixed networks. Recently, a subgradient-push approach built upon the well-known push-sum algorithm [26] is proposed in [44] to account for asynchronous communication over directed networks.

2.2.3 Convergence rate comparison

The convergence rate of distributed algorithms established in the literature is always inferior to their central counterparts. The difference of the convergence performance is further enlarged when it comes to stochastic networks and asynchronous implementation. In particular, both the dual-averaging method [9] and subgradient-push method [44] can only achieve the ergodic[6] rate of $O(\ln k/\sqrt{k})$ in terms of the *objective error* (OBE) with a decaying stepsize of $1/\sqrt{k}$, and its improvement to $O(\ln k/k)$ is obtained in [10] with the accelerated *Nesterov* method

[5]The stochastic model therein is more general as they allow the link failures to be dependent.
[6]We refer to ergodic convergence rate as the rate obtained in terms of running average.

and in [11] for strongly convex functions with Lipschitz gradients respectively. In contrast, with D-FBBS algorithm, we establish a non-ergodic convergence rate of $o(1/k)$ in terms of the fixed point residual (FPR) for fixed networks which is comparable with the centralized counterpart[7]. With an extra assumption of strong convexity of the cost function, we obtain an ergodic convergence rate of $O(1/k)$ for stochastic networks with the same algorithm. For AugDGM under fixed networks, we establish an ergodic convergence rate of $O(1/\sqrt{k})$ in terms of OBE for coercive and convex functions with Lipschitz gradients while, for AsynDGM under stochastic networks even in the context of asynchronous settings, we obtain an ergodic convergence rate of $O(1/\sqrt{k})$ for strongly convex functions having Lipschitz gradients. In these regards, we make a significant improvement on the convergence performance of distributed algorithms for distributed optimization problems, making them more suitable for asynchronous implementation over stochastic networks.

[7]Note that a very recent work also achieve the similar convergence rate as this work for fixed networks [66] (see Section 5.3.4 for the detailed analysis).

Chapter 3

Distributed Optimization in Networked Systems

This chapter introduces the general distributed optimization problem we will be dealing with in subsequent chapters and provide some insights on distributed optimization. In particular, in Section 3.1, we will introduce the topology model, including the coupling structure among agents and the topology of communication network, as well as the formulation of the general optimization model to be considered. The well-known (dynamic) average consensus protocols are then reviewed from a new perspective in Section 3.2, followed by some insights on distributed optimization in Section 3.3, with an emphasis on the philosophy behind it.

3.1 Topology and Optimization Model

We consider a large-scale networked system consisting of a large number of subsystems (agents) each of which has a local objective to be optimized. We assume agents are allowed to exchange information with each other in order to achieve certain network (system) objective. In particular, we use $\mathcal{G}_p = (\mathcal{V}_p, \mathcal{E}_p)$ to denote the coupling structure of the system. On the other hand, due to either the privacy requirement or the limitation of communication resources, agents have to exchange information with its immediate neighbors to come up with aggregated information via certain communication network underpinned by a graph $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c)$.

Most existing networked systems naturally exhibits certain distributed coupling structure which can be captured by a graph. For instance, in wind farm systems, only the upstream wind turbine will impact the power generation of the down stream turbine. For this kind of system, we only require $\mathcal{G}_p \subseteq \mathcal{G}_c$ and we can design a trivial and more efficient communication protocol (e.g., simply summing up the cost received from its downstream neighbor) that relies on merely the interacting neighbors, leading to distributed scheme or algorithms. However, this kind of protocol will disclose the information to its neighbors and is vulnerable to packet loss and topology changing. It somewhat resembles centralized approach and, in fact, it is the inherent distributed structure of the physical interaction that leads to the distributed scheme. In order to account for the scenarios where $\mathcal{G}_p \nsubseteq \mathcal{G}_c$ and for generality, we assume each local objective is dependent of the entire decision vector, yielding a fully coupled structure, i.e., $\mathcal{G}_p$ being a fully connected graph. In this case, we need to design some distributed algorithms that can acquire the aggregate information over the whole network by merely local communication and computation. As we will show later, consensus protocols will play a key role in achieving this goal. In addition, we assume the whole networked system can be properly partitioned in such a way that each subsystem is loosely coupled with each other. By being loosely coupled, we mean, by various means, each subsystem can be approximated as an "oracle" system such that, whenever fed with certain input θ, the system will return the corresponding output $f(\theta)$. This is particularly true when it comes to optimize the quasi-steady-state performance of dynamic systems, e.g., a wind turbine always working at certain operation point and producing certain power corresponding to this operation point. We thus assume a fully coupling structure of the problem for generality and model the communication network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with each node $i \in \mathcal{V} = \{1, 2, ..., m\}$ representing each agent and each directed edge $e \in \mathcal{E}$ indicating the direction of allowed information flow.

Figure 3.1 illustrates the general distributed optimization model. Given an above-mentioned networked system consisting of m agents, the objective of the network is to minimize the following function in a cooperative way:

$$F(\theta) = \sum_{i=1}^{m} f_i(\theta) \tag{DOP}$$

where $\theta \in \mathcal{R}^d$ is the global parameter to be optimized while $f_i : \mathcal{R}^d \to \mathcal{R} \cup \{\pm\infty\}$ is the local cost function available only to agent i.

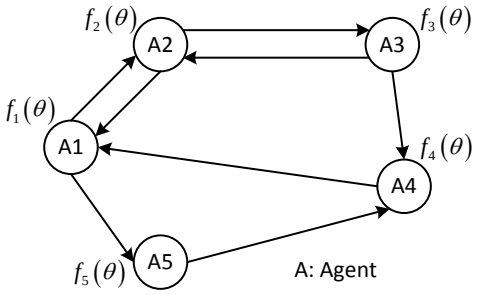


FIGURE 3.1: An illustration of a general distributed optimization model. The solid lines indicate the information flow while the circles denote the agents that are fully coupled with each other through the global decision vector θ.

3.2 Canonical Distributed Optimization

We show that the well-known (dynamic) average consensus protocol can be regarded as a canonical distributed optimization problem in the sense that it can be derived from the perspective of distributed optimization. We also provide the big picture behind distributed optimization by re-examining the fundamental questions of what coordination is and how we can achieve it.

3.2.1 Consensus protocol revisited

Consensus theory can be dated back to the DeGroot learning model where a group of agents are to reach an agreement on the belief of certain subject via sharing their individual opinions [72]. The formal definition of consensus seeking is as follows:

Definition 3.1 (Consensus Seeking [72])**.** Given a sequence $\{x_k\}_{k\geq 0}$ of m dimension, we say a consensus is reached if $\lim_{k\to\infty} \|x_{i,k} - x_{j,k}\| = 0, \forall i, j \in \{1, 2, ..., m\}$.

Now, consider the canonical distributed optimization problem[1] as follows:

$$\min_{x_i} \sum_{i=1}^{m} (x_i - r_i)^2, \ \ s.t.\ x_i = x_j, \forall i, j \in \mathcal{V},$$

[1]It is not difficult to see that this is an alternative form conforming to the general distributed optimization problem (DOP).

which can be shown to be equivalent[2] to

$$\min_x = \frac{1}{2}\|x-r\|^2, \ s.t. \ Kx = 0,$$

where $K^TK = L$ with L being the Laplacian matrix, we introduce the Lagrangian associated with the above optimization problem:

$$\psi(x,y) = \frac{1}{2}\|x-r\|^2 - y^TKx.$$

Applying the inexact Uzawa Method to the above Lagrangian yields

$$x = \inf_x \psi(x,y) \tag{3.1a}$$

$$\dot{y} = -Kx. \tag{3.1b}$$

Solving (3.1b) analytically leads to

$$x = r + K^Ty \tag{3.2a}$$

$$\dot{y} = -Kx. \tag{3.2b}$$

Taking the derivative of both sides of (3.2a) and combining with (3.2b) leads to

$$\dot{x} = -K^TKx + \dot{r} = -Lx + \dot{r}, \tag{3.3}$$

This is exactly the dynamic average consensus. In addition, it is not difficult to see that when $r = 0$, the above dynamics will reduce to

$$\dot{x} = -Lx, \tag{3.4}$$

which is the well-known consensus protocol.

Remark 3.1. As an extension to the basic consensus protocol (3.4), dynamic average consensus (3.3) can ensure that the instantaneous sum of the state of all agents is the same as that of the reference input. This conservation property allows us to track the time-varying average of the reference input and thus provides *a feasible way to develop distributed versions of many existing schemes and algorithms.*

[2]This is true when we have certain property for the Laplacian matrix, i.e., $null\{L\} = null\{K\} = span\{\mathbf{1}\}$ (see Section 5.3.1 for the detailed analysis).

3.2.2 Consensus mechanism for coordination

Perhaps the most fundamental concern related to coordinated control and estimation problems is on the formal definition of coordination which essentially characterizes the associated distributed schemes and algorithms. In constrained optimization, the Lagrange multiplier is often introduced via dual decomposition to account for global constraints, functioning as a coordinator (e.g., the price in economics). One may regard this as some sort of coordination among agents by satisfying certain constraints. However, this is done in a parallel way in the sense that there is always an external center taking charge of the update of the Lagrange multiplier for ensuring constraints. Coordination is, instead, more about distributed implementation where there is no super-center providing "instructions" for the process of coordination. This is particularly true in distributed computation where consensus theory is widely applied to achieve synchronization among agents' computation in the absence of global clock. The main purpose of consensus protocol is to propagate (diffuse) information over the network in a completely distributed way, guaranteeing the consistency of data updating. Thus, consensus mechanism therein plays the similar role as a global clock. Although the consensus mechanism is more fundamental than satisfying constraints in coordination, it can be stated in terms of a global *consistency constraint* such as $Lx = 0$, which, as we will show later, allows us to transform the original problem (DOP) to many equivalent forms on which various distributed algorithms can be developed.

3.3 The Evolution and Philosophy

It is well known that centralized optimization suffers from high computational complexity especially in dealing with large-scale problems. A popular way to tackle this issue is to decompose the problem into several relatively light subproblems each of which is then allocated to an agent with limited capacity, leading to a parallel optimization technique such as the well-known ADMM [48]. This kind of technique is, however, not totally distributed in the sense that there is still a center unit involved taking charge of the update of some important parameter (e.g., Lagrangian multiplier) which is centrally stored. Distributed optimization pushes the idea further towards a totally distributed implementation by introducing

to each agent a local copy of the parameter that is locally stored. In so doing, agents are allowed to operate based on the local copy of the information on its own without much intervention with its neighbors and only needs to exchange few data for ensuring consistency via certain kind of consensus mechanism. Note that distributed optimization differs from decentralized optimization in that it explicitly takes into account the impacts of the interest of each agent on others (thus yielding Pareto optimum) rather than simply ignores it leading to Nash equilibrium which is usually sub-optimal. Figure 3.2 illustrates the basic difference among centralized, parallel and distributed optimization.

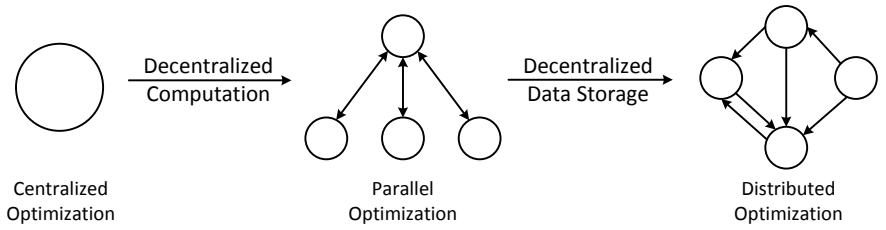


FIGURE 3.2: An illustration of the evolution from centralized optimization to parallel optimization and distributed optimization.

We now summarize the above analysis into the following basic steps as the philosophy behind distributed optimization techniques (see Figure 3.3 for the detail):

1) Decouple the problem by introducing auxiliary variables (local copies),
2) Do local operations, such as local update for optimization or local learning (e.g., gradient search or orthogonal perturbation) and local communication,
3) Employ certain consensus mechanism to ensure consistency.

The above philosophy only provides a general guideline for developing distributed optimization techniques. There is much room to be explored beyond the framework in order to achieve specific requirements. For instance, it is not necessary to separate the local update step (2) and the consensus step (3). Indeed, these two steps can be coupled into each other and carried out simultaneously. Most existing distributed optimization techniques follow this idea of combining the consensus step and local update for optimization step into one single step. In contrast, as we will see later, the main idea of our approaches proposed in this thesis is the

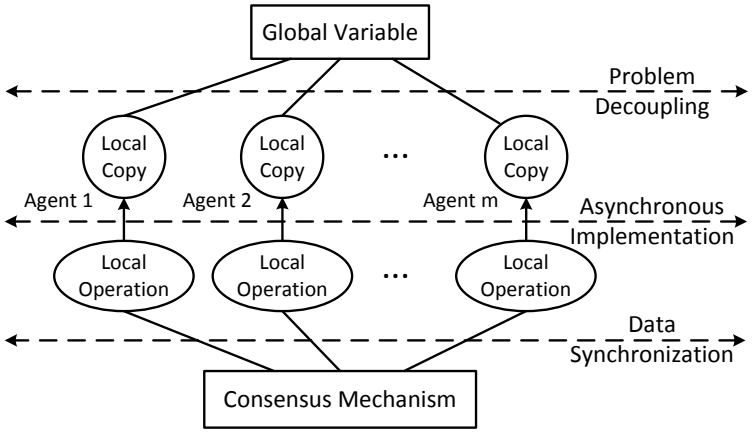


FIGURE 3.3: The philosophy behind distributed optimization.

introduction of an extra step for consensus on certain parameters, leading to algorithms or schemes in an *augmented form*. In so doing, we can effectively separate the consensus step and the local optimization step, eliminating the steady-state error which cannot be avoided by most existing algorithms that have one single combined step. This phenomenon somewhat resembles the PI feedback control in classical control theory where the integral part, different from the P-control, is introduced to compensate for the unknown constant disturbance. What is more interesting and important is that by introducing the extra step for consensus, the algorithm turns out to be more capable in dealing with time-varying (stochastic) networks even under asynchronous implementation.

Part I

Coordinated Control

Chapter 4

Distributed Optimization in Networked Control Systems: A Simultaneous Perturbation Approach

This chapter is concerned with the optimization of the steady-state performance of large-scale networked dynamic systems in a distributed manner. We first formulate the problem in Chapter 4.1, followed by some preliminaries for this chapter. The specific approaches as well as the corresponding stability analysis is then given in Chapters 4.3 and 4.4. We finally apply the proposed approach into a simulated wind farm system and compare it with the state-of-the-art technique in Chapter 4.5.

4.1 Problem Statement

4.1.1 Multi-agent dynamics

We consider a large-scale dynamic system consisting of m agents, each of which can be depicted as follows:

$$\begin{cases} \dot{x}_i = f_i(x, u_i) \\ z_i = h_i(x), \quad i \in \mathcal{V} := \{1, 2, ..., m\} \end{cases} \tag{4.1}$$

where $x_i \in \mathcal{R}^{n_i}$ is the state of agent i and $u_i \in \mathcal{R}$ is the control input[1] while $x = [x_1^T, x_2^T, ..., x_m^T]^T \in \mathcal{R}^n$ and $u = [u_1, u_2, ..., u_m]^T \in \mathcal{R}^m$ denote the state and control input of the whole system, respectively, $z_i \in \mathcal{R}$ is the cost of agent i, and $f_i : \mathcal{R}^n \times \mathcal{R} \to \mathcal{R}^{n_i}$ and $h_i : \mathcal{R}^n \to \mathcal{R}$ are both $\mathcal{C}^1$ functions. Note that the analytical form of h_i can be unknown as long as its value can be measured. In addition, agents are assumed to be loosely coupled through dynamics but could be strongly coupled in performance. For instance, in wind farm systems, each turbine can be controlled locally but its power generation depends on others due to wake effect. To make this precise, we make the following assumptions.

Assumption 4.1. There exists a locally Lipschitz function $l : \mathcal{R}^m \to \mathcal{R}^n$ and a control law $u = \varphi(x, \theta)$, where $\theta \in \mathcal{R}^m$ can be regarded as the reference point for the system, such that $f(x, \varphi(x, \theta)) = 0$ if and only if $x = l(\theta)$.

Assumption 4.2. The system (4.1) is globally asymptotically stable (GAS), uniformly in $\theta = [\theta_1, \theta_2, ..., \theta_m]^T \in \mathcal{R}^m$.

Remark 4.1. Assumptions 4.1 and 4.2 essentially tell us that the system could be stabilized in a distributed manner. A simple example satisfying this assumption is that of coordinated control of wind farm systems where each wind turbine is not coupled physically with each other and thus can be controlled individually.

4.1.2 Communication network

We assume agents are able to communicate with their immediate neighbors for collaboration through a communication network described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In addition, we model the communication process as a continuous dynamic system (see the brief analysis for the rationales in Remark 4.2) and do not take into account the effect of quantization and packet loss involved in the communication network. Instead, for brevity, we only consider fixed communication topology[2]. To be precise, we make the following assumption on the communication network:

Assumption 4.3. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ underpinning the communication network is fixed, balanced and strongly connected such that we have $\mathbf{1}^T L = 0$ and $L\mathbf{1} = 0$, where L is the Laplacian matrix of the graph.

[1] To simplify the presentation, we assume each subsystem has only one scalar input.
[2] As we will show in the sequel, the result is, in fact, not limited to fixed networks.

Remark 4.2. In networked control systems, it has been shown that the controller designed without considering the network is able to preserve the stability properties[3] of the system presented with network so long as a Lyapunov UGAS protocol is employed and certain parameter associated with the network, e.g., maximum allowable transmission interval (MATI), is designed to be sufficiently small [73, 74].

4.1.3 Dynamic optimal consensus problem

We consider a distributed optimization problem in which several dynamic agents are, based on local resources, to collaboratively seek the optimum of the sum of their individual costs. Each agent is assumed to only have access to local cost which can be measured in some way (local sensing) and take charge of a subset of the components of the global decision vector (local action).

Specifically, we focus on dynamic systems that are always operating at its equilibrium point or systems exhibiting two-time-scale structure such that the transient dynamics can be ignored leading to static reference-to-output equilibrium map [33]. It basically means that, given any reference point, the system will be asymptotically stabilized in its steady-state in a relatively fast way. However, we do not restrict our attention to this kind of dynamic optimization problems. It will become clear that the proposed scheme can be also employed to solve static (mathematical) optimization problems in which the true gradient is difficult to obtain explicitly but can be approximately estimated from measurements, e.g., neural network training [75]. In addition, we assume that the cost of each agent is transferable thus they can be simply summed up for optimization, resulting in a Pareto-optimal solution (cf. Remark 4.3). For cases where costs are not transferable, other methods, such as objective product method, can be employed to account for the fairness issue [76].

The above problem[4], which we term dynamic optimal consensus problem (DOCP), can be formulated as follows:

$$\theta^\star = \arg\min_{\theta \in \mathcal{R}^m} \sum_{i=1}^{m} J_i(\theta) \tag{DOCP}$$

[3]Here, the stability is preserved in the sense that the nominal system which is GAS and has no network issue involved will still be semi-globally practically asymptotically stable in certain parameter, e.g., MATI, when there is network presence in the same system.

[4]Without loss of generality, we will only consider the minimization problem as maximization problems can be treated identically.

where $\theta = [\theta_1, \theta_2, ..., \theta_m]^T \in \mathcal{R}^m$ is the global decision vector and J_i denotes the cost function of agent i. As mentioned earlier, for simplicity, we assume $\theta_i \in \mathcal{R}, \forall i \in \mathcal{V}$, meaning that each agent is responsible for one-dimensional component of the global decision vector. In addition, we make the following assumptions on the cost function as well as the optimization problem:

Assumption 4.4. The cost functions $J_i = h_i \circ l(\theta)$, $\forall i \in \mathcal{V}$ are twice continuously differentiable, i.e., $J_i \in \mathcal{C}^2$, $\forall i \in \mathcal{V}$ and their level sets $\Omega_c^i = \{\theta \in \mathcal{R}^n | J_i(\theta) \leq c\}$, $\forall i \in \mathcal{V}$ are bounded for all values of c.

Assumption 4.5. There exists a unique solution $\theta^\star$ attaining the Pareto-optimum (i.e., global optimum) of the DOCP problem.

4.2 Preliminaries

4.2.1 Singular perturbation and averaging theory

Let us consider the following parameterized or perturbed system:

$$\dot{x} = f(x, \varepsilon) \tag{4.2}$$

where ε is a small positive number and $f : \mathcal{R}^n \times [-\varepsilon_0, \varepsilon_0] \to \mathcal{R}^n$ is piecewise continuous and locally Lipschitz in (x, ε). The goal of the perturbation method is to exploit the smallness of the perturbation parameter ε [77]. Thus, instead of studying the perturbed system directly, we carry out the stability analysis on the nominal or unperturbed system which is obtained by setting $\varepsilon = 0$:

$$\dot{x} = f(x, 0) \tag{4.3}$$

A system exhibiting two-time scale structure can be casted into the following standard singularly perturbed model via proper coordinate transformation [77]:

$$\dot{x} = f(x, z, \varepsilon) \tag{4.4a}$$

$$\varepsilon \dot{z} = g(x, z, \varepsilon) \tag{4.4b}$$

where f and g are continuously differentiable functions in specific domains.

With the above model, we present the following important definitions and lemma:

Definition 4.1 (USPAS [78])**.** The parameterized system $\dot{x} = f(t, x, \varepsilon)$ is said to be uniformly semi-globally practically asymptotically stable (USPAS) on ε if there exists $\kappa \in \mathcal{KL}$[5] and, for each pair of strictly positive numbers (Δ, δ), there exists a real number $\varepsilon^\star = \varepsilon^\star(\Delta, \delta) > 0$ such that for all initial condition x_0 with $\|x_0\| \in \Delta$ and for each $\varepsilon \in (0, \varepsilon^\star)$, we have $\|x(t)\| \leq \kappa(\|x_0\|, t - t_0) + \delta, \forall t \geq t_0 \geq 0$.

Definition 4.2 (Average)**.** A locally Lipschitz function $\rho : \mathcal{R}_{\geq 0} \times \mathcal{R}^N \to \mathcal{R}^N$, is said to have an average $\rho^{av}(x)$ if there exists a period T such that

$$\rho^{av}(x) = \frac{1}{T} \int_t^{t+T} \rho(\tau, x) d\tau, \ \forall t \in \mathcal{R}_{\geq 0}$$

exists and

$$\left\| \int_t^{t+s} \rho(\tau, x) - \rho^{av}(x) d\tau \right\| \leq K, \ 0 \leq s \leq T, \ \forall t \in \mathcal{R}_{\geq 0}$$

where K is $O(1)$ positive constant.

Lemma 4.1 ([34, 79])**.** *Consider the following two-time scale system:*

$$\begin{aligned} \dot{x} &= f(t, x, z, \varepsilon) \\ \varepsilon \dot{z} &= g(t, x, z, \varepsilon) \end{aligned} \tag{4.5}$$

where $\varepsilon \in \mathcal{R}$ is a small positive number. Suppose the following conditions hold:

- *the algebraic equation $0 = g(t, x, z, 0)$ has an isolated root $z = h(x, t)$,*
- *the functions f, g, h and $\frac{\partial h}{\partial x}$ are locally Lipschitz in (x, z, ε), uniformly in t,*
- *the reduced system $\dot{x} = f(x, h(x, t), 0)$ is USPAS on ε, and*
- *the origin of the boundary-layer system*

$$\frac{dy}{d\tau} = g(t, x, y + h(x, t), 0)$$

with $y = z - h$, $t = \varepsilon\tau$, is GAS, uniformly in (t, x).

Then, the singularly perturbed system is USPAS on ε.

[5] Refer to [77, Def. 4.2 and 4.3] for the specific definition of $\mathcal{KL}$ functions.

4.2.2 Cooperative and non-cooperative game

In a nutshell, game theory is a study of team decision making problems. There are two kinds of games according to the way the players play the game, namely non-cooperative games and cooperative games. In non-cooperative games, each player takes actions independently in order to minimize its own loss function without taking into account others' interests, leading to the outcome of Nash equilibrium. On the other hand, in cooperative games, players have to sacrifice their own benefits for achieving global results, yielding Pareto-optimal outcomes.

To be more accurate, we provide the formal definitions of Nash Equilibrium and Pareto-optimum for an m-player nonzero-sum game (Θ, J) where $\Theta = \Theta_1 \times \Theta_2 \cdots \times \Theta_m$ is the set of strategy profiles with Θ_i denoting the strategy set for player $i \in \mathcal{V}$ and $J = [J_1(\theta), J_2(\theta), ..., J_m(\theta)]^T$ is the cost[6] for $\theta \in \Theta$.

Definition 4.3 (Nash Equilibrium [20])**.** A strategy profile $\theta^\star = [\theta_1^\star, \theta_2^\star, ..., \theta_m^\star]^T \in \Theta$ with $\theta_i^\star \in \Theta_i,\ i \in \mathcal{V}$ is said to constitute a Nash equilibrium solution for an m-player nonzero-sum game if the following conditions hold

$$J_i(\theta_i, \theta_{-i}^\star) \geq J_i(\theta_i^\star, \theta_{-i}^\star),\ \forall \theta_i \in \Theta_i,\ i \in \mathcal{V}$$

where $\theta_{-i}^\star$ denotes the strategies of all other players.

Definition 4.4 (Pareto-Optimum [76])**.** A strategy profile $\theta^\star = [\theta_1^\star, \theta_2^\star, ..., \theta_m^\star]^T \in \Theta$ with $\theta_i^\star \in \Theta_i\ i \in \mathcal{V}$ is said to constitute a Pareto-optimal solution for an m-player nonzero-sum game if there does not exist another strategy $\theta \in \Theta$ such that $J_i(\theta^\star) \geq J_i(\theta),\ \forall i \in \mathcal{V}$ and $J_k(\theta^\star) > J_k(\theta)$ for at least one player $k \in \mathcal{V}$.

Remark 4.3. The most common way to obtain the Pareto-optimal solution is to use the weighted sum method, i.e., minimizing $\sum_{i=1}^{N} w_i J_i(\theta)$, which admits a unique solution and is sufficient for achieving Pareto optimality [76]. In addition, if one player can losslessly transfer part of its cost to another player (e.g., they have a common currency to evaluate their cost), then we can simply optimize their sum for Pareto-optimality.

[6] Without loss of generality, we will deal with cost instead of payoff as used in game theory.

4.3 A Distributed Simultaneous Perturbation Approach

In this section, we present a distributed simultaneous perturbation approach (D-SPA). In this scheme, the simultaneous perturbation technique using orthogonal signals is employed to obtain the pseudo-gradient. In addition, dynamic average consensus is introduced for distributed implementation of this technique.

4.3.1 Simultaneous perturbation for gradient extraction

We show that employing orthogonal perturbation signals allows us to extract the gradient information for a given performance function.

Definition 4.5 (Orthogonal Perturbation Signals)**.** A number of $\mathcal{C}^1$ signals $\mu = [\mu_1(t), \mu_2(t), ..., \mu_m(t)]^T$ are said to be orthogonal perturbation signals if there exists a period[7] T such that, for any $t \geq 0$, the following conditions are satisfied:

$$\frac{1}{T}\int_t^{t+T} \mu_i(\tau)^2 d\tau = a^2, \ \frac{1}{T}\int_t^{t+T} \mu_i(\tau) d\tau = 0, \tag{4.6a}$$

$$\frac{1}{T}\int_t^{t+T} \mu_i(\tau) \cdot \mu_j(\tau) d\tau = 0, \ \forall i \neq j \in \mathcal{V}, \tag{4.6b}$$

$$a = O(\frac{1}{T}), \ \ \|\mu_i(t)\| = O(a) \ \forall i \in \mathcal{V}. \tag{4.6c}$$

where "a" is the root mean square (RMS) amplitude of the signal $\mu_i, i \in \mathcal{V}$.

Remark 4.4. Although we focus on deterministic signals, stochastic signals are also good candidates for perturbation [80, 81].

Lemma 4.2 (Approximate Gradient System)**.** *Consider the following system*

$$\dot{\theta} = -\delta[\psi(\theta + \mu) + C] \otimes \mu \tag{4.7}$$

where $\psi : \mathcal{R}^N \to \mathcal{R}$ is a $\mathcal{C}^2$ function, C is some constant and μ is the orthogonal perturbation signals (cf. Definition 4.5). Then, for any given arbitrary compact

[7] Here "T" refers to the least common multiple of the time periods of the perturbation signal.

domain Ω and sufficiently small δ and a, the system can be rewritten as a gradient system with perturbation as follows:

$$\dot{\theta}^{av} = -a^2\delta\left[\nabla\psi(\theta^{av}) + O(a+\delta)\right] \tag{4.8}$$

where ∇ denotes the differential operator.

Proof: See Appendix A.

Remark 4.5. The correlation is made between the performance function and the local perturbation signal, leading to local learning. This, in fact, offers a natural way for distributed implementation.

4.3.2 Dynamic average consensus

Different from conventional consensus, dynamic average consensus is trying to track in real-time the average of input references [82, 83]. The dynamic average consensus protocol for each agent $i \in \mathcal{V}$ is designed as follows (cf. Section 3.2):

$$\dot{y}_i = -\beta \sum_{j\in\mathcal{N}_i} l_{ij} y_j + \dot{z}_i \tag{4.9}$$

where y_i is the introduced auxiliary variable for agent i and l_{ij} is the weight that agent i use to incorporate the data from agent $j \in \mathcal{N}_i$ with $\mathcal{N}_i$ denoting the index set of the neighbors of agent i, or, in another form[8]:

$$\begin{aligned} \dot{\zeta}_i &= -\beta \sum_{j\in\mathcal{N}_i} l_{ij} y_j \\ y_i &= \zeta_i + z_i \end{aligned} \tag{4.10}$$

which can be further written in a compact form for the whole system as follows:

$$\begin{aligned} \dot{\zeta} &= -\beta L(\zeta + z) \\ y &= \zeta + z \end{aligned} \tag{4.11}$$

where z is the exogenous reference input of each agent.

[8]This form is different from (4.9) in that it can account for non-differentiable reference inputs.

Lemma 4.3 (Conservation Property I)**.** *Consider the system (4.11). Let Assumption 4.3 hold. Then,* $\sum_{i\in\mathcal{V}} y_i(t) - \sum_{i\in\mathcal{V}} z_i(t) = C,\ \forall t \geq 0$, *where* $C = \sum_{i\in\mathcal{V}} \zeta_i(0)$.

By pre-multiplying both sides of (4.11) by $\mathbf{1}^T$ and knowing that $\mathbf{1}^T L = 0$, we can immediately obtain the result since $\sum_{i\in\mathcal{V}} \zeta_i(t) = \sum_{i\in\mathcal{V}} \zeta_i(0)$ and $\sum_{i\in\mathcal{V}} y_i(t) = \sum_{i\in\mathcal{V}} \zeta_i(t) + \sum_{i\in\mathcal{V}} z_i(t),\ \forall t \geq 0$. This lemma essentially tells us that the instantaneous sum of the state of all agents is always the same as that of the reference inputs. This conservation property is important in that it allows us to track the time-varying average of the reference inputs.

4.3.3 The general overall scheme

The D-SPA scheme intertwines the above simultaneous perturbation technique and dynamic average consensus protocol by introducing certain auxiliary variables. Figure 4.1 shows the basic scheme for a static reference-to-output map. The corre-

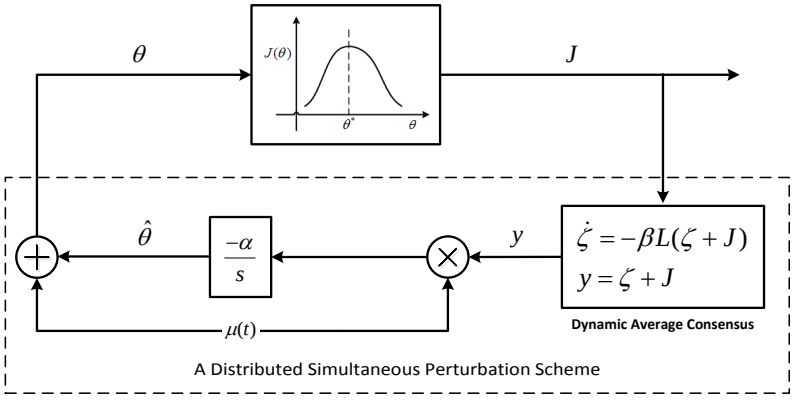


FIGURE 4.1: A scheme of distributed simultaneous perturbation approach

sponding continuous-time dynamic system of the scheme is given as follows:

$$\dot{\zeta} = -\beta L(\zeta + J(\hat{\theta}(t) + \mu(t))) \tag{4.12a}$$

$$\dot{\hat{\theta}} = -\alpha(\zeta + J) \odot \mu(t) \tag{4.12b}$$

where β and α are both $O(1)$ positive constants which encode the speed of the average-consensus process and the optimum seeking process respectively, $\hat{\theta}$ the estimated optimum, $\zeta = [\zeta_1, \zeta_2, ..., \zeta_m]^T$ the introduced auxiliary variable, $J =$

$[J_1, J_2, ..., J_m]^T$ the cost[9] of each subsystem, μ the orthogonal perturbation signal (cf. Definition 4.5) and L the Lapalacian Matrix.

4.4 Specific Schemes and Stability Analysis

4.4.1 Basic scheme

In this section, we establish the stability property of the proposed scheme taking into account the physical dynamics (4.1). We first provide the stability analysis based on the simplified version of the scheme that do not have high and low pass filters involved in the scheme as shown in Figure 4.2.

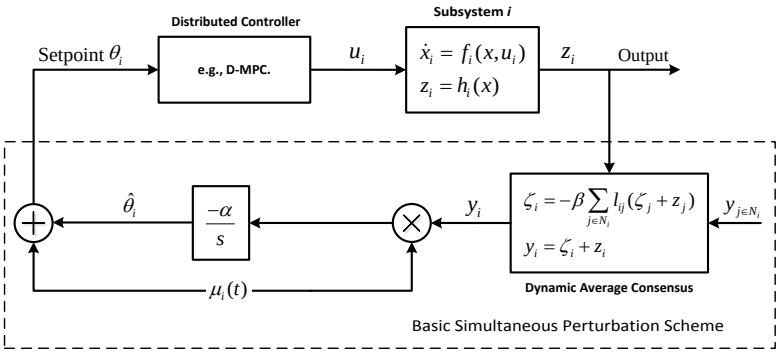


FIGURE 4.2: A basic scheme of distributed simultaneous perturbation approach

The overall dynamics corresponding to the basic scheme can be depicted as follows:

$$\begin{cases} \dot{x} &= f(x, \varphi(x, \hat{\theta} + \mu)) \\ \dot{\zeta} &= -\beta L(\zeta + h(x)) \\ \dot{\hat{\theta}} &= -\alpha(\zeta + h(x)) \odot \mu \end{cases} \tag{4.13}$$

Before preceding to our main results, we give the following lemma which guarantees the preservation of stability (cf. footnote 3) between a parameterized system and its corresponding nominal system.

