2019

Time-varying optimization using primal-dual type methods: Solution tracking and model training

Yijian Zhang
Iowa State University

Time-varying optimization using primal-dual type methods: Solution tracking and model training

by

Yijian Zhang

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:
Mingyi Hong, Co-major Professor
Lizhi Wang, Co-major Professor
Guiping Hu
Sarah Ryan
Zhengdao Wang

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2019

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this dissertation. First and foremost, I would like to express my gratitude to my advisor Dr. Mingyi Hong for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge.

I would also like to thank my co-advisor Dr. Lizhi Wang and the rest of my committee members Dr. Guiping Hu, Dr. Sarah Ryan and Dr. Zhengdao Wang for their insightful comments and encouragement, also for the hard question which incented me to improve my dissertation from various perspectives. I would additionally like to thank Dr. Emiliano Dall'Anese from University of Colorado for his guidance throughout our collaboration in the past four years.

ABSTRACT

This dissertation considers general time-varying optimization problems that arise in many network control and machine learning applications. The problem takes the form of minimizing the sum of separable convex/non-convex cost functions subject to equality constraints, and both the objective and constraints are parameterized by some time-varying parameters. To cope with the problem dynamics, we design dynamic/stochastic algorithms based on primal-dual type methods, e.g. alternating direction method of multipliers (ADMM) and primal-dual decomposition (PDD). Depending on specific application, our algorithms can accomplish two major tasks: i) continuously track optimal solutions for each time instance; ii) learn the general pattern of given data and produce one solution that fits all time-varying parameters.

The first part of the dissertation focuses on problems with changing optimal solutions. Specifically, our considered problem is changing in real time and no iterative algorithm can solve to convergence for the smallest time interval. We aim at designing algorithms that can run limited iterations for each time instance and still stay close to optimal solutions. To this end, we design a primal-dual type method based on ADMM, where we leverage proximal gradient in the primal steps, and modify the dual steps by adding some perturbation to accommodate the time-varying nature of the problem. We show that, under mild assumptions, the proposed algorithm is able to track the change of problem, meaning it will always stay in a neighborhood around the optimal or approximate optimal solutions for each time instance. Moreover, our analysis indicates an interesting trade-off between solution accuracy and convergence speed. In cases where gradient information is not available or difficult to compute, we develop a suitable time-varying algorithm by using only function value information (also known as the zeroth-order information). Through a two-point estimation of gradient, we observe similar performance as gradient based methods and convergence in expectation is proved under suitable assumptions. As an extension of this time-varying framework, static optimization with randomness in updates are discussed with applications in power systems. Specifically, an ADMM-based distributed optimal power flow (OPF) controller for renewable energy source (RES) is designed to steer RES output powers to approximate AC OPF solutions.

The second part of the dissertation, we further discover that the time-varying framework is also applicable to cases where all changing parameters can fit one solution, i.e. training. This type of problem is the core of many machine learning models that aiming at extracting data pattern. We specifically focus on deep neural network (DNN) and model the training of DNN into an equality constrained optimization problem by introducing auxiliary variables for each hidden layer. The resulting formulation is highly nonconvex and time-varying in that each time only part of the data is available and as time goes by data comes in sequentially. We design another primal-dual type method called double stochastic primal-dual decomposition (DSPD) for the neural network training problem. We demonstrate that the developed algorithm is effective by: 1) performing theoretical analysis to show that the stochastic DSPD algorithm can reach stationary solution of the training problem; 2) conducting comprehensive comparison with state-of-the-art algorithms and show that the proposed algorithm achieves some early stage advantage, that is, the training error decreases faster in the first a few iterations.

CHAPTER 1. GENERAL INTRODUCTION

In this dissertation we address dynamic/time-varying (distributed) optimization problems where both objective and constraints are time-varying [2] [3]. This is closely related to many engineering problems in power systems, signal processing and machine learning just to name a few. To proceed, we separate our discussion into 2 parts: solution tracking and model training. For the tracking task, we focus on keeping close to optimal solutions of a time-varying problem, while the training task aims at finding one solution that fits all time-varying parameters.

For the tracking task, there have been many works discussing how to model and control the system output so to continuously optimize overall or individual performance with respect to each network node, whilst meeting system and node constraints that evolve in real time [4–7]. To model the time-varying objective we consider the following general formulation:

$$\min_{\mathbf{x}\in\mathbb{R}^m,\mathbf{y}\in\mathbb{R}^n} \; f(\mathbf{x};t)+g(\mathbf{y};t) \tag{P1}$$

$$\text{s.t. } \mathbf{A}(t)\mathbf{x}+\mathbf{B}(t)\mathbf{y}=\mathbf{b}(t) \tag{1.1a}$$

$$\mathbf{x}\in\mathcal{X}(t),\mathbf{y}\in\mathcal{Y}(t), \tag{1.1b}$$

where $t\in\mathbb{R}^+$ is the time index; $f(\cdot,t),g(\cdot,t)$ are convex functions uniformly for all t. Depending on different applications, these functions are used to model costs such as system operation costs, or the (negative) business profits. For example, in power systems they are used to model the power losses and/or deviations from the nominal voltage profile and cost of/reward for ancillary service provisioning, respectively; $\mathbf{A}(t)\in\mathbb{R}^{\ell\times m},\mathbf{B}(t)\in\mathbb{R}^{\ell\times n}$ are time-varying matrices, $\mathcal{X}(t),\mathcal{Y}(t)$ are constraint sets that are convex uniformly in time and are easy to apply projection, e.g. bound or ball constraints. For specific problems with inequality constraints, one can always add slack variables to convert them to equality constraints and thus fit in the formulation of (1.1a). Problem (P1) represents a general time-varying optimization problem, meaning the optimal solutions are changing over time. Solving such problem requires algorithms to have the ability to track the optimal trajectory in real-time. Note that here we only present 1.1 in the form of two blocks

$\mathbf{x}, \mathbf{y}$ because it will have convergence guarantee as stated in later chapters. For problems with more than 2 blocks, there may not be a convergence guarantee, however, there is empirical evidence that they can still be solved to convergence for each time instance.

Previous efforts [8, 9] have treated dynamic problems as a series of static ones and assuming separate time scales so that convergence is reached for each discrete time. However, this is not realistic in practice especially when system parameters change in a fast pace that the algorithm will keep updating according to past information; [5, 6, 10] successfully approach dynamic problem in continuous time, but only for isolated systems where time-varying exogenous inputs that dispersed in the network are not considered.

In recent years, many works have focused on real-time implementation: [5] presents a control algorithm for real-time multi-agent systems with the ability to track optimal trajectory, however, only cost function is time-varying; [11] proposes an online algorithm for optimal power flow problem based on quasi-Newton method. It can be shown that proposed algorithm is able to provide suboptimal solution at a fast timescale. The tracking ability hinges on the accurate estimation of second order information; For the same application, [12] leverages dual subgradient method and system feedbacks to design a tracking algorithm called double smoothing. Regularization term is added in both primal and dual subproblems to prove Q-linear convergence to a neighborhood of optimal solution for each time instance; [3] [13] further extend double smoothing algorithm to more general settings and provide a thorough regret analysis.