[9]The cost refers to the one induced by the operation of the system working at steady-state.

Lemma 4.4. *Consider a parameterized system $\dot{x} = \phi(t, x, \varepsilon)$, where $\phi : \mathcal{R}_{\geq 0} \times \mathcal{R}^n \times \mathcal{R} \to \mathcal{R}^n$ and its partial derivatives with respect to (x, ε) are locally Lipschitz in $\mathcal{R}^n \times \mathcal{R}$, uniformly in t. Suppose that the nominal system $\dot{x} = \phi(t, x, 0)$ is uniformly globally asymptotically stable (UGAS), then the original system is USPAS on ε.*

Proof: See Appendix A.

Theorem 4.1. *Suppose that Assumptions 4.3, 4.4 and 4.5 hold.Let $\alpha = \alpha_K w \delta$ and $\delta = O(w)$, where $w = \frac{1}{T}$ is the base frequency and α_K is $O(1)$ positive constant. Then, the system (4.12) is USPAS on $[a]$ with respect to the Pareto-optimum of the DOCP problem, where 'a' is the RMS amplitude of the perturbation signal.*

Proof. Consider the dynamic system (4.12). Let $\tau = wt$ be the new time variable and $\bar{\mu}(\tau) = \mu(t)$ the scaled signal with unit time period. Since $\alpha = \alpha_K w \delta$, (4.12) can be rewritten as follows

$$w \frac{d\zeta}{d\tau} = -\beta L(\zeta + J(\hat{\theta} + \bar{\mu}(\tau))) \tag{4.14a}$$

$$\frac{d\hat{\theta}}{d\tau} = -\delta \alpha_K (\zeta + J(\hat{\theta} + \bar{\mu}(\tau))) \odot \bar{\mu}(\tau) \tag{4.14b}$$

When w is small, the above system exhibits a two-time-scale structure and is thus ready to be analyzed using singular perturbation theory [77]. Letting $y = \zeta + J(\hat{\theta} + \bar{\mu})$ be the new state and knowing that $\bar{\mu}$ is a $\mathcal{C}^1$ function, (4.14a) can be rewritten as (note that $\tau = wt$)

$$\frac{dy}{dt} = -\beta L y + w \nabla J(\hat{\theta} + \bar{\mu}) \cdot \left(\frac{d\hat{\theta}}{d\tau} + \frac{d\bar{\mu}(\tau)}{d\tau} \right) \tag{4.15}$$

where ∇J denotes the derivative of $J(\cdot)$. Setting $w = 0$ gives the following boundary-layer system:

$$\frac{dy}{dt} = -\beta L y \tag{4.16}$$

Let $V_1 = \frac{1}{2} y^T y$ be the Lyapunov function of the system. Then, we have

$$\dot{V}_1 = -\beta y^T L y = -\frac{\beta}{2}(y^T L y + y^T L^T y)$$

$$\begin{cases} \leq -\beta \lambda_2(L_s) \|y\| < 0, & \forall y \in \{y | Ly \neq 0\} \\ = 0, & \forall y \in \{y | Ly = 0\} \end{cases}$$

where $L_s = \frac{L+L^T}{2}$ and $\lambda_2(L_s)$ is the algebraic connectivity of the communication graph which is positive by Assumption 4.3 [24]. Hence, by Lasalle's theorem, the system will exponentially converge to the invariant manifold $\{y|Ly = 0\}$, uniformly in $\hat{\theta} + \bar{\mu}$. In addition, according to Conservation Property I in Lemma 4.3, we have

$$\sum_{i=1}^{N} y_i = \sum_{i=1}^{N} J_i(\hat{\theta} + \bar{\mu}) + C$$

where $C = \sum_{i=1}^{N} y_i(0)$. Thus, let $\bar{y} = \frac{\mathbf{1}^T y}{N}$ and $\bar{J} = \frac{\mathbf{1}^T J}{N}$ be the mean value of the entries of y and J respectively and $\bar{C} = \frac{\mathbf{1}^T y(0)}{N}$, we obtain the invariant manifold (i.e., the unique isolated root)

$$y^\star = \bar{y} \otimes \mathbf{1} = \left(\bar{J}(\hat{\theta} + \bar{\mu}) + \bar{C}\right) \otimes \mathbf{1} \tag{4.17}$$

on which the motion is carried out following the slow dynamics described by the reduced model (4.14b).

Now, let us consider the reduced system:

$$\frac{d\hat{\theta}}{d\tau} = -\delta\alpha_K \left[\left(\bar{J}(\hat{\theta} + \bar{\mu}) + \bar{C}\right) \otimes \mathbf{1}\right] \odot \bar{\mu}$$

which is equivalent to

$$\frac{d\hat{\theta}}{d\tau} = -\delta\alpha_K \left(\bar{J}(\hat{\theta} + \bar{\mu}) + \bar{C}\right) \otimes \bar{\mu}. \tag{4.18}$$

Invoking Lemma 4.2 and knowing that $\delta = O(w)$ and $a = O(w)$ (cf. Definition 4.5), we obtain the following averaged model in $\sigma = a^2\delta\tau$ time scale:

$$\frac{d\hat{\theta}^{av}}{d\sigma} = -\alpha_K \left(\nabla\bar{J}(\hat{\theta}^{av}) + O(w)\right) \tag{4.19}$$

Let $V_2 = \bar{J}(\hat{\theta}^{av}) - \bar{J}(\theta^\star)$, where $\theta^\star$ is the Pareto optimum of the DOCP problem, be the Lyapunov function for the nominal gradient system $\dot{\hat{\theta}}^{av} = -\alpha_K \nabla\bar{J}(\hat{\theta}^{av})$. Then, we have

$$\dot{V}_2 = \nabla\bar{J}(\hat{\theta}^{av})\dot{\hat{\theta}}^{av} = -\alpha_K \left\|\nabla\bar{J}(\hat{\theta}^{av})\right\|^2 \leq 0$$

with equality if only if $\nabla\bar{J}(\hat{\theta}^{av}) = 0$. Thus, by Assumption 4.4 and Lasalle's theorem, the nominal system of (4.19) is UGAS. In addition, knowing that $\bar{J}$ is a $\mathcal{C}^2$ function from Assumption 4.4, by Lemma 4.4, we claim that the original

averaged system (4.19) is USPAS on $[w]$ and so is the original reduced system (4.18). Further, since the boundary-layer system (4.16) is GAS, uniformly in $\hat{\theta}+\bar{\mu}$ and the dynamic functions (4.15) and (4.14b) as well as their first derivatives are locally Lipschitz in $(\hat{\theta}, y, w)$, it follows from Lemma 4.1 that the system (4.12) is USPAS on $[w]$ and, recalling that $a = O(w)$, is thus also USPAS on $[a]$ with respect to the Pareto-optimum of the DOCP problem. □

Remark 4.6. It is important to note that the stability result obtained using singular perturbation techniques does not provide much guidance in improving the convergence performance of the scheme. For instance, a proper tuning of certain parameters, such as "α", will increase the speed of convergence [34].

Remark 4.7. Note that the assumption that the feedback gain α is chosen to be the same for all subsystems is merely for simplicity. Indeed, it is not difficult to show that the scheme can still be guaranteed to converge to the neighborhood of Pareto-optimum with different feedback gains for different agents, which is crucial considering the heterogeneous property of subsystems in practice, making it possible for asynchronous implementation as well.

Theorem 4.2. *Suppose all the assumptions of theorem 4.1 hold. In addition, Assumptions 4.1 and 4.2 are satisfied. Let $\alpha = \alpha_K w\delta$ and $\delta = O(w)$. Then, the dynamic system (4.1), under the proposed basic D-SPA scheme, is USPAS on $[a]$ with respect to the Pareto-optimum of the DOCP problem.*

Proof. Consider the overall system (4.13). Let $\tau = wt$ be the new time variable, $\bar{\mu}(\tau) = \mu(t)$ the scaled signal and $y = \zeta + h(x)$ the new state. Since $\alpha = \alpha_K w\delta$, the overall system (4.13) can be rewritten in singularly perturbed form as follows:

$$w\frac{d}{d\tau}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f(x, \varphi(x, \hat{\theta}+\bar{\mu}(\tau))) \\ -\beta L y + \nabla h \cdot f \end{bmatrix} \tag{4.20a}$$

$$\frac{d\hat{\theta}}{d\tau} = -\delta\alpha_K y \odot \bar{\mu}(\tau) \tag{4.20b}$$

where ∇h denotes the derivative of $h(\cdot)$. Then, let us first consider the following boundary-layer system:

$$\frac{dx}{dt} = f(x, \varphi(x, \hat{\theta}+\bar{\mu}(\tau))) \tag{4.21a}$$

$$\frac{dy}{dt} = -\beta L y + \nabla h(x) \cdot f(x, \phi(x, \hat{\theta}+\bar{\mu}(\tau))) \tag{4.21b}$$

Similar with the proof in Theorem 4.1, we can show that the nominal system $\frac{dy}{dt} = -\beta L y$ is exponentially stable with respect to the isolated root of (4.21b), i.e., $y^\star = [\bar{h}(x) + \bar{C}] \otimes \mathbf{1}$, where $\bar{h} = \frac{\mathbf{1}^T h}{N}$. Since h and f are both $\mathcal{C}^1$ functions, $\nabla h \cdot f$ is bounded for any given bounded $x - x^\star$, where $x^\star = l(\hat{\theta} + \bar{\mu})$ is the isolated root of (4.21a). Thus, it is not difficult to show that the subsystem (4.21b) is, with $x - x^\star$ viewed as the input, input-to-state stable (ISS) [77, Lem. 4.6]. In addition, by Assumption 4.2, we know that the physical system (4.21a) is GAS, uniformly in $\hat{\theta} + \bar{\mu}$, it follows that the system (4.21) is GAS, uniformly in $\hat{\theta} + \bar{\mu}$ [77, Lem. 4.7].

Then, "freezing" $[x, y]^T$ at its equilibrium leads to the following reduced system (Recall that $\bar{J} = \bar{h} \circ m$):

$$\frac{d\hat{\theta}}{d\tau} = -\delta\alpha_K \left[\left(\bar{J}(\hat{\theta} + \bar{\mu}) + \bar{C} \right) \otimes \mathbf{1} \right] \odot \bar{\mu} \tag{4.22}$$

which is already shown from Theorem 4.1 to be USPAS on $[w]$. Further, it is not difficult to show that the dynamic functions in (4.20) and their first derivatives are locally Lipschitz in $(x, y, \hat{\theta}, w)$. Thus, by Lemma 4.1, we conclude that the system (4.13) is USPAS on $[w]$ and, recalling that $a = O(w)$, is thus also USPAS on $[a]$ with respect to the Pareto-optimum of the DOCP problem.

4.4.2 High-order scheme

In this section, we will introduce the high-order simultaneous perturbation scheme which incorporates the low pass and high pass filters as shown in Figure 4.3.

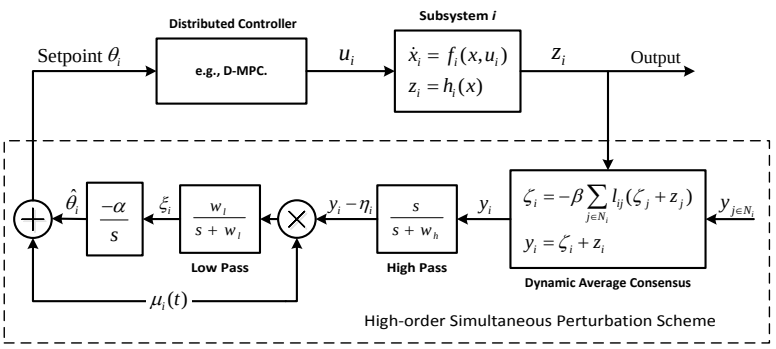


FIGURE 4.3: A high-order scheme of distributed simultaneous perturbation approach

The overall dynamics of the high-order scheme can be depicted as follows:

$$\begin{cases} \dot{x} &= f(x, \varphi(x, \hat{\theta} + \mu)) \\ \dot{\zeta} &= -\beta L(\zeta + h(x)) \\ \dot{\hat{\theta}} &= -\alpha\xi \\ \dot{\xi} &= -w_l\xi + w_l(\zeta + h(x) - \eta) \odot \mu \\ \dot{\eta} &= -w_h\eta + w_h(\zeta + h(x)) \end{cases} \tag{4.23}$$

where w_l and w_h denote the cut-off frequencies of the low pass filter and the high pass filter respectively. As we will show in our proof that the filters introduced in the scheme will not impact the stability of the system but enhance the convergence performance, e.g., attenuating the output oscillation [34].

Theorem 4.3. *Suppose all the assumptions of theorem 4.2 hold. Let $w_l = w_L w\delta$, $w_h = w_H w\delta$, $\alpha = \alpha_K w\delta$ and $\delta = O(w)$, where w_L, w_H and α_K are $O(1)$ positive constants. Then, the system (4.1), under the proposed high-order D-SPA scheme, is USPAS on $[a]$ with respect to the Pareto-optimum of the DOCP problem.*

Proof. Consider the dynamic system (4.23). Let $\tau = wt$ be the new time variable, $\bar{\mu}(\tau) = \mu(t)$ the scaled signal and $y = \zeta + h(x)$ the new state. Since $w_l = w_L w\delta, w_h = w_H w\delta, \alpha = \alpha_K w\delta$, (4.23) can be rewritten in singularly perturbed form as follows:

$$w\frac{d}{d\tau}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f(x, \varphi(x, \hat{\theta} + \bar{\mu}(\tau))) \\ -\beta L y + \nabla h \cdot f \end{bmatrix} \tag{4.24a}$$

$$\frac{d}{d\tau}\begin{bmatrix} \hat{\theta} \\ \xi \\ \eta \end{bmatrix} = \delta \begin{bmatrix} -\alpha_K\xi \\ -w_L\xi + w_L(y - \eta) \odot \bar{\mu} \\ -w_H(\eta - y) \end{bmatrix} \tag{4.24b}$$

Thus, fixing $[x, y]^T$ at its equilibrium leads to the following reduced system:

$$\frac{d}{d\tau}\begin{bmatrix} \hat{\theta} \\ \xi \\ \eta \end{bmatrix} = \delta \begin{bmatrix} -\alpha_K\xi \\ -w_L\xi + w_L\left((\bar{J} + \bar{C}) \otimes \mathbf{1} - \eta\right) \odot \bar{\mu} \\ -w_H\eta + w_H\left(\bar{J}(\hat{\theta} + \bar{\mu}) + \bar{C}\right) \otimes \mathbf{1} \end{bmatrix} \tag{4.25}$$

Using the same technique as in the proof of Lemma 4.2 and knowing that $\int_0^1 \eta \odot$

$\bar{\mu}(\sigma)d\sigma = 0$ and $\int_0^1 \bar{J}(\hat{\theta} + \bar{\mu}(\sigma)) \otimes \mathbf{1} d\sigma = \bar{J}(\hat{\theta}) \otimes \mathbf{1} + O(a^2)$ by first-order approximation, it is not difficult to show that the above system (4.25) can be represented as an averaged system with perturbation:

$$\frac{d}{d\tau}\begin{bmatrix} \hat{\theta}^{av} \\ \xi^{av} \\ \eta^{av} \end{bmatrix} = \delta \begin{bmatrix} -\alpha_K \xi^{av} \\ -w_L \xi^{av} + w_L a^2 \left(\nabla \bar{J}(\hat{\theta}^{av}) + r \right) \\ -w_H \eta^{av} + w_H \left(\bar{J}(\hat{\theta}^{av}) + \bar{C} \right) \otimes \mathbf{1} + r \end{bmatrix} \tag{4.26}$$

where $r = O(w)$ is the perturbation term obtained by knowing $a = O(w)$, $\delta = O(w)$. The above nominal system can be rewritten in $\sigma = \delta\tau$ time scale as follows:

$$\frac{d}{d\sigma}\begin{bmatrix} \hat{\theta}^{av} \\ \xi^{av} \end{bmatrix} = \begin{bmatrix} -\alpha_K \xi^{av} \\ -w_L \xi^{av} + w_L a^2 \nabla \bar{J}(\hat{\theta}^{av}) \end{bmatrix} \tag{4.27a}$$

$$\frac{d\eta^{av}}{d\sigma} = -w_H \eta^{av} + w_H \left(\bar{J}(\hat{\theta}^{av}) + \bar{C} \right) \otimes \mathbf{1} \tag{4.27b}$$

Let $V_3 = \bar{J}(\hat{\theta}^{av}) - \bar{J}(\theta^\star) + \frac{1}{2}\frac{\alpha_K}{w_L a^2} \xi^{avT} \xi^{av}$ be the Lyapunov function for the subsystem (4.27a). Taking the derivative along the trajectory yields

$$\begin{aligned} \dot{V}_3 &= -\alpha_K \nabla \bar{J}(\hat{\theta}^{av})^T \xi^{av} + \frac{\alpha_K}{a^2} \xi^{avT}(-\xi^{av} + a^2 \nabla \bar{J}(\hat{\theta}^{av})) \\ &= -\frac{\alpha_K}{a^2} \xi^{avT} \xi^{av} \leq 0 \end{aligned}$$

with equality if only if $\xi^{av} = 0$. Thus, by Assumption 4.4 and Lasalle's theorem, the system (4.27a) is UGAS. In addition, letting $\tilde{\eta}^{av} = \eta^{av} - (\bar{J}(\theta^\star) + \bar{C}) \otimes \mathbf{1}$, the system (4.27b) can be rewritten as

$$\frac{d\tilde{\eta}^{av}}{d\tau} = -w_H \tilde{\eta}^{av} + w_H \left(\bar{J}(\hat{\theta}^{av}) - \bar{J}(\theta^\star) \right) \otimes \mathbf{1} \tag{4.28}$$

Since $\bar{J}$ is a $\mathcal{C}^2$ function thus locally Lipschitz in its argument, the value of $\bar{J}(\hat{\theta}^{av}) - \bar{J}(\theta^\star)$ is bounded for any given bounded "$\hat{\theta}^{av} - \theta^\star$". It is obvious that the origin of the unforced system $\frac{d\tilde{\eta}^{av}}{d\tau} = -w_H \tilde{\eta}^{av}$ is globally exponentially stable, thus we claim that the nominal subsystem (4.27b), with "$\hat{\theta}^{av} - \theta^\star$" viewed as input, is ISS [77, Lem. 4.6]. Further, recalling that the system (4.27a) is UGAS, it follows that the nominal averaged system (4.27) is UGAS [77, Lem. 4.7] .

Moreover, since $\bar{J}$ is a $\mathcal{C}^2$ function, by Lemma 4.4, we claim that the original averaged system (4.26) is USPAS on $[w]$ and so is the original (reduced) system (4.25).

As shown before, according to Assumption 4.2, the boundary-layer system (4.24a) is GAS with respect to the isolated root $x^\star = l(\hat{\theta}+\bar{\mu}), y^\star = [\bar{J}(\hat{\theta}+\bar{\mu})+\bar{C}]\otimes\mathbf{1}$, uniformly in $\hat{\theta}+\bar{\mu}$. Further, it is not difficult to show that the functions in (4.24) are locally Lipschitz in their arguments. Thus, by Lemma 4.1, we conclude that the system (4.1) under the high-order D-SPA scheme (4.23) is USPAS on $[w]$ and, recalling that $a = O(w)$, is thus also USPAS on $[a]$ with respect to the Pareto-optimum of the DOCP problem. □

Remark 4.8. As we will see in the proof of the above theorem, there are multiple time-scales involved in the above dynamic system (4.23). Specifically, we have the following time-scales:

1) *fastest*: the plant and the dynamic average consensus filter;

2) *medium*: the probing frequencies;

3) *slow*: the high/low pass filters in the scheme.

Remark 4.9. The interplay of several perturbation parameters is very complicated and the result is very limited especially for semi-globally practically asymptotically stable systems [34]. To circumvent this difficulty, we introduce the conditions $a = O(w)$ and $\delta = O(w)$ in the stability analysis to make the system amenable to using techniques under the framework of scalar perturbation.

4.4.3 Distributed extremum seeking control

In this section, we show that when we choose sinusoidal signal as the perturbation signal, which is the most popular choice, we can come up with a distributed version of multivariable extremum seeking control [16, 35], termed D-ESC. In particular, we set $\mu(t) = a[\sin(w_1 t), \sin(w_2 t), ..., \sin(w_m t)]^T$ and, then, we can easily show that these perturbation signals are mutually orthogonal as follows:

$$\delta_{jk} = \frac{w_g}{2\pi}\int_0^{\frac{2\pi}{w_g}} \sin(w_j t)\sin(w_k t)\mathrm{d}t = \begin{cases} 0, & w_j \neq w_k, \\ \frac{1}{2}, & w_j = w_k. \end{cases} \quad \forall i,k \in \mathcal{V}. \tag{4.29}$$

where w_g is the greatest common divisor of all the frequencies w_i of chosen signals.

Remark 4.10. In the above analysis, we only consider first-order approximation of the gradient. However, if certain additional orthogonality conditions hold, higher-order approximation with more accuracy can be achieved [75].

4.5 Application to Wind Farm Systems

4.5.1 Dynamic modeling and wake effect of wind farms

We consider the control parameter of a wind turbine as the axial induction factor, the fractional decrease between the velocity of the upstream wind and downstream wind seen by the turbine, which can be controlled by properly tuning the blade pitch angle and the tip speed ratio of rotors as shown in Figure 4.4.

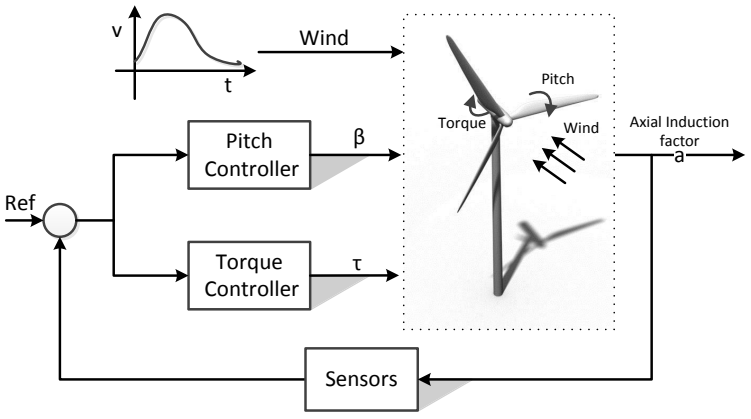


FIGURE 4.4: Illustration of the control of wind turbines.

The Park model is one of the most popular wake models which give the velocity profile of wind farm [84, 85]. Figure 4.5 illustrates the interaction between an upstream turbine T_1 and a downstream turbine T_2 operating in a free stream velocity V_∞. We can see from the figure that the axial induction factor a_1 of the upstream turbine is coupled into the power generation of the downstream turbine due to the wake effect induced by the overlapping area $A_{1\to 2}^{overlap}$ of these two wind turbines.

In particular, let us consider a wind farm system consisting of multiple wind turbines that are densely deployed. Then, according to the Park model, the effective wind velocity seen by turbine i can be calculated as

$$V_j(a) = V_\infty(1 - \delta V_j(a))$$

where $a = [a_1, a_2, ..., a_m]^T$ denotes the axial induction factors of all turbines and $\delta V_j(a)$ is the aggregated velocity deficit created by all the upstream turbines of

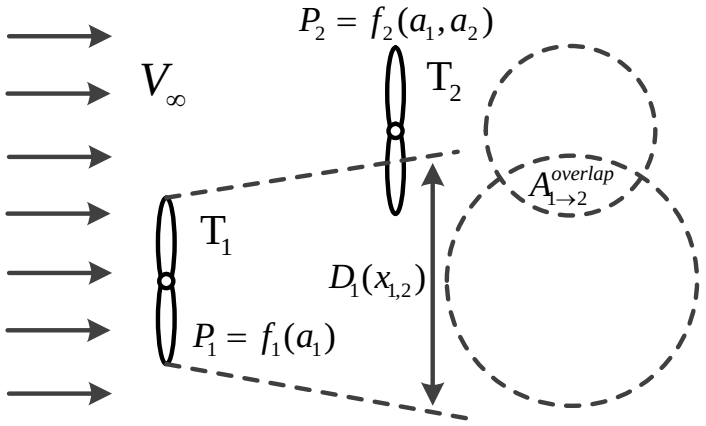


FIGURE 4.5: Illustration of the wake effect of wind turbines.

turbine i which can be expressed as

$$\delta V_j(a) = 2\sqrt{\sum_{i\in\mathcal{V}:x_j<x_i}\left(a_i\left(\frac{D_i}{D_i+2kx_{ij}}\right)^2\frac{A_{i\to j}^{overlap}}{A_i}\right)}$$

with A_i being the area swept by the blades of turbine i, $A_{i\to j}^{overlap}$ the area of overlap between the wake created by turbine j and the disc created by the blades of turbine i, x_{ij} the distance between turbine i and turbine j, k the roughness coefficient which can be determined empirically for specific environment and $D_i(x_{ij}) = D_i + 2kx_{ij}$ the diameter of the wake induced by turbine i at a distance x_{ij}.
The power generated by turbine i can be thus calculated as

$$P_i(a) = \frac{1}{2}\rho A_i C_P(a_i) V_i(a)^3$$

where ρ is the air density and C_P is the power efficiency coefficient which can be expressed as $C_P(a_i) = 4a_i(1-a_i)^2$.

4.5.2 Coordinated control of wind farm systems

We apply our proposed approach to a simulated wind farm system. Coordinated control of wind farms has been receiving considerable attention recently for energy maximization [84–87]. By taking into account the aerodynamic interaction

among wind turbines, we can have an energy gain of up to *25%* under certain wind conditions with respect to the greedy policy where each turbine operates individually [84]. In the simulation, we consider an offshore wind farm consisting of 4×3 wind turbines as illustrated in Figure 4.6.

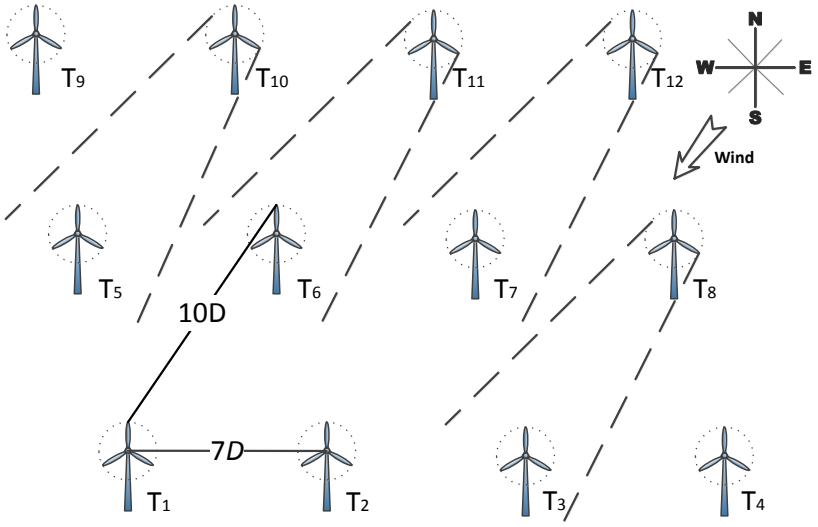


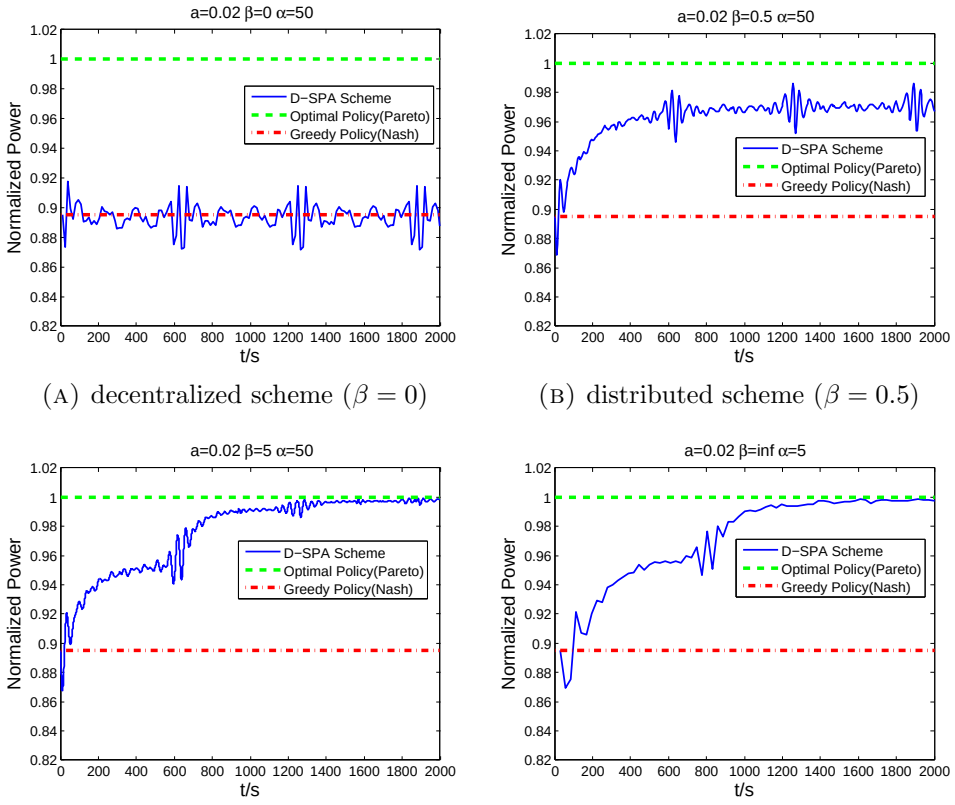
FIGURE 4.6: The layout of a wind farm consisting of 4x3 wind turbines. This layout resembles the Horns Rev wind farm constituted by Vestas V80 2MW turbines with a diameter D of 80 meters. The turbines are spaced evenly with an interval of 7 turbine diameters in north and east direction and 10 diameters in north-east direction.

We consider the quasi-steady-state model (cf. Equation (4.25)). That is, we do not consider the dynamics of wake traveling and wind turbines[10] but assume there is already a distributed controller properly designed to make sure the wind farm system is asymptotically stable for any given setpoint in certain domain (e.g., [0.1,0.5]) for axial induction factor [88]. In addition, we assume each turbine can communicate with its immediate neighboring turbines and the Laplacian matrix L underpinning the communication network is designed as follows[11]:

$$l_{ij} = \begin{cases} \frac{-1}{5}, & j \in \mathcal{N}_i \text{ and } j \neq i \\ \frac{|\mathcal{N}_i|}{5}, & j = i \end{cases} \tag{4.30}$$

[10]Note that the simulation time may thus not exactly represent the physical time.

[11]Note that in our simulation the dynamic average consensus filter is properly initialized to make sure its output be close to zero for better convergence performance.



(A) decentralized scheme ($\beta = 0$)

(B) distributed scheme ($\beta = 0.5$)

(C) distributed scheme ($\beta = 5$)

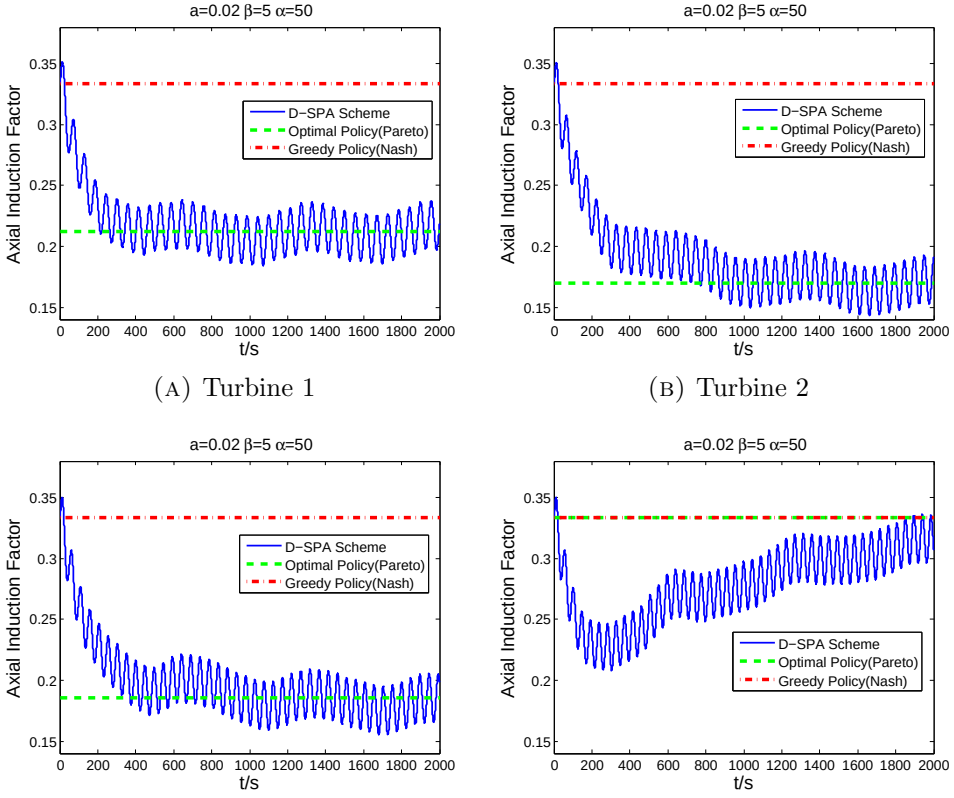
(D) centralized scheme ($\beta = \infty$)

FIGURE 4.7: Time History of the overall power generation of 4x3 windfarm under south wind with different speeds of the consensus process. It shows that there will be no coordination among turbines when $\beta = 0$, leading to Nash Equilibrium, while we can achieve Pareto optimum when the consensus process is instantaneous, corresponding to $\beta = \infty$. In practice, the outcome is somewhere in-between when β is certain positive constant.

The model parameters for the wind farm were set as $k = 0.04$, $\rho = 1.225(kg/m^3)$ and $U_\infty = 8(m/s)$ in all wind directions. In the D-SPA scheme, we chose sinusoidal signal as the perturbation signal (cf. Section 4.4.3) and the frequencies were designed as $[11, 12, ..., 22] \cdot 10^{-2}$ rad/s for each turbine respectively.

We used the high-order scheme with $w_l = w_h = 0.01$ rad/s in the simulation due to its better convergence performance[12]. The simulation was conducted for west, south and northeast wind conditions having the most potential of energy gain. To show the effectiveness of the proposed scheme, the results are compared with the greedy policy ($\theta = 1/3$) and optimal policy.

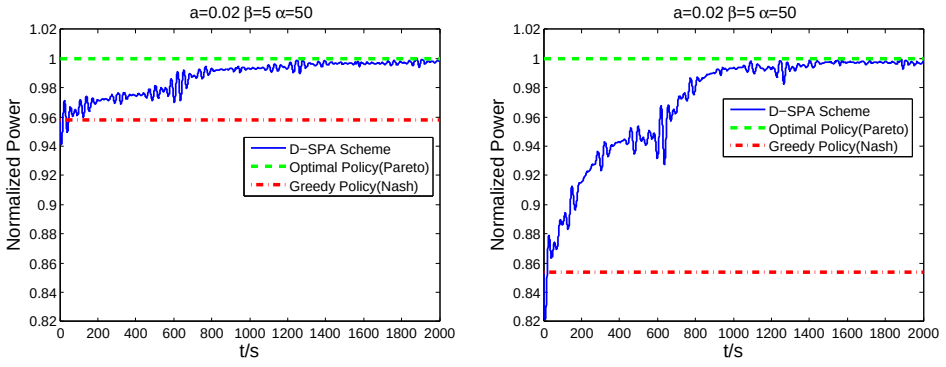
[12] As a rule of thumb, the cut-off frequency of high/low-pass filters is chosen around 1/10 of the least probing frequency.



(A) Turbine 1 (B) Turbine 2

(C) Turbine 3 (D) Turbine 4

FIGURE 4.8: Trajectories of the axial induction factor of Turbines 1, 2, 3 and 4 under west wind. It shows that each turbine manages to work at the best operating point corresponding to the optimal policy.



(A) Northeast Wind (B) West Wind

FIGURE 4.9: Time History of the overall power generation of the 4x3 windfarm under (a) north-east wind and (b) west wind. The figures show that the proposed scheme with the same parameter setting is able to deal with the topology change of interactions among wind turbines due to the change of wind direction.

Figure 4.7 demonstrates that the coordination level of turbines is dependent on the speed of the consensus process. The faster the consensus process being carried out, the higher is the achieved efficiency of the wind farm power generation. The outcome of the scenario without communication ($\beta = 0$) corresponding to greedy policy is of Nash Equilibrium while the exact Pareto optimum can be attained using optimal policy corresponding to the consensus process being instantaneous ($\beta = \infty$). However, in real applications, the outcome will be somewhere in between as it needs some time for the consensus process to converge. This illustrates that the communication part in networked control systems plays a key role not only for stability but also for optimization.

Figure 4.8 shows the trajectories of the axial induction factor of selected wind turbines when the wind farm is under west wind direction and controlled under the D-SPA scheme with relatively fast consensus process $\beta = 5$. As can be seen from the figure, each turbine is converging to the operating point corresponding to the optimal policy. However, there is still some gap due to the inability of the dynamic average process to track the varying perturbation signals in real time. This can be mitigated by choosing perturbation signals with low frequency but doing so will lead to larger oscillation in output [35].

Figure 4.9 illustrates the power efficiency achievement using the D-SPA scheme with the same parameter setting for north-east wind and south wind respectively. It essentially indicates that the proposed scheme does not require the information of the topology of aerodynamic interactions of wind turbines and thus is robust to the topology change due to the variation of the wind direction. However, it suffers from the drawback of slow convergence speed common to model-free approaches.

4.5.3 Comparisons with state-of-the-art techniques

The proposed approach belongs to the family of the "map-free" MPPT algorithms which can seek the optimum without knowing much knowledge of the reference-output map. Most of the conventional MPPT algorithms are centralized and not readily transferable to distributed version as the topology of the system will come into play. To the best of our knowledge, there are few distributed MPPT approaches in the existing literature. In particular, [85] proposed a GA-MMPT approach which requires one to explicitly deal with the topology. Thus, it is not

practical considering that the wind condition is changing constantly. Another approach (GT-MPPT) proposed by [84] is based on random search, which is already shown to converge slowly [85]. As with most ES-based approaches, our approach can be applied without modifications to the existing system so long as there exists a stabilizing control law [88]. In addition, since we use simultaneous perturbation technique, our algorithm does not need synchronization for implementation. In contrast, the conventional Perturb and Observe (P&O) MPPT algorithm should be properly synchronized and carried out in a predefined sequence (This somewhat resembles centralized information), making it complicated for practical implementation. Moreover, since we employ consensus mechanism for coordination, it will be more robust to the topology changes of the system as compared to the existing distributed MMPT approaches where only certain topology is considered and this is evident in our simulation for different wind directions.

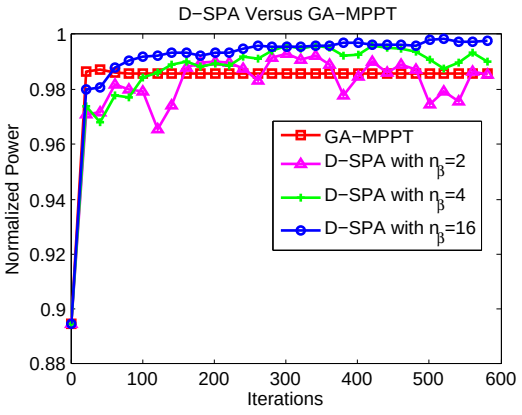


FIGURE 4.10: Evolution history of the normalized power generation of GA-MMPT (square) and D-SPA with three parameter settings: (1) $a = 0.02, \alpha = 50, n_\beta = 2$ (diamond) (2) $a = 0.02, \alpha = 50, n_\beta = 4$ (cross) and (3) $a = 0.02, \alpha = 50, n_\beta = 16$ (circle). It shows that the proposed D-SPA approach is able to obtain more wind energy so long as there is enough number of inner loops of consensus being carried out in each iteration.

Furthermore, we have also compared the performance with the existing state-of-the-art distributed MPPT approach (GA-MPPT) [85] in terms of convergence speed and accuracy. The simulation result is shown in Figure 4.10 which plots the normalized overall power generation of the wind farm with respect to iterations. In order to make a fair comparison, each round of the update of the control variables is regarded as an iteration and only a peer-to-peer network is allowed for information sharing. In addition, a discrete-time counterpart[13] of the proposed approach

[13] We choose $\Delta t = 0.1$ as the time interval for discretization.

is also developed (cf. Equation (4.31)) for comparison. It should be noted that the parameter β in the proposed continuous scheme will become the number of inner loop (denoted as n_β) of consensus being carried out within each step of perturbation in the discrete counterpart. The bigger the value of β, the higher the number of inner loops of consensus are executed in each iteration.

$$y(k+1) = W^{n_\beta}(y(k) + \Delta J(k)) \tag{4.31a}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \alpha \cdot (y(k+1) - y(k)) \odot \mu(k) \tag{4.31b}$$

where $\Delta J(k) = J(\hat{\theta}(k) + \mu(k)) - J(\hat{\theta}(k))$ is the difference between the power generation before and after perturbation, $\mu(k) = a[\sin(w_1 k), \sin(w_2 k), ..., \sin(w_m k)]^T$ the perturbation signals for each turbine, α the stepsize of gradient search, $y(k)$ the auxiliary variable and $W = I - 0.1L$ the weight matrix designed for average consensus. In our simulation, we consider the same wind farm system under South wind as before. The parameter setting for GA-MPPT is $K = 0.01$, which yields fast convergence without much overshooting. For our algorithm, we consider three scenarios with parameter settings: $a = 0.02, \alpha = 50, n_\beta = 2$, $a = 0.02, \alpha = 50, n_\beta = 4$ and $a = 0.02, \alpha = 50, n_\beta = 16$ respectively. The probing frequencies of the perturbation signals are chosen as the same with the previous continuous example for all scenarios. It follows from Figure 4.10 that GA-MPPT has a slightly faster and smoother convergence property due to the simplicity of its perturbation technique resulting in more accurate gradient estimation. However, as mentioned above, this requires perfect synchronization among modules which is not practical or requires "centralized control". We can also see from the figure that it will get stuck at a distance away from the optimal point, which is expected as it does not incorporate the information of power generation of multi-hop neighboring turbines on which they have impact. In contrast, our approach is able to extract more wind energy so long as enough number of inner loops of consensus are carried out in each iteration.

4.6 Summary

In this chapter, we have proposed a distributed simultaneous perturbation approach for optimizing the steady-state performance of large-scale networked dynamic systems. The approach employs the simultaneous perturbation technique as well as a consensus mechanism, yielding a distributed model-free technique which has the

potential to accommodate slowly changing environments. We have also proved the convergence of the scheme to the neighborhood of Pareto optimum using singular perturbation and averaging techniques. In the simulation of coordinated control of a wind farm system and the comparison with existing state-of-art distributed MPPTs, we have verified that, with proper tuning of the parameters under design, the proposed scheme is able to improve the energy efficiency to the extent that increases with the speed of the consensus process. The proposed scheme is expected to be applicable to other large-scale dynamic systems for which the reference-to-output map is hard to obtain and distributed implementation is necessary.

Part II

Coordinated Estimation

Chapter 5

Distributed Optimization in Sensor Networks: Fixed Networks and Synchronous Implementation

This chapter deals with the distributed estimation problem in large-scale sensor networks where the network is fixed and the algorithm is synchronous. Two basic algorithms are proposed in this chapter to solve the problem. We state the general problem of our interest in Section 5.1 and provide the preliminaries that are crucial in developing the proposed algorithms in Section 5.2. We then investigate the convergence performance of the proposed algorithms in Section 5.3 and 5.4 and finally apply the algorithms to sensor fusion problems in Section 5.5.

5.1 Problem Statement

We consider the distributed estimation problem involved in large-scale sensor networks where a large number of sensors are collaborating with each other to estimate certain parameters, e.g., temperature of a room or position of a source. This kind of estimation problem can be formulated as the following optimization problem

which, indeed, is equivalent to the problem (DOP):

$$\begin{aligned} \min_{x \in \mathcal{R}^{md}} f(x) &= \sum_{i=1}^{m} f_i(x_i) \\ \mathbf{s.t.}\ x_i &= x_j,\ \forall i, j \in \mathcal{V} \end{aligned} \tag{EDOP}$$

where $x_i \in \mathcal{R}^d$ is the local estimate of agent i about the global optimum $\theta^\star$ while $x = [x_1^T, x_2^T, ..., x_m^T]^T \in \mathcal{R}^{m \times d}$ is the collection of the estimates of all agents and $f_i : \mathcal{R}^d \to \mathcal{R}$ is the local objective function known only to agent i.

For the EDOP problem to be feasible, we make the following assumption on the existence of the optimal solution:

Assumption 5.1. There exists an optimum $x^\star = \mathbf{1} \otimes \theta^\star$ to the EDOP problem such that $f^\star := f(x^\star) = \min_{\theta \in \mathcal{R}^d} F(\theta)$.

Remark 5.1. In the sequel, we only consider the case $d = 1$ since the analysis for the case $d > 1$ is similar except for that we need to deal with Kronecker product and the results developed can be easily extended to multi-dimensional cases. In addition, once it is clear, we will suppress the subscript of variables for brevity.

5.2 Preliminaries

5.2.1 Monotone operator, saddle point and Fenchel's dual

An operator T on an Euclidean Space $\mathcal{H}$ is a set-valued mapping, i.e., $T : \mathcal{H} \to 2^{\mathcal{H}}$ and the graph of T is defined as gra $T = \{(x, y) \in \mathcal{H} \times \mathcal{H} | y \in Tx\}$.

Definition 5.1. An operator T is said to be monotone if

$$\forall (x, y), (x', y') \in \text{gra } T \qquad \langle x - x', y - y' \rangle \geq 0,$$

and strongly monotone if

$$\forall (x, y), (x', y') \in \text{gra } T \qquad \langle x - x', y - y' \rangle \geq \alpha \left\| x - x' \right\|^2.$$

A monotone operator is called maximal if there is no monotone operator T' such that gra $T \subset$ gra T', e.g., ∂f of a function $f \in \Gamma(\mathcal{H})$ is maximally monotone.

Further, the resolvent of T is defined as $J_{\tau T} = (I + \tau T)^{-1}$.

Definition 5.2 (Proximity Operator)**.** A resolvent operator with $\tau > 0$ is called as the proximity operator of a convex functional f and is defined as follows:

$$\mathbf{prox}_{\tau f}(v) = \arg\min_{x \in \mathcal{H}} \{f(x) + \frac{1}{2\tau} \|x - v\|^2\}.$$

Remark 5.2. The proximity operator plays a key role in proximal point algorithms which subsume many well known algorithms, such as the well-known ADMM and augmented Lagrangian method.

Definition 5.3 (Saddle Point)**.** A pair $(x^\star, y^\star)$ is said to constitute a saddle point if the following condition holds

$$\psi(x^\star, y) \leq \psi(x^\star, y^\star) \leq \psi(x, y^\star), \forall (x, y) \in \mathcal{D} \tag{5.1}$$

where $\psi(x, y)$ is the Lagrangian.

Definition 5.4 (Fenchel's dual)**.** Let $f \in \Gamma(\mathcal{H}) : \mathcal{H} \to \mathcal{R} \cup \{\pm\infty\}$ be a convex functional. Then, its convex conjugate (Fenchel's dual) f^* is defined as follows:

$$f^*(y) := \sup_{x \in \mathcal{H}} \{\langle x, y \rangle - f(x)\}$$

where y is the dual variable.

5.2.2 Bregman distance and G-space

Definition 5.5 (Bregman Distance)**.** Consider a convex functional $f : \mathcal{H} \to \mathcal{R} \cup \{\pm\infty\}$, the Bregman distance between $x \in \mathcal{H}$ and $x' \in \mathcal{H}$ is defined as

$$D_f^q(x, x') = f(x) - f(x') - \langle q, x - x' \rangle$$

where $q \in \partial f(x')$ is a subgradient evaluated at x'.

Bregman distance has three basic properties: $D(x, y) \geq 0$, $D(x, y) \neq D(y, x)$ and $D(y, x) \geq D(z, x)$ if $z \in [x, y]$

Remark 5.3. The Bregman distance is not the distance in usual sense but performs a similar function as distance, e.g., in the Bregman proximal point algorithm, where the Euclidean distance is replaced with Bregman distance.

Definition 5.6 (G-space and Induced Norm)**.** Given a symmetric positive definite matrix G, a G-space as well as its induced norm are defined as follows: $\langle x, x' \rangle_G = \langle Gx, x' \rangle$ and $\|x\|_G = \sqrt{\langle Gx, x \rangle}$, $\forall x, x' \in \mathcal{H}$.

5.2.3 Some basic relations

The following relations will be frequently used and crucial to our subsequent proofs in the convergence analysis in Section 5.4.2 and Section 6.4.3.

Proposition 5.1. *Consider two vectors $x, y \in \mathcal{R}^m$ and an orthogonal projection matrix $Q \in \mathcal{R}^{m \times m}$. Let $\bar{z} = \frac{\mathbf{1}\mathbf{1}^T}{m} z$ and $\tilde{z} = z - \bar{z}$, $z = x, y$. Then, we have*

(i) $x^T \bar{y} = \bar{x}^T \bar{y}$;

(ii) $x \odot y - \bar{x} \odot \bar{y} = x \odot \tilde{y} + \tilde{x} \odot \bar{y}$;

(iii) $\overline{x \odot y} = \bar{x} \odot \bar{y} + \overline{\tilde{x} \odot \tilde{y}}$;

(iv) $\frac{1}{\sqrt{m}}(\|\bar{x}\| \, \|\bar{y}\| - \|\tilde{x}\| \, \|\tilde{y}\|) \leq \|\overline{x \odot y}\| \leq \frac{1}{\sqrt{m}} \|x\| \, \|y\|$;

(v) $Q^2 = Q$ *and* $\|Qx\| \leq \|x\|$.

Proof. See Appendix B.

5.3 Distributed Bregman Forward-Backward Splitting Algorithm

We present a distributed algorithm, termed Distributed Forward-Backward Bregman Splitting (D-FBBS), to solve the above EDOP problem as well as its dual. We first introduce the Bregman Iterative Regularization as the basis of developing the algorithm. The forward-backward splitting technique is then employed to split the optimization problem, leading to a separated one which can be solved more efficiently than the original combined one in a distributed way with much cheaper computation and less communication over the network.

5.3.1 Primal-dual formulation

For the EDOP problem to be feasible, we make the following assumptions.

Assumption 5.2. The cost functions are proper, closed and convex, i.e., $f_i \in \Gamma(\mathcal{H}), \forall i \in \mathcal{V}$.

Assumption 5.3. The non-negative weight matrix[1] W associated with the communication graph satisfies the following conditions:

- Positive-definite: $W^T = W$ and $W > 0$,
- Stochasticity: $W\mathbf{1} = \mathbf{1}$ or $\mathbf{1}^T W = \mathbf{1}^T$,
- Connectivity: $\rho\left(W - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) < 1$.

With the above assumptions, the EDOP problem can be shown to be equivalent to the following optimal consensus problem (OCP):

$$\min_{x \in \mathcal{R}^m} f(x) = \sum_{i=1}^{m} f_i(x_i) \quad \mathbf{s.t.}\ (I - W)x = 0. \tag{OCP}$$

Similar with [45, Lem. 3.1], by noticing that W has a simple eigenvalue one and the corresponding eigenvector '$\mathbf{1}$' under Assumption 5.3 [89] and thus $null\{I - W\} = span\{\mathbf{1}\}$, together with Assumption 5.2, the OCP problem is equivalent to the following problem which is an alternative form of the EDOP problem:

$$\min_{x \in \mathcal{R}^m} f(x) + \iota_{\mathcal{C}}(x), \tag{5.2}$$

where $\iota_{\mathcal{C}}(x)$ is the indicator function defined as follows

$$\iota_{\mathcal{C}}(x) = \begin{cases} 0, & \text{if } x \in \mathcal{C} := \{[\theta, \theta, ..., \theta]^T | \theta \in \mathcal{R}\} \\ \infty, & \text{otherwise,} \end{cases}$$

Correspondingly, the dual formulation of the problem (5.2) is [54, Def. 15.10]:

$$\min_{y \in \mathcal{R}^m} f^*(y) + \iota_{\mathcal{C}}^*(-y), \tag{5.3}$$

[1] Here we assume each communication link is associated with a positive weight w_{ij}.

where f^* and $\iota_{\mathcal{C}}^*(\cdot)$ are the convex conjugates of f and $\iota_{\mathcal{C}}(\cdot)$ respectively. Likewise, since $\iota_{\mathcal{C}}^*(\cdot)$ indicates the orthogonal space of $\mathcal{C}$ (denoted as $\mathcal{C}^\perp$) the problem (5.3) is equivalent to the following optimal exchange problem (OEP):

$$\min_{y\in\mathcal{R}^m} f^*(y) = \sum_{i=1}^{m} f_i^*(y_i) \quad \text{s.t. } \mathbf{1}^T y = 0, \tag{OEP}$$

where $y = [y_1, y_2, ..., y_m]^T \in \mathcal{R}^m$ is the dual variable.

Remark 5.4. As we will show later, the proposed distributed algorithm can solve both the primal and dual problem simultaneously. In other words, it provides an alternative *distributed* way to solve the OEP problem[2] which usually needs to be solved by centralized or parallel approach [48]. In the sequel, the above OCP and OEP problems will be termed together as a *primal-dual* problem.

5.3.2 Some basic techniques

Bregman Iterative Regularization

In [57], a Bregman-based method is introduced and revealed to be very efficient in image processing. In order to improve the quality of image recovery, Bregman iterative regularization method attempts to solve the following optimization problem

$$\min_x J(x) + H(x)$$

iteratively by the following algorithm

$$\begin{aligned} x_{k+1} &= \operatorname*{argmin}_x \ (D_J^{y_k}(x, x_k) + H(x)) \\ y_{k+1} &= y_k - \nabla H(x_{k+1}), \end{aligned} \tag{5.4}$$

where J is a convex regularization functional, H is convex fitting functional in inverse problems, e.g., $H(x) = \|Ax - b\|^2$ for linear problems, and $D_f^{y_k}(x, x_k)$ is the Bregman distance between x and x_k. Bregman iteration has an elegant interpretation as it resembles the feedback control in control theory, i.e., feeding back the error into the input. It was shown in [61] that the above algorithm is, in fact,

[2]This problem arises from many real application areas, such as smart grid, network utility maximization and feature splitting in machine learning.

equivalent to augmented Lagrangian method for linear problems and solves the following equality-constrained problem:

$$\min_x J(x) \qquad \textbf{s.t. } H(x) = 0. \tag{5.5}$$

Inspired from the above analysis, following (5.4), we can easily come up with an algorithm for solving the OCP problem under fixed networks as follows:

$$x_{k+1} = \arg\min_{x \in \mathcal{R}^m} \left(D_f^{y_k}(x, x_k) + \frac{1}{2\gamma} \|x\|_{I-W}^2 \right) \tag{5.6a}$$

$$y_{k+1} = y_k - \frac{1}{\gamma}(I - W)x_{k+1}. \tag{5.6b}$$

However, the above algorithm cannot be carried out in a distributed way since the x-update requires the linear operator W to be evaluated implicitly thus requiring either the global knowledge of the network or infinite inner-loops of consensus to solve the subproblem (5.6a), which is not practical.

Forward-Backward Splitting Techniques (FBS)

In composite optimization problems, one always needs to solve the following inclusion: $0 \in (A + B)z$, where A and B are two operators. It is usually expensive to solve two operators together and a more efficient way is to split them into separated parts each of which is relatively easier to be evaluated. There are several splitting techniques proposed in the existing literature. Forward-backward splitting is the one dedicated for inclusion problems where A is maximally monotone and B is co-coercive. Specifically, the above inclusion can be rewritten in a split form as: $0 \in (I + \tau A)z - (I - \tau B)z$, yielding the forward-backward splitting algorithm:

$$z_{k+1} = \mathbf{prox}_{\tau A}(I - \tau B)z_k, \tag{5.7}$$

where $\mathbf{prox}_{\tau A} = (I + \tau A)^{-1}$ is the proximity operator of A. The convergence of the forward-backward splitting algorithm is guaranteed by the following Proposition.

Proposition 5.2 (Forward-Backward Splitting [54, 90])**.** *Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be maximally monotone, $B : \mathcal{H} \to \mathcal{H}$ be κ-cocoercive for some $\kappa \in (0 + \infty]$ and let $\tau \in [0, 2\kappa]$. Suppose $zer(A + B) \neq \varnothing$. Then, the sequence $(z_k)_{k \in \mathbb{N}}$ generated by the algorithm (5.7) will converge to $z^\star \in zer(A + B)$.*

5.3.3 D-FBBS algorithm for fixed networks

As abovementioned, the algorithm (5.6) does not permit for distributed implementation. To see this, let us consider the x-update (5.6a). By the necessary condition of optimality we have

$$\gamma y_k \in (I - W + \gamma \partial f)(x_{k+1}). \tag{5.8}$$

It is clear that the above inclusion involves computing the inverse of W, thus suffering from the aforementioned issues. In order to be able to solve it in a distributed way, we propose a forward-backward splitting approach as follows (cf. Prop. 5.2):

$$x_{k+1} = \mathbf{prox}_{\gamma f}(W x_k + \gamma y_k), \tag{5.9}$$

where $\mathbf{prox}_{\gamma f} = (I + \gamma \partial f)^{-1}$ is the proximity operator of f with parameter γ. According to Proposition 5.2, given certain y_k, (5.9) is equivalent to (5.8) only when it runs infinite steps. However, we will show that the OCP problem can still be solved when (5.9) is executed only once per each iteration, which yields the following algorithm:

$$\gamma y_k - (x_{k+1} - W x_k) \in \gamma \partial f(x_{k+1}) \tag{5.10a}$$

$$(I - W)x_{k+1} + \gamma(y_{k+1} - y_k) = 0, \tag{5.10b}$$

or, equivalently,

$$x_{k+1} = \underset{x \in \mathcal{R}^m}{\arg\min} \left(D_f^{y_k}(x, x_k) + \frac{1}{2\gamma} \|x - W x_k\|^2 \right) \tag{5.11a}$$

$$y_{k+1} = y_k - \frac{1}{\gamma}(I - W)x_{k+1}. \tag{5.11b}$$

Remark 5.5. It can be seen from the above algorithm that it can be carried out in a distributed manner since each agent only requires local information to solve its own optimization subproblem. In particular, at each iteration, each agent collects the information from its neighbors and solve the local optimization problem (5.11a) based on the obtained weighted average. The estimated optimum of the next iteration is then communicated to its neighbors for updating the dual variable according to the step (5.11b) which can be also done locally.

We summarize the above distributed algorithm–Forward-Backward Bregman Splitting (D-FBBS)–in *Algorithm* 1.

Algorithm 1 D-FBBS for Fixed Networks

1: **Initialization**: $y_{i,0} = 0, \forall i \in \mathcal{V}$ such that $\mathbf{1}^T y_0 = 0$, while the initial guess of x_0 can be arbitrarily assigned.

2: **Primal Update**: For each agent $i \in \mathcal{V}$, compute:

$$x_{i,k}^{av} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij} x_{j,k}$$

$$x_{i,k+1} = \arg\min_{x_i \in \mathcal{R}^d} \left(D_f^{y_{i,k}}(x_i, x_{i,k}) + \frac{1}{2\gamma} \left\| x_i - x_{i,k}^{av} \right\|^2 \right)$$

3: **Dual Update**: For each agent $i \in \mathcal{V}$,

$$y_{i,k+1} = y_{i,k} - \frac{1}{\gamma} \sum_{j \in \mathcal{N}_i} w_{ij} \left(x_{i,k+1} - x_{j,k+1} \right)$$

4: Set k $\rightarrow$ k+1 and go to Step 2 until certain stopping criteria is satisfied (cf. Proposition 5.4).

Additionally, for facilitating our sequent analysis, it is beneficial to add (5.10a) with (5.10b), which leads to:

$$\gamma y_{k+1} - W(x_{k+1} - x_k) \in \gamma \partial f(x_{k+1}) \tag{5.12a}$$

$$(I - W)x_{k+1} + \gamma(y_{k+1} - y_k) = 0. \tag{5.12b}$$

Remark 5.6. It is clear from (5.12) that the proposed algorithm after splitting is no longer the exact Bregman iterative method since $y_{k+1} \notin \partial f(x_{k+1})$. Thus, it also can be understood as an inexact version of Bregman iterative regularization.

5.3.4 Theoretical connections to existing algorithms

In this section, we provide a variant of the proposed algorithm, termed '*Inexact* D-FBBS', to tackle cost functions with certain property, i.e., having Lipschitz gradients and show the specific connections of D-FBBS with some existing well-known algorithms. In particular, under the above framework, we can perform another proper forward-backward splitting for functions having Lipschitz gradients, which essentially belongs to inexact Uzawa methods. We will also show that, via proper

change of variables, it has close connections with existing well known methods, such as preconditioned augmented Lagrangian methods and primal-dual approaches.

In order to carry out the following analysis, we first introduce the augmented Lagrangian associated with the above *primal-dual* problem as follows:

$$\psi(x, y) = f(x) - y^T x + \frac{1}{2\gamma} \|x\|^2_{I-W} . \tag{5.13}$$

Remark 5.7. The dual variable y plays a key role in reconciling the discrepancy of the interests of different agents for achieving global optimum.

The following lemmas will be useful in the subsequent analysis of the connections of the proposed algorithm to some existing algorithms and the convergence analysis.

Lemma 5.1. *Let P be a $m \times m$ matrix such that $null(P) = span\{\mathbf{1}\}$. Then, for each $y \in span^{\perp}\{\mathbf{1}\}$, there exists a unique $y' \in span^{\perp}\{\mathbf{1}\}$ such that $y = Py'$ and vice versa, i.e., the P-transformation between y and y' is bijective.*

Proof. See Appendix B.

Lemma 5.2 (Conservation Property II)**.** *Consider the sequence $\{y_k\}_{k\geq 0}$ generated by (5.12b). Suppose $\mathbf{1}^T y_0 = 0$ and Assumption 5.3 holds, then $\mathbf{1}^T y_k = 0, \forall k \geq 0$.*

The above conservation property is immediately followed by pre-multiplying both sides of (5.12b) by $\mathbf{1}^T$ and knowing from Assumption 5.3 that $\mathbf{1}^T(I - W) = 0$.

Inexact Uzawa Method

The proposed algorithm is still expensive in the sense that at each iteration we need to evaluate the inverse of $I + \gamma\partial f$. However, if we know that $f \in \Gamma(\mathcal{H})$ has Lipschitz gradients, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \leq \kappa \|x - y\| , \ \forall x, y \in \mathcal{D}, \tag{5.14}$$

then f can be also evaluated forwardly in a cheaper way. That is, applying the forward-backward splitting technique gives

$$x_{k+1} = Wx_k - \gamma(\nabla f(x_k) - y_k), \tag{5.15}$$

which amounts to an *inexact* Uzawa Method being applied to the augmented Lagrangian (5.13), i.e.,

$$x_{k+1} = x_k - \gamma \nabla_x \psi(x). \tag{5.16}$$

Thus, the above analysis leads to the following variant (*Inexact* D-FBBS) of the proposed distributed algorithm:

$$x_{k+1} = Wx_k - \gamma(\nabla f(x_k) - y_k) \tag{5.17a}$$

$$y_{k+1} = y_k - \frac{1}{\gamma}(I - W)x_{k+1}. \tag{5.17b}$$

Suppose $y_0 = 0$. Summing (5.17b) over k and substituting it into (5.17a) yields

$$\underbrace{x_{k+1} = Wx_k - \gamma \nabla f(x_k)}_{\text{DSM}} \underbrace{- \sum_{i=0}^{k} (I - W)x_i}_{\text{Correction}},$$

which can be termed corrected DSM and is, indeed, equivalent to EXTRA with $W = \tilde{W} = \frac{I+W'}{2}$, where $\tilde{W}$ and W' are two properly designed weight matrices [66, Eq. (2.13)]. Thus, the convergence result[3] for the OCP problem follows from the similar analysis therein, i.e., $\gamma \leq \frac{2\lambda_{min}(W)}{\kappa}$.

The D-FBBS algorithm is also closely related (or equivalent) to some well-known existing algorithms by using preconditioned technique or coordination transform.

Preconditioned Augmented Lagrangian Method

To avoid the computation of the inverse of the weight matrix W, a clever way is to add an extra proximity term as follows

$$x_{k+1} = \arg\min_{x \in \mathcal{R}^m} \psi(x, y_k) + \frac{1}{2\gamma} \|x - x_k\|_W^2 \tag{5.18a}$$

$$y_{k+1} = y_k - \frac{1}{\gamma}(I - W)x_{k+1}, \tag{5.18b}$$

where ψ is the Lagrangian defined in (5.13). Introducing the prox-term allows the x-update step to be evaluated explicitly. Note that in the algorithm of Bregman Operator Splitting the above prox-term is replaced with the Bregman distance

[3]Note that we can obtain the linear convergence result as well if the cost function is known to be also strongly convex.

induced by a strongly convex function [59]. It is not difficult to verify that the above algorithm with $\mathbf{1}^T y_0 = 0$ is equivalent to the proposed *Algorithm* 1. Moreover, setting $y_0 = 0$, summing (5.18b) over k and substituting it into (5.18a) yields

$$x_{k+1} = \mathbf{prox}_{\gamma f}\left(Wx_k - \sum_{i=0}^{k}(I-W)x_i\right),$$

which, similar as before, can be shown to be equivalent to P-EXTRA with $W = \tilde{W} = \frac{I+W'}{2}$ [67]. However, we will show that our algorithm not only solve the primal OCP problem but also the dual OEP problem. In this regard, our convergence analysis is also different from theirs.

Moreover, it is not difficult to verify that the above algorithm is also equivalent to the Jacobi variant of distributed Augmented Lagrangian (AL) methods proposed in [65] when the inner iterations of consensus are carried out only once, i.e., $\tau = 1$. Note that their convergence result does not support this case.

Primal-Dual Approach

In [56], a general *primal-dual* proximal point algorithm is proposed as follows:

$$\begin{aligned} x_{k+1} &= \mathbf{prox}_{\gamma f}(x_k + \gamma K \check{y}_k) \\ y_{k+1} &= \mathbf{prox}_{\delta g}(y_k - \delta K^T x_{k+1}) \\ \check{y}_{k+1} &= y_{k+1} + \theta(y_{k+1} - y_k),\ \theta \in [0,1] \end{aligned} \tag{5.19}$$

to solve the generic saddle point problem in image processing. In particular, when $g = 0$ and $\theta = 1$, it solves the following *symmetric* saddle point problem:

$$0 \in \left\{ \begin{bmatrix} \partial f & K \\ K^T & O \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right\} \tag{5.20}$$

Let $K = \sqrt{I-W}$. Since $null\{K\} = null\{I-W\} = span\{\mathbf{1}\}$, we know from Lemma 5.1 that for each $y' \in span^{\perp}\{\mathbf{1}\}$ there exists a unique $y \in span^{\perp}\{\mathbf{1}\}$ such that $y' = Ky$. Thus, the *primal-dual* algorithm (5.19) can be rewritten as [68, 91]:

$$\begin{aligned} x_{k+1} &= \mathbf{prox}_{\gamma f}(x_k + \gamma(2y'_k - y'_{k-1})) \\ y'_{k+1} &= y'_k - \delta(I-W)x_{k+1}. \end{aligned} \tag{5.21}$$

It is easy to verify that when $\delta = \frac{1}{\gamma}, \mathbf{1}^T y_0 = 0$ (Thus $\mathbf{1}^T y_k = 0, \forall k$ by Lemma 5.2) the above algorithm is equivalent to the proposed distributed *Algorithm 1* and it, in fact, solves the following *asymmetric* saddle point problem :

$$0 \in \left\{ \begin{bmatrix} \partial f & I \\ I - W & O \end{bmatrix} \begin{bmatrix} x \\ y' \end{bmatrix} \right\}, \ \mathbf{1}^T y' = 0, \tag{5.22}$$

which is exactly the optimality conditions (5.23).

Remark 5.8. The algorithm developed based on the symmetric formulation (5.20) is not suitable for stochastic networks as the saddle point varies with the topology, i.e., depending on the weight matrix W and thus K. As such, the above connection via proper change of variables will be no longer valid for stochastic networks.

5.3.5 Convergence analysis

The KKT conditions for optimality of the above *primal-dual* problem are:

$$\text{(Primal Feasibility)} \qquad (I_{md} - W \otimes I)x^\star = 0, \tag{5.23a}$$

$$\text{(Dual Feasibility)} \qquad (\mathbf{1} \otimes I)^T y^\star = 0, \tag{5.23b}$$

$$\text{(Lagrangian Optimality)} \qquad y^\star \in \partial f(x^\star). \tag{5.23c}$$

Remark 5.9. As we will show later, the dual feasibility (5.23b), with proper initialization, can be always guaranteed by the proposed algorithm (cf. Lemma 5.2). Thus, in the sequel, we will focus on the domain $\mathcal{D} = \{(x, y) | x \in \mathcal{R}^m, \ y \in \mathcal{C}^\perp\}$.

Assumption 5.4. The augmented Lagrangian ψ, as defined in (5.13), has a saddle point $(x^\star, y^\star) \in \mathcal{D}$.

The following proposition shows the equivalence between the saddle point of (5.13) and the optimality conditions (5.23).