Chapter 2: In light of the aforementioned works, this chapter proposes a general framework based on alternating direction method of multipliers (ADMM) [14] to solve time-varying problems aroused from network control applications. We modify ADMM by introducing perturbation to dual variables. This step gives us the privilege to exchange solution accuracy for convergence speed, which essentially helps guarantee tracking ability of our algorithm. We compare ADMM with state of the art method, i.e. dual subgradient method, and conclude that we are able to deal with a wider range of problems and improve stability, especially for problems with ill-conditioned dual functions [15] [16]. Tracking ability is proved in the sense that iterates generated from our algorithm is able to converge to a neighborhood of optimal solution in real time.

Chapter 3: In this chapter, we seek to build a *zeroth-order* dynamic distributed algorithm for *time-varying* optimization problems. Specifically, we consider cases where gradient of objective is computationally expensive or the explicit objective formulation is unknown, and we only have access to function values of the objective, i.e. zeroth-order information. This setup usually comes from network control applications where each user node wants to preserve privacy. For each time instance, we query the function values from a zeroth-order oracle and construct gradient estimations to feed to our algorithm. By doing this, we avoid expensive gradient evaluation and decrease computation complexity. The resulting algorithm is expected to perform faster (when gradient evaluation is expensive) and have comparable tracking ability as first-order methods (in which exact gradient information is required). We provide our design of algorithms and some numerical results on the same power system model.

Chapter 4: As a direct extension of this time-varying framework, static case optimization with randomness in updates is also studied. Specifically, an application in designing a distributed optimal power flow controller is presented, where we leverage the ADMM based time-varying optimization framework to steer the renewable energy source output powers to solutions of AC optimal power flow (OPF) problems. Convergence of the RES-inverter output powers to solutions of the approximate AC OPF problem is established under suitable conditions on the mismatches between the commanded setpoints and actual RES output powers. Overall, since the proposed scheme can be cast as an ADMM with inexact primal and dual updates, the convergence results can be applied to more general distributed optimization settings.

For the training task, we specifically focus on training of deep neural network. For the last two decades, the state-of-the-art training approach for DNNs has been the stochastic gradient descent (SGD) based methods [17], which are built based on using the chain rule to compute the gradient of the loss function w.r.t the weight parameters (i.e. backpropagation) [18]. The SGD is easy to implement and suitable for GPU computing. Extensive research has been done to improve training speed of SGD. The authors of [19, 20] introduce Polyak momentum and Nesterov momentum respectively that takes average of a few last iterates to accelerate convergence. In AdaGrad [21], AdaDelta [22], RMSProp [23] and Adam [24] the main focus is on using different adaptive learning rates, and those methods have been shown to achieve much improves performances in practice.

Despite all these efforts, the SGD-based algorithms still suffer from many numerical difficulties, among which the most challenging one is the *vanishing gradient* [25]. This difficulty comes from the fact that gradients for variables belonging to shallower layers are dependent on those for deeper layers (through nonlinear function composition), which makes the gradients decrease exponentially with the increased layers of the network, and eventually results in slow convergence. Numerous works have tried to address this issue. In [26] the authors found that non-saturating activation functions such as rectified linear units (ReLU) can help alleviate vanishing gradient problem. In [27–29] a normalized initialization is considered, while [30] considers intermediate normalization layers, enabling SGD to start converging for very deep networks. The work [31] proposes a new network called residual net that has "shortcut" connections among layers. The structure has a shorter path from input to output and consequently achieves significantly increased performs in practice.

Recently, a new training framework is developed in effort to avoid the process of error backpropagation [32]. The authors propose the idea of splitting neural network layers by introducing auxiliary variables. The training task is then formulated as an equality constrained optimization problem. Numerous works have leveraged this splitting idea, e.g. [33] proposes to use ADMM algorithm to solve the equality constrained problem and suggests that vanishing gradient problem may possibly be alleviated by this framework; [34,35] utilize block coordinate descent (BCD) algorithm to tackle the problem. Convergence is proved under suitable assumption on the objective function. Although all these works have shown superiority over SGD-based method, the comparison may not be fair because all aforementioned splitting layer works are batch algorithms. In practice, batch algorithm is not realistic due to data size and computation limit. To the best of our knowledge, there is no stochastic version primal-dual algorithm or stochastic BCD algorithm for training neural network. The corresponding convergence property also remains unknown.

Chapter 5: In this chapter, we further extend our time-varying framework to the application of deep learning. Training a neural network is to formulate an optimization problem and each time pass in part of the data to partially solve the problem. Therefore, this problem is essentially a different type of time-varying optimization with the goal of learning one set of solution instead of tracking multiple solutions. In this work, we also leverage the splitting layer idea and propose a novel training framework that only requires part of the data to be available at each time instance. The resulting algorithm can fully extract gradient information for variables of deeper layers at early stage of training without needing to go through the function composition across layers. A double stochastic primal-dual (DSPD) method is developed, in which the equality constraints are first relaxed, and then gradually tightened by having increased penalization as the algorithm proceeds. The DSPD enjoys rigorous convergence guarantees. Its inner loop is a (variance reduced) double stochastic block coordinate descent method, which converges with a sublinear rate; its outer loop executes certain dual ascent scheme, which ensures that DSPD converges to first-order stationary solution (almost surely). Numerical experiments on MNIST dataset demonstrate the effectiveness of the proposed framework.

CHAPTER 2. TIME-VARYING OPTIMIZATION WITH APPLICATIONS IN POWER SYSTEMS

2.1 Introduction

This chapter proposes the development of an online algorithm for time-varying convex problems based on alternating direction method of multipliers (ADMM) [14]. Using a quadratic regularization term, ADMM can allow one to deal with nonsmooth terms, and it exhibits improved convergence properties relative to dual (sub)gradient methods, especially for problems with ill-conditioned dual functions [15] [16]. The paper presents a new algorithm that has following characteristics: i) at each step the primal subproblems are solved via proximal gradient descent – providing favorable scalability to large-scale problems and accommodating non-smooth objectives; ii) a dual perturbation method is utilized, where the dual variables are suitably perturbed at every iteration to gain in convergence rate. Related works along this line include the following: [36] leverages ADMM to solve a real-time multi-agent problem. But it differs from the present work because it considers only consensus constraints (a special case of our general formulation); [37] considers a dynamic sharing problem, and convergence to a neighborhood is provided under standard assumptions; however, the constraint is also a special case of our formulation. Our previous work [38] applies ADMM to track a solution of a domain-specific optimal power flow problem; however, this work significantly extends [38] in the following ways: 1) we extend the work to a more general time-varying optimization problem (with applications not only in power systems); 2) a perturbed ADMM is proposed, and convergence is proved under milder conditions, which enables a wider range of constraints and objectives; 3) the resulting algorithm is able to incorporate feedback in multiple places to enable distributed implementation. Overall our work has the following contributions: i) An ADMM-based dynamic algorithm is proposed to solve a family of time-varying problems. ii) Tracking ability is rigorously proved under mild assumptions; nonsmooth terms are allowed in the objective and no full row-rank assumption is made for the constraints.