Proposition 5.3 (Optimality as Saddle Points)**.** *Consider a pair $(x^\star, y^\star) \in \mathcal{D}$. Then, it is a saddle point of the Lagrangian (5.13) if and only if it satisfies the optimality conditions (5.23). Moreover, if it is a saddle point, then $x^\star$ solves the OCP problem and $y^\star$ solves the OEP problem respectively.*

Proof. See Appendix B.

Before proceeding to the main result, let us first establish the following lemmas:

Lemma 5.3. *Let* $\tilde{x} = (I - \frac{\mathbf{1}\mathbf{1}^T}{m})x$ *and* $u_{k+1} = \|\tilde{x}_{k+1}\|^2_{I-W} + \|x_{k+1} - x_k\|^2_W$. *Suppose Assumptions 5.2 and 5.3 hold, the sequence* $\{u_k\}_{k\geq 0}$ *generated from the D-FBBS algorithm (5.11) is monotonically non-increasing. We even have*

$$u_{k+1} \leq u_k - \|x_{k+1} - x_k\|^2_{I-W}. \tag{5.24}$$

Proof. Let $\bar{x} \in span\{\mathbf{1}\}$. Knowing that ∂f is maximally monotone in the domain $\mathcal{R}^{md}$ by Assumption 5.2 and that $\gamma y_{k+1} - W(x_{k+1} - x_k) \in \gamma\partial f(x_{k+1})$ and $\gamma y_k - W(x_k - x_{k-1}) \in \gamma\partial f(x_k)$ from (5.12a), we have

$$\begin{aligned}
&\langle \gamma(y_{k+1} - y_k) - W(x_{k+1} - x_k) + W(x_k - x_{k-1}), x_{k+1} - x_k\rangle \\
&\quad = \langle -W(x_{k+1} - x_k) + W(x_k - x_{k-1}), x_{k+1} - x_k\rangle \\
&\qquad - \langle (I - W)(x_{k+1} - \bar{x}), x_{k+1} - x_k\rangle \geq 0,
\end{aligned} \tag{5.25}$$

where we have used (5.12b) to replace $\gamma(y_{k+1} - y_k)$ and the stochasticity of W (cf. Assumption 5.3) to obtain the last term, i.e., $(I - W)x = (I - W)(x - \bar{x})$.

Recalling that W is symmetric and positive definite by Assumption 5.3, the above inequality further leads to

$$\begin{aligned}
&\langle (I - W)(x_{k+1} - \bar{x}), x_{k+1} - x_k\rangle \\
&\quad \leq \langle -W(x_{k+1} - x_k) + W(x_k - x_{k-1}), x_{k+1} - x_k\rangle \\
&\quad = -\|x_{k+1} - x_k\|^2_W + \langle W(x_k - x_{k-1}), x_{k+1} - x_k\rangle \\
&\quad \leq -\|x_{k+1} - x_k\|^2_W + \frac{\|x_{k+1} - x_k\|^2_W + \|x_k - x_{k-1}\|^2_W}{2} \\
&\quad = \frac{-\|x_{k+1} - x_k\|^2_W + \|x_k - x_{k-1}\|^2_W}{2}.
\end{aligned} \tag{5.26}$$

Combining the following identity

$$\begin{aligned}
&2\langle (I - W)(x_{k+1} - \bar{x}), x_{k+1} - x_k\rangle \\
&\qquad = \|x_{k+1} - \bar{x}\|^2_{I-W} - \|x_k - \bar{x}\|^2_{I-W} + \|x_{k+1} - x_k\|^2_{I-W}
\end{aligned} \tag{5.27}$$

with (5.26) yields

$$\begin{aligned}
&\|x_{k+1} - \bar{x}\|^2_{I-W} - \|x_k - \bar{x}\|^2_{I-W} + \|x_{k+1} - x_k\|^2_{I-W} \\
&\qquad \leq -\|x_{k+1} - x_k\|^2_W + \|x_k - x_{k-1}\|^2_W
\end{aligned} \tag{5.28}$$

which is equivalent to

$$\begin{aligned}\|\tilde{x}_{k+1}\|_{I-W}^2 - \|\tilde{x}_k\|_{I-W}^2 + \|x_{k+1} - x_k\|_{I-W}^2 \\ \leq -\|x_{k+1} - x_k\|_W^2 + \|x_k - x_{k-1}\|_W^2 .\end{aligned} \tag{5.29}$$

Let $u_{k+1} = \|\tilde{x}_{k+1}\|_{I-W}^2 + \|x_{k+1} - x_k\|_W^2$. Rearranging the terms of (5.29) leads to (5.24).

The following lemma tells us that once the sequence generated by the proposed D-FBBS algorithm reaches consensus, the consensus value is the solution to the *primal-dual* problem.

Lemma 5.4. *Consider the sequence $\{(x_k, y_k)\}_{k\geq 0}$ generated by the D-FBBS algorithm (5.11). Suppose Assumptions 5.2, 5.3 and 5.4 hold and the sequence $\{x_k\}_{k\geq 0}$ converges to some value $x^\star = \mathbf{1} \otimes \theta^\star$ for certain $\theta^\star \in \mathcal{R}$. Then, with proper initialization of $\mathbf{1}^T y_0 = 0$, the sequence $\{(x_k, y_k)\}_{k\geq 0}$ will converge to a saddle point $(x^\star, y^\star)$ of the* primal-dual *problem.*

Proof. See Appendix B.

The following proposition aims to establish the relationship between the objective suboptimality and the residuals.

Proposition 5.4 (Stopping Criteria)**.** *The objective suboptimality of $\{(x_k, y_k)\}_{k\geq 0}$ is bounded by the residuals of the optimality conditions as follows:*

$$\psi(x_k, y_k) - f^\star \leq \frac{1}{\gamma} \|x_k - x^\star\|_W \|x_{k+1} - x_k\|_W + \frac{1}{2\gamma} \|\tilde{x}_k\|_{I-W}^2 .$$

Proof. See Appendix B.

Remark 5.10. According to Proposition 5.4, if one can estimate the upper bound of $\|x_k - x^\star\|_W$, we can have a non-ergodic convergence rate of $o(\frac{1}{\sqrt{k}})$ in terms of the objective error.

Now, we are ready to present the main convergence result for fixed networks.

Theorem 5.1. *Suppose Assumptions 5.2, 5.3 and 5.4 hold. Then, the sequence $\{(x_k, y_k)\}_{k\geq 0}$ generated by Algorithm 1 will converge to a saddle point $(x^\star, y^\star)$ of the* primal-dual *problem. Moreover, the fixed point residual in terms of $\|x_k - \bar{x}_k\|_{I-W}^2$ and $\|x_{k+1} - x_k\|_W^2$ will decease at a non-ergodic rate of $o(\frac{1}{k})$.*

Proof. Consider the iteration (5.12). Since $\mathbf{1}^T y_k = 0, \forall k \geq 0$ by Lemma 5.2 and $null(I - W) = span\{\mathbf{1}\}$, there exists a unique $y'_k \in span^{\perp}\{\mathbf{1}\}$ such that $y_k = (I - W)y'_k$ by Lemma 5.1. Knowing that ∂f is maximally monotone by Assumption 5.2 and $\gamma y^\star \in \gamma \partial f(x^\star)$ from the optimality conditions (5.23), together with (5.12a) we have

$$\begin{aligned}&\langle \gamma(y_{k+1} - y^\star) - W(x_{k+1} - x_k), x_{k+1} - x^\star \rangle \\ &= \left\langle \gamma(I - W)(y'_{k+1} - y'^\star), x_{k+1} - x^\star \right\rangle - \langle W(x_{k+1} - x_k), x_{k+1} - x^\star \rangle \geq 0.\end{aligned} \tag{5.30}$$

Since W is symmetric by Assumption 5.3, with (5.12b) we have

$$\begin{aligned}&\langle \gamma(y_{k+1} - y^\star) - W(x_{k+1} - x_k), x_{k+1} - x^\star \rangle \\ &= \gamma \left\langle (I - W)(x_{k+1} - x^\star), y'_{k+1} - y'^\star \right\rangle - \langle W(x_{k+1} - x_k), x_{k+1} - x^\star \rangle \\ &= \gamma^2 \left\langle -(y_{k+1} - y_k), y'_{k+1} - y'^\star \right\rangle - \langle W(x_{k+1} - x_k), x_{k+1} - x^\star \rangle \\ &= \gamma^2 \left\langle -(I - W)(y'_{k+1} - y'_k), y'_{k+1} - y'^\star \right\rangle - \langle W(x_{k+1} - x_k), x_{k+1} - x^\star \rangle \geq 0.\end{aligned} \tag{5.31}$$

Then, using the similar identity as (5.27) yields

$$\begin{aligned}&\gamma^2 \left\| y'_{k+1} - y'^\star \right\|^2_{I-W} - \gamma^2 \left\| y'_k - y'^\star \right\|^2_{I-W} + \left\| x_{k+1} - x^\star \right\|^2_W - \left\| x_k - x^\star \right\|^2_W \\ &\quad \leq -\gamma^2 \left\| y'_{k+1} - y'_k \right\|^2_{I-W} - \left\| x_{k+1} - x_k \right\|^2_W .\end{aligned} \tag{5.32}$$

Let $\tilde{x}_k = (I - \frac{\mathbf{1}\mathbf{1}^T}{m})x_k$. Recalling that $y_{k+1} = (I - W)y'_{k+1}, y_k = (I - W)y'_k$ we have

$$\begin{aligned}\gamma^2 \left\| y'_{k+1} - y'_k \right\|_{I-W} &= \gamma^2 \left\langle y'_{k+1} - y'_k, y_{k+1} - y_k \right\rangle = -\gamma \left\langle y'_{k+1} - y'_k, (I - W)x_{k+1} \right\rangle \\ &= -\left\langle \gamma(I - W)(y'_{k+1} - y'_k), x_{k+1} \right\rangle = \left\| x_{k+1} \right\|^2_{I-W} = \left\| \tilde{x}_{k+1} \right\|^2_{I-W} ,\end{aligned} \tag{5.33}$$

where we have used (5.12b) to obtain the last two equalities. Thus, the above relation (5.32) can be rewritten as

$$\begin{aligned}&\gamma^2 \left\| y'_{k+1} - y'^\star \right\|^2_{I-W} - \gamma^2 \left\| y'_k - y'^\star \right\|^2_{I-W} + \left\| x_{k+1} - x^\star \right\|^2_W \\ &\qquad - \left\| x_k - x^\star \right\|^2_W \leq - \left\| \tilde{x}_{k+1} \right\|^2_{I-W} - \left\| x_{k+1} - x_k \right\|^2_W .\end{aligned} \tag{5.34}$$

Let $V_k = \gamma^2 \left\| y'_k - y'^\star \right\|^2_{I-W} + \left\| x_k - x^\star \right\|^2_W$ (note that $V_k \geq 0$ by Assumption 5.3). Summing (5.34) over k from 0 to $t - 1$ yields

$$\sum_{k=0}^{t-1} \left(\left\| \tilde{x}_{k+1} \right\|^2_{I-W} + \left\| x_{k+1} - x_k \right\|^2_W \right) \leq V_0 - V_t < \infty. \tag{5.35}$$

By Lemma 2, $u_{k+1} = \|\tilde{x}_{k+1}\|^2_{I-W} + \|x_{k+1} - x_k\|^2_W$ is monotonically non-increasing. Thus, we have

$$tu_t \leq \sum_{k=0}^{t-1} u_{k+1} \leq V_0, \tag{5.36}$$

yielding $u_t \leq \frac{V_0}{t} = o(\frac{1}{t})$. Since $u_t \geq 0$ by Assumption 5.3, we have $\lim_{t\to\infty} u_t = 0$. In addition, from (5.34) we know that V_k is bounded and so is x_k, by standard analysis for weak cluster points, it follows that x_k will converge to some $x^\star \in span\{\mathbf{1}\}$. Thus, by Lemma 5.4, we conclude that the sequence $\{(x_k, y_k)\}_{k\geq 0}$ will converge to a saddle point $(x^\star, y^\star)$ with a non-ergodic rate of $o(\frac{1}{k})$.

5.4 Augmented Distributed Gradient Methods

The proposed D-FBBS algorithm in the previous section has good convergence performance comparable to the centralized counterpart. However, it requires the weight matrix to be positive definite and symmetric. This will be problematic when it comes to consensus protocol design, restricting its application to general (directed) communication graphs. To deal with general graphs as well as heterogeneous computations involved in different agents, we proposed a new algorithm which can be regarded as an augmented version of the existing DSM algorithm [27]. The proposed algorithm resembles (approximates) its centralized counterpart and is, in fact, running in analogy to it in the average space.

5.4.1 AugDGM algorithm for fixed networks

Different from the previous D-FBBS algorithm, we make the following assumptions on the weight matrix[4] as well as the cost functions:

Assumption 5.5. The weight matrix $W = \{w_{ij}\}$ associated with the communication graph satisfies $\mathbf{1}^T W = \mathbf{1}^T$, $W\mathbf{1} = \mathbf{1}$, and $\eta = \rho(W - \frac{\mathbf{1}\mathbf{1}^T}{m}) < 1$ (see [89] for the details on the design of the weight matrix).

Assumption 5.6. Each objective function f_i is convex and coercive[5] such that $\|x_i\| \to \infty$ leads to $f_i(x_i) \to \infty$.

[4]The underpinning graph can be directed as long as it is balanced and the assumption of the weight matrix is satisfied (cf. Assumption 5.5).

[5]It can be relaxed to only require the overall function f to be convex and coercive.

Assumption 5.7. Each objective function f_i is continuously differentiable and has Lipschitz gradient as follows:

$$\|g_i(x_i) - g_i(x_i')\| \leq L_i \|x_i - x_i'\|, \forall x_i, x_i' \in \mathcal{R}$$

where L_i is the Lipschitz constant while $g_i(x_i)$ and $g_i(x_i')$ are the gradients of f_i evaluated at x_i and x_i' respectively.

Remark 5.11. It follows immediately from Assumptions 5.6 and 5.7 that the global function f is convex, coercive and has Lipschitz gradient with Lipschitz constant $L = \max\{L_i\}$ since we have for any $x, y \in \mathcal{H}$

$$\|g(x) - g(y)\| = \left\| \begin{bmatrix} g_1(x_1) - g_1(y_1) \\ g_2(x_2) - g_2(y_2) \\ \cdots \\ g_m(x_m) - g_m(y_m) \end{bmatrix} \right\| \leq \sqrt{\sum_{i=1}^{m} L_i^2 \|x_i - y_i\|^2} \leq L \|x - y\|.$$

To exactly solve the EDOP problem under the above assumptions, we propose a new augmented distributed gradient method (termed AugDGM) as detailed in *Algorithm* 2. In contrast to most of the existing algorithms, the proposed algorithm involves an extra step of consensus on the gradients:

- Local update step for optimization. For agreement of the estimates of all agents to the global optimum, we use the following rule for update [23, 27]:

$$\begin{aligned} s_{i,k+1} &= x_{i,k} - \gamma_i \cdot y_{i,k} \\ x_{i,k+1} &= s_{i,k} + \sum_{j \in \mathcal{N}_i} w_{ij}(s_{j,k} - s_{i,k}), \end{aligned} \tag{5.37}$$

  where $s_{i,k}$ is the intermediate variable of agent i to be sent to its neighbors at time k, $x_{i,k}$ the estimate of agent i obtained at time k and $\gamma_i \in \{0\} \cup [\gamma_{\min}, \gamma_{\max}]$ is the stepsize chosen by agent i all the time.

- Dynamic average consensus step. To ensure that the algorithm has the ability to seek the exact optimum, we employ dynamic average consensus to track the average of the gradients of objective functions [83]

$$y_{i,k+1} = y_{i,k} + \sum_{j \in \mathcal{N}_i} w_{ij}(y_{j,k} - y_{i,k}) + \Delta g_{i,k}, \tag{5.38}$$

where $\Delta g_{i,k} = g_i(x_{i,k+1}) - g_i(x_{i,k})$ and $y_{i,k}$ is the introduced auxiliary variable tracking the average of the gradients $g(x_k)$.

By properly intertwining the above two steps, the proposed distributed algorithm can be rewritten in a compact form as follows:

$$x_{k+1} = W[x_k - \gamma \odot y_k] \tag{5.39a}$$

$$y_{k+1} = W[y_k + \Delta g_k], \tag{5.39b}$$

where y_k is the introduced auxiliary variable, $\Delta g_k = g(x_{k+1}) - g(x_k)$ the incremental change of the gradients and γ the vector of stepsize chosen by all agents.

Algorithm 2 AugDGM for Fixed Networks

1: **Initialization**: $\forall$ agent $i \in \mathcal{V}$: $x_{i,0}$ arbitrarily assigned while $y_{i,0} = g_i(x_{i,0})$.
2: **Local Optimization**: $\forall$ agent $i \in \mathcal{V}$, computes:

$$\begin{aligned} s_{i,k} &= x_{i,k} - \gamma_i \cdot y_{i,k} \\ x_{i,k+1} &= s_{i,k} + \sum_{j \in \mathcal{N}_i} w_{ij}(s_{j,k} - s_{i,k}) \end{aligned} \tag{5.40}$$

3: **Dynamic Average Consensus**: $\forall$ agent $i \in \mathcal{V}$, computes:

$$\begin{aligned} q_{i,k} &= y_{i,k} + g_i(x_{i,k+1}) - g_i(x_{i,k}) \\ y_{i,k+1} &= q_{i,k} + \sum_{j \in \mathcal{N}_i} w_{ij}\,(q_{j,k} - q_{i,k}) \end{aligned} \tag{5.41}$$

4: Set k $\rightarrow$ k+1 and go to Step 2.

5.4.2 Convergence analysis

To quantify the variation of the stepsizes used by agents, we introduce the following parameter which will be crucial in the subsequent convergence analysis.

Definition 5.7 (Heterogeneity of Stepsize I)**.** Let γ be the vector of the stepsizes chosen by the agents. Then, the heterogeneity of stepsize (HoS) is defined as

$$\Delta_\gamma = \frac{\|\tilde{\gamma}\|}{\|\bar{\gamma}\|},$$

where $\bar{\gamma} = \frac{\mathbf{1}\mathbf{1}^T}{m}\gamma$ and $\tilde{\gamma} = \gamma - \bar{\gamma}$.

Let us first consider a scalar sequence and its associated "L2-stability" result.

Lemma 5.5. *Let* $\{\upsilon_k\}_{k\geq 0}$ *and* $\{\omega_k\}_{k\geq 0}$ *be positive scalar sequences such that for all* $k \geq 0$

$$\upsilon_{k+1} \leq \eta\upsilon_k + \omega_k, \tag{5.42}$$

where $\eta \in (0,1)$ *is the decaying factor. Let* $\Upsilon_k = \sqrt{\sum_{i=0}^{k} \upsilon_i^2}$ *and* $\Omega_k = \sqrt{\sum_{i=0}^{k} \omega_i^2}$ *be the square root of "energy" from* 0 *to* k*. Then, we have*

$$\Upsilon_k \leq p\Omega_k + q,$$

where $p = \frac{\sqrt{2}}{1-\eta}$ *and* $q = \sqrt{\frac{2}{1-\eta^2}}\upsilon_0$.

Proof. See Appendix B.

To facilitate our subsequent analysis, let us consider the following auxiliary sequence which runs in analogy with (5.39a):

$$\bar{x}_{k+1} = \bar{x}_k - \overline{\gamma \odot y}_k, \tag{5.43}$$

where $\bar{x}_k = \frac{\mathbf{1}\mathbf{1}^T}{m} x_k$ and $\overline{\gamma \odot y}_k = \frac{\mathbf{1}\mathbf{1}^T}{m}(\gamma \odot y_k)$.

Likewise, with Assumption 5.5, projecting (5.39b) into the average space gives

$$\bar{y}_{k+1} = \bar{y}_k + \bar{g}_{k+1} - \bar{g}_k, \tag{5.44}$$

where $\bar{y}_k = \Pi_{\|} y_k$ and $\bar{g}_k = \Pi_{\|} g(x_k)$ and we have the following conservation property for the above sequence.

Lemma 5.6 (Conservation Property III)**.** *Consider the sequence (5.44). Let* $y_0 = g(x_0)$*. Suppose Assumption 5.5 holds. Then, we have* $\bar{y}_k = \bar{g}_k, \forall k \geq 0$.

This lemma is immediately followed by summing (5.44) over k and knowing the fact that $y_0 = g(x_0)$.

Remark 5.12. It follows from Proposition 5.1-(iii) and Lemma 5.6 that $\overline{\gamma_k \odot y_k} = \bar{\gamma} \odot \bar{y}_k + \overline{\tilde{\gamma} \odot \tilde{y}_k} = \bar{\gamma} \odot \bar{g}_k + \overline{\tilde{\gamma} \odot \tilde{y}_k}$. Thus, the proposed AugDGM algorithm, in fact, resembles the centralized counterpart with some approximation error.

Before proceeding to the main result, we present our next important lemma.

Lemma 5.7. *Consider the algorithm 2 and suppose Assumptions 5.5 and 5.7 hold. Let* $X_k = \sqrt{\sum_{i=0}^{k} \|\tilde{x}_i\|^2}$, $Y_k = \sqrt{\sum_{i=0}^{k} \|\tilde{y}_i\|^2}$ *and* $Z_k = \sqrt{\sum_{i=0}^{k} \|\overline{\gamma \odot y}_i\|^2}$ *be the signal energy from* 0 *to* k *and* $\gamma_{max} = \max\{\gamma_i\}$, $\beta = \gamma_{max} L$ *and* $\eta' = \eta + \beta(1+\Delta_\gamma)$. *If* $\beta < \frac{(1-\eta)^2}{(1+\Delta_\gamma)(2\eta^3+2\eta^2-\eta+1)}$ *such that* $\rho_1\rho_2 < 1$ *and* $\eta' < 1$, *then we have*

$$X_k \leq \frac{\rho_1 p_2 + p_1}{1-\rho_1\rho_2} Z_k + \frac{q_1 + \rho_1 q_2}{1-\rho_1\rho_2} \tag{5.45a}$$

$$Y_k \leq \frac{\rho_2 p_1 + p_2}{1-\rho_1\rho_2} Z_k + \frac{q_2 + \rho_2 q_1}{1-\rho_1\rho_2}, \tag{5.45b}$$

where $\rho_1 = \frac{\sqrt{2}\eta\gamma_{max}(1+\Delta_\gamma)}{1-\eta}$, $p_1 = \frac{\sqrt{2}\eta\Delta_\gamma}{1-\eta}$, $q_1 = \frac{\sqrt{2}\|\tilde{x}_0\|}{\sqrt{1-\eta^2}}$ *and* $\rho_2 = \frac{\sqrt{2}\eta(1+\eta)L}{1-\eta'}$, $p_2 = \frac{\sqrt{2}\eta L(1+\Delta_\gamma)}{1-\eta'}$, $q_2 = \frac{\sqrt{2}\|\tilde{y}_0\|}{\sqrt{1-\eta'^2}}$.

Proof. See Appendix B. □

Now, we are ready to present our main result.

Theorem 5.2. *Consider the distributed algorithm 2 with* $y_0 = g(x_0)$ *and suppose Assumptions 5.1, 5.5, 5.6 and 5.7 hold. Then, there exists a positive number* $\gamma^\star(\eta, \Delta_\gamma)/L$ *such that if* $\gamma_{\max} < \gamma^\star$, *we have* $\lim_{k\to\infty} \|x_k - \bar{x}_k\| = 0$ *and* $\lim_{k\to\infty} f(x_k) = f^\star$, *where* $f^\star$ *is the optimal value of the EDOP problem.*

Proof. Consider the sequence (5.43). Since f has Lipschitz gradient by Assumption 5.7 and Remark 5.11, we have for $\forall x, x' \in \mathcal{R}^m$

$$f(x') \leq f(x) + g(x)^T(x'-x) + \frac{L}{2}\|x'-x\|^2.$$

Let $\Delta\bar{x}_k = \bar{x}_{k+1} - \bar{x}_k = -\overline{\gamma \odot y}_k$. Plugging $x' = \bar{x}_{k+1}$ and $x = \bar{x}_k$ into the above relation yields

$$\begin{aligned} f(\bar{x}_{k+1}) &\leq f(\bar{x}_k) + g(\bar{x}_k)^T\Delta\bar{x}_k + \frac{L}{2}\|\Delta\bar{x}_k\|^2 \\ &\leq f(\bar{x}_k) + g(x_k)^T\Delta\bar{x}_k + \frac{L}{2}\|\Delta\bar{x}_k\|^2 + (g(\bar{x}_k) - g(x_k))^T\Delta\bar{x}_k \\ &\leq f(\bar{x}_k) - \bar{y}_k^T\overline{\gamma \odot y}_k + \frac{L}{2}\|\Delta\bar{x}_k\|^2 + (g(\bar{x}_k) - g(x_k))^T\Delta\bar{x}_k, \end{aligned} \tag{5.46}$$

where for the last inequality we have used the fact that $g(x_k)^T\Delta\bar{x}_k = \bar{g}_k^T\Delta\bar{x}_k = \bar{y}_k^T\Delta\bar{x}_k$, the first equality of which follows from Proposition 5.1-(i) while the second is due to Conservation Property III in Lemma 5.6.

Let us first bound the second term. Using Proposition 5.1-(iii) we have

$$\begin{aligned}\bar{y}_k^T\overline{\gamma \odot y}_k &= \frac{\sqrt{m}}{\|\bar{\gamma}\|}(\overline{\gamma \odot y}_k - \overline{\tilde{\gamma} \odot \tilde{y}}_k)^T\overline{\gamma \odot y}_k \\ &\geq \frac{\sqrt{m}}{\|\bar{\gamma}\|}(\|\overline{\gamma \odot y}_k\|^2 - \left\|\overline{\tilde{\gamma} \odot \tilde{y}}_k\right\| \|\overline{\gamma \odot y}_k\|) \\ &\geq \frac{\sqrt{m}}{\|\bar{\gamma}\|}\|\overline{\gamma \odot y}_k\|^2 - \frac{\|\tilde{\gamma}\|}{\|\bar{\gamma}\|}\|\tilde{y}_k\| \|\overline{\gamma \odot y}_k\| \\ &\geq \frac{1}{\gamma_{max}}\|\overline{\gamma \odot y}_k\|^2 - \Delta_\gamma \|\tilde{y}_k\| \|\overline{\gamma \odot y}_k\|,\end{aligned} \tag{5.47}$$

where we have employed Proposition 5.1-(iv) to obtain the third inequality as well as the definition of HoS (cf. Definition 5.7) for the last inequality. Then, let us consider the last deviate term. By Assumption 5.7 and Remark 5.11, we obtain

$$\left|(g(\bar{x}_k) - g(x_k))^T\Delta\bar{x}_k\right| \leq L\|\tilde{x}_k\| \|\Delta\bar{x}_k\|. \tag{5.48}$$

Then, combining (5.46), (5.47) and (5.48) leads to

$$f(\bar{x}_{k+1}) \leq f(\bar{x}_k) - \left(\frac{1}{\gamma_{max}} - \frac{L}{2}\right)\|\Delta\bar{x}_k\|^2 + (\Delta_\gamma\|\tilde{y}_k\| + L\|\tilde{x}_k\|)\|\Delta\bar{x}_k\|. \tag{5.49}$$

Summing the above inequality over k from 0 to t, we have

$$\begin{aligned}f(\bar{x}_{t+1}) \leq f(\bar{x}_0) &- \left(\frac{1}{\gamma_{max}} - \frac{L}{2}\right)\sum_{k=0}^{t}\|\Delta\bar{x}_k\|^2 \\ &+ \Delta_\gamma\sum_{k=0}^{t}\|\tilde{y}_k\| \|\Delta\bar{x}_k\| + L\sum_{k=0}^{t}\|\tilde{x}_k\| \|\Delta\bar{x}_k\|.\end{aligned} \tag{5.50}$$

Using Cauchy-Schwarz inequality and recalling that $\Delta\bar{x}_k = -\overline{\gamma \odot y}_k$, we obtain

$$f(\bar{x}_{t+1}) \leq f(\bar{x}_0) - \left(\frac{1}{\gamma_{max}} - \frac{L}{2}\right)Z_t^2 + \Delta_\gamma Y_t Z_t + LX_t Z_t. \tag{5.51}$$

Suppose all the assumptions of Lemma 5.7 hold and $\beta < \frac{(1-\eta)^2}{(1+\Delta_\gamma)(2\eta^3+2\eta^2-\eta+1)}$ such that $\rho_1\rho_2 < 1$ and $\eta' < 1$, then invoking Lemma 5.7 we have

$$f(\bar{x}_{t+1}) \leq f(\bar{x}_0) - \mu Z_t^2 + \nu Z_t, \tag{5.52}$$

where

$$\begin{cases} \mu = \frac{1}{\gamma_{max}} - \frac{L}{2} - \frac{L(p_1+\rho_1 p_2)+\Delta_\gamma(p_2+\rho_2 p_1)}{1-\rho_1\rho_2} \\ \nu = \frac{L(q_1+\rho_1 q_2)+\Delta_\gamma(q_2+\rho_2 q_1)}{1-\rho_1\rho_2} \end{cases} \tag{5.53}$$

Let $u_t = f(\bar{x}(t)) - f^\star$. Then, since $u_t \geq 0, \forall t \geq 0$, (5.52) can be rewritten as

$$-\mu Z_t^2 + \nu Z_t + u_0 \geq 0. \tag{5.54}$$

Additionally, it is not difficult to show that $\mu > 0$ when $0 < \beta < \frac{b-\sqrt{b^2-4ac}}{2a}$ where

$$\begin{cases} a &= (1-\eta^2)(1-\eta)(1+\Delta_\gamma) \\ b &= 4\eta(\eta^2+1)\Delta_\gamma^2 + (4\eta^3 - 4\eta^2 + 6\eta + 2)\Delta_\gamma \\ &\quad + 4\eta^3 + 5\eta^2 - 4\eta + 3 \\ c &= 2(1-\eta)^2 \end{cases} \tag{5.55}$$

which further implies that

$$\lim_{t\to\infty} Z_t \leq Z_\infty = \frac{\nu + \sqrt{\nu^2 - 4\mu u_0}}{2\mu} < \infty. \tag{5.56}$$

Thus, applying the monotone convergence theorem to the above relation, we have $\lim_{k\to\infty} \|\overline{\gamma \odot y}_k\| = 0$. Also, from (5.45b) of Lemma 5.7 and (5.56), we know that

$$\lim_{k\to\infty} Y_k \leq Y_\infty \leq \frac{(\rho_2 p_1 + p_2)Z_\infty + (q_2 + \rho_2 q_1)}{(1-\rho_1\rho_2)} < \infty,$$

yielding $\lim_{k\to\infty} \|\tilde{y}_k\| = 0$. Together with Proposition 5.1-(iv) we further have

$$\lim_{k\to\infty} \|\bar{y}_k\| \leq \lim_{k\to\infty} \left(\frac{\sqrt{m}\,\|\overline{\gamma \odot y}_k\|}{\|\bar{\gamma}\|} + \Delta_\gamma \|\tilde{y}_k\| \right) = 0.$$

Likewise, using (5.45a) of Lemma 5.7 and (5.56) yields $\lim_{k\to\infty} \|\tilde{x}_k\| = 0$.

Since f is convex by Assumption 5.6 and Remark 5.11, for any $\bar{x}_k \in \mathcal{R}^m$ we have

$$\begin{aligned} f(\bar{x}_k) - f(x^\star) &\leq g(\bar{x}_k)^T(\bar{x}_k - x^\star) \\ &= g(x_k)^T(\bar{x}_k - x^\star) + (g(\bar{x}_k) - g(x_k))^T(\bar{x}_k - x^\star) \\ &= \bar{g}_k^T(\bar{x}_k - x^\star) + (g(\bar{x}_k) - g(x_k))^T(\bar{x}_k - x^\star) \\ &\leq \|\bar{g}_k\|\,\|\bar{x}_k - x^\star\| + L\,\|\tilde{x}_k\|\,\|\bar{x}_k - x^\star\|, \end{aligned} \tag{5.57}$$

where $x^\star$ is an optimum to the EDOP problem and we have used Proposition 5.1-(i) to obtain the first term of the second inequality.

Moreover, we know $f(\bar{x}_k)$ is bounded by virtue of (5.52) and (5.56), which implies that $\|\bar{x}_k - x^\star\|$ is also bounded since function f is coercive by Assumption 5.6. Thus, in view of (5.57) and recalling that $\bar{y}_k = \bar{g}_k$, we claim that

$$\lim_{k\to\infty} f(\bar{x}_k) = f(x^\star) = f^\star. \tag{5.58}$$

Further, by mean value theorem, we have

$$f(x_k) = f(\bar{x}_k) + g(\bar{x}_k + \xi\tilde{x}_k)^T\tilde{x}_k, \tag{5.59}$$

where $0 \leq \xi \leq 1$ is some positive number.

Since $\|\bar{x}_k + \xi\tilde{x}_k\| \leq \|\bar{x}_k\| + \xi\|\tilde{x}_k\|$ is bounded as shown above and g is Lipschitz continuous, thus $\|g(\bar{x}_k + \xi\tilde{x}_k)\|$ is also bounded. Then, from (5.59) and recalling that $\lim_{k\to\infty}\|\tilde{x}_k\| = 0$, we have

$$\lim_{k\to\infty} |f(x_k) - f(\bar{x}_k)| \leq \lim_{k\to\infty} \|g(\bar{x}_k + \xi\tilde{x}_k)\|\,\|\tilde{x}_k\| = 0. \tag{5.60}$$

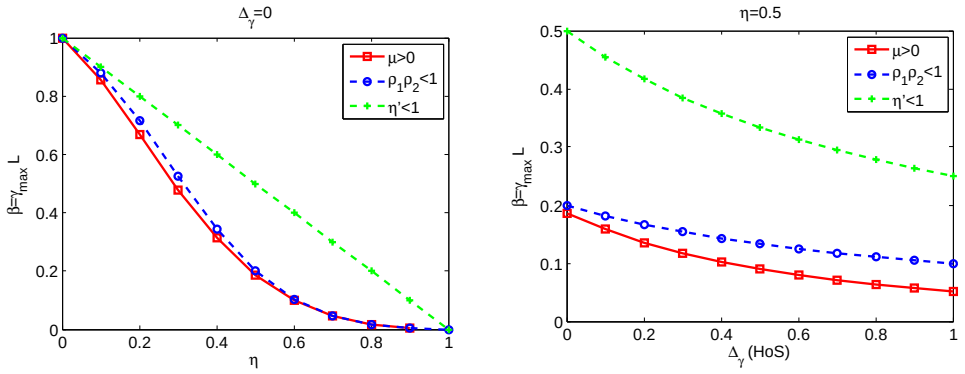
Combining (5.58) and (5.60) yields $\lim_{k\to\infty} f(x_k) = f^\star$, which completes the proof. □

Remark 5.13. The estimated theoretical upperbound of the stepsize is given in Equation (5.55). For illustration, Figure 5.1 plots the estimated upper bound of β with respect to the spectral radius η given certain value of Δ_γ and with respect to Δ_γ given certain value of η respectively to ensure certain conditions (i.e., $\eta' < 1$, $\rho_1\rho_2 < 1$ and $\mu > 0$) to be satisfied for convergence.

Corollary 5.3. *Consider the distributed algorithm 2 with $y_0 = g(x_0)$. Suppose all the assumptions of Theorem 5.2 hold and all agents use the same stepsize ($\Delta_\gamma = 0$). Then, there exists a positive number $\gamma^\star(\eta, \Delta_\gamma)/L$ such that if $\gamma_{\max} < \gamma^\star$, we have $\lim_{k\to\infty}\|x_k - \bar{x}_k\| = 0$ and $\lim_{k\to\infty} f(x_k) = f^\star$.*

Proof. It directly follows from Theorem 5.2 and noticing that $\Delta_\gamma = 0$.

Theorem 5.4. *Consider the distributed algorithm 2 with $y_0 = g(x_0)$. Let $\hat{x}_k = \frac{1}{t}\sum_{k=0}^{t-1} x_k$ be the running average and suppose all the assumptions of Theorem 5.2*



(A) β versus η with $\Delta_\gamma = 0$ (B) β versus Δ_γ with $\eta = 0.5$

FIGURE 5.1: Plots of the estimated upper bound of β to ensure certain conditions: $\eta' < 1$, $\rho_1\rho_2 < 1$ and $\mu > 0$ for convergence of the algorithm.

hold and all agents use the homogeneous (same) stepsize ($\Delta_\gamma = 0$). Then, there exists a positive number $\gamma^\star := \varphi(\eta, \Delta_\gamma)/L$ such that, if $\gamma_{\max} < \gamma^\star$, we have $\left\|\hat{x}_k - \hat{\bar{x}}_k\right\| \leq O(\frac{1}{\sqrt{k}})$ and $|f(\hat{x}_k) - f^\star| \leq O(\frac{1}{\sqrt{k}})$.

Proof. Consider the sequence (5.43). Let $x^\star \in \mathcal{R}$ be an optimum of the EDOP problem. Then, we have

$$\begin{aligned} \|\bar{x}_{k+1} - x^\star\|^2 &= \|\bar{x}_k - \overline{\gamma \odot y_k} - x^\star\|^2 \\ &\leq \|\bar{x}_k - x^\star\|^2 - 2\left\langle \overline{\gamma \odot y_k}, \bar{x}_k - x^\star \right\rangle + \|\overline{\gamma \odot y_k}\|^2 . \end{aligned} \tag{5.61}$$

Let us consider the middle term.

$$\begin{aligned} \left\langle \overline{\gamma \odot y_k}, \bar{x}_k - x^\star \right\rangle &\overset{(a)}{=} \left\langle \bar{\gamma} \odot \bar{y}_k + \overline{\tilde{\gamma} \odot \tilde{y}_k}, \bar{x}_k - x^\star \right\rangle \\ &\overset{(b)}{\geq} \frac{\|\bar{\gamma}\|}{\sqrt{m}} \left\langle \bar{y}_k, \bar{x}_k - x^\star \right\rangle \overset{(c)}{=} \frac{\|\bar{\gamma}\|}{\sqrt{m}} \left\langle g(x_k), \bar{x}_k - x^\star \right\rangle \\ &\overset{(d)}{\geq} \frac{\|\bar{\gamma}\|}{\sqrt{m}} \left(f(x_k) - f^\star\right) - \frac{\|\bar{\gamma}\|}{\sqrt{m}} \|g(x_k)\| \, \|\tilde{x}_k\| , \end{aligned} \tag{5.62}$$

where (a) is due to Prob. 5.1-(iii), (b) is clear since $\|\tilde{\gamma}\| = 0$, (c) is derived from Conversation Property III (cf. Lemma 5.6) and Prob. 5.1-(i) and (d) is obtained using the convexity of f and the Cauchy-Schwarz inequality.

Note that $\bar{x}_k$ is bounded with sufficiently small $\gamma_{\max}$ and so is x_k (cf. Theorem 5.2). Since g is continuous and thus bounded for any compact domain $\mathcal{D}$, we have $\|g(x)\| \leq C$, where C is certain positive number.

Then, combining (5.61) and (5.62) yields

$$\|\bar{x}_{k+1} - x^\star\|^2 \leq \|\bar{x}_k - x^\star\|^2 - \frac{2\,\|\bar{\gamma}\|}{\sqrt{m}}\,(f(x_k) - f^\star) + \frac{2\,\|\bar{\gamma}\|\,C}{\sqrt{m}}\,\|\tilde{x}_k\| + \|\overline{\gamma \odot y_k}\|^2\,, \tag{5.63}$$

Using convexity of f, we also have

$$|f(x_k) - f(\bar{x}_k)| \leq |\langle g(x_k), x_k - \bar{x}_k\rangle| \leq \|g(x_k)\|\,\|\tilde{x}_k\| \leq C\,\|\tilde{x}_k\|\,,$$

which further allows us to obtain

$$\begin{aligned} &f(x_k) - f^\star \\ &= f(x_k) - f(\bar{x}_k) + |f(x_k) - f(\bar{x}_k)| - |f(x_k) - f(\bar{x}_k)| + f(\bar{x}_k) - f^\star \\ &\geq |f(x_k) - f^\star| - 2\,|f(x_k) - f(\bar{x}_k)| \geq |f(x_k) - f^\star| - 2C\,\|\tilde{x}_k\|\,. \end{aligned} \tag{5.64}$$

Combining (5.63) and (5.64) and rearranging the terms we have

$$\begin{aligned} &\frac{2\,\|\bar{\gamma}\|}{\sqrt{m}}\,|f(x_k) - f^\star| \\ &\qquad \leq \|\bar{x}_k - x^\star\|^2 - \|\bar{x}_{k+1} - x^\star\|^2 + \frac{6\,\|\bar{\gamma}\|\,C}{\sqrt{m}}\,\|\tilde{x}_k\| + \|\overline{\gamma \odot y_k}\|^2\,. \end{aligned} \tag{5.65}$$

Summing the above inequality over k from 0 to $t-1$ leads to

$$\begin{aligned} \frac{2\,\|\bar{\gamma}\|}{\sqrt{m}} \sum_{k=0}^{t-1} |f(x_k) - f^\star| &\leq \|\bar{x}_0 - x^\star\|^2 \\ &\quad - \|\bar{x}_t - x^\star\|^2 + \frac{6\,\|\bar{\gamma}\|\,C}{\sqrt{m}} \sum_{k=0}^{t-1} \|\tilde{x}_k\| + \sum_{k=0}^{t-1} \|\overline{\gamma \odot y_k}\|^2\,. \end{aligned} \tag{5.66}$$

In addition, using the following Cauchy-Schwarz inequality

$$a_1 + a_2 + ... + a_m \leq \sqrt{m}\sqrt{a_1^2 + a_2^2 + ... + a_m^2},$$

where $a_1, a_2, ..., a_m$ are positive numbers, we have

$$\sum_{k=0}^{t-1} \|\tilde{x}_k\| \leq \sqrt{t}\sqrt{\sum_{k=0}^{t-1} \|\tilde{x}_k\|^2}. \tag{5.67}$$

Thus, dividing both sides of (5.66) by $\frac{2\|\bar{\gamma}\|}{\sqrt{m}}t$ we have

$$\frac{1}{t}\sum_{k=0}^{t-1}|f(x_k)-f^\star| \leq \frac{\sqrt{m}(Z_\infty + A_0)}{2\|\bar{\gamma}\|\, t} + \frac{3CX_\infty}{\sqrt{t}}. \tag{5.68}$$

where $A_0 = \|\bar{x}_0 - x^\star\|^2$.
Let $\hat{\bar{x}}_t = 1/t\sum_{k=0}^{t-1}\bar{x}_k$ be the running average. Using the convexity of f we have

$$|f(\hat{x}_t)-f^\star| \leq \frac{1}{t}\sum_{k=0}^{t-1}|f(x_k)-f^\star| \leq \frac{\sqrt{m}(Z_\infty + A_0)}{2\|\bar{\gamma}\|\, t} + \frac{3CX_\infty}{\sqrt{t}}. \tag{5.69}$$

Likewise, diving both sides of (5.67) by t we have

$$\frac{1}{t}\sum_{k=0}^{t-1}\|\tilde{x}_k\| \leq \frac{1}{\sqrt{t}}\sqrt{\sum_{k=0}^{t-1}\|\tilde{x}_k\|^2} \leq \frac{1}{\sqrt{t}}X_\infty. \tag{5.70}$$

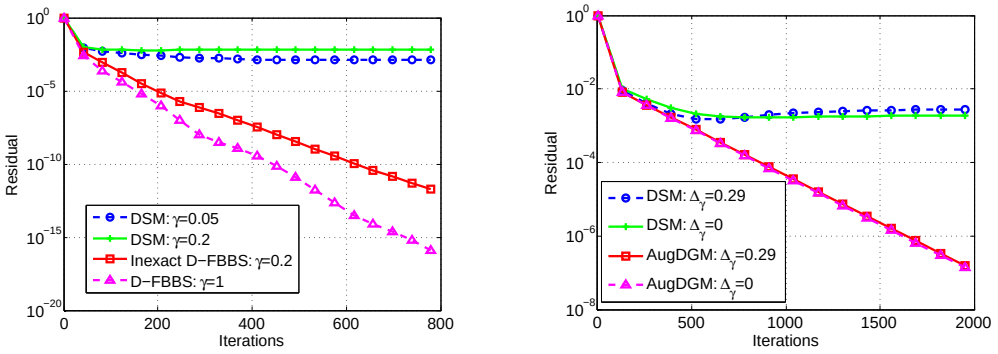
Let $\hat{\tilde{x}}_k = 1/t\sum_{k=0}^{t-1}\tilde{x}_k$. Again using the convexity of norm we have $\left\|\hat{x}_k - \hat{\tilde{x}}_k\right\| \leq \frac{1}{t}\sum_{k=0}^{t-1}\|\tilde{x}_k\| \leq \frac{1}{\sqrt{t}}X_\infty$. The rest of the proof follows from the fact that X_∞ and Z_∞ are bounded as previously shown in Theorem 5.2 with sufficiently small $\gamma_{\max}$.

5.5 Application to Sensor Fusion Problems

In this section, we report some simulations to show the effectiveness of the proposed algorithms over fixed networks. In particular, we consider a canonical distributed estimation problem. Each sensor is assumed to measure certain unknown parameter $\theta \in \mathcal{R}^d$ with some Gaussian noise ω_i, i.e., $z_i = M_i\theta + \omega_i$, where $M_i \in R^{r\times d}$ is the measurement matrix of sensor i and $z_i \in R^r$ is the measurement data collected by sensor i. Thus, the maximum likelihood estimation with regularization can be casted as the following minimization problem:

$$\theta^\star = \arg\min_{\theta\in\mathcal{R}^d}\left(\sum_{i=1}^{m}\|z_i - M_i\theta\|^2 + \lambda\|\theta\|^2\right) \tag{5.71}$$

where λ is the regularization parameter.



(A) D-FBBS Versus DSM (B) AugDGM Versus DSM

FIGURE 5.2: Comparison of the proposed algorithms with the best known DSM algorithm over a fixed network. (a) Plot of the relative FPR versus the number of iterations for DSM, *inexact* D-FBBS and D-FBBS respectively. (b) Plot of the relative FPR versus the number of iterations for DSM and AugDGM. The stepsizes for DSM and D-FBBS algorithms are optimized manually.

Parameter Setting: We set $d = 4, r = 1, m = 50$ for all algorithms. The measurement matrix is generated from a uniform distribution in the unit $R^{r\times d}$ space and the noise follows a i.i.d. Gaussian process with zero mean and certain variance $\mathcal{N}(0, 0.2)$. The regularization parameter is set as $\lambda = 0$ for both algorithms (note that we do not need the cost function to be strongly convex for the algorithm to converge for fixed networks). The weight matrix is designed using the simple rule in [89, 92], i.e., $W = I - \alpha L$ with $\alpha = \frac{1}{2+d_{\max}}$ for D-FBBS[6] and $\alpha = \frac{1}{2d_{\max}}$ for AugDGM, where $d_{\max}$ is the maximum degree of the communication graph. A stepsize of $\gamma = 0.05 \cdot \mathbf{1}$ and $\gamma = [0.26, 0.27, ..., 0.75]^T$ is used to simulate the case of homogeneous and heterogeneous computation respectively. We compare our results with DSM [27] in terms of the relative FPR $e = \frac{\|x_k - x^\star\|^2}{\|x_0 - x^\star\|^2}$.

Discussions: Figure 5.2a plots the relative FPR of D-FBBS, its inexact version and DSM with respect to the number of iterations for a fixed network employing constant stepsizes. It follows from the figure that all algorithms have similar convergence performance at the initial stage. However, DSM gets stuck after several iterations, which is not surprising as implied by the theoretical result [93], while the proposed D-FBBS and its *inexact* version still converge linearly to the optimum. Note that the *inexact* D-FBBS have comparable computational complexity with DSM as both are using gradient-based search.

[6]The value of α is such designed that the weight matrix is positive-definite.

Figure 5.2b shows the estimation results of the proposed AugDGM and DSM under homogeneous (same) stepsize and heterogeneous (different) stepsize rules respectively. Likewise, it follows from the figure that the performance of both algorithms are quite similar in the initial stage for both scenarios. However, DSM gets stuck after several iterations, resulting in an estimation error which is further enlarged under the heterogeneous setting, while AugDGM still progresses linearly to the exact optimum. It can also be observed that AugDGM performs almost the same under both scenarios, implying that it is robust to the heterogeneity of stepsize.

Remark 5.14. In our simulations, we found that the algorithm is still able to seek the exact optimum even using smaller γ than indicated by Theorem 6.1, which leads to faster convergence. This implies that the bound obtained is a bit conservative.

5.6 Summary

In this chapter, we have proposed two basic distributed algorithms, namely D-FBBS and AugDGM, to solve the general distributed estimation problem encountered in large-scale sensor networks where the communication graph is fixed and the algorithm is synchronously running. Both algorithms are in augmented form involving an extra step of consensus and shown to be able to seek the exact optimum even with constant stepsizes. In developing the D-FBBS algorithm, we introduced a distributed algorithm framework based on the Bregman method and operator splitting. This framework allows us to easily come up with efficient distributed algorithms for problems with certain structures. We have also established a non-ergodic convergence rate of $o(\frac{1}{k})$ in terms of FPR for the D-FBBS algorithm for general convex functions and an ergodic convergence rate of $O(\frac{1}{\sqrt{k}})$ in terms of OBE for the AugDGM algorithm employing homogeneous (same) stepsizes for coercive and convex functions with Lipschitz gradients. In addition, we have shown that the AugDGM algorithm is able to tackle with more general (balanced directed) graphs with heterogeneous computation (i.e., using different stepsizes), which, as we will see later, lends itself to asynchronous scenarios.

Chapter 6

Distributed Optimization in Sensor Networks: Stochastic Networks and Asynchronous Implementation

This chapter extends the two basic algorithms developed in the previous chapter to stochastic networks and asynchronous scenarios. In particular, we formulate the problem in Chapter 6.1 with emphasis on the stochastic modeling of the communication graph. We provide some preliminaries related to probability theory in Chapter 6.2. The application of D-FBBS to stochastic networks as well as its convergence analysis is made in Chapter 6.3. To further deal with asynchronous scenarios, we propose an asynchronous version of AugDGM in Chapter 6.4 where the asynchronous implementation model is also introduced with illustration.

6.1 Problem Statement

We consider the same EDOP problem as in the previous chapter except that we are now dealing with asynchronous scenarios over stochastic networks. In particular, we assume agents are communicating with each other through a random network captured by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In large-scale sensor networks, sensors are no longer guaranteed to be able to communicate with any agent at any time. Instead,

we assume each communication link is subject to random failures. That is, at each time tick, the communication link $e_{ij} \in \mathcal{E}$ will be active or deactive with probabilities p_{ij} and $1 - p_{ij}$ respectively. Once the link is being activated, the associated agent will be active as well. Thus, each agent is awake or asleep with probabilities $p_i = \sum_{j \in \mathcal{N}_i} p_{ij}$ and $1 - p_i$ respectively.

6.2 Preliminaries

6.2.1 Induced norm and its properties

Definition 6.1 (Inner Product and Induced Norm)**.** Given two random vectors $x, y \in \mathcal{R}^m$ and a square random matrix $A \in \mathcal{R}^{m \times m}$, the inner product is defined in expectation as $\langle x, y \rangle_E = E\left[\langle x, y \rangle\right] = E\left[x^T y\right]$, where $\langle \cdot, \cdot \rangle$ is the inner product in Euclidean space. In addition, we define the induced vector norm and matrix norm as $\|x\|_E = \sqrt{E\left[\|x\|^2\right]}$ and $\|A\|_E = \sup_{\|x\|_E = 1} \|Ax\|_E$ respectively.

Lemma 6.1. *Let $x, y \in \mathcal{R}^n$ be random vectors and $A_{n \times n}$ be a square random matrix. Then, we have the triangle inequality:*

$$\|x + y\|_E \leq \|x\|_E + \|y\|_E ,$$

and the Cauchy-Schwarz inequality:

$$\langle x, y \rangle_E \leq \|x\|_E \|y\|_E .$$

Further, if A is independent of x, we further have

$$\|A\|_E = \sqrt{\rho(E[A^T A])},$$

where $\rho(\cdot)$ is the spectral radius.

Proof. See Appendix B.

6.2.2 Convergence concepts in probability

The following convergence concepts are very important to the developed results.

Definition 6.2 (Mean Square Convergence)**.** X_n converges in mean square to the random variable X as $n \to \infty$ if $\|X_n - X\|_E \to 0$ as $n \to \infty$.

Definition 6.3 (Probability Convergence)**.** X_n converges in probability to the random variable X as $n \to \infty$ if $\forall \varepsilon > 0$, we have $P(|X_n - X|) > \varepsilon \to 0$ *as* $n \to \infty$.

Definition 6.4 (Almost Sure Convergence)**.** X_n converges almost surely $(a.s.)$ to the random variable X as $n \to \infty$ if $P(\{w : X_n(w) \to X(w)\}$ as $n \to \infty) = 1$.

6.2.3 Some basic inequalities and lemmas

The following inequalities are very crucial in developing the subsequent results.

Proposition 6.1 (Jensen's inequality)**.** *Let X be a random variable and f a convex function. Suppose that X and $g(X)$ are integrable. Then, $f(E[X]) \leq E[f(X)]$.*

Proposition 6.2 (Markov's inequality)**.** *Let X be the random variable. Then we have $P(|X| > \varepsilon) \leq \frac{E[|X|^p]}{\varepsilon^p}$.*

The following lemmas will be useful in showing the almost sure convergence result.

Lemma 6.2 (Theorem 4.4-(j), p.52 [94])**.** *Let X be a non-negative random variable. If $E[X] < \infty$ then $X < \infty$ a.s..*

Lemma 6.3 (Corollary 5.2, p.56 [94])**.** *Let $\{X_k\}_{k\geq 0}$ be a collection of non-negative random variables. Then $E[\sum_{k=0}^{\infty} X_k] = \sum_{k=0}^{\infty} E[X_k]$.*

Lemma 6.4 (Borel-Cantelli Lemma)**.** *Let $\{E_n, n \geq 1\}$ be arbitrary events. Then $\sum_{n=1}^{\infty} P(E_n) < \infty \Rightarrow P(\limsup A_n) = 0$.*

Lemma 6.5 (Robbins and Siegmung [95])**.** *Let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space and $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_k$ be a sequence of σ-subfields of $\mathcal{F}$. In addition, let v_k, a_k, w_k be non-negative random variables and let the following relation hold with probability one for any $k \geq 0$:*

$$E(v_{k+1}|\mathcal{F}_k) \leq (1 + a_k)v_k - u_k + w_k,$$

where $\sum_{k=0}^{\infty} a_k < \infty$ a.s., $\sum_{k=0}^{\infty} w_k < \infty$ a.s.. Then, the sequence $\{v_k\}_{k\geq 0}$ will converge to some random variable v a.s. and we further have $\sum_{i=0}^{\infty} u_k < \infty$ a.s..

6.3 Distributed Bregman Forward-Backward Splitting Algorithm

We show that under a stronger assumption on the cost functions, the previously proposed D-FBBS algorithm can be employed to solve the same distributed estimation problem over stochastic networks. Note that most of the analysis of the algorithm for fixed networks is not readily transferable to stochastic networks as the Lyapunov function employed therein is dependent on the network (i.e., the weight matrix W) thus varying with time. We have to find a new (common) Lyapunov function that is immune to varying (stochastic) networks. Bregman distance thus, as we will show, plays a key role in the subsequent convergence analysis.

6.3.1 D-FBBS algorithm for stochastic networks

For the EDOP problem to be feasible, we make the following assumption:

Assumption 6.1. Let $\{W_k\}_{k\geq 0}$ be the i.i.d. stochastic weight matrix sequence and $\overline{W} = E(W_k)$ be the mean. Then, the following conditions hold: $W_k^T = W_k,\ W_k > 0,\ W_k\mathbf{1} = \mathbf{1},\ \forall k \geq 0,$ and $\rho\left(\overline{W} - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) < 1$.

With the above assumption, similar with that of fixed networks (cf. Section 5.3.1), it is not difficult to see that the EDOP problem is equivalent to the following stochastic optimal consensus problem[1] (SOCP):

$$\min_{x\in\mathcal{R}^{md}} f(x) = \sum_{i=1}^{m} f_i(x_i) \qquad \mathbf{s.t.}\ (I - E\left[W_k^T W_k\right])x = 0 \tag{SOCP}$$

Assumption 6.2. The cost function f is strongly convex, i.e., $D_f^{\partial f(y)}(x,y) \geq \frac{m}{2}\|x-y\|^2$.

To solve the SOCP problem, we use the algorithm as follows:

$$\begin{aligned} x_{k+1} &= \arg\min_{x\in\mathcal{R}^m} \left(D_f^{y_k}(x, x_k) + \frac{1}{2\gamma}\|x - W_k x_k\|^2 \right) \\ y_{k+1} &= y_k - \frac{1}{\gamma}(I - W_k)x_{k+1}, \end{aligned} \tag{6.1}$$

[1]Note that, when $W_k^T = W_k, \forall k \geq 0$, we have $\rho(E\left[W_k^T W_k\right] - \frac{\mathbf{1}\mathbf{1}^T}{m}) = \rho^2(E\left[W_k\right] - \frac{\mathbf{1}\mathbf{1}^T}{m})$.

which is equivalent to

$$\gamma y_k - (x_{k+1} - W_k x_k) \in \gamma \partial f(x_{k+1}) \tag{6.2a}$$
$$(I - W_k)x_{k+1} + \gamma(y_{k+1} - y_k) = 0. \tag{6.2b}$$

As before, adding (6.2a) with (6.2b) yields:

$$\gamma y_{k+1} - W_k(x_{k+1} - x_k) \in \gamma \partial f(x_{k+1}) \tag{6.3a}$$
$$(I - W_k)x_{k+1} + \gamma(y_{k+1} - y_k) = 0. \tag{6.3b}$$

The proposed algorithm for stochastic networks is summarized in *Algorithm* 3.

Algorithm 3 D-FBBS for Stochastic Networks

1: **Initialization**: $y_{i,0} = 0, \forall i \in \mathcal{V}$ such that $\mathbf{1}^T y_0 = 0$, while the initial guess of x_0 can be arbitrarily assigned.
2: **Primal Update**: For each agent $i \in \mathcal{V}$, compute:

$$x_{i,k}^{av} = \sum_{j \in \mathcal{N}_i \cup \{i\}} w_{ij,k} x_{j,k}$$
$$x_{i,k+1} = \arg\min_{x_i \in \mathcal{R}^d} \left(D_f^{y_{i,k}}(x_i, x_{i,k}) + \frac{1}{2\gamma} \left\| x_i - x_{i,k}^{av} \right\|^2 \right)$$

3: **Dual Update**: For each agent $i \in \mathcal{V}$,

$$y_{i,k+1} = y_{i,k} - \frac{1}{\gamma} \sum_{j \in \mathcal{N}_i} w_{ij,k} \left(x_{i,k+1} - x_{j,k+1} \right)$$

4: Set k $\rightarrow$ k+1 and go to Step 2

6.3.2 Convergence analysis

We present the main result of the D-FBBS algorithm for stochastic networks.

Theorem 6.1. *Let* $\lambda_{max} = \max\{\{\lambda(W_k)\}_{k\geq 0}\} \in (0, 1]$ $\forall k \geq 0$ *and* $\lambda_{min} = \min\{\{\lambda(W_k)\}_{k\geq 0}\} \in (0, 1]$ $\forall k \geq 0$. *Suppose Assumptions 5.4, 6.1, 6.2 hold and*

$$\gamma > \frac{2(1-\mu)(1-\lambda_{min})}{m\mu} + \frac{2\lambda_{max}}{m},$$

where $\mu \in (0,1)$. *Then, the sequence* $\{(x_k, y_k)\}_{k\geq 0}$ *generated by the D-FBBS algorithm (6.1) will converge almost surely to the optimal solution of the SOCP problem. Moreover, let* $\hat{x}_k = \frac{1}{k}\sum_{t=0}^{k-1} x_t$ *and* $\hat{\bar{x}}_k = \frac{1}{k}\sum_{t=0}^{k-1} \bar{x}_t$ *be the running average of* x_t *and* $\bar{x}_t$ *respectively. Then, the fixed point residual in terms of* $E\left[\|\hat{x}_{k+1} - \hat{x}_k\|^2\right]$ *and* $E\left[\left\|\hat{x}_k - \hat{\bar{x}}_k\right\|^2_{I-\overline{W}}\right]$ *will decrease at an ergodic rate of* $O(\frac{1}{k})$.

Proof. Let us first consider an interesting recursive relation as follows:

$$\begin{aligned} &D_{\gamma f}^{q_{k+1}}(x, x_{k+1}) - D_{\gamma f}^{q_k}(x, x_k) + D_{\gamma f}^{q_k}(x_{k+1}, x_k) \\ &\qquad = \gamma f(x) - \gamma f(x_{k+1}) - \langle q_{k+1}, x - x_{k+1}\rangle - \gamma f(x) + \gamma f(x_k) \\ &\qquad\quad + \langle q_k, x - x_k\rangle + \gamma f(x_{k+1}) - \gamma f(x_k) \\ &\qquad\quad - \langle q_k, x_{k+1} - x_k\rangle = \langle q_{k+1} - q_k, x_{k+1} - x\rangle . \end{aligned} \tag{6.4}$$

Let $q_k = \gamma y_k - W_k(x_k - x_{k-1})$. Using (6.3b) we have

$$\begin{aligned} q_{k+1} - q_k &= \gamma(y_{k+1} - y_k) - W_k(x_{k+1} - x_k) + W_{k-1}(x_k - x_{k-1}) \\ &= -(I - W_k)x_{k+1} - W_k(x_{k+1} - x_k) + W_{k-1}(x_k - x_{k-1}). \end{aligned} \tag{6.5}$$

Let $\bar{x} \in span\{\mathbf{1}\}$. Combining (6.4) and (6.5) yields

$$\begin{aligned} &D_{\gamma f}^{q_{k+1}}(\bar{x}, x_{k+1}) - D_{\gamma f}^{q_k}(\bar{x}, x_k) + D_{\gamma f}^{q_k}(x_{k+1}, x_k) \\ &= \langle -W_k(x_{k+1} - x_k) + W_{k-1}(x_k - x_{k-1}), x_{k+1} - \bar{x}\rangle - \|x_{k+1} - \bar{x}\|^2_{I-W_k} \\ &= -\|x_{k+1} - \bar{x}\|^2_{I-W_k} - \langle W_k(x_{k+1} - x_k), x_{k+1} - \bar{x}\rangle \\ &\quad + \langle W_{k-1}(x_k - x_{k-1}), x_k - \bar{x}\rangle + \langle W_{k-1}(x_k - x_{k-1}), x_{k+1} - x_k\rangle \\ &\leq -\|x_{k+1} - \bar{x}\|^2_{I-W_k} - \langle W_k(x_{k+1} - x_k), x_{k+1} - \bar{x}\rangle + \langle W_{k-1}(x_k - x_{k-1}), x_k - \bar{x}\rangle \\ &\quad - \|x_{k+1} - x_k\|^2_{\frac{W_k}{2}} + \|x_k - x_{k-1}\|^2_{\frac{W_{k-1}}{2}} + \|x_{k+1} - x_k\|^2_{\frac{W_k + W_{k-1}}{2}} , \end{aligned} \tag{6.6}$$

where we have used the following inequality to obtain the third relation:

$$\begin{aligned} \langle W_{k-1}(x_k - x_{k-1}), x_{k+1} - x_k\rangle &\leq \|x_{k+1} - x_k\|^2_{\frac{W_{k-1}}{2}} + \|x_k - x_{k-1}\|^2_{\frac{W_{k-1}}{2}} \\ &= \|x_{k+1} - x_k\|^2_{\frac{W_k + W_{k-1}}{2} - \frac{W_k}{2}} + \|x_k - x_{k-1}\|^2_{\frac{W_{k-1}}{2}} . \end{aligned} \tag{6.7}$$

Let $V_{k+1} = D_{\gamma f}^{q_{k+1}}(\bar{x}, x_{k+1}) + \langle W_k(x_{k+1} - x_k), x_{k+1} - \bar{x}\rangle + \|x_{k+1} - x_k\|^2_{\frac{W_k}{2}}$, which is

positive if $\gamma > \frac{\lambda_{max}}{m}$ since

$$\begin{aligned} V_{k+1} &= D_{\gamma f}^{q_{k+1}}(\bar{x}, x_{k+1}) - \|x_{k+1} - \bar{x}\|_{\frac{W_k}{2}}^2 + \left\|x'_{k+1} - \bar{x}\right\|_{\frac{W_k}{2}}^2 \\ &\geq \|x_{k+1} - \bar{x}\|_{\frac{m\gamma I - W_k}{2}}^2 \geq \|x_{k+1} - \bar{x}\|_{\frac{m\gamma - \lambda_{max}}{2} I}^2 > 0, \end{aligned} \tag{6.8}$$

where $x'_{k+1} = 2x_{k+1} - x_k$ and we have used Assumption 6.2 to obtain

$$D_{\gamma f}^{q_k}(x_{k+1}, \bar{x}) \geq \frac{\gamma m}{2} \|x_{k+1} - \bar{x}\|^2 .$$

Then, (6.6) can be rewritten as

$$D_{\gamma f}^{q_k}(x_{k+1}, x_k) - \|x_{k+1} - x_k\|_{\frac{W_k + W_{k-1}}{2}}^2 + \|x_{k+1} - \bar{x}\|_{I - W_k}^2 \leq V_k - V_{k+1}. \tag{6.9}$$

Summing (6.9) over k from 0 through $t-1$ leads to

$$\begin{aligned} \sum_{k=0}^{t-1} \left(D_{\gamma f}^{q_k}(x_{k+1}, x_k) - \|x_{k+1} - x_k\|_{\frac{W_k + W_{k-1}}{2}}^2 \right) \\ + \sum_{k=0}^{t-1} \|x_{k+1} - \bar{x}\|_{I - W_k}^2 \leq V_0 - V_t \leq V_0 < \infty. \end{aligned} \tag{6.10}$$

Applying the basic inequaltiy in G-space $\|a+b\|_G^2 \geq (1-\frac{1}{\mu}) \|a\|_G^2 + (1-\mu) \|b\|_G^2$, $\forall \mu \geq 0$, we have

$$\|x_{k+1} - \bar{x}\|_{I - W_k}^2 \geq (1 - \frac{1}{\mu}) \|x_{k+1} - x_k\|_{I - W_k}^2 + (1 - \mu) \|x_k - \bar{x}\|_{I - W_k}^2 , \tag{6.11}$$

where we require $\mu \in (0, 1)$. Thus, we have

$$\sum_{k=0}^{t-1} \left(D_{\gamma f}^{q_k}(x_{k+1}, x_k) - \|x_{k+1} - x_k\|_{(\lambda_{max} + \rho) I}^2 \right) + \sum_{k=0}^{t-1} (1 - \mu) \|x_k - \bar{x}\|_{I - W_k}^2 \leq V_0 < \infty, \tag{6.12}$$

where $\rho = \frac{(1-\mu)(1-\lambda_{min})}{\mu}$. Knowing that W_k is independent of the past states, taking total expectation yields

$$\sum_{k=0}^{t-1} E \left[\|x_{k+1} - x_k\|_{(\frac{\gamma m}{2} - \lambda_{max} - \rho) I}^2 \right] + \sum_{k=0}^{t-1} (1 - \mu) E \left[\|x_k - \bar{x}_k\|_{I - \overline{W}}^2 \right] \leq V_0 < \infty, \tag{6.13}$$

where we have used the fact that $\|x_k - \bar{x}\|_{I - \overline{W}}^2 = \|x_k - \bar{x}_k\|_{I - \overline{W}}^2$ (cf. Assumption 6.1) to replace the second term. Suppose $\gamma > \frac{2(\lambda_{max} + \rho)}{m}$ and let $t \to \infty$. Then,

by Markov's inequality [94], using $(a+b)^2 < 2a^2 + 2b^2$ we have for any $\varepsilon > 0$

$$\begin{aligned}
&\sum_{k=0}^{\infty} P(\|x_{k+1} - x_k\|_{(\frac{\gamma m}{2} - \lambda_{max} - \rho)I} + \sqrt{(1-\mu)}\,\|x_k - \bar{x}_k\|_{I-\overline{W}} > \varepsilon) \\
&\leq \frac{2\sum_{k=0}^{\infty} E\left[\|x_{k+1} - x_k\|^2_{(\frac{\gamma m}{2} - \lambda_{max} - \rho)I}\right] + 2\sum_{k=0}^{\infty}(1-\mu)E\left[\|x_k - \bar{x}_k\|^2_{I-\overline{W}}\right]}{\varepsilon^2} \\
&\leq \frac{2V_0}{\varepsilon^2} < \infty.
\end{aligned} \tag{6.14}$$

Thus, by Borel-Cantelli Lemma and Prop. 1.2 [94, p.206], we have $\lim_{k\to\infty} x_{k+1} = x_k$ and $\lim_{k\to\infty} x_k = \bar{x}_k$ with probability one. In addition, from (6.9) we know that V_k is bounded and so is x_k by (6.8). Thus, by standard analysis of weak cluster points, we claim that the sequence $\{x_k\}_{k\geq 0}$ will converge almost surely to some value $x^\star \in span\{\mathbf{1}\}$. Then, invoking Lemma 5.4, we conclude that the sequence $\{x_k\}_{k\geq 0}$ will converge almost surely to the optimal solution of the SOCP problem.