The remainder of paper is organized as follows. Section 2 will give the general time-varying problem formulation. Section 3 will introduce our dynamic algorithm. Section 4 will apply proposed algorithm to two applications, one is in power systems, the other one is route selection. Tracking ability is shown in 5 and 6 in the form of convergence analysis and simulation, respectively.

2.2 Problem Formulation

2.2.1 Problem Setup

Consider the time-varying problem (P1). Assume that the temporal domain is discretized as $t_k = k\tau$, $k \in \mathbb{N}$ and $\tau > 0$ is a given sampling time that is small enough to capture the dynamics. At time k, if the associated problem problem (P1) is solved to global optimality, then we say that a *perfect tracking* is achieved. However, in many applications (such as those to be specified shortly) such perfect tracking may not be possible because before the problem at time k[1] is solved, it may have already evolved to a new problem. Specifically, iterative algorithms often involve multiple iterations of computing and communication, and by the time algorithms converge for time k, problem parameters such as $\mathbf{A}^{(k)}, \mathbf{B}^{(k)}, \mathbf{b}^{(k)}$ might have already changed. Therefore it is desirable to design algorithms with certain "tracking ability", which means that the iterates can be continuously steered to stay close to the time-varying optimal solutions.

To derive algorithms that possess the above notion of "tracking" ability, let us reformulate problem (3.11) as follows. First, we rewrite the time-varying constraint sets $\mathcal{X}^{(k)}, \mathcal{Y}^{(k)}$ into indicator functions in the objective; and then we separate objective into non-differential functions $f_0^{(k)}(\mathbf{x}), g_0^{(k)}(\mathbf{y})$ and differential

[1] Throughout this paper, boldface characters denote vectors or matrices; characters with superscript (k) denote time-varying iterates and parameters; for a given vector $\mathbf{x}$ and matrix $\mathbf{G}$, $\|\mathbf{x}\|^2_{\mathbf{G}} := \mathbf{x}^T\mathbf{G}\mathbf{x}$; $< \mathbf{x}, \mathbf{y} >$ denotes the inner product between the vectors $\mathbf{x}$ and $\mathbf{y}$.

functions $f_1^{(k)}(\mathbf{x}), g_1^{(k)}(\mathbf{y})$. At time k we consider the following time-varying problem:

$$\min_{\mathbf{x}\in\mathbb{R}^m,\mathbf{y}\in\mathbb{R}^n} \quad f^{(k)}(\mathbf{x}) + g^{(k)}(\mathbf{y}) \tag{P2}$$

$$\text{s.t. } \mathbf{A}^{(k)}\mathbf{x} + \mathbf{B}^{(k)}\mathbf{y} = \mathbf{b}^{(k)} \tag{2.1a}$$

$$f^{(k)}(\mathbf{x}){:=}f_0^{(k)}(\mathbf{x}) + f_1^{(k)}(\mathbf{x}), g^{(k)}(\mathbf{y}){:=}g_0^{(k)}(\mathbf{y}) + g_1^{(k)}(\mathbf{y}),$$

$$\mathbf{A}^{(k)} := \mathbf{A}(t_k), \mathbf{B}^{(k)} := \mathbf{B}(t_k), \mathbf{b}^{(k)} := \mathbf{b}(t_k),$$

$$\mathcal{X}^{(k)} := \mathcal{X}(t_k), \mathcal{Y}^{(k)} := \mathcal{Y}(t_k).$$

Throughout the entire paper, our discussion on (P2) and proposed algorithm are based on the following assumption:

Assumption 1. *For each time k, (P2) is feasible.*

If Assumption 1 does to hold at a time k, the problem formulation would not be well posed since there is no solution trajectory to track. We further characterize the *drift* of the optimal solution as follows.

Assumption 2. *The successive difference between optimal solutions are bounded:*

$$\|\mathbf{x}^{*,(k+1)} {-} \mathbf{x}^{*,(k)}\| \leq \sigma_{\mathbf{x}}, \|\mathbf{y}^{*,(k+1)} - \mathbf{y}^{*,(k)}\| \leq \sigma_{\mathbf{y}}, \tag{2.2}$$

where $\mathbf{x}^{*,(k)}, \mathbf{y}^{*,(k)}$ are optimal solutions of (P2) at time k; $\sigma_{\mathbf{x}} > 0, \sigma_{\mathbf{y}} > 0$ are some constants. Also the problem parameters are bounded as:

$$\|\mathbf{A}^{(k+1)} - \mathbf{A}^{(k)}\| \leq \sigma_{\mathbf{A}}, \; \|\mathbf{B}^{(k+1)} - \mathbf{B}^{(k)}\| \leq \sigma_{\mathbf{B}} \tag{2.3}$$

$$\|\mathbf{A}^{(k)}\| \leq \tilde{\sigma}_{\mathbf{A}}, \; \|\mathbf{B}^{(k)}\| \leq \tilde{\sigma}_{\mathbf{B}}, \; \|\mathbf{b}^{(k+1)} - \mathbf{b}^{(k)}\| \leq \sigma_{\mathbf{b}} \tag{2.4}$$

where $\sigma_{\mathbf{A}}, \sigma_{\mathbf{B}}, \tilde{\sigma}_{\mathbf{A}}, \tilde{\sigma}_{\mathbf{B}}, \sigma_{\mathbf{b}}$ are some given positive constants.

Assumption 2 is common in time-varying optimization [2, 3, 6, 11]; worst-case bounds for (2.2) can be obtained assuming that the sets $\mathcal{X}^{(k)}, \mathcal{Y}^{(k)}$ are compact uniformly in time. Another approach is to measure the distance based on the *optimal drift*, without assuming a specific bound; see, e.g., [36,37]. The parameters $\sigma_{\mathbf{x}}$ and $\sigma_{\mathbf{y}}$ quantify maximum variation of the optimal solutions over two consecutive time steps, and they are always finite (because the problem is assumed to be always feasible); since the paper deals with a tracking problem, conventional wisdom would suggest that better tracking performance can be achieved when (P2) is not changing rapidly; this will be confirmed in the convergence results presented later in the paper.