Moreover, let $\hat{x}_t = \frac{1}{t}\sum_{k=0}^{t-1} x_k$. Multiplying both sides of (6.13) by $\frac{1}{t}$ and using the Jensen' inequality yields

$$E\left[\left\|\hat{x}_{t+1} - \hat{x}_t + \frac{1}{t}(\hat{x}_{t+1} - x_0)\right\|^2_{(\frac{\gamma m}{2} - \lambda_{max} - \rho)I}\right] + (1-\mu)E\left[\left\|\hat{x}_t - \hat{\bar{x}}_t\right\|^2_{I-\overline{W}}\right] \leq \frac{V_0}{t}, \tag{6.15}$$

where $\hat{\bar{x}}_t = \frac{1}{t}\sum_{k=0}^{t-1}\bar{x}_k$. Again, using the basic inequality $\|a+b\|^2 \geq (1-\frac{1}{\nu})\|a\|^2 + (1-\nu)\|b\|^2$, $\forall \nu \geq 0$, we have

$$\begin{aligned}
&E\left[(1-\nu)\|\hat{x}_{t+1} - \hat{x}_t\|^2_{(\frac{\gamma m}{2} - \lambda_{max} - \rho)I}\right] + (1-\mu)E\left[\left\|\hat{x}_t - \hat{\bar{x}}_t\right\|^2_{I-\overline{W}}\right] \\
&\qquad \leq \frac{V_0}{t} + \frac{1}{t^2}(\frac{1}{\nu} - 1)\|\hat{x}_{t+1} - x_0\|^2_{(\frac{\gamma m}{2} - \lambda_{max} - \rho)I},
\end{aligned} \tag{6.16}$$

where we require $\nu \in (0,1)$. Since x_t is a convergent sequence and so is $\hat{x}_t$, thus $\hat{x}_t$ is uniformly bounded, implying that the last term of the above relation is of $O(\frac{1}{t^2})$. It follows then that the fixed point residual in terms of the running average $\hat{x}_t$ will decrease at an ergodic rate of $O(\frac{1}{t})$.

6.4 Asynchronous Distributed Gradient Methods

As we have shown previously in Chapter 5, the D-FBBS algorithm, though permitting better convergence rate, heavily depends on the topology of the network, restricting its application to asynchronous implementation[2] or being very complex in analysis for asynchronous scenarios. In this section, we show that the asynchronous version of AugDGM is, in fact, not only capable of dealing with stochastic networks but also, most importantly, able to account for asynchronous implementation.

6.4.1 Asynchronous implementation

In distributed algorithms, due to the absent of global clocks or synchronization mechanisms, it is very common that different agents will end up with acting very differently, resulting in heterogeneous (asynchronous) issues. In asynchronous implementation, agents are only to act when they are activated (i.e., awake) while keeping idle when not (i.e., asleep). As a result, it is very likely that some may execute more iterations and communicate more frequently than others.

We assume there exists a virtual global clock that ticks whenever any local clock ticks. Let T_k be the time of k-th tick of the virtual global clock. Then, at each tick T_k, a subset of agents will be activated and each of them attempts to give the best estimate of the global optimum based on its local observation and the data received from its immediate neighbors. Thus, during the interval $[T_k, T_{k+1})$, agents not only need to update their local estimates but also have to make consensus with its neighbors in order to achieve consistency on the estimates (cf. Figure 6.1). The algorithm can be totally asynchronous and goes through the below three phases:

- *Initial Phase*: proper initialization,
- *Action Phase*: at each tick of virtual global clock, a subset of agents being activated will receive the estimates from their active neighbors, update their own estimates and then broadcast the new value to the neighbors,
- *Idle Phase*: doing nothing at this stage.

[2]Note that it differs from the randomized Gauss-Seidel iteration [65] in that the latter requires the weight matrix to be the same all the time.

In action phase, each agent carries out several local steps, such as local communication for consensus and local update for optimization (cf. Section 5.4).

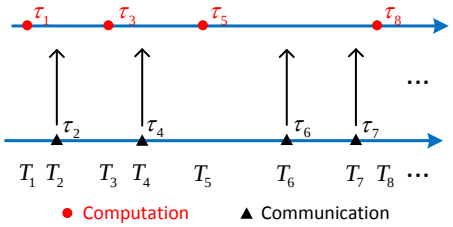


FIGURE 6.1: An illustration of asynchronous implementation of distributed algorithms.

6.4.2 AsynDGM algorithm for stochastic networks

We use the asynchronous version of AugDGM (termed AsynDGM) to solve the problem over stochastic networks under asynchronous implementation. In particular, we use the following algorithm to solve the SOCP problem:

$$x_{k+1} = W_k \left[x_k - \gamma_k \odot y_k\right] \tag{6.17a}$$

$$y_{k+1} = W_k y_k + \Delta g_k, \tag{6.17b}$$

where x_k and y_k are the collective vectors, $\Delta g_k = g(x_{k+1}) - g(x_k)$ is the incremental change of the gradients and $\gamma_k = [\gamma_{1,k}, \gamma_{2,k}, ..., \gamma_{m,k}]^T$ is the vector of stepsize with the component being zero when the corresponding agent is inactive.

Remark 6.1. Since we are focused on asynchronous implementation, W_k and γ_k are dependent. That is, for any $i \in \mathcal{V}$, agent i only carries out the above computation steps when there is communication involved, i.e., $w_{ii} \neq 1$. It is not difficult to see that if $\gamma_{i,k} = 0$, then $x_{i,k+1} = x_{i,k}$ and so is $y_{i,k+1} = y_{i,k}$, corresponding to the inactivity of the agent i. Indeed, the stepsize $\gamma_{i,k}$ chosen by agent i at time k is a random variable following Bernoulli process.

Let $\mathcal{V}_k^+$ denote the set of agents being activated at time k while $\mathcal{V}_k^-$ the set of deactivated agents. We summarize the proposed AsynDGM in *Algorithm* 4.

Different from the D-FBBS algorithm, we make the following assumption on the weight matrix which is less restrictive than Assumption 6.1:

Algorithm 4 AsynDGM for Stochastic Networks

1: **Initialization**: $\forall$ agent $i \in \mathcal{V}$: $x_{i,0}$ randomly assigned while $y_{i,0} = g(x_{i,0})$.
2: **Local Optimization**: $\forall$ agent $i \in \mathcal{V}_k^+$, computes:

$$\begin{aligned} s_{i,k} &= x_{i,k} - \gamma_{i,k} \cdot y_{i,k} \\ x_{i,k+1} &= s_{i,k} + \sum_{j \in \mathcal{N}_i \cap \mathcal{V}_k^+} w_{ij,k}(s_{j,k} - s_{i,k}) \end{aligned} \tag{6.18}$$

and set $x_{i,k+1} = x_{i,k}$ $\forall$ agent $i \in \mathcal{V}_k^-$.
3: **Dynamic Average Consensus**: $\forall$ agent $i \in \mathcal{V}_k^+$, computes:

$$y_{i,k+1} = y_{i,k} + \sum_{j \in \mathcal{N}_i \cap \mathcal{V}_k^+} w_{ij,k}\left(y_{j,k} - y_{i,k}\right) + \Delta g_{i,k} \tag{6.19}$$

and set $y_{i,k+1} = y_{i,k}$ $\forall$ agent $i \in \mathcal{V}_k^-$.
4: Set k $\rightarrow$ k+1 and go to Step 2.

Assumption 6.3. The weight matrix is drawn i.i.d. from a probability space $\mathcal{F} = (\Omega, \mathcal{B}, \mathcal{P})$ such that each W_k is doubly stochastic and $E(W_k^T W_k)$ has the second largest eigenvalue *strictly* less than 1, i.e., $\forall k \geq 0$,

$$\mathbf{1}^T W_k = \mathbf{1}^T, \ W_k \mathbf{1} = \mathbf{1}, \tag{6.20a}$$

$$\eta^2 = \rho\left(E(W_k^T W_k) - \frac{\mathbf{1}\mathbf{1}^T}{m}\right) < 1. \tag{6.20b}$$

Further, we use $\mathcal{F}_k$ to denote the σ-algebra generated by the entire history of weight matrix as well as the random initialization, i.e., for $k \geq 1$,

$$\mathcal{F}_k = \{x_0; W_i, 1 \leq i \leq k-1\}.$$

Remark 6.2. The above consensus protocol should be regarded as gossip-like protocols as multiple links will be activated simultaneously at certain time tick [25].

Similar with that of fixed networks (cf. Definition 5.7), we introduce a new important parameter that quantifies the variation of the stepsizes used by agents over time, i.e., how "asynchronous" the algorithm is running.

Definition 6.5 (Heterogeneity of Stepsize II)**.** Let γ be the vector of the stepsizes sampled from a probability space by the agents. The heterogeneity of stepsize

(HoS) is defined as

$$\Delta_\gamma = \frac{\|\tilde{\gamma}\|_E}{\|\bar{\gamma}\|_E},$$

where $\bar{\gamma} = \Pi_{\|}\gamma$ is the average vector and $\tilde{\gamma} = \gamma - \bar{\gamma}$ is the deviation vector. If all agents are using the same stepsize all the time, then $\Delta_\gamma = 0$, corresponding to synchronous running of the algorithm.

6.4.3 Basic convergence analysis

The following analysis is carried out based on the approximate average analogue of the algorithm as well as the approximation error between the distributed algorithm and its centralized counterpart.

Lemma 6.6. *Consider the distributed algorithm 4 and suppose Assumption 5.7 and 6.3 hold. Let* $X_k^e = \sqrt{\sum_{i=0}^k \|\tilde{x}_i\|_E^2}, Y_k^e = \sqrt{\sum_{i=0}^k \|\tilde{y}_i\|_E^2}, Z_k^e = \sqrt{\sum_{i=0}^k \|\bar{y}_i\|_E^2}$ *be the expected "energy" from 0 to k,* $\beta = \gamma_{\max} L$ *and* $\eta' = \eta + \beta(1 + \Delta_\gamma)$. *If* $\beta < \frac{(1-\eta)^2}{3+\eta+\Delta_\gamma(1-\eta)}$ *such that* $\rho_1\rho_2 < 1$ *and* $\eta' < 1$, *then we have*

$$X_k^e \leq \frac{\rho_1 p_2 + p_1}{1 - \rho_1\rho_2} Z_k^e + \frac{q_1 + \rho_1 q_2}{1 - \rho_1\rho_2} \tag{6.21a}$$

$$Y_k^e \leq \frac{\rho_2 p_1 + p_2}{1 - \rho_1\rho_2} Z_k^e + \frac{q_2 + \rho_2 q_1}{1 - \rho_1\rho_2}, \tag{6.21b}$$

where $\rho_1 = \frac{\sqrt{2}\gamma_{\max}}{1-\eta}$, $p_1 = \frac{\sqrt{2}\Delta_\gamma\bar{\sigma}_\gamma}{(1-\eta)\sqrt{m}}$, $q_1 = \frac{\sqrt{2}\|\tilde{x}_0\|_E}{\sqrt{1-\eta^2}}$, *and* $\rho_2 = \frac{\sqrt{2}(1+\eta)L}{1-\eta'}$, $p_2 = \frac{\sqrt{2}(1+\Delta_\gamma)L\bar{\sigma}_\gamma}{(1-\eta')\sqrt{m}}$, $q_2 = \frac{\sqrt{2}\|\tilde{y}_0\|_E}{\sqrt{1-\eta'^2}}$.

Proof. See Appendix B.

The following theorem is one of our main results that shows the basic convergence of the proposed algorithm.

Theorem 6.2. *Consider the distributed algorithm 4 with* $y_0 = g(x_0)$ *and suppose Assumptions 5.1, 5.6, 5.7 and 6.3 hold. Then, there exists a positive number* $\gamma^\star := \varphi(\eta, \Delta_\gamma)/L$ *such that, if* $\gamma_{\max} < \gamma^\star$, *we have* $\lim_{k\to\infty} \|x_k - \bar{x}_k\| = 0$ *a.s. and* $\lim_{k\to\infty} f(x_k) = f^\star$ *a.s., where* $f^\star$ *is the optimal value of the SOCP Problem.*

Proof. Consider the sequence (5.43). Since f has Lipschitz gradient by Assumption 5.7, we have for $\forall x, x' \in \mathcal{R}^m$

$$f(x') \leq f(x) + \langle g(x), x' - x\rangle + \frac{L}{2}\|x' - x\|^2.$$

Taking conditional expectation on $\mathcal{F}_k$ and plugging $x' = \bar{x}_{k+1}$ and $x = \bar{x}_k$ into the above relation yields

$$\begin{aligned} E\left[f(\bar{x}_{k+1})|\mathcal{F}_k\right] &\leq f(\bar{x}_k) - E\left[\langle g(\bar{x}_k), \overline{\gamma_k \odot y_k}\rangle \,|\mathcal{F}_k\right] + \frac{L}{2} E\left[\|\overline{\gamma_k \odot y_k}\|^2 \,|\mathcal{F}_k\right] \\ &= f(\bar{x}_k) - E\left[\langle g(x_k), \overline{\gamma_k \odot y_k}\rangle \,|\mathcal{F}_k\right] + \frac{L}{2} E\left[\|\overline{\gamma_k \odot y_k}\|^2 \,|\mathcal{F}_k\right] \\ &\quad - E\left[\langle g(\bar{x}_k) - g(x_k), \overline{\gamma_k \odot y_k}\rangle \,|\mathcal{F}_k\right] \\ &= f(\bar{x}_k) - E\left[\langle \bar{y}_k, \overline{\gamma_k \odot y_k}\rangle \,|\mathcal{F}_k\right] + \frac{L}{2} E\left[\|\overline{\gamma_k \odot y_k}\|^2 \,|\mathcal{F}_k\right] \\ &\quad - E\left[\langle g(\bar{x}_k) - g(x_k), \overline{\gamma_k \odot y_k}\rangle \,|\mathcal{F}_k\right], \end{aligned} \tag{6.22}$$

where we have used Proposition 5.1-(i) and Conservation Property III (cf. Lemma 5.6) to obtain the last inequality.
Consider the second term. Let $\bar{\mu}_\gamma = E\left[\|\bar{\gamma}_k\|\right], \tilde{\mu}_\gamma = E\left[\|\tilde{\gamma}_k\|\right]$. Recall that γ_k is i.i.d. and independent of $\mathcal{F}_k$ (cf. Assumption 6.3). Then, using Prop. 5.1-(iv) we have

$$E\left[\langle \bar{y}_k, \overline{\gamma_k \odot y_k}\rangle \,|\mathcal{F}_k\right] = \|\bar{y}_k\| \, E\left[\|\overline{\gamma_k \odot y_k}\| \,|\mathcal{F}_k\right] \geq \frac{1}{\sqrt{m}} \left(\bar{\mu}_\gamma \|\bar{y}_k\|^2 - \tilde{\mu}_\gamma \|\bar{y}_k\| \, \|\tilde{y}_k\|\right). \tag{6.23}$$

For the third term, taking square root we have

$$\begin{aligned} \sqrt{E\left[\|\overline{\gamma_k \odot y_k}\|^2 \,|\mathcal{F}_k\right]} &= \sqrt{E\left[\left\|\bar{\gamma}_k \odot \bar{y}_k + \overline{\tilde{\gamma}_k \odot \tilde{y}_k}\right\|^2 |\mathcal{F}_k\right]} \\ &\leq \sqrt{E\left[\|\bar{\gamma}_k \odot \bar{y}_k\|^2 \,|\mathcal{F}_k\right]} + \sqrt{E\left[\left\|\overline{\tilde{\gamma}_k \odot \tilde{y}_k}\right\|^2 |\mathcal{F}_k\right]} \\ &\leq \frac{1}{\sqrt{m}} \left(\bar{\sigma}_\gamma \|\bar{y}_k\| + \tilde{\sigma}_\gamma \|\tilde{y}_k\|\right). \end{aligned} \tag{6.24}$$

Now, let us consider the last deviate term. By Assumption 5.7 and using Cauchy-Schwarz inequality, we obtain

$$\begin{aligned} E\left[\langle g(\bar{x}_k) - g(x_k), \overline{\gamma_k \odot y_k}\rangle \,|\mathcal{F}_k\right] &\leq E\left[\|g(\bar{x}_k) - g(x_k)\| \, \|\overline{\gamma_k \odot y_k}\| \,|\mathcal{F}_k\right] \\ &\leq L \|\tilde{x}_k\| \, E\left[\|\overline{\gamma_k \odot y_k}\| \,|\mathcal{F}_k\right] \leq L \|\tilde{x}_k\| \sqrt{E\left[\|\overline{\gamma_k \odot y_k}\|^2 \,|\mathcal{F}_k\right]} \\ &\leq \frac{L}{\sqrt{m}} \left(\bar{\sigma}_\gamma \|\tilde{x}_k\| \, \|\bar{y}_k\| + \tilde{\sigma}_\gamma \|\tilde{x}_k\| \, \|\tilde{y}_k\|\right). \end{aligned} \tag{6.25}$$

Combining (6.22)–(6.25) and letting $v_t = f(\bar{x}_t) - f^\star$ leads to

$$E\left[v_{k+1}|\mathcal{F}_k\right] \leq v_k - \frac{1}{\sqrt{m}}\left(\bar{\mu}_\gamma \left\|\bar{y}_k\right\|^2 - \tilde{\mu}_\gamma \left\|\bar{y}_k\right\| \left\|\tilde{y}_k\right\|\right) + \frac{L}{m}\left(\bar{\sigma}_\gamma^2 \left\|\bar{y}_k\right\|^2 + \tilde{\sigma}_\gamma^2 \left\|\tilde{y}_k\right\|^2\right) + \frac{L}{\sqrt{m}}\left(\bar{\sigma}_\gamma \left\|\tilde{x}_k\right\| \left\|\bar{y}_k\right\| + \tilde{\sigma}_\gamma \left\|\tilde{x}_k\right\| \left\|\tilde{y}_k\right\|\right). \tag{6.26}$$

Taking the total expectation and summing the above inequality over k from 0 to t, rearranging terms we have

$$E\left[v_{k+1}\right] \leq E\left[v_0\right] - \left(\frac{\bar{\mu}_\gamma}{\sqrt{m}} - \frac{\bar{\sigma}_\gamma^2 L}{m}\right)\sum_{k=0}^{t} E\left[\left\|\bar{y}_k\right\|^2\right] + \frac{\tilde{\mu}_\gamma}{\sqrt{m}}\sum_{k=0}^{t} E\left[\left\|\bar{y}_k\right\| \left\|\tilde{y}_k\right\|\right] + \frac{\tilde{\sigma}_\gamma^2 L}{m}\sum_{k=0}^{t} E\left[\left\|\tilde{y}_k\right\|^2\right] + \frac{\bar{\sigma}_\gamma L}{\sqrt{m}}\sum_{k=0}^{t} E\left[\left\|\tilde{x}_k\right\| \left\|\bar{y}_k\right\|\right] + \frac{\tilde{\sigma}_\gamma L}{\sqrt{m}}\sum_{k=0}^{t} E\left[\left\|\tilde{x}_k\right\| \left\|\tilde{y}_k\right\|\right]. \tag{6.27}$$

Using Cauchy-Schwarz inequality (cf. Lemma 6.1) yields

$$E\left[v_{t+1}\right] \leq E\left[v_0\right] - aZ_t^{e2} + bZ_t^e Y_t^e + cY_t^{e2} + dX_t^e Z_t^e + eX_t^e Y_t^e, \tag{6.28}$$

where

$$a = \frac{\bar{\mu}_\gamma}{\sqrt{m}} - \frac{\bar{\sigma}_\gamma^2 L}{m},\ b = \frac{\tilde{\mu}_\gamma}{\sqrt{m}},\ c = \frac{\tilde{\sigma}_\gamma^2 L}{m},\ d = \frac{\bar{\sigma}_\gamma L}{\sqrt{m}},\ e = \frac{\tilde{\sigma}_\gamma L}{\sqrt{m}}.$$

Let us first consider the first term. Recalling that $\bar{\mu}_\gamma = E\left[\left\|\bar{\gamma}_k\right\|\right]$, we have

$$a \geq \frac{1}{\sqrt{m}} E\left[\frac{\left\|\bar{\gamma}_k\right\|}{\sqrt{m}\gamma_{\max}}\left\|\bar{\gamma}_k\right\|\right] - \frac{L}{m}\bar{\sigma}_\gamma^2 \geq \left(\frac{1}{\beta} - 1\right) R,$$

where $\beta = \gamma_{\max} L$ and $R = \frac{\bar{\sigma}_\gamma^2 L}{m}$.

For the second term, using Cauchy-Schwarz inequality yields

$$b = \frac{\tilde{\mu}_\gamma}{\sqrt{m}} \leq \frac{\tilde{\sigma}_\gamma}{\sqrt{m}} = \frac{\bar{\sigma}_\gamma \Delta_\gamma}{\sqrt{m}}.$$

Suppose $\beta < \frac{(1-\eta)^2}{3+\eta+\Delta_\gamma(1-\eta)}$ such that $\rho_1\rho_2 < 1$ and $\eta' < 1$. Since Assumptions 5.7 and 6.3 hold, invoking Lemma 6.6 we have

$$bZ_t^e Y_t^e \leq b_1 Z_t^{e2} + b_2 Z_t^e,$$

where

$$b_1 = \frac{2\Delta_\gamma(1+\eta) + \sqrt{2}\Delta_\gamma(1+\Delta_\gamma)(1-\eta)}{(1-\eta)(1-\eta') - 2(1+\eta)\beta} R,\ b_2 > 0.$$

Similarly, applying the above analysis to the rest terms yields

$$\begin{cases} cY_t^{e2} & \leq c_1 Z_t^{e2} + c_2 Z_t^e + c_3, \\ dX_t^e Z_t^e & \leq d_1 Z_t^{e2} + d_2 Z_t^e, \\ eX_t^e Y_t^e & \leq e_1 Z_t^{e2} + e_2 Z_t^e + e_3, \end{cases}$$

where

$$\begin{aligned} c_1 &= \beta^2 \left(\frac{2\Delta_\gamma(1+\eta) + \sqrt{2}\Delta_\gamma(1+\Delta_\gamma)(1-\eta)}{(1-\eta)(1-\eta') - 2(1+\eta)\beta} \right)^2 \cdot R, \\ d_1 &= \frac{2(1+\Delta_\gamma)\beta + \sqrt{2}(1-\eta')\Delta_\gamma}{(1-\eta)(1-\eta') - 2(1+\eta)\beta} \cdot R, \\ e_1 &= \beta \frac{2\Delta_\gamma(1+\eta) + \sqrt{2}\Delta_\gamma(1+\Delta_\gamma)(1-\eta)}{(1-\eta)(1-\eta') - 2(1+\eta)\beta} \cdot \frac{2(1+\Delta_\gamma)\beta + \sqrt{2}(1-\eta')\Delta_\gamma}{(1-\eta)(1-\eta') - 2(1+\eta)\beta} \cdot R, \\ & c_2 > 0, c_3 > 0, d_2 > 0, e_2 > 0, e_3 > 0. \end{aligned}$$

Combining the right-hand side terms of (6.28) leads to

$$E\left[v_{t+1}\right] \leq E\left[v_0\right] - a_0 Z_t^{e2} + b_0 Z_t^e + c_0, \tag{6.29}$$

where a_0, b_0, c_0 are constants depending on η, β and Δ_γ. Since $v_t \geq 0, \forall t > 0$, (6.29) can be rewritten as

$$-a_0 Z_t^{e2} + b_0 Z_t^e + c_0 + E\left[v_0\right] \geq 0. \tag{6.30}$$

Additionally, it is not difficult to show that $a_0 > 0$ when the stepsize is sufficiently small, i.e., $\beta \ll 1$.

Since $b_0 > 0, c_0 + E\left[v_0\right] > 0$, it follows from (6.30) that

$$\lim_{t\to\infty} Z_t^e \leq Z_\infty^e < \infty. \tag{6.31}$$

Thus, by Markov's inequality [94], we have for any $\varepsilon > 0$

$$\sum_{k=0}^{\infty} P(\|\bar{y}_k\| > \varepsilon) \leq \frac{\sum_{k=0}^{\infty} \|\bar{y}_k\|_E^2}{\varepsilon^2} = \frac{{Z_\infty^e}^2}{\varepsilon^2} < \infty.$$

Then, by Borel-Cantelli Lemma and Proposition 1.2 [94, p.206], we have $\lim_{k\to\infty} \|\bar{y}_k\| = 0$ $a.s.$. Also, from (6.21b) of Lemma 6.6 and (6.31), we know that

$$\lim_{k\to\infty} Y_k^e \leq Y_\infty^e \leq \frac{(\rho_2 p_1 + p_2) Z_\infty^e + (q_2 + \rho_2 q_1)}{(1 - \rho_1 \rho_2)} < \infty.$$

which, as shown above, yields $\lim_{k\to\infty} \|\tilde{y}_k\| = 0$ $a.s.$.

Likewise, using (6.21a) of Lemma 6.6 and (6.31) we have $\lim_{k\to\infty} \|\tilde{x}_k\| = 0$ $a.s.$.

In addition, using $(a+b)^2 \leq 2a^2 + 2b^2, \forall a, b \in \mathcal{R}$, (6.26) can be rewritten as

$$E\left[v_{k+1}|\mathcal{F}_k\right] \leq v_k - \frac{\bar{\mu}_\gamma}{\sqrt{m}} \|\bar{y}_k\|^2 + \epsilon_k \tag{6.32}$$

where

$$\begin{aligned} \epsilon_k =& \frac{L(\bar{\sigma}_\gamma + \tilde{\sigma}_\gamma)}{2\sqrt{m}} \|\tilde{x}_k\|^2 + \left(\frac{\tilde{\mu}_\gamma + L\tilde{\sigma}_\gamma}{2\sqrt{m}} + \frac{L\tilde{\sigma}_\gamma^2}{m}\right) \|\tilde{y}_k\|^2 \\ &+ \left(\frac{\tilde{\mu}_\gamma + L\bar{\sigma}_\gamma}{2\sqrt{m}} + \frac{L\bar{\sigma}_\gamma^2}{m}\right) \|\bar{y}_k\|^2 . \end{aligned} \tag{6.33}$$

Moreover, we have shown in the above that $\sum_{k=0}^{\infty} E\left[\|\tilde{x}_k\|^2\right]$, $\sum_{k=0}^{\infty} E\left[\|\tilde{y}_k\|^2\right]$ and $\sum_{k=0}^{\infty} E\left[\|\bar{y}_k\|^2\right]$ are all bounded, thus using Lemma 6.2 and 6.3 we obtain $\sum_{k=0}^{\infty} \epsilon_k < \infty$ $a.s.$. Since $\epsilon_k > 0$, Lemma 6.5 applies and the sequence $\{v_k\}_{k\geq 0}$ converges to some random variable almost surely and thus bounded almost surely and so is the sequence $\{f(\bar{x}_k)\}_{k\geq 0}$. Since f is coercive and thus has compact level set (cf. Assumption 5.6 and Remark 5.11), this implies that $\bar{x}_k$ is bounded with probability one. Thus, there must exists a subsequence $\bar{x}_{k_j}$ converging to some limit point $\bar{x}_\infty$ for which $\Pi_{\|} g(\bar{x}_\infty) = 0$, i.e., $\sum_{i=1}^{m} g_i(\theta_\infty) = 0$, for certain $\theta_\infty \in \mathcal{R}$ (note that $\lim_{k\to\infty} \bar{y}_k = \Pi_{\|} g(x_k) = 0$ and $\lim_{k\to\infty} \tilde{x}_k = 0$). This in turn, by convexity of f, implies that $\bar{x}_\infty$ is the optimal solution to the EDOP problem. Also, from (5.43) and knowing that $\overline{\gamma_k \odot y_k}$ converges to 0, we observe that $\lim_{k\to\infty} \bar{x}_{k+1} - \bar{x}_k = 0$, which implies that the subsequence $\bar{x}_{k_j - 1}$ also converges to the same limit point $\bar{x}_\infty$. In all, we conclude that

$$\lim_{k\to\infty} f(\bar{x}_k) = f(\bar{x}_\infty) = f^\star \ a.s.. \tag{6.34}$$

Further, since $\lim_{k\to\infty} \|\tilde{x}_k\| = 0$ $a.s.$ as shown above, by [94, Th. 10.1, p.244], using the continuity of f and norm and the fact of the convergence of the sequence

$\{f(\bar{x}_k)\}_{k\geq 0}$ we obtain

$$\begin{aligned}\lim_{k\to\infty}\left(f(x_k)-f(\bar{x}_k)\right) &= \lim_{k\to\infty}\left(f(\bar{x}_k+\tilde{x}_k)-f(\bar{x}_k)\right)\\ &= f(\lim_{k\to\infty}\bar{x}_k+0)-f(\lim_{k\to\infty}\bar{x}_k)=0 \ a.s..\end{aligned} \tag{6.35}$$

By [94, Th. 11.1], using (6.34) and (6.35) completes the proof.

Corollary 6.3. *Consider the algorithm (6.17) with $y_0 = g(x_0)$. Suppose all the assumptions of Theorem 6.2 hold and the computation processes of agents are synchronous (i.e., $\Delta_\gamma = 0$). If $\gamma_{\max} < \frac{\eta^2-\eta+4-\sqrt{\eta^4-6\eta^3+13\eta^2-4\eta+12}}{2(1+\eta)L}$, then we have $\lim_{k\to\infty}\|\tilde{x}_k\| = 0$ a.s. and $\lim_{k\to\infty} f(x_k) = f^\star$.*

Proof. Since $\Delta_\gamma = 0$, the coefficient a_0 of (6.29) can be calculated as follows:

$$a_0 = \left[\left(\frac{1}{\beta}-1\right)-\frac{2\beta}{(1-\eta)(1-\eta')-2(1+\eta)\beta}\right]\cdot R.$$

Since $\beta < \frac{(1-\eta)^2}{3+\eta+\Delta_\gamma(1-\eta)}$. Then, simple calculation shows that $a_0 > 0$ is equivalent to

$$(1+\eta)\beta^2-\left(\eta^2-\eta+4\right)\beta+(1-\eta)^2>0.$$

Then, knowing that $\beta = \gamma_{\max}L$ and taking the smaller root completes the proof.

Figure 6.2 plots the estimated upper bound of $\beta = \gamma_{\max}L$ in terms of the spectral radius η with $\Delta_\gamma = 0$ to ensure certain conditions for convergence.

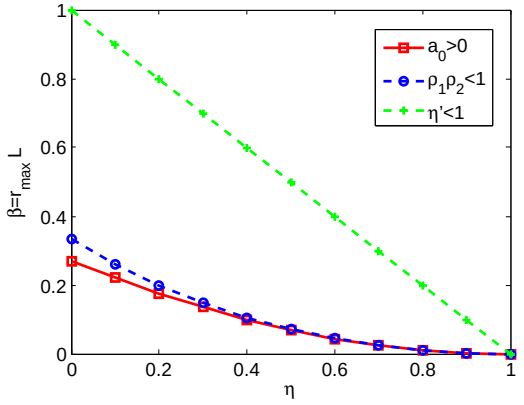


FIGURE 6.2: Plots of the estimated upper bound of β versus η with $\Delta_\gamma = 0$ to ensure certain conditions: $\eta' < 1$, $\rho_1\rho_2 < 1$ and $\mu > 0$.

6.4.4 Convergence rate analysis for strongly convex functions

We make a stronger assumption on the cost function in order to derive the convergence rate for the proposed algorithm. Note that the following assumption is not necessary when the algorithm is synchronously running[3].

Assumption 6.4. Each objective function f_i is l_i-strongly convex with $l_i > 0$.

Remark 6.3. The above assumption implies the uniqueness of the optimal solution. In addition, it is easy to show that the overall function f is also l-strongly convex with $l = \min\{l_i\}$.

Theorem 6.4. *Consider the distributed algorithm 4 with $y_0 = g(x_0)$. Let $\hat{x}_k = \frac{1}{t}\sum_{k=0}^{t-1} x_k$ be the running average. Suppose Assumptions 5.1, 5.7, 6.3 and 6.4 hold. Then, there exists a positive number $\gamma^\star := \varphi(\eta, \Delta_\gamma)/L$ such that, if $\gamma_{\max} < \gamma^\star$, we have $E\left[\left\|\hat{x}_k - \hat{\bar{x}}_k\right\|\right] \leq O(\frac{1}{\sqrt{k}})$ and $E\left[|f(\hat{x}_k) - f^\star|\right] \leq O(\frac{1}{\sqrt{k}})$.*

Proof. Consider the sequence (5.43). Let $x^\star \in \mathcal{R}$ be an optimum of the EDOP problem. Then, we have

$$\begin{aligned} E\left[\|\bar{x}_{k+1} - x^\star\|^2 |\mathcal{F}_k\right] &= E\left[\|\bar{x}_k - \overline{\gamma_k \odot y_k} - x^\star\|^2 |\mathcal{F}_k\right] \\ &\leq \|\bar{x}_k - x^\star\|^2 - 2E\left[\langle \overline{\gamma_k \odot y_k}, \bar{x}_k - x^\star\rangle |\mathcal{F}_k\right] + E\left[\|\overline{\gamma_k \odot y_k}\|^2 |\mathcal{F}_k\right]. \end{aligned} \tag{6.36}$$

Let us first consider the second term.

$$\begin{aligned} & E\left[\langle \overline{\gamma_k \odot y_k}, \bar{x}_k - x^\star\rangle |\mathcal{F}_k\right] \\ &\overset{(a)}{=} E\left[\langle \bar{\gamma}_k \odot \bar{y}_k, \bar{x}_k - x^\star\rangle |\mathcal{F}_k\right] + E\left[\left\langle \overline{\tilde{\gamma}_k \odot \tilde{y}_k}, \bar{x}_k - x^\star\right\rangle |\mathcal{F}_k\right] \\ &\overset{(b)}{\geq} \frac{\bar{\mu}_\gamma}{\sqrt{m}} \langle \bar{y}_k, \bar{x}_k - x^\star\rangle - \frac{\tilde{\mu}_\gamma}{\sqrt{m}} \|\tilde{y}_k\| \|\bar{x}_k - x^\star\| \\ &\overset{(c)}{\geq} \frac{\bar{\mu}_\gamma}{\sqrt{m}} \left\langle \Pi_{\|} g(\bar{x}_k), \bar{x}_k - x^\star\right\rangle - \frac{\tilde{\mu}_\gamma}{\sqrt{m}} \|\tilde{y}_k\| \|\bar{x}_k - x^\star\| \\ &\quad + \frac{\bar{\mu}_\gamma}{\sqrt{m}} \left\langle \Pi_{\|} g(x_k) - \Pi_{\|} g(\bar{x}_k), \bar{x}_k - x^\star\right\rangle, \end{aligned} \tag{6.37}$$

where (a) is due to Prob. 5.1-(iii), (b) is obtained from Prob. 2-(iv) and (c) is derived from the conversation property $\bar{y}_k = \bar{g}_k = \Pi_{\|} g(x_k), \forall k > 0$ (cf. Lemma 5.6).

[3] In this case, the convergence proof follows from the similar lines as in Theorem 5.4.