Assumption 3. *For each time k, $f^{(k)}, g^{(k)}$ satisfy*

$$\langle \partial f^{(k)}(\mathbf{x}_1) - \partial f^{(k)}(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle \geq \tilde{v}_f \|\mathbf{x}_1 - \mathbf{x}_2\|, \forall \mathbf{x}_1, \mathbf{x}_2 \tag{2.5}$$

$$\langle \partial g^{(k)}(\mathbf{y}_1) - \partial g^{(k)}(\mathbf{y}_2), \mathbf{y}_1 - \mathbf{y}_2 \rangle \geq \tilde{v}_g \|\mathbf{y}_1 - \mathbf{y}_2\|, \forall \mathbf{y}_1, \mathbf{y}_2 \tag{2.6}$$

where $\tilde{v}_f, \tilde{v}_g$ are uniform lower bounds of strongly convex constants for $f^{(k)}, g^{(k)}$. Functions $f_1^{(k)}, g_1^{(k)}$ have Lipschitz continuous gradients,

$$\|\nabla f_1^{(k)}(\mathbf{x}_1) - \nabla f_1^{(k)}(\mathbf{x}_2)\| \leq \tilde{L}_f \|\mathbf{x}_1 - \mathbf{x}_2\|, \forall \mathbf{x}_1, \mathbf{x}_2 \tag{2.7}$$

$$\|\nabla g_1^{(k)}(\mathbf{y}_1) - \nabla g_1^{(k)}(\mathbf{y}_2)\| \leq \tilde{L}_g \|\mathbf{y}_1 - \mathbf{y}_2\|, \forall \mathbf{y}_1, \mathbf{y}_2 \tag{2.8}$$

where $\tilde{L}_f, \tilde{L}_g$ are uniform upper bounds of Lipschitz constants for $\nabla f_1^{(k)}, \nabla g_1^{(k)}$.

Assumption 4. *For each time k, objective functions $f^{(k)}(\mathbf{x}), g^{(k)}(\mathbf{y})$ are coercive, i.e.*

$$f^{(k)}(\mathbf{x}) \to \infty \text{ as } \|\mathbf{x}\| \to \infty, \; g^{(k)}(\mathbf{y}) \to \infty \text{ as } \|\mathbf{y}\| \to \infty$$

Assumption 4 will be instrumental to ensure that the iterates are bounded; that is, continuous coercive functions' level sets $\{\mathbf{x} | f(\mathbf{x}) \leq \mu_1, \forall \mu_1\}, \{\mathbf{y} | g(\mathbf{y}) \leq \mu_2, \forall \mu_2\}$ are always compact, thus we have optimal solutions are bounded, i.e.

$$\|\mathbf{x}^{*,(k)}\| \leq \sigma_1, \; \|\mathbf{y}^{*,(k)}\| \leq \sigma_2$$

for some positive constants σ_1, σ_2.

Remark. Problem (2.1) is time-varying (or dynamic) because its cost function and constraints evolve over time. In this context, we consider the problem of tracking an optimal solution trajectory. Relative to other existing online optimization or leaning settings, our time-varying scenario involves a changing objective and constraints, not just increasing data or processing measurements in a sequential fashion; furthermore, our scenario does not rely on feedback, so there is no exploration-exploitation tradeoff.

2.3 ADMM for Time-Varying Optimization

This section presents an ADMM-based algorithm to track an optimal solution trajectory of the time-varying problem (P2). As summarized in Table I, the proposed algorithm exhibits linear convergence guarantees under less stringent conditions relative to existing ADMM-based methods (even for static problems). In fact, although classic ADMM is conceptually simple and easy to implement, the conditions under which it is convergent is shown to be quite restrictive [15]. We propose a new algorithm by leveraging the idea of dual perturbation [39,40] and gradient steps; this will provide a way to demonstrate convergence for a larger family of problems. However, a linear convergence rate at milder conditions comes at the cost of ensuring tracking of an approximate Karush-Kuhn-Tucker (KKT) point [3,39,40].

Accordingly, we propose to add a small perturbation to the dual variable $\boldsymbol{\lambda}$ in the form of $1-\beta\gamma$, where $\gamma > 0$ is the perturbation parameter and $\beta\gamma \in (0,1)$. The perturbed augmented Lagrangian function is then defined as

$$\begin{aligned}
&\mathcal{L}^{(k)}(\mathbf{x},\mathbf{y};\boldsymbol{\lambda}) = \mathcal{L}_1^{(k)}(\mathbf{x},\mathbf{y};\boldsymbol{\lambda}) + f_0^{(k)}(\mathbf{x}) + g_0^{(k)}(\mathbf{y}) \\
&\mathcal{L}_1^{(k)}(\mathbf{x},\mathbf{y};\boldsymbol{\lambda}){=}f_1^{(k)}(\mathbf{x}){+}g_1^{(k)}(\mathbf{y}) + \frac{\beta}{2}\|\mathbf{A}^{(k)}\mathbf{x}{+}\mathbf{B}^{(k)}\mathbf{y}{-}\mathbf{b}^{(k)}\|^2 \\
&-(1-\beta\gamma)\boldsymbol{\lambda}^T(\mathbf{A}^{(k)}\mathbf{x}+\mathbf{B}^{(k)}\mathbf{y}{-}\mathbf{b}^{(k)}).
\end{aligned} \tag{2.9}$$

Following standard gradient-based ADMM algorithm [41], to update $\mathbf{x}$ and $\mathbf{y}$, one performs the following optimization

$$\mathbf{y}^{(k+1)}{=}\arg\min_{\mathbf{y}} \left\langle \frac{\partial\mathcal{L}_1^{(k+1)}(\mathbf{x}^{(k)},\mathbf{y}^{(k)};\boldsymbol{\lambda}^{(k)})}{\partial\mathbf{y}^{(k)}}, \mathbf{y}-\mathbf{y}^{(k)} \right\rangle + g_0^{(k+1)}(\mathbf{y}) + \frac{1}{2\alpha_2}\|\mathbf{y}-\mathbf{y}^{(k)}\|^2,$$

$$\mathbf{x}^{(k+1)}{=}\arg\min_{\mathbf{x}} \left\langle \frac{\partial\mathcal{L}_1^{(k+1)}(\mathbf{x}^{(k)},\mathbf{y}^{(k+1)};\boldsymbol{\lambda}^{(k)})}{\partial\mathbf{x}^{(k)}}, \mathbf{x}-\mathbf{x}^{(k)} \right\rangle + f_0^{(k+1)}(\mathbf{x}) + \frac{1}{2\alpha_1}\|\mathbf{x}-\mathbf{x}^{(k)}\|^2,$$

where α_1, α_2 are step sizes. Now we are ready to give a perturbed version of gradient-based ADMM[2]:

$$\mathbf{y}^{(k+1)} = \operatorname*{prox}_{g_0^{(k+1)}}\left(\mathbf{y}^{(k)} - \alpha_2 \frac{\partial \mathcal{L}_1^{(k+1)}(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}; \boldsymbol{\lambda}^{(k)})}{\partial \mathbf{y}^{(k)}}\right), \tag{2.10a}$$

$$\mathbf{x}^{(k+1)} = \operatorname*{prox}_{f_0^{(k+1)}}\left(\mathbf{x}^{(k)} - \alpha_1 \frac{\partial \mathcal{L}_1^{(k+1)}(\mathbf{x}^{(k)}, \mathbf{y}^{(k+1)}; \boldsymbol{\lambda}^{(k)})}{\partial \mathbf{x}^{(k)}}\right), \tag{2.10b}$$

$$\boldsymbol{\lambda}^{(k+1)} = (1-\beta\gamma)\boldsymbol{\lambda}^{(k)} - \beta\Big(\mathbf{A}^{(k+1)}\mathbf{x}^{(k+1)} + \mathbf{B}^{(k+1)}\mathbf{y}^{(k+1)} - \mathbf{b}^{(k+1)}\Big). \tag{2.10c}$$

Compared to classical ADMM-based algorithms, the differences here are in the proximal gradient steps in the primal update and the (small) perturbation added to $\boldsymbol{\lambda}$. The proximal gradient steps may provide favorable computational gains when applied to a large-scale problem; it also facilitate ones to develop measurement-based algorithms as in, e.g., [3].