Since the cost function f is l-strongly convex and has Lipschitz gradient, we have

$$\begin{aligned}\|\bar{x}_k - x^\star\| &\leq \frac{1}{l}\left\|\Pi_{\|} g(\bar{x}_k)\right\| = \frac{1}{l}\left\|\Pi_{\|} g(\bar{x}_k) - \Pi_{\|} g(x_k) + \Pi_{\|} g(x_k)\right\| \\ &\leq \frac{1}{l}\|\bar{y}_k\| + \frac{L}{l}\|\tilde{x}_k\|,\end{aligned} \tag{6.38}$$

and using $\bar{a}^T\bar{b} = a^T\bar{b}$ (cf. Proposition 5.1-(i)) to eliminate $\Pi_{\|}$ in the last inequality of (6.37) we further have

$$\begin{aligned}&E\left[\langle\overline{\gamma_k \odot y_k}, \bar{x}_k - x^\star\rangle \,|\mathcal{F}_k\right] \\ &\geq \frac{\bar{\mu}_\gamma}{\sqrt{m}}\langle g(\bar{x}_k), \bar{x}_k - x^\star\rangle - \frac{\tilde{\mu}_\gamma}{\sqrt{m}}\|\tilde{y}_k\|\,\|\bar{x}_k - x^\star\| + \frac{\bar{\mu}_\gamma}{\sqrt{m}}\langle g(x_k) - g(\bar{x}_k), \bar{x}_k - x^\star\rangle \\ &\geq \frac{\bar{\mu}_\gamma}{\sqrt{m}}(f(\bar{x}_k) - f^\star) - \left(\frac{\tilde{\mu}_\gamma}{\sqrt{m}}\|\tilde{y}_k\| + \frac{\bar{\mu}_\gamma}{\sqrt{m}}L\|\tilde{x}_k\|\right)\left(\frac{1}{l}\|\bar{y}_k\| + \frac{L}{l}\|\tilde{x}_k\|\right).\end{aligned} \tag{6.39}$$

Combining (6.36) and (6.39) and using (6.24) and the relation $2ab \leq a^2 + b^2$ yield

$$E\left[\|\bar{x}_{k+1} - x^\star\|^2 \,|\mathcal{F}_k\right] \leq \|\bar{x}_k - x^\star\|^2 - \frac{2\bar{\mu}_\gamma}{\sqrt{m}}(f(\bar{x}_k) - f^\star) + \epsilon_k, \tag{6.40}$$

where

$$\begin{aligned}\epsilon_k =& \frac{(\bar{\mu}_\gamma + \tilde{\mu}_\gamma)L + 2\bar{\mu}_\gamma L^2}{l\sqrt{m}}\|\tilde{x}_k\|^2 \\ &+ \left(\frac{(1+L)\tilde{\mu}_\gamma}{l\sqrt{m}} + \frac{2\tilde{\sigma}_\gamma^2}{m}\right)\|\tilde{y}_k\|^2 + \left(\frac{\tilde{\mu}_\gamma + \bar{\mu}_\gamma L}{l\sqrt{m}} + \frac{2\bar{\sigma}_\gamma^2}{m}\right)\|\bar{y}_k\|^2\end{aligned} \tag{6.41}$$

Using convexity of f, we have

$$f(x_k) - f(\bar{x}_k) \leq \langle g(\bar{x}_k), x_k - \bar{x}_k\rangle.$$

Since f is strongly convex thus coercive, it follows from Theorem 6.2 that there exists a $\gamma^\star$ such that if $\gamma_{\max} \leq \gamma^\star$, $\bar{x}_k$ will be bounded almost surely.
Thus, knowing that the gradient g is bounded for any compact domain $\mathcal{D}$, i.e., $\|g(x)\| \leq C, x \in \mathcal{D}$, we further have

$$|f(x_k) - f(\bar{x}_k)| \leq \|g(\bar{x}_k)\|\,\|x_k - \bar{x}_k\| \leq C\|\tilde{x}_k\|. \tag{6.42}$$

Then, using the above equation and the relation $|a|+|b| \geq |a+b|, \forall a, b \in \mathcal{R}$, (6.40)

can be rewritten as

$$E\left[\|\bar{x}_{k+1}-x^\star\|^2|\mathcal{F}_k\right] \leq \|\bar{x}_k-x^\star\|^2 - \frac{2\bar{\mu}_\gamma}{\sqrt{m}}|f(x_k)-f^\star| + \frac{2\bar{\mu}_\gamma C}{\sqrt{m}}\|\tilde{x}_k\| + \epsilon_k. \quad (6.43)$$

Taking the total expectation of the above inequality and rearranging terms we have

$$\begin{aligned}\frac{2\bar{\mu}_\gamma}{\sqrt{m}}&E\left[|f(x_k)-f^\star|\right] \\ &\leq E\left[\|\bar{x}_k-x^\star\|^2\right] - E\left[\|\bar{x}_{k+1}-x^\star\|^2\right] + \frac{2\bar{\mu}_\gamma C}{\sqrt{m}}E\left[\|\tilde{x}_k\|\right] + E\left[\epsilon_k\right].\end{aligned} \quad (6.44)$$

Summing the above inequality over k from 0 to $t-1$ leads to

$$\begin{aligned}\frac{2\bar{\mu}_\gamma}{\sqrt{m}}\sum_{k=0}^{t-1}&E\left[|f(x_k)-f^\star|\right] \leq E\left[\|\bar{x}_0-x^\star\|^2\right] \\ &- E\left[\|\bar{x}_t-x^\star\|^2\right] + \frac{2\bar{\mu}_\gamma C}{\sqrt{m}}\sum_{k=0}^{t-1}E\left[\|\tilde{x}_k\|\right] + \sum_{k=0}^{t-1}E\left[\epsilon_k\right].\end{aligned} \quad (6.45)$$

From Theorem 6.2, we know that $\sum_{k=0}^{\infty}E\left[\epsilon_k\right] < B_0$ *a.s.* with sufficiently small $\gamma_{\max}$, where B_0 is some positive number. Also, using the Cauchy-Schwarz inequality

$$a_1+a_2+...+a_m \leq \sqrt{m}\sqrt{a_1^2+a_2^2+...+a_m^2},$$

where $a_1, a_2, ..., a_m$ are positive numbers, we have

$$\sum_{k=0}^{t-1}E\left[\|\tilde{x}_k\|\right] \leq \sqrt{t}\sqrt{\sum_{k=0}^{t-1}E\left[\|\tilde{x}_k\|\right]^2} \leq \sqrt{t}\sqrt{\sum_{k=0}^{t-1}\|\tilde{x}_k\|_E^2}. \quad (6.46)$$

Thus, dividing both sides of (6.45) by $\frac{2\bar{\mu}_\gamma}{\sqrt{m}}t$ we have

$$\frac{1}{t}\sum_{k=0}^{t-1}E\left[|f(x_k)-f^\star|\right] \leq \frac{\sqrt{m}\left(A_0+B_0\right)}{2\bar{\mu}_\gamma t} + \frac{CX_\infty}{\sqrt{t}}. \quad (6.47)$$

where $X_\infty = \lim_{t\to\infty}\sqrt{\sum_{k=0}^{t-1}\|\tilde{x}_k\|_E^2}$ and $A_0 = E\left[\|\bar{x}_0-x^\star\|^2\right]$.
Let $\hat{\bar{x}}_t = 1/t\sum_{k=0}^{t-1}\bar{x}_k$ be the running average. Applying Jensen inequality to the above relation (6.47) we have

$$E\left[|f(\hat{\bar{x}}_t)-f^\star|\right] \leq \frac{1}{t}\sum_{k=0}^{t-1}E\left[|f(x_k)-f^\star|\right] \leq \frac{\sqrt{m}\left(A_0+B_0\right)}{2\bar{\mu}_\gamma t} + \frac{CX_\infty}{\sqrt{t}}. \quad (6.48)$$

In addition, diving both sides of (6.46) by t we have

$$\frac{1}{t}\sum_{k=0}^{t-1} E\left[\|\tilde{x}_k\|\right] \leq \frac{1}{\sqrt{t}}\sqrt{\sum_{k=0}^{t-1}\|\tilde{x}_k\|_E^2} \leq \frac{1}{\sqrt{t}}X_\infty. \tag{6.49}$$

Let $\hat{\tilde{x}}_k = 1/t\sum_{k=0}^{t-1}\tilde{x}_k$. Applying the Jensen inequality we have $E\left[\left\|\hat{x}_k - \hat{\tilde{x}}_k\right\|\right] \leq \frac{1}{t}\sum_{k=0}^{t-1} E\left[\|\tilde{x}_k\|\right] \leq \frac{1}{\sqrt{t}}X_\infty$. The rest of the proof follows from the fact that X_∞ is bounded as previously shown in Theorem 6.2 with sufficiently small $\gamma_{\max}$.

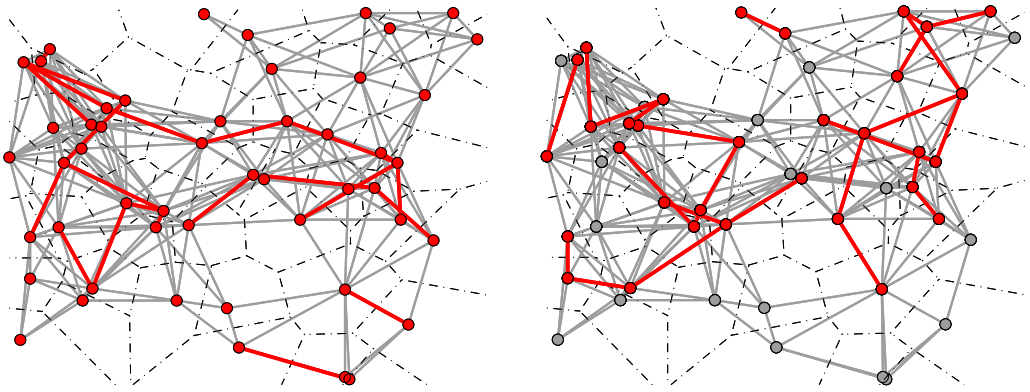
Corollary 6.5. *Consider the algorithm (6.17) with $y_0 = g(x_0)$. Suppose all the assumptions of Theorem 6.4 hold and the computation processes of agents are synchronous (i.e., $\Delta_\gamma = 0$). Then, if $\gamma_{\max} < \frac{\eta^2-\eta+4-\sqrt{\eta^4-6\eta^3+13\eta^2-4\eta+12}}{2(1+\eta)L}$, we have $E\left[\left\|\hat{x}_k - \hat{\tilde{x}}_k\right\|\right] \leq O(\frac{1}{\sqrt{k}})$ and $E\left[|f(\hat{x}_k) - f^\star|\right] \leq O(\frac{1}{\sqrt{k}})$.*

Proof. The proof is similar to that of Corollary 6.3.

6.5 Application to Sensor Fusion Problems

In this section, we report some simulations to show the effectiveness of the proposed algorithms over stochastic and asynchronous scenarios. We consider the same distributed estimation problem as before (cf. Section 5.5) over a stochastic network (cf. Figure 6.3a) where each communication link is subject to random failure following certain Bernoulli Process. That is, in each iteration, each communication link will be activated with probability of p and deactivated with $1-p$. Thus, when $p = 1$, the random network will reduce to a fixed network.

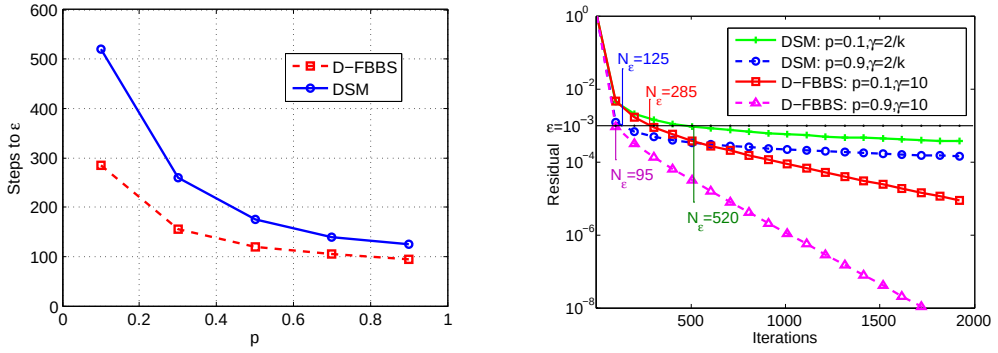
Parameter Setting: We use the same parameter setting as before (cf. Section 5.5) except for that we use $\lambda = 0.1$ for the D-FBBS algorithm for the requirement of strong convexity of the cost function and the weight matrix now becomes $W_k = I - \alpha_k L$ with $\alpha_k = \frac{1}{2+d_{\max,k}}$ for D-FBBS and $\alpha_k = 2d_{\max,k}$ for AsynDGM, where $d_{\max,k}$ is the maximum degree of the communication graph at time k [89, 92]. The simulation is carried out over a stochastic network and all the results are averaged over 20 Monte-Carlo runs. We conduct two separate simulations for D-FBBS and AsynDGM over a stochastic scenario (cf. Figure 6.3a) and an asynchronous scenario (cf. Figure 6.3b), respectively. The result of D-FBBS is then compared with DSM [27] in terms of the relative FPR and the result of AsynDGM with the



(A) stochastic scenario (B) asynchronous scenario

FIGURE 6.3: A snapshot of a random sensor network of 50 nodes. The red lines denote the communication links being activated while the gray lines stand for no communication being carried out at this moment. Correspondingly, the red dots denote the active nodes while the gray dots stand for the deactive nodes.

best known asynchronous distributed algorithm[4]–RandBroadcast [62]–in terms of the relative objective error (OBE), i.e., $\frac{\|f(x_k)-f^\star\|}{\|f(x_0)-f^\star\|}$, respectively.



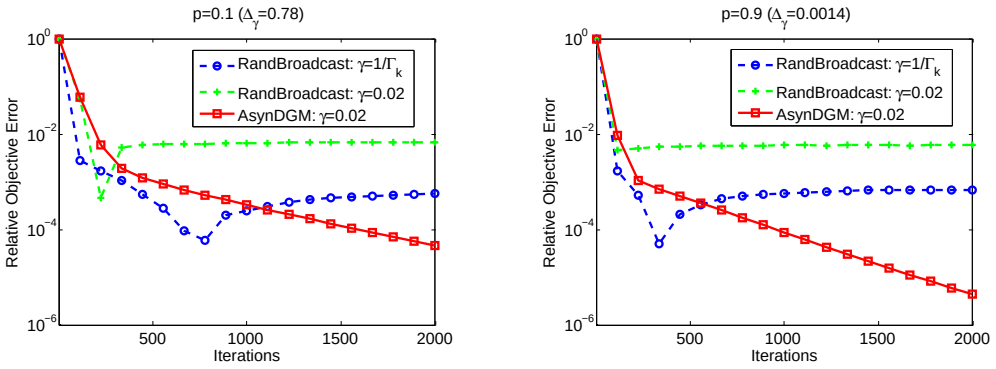
(A) (B)

FIGURE 6.4: Performance Comparison between D-FBBS and DSM. (a) Plot of the number of iterations required to reach a fixed accuracy $\epsilon = 0.001$ for both DSM and D-FBBS. The stepsize $\gamma = 2/k$ for DSM is optimized by hand while the stepsize $\gamma = 10$ for D-FBBS is calculated based on Theorem 6.1. The results are averaged over 20 Monte-Carlo runs. (b) Plot of the relative FPR versus the number of iterations for both DSM and D-FBBS under two different probabilities of link failure, i.e., $p = 0.1$ (low) and $p = 0.9$ (high).

Discussions: Figure 6.4a and Figure 6.4b illustrate the simulation results for stochastic networks. Figure 6.4a shows that the proposed D-FBBS algorithm always needs less iterations to reach the specified accuracy of $\epsilon = 0.001$ as compared

[4]In our simulation, we use gossip-like protocols (cf. Remark 6.2) rather than the random broadcast scheme for both algorithms for fair comparison.

with DSM. The advantage is more significant when the communication link has low probability of being activated at each iteration, i.e., communication process being more "asynchronous". In particular, we can observe from Figure 6.4b that the proposed D-FBBS algorithm require $N_\epsilon = 95$ iterations to reach the specified accuracy while DSM needs $N_\epsilon = 125$ iterations when the communication link is activated with a high probability of $p = 0.9$. In addition, when p becomes small (e.g., $p = 0.1$), the difference of iterations required is enlarged, i.e., $N_\epsilon = 285$ for D-FBBS and $N_\epsilon = 520$ for DSM. Moreover, similar to the case of fixed network, both algorithms have similar performance in the beginning but the proposed D-FBBS algorithm still progresses at a linear convergence rate afterwards.



(A) High probability of link failure

(B) Low probability of link failure

FIGURE 6.5: Plot of the relative objective error with respect to the number of iterations for both AsynDGM and RandBroadcast algorithms under: (a) low and (b) high probability of link failure.

Figure 6.5 plots the relative objective error of both AsynDGM and RandBroadcast algorithms, which are asynchronously implemented, with respect to the number of iterations over a stochastic network under high and low probability of link failure respectively. It follows from the figure that the two algorithms have similar convergence performance at the initial stage. However, RandBroadcast using constant stepsize gets stuck after several iterations which can be alleviated to some extent by employing decaying stepsize $\frac{1}{\Gamma_{i,k}}$[5], while the proposed AsynDGM algorithm still progresses almost linearly to the optimum. Different from D-FBBS, the advantage of the AsynDGM algorithm over the RandBroadcast algorithm will be more significant when the communication network has low probability of link failure (i.e., the network being more "synchronous").

[5]Here, $\Gamma_{i,k}$ denotes the number of updates agent i has carried out before time k [62].

6.6 Summary

In this chapter, we have investigated the convergence performance of the two previously proposed algorithms, namely D-FBBS and AsynDGM, over stochastic networks even under asynchronous implementation. We have shown that both algorithms are able to seek the exact optimum even with constant stepsizes so long as it is chosen properly within certain theoretical bound. For the D-FBBS algorithm under stochastic networks, with an extra assumption of strong convexity on the cost functions, we have shown that the algorithm is guaranteed to converge to the optimum almost surely and an ergodic convergence rate of $O(1/\sqrt{k})$ can be established in terms of FPR. Further, for the AsynDGM algorithm under asynchronous implementation, we have shown its convergence to the optimum almost surely for coercive and convex functions with Lipschitz gradients and established an ergodic convergence rate of $O(1/\sqrt{k})$ in terms of OBE for functions that are also strongly convex. Note that the above obtained convergence rates are the known best rates under the same setting as this research work. We have also reported an example to illustrate the effectiveness of both proposed algorithms.

Chapter 7

Conclusion and Future Work

7.1 Discussions and Summary

In this thesis, we have proposed several schemes and algorithms for the distributed optimization problem involved in large-scale networked systems that arises from many application domains, such as sensor fusion, resource allocation and distributed learning. Distributed methods for optimization problems in these systems are very important either from the perspective of robustness or computational complexity. Many existing works have been devoted to this area but most of them are not able to account for practical issues such as heterogeneity, varying topology and asynchronous implementation that are common in real applications. In this research work, we aim to develop new schemes and algorithms that are more capable in dealing with the above issues. Specifically, we have made several contributions to the community of distributed optimization in the following aspects:

- A general philosophy has been proposed based on the fundamental analysis of consensus mechanism for coordination, and it allows us to easily come up with certain distributed schemes and algorithms for specific problems.
- A novel distributed derivative-free approach (cf. D-SPA) has been proposed to solve the dynamic optimal consensus problem in large-scale networked control systems based on simultaneous perturbation and consensus strategies. The proposed scheme is shown to be able to achieve Pareto-optimum rather than Nash equilibrium of the whole system under control. A semi-global stability

result has also been obtained for more possible applications. Indeed, it is expected that the proposed scheme is applicable to many other large-scale dynamic systems where the analytical form of the performance is hard to obtain and distributed implementation is a necessity.

- Two basic distributed algorithms have been proposed to solve the optimal consensus problem in large-scale sensor networks under fixed topology and synchronous implementation. Both are able to seek the exact optimum even with constant stepsize without the assumption of boundedness of (sub)-gradient. A non-ergodic convergence rate of $o(\frac{1}{k})$ in terms of FPR has been established for the D-FBBS algorithm for general proper convex functions while an ergodic convergence rate of $O(\frac{1}{\sqrt{k}})$ in terms of OBE is obtained for the AugDGM algorithm under homogeneous computation (i.e., using same stepsizes) for coercive and convex functions having Lipschitz gradients. What is more interesting is that the techniques used in developing these algorithms such as the Bregman method and operator splitting, turns out to be very useful and, in fact, provide a framework for us to easily come up with certain distributed algorithms for specific problems with certain structure.

- The proposed algorithms have been shown to be very capable in dealing with stochastic networks even under asynchronous settings. In particular, both algorithms are shown to be able to seek the exact optimum almost surely even with constant stepsizes. Besides, an ergodic convergence rate of $O(\frac{1}{k})$ in terms of FPR has been established for strongly convex functions for the D-FBBS algorithm over stochastic networks and an ergodic convergence rate of $O(\frac{1}{\sqrt{k}})$ in terms of OBE is obtained for strongly convex functions with Lipschitz gradients for the AsynDGM algorithm over stochastic networks even under asynchronous implementation. To the best of our knowledge, these are the known best rates under the same setting as this work. In this regard, our result is a significant improvement of the existing distributed algorithms, making them amenable to stochastic and asynchronous scenarios.

7.2 Extensions and Future Work

The results developed in this research work are still quite limited in the sense that the convergence performance of the proposed distributed schemes and algorithms

are always inferior to their centralized counterparts and the difference is even more significant when it comes to time-varying topology and asynchronous implementation. Thus, it would be of great interest and importance to find the *fundamental (theoretical) limit* of the convergence performance for distributed methods over general (unbalanced directed) graphs where algorithms may run in a totally asynchronous way, which will also serve as a general guideline in designing these methods as well as the practical experiments used for verification. Some extensions can also be made to account for *constraints*, *quantization effects* as well as *asymmetric issues* induced by general graphs and asynchrony of algorithms.

7.2.1 Large-scale distributed learning

Distributed optimization has recently become a hot topic in the machine learning community due to its ability to deal with large-scale datasets. Most existing algorithms, however, is not distributed and require somewhat supervisory control over the whole network. Thus, it would be of interest if we can design some distributed algorithm that can run over a large-scale data processing center where several processors are geographically scattered and each is responsible for a partial dataset. In machine learning, we always encounter the optimization problem of composite cost functions with a regularization term encoding some prior knowledge or the feasible set (e.g., indicator function). An optimal consensus problem for this kind of cost function can be thus depicted as follows:

$$\min_{\theta\in\mathcal{R}^d} f(\theta) = \sum_{i=1}^{m} f_i(\theta) + g_i(\theta),$$

where g_i is the regularization term associated with agent i. The above formulation is, in fact, corresponding to splitting across datasets in machine learning.

7.2.2 Constraints and quantization effects

In most real applications, systems are usually subject to some constraints, be it local (hard) $\mathcal{X}_i \subset \mathcal{R}^d$[1] or global (soft) $g(\theta) \leq 0$ (cf. Equation (7.1)). Thus, it is desirable to design some algorithm that can account for these constraints. However,

[1]Note that the intersection of all local sets $\mathcal{X}_i$ can be empty.

introducing global constraints may render the problem to be non-decomposable, imposing much challenges on the design of distributed algorithms.

$$\min_{\{\theta \in \cap_i \mathcal{X}_i\}} f(\theta) = \sum_{i=1}^{m} f_i(\theta), \quad \textbf{s.t. } g_i(\theta) \leq 0, \ \forall i \in \mathcal{V}. \tag{7.1}$$

On the other hand, since distributed optimization is carried out over a network which is unreliable and has limited bandwidth, quantization is necessary for real implementation. Thus, it would be of interest if we can design a *proper encoder/decoder* or *event-trigger-based mechanism* to alleviate the communication burden and design new algorithms that can account for packet loss and delay or investigate the performance limit of algorithms with respect to these parameters.

7.2.3 Towards general graphs and total asynchrony

It is well known that doubly-stochastic weight matrix is difficult to design and maintain in real applications where the graph may not be balanced and the algorithm may be executed asynchronously. There are many existing algorithms, such as those based on the push-sum protocol, that can operate on column-stochastic weight matrix which are relatively simple to design. It would be of interest if we could come up with a new way for compensating the errors either due to the asymmetric "data flow" over unbalanced graphs or the asynchronous implementation of algorithms over heterogeneous nodes or both simultaneously based on fixed point and operator splitting theory. Note that the effect induced by unbalanced graphs somewhat resembles that of asynchronous implementations.

In this thesis, we have observed that consensus-based algorithms generally require weak assumptions on the communication graph but the obtained convergence rates are always inferior to its centralized counterpart. In contrast, decomposition-based algorithms are designed based on well-established optimization theory and can usually obtain very good convergence rates that are comparable with the centralized counterparts but they require strict assumptions on the weight matrix, making it not so practical in real applications. Thus, it would be of great interest to investigate the possibility of filling the gap between consensus-based and decomposition-based approaches especially in dealing with stochastic and asynchronous scenarios by seeking new mathematical tools for alternative ways of proofs.

Appendix A

Proofs for Part I

A.1 Proof of Lemma 4.2

Consider the system (4.7). Let Ω be any given arbitrary compact domain. Since ψ is locally Lipschitz in θ and $\left\|\frac{\mu}{a}\right\|$ is bounded according to Definition 4.5. Thus, for any $\theta \in \Omega$ the average

$$\psi^{av}(\theta) = \frac{1}{T}\int_0^T [\psi(\theta + \mu(\tau)) + C] \otimes \frac{\mu(\tau)}{a} d\tau$$

exists (cf. Definition 4.2). Let us consider the change of variables $\theta = \theta^{av} - \varepsilon u(t, \theta^{av})$, where $\varepsilon = a\delta$ and

$$u(t, \theta^{av}) = \int_0^t [\psi(\theta^{av} + \mu(\tau)) + C] \otimes \frac{\mu(\tau)}{a} - \psi^{av}(\theta^{av}) d\tau.$$

Then, differentiating both sides gives

$$\dot{\theta} = \dot{\theta}^{av} - \varepsilon \frac{\partial u}{\partial t} - \varepsilon \frac{\partial u}{\partial \theta^{av}} \dot{\theta}^{av}.$$

Substituting (4.7) into the above equation and simple calculation yields

$$\begin{aligned}\left[I-\varepsilon\frac{\partial u}{\partial\theta^{av}}\right]\dot{\theta}^{av} &= \varepsilon\frac{\partial u}{\partial t}-\delta[\psi(\theta+\mu)+C]\otimes\mu\\ &= \varepsilon\left[\psi(\theta^{av}+\mu)\otimes\frac{\mu}{a}-\psi^{av}(\theta^{av})\right]-\delta\psi(\theta+\mu)\otimes\mu\\ &= -\varepsilon\psi^{av}(\theta^{av})+\delta\left[\psi(\theta^{av}+\mu)-\psi(\theta^{av}-\varepsilon u+\mu)\right]\otimes\mu\\ &= -\varepsilon\psi^{av}(\theta^{av})+\varepsilon^2 u\left[\frac{\partial\psi}{\partial\theta}(\xi+\mu)\otimes\frac{\mu}{a}\right],\end{aligned} \tag{A.1}$$

where $\xi\in\{\theta|\,\|\theta-\theta^{av}\|\leq\varepsilon u\}$ and we have used the mean value theorem to obtain the last equality. In addition, for any $\theta^{av}\in\Omega$, $\frac{\partial u}{\partial\theta^{av}}$ is bounded such that the inverse matrix $\left[I-\varepsilon\frac{\partial u}{\partial\theta^{av}}\right]^{-1}$ exists and can be approximated as $I+O(\varepsilon)$ for sufficiently small ε. Thus, knowing that u, $\frac{\partial\psi}{\partial\theta}$ and $\left\|\frac{\mu}{a}\right\|$ are bounded for $\forall\theta\in\Omega$ (cf. Definition 4.5 and 4.2), the above dynamic equation can be rewritten as follows:

$$\dot{\theta}^{av}=-\varepsilon\psi^{av}(\theta^{av})+O(\varepsilon^2). \tag{A.2}$$

Moreover, since ψ is a $\mathcal{C}^2$ function, by Taylor expansion, we have the following first-order approximation for the average function as follows:

$$\begin{aligned}\psi^{av}(\theta^{av}) &= \frac{1}{T}\int_t^{t+T}[\psi(\theta^{av}+\mu(\tau))+C]\otimes\frac{\mu(\tau)}{a}d\tau\\ &= \frac{1}{T}\int_t^{t+T}\left[(\psi(\theta^{av})+\nabla\psi(\theta^{av})^T\mu+C)\otimes\frac{\mu}{a}+r\right]d\tau\\ &= \frac{1}{T}\int_t^{t+T}[\psi(\theta^{av})+C]\otimes\frac{\mu(\tau)}{a}d\tau\\ &\quad+\left[\frac{1}{Ta}\int_t^{t+T}\mu(\tau)^T\otimes\mu(\tau)d\tau\right]\cdot\nabla\psi(\theta^{av})+r\\ &= a\nabla\psi(\theta^{av})+r,\end{aligned} \tag{A.3}$$

where $r=O(a^2)$ and ∇ denotes the gradient operator and we have employed certain conditions of Definition 4.5 and the following relation to obtain the last equality:

$$(a^T\cdot b)\otimes c=(b^T\otimes c)\cdot a,$$

where a, b and c are all column vectors with the same dimension. Then, combining Equation (A.3) with Equation (A.2) completes the proof.

A.2 Proof of Lemma 4.4

Consider a system $\dot{x} = \phi(t, x, \varepsilon)$ with parameter $\varepsilon \in \mathcal{R}$. Given any pair (δ, Δ) with $\delta < \Delta$, define a compact domain $\Omega : \{x | \delta < \|x\| < \Delta\}$. Since the nominal system $\dot{x} = \phi(t, x, 0)$ is UGAS and ϕ and its partial derivatives w.r.t. (x, ε) are locally Lipschitz in $\mathcal{R}^n \times \mathcal{R}$, uniformly in t, by Khalil [77, Th. 4.16] there exists a continuously differentiable function $W : \mathcal{R}_{\geq 0} \times \Omega \to \mathcal{R}_{\geq 0}$ that satisfies the following inequalities:

$$\begin{aligned} \gamma_1(\|x\|) \leq W(t, x) \leq \gamma_2(\|x\|) \\ \frac{\partial W}{\partial t} + \frac{\partial W}{\partial x}\phi(t, x, 0) \leq -\gamma_3(\|x\|) \end{aligned} \tag{A.4}$$

for all $x \in \Omega$ and some $\mathcal{K}_\infty$ functions γ_1 to γ_3.

Then, taking the derivative of W along the trajectory of the original system gives

$$\begin{aligned} \dot{W} &= \frac{\partial W}{\partial t} + \frac{\partial W}{\partial x}\phi(t, x, 0) + \frac{\partial W}{\partial x}^T [\phi(t, x, \varepsilon) - \phi(t, x, 0)] \\ &\leq -\gamma_3(\|x\|) + \left\| \frac{\partial W}{\partial x} \right\| \|\phi(t, x, \varepsilon) - \phi(t, x, 0)\| \\ &\leq -\gamma_3(\|x\|) + kL_1\varepsilon < 0. \ \forall \varepsilon < \min\{\frac{\gamma_3(\delta)}{kL_1}, \varepsilon_0\}, \end{aligned} \tag{A.5}$$

where k is the upper bound of $\left\| \frac{\partial W}{\partial x} \right\|$ on $x \in \Omega$, ε_0 is a small positive number and L_1 is the Lipschitz constant of ϕ in $[0, \varepsilon_0]$. It follows from the above that there exist ε^* such that the system is UGAS for each $\varepsilon \in (0, \varepsilon^*)$. Then, by Definition 4.1, we claim that the system is USPAS on ε.

Appendix B

Proofs for Part II

B.1 Proof of Lemma 5.1

Since $null(P) = span\{\mathbf{1}\}$, we have $rank(P) = m - 1$. In addition, $y \in span^{\perp}\{\mathbf{1}\}$ implies that $\mathbf{1}^T y = 0$ and in turn that $rank([P\ y]) = m - 1$. Thus, by basic linear algebra, there exists a solution y' such that $Py' = y$. Since $y' \in span^{\perp}\{\mathbf{1}\}$ implying that $\mathbf{1}^T y' = 0$, we have an augmented system of equation $[P^T\ \mathbf{1}]^T y' = [y\ 0]^T$. Since $rank([P^T\ \mathbf{1}]) = m$, again by basic linear algebra, we conclude that y' is unique. The proof for the reverse is similar.

B.2 Proof of Lemma 5.4

Since $\{x_k\}_{k\geq 0}$ has limit point $x^\star = \mathbf{1}\otimes\theta^\star$, it follows from (5.12b) that $\lim_{k\to\infty} y_{k+1} - y_k = -\frac{1}{\gamma}(I-W)x^\star = 0$. Recalling that $q_k = y_k - W(x_k - x_{k-1})$, from (5.12a) we have $q_k \in \gamma\partial f(x_k)\ \forall k$. Thus, by [29, Prop. 4.2.3] we know that the sequence $\{q_k\}_{k\geq 0}$ is bounded. In addition, we know that $\lim_{k\to\infty} q_{k+1} - q_k = \lim_{k\to\infty}(y_{k+1} - y_k) - W\lim_{k\to\infty}[(x_{k+1} - x_k) - (x_k - x_{k-1})] = 0$. By stardard analysis for weak cluster points, we claim that $\{q_k\}_{k\geq 0}$ is a convergent sequence. Let $y^\star$ be its limit point. We have $y^\star \in \partial f(x^\star)$ again by [29, Prop. 4.2.3]. Further, by Conservation Property II (cf. Lemma 5.2), we have $\mathbf{1}^T y^\star = 0$. Hence, all the optimality conditions (5.23) are satisfied, meaning that $(x^\star, y^\star)$ is a saddle point to the *primal-dual* problem.