We remark that adding small perturbation in dual variable is a very useful technique to ensure convergence. To give some intuition of why we design our algorithm this way, let us consider a toy example as follows:

$$\min_{\mathbf{x}} \quad 0, \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = 0 \tag{2.11}$$

where $\mathbf{A}$ is some fixed matrix, not necessarily positive semidefinite. The optimality condition for the above problem can be written down as the following saddle point problem

$$\min_{\mathbf{x}} \max_{\boldsymbol{\lambda}} \ \mathbf{x}^T \mathbf{A} \boldsymbol{\lambda}. \tag{2.12}$$

One can apply the alternating gradient descent/ascent method for solving problem (2.12), whose steps are similar as (2.10) and are given below

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha(\mathbf{A}\boldsymbol{\lambda}^k), \ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \beta(\mathbf{A}^T\mathbf{x}^k).$$

An interesting observation (see Figure 2.2 where we plot $\mathbf{x}^T\mathbf{A}\boldsymbol{\lambda}$ using a random matrix $\mathbf{A}$) is that the algorithm will diverge if no perturbation is added to $\mathbf{y}$ as shown in Figure 2.1a; also see [42] for a formal proof. However, once a small perturbation is added to $\mathbf{y}$ in both primal and dual updates, i.e.

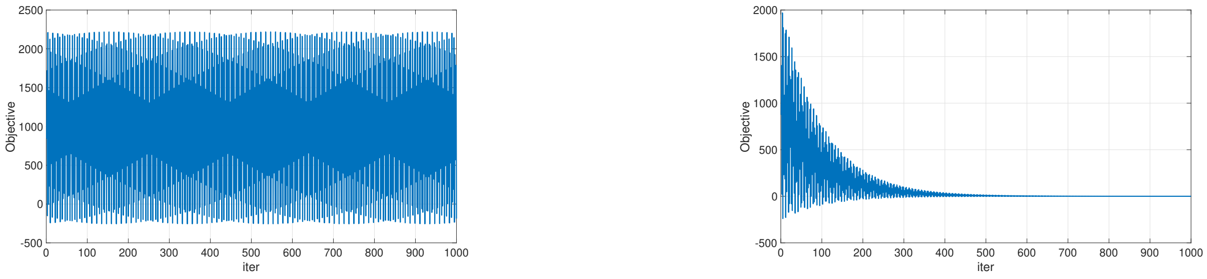
$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha(\mathbf{A}\boldsymbol{\lambda}^k(1-\gamma\beta)), \ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k(1-\gamma\beta) + \beta(\mathbf{A}^T\mathbf{x}^k),$$

where $\gamma > 0$ is a small number, the algorithm will converge as shown in Figure 2.1b. This example serves as an motivation to use the perturbation technique.

[2] proximal operator is defined as $\operatorname{prox}_h(\mathbf{x}) = \arg\min_{\mathbf{z}} \|\mathbf{z} - \mathbf{x}\|^2 + h(\mathbf{z})$.

Table 2.1: Trade-off between optimality and conditions for linear convergence.

	Strong Convexity	**Lipschitz Continuity**	**Full Row Rank**	**Optimality**
Classic ADMM	$f^{(k)}$	$\nabla f^{(k)}$	$\mathbf{A}^{(k)}, (\mathbf{B}^{(k)})^T$	Optimal solution
	$f^{(k)}, g^{(k)}$	$\nabla f^{(k)}$	$\mathbf{A}^{(k)}$	
	$f^{(k)}$	$\nabla f^{(k)}, \nabla g^{(k)}$	$(\mathbf{B}^{(k)})^T$	
	$f^{(k)}, g^{(k)}$	$\nabla f^{(k)}, \nabla g^{(k)}$	✗	
Proposed Algorithm	$f^{(k)}, g^{(k)}$	✗	✗	Perturbed solution [cf. (2.13)]



(a) Performance without perturbation

(b) Performance with perturbation.

Figure 2.2: Example of trends of the objective value of (2.12) for methods with and without perturbation.

2.4 Convergence Analysis

In this section we provide analytical results for convergence and tracking ability of the proposed algorithm. The overall analysis is grounded on the fact that the algorithm would exhibit linear convergence in a static setting; once linear convergence is guaranteed for a static problem, we can prove that our proposed algorithm is able to provide the desired tracking ability.

From [15], it is known that existing ADMM has relatively strict conditions for linear convergence and these conditions may not hold true in some applications (it will not hold for one application presented in Section 4.4); for example, the coefficient matrices in constraints (2.1a) might not have full row rank. Further, in some applications, the objective function of (2.1) may also contain non-smooth terms, which can jeopardize the Lipschitz continuity property. In contrast, the proposed algorithm could be utilized in a wider range of time-varying optimization problems. It is also worth pointing out that [15] deals with static optimization problems; here, the focus is on time-varying settings. Next, we analyze the convergence of the algorithm. To proceed, we concatenate primal and dual optimizer as $\{\mathbf{u}^*\} = \{\mathbf{x}^*; \mathbf{y}^*; \boldsymbol{\lambda}^*\}$ (for static case) as the optimizer of $\max\limits_{\boldsymbol{\lambda}} \min\limits_{\mathbf{x},\mathbf{y}} \mathcal{L}^{(k)}$ at time k. For notation simplicity we neglect superscript k for static case and we have:

$$\mathbf{A}^T\boldsymbol{\lambda}^* - \nabla f_1(\mathbf{x}^*) \in \partial f_0(\mathbf{x}^*) \tag{2.13a}$$

$$\mathbf{B}^T\boldsymbol{\lambda}^* - \nabla g_1(\mathbf{y}^*) \in \partial g_0(\mathbf{y}^*) \tag{2.13b}$$

$$\mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{y}^* - \mathbf{b} + \gamma\boldsymbol{\lambda}^* = 0 \tag{2.13c}$$

Condition (2.13) is a perturbed version of KKT conditions, related to approximate KKT (AKKT) [43, 44]. Basically, optimizer $\mathbf{u}^*$ is not necessarily the KKT point of original problem (P2), but rather an approximate solution within a range of the KKT point of (P2). The detailed proof of why $\mathbf{u}^*$ is an AKKT point of (P2) is beyond the scope of this paper. We refer readers to [44, section 3] for detailed discussion on connections between AKKT and KKT conditions as well as AKKT proofs.

As an intermediate step, we consider the case where our algorithm is applied to a static problem; the result for the static case will then be utilized in the proof of our main result.

Lemma 1. *For a fixed given time, let $\{\mathbf{u}^k\} = \{\mathbf{x}^k; \mathbf{y}^k; \boldsymbol{\lambda}^k\}$ be the sequence generated by our algorithm. Further, let $\mathbf{u}^*$ be an approximate KKT point of (P2), we have the following inequality:*

$$\|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{G}}^2 \geq (1+\delta)\|\mathbf{u}^{k+1} - \mathbf{u}^*\|_{\mathbf{G}}^2, \tag{2.14}$$

where $\mathbf{G} = \begin{pmatrix} \frac{1}{\alpha_1}\mathbf{I} & 0 & 0 \\ 0 & \frac{1}{\alpha_2}\mathbf{I} & 0 \\ 0 & 0 & \frac{1}{\beta}\mathbf{I} \end{pmatrix}$ *is positive definite and:*

$$\delta = \min\left(\frac{\tilde{v}_f}{(1+\beta\gamma)\tilde{\sigma}_{\mathbf{A}}^2 + \frac{\tilde{L}_f^2}{\tilde{v}_f}}, \frac{\tilde{v}_g^2}{4\beta^2\tilde{\sigma}_{\mathbf{B}}^4 + 2\tilde{L}_g^2}, \beta\gamma \right).$$

Dual step size β and perturbation constant γ satisfy: $\beta\gamma + \beta \leq 1, \beta \leq 1$. Finally, we can choose step sizes as follows:

$$\alpha_1 = \frac{1}{(1+\beta\gamma)\tilde{\sigma}_{\mathbf{A}}^2 + \frac{\tilde{L}_f^2}{\tilde{v}_f}}, \ \alpha_2 = \frac{1}{\frac{2\beta^2 \max \tilde{\sigma}_{\mathbf{B}}^4}{\tilde{v}_g} + \frac{\tilde{L}_g^2}{\tilde{v}_g}}$$

Remark. We can further specify $\beta = 0.5$ and $\gamma = 1$ (other choices would also be fine as long as they satisfy conditions in Lemma 1) so that $\alpha_1, \alpha_2, \delta$ depend only on the problem itself; i.e,

$$\delta = \min\left(\frac{\tilde{v}_f}{\frac{3}{2}\tilde{\sigma}_{\mathbf{A}}^2 + \frac{\tilde{L}_f^2}{\tilde{v}_f}}, \frac{\tilde{v}_g^2}{\tilde{\sigma}_{\mathbf{B}}^4 + 2\tilde{L}_g^2}, \frac{1}{2}\right), \tag{2.15}$$

$$\alpha_1 = \frac{1}{\frac{3}{2}\tilde{\sigma}_{\mathbf{A}}^2 + \frac{\tilde{L}_f^2}{\tilde{v}_f}}, \ \alpha_2 = \frac{1}{\frac{\max \tilde{\sigma}_{\mathbf{B}}^4}{2\tilde{v}_g} + \frac{\tilde{L}_g^2}{\tilde{v}_g}}. \tag{2.16}$$

Notice that as long as one picks δ as in (2.15), there exist suitable α_1, α_2 to ensure convergence (see (2.41)–(2.42) in the proof). Inequality (2.14) indicates that the iterates generated by the algorithm are contracting updates. In fact, it can be regarded as linear convergence of $\mathbf{u}^k$ for each fixed time instance (i.e. static case). The proof is relegated to appendix. Now that we are guaranteed (2.14) is true, we can now proceed to state the tracking ability of our proposed algorithm.

Theorem 1. *At each time instance k, suppose that all assumptions hold; we concatenate primal and dual variables as $\{\mathbf{w}^{(k)}\} = \{\mathbf{x}^{(k)}; \mathbf{y}^{(k)}; \boldsymbol{\lambda}^{(k)}\}$ as iterates generated by our algorithm and $\{\mathbf{w}^{*,(k)}\} = \{\mathbf{x}^{*,(k)}, \mathbf{y}^{*,(k)}, \boldsymbol{\lambda}^{*,(k)}\}$ be an optimizer of $\max_{\boldsymbol{\lambda}} \min_{\mathbf{x},\mathbf{y}} \mathcal{L}^{(k)}$. It holds that:*

$$\lim_{k\to\infty} \sup \|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \leq \frac{\psi(\sigma)}{\delta}, \tag{2.17}$$

where $\psi(\sigma) = \sqrt{\frac{\sigma_{\mathbf{x}}^2}{\alpha_1} + \frac{\sigma_{\mathbf{y}}^2}{\alpha_2} + 2\sigma_{\boldsymbol{\lambda}}^2}$ and

$$\sigma_{\boldsymbol{\lambda}} = \tilde{\sigma}_{\mathbf{A}}\sigma_{\mathbf{x}} + \tilde{\sigma}_{\mathbf{B}}\sigma_{\mathbf{y}} + \sigma_{\mathbf{b}} + \sigma_{\mathbf{A}}\sigma_1 + \sigma_{\mathbf{B}}\sigma_2.$$

Proof of Theorem 1. According to linear convergence result for a fixed time we have

$$\|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \leq \frac{1}{1+\delta}\|\mathbf{w}^{(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}},$$

For notation simplicity we define $r = \frac{1}{1+\delta} \in (0, 1)$. Based on this and triangle inequality we have

$$
\begin{aligned}
&\|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \leq r\|\mathbf{w}^{(k-1)} - \mathbf{w}^{*,(k-1)} + \mathbf{w}^{*,(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \\
&\leq r\|\mathbf{w}^{(k-1)} - \mathbf{w}^{*,(k-1)}\|_{\mathbf{G}} + r\|\mathbf{w}^{*,(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \\
&\leq r\left(r\|\mathbf{w}^{(k-2)} - \mathbf{w}^{*,(k-2)}\|_{\mathbf{G}} + r\|\mathbf{w}^{*,(k-2)} - \mathbf{w}^{*,(k-1)}\|_{\mathbf{G}}\right) \\
&+ r\|\mathbf{w}^{*,(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \quad \ldots \\
&\leq r^k\|\mathbf{w}^{(0)} - \mathbf{w}^{*,(0)}\|_{\mathbf{G}} + \sum_{i=1}^{k} r^{k-i+1}\|\mathbf{w}^{*,(i-1)} - \mathbf{w}^{*,(i)}\|_{\mathbf{G}}.
\end{aligned}
$$