B.3 Proof of Lemma 5.5

Consider the sequence (5.42). Applying the relation recursively yields

$$\upsilon_k \leq \eta^k \upsilon_0 + \sum_{i=0}^{k-1} \eta^i \omega_{k-1-i}.$$

Taking square of both sides of the above equation gives

$$\begin{aligned} \upsilon_k^2 &\overset{(a)}{\leq} 2\eta^{2k}\upsilon_0^2 + 2\left(\sum_{i=0}^{k-1} \eta^i \omega_{k-1-i}\right)^2 \\ &\overset{(b)}{\leq} 2\eta^{2k}\upsilon_0^2 + 2\left(\sum_{i=0}^{k-1} (\eta^{\frac{i}{2}})^2\right)\left(\sum_{i=0}^{k-1} (\eta^{\frac{i}{2}}\omega_{k-1-i})^2\right) \\ &\leq 2\eta^{2k}\upsilon_0^2 + \frac{2}{1-\eta}\sum_{i=0}^{k-1} \eta^i \omega_{k-1-i}^2, \end{aligned} \tag{B.1}$$

where (a) follows from $(a+b)^2 \leq 2a^2 + 2b^2, \forall a, b \in \mathcal{R}$ and (b) due to the Cauchy-Schwarz inequality. Summing the above relations over k from 1 to t and adding υ_0^2 to both sides yields

$$\begin{aligned} \Upsilon_t^2 &\leq 2\sum_{k=0}^{t} \eta^{2k}\upsilon_0^2 + \frac{2}{1-\eta}\sum_{k=1}^{t}\sum_{i=0}^{k-1} \eta^i \omega_{k-1-i}^2 \\ &= \frac{2}{1-\eta^2}\upsilon_0^2 + \frac{2}{1-\eta}\sum_{i=0}^{t-1}\sum_{k=0}^{t-1-i} \eta^k \omega_i^2 \\ &\leq \frac{2}{1-\eta^2}\upsilon_0^2 + \frac{2}{1-\eta}\left(\sum_{k=0}^{t-1} \eta^k\right)\left(\sum_{i=0}^{t} \omega_i^2\right) \\ &\leq \frac{2}{1-\eta^2}\upsilon_0^2 + \frac{2}{(1-\eta)^2}\Omega_t^2. \end{aligned} \tag{B.2}$$

Taking the square roots of both sides and using the relation $\sqrt{a^2+b^2} \leq a+b, \forall a, b \in \mathcal{R}$, we obtain

$$\Upsilon_t \leq \sqrt{\frac{2}{1-\eta^2}}\upsilon_0 + \frac{\sqrt{2}}{1-\eta}\Omega_t.$$

Changing t to k and letting $p = \frac{\sqrt{2}}{1-\eta}$ and $q = \sqrt{\frac{2}{1-\eta^2}}\upsilon_0$ completes the proof.

B.4 Proof of Lemma 5.7

Consider the dynamic average consensus step (5.38). Subtracting (5.44) from (5.39b) and letting $J = W - \frac{\mathbf{1}\mathbf{1}^T}{m}$, we have

$$\|\tilde{y}_{k+1}\| = \|J\tilde{y}_k + J\Delta g_k\| \leq \eta \|\tilde{y}_k\| + \eta \|\Delta g_k\| . \tag{B.3}$$

Then, by Assumption 5.7 and Remark 5.11 we have

$$\begin{aligned}
&\|\tilde{y}_{k+1}\| \leq \eta(\|\tilde{y}_k\| + \|\Delta g_k\|) \\
&\leq \eta(\|\tilde{y}_k\| + L \|x_{k+1} - x_k\|) \\
&= \eta(\|\tilde{y}_k\| + L \|\tilde{x}_{k+1} + \bar{x}_{k+1} - \bar{x}_k - \tilde{x}_k\|) \\
&\leq \eta \|\tilde{y}_k\| + \eta L \left(\|\tilde{x}_{k+1}\| + \|\overline{\gamma \odot y}_k\| + \|\tilde{x}_k\|\right),
\end{aligned} \tag{B.4}$$

where we have employed the relation (5.43) to substitute the third term of the last inequality. Now, let us consider the local update step (5.37). Similar with the above analysis for y-update, using (5.43) and (5.39a) we obtain for all $k \geq 0$

$$\begin{aligned}
&\|\tilde{x}_{k+1}\| = \|x_{k+1} - \bar{x}_{k+1}\| \\
&= \|J(x_k - \bar{x}_k) - J(\gamma \odot y_k)\| \\
&= \|J(x_k - \bar{x}_k) - J(\gamma \odot y_k - \bar{\gamma} \odot \bar{y}_k)\| \\
&= \|J(x_k - \bar{x}_k) - J(\gamma \odot \tilde{y}_k + \tilde{\gamma} \odot \bar{y}_k)\| \\
&\leq \eta \|\tilde{x}_k\| + \eta \|\gamma \odot \tilde{y}_k\| + \eta/\sqrt{m} \|\tilde{\gamma}\| \|\bar{y}_k\| \\
&\leq \eta \|\tilde{x}_k\| + \eta\gamma_{max} \|\tilde{y}_k\| + \eta \|\tilde{\gamma}\| / \|\bar{\gamma}\| \|\bar{\gamma} \odot \bar{y}_k\| \\
&\leq \eta \|\tilde{x}_k\| + \eta\gamma_{max}(1 + \Delta_\gamma) \|\tilde{y}_k\| + \eta\Delta_\gamma \|\overline{\gamma \odot y}_k\| ,
\end{aligned} \tag{B.5}$$

where we have used Proposition 5.1-(ii) for the third inequality and Proposition 5.1-(iv) as well as the definition of HoS (cf. Definition 5.7) for the last inequality. Combining (B.4) and (B.5) yields (note that $\eta < 1$)

$$\begin{aligned}
\|\tilde{y}_{k+1}\| \leq{}& \eta \|\tilde{y}_k\| + \eta L(\eta \|\tilde{x}_k\| + \gamma_{max}(1 + \Delta_\gamma) \|\tilde{y}_k\| \\
&+ \Delta_\gamma \|\overline{\gamma \odot y}_k\| + \|\overline{\gamma \odot y}_k\| + \|\tilde{x}_k\|) \\
={}& \eta' \|\tilde{y}_k\| + \eta L(1 + \eta) \|\tilde{x}_k\| + \eta L(1 + \Delta_\gamma) \|\overline{\gamma \odot y}_k\| ,
\end{aligned} \tag{B.6}$$

where $\eta' = \eta + \gamma_{max} L(1 + \Delta_\gamma)$.
Then, let us first consider (B.5) and let

$$w_k = \eta\gamma_{max}(1 + \Delta_\gamma) \|\tilde{y}_k\| + \eta\Delta_\gamma \|\overline{\gamma \odot y}_k\|$$

and $\beta = \gamma_{max} L$. Since $\eta < 1$ by Assumption 5.5, then by Lemma 5.5 we have

$$X_k \leq \rho_1 Y_k + p_1 Z_k + q_1, \tag{B.7}$$

where $\rho_1 = \frac{\sqrt{2}\eta\gamma_{max}(1+\Delta_\gamma)}{1-\eta}$, $p_1 = \frac{\sqrt{2}\Delta_\gamma\eta}{1-\eta}$, $q_1 = \frac{\sqrt{2}\|\tilde{x}_0\|}{\sqrt{1-\eta^2}}$.
Likewise, since $\beta < \frac{(1-\eta)^2}{(1+\Delta_\gamma)(2\eta^3+2\eta^2-\eta+1)} < \frac{1-\eta}{1+\Delta_\gamma}$ and thus $\eta' < 1$, using (B.6) and invoking Lemma 5.5 it follows that

$$Y_k \leq \rho_2 X_k + p_2 Z_k + q_2, \tag{B.8}$$

where $\rho_2 = \frac{\sqrt{2}\eta(1+\eta)L}{(1-\eta')}$, $p_2 = \frac{\sqrt{2}\eta(1+\Delta_\gamma)L}{(1-\eta')}$, $q_2 = \frac{\sqrt{2}\|\tilde{y}_0\|}{\sqrt{1-\eta'^2}}$. Since $\beta < \frac{(1-\eta)^2}{(1+\Delta_\gamma)(2\eta^3+2\eta^2-\eta+1)}$ and thus $\rho_1\rho_2 < 1$, combining (B.7) and (B.8) completes the proof.

B.5 Proof of Lemma 6.1

The first two inequalities follow from the Minkowski inequality and Cauchy-Schwarz inequality respectively [94]. For the third relation, using the smoothing lemma [94] and knowing that A is independent of x we have

$$\begin{aligned} E\left[\|Ax\|^2\right] &= E[x^T E[A^T A|x]x] \\ &= E[x^T E[A^T A]x] \\ &\leq \rho(E[A^T A])E[x^T x], \end{aligned} \tag{B.9}$$

where $\rho(\cdot)$ is the spectral radius. With the definition of the induced norm of matrix $\|A\|_E = \sup_{\|x\|_E=1} \|Ax\|_E$, taking the square root of both sides completes the proof.

B.6 Proof of Lemma 6.6

Notice that $W_k\Pi_{||} = \Pi_{||}W_k = \Pi_{||}$, $\forall k \geq 0$ by Assumption 6.3. Subtracting (5.43) from (6.17a) yields for all $k \geq 0$,

$$\tilde{x}_{k+1} = A_k\tilde{x}_k - A_k\left(\gamma_k \odot y_k\right), \tag{B.10}$$

where $A_k = (I - \Pi_{||})W_k = \Pi_{\perp}W_k$.

Then, taking the total expected norm of both sides gives

$$\begin{aligned}
\|\tilde{x}_{k+1}\|_E &= \|A_k\tilde{x}_k - A_k(\gamma_k \odot y_k)\|_E \\
&\overset{(a)}{=} \|A_k\tilde{x}_k - A_k\left(\gamma_k \odot y_k - \bar{\gamma}_k \odot \bar{y}_k\right)\|_E \\
&\overset{(b)}{=} \|A_k\tilde{x}_k - A_k\left(\gamma_k \odot \tilde{y}_k + \tilde{\gamma}_k \odot \bar{y}_k\right)\|_E \\
&\overset{(c)}{=} \|A_k\tilde{x}_k\|_E + \|A_k\|_E\,\|(\gamma_k \odot \tilde{y}_k + \tilde{\gamma}_k \odot \bar{y}_k)\|_E \\
&\overset{(d)}{\leq} \|A_k\tilde{x}_k\|_E + \|\gamma_k \odot \tilde{y}_k\|_E + \frac{\tilde{\sigma}_\gamma}{\sqrt{m}}\,\|\bar{y}_k\|_E \\
&\overset{(e)}{\leq} \eta\,\|\tilde{x}_k\|_E + \gamma_{\max}\,\|\tilde{y}_k\|_E + \Delta_\gamma\frac{\bar{\sigma}_\gamma}{\sqrt{m}}\,\|\bar{y}_k\|_E\,,
\end{aligned} \tag{B.11}$$

where (a) is clear, (b) due to Prop. 5.1-(ii), (c) and (d) obtained using Lemma 6.1 and knowing that $\|A_k\|_E \leq 1$, while in (e) we have used Lemma 1 to obtain the first term (note that A_k is independent of $\tilde{x}_k$) and the following relation

$$\|\gamma_k \odot \tilde{y}_k\|_E = \|\mathrm{diag}\{\gamma_k\}\tilde{y}_k\|_E \leq \gamma_{\max}\,\|\tilde{y}_k\|_E$$

as well as the fact that $\tilde{\sigma}_\gamma = \Delta_\gamma\bar{\sigma}_\gamma$ (cf. Definition 5.7) for the second term and third term respectively.

Similarly, following the same argument as in obtaining (B.10), we can deduce from (6.17b) and (5.44) that

$$\tilde{y}_{k+1} = A_k\tilde{y}_k + \Pi_{\perp}\Delta g_k. \tag{B.12}$$

Taking the total expected norm of both sides yields

$$\begin{aligned}
&\|\tilde{y}_{k+1}\|_E \le \|A_k\tilde{y}_k\|_E + \|\Pi_\perp \Delta g_k\|_E \\
&\overset{(a)}{\le} \eta \|\tilde{y}_k\|_E + L\|\Pi_\perp\|_E \|x_{k+1} - x_k\|_E \\
&\overset{(b)}{\le} \eta \|\tilde{y}_k\|_E + L\|\tilde{x}_{k+1} - \overline{\gamma_k \odot y_k} - \tilde{x}_k\|_E \\
&\le \eta \|\tilde{y}_k\|_E + L\left(\|\tilde{x}_{k+1}\|_E + \|\overline{\gamma_k \odot y_k}\|_E + \|\tilde{x}_k\|_E\right) \\
&\overset{(c)}{\le} \eta \|\tilde{y}_k\|_E + L\left(\|\tilde{x}_{k+1}\|_E + \Delta_\gamma \frac{\bar{\sigma}_\gamma}{\sqrt{m}} \|\tilde{y}_k\|_E + \frac{\bar{\sigma}_\gamma}{\sqrt{m}} \|\bar{y}_k\|_E + \|\tilde{x}_k\|_E\right),
\end{aligned} \tag{B.13}$$

where (a) is obtained from Lemma 6.1 and Assumption 5.7, (b) due to (5.43) and (c) deduced from (6.24) and $\tilde{\sigma}_\gamma = \Delta_\gamma \bar{\sigma}_\gamma$.
Combining (B.11) and (B.13) yields

$$\begin{aligned}
&\|\tilde{y}_{k+1}\|_E \\
&\le \left(\eta + \gamma_{\max} L + \Delta_\gamma L \frac{\bar{\sigma}_\gamma}{\sqrt{m}}\right) \|\tilde{y}_k\|_E + (1+\eta)L\|\tilde{x}_k\|_E + (1+\Delta_\gamma)L\frac{\bar{\sigma}_\gamma}{\sqrt{m}}\|\bar{y}_k\|_E \\
&\le \eta' \|\tilde{y}_k\|_E + (1+\eta)L\|\tilde{x}_k\|_E + (1+\Delta_\gamma)L\frac{\bar{\sigma}_\gamma}{\sqrt{m}}\|\bar{y}_k\|_E,
\end{aligned} \tag{B.14}$$

where $\eta' = \eta + (1+\Delta_\gamma)\beta$ and we have used the fact that $\frac{\bar{\sigma}_\gamma}{\sqrt{m}} \le \gamma_{\max}$ to obtain the last inequality.
Consider the relation (B.11). Let

$$w_k = \gamma_{\max}\|\tilde{y}_k\|_E + \Delta_\gamma \frac{\bar{\sigma}_\gamma}{\sqrt{m}}\|\bar{y}_k\|_E.$$

Then, by Lemma 5.5 we have

$$X_k^e \le \rho_1 Y_k^e + p_1 Z_k^e + q_1, \tag{B.15}$$

where $\rho_1 = \frac{\sqrt{2}\gamma_{\max}}{1-\eta}$, $p_1 = \frac{\sqrt{2}\Delta_\gamma\bar{\sigma}_\gamma}{(1-\eta)\sqrt{m}}$, $q_1 = \frac{\sqrt{2}\|\tilde{x}_0\|_E}{\sqrt{1-\eta^2}}$. Likewise, let $w_k' = (1+\eta)L\|\tilde{x}_k\|_E + (1+\Delta_\gamma)L\frac{\bar{\sigma}_\gamma}{\sqrt{m}}\|\bar{y}_k\|_E$. Noticing that $\eta' < 1$ since $\beta < \frac{(1-\eta)^2}{3+\eta+\Delta_\gamma(1-\eta)} < \frac{1-\eta}{1+\Delta_\gamma}$, it thus follows from (B.14) and Lemma 5.5 that

$$Y_k^e \le \rho_2 X_k^e + p_2 Z_k^e + q_2, \tag{B.16}$$

where $\rho_2 = \frac{\sqrt{2}(1+\eta)L}{1-\eta'}$, $p_2 = \frac{\sqrt{2}(1+\Delta_\gamma)L\bar{\sigma}_\gamma}{(1-\eta')\sqrt{m}}$, $q_2 = \frac{\sqrt{2}\|\tilde{y}_0\|_E}{\sqrt{1-\eta'^2}}$. Since $\beta < \frac{(1-\eta)^2}{3+\eta+\Delta_\gamma(1-\eta)}$ and thus $\rho_1\rho_2 < 1$. Then, combining (B.15) and (B.16) completes the proof.

B.7 Proof of Proposition 5.1

(i) It is clear by noting that $\Pi_{\|}^2 = \Pi_{\|}$, where $\Pi_{\|} = \frac{\mathbf{1}\mathbf{1}^T}{m}$;

(ii) $x \odot y - \bar{x} \odot \bar{y} = x \odot y - x \odot \bar{y} + x \odot \bar{y} - \bar{x} \odot \bar{y} = x \odot \tilde{y} + \tilde{x} \odot \bar{y}$;

(iii) Recall that $x = \bar{x} + \tilde{x}$ and $y = \bar{y} + \tilde{y}$. Then, we have $\overline{x \odot y} = \overline{(\bar{x} + \tilde{x}) \odot (\bar{y} + \tilde{y})} = \overline{\bar{x} \odot \bar{y}} + \overline{\bar{x} \odot \tilde{y}} + \overline{\tilde{x} \odot \bar{y}} + \overline{\tilde{x} \odot \tilde{y}} = \bar{x} \odot \bar{y} + \overline{\tilde{x} \odot \tilde{y}}$, where the last equality follows from the facts that $\Pi_{\|}\tilde{x} = \Pi_{\|}\tilde{y} = 0$.

(iv) Consider the right-hand side. $\|\overline{x \odot y}\| = \left\| \frac{\langle x,y \rangle}{m} \odot \mathbf{1} \right\| = \frac{1}{\sqrt{m}} \|\langle x, y \rangle\| \leq \frac{1}{\sqrt{m}} \|x\| \|y\|$. For the left-hand side, using (iii) we have $\|\overline{x \odot y}\| = \left\|\bar{x} \odot \bar{y} + \overline{\tilde{x} \odot \tilde{y}}\right\| \geq \|\bar{x} \odot \bar{y}\| - \left\|\overline{\tilde{x} \odot \tilde{y}}\right\| \geq \frac{1}{\sqrt{m}}(\|\bar{x}\| \|\bar{y}\| - \|\tilde{x}\| \|\tilde{y}\|)$;

(v) It is clear from the definition of projection matrix and Cauchy-Schwarz Inequality.

B.8 Proof of Proposition 5.3

Since $\mathbf{1}^T y = 0$ and $null(I - W) = span\{\mathbf{1}\}$, together with (5.23a) we have $\psi(x^\star, y) = f(x^\star) - y^T x^\star + \frac{1}{2\gamma} \|x^\star\|_{I-W}^2 = f(x^\star) = \psi(x^\star, y^\star)$. Thus, the left-hand side of condition (5.1) is proved. Then, from (5.23a) and (5.23c), we have $0 \in \partial f(x^\star) + \frac{1}{\gamma}(I - W)x^\star - y^\star = \partial \psi_x(x^\star, y^\star)$, which implies that $x^\star$ minimizes $\psi(x, y^\star)$. Thus, we prove the right-hand side of condition (5.1). Conversely, assume that $(x^\star, y^\star)$ is a saddle point such that the condition (5.1) holds. Then, from the left-hand side of condition (5.1) we have

$$\begin{aligned} \sup_{\mathbf{1}^T y = 0} \psi(x^\star, y) &= \sup_{\mathbf{1}^T y = 0} f(x^\star) - y^T x^\star + \frac{1}{2\gamma} \|x^\star\|_{I-W}^2 \\ &= \psi(x^\star, y^\star) < \infty, \end{aligned}$$

which is only possible when $x^\star \in span\{\mathbf{1}\}$. Thus we have $x^\star \in null\{I - W\}$ or namely $(I - W)x^\star = 0$. In addtion, from the righthand side of condition (5.1), we know that $x^\star$ minimizes $\psi(x, y^\star)$, implying that $0 \in \partial f(x^\star) + \frac{1}{\gamma}(I - W)x^\star - y^\star$. Since we have shown that $(I - W)x^\star = 0$, we have $0 \in \partial f(x^\star) - y^\star$, i.e., $y^\star \in \partial f(x^\star)$. Since $y^\star \in span^\perp\{\mathbf{1}\}$ and thus $\mathbf{1}^T y^\star = 0$, all the optimality conditions (5.23) hold.

Moreover, since the above analysis shows that every saddle point satisfies the optimality conditions (5.23), it follows from [54, Th. 19.1] that $x^\star$ is a primal solution to the OCP problem and $y^\star$ is a dual solution to the OEP problem respectively.

B.9 Proof of Proposition 5.4

From the definition of Lagrangian (5.13), recalling that $q_k = \gamma y_k - W(x_k - x_{k-1})$ and $f^\star = \psi(x^\star, y^\star)$ we have

$$
\begin{aligned}
&\gamma\left(\psi(x_k, y_k) - f^\star\right) \\
&= \gamma f(x_k) - \gamma y_k^T x_k + \frac{1}{2}\left\|\tilde{x}_k\right\|_{I-W}^2 - \gamma f(x^\star) \\
&= -\left[\gamma f(x^\star) - \gamma f(x_k) - \langle q_k, x^\star - x_k\rangle\right] - \langle W(x_k - x_{k-1}), x_k - x^\star\rangle + \frac{1}{2}\left\|\tilde{x}_k\right\|_{I-W}^2 \\
&= -D_{\gamma f}^{q_k}(x^\star, x_k) + \langle W(x_k - x_{k-1}), x_k - x^\star\rangle + \frac{1}{2}\left\|\tilde{x}_k\right\|_{I-W}^2 \\
&\leq \langle W(x_k - x_{k-1}), x_k - x^\star\rangle + \frac{1}{2}\left\|\tilde{x}_k\right\|_{I-W}^2 \\
&\leq \left\|x_k - x_{k-1}\right\|_W \left\|x_k - x^\star\right\|_W + \frac{1}{2}\left\|\tilde{x}_k\right\|_{I-W}^2,
\end{aligned}
\tag{B.17}
$$

where we have used the non-negativity of Bregman distance to obtain the second last inequality. Dividing by γ both sides of the above relation yields the claim.

Author's Publications

Journal Articles

- **Jinming Xu** and Yeng Chai Soh, "A Distributed Simultaneous Perturbation Approach for Large-scale Dynamic Optimization Problems," *accepted by Automatica as a Regular Paper.*
- **Jinming Xu**, Shanying Zhu, Yeng Chai Soh and Lihua Xie, "A Bregman Splitting Algortihm for Distributed Optimization Problems over Networks," *submitted to IEEE Transactions on Signal Processing.*
- **Jinming Xu**, Shanying Zhu, Yeng Chai Soh and Lihua Xie, "Convergence of Asynchronous Distributed Gradient Methods over Stochastic Networks," *submitted to IEEE Transactions on Automatic Control.*

Conference Proceedings

- **Jinming Xu** and Yeng Chai Soh, "Distributed Extremum Seeking Control of Networked Large-scale Systems under Constraints," in *Proceedings of 52nd IEEE Conference on Decision and Control, 2013.*
- **Jinming Xu**, Shanying Zhu, Yeng Chai Soh and Lihua Xie, "Augmented Distributed Gradient Methods for Multi-agent Optimization under Uncoordinated Constant Stepsizes," in *Proceedings of 54th IEEE Conference on Decision and Control, 2015.*
- **Jinming Xu**, Shanying Zhu, Yeng Chai Soh and Lihua Xie, "A forward-backward Bregman splitting scheme for regularized distributed optimization problems," *submitted to CDC 2016.*

Bibliography

[1] J. Baillieul and P. J. Antsaklis. Control and communication challenges in networked real-time systems. In *Proceedings of the IEEE*, volume 95, pages 9–28, Jun 2007.

[2] A. Aquino-Lugo. *Distributed and decentralized control of the power grid*. PhD thesis, University of Illinois at Urbana-Champaign, 2010.

[3] R. L. Raffard, C. J. Tomlin, and S. P. Boyd. Distributed optimization for cooperative agents: Application to formation flight. In *Proceedings of 43rd IEEE Conference on Decision and Control*, pages 2453–2459, Dec. 2004.

[4] M. Rabbat and R. Nowak. Distributed optimization in sensor networks. In *Proceedings of 3rd International Symposium on Information Processing in Sensor Networks*, pages 20–27, April 2004.

[5] L. Xiao, M. Johansson, and S. P. Boyd. Simultaneous routing and resource allocation via dual decomposition. *IEEE Trans. Commun.*, 52(7):1136–1144, July 2004.

[6] J. Chen and A. H. Sayed. Diffusion adaptation strategies for distributed optimization and learning over networks. *IEEE Trans. Signal Process*, 60(8): 4289–4305, May 2012.

[7] J. B. Predd, S. R. Kulkarni, and H. V. Poor. A collaborative training algorithm for distributed learning. *IEEE Trans. Inf. Theory*, 55(4):1856–1871, April 2009.

[8] J. M. Maestre, D. Muoz de la Pea, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32:153–176, 2011.

[9] J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans. Autom. Control*, 57(3):592–606, March 2012.

[10] D. Jakovetic, J. Xavier, and J. M. F. Moura. Fast distributed gradient methods. *IEEE Trans. Autom. Control*, 59(5):1131–1146, May 2014.

[11] A. Nedic and A. Olshevsky. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *arXiv:1406.2075*, Jun 2014.

[12] J. B. Rawlings and B. T. Stewart. Coordinating multiple optimization-based controllers: New opportunities and challenges. *Journal of Process Control*, 18: 839–845, 2008.

[13] W. B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Trans. Autom. Control*, 52:1249–1263, July 2007.

[14] T. Keviczky, F. Borrelli, and G. J. Balas. Decentralized receding horizon control for large scale dynamically decoupled systems. *Automatica*, 42:2105–2115, 2006.

[15] F. Borrelli and T. Keviczky. Distributed LQR design for identical dynamically decoupled systems. *IEEE Trans. Autom. Control*, Sep. 2008.

[16] K. B. Ariyur and M. Krstic. *Real-time optimization by Extremum-seeking Control*. Hoboken, NJ : Wiley Interscience, 2003.

[17] Y. Tan, W. Moase, C. Manzie, D. Neic, and I. Mareels. Extremum seeking from 1922 to 2010. In *Proceedings of 29th Chinese Control Conference*, pages 14–26, July 2010.

[18] M. S. Stankovic, K. H. Johansson, and D. M. Stipanovic. Distributed seeking of nash equilibria with applications to mobile sensor networks. *IEEE Trans. Autom. Control*, 57(4):904–919, April 2012.

[19] P. Frihauf, M. Krstic, and T. Basar. Nash equilibrium seeking in noncooperative games. *IEEE Trans. Autom. Control*, 57(5):1192–1207, Oct. 2012.

[20] T. Basar and G. J. Olsder. *Dynamic noncooperative game theory*. Society for Industrial Mathematics, 1999.

[21] S. L. Waslander, G. Inalhan, and C. J. Tomlin. Decentralized optimization via Nash bargaining. In *Theory and Algorithms for Cooperative Systems*, pages 565–583. World Scientific Pub. Co. Inc., Destin, 2003.

[22] E. Semsar-Kazerooni and K. Khorasani. Multi-agent team cooperation: A game theory approach. *Automatica*, 45:2205–2213, 2009.

[23] J. Tsitsiklis, D. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. Autom. Control*, 31(9):803–812, Sep. 1986.

[24] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. In *Proceedings of the IEEE*, volume 95, pages 215–233, Jan 2007.

[25] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2508–2530, June 2006.

[26] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 482–491, Oct 2003.

[27] A. Nedic and A. Ozdaglar. Distributed subgradient methods for Multi-agent optimization. *IEEE Trans. Autom. Control*, 54(1):48–61, Jan 2009.

[28] H. Terelius, U. Topcu, and R. M. Murray. Decentralized multi-agent optimization via dual decomposition. In *Proceedings of the 18th IFAC World Congress*, number 18, pages 11245–11251, Aug 2011.

[29] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[30] S. H. Low and D. E. Lapsley. Optimization flow control. I. Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7:861–874, 1999.

[31] J. Xu and Y. C. Soh. Distributed extremum seeking control of networked large-scale systems under constraints. In *Proceedings of IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 2187–2192, 2013.

[32] K. Kvaternik and L. Pavel. An analytic framework for decentralized extremum seeking control. In *Proceedings of American Control Conference (ACC), 2012*, pages 3371–3376, 2012.

[33] M. Krstic and H. H. Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36:595–601, 2000.

[34] Y. Tan, D. Neic, and I. Mareels. On non-local stability properties of extremum seeking control. *Automatica*, 42:889–903, 2006.

[35] M. Rotea. Analysis of multivariable extremum seeking algorithms. In *Proceedings of the 2000 American Control Conference.*, pages 433–437, 2000.

[36] J. C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998.

[37] Nusawardhana and S. H. Zak. Simultaneous perturbation extremum seeking method for dynamic optimization problems. In *Proceedings of the 2004 American Control Conference*, volume 3, pages 2805–2810, 2004.

[38] A. Ghaffari, M. Krstic, and D. Nešić. Multivariable newton-based extremum seeking. *Automatica*, 48(8):1759–1767, 2012.

[39] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato. Newton-Raphson consensus for distributed convex optimization. In *Proceedings of 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 5917–5922, Dec 2011.

[40] A. Nedic, A. Ozdaglar, and P. A. Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Trans. Autom. Control*, 55(4):922–938, April 2010.

[41] M. Zhu and S. Martinez. On distributed convex optimization under inequality and equality constraints. *IEEE Trans. Autom. Control*, 57(1):151–164, Jan 2012.

[42] D. Yuan, S. Xu, and H. Zhao. Distributed Primal-Dual subgradient method for Multi-agent optimization via consensus algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41:1715–1724, 2011.

[43] K. Srivastava and A. Nedic. Distributed asynchronous constrained stochastic optimization. *IEEE J. Sel. Topics Signal Process.*, 5(4):772–790, Aug 2011.

[44] A. Nedic and A. Olshevsky. Distributed optimization over time-varying directed graphs. *IEEE Trans. Autom. Control*, 60(3):601–615, March 2015.

[45] B. Gharesifard and J. Cortes. Distributed continuous-time convex optimization on weight-balanced digraphs. *IEEE Trans. Autom. Control*, 59(3):781–786, March 2014.

[46] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta numerica*, 14(1):1–137, 2005.

[47] M. Fortin and R. Glowinski. *Augmented Lagrangian methods: Applications to the numerical solution of boundary-value problems*. Elsevier, 2000.

[48] 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, Jan 2011. ISSN 1935-8237.

[49] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Trans. Signal Process*, 61(10):2718–2723, May 2013.

[50] E. Wei and A. E. Ozdaglar. Distributed alternating direction method of multipliers. In *Proceedings of IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 5445–5450, 2012.

[51] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Trans. Signal Process*, 62(7):1750–1761, Feb. 2014.

[52] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.*, 14(5):877–898, 1976.

[53] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

[54] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer Science & Business Media, 2011.

[55] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[56] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[57] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[58] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–343, 2009.

[59] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM J. Imaging Sci.*, 3(3):253–276, 2010.

[60] K. Srivastava, A. Nedić, and D. Stipanović. Distributed bregman-distance algorithms for min-max optimization. In *Agent-Based Optimization*, pages 143–174. Springer, 2013.

[61] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM J. Imaging Sci.*, 1(1):143–168, 2008.

[62] A. Nedic. Asynchronous broadcast-based convex optimization over a network. *IEEE Trans. Autom. Control*, 56(6):1337–1351, June 2011.

[63] E. Wei and A. Ozdaglar. On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 551–554, 2013.

[64] J. Wang and N. Elia. A control perspective for centralized and distributed convex optimization. In *Proceedings of 50th IEEE Conference on Decision and Control and European Control Conference*, pages 3800–3805, Dec 2011.

[65] D. Jakovetic, J. M. F. Moura, and J. Xavier. Linear convergence rate of a class of distributed augmented lagrangian algorithms. *IEEE Trans. Autom. Control*, 60(4):922–936, Oct 2015.

[66] W. Shi, Q. Ling, G. Wu, and W. Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25 (2):944–966, 2015.

[67] W. Shi, Q. Ling, G. Wu, and W. Yin. A proximal gradient algorithm for decentralized composite optimization. *IEEE Trans. Signal Process.*, 63(22): 6013–6023, Nov 2015.

[68] P. Bianchi and W. Hachem. A primal-dual algorithm for distributed optimization. In *Proceedings of 2014 IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 4240–4245, 2014.

[69] I. Lobel and A. Ozdaglar. Distributed subgradient methods for convex optimization over random networks. *IEEE Trans. Autom. Control*, 56(6):1291–1306, Nov 2011.

[70] F. Zanella, D. Varagnolo, A. Cenedese, G. Pillonetto, and L. Schenato. Asynchronous Newton-Raphson consensus for distributed convex optimization. In *Proceedings of 3rd IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys 12)*, pages 133–138, 2012.

[71] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem. Asynchronous distributed optimization using a randomized alternating direction method of multipliers. In *Proceedings of IEEE 52nd Annual Conference on Decision and Control (CDC)*, pages 3671–3676, 2013.

[72] M. H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[73] G. C. Walsh, H. Ye, and L. G. Bushnell. Stability analysis of networked control systems. *IEEE Trans. Control Syst. Technol.*, 10(3):438–446, 2002.

[74] D. Nešić and A. R. Teel. Input-to-state stability of networked control systems. *Automatica*, 40(12):2121–2128, 2004.

[75] A. Dembo and T. Kailath. Model-free distributed learning. *IEEE Trans. Neural Netw.*, 1(1):58–70, Mar 1990.

[76] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[77] H. K. Khalil. *Nonlinear systems*. Prentice Hall, 2002.

[78] D Nešić and AR Teel. Input-to-state stability for nonlinear time-varying systems via averaging. *Mathematics of Control, Signals and Systems*, 14(3):257–280, 2001.

[79] A. R. Teel, L. Moreau, and D. Nešić. A unified framework for input-to-state stability in systems with two time scales. *IEEE Trans. Autom. Control*, 48 (9):1526–1544, Sep 2003.

[80] C. Manzie and M. Krstic. Extremum seeking with stochastic perturbations. *IEEE Trans. Autom. Control*, 54(3):580–585, March 2009.

[81] S. Z. Khong, Y. Tan, C. Manzie, and D. Nešić. Extremum seeking of dynamical systems via gradient descent and stochastic approximation methods. *Automatica*, 56:44–52, 2015.

[82] D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Dynamic consensus on mobile networks. In *IFAC world congress*. Prague Czech Republic, 2005.

[83] M. Zhu and S. Martinez. Discrete-time dynamic average consensus. *Automatica*, 46(2):322–329, 2010.

[84] J. R. Marden, S. D. Ruben, and L. Y. Pao. A model-free approach to wind farm control using game theoretic methods. *IEEE Trans. Control Syst. Technol.*, 21(4):1207–1214, May 2013.

[85] P. M. O. Gebraad and J. W. Wingerden. Maximum power-point tracking control for wind farms. *Wind Energy*, 18(3):429–447, 2015.

[86] K. E. Johnson and N. Thomas. Wind farm control: Addressing the aerodynamic interaction among wind turbines. In *Proceedings of the 2009 American Control Conference*, pages 2104–2109, 2009.

[87] L. Y. Pao and K. E. Johnson. Control of wind turbines. *IEEE Control Systems*, 31(2):44–62, March 2011.

[88] A. Ghaffari, M. Krstic, and S. Seshagiri. Power optimization and control in wind energy conversion systems using extremum seeking. *IEEE Trans. Control Syst. Technol.*, 22(5):1684–1695, Sep 2014.

[89] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[90] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[91] L. Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.*, 158(2):460–479, 2013.

[92] S. Kar and J. M. F. Moura. Sensor networks with random links: Topology design for distributed consensus. *IEEE Trans. Signal Process*, 56(7):3315–3326, July 2008.

[93] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *arXiv preprint arXiv:1310.7063*, 2013.

[94] A. Gut. *Probability: A Graduate Course.* New York, NY : Springer Science+Business Media, Inc., 2005.

[95] B. T. Polyak. *Introduction to Optimization.* New York: Optimization Software Inc., 1987.