Notice that we only have bounded primal drift, so we need to use other terms to bound dual drift. From (2.2), (2.13c) and the fact that we have chosen $\gamma = 1$, we know that

$$
\begin{aligned}
&\mathbf{A}^{(k+1)}\mathbf{x}^{*,(k+1)} - \mathbf{A}^{(k)}\mathbf{x}^{*,(k)} + \mathbf{B}^{(k+1)}\mathbf{y}^{*,(k+1)} - \mathbf{B}^{(k)}\mathbf{y}^{*,(k)} \\
&+ \mathbf{b}^{(k)} - \mathbf{b}^{(k+1)} + \boldsymbol{\lambda}^{*,(k+1)} - \boldsymbol{\lambda}^{*,(k)} = 0 \\
&\Rightarrow \|\boldsymbol{\lambda}^{*,(k+1)} - \boldsymbol{\lambda}^{*,(k)}\| \\
&\leq \|\mathbf{A}^{(k+1)}(\mathbf{x}^{*,(k+1)} - \mathbf{x}^{*,(k)}) + (\mathbf{A}^{(k+1)} - \mathbf{A}^{(k)})\mathbf{x}^{*,(k)}\| \\
&+ \|\mathbf{B}^{(k+1)}(\mathbf{y}^{*,(k+1)} - \mathbf{y}^{*,(k)}) + (\mathbf{B}^{(k+1)} - \mathbf{B}^{(k)})\mathbf{y}^{*,(k)}\| \\
&+ \|\mathbf{b}^{(k)} - \mathbf{b}^{(k+1)}\| \\
&\leq \|\mathbf{A}^{(k+1)}\|\|\mathbf{x}^{*,(k+1)} - \mathbf{x}^{*,(k)}\| + \|\mathbf{B}^{(k+1)}\|\|\mathbf{y}^{*,(k+1)} - \mathbf{y}^{*,(k)}\| \\
&+ \|\mathbf{b}^{(k)} - \mathbf{b}^{(k+1)}\| + \|\mathbf{A}^{(k+1)} - \mathbf{A}^{(k)}\|\|\mathbf{x}^{*,(k)}\| \\
&+ \|\mathbf{B}^{(k+1)} - \mathbf{B}^{(k)}\|\|\mathbf{y}^{*,(k)}\| \\
&\leq \tilde{\sigma}_{\mathbf{A}}\sigma_{\mathbf{x}} + \tilde{\sigma}_{\mathbf{B}}\sigma_{\mathbf{y}} + \sigma_{\mathbf{b}} + \sigma_{\mathbf{A}}\sigma_1 + \sigma_{\mathbf{B}}\sigma_2 \triangleq \sigma_{\boldsymbol{\lambda}}.
\end{aligned}
$$

Now we can make the following conclusion:

$$
\|\mathbf{w}^{*,(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \leq \sqrt{\frac{\sigma_{\mathbf{x}}^2}{\alpha_1} + \frac{\sigma_{\mathbf{y}}^2}{\alpha_2} + 2\sigma_{\boldsymbol{\lambda}}^2} \triangleq \psi(\sigma).
$$

Taking $k \to +\infty$, we can derive

$$\begin{aligned} &\lim_{k\to\infty} \|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \\ &\le \lim_{k\to\infty} \left(\frac{r(1-r^k)}{1-r}\psi(\sigma) + r^k \|\mathbf{w}^{(0)} - \mathbf{w}^{*,(0)}\|_{\mathbf{G}} \right) \\ \Rightarrow &\limsup_{k\to\infty} \|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \le \frac{r}{1-r}\psi(\sigma) = \frac{\psi(\sigma)}{\delta}. \end{aligned}$$

The desired result is obtained. ■

Remark. The final bound in Theorem 1 may not necessarily be tight. However, it illustrates how constant δ, successive differences between parameters and optimal solutions can affect the final tracking bound. This can serve as a guideline for modelling future applications to ensure tracking ability. The results of Theorem 1 can be further extended to network control applications where system measurements are involved in both objective function and constraints. The measurements are used to replace terms originally generated from iterates. It can be shown that as long as the difference between measurements and original terms are within a bound, we are able to generalize Theorem 1 by manipulating the constant on the right hand side of (2.17).

2.5 Example of Motivating Applications

In this section, we briefly outline two examples of application of the proposed methodology and algorithmic framework. In particular:

(i) Power systems: we consider a distribution network featuring distributed energy resources (DERs), and we apply the proposed methodology to drive the DER output powers to the solution of an optimization problem encapsulating voltage constraints and given performance objectives. Differently from e.g. [4, 45], we demonstrate that the proposed methodology is amenable to settings where the distribution system is partitioned in areas; each area is autonomously controlled, and it "trades" power with adjacent areas based on given economic objectives [46].

(ii) Route choice in a road networks: we consider a problem where drivers try to minimize the traveling time, while possibly avoiding substantial detours from their preferred route. The routing problem is probabilistic, in the sense that the problem produces probabilities mass functions that the driver utilizes to pick the routes.

the problem inputs include terms that account for traffic conditions and capacity constraints; see e.g. [47] and pertinent references there in for a detailed model.

2.5.1 Real Time OPF with Multi-area Consensus Constraints

Similar to [46], consider partitioning a power distribution network into C clusters, and denote as $\mathcal{C}_i$ the set of electrical nodes within cluster $i = 1, \dots, C$. Two clusters i and j are adjacent if there is at least an electrical node i such that $i \in \mathcal{C}_i$ and $i \in \mathcal{C}_j$. Let $\mathcal{B}_{i,j} := \mathcal{C}_i \cap \mathcal{C}_j$ be the set of *boundary nodes* connecting cluster i to cluster j, and define $\mathcal{B}_i := \cup_{j \neq i} \mathcal{B}_{i,j}$. Further, let $\mathcal{I}_i := \mathcal{C}_i \backslash \mathcal{B}_i$ be the set of *internal nodes* for cluster i. For future developments, let $N_i := |\mathcal{I}_i|$ be the number of internal nodes if cluster i, and let $\mathcal{N}_i \subset \{1, \dots, C\}$ be the set of neighboring clusters of the ith one (i.e., cluster connected to the ith one).

Let $\mathbf{x}_j^i := [P_j^i, Q_j^i]^\mathsf{T} \in \mathbb{R}^2$ collect the net active and reactive powers injected by DERs at the node $j \in \mathcal{I}_i$ of cluster i. Particularly, $\mathbf{x}_j^i$ can represent the powers injected by *one* DER located at node j, or the aggregate net power injections of *a group of* DERs located at node j (e.g., a household with multiple controllable devices) and we stack the setpoints $\{\mathbf{x}_j^i\}_{j \in \mathcal{I}_i}$ in the vector $\mathbf{x}^i \in \mathbb{R}^{2N_i}$. If no controllable DERs are present at a given location, the corresponding vector $\mathbf{x}_j^i$ is set to $\mathbf{0}$[3]. On the other hand, $\boldsymbol{\ell}_j^i \in \mathbb{R}^2$ denotes the net non-controllable loads at node $j \in \mathcal{I}_i$, and $\boldsymbol{\ell}^i \in \mathbb{R}^{2N_i}$ stacks the loads $\{\boldsymbol{\ell}_j^i\}_{j \in \mathcal{I}_i}$. It is assumed that no DERs and no non-controllable loads are located at the boundary nodes $\mathcal{B}_{i,j}$.

Let $V_j^i \in \mathbb{C}$ denote the complex line-to-ground voltage phasor at node j of cluster i, and let $\mathbf{v}^i := [\{|V_j^i|, j \in \mathcal{I}_i\}]^\mathsf{T}$ be the vector of voltage magnitudes of the internal nodes $\mathcal{I}_i$. For each pair of neighboring clusters (i, j), let $\mathbf{x}_n^{j \to i} := [P_n^{j \to i}, Q_n^{j \to i}]^\mathsf{T} \in \mathbb{R}^2$ represent the active and reactive powers flowing into area i from area j through node $n \in \mathcal{B}_{i,j}$; on the other hand, $\mathbf{x}_n^{i \to j} \in \mathbb{R}^2$ contains the active and reactive powers flowing into area j from area i through node $n \in \mathcal{B}_{i,j}$. From Kirchhoff's Law, it holds that $\mathbf{x}_n^{j \to i} + \mathbf{x}_n^{i \to j} = \mathbf{0}$. To facilitate the syntheses of computationally-affordable algorithms, we leverage the following approximate linear relationship between net injected power and voltage magnitude (see e.g., [48, 49] and references therein):

[3] For notational simplicity, the model is outlined for balanced systems and for the case where one household/building with DERs is located at a node. However, the model can be trivially extended to multiphase networks [48] and for the case where multiple households/buildings with DERs are located at a node (at the cost of increasing the complexity of the notation).

$$\tilde{\mathbf{v}}^i := \sum_{j\in\mathcal{I}_i} \mathbf{A}^i_j(\mathbf{x}^i_j - \boldsymbol{\ell}^i_j) + \sum_{j\in\mathcal{N}_i}\sum_{n\in\mathcal{B}_{i,j}} \mathbf{A}^{j\to i}_n \mathbf{x}^{j\to i}_n + \mathbf{a}, \tag{2.18a}$$

$$= \mathbf{A}^i(\mathbf{x}^i - \boldsymbol{\ell}^i) + \sum_{j\in\mathcal{N}^i} \mathbf{A}^{j\to i}\mathbf{x}^{j\to i} + \mathbf{a}, \tag{2.18b}$$

where $\mathbf{A}^i = [\mathbf{A}^i_j]_{j\in\mathcal{I}_i}, \mathbf{A}^{j\to i} = [\mathbf{A}^{j\to i}_n]_{n\in\mathcal{B}_{i,j}}, \mathbf{a}$ are time-varying problem parameters derived from linearized power flow equation. Another linear relationship between net injected power and power between clusters is captured in the following equation:

$$\mathbf{x}^{j\to i} := \sum_{k\in\mathcal{I}_i} \mathbf{M}^{j\to i}_k(\mathbf{x}^i_k - \boldsymbol{\ell}^i_k) + \mathbf{m}^{j\to i},$$

$$+ \sum_{k\in\mathcal{N}_i\setminus\{j\}}\sum_{n\in\mathcal{B}_{i,k}} \mathbf{M}^{k,j\to i}_n \mathbf{x}^{k\to i}_n \tag{2.19a}$$

$$= \mathbf{M}^{j\to i}(\mathbf{x}^i - \boldsymbol{\ell}^i) + \mathbf{m}^{j\to i} + \sum_{k\in\mathcal{N}_i\setminus\{j\}} \mathbf{M}^{k,j\to i}\mathbf{x}^{j\to i}, \tag{2.19b}$$

where $\mathbf{M}^{j\to i} = [\mathbf{M}^{j\to i}_k]_{k\in\mathcal{I}_i}, \mathbf{M}^{k,j\to i} = [\mathbf{M}^{k,j\to i}_n]_{n\in\mathcal{B}_{i,k}},{}^{j\to i}$ are also time-varying problem parameters depending on the actual network physics. All model parameters in (2.18)–(2.19) can be obtained as shown in [48]. Now we are ready to state our real-time OPF problem as follows:

$$\min_{\{\mathbf{x}^i\},\{\mathbf{x}^{j\to i}_n,\mathbf{x}^{i\to j}_n\}} \sum_{i=1}^{C}[f^i(\mathbf{x}^i) + g^i(\{\mathbf{x}^{j\to i}\})] \tag{P3}$$

$$\text{s.t. } \mathbf{x}^i_j \in \mathcal{Y}^i_j, \forall\, j \in \mathcal{I}^i,\ i = 1,\dots C \tag{2.20a}$$

$$v^{\min}\mathbf{1} \le \tilde{\mathbf{v}}^i \le v^{\max}\mathbf{1}, \forall\, i = 1,\dots C \tag{2.20b}$$

$$\mathbf{x}^{j\to i} = \mathbf{M}^{j\to i}(\mathbf{x}^i - \boldsymbol{\ell}^i) + \mathbf{m}^{j\to i} + \sum_{k\in\mathcal{N}_i\setminus\{j\}} \mathbf{M}^{k,j\to i}\mathbf{x}^{j\to i}$$

$$, \forall\, j \in \mathcal{N}_i,\ i = 1,\dots,C \tag{2.20c}$$

$$\mathbf{x}^{j\to i} + \mathbf{x}^{i\to j} = \mathbf{0},\ \forall \text{ neighboring areas } (i,j) \tag{2.20d}$$

where the time-varying objective function models the amount of real power curtailed and the amount of reactive power injected or absorbed (which leads to non-smooth term in the objective, e.g. ℓ_1 term). For notation simplicity, we write objective function in (P3) as $\Psi(\mathbf{x})$. Consider $\mathbf{M}^{j\to i}$ consists of $1, 0$, with 1 for real power, 0 for reactive power. Putting (2.19b) back to (2.18b), adding slack variables $\boldsymbol{\gamma}^i, \boldsymbol{\beta}^i$ to (2.20b) formulate equality constraints, and adding strongly convex term w.r.t $\boldsymbol{\gamma} = \{\boldsymbol{\gamma}^i\}, \boldsymbol{\beta} = \{\boldsymbol{\beta}^i\}$ we have the following formulation:

$$\min_{\{\{\mathbf{x}^i\},\{\mathbf{x}_n^{j\to i},\mathbf{x}_n^{i\to j}\}\},\{\boldsymbol{\gamma}^i,\boldsymbol{\beta}^i\geq 0\}} \quad \Psi(\mathbf{x}) + a\|\boldsymbol{\gamma}\|^2 + b\|\boldsymbol{\beta}\|^2 \tag{P4}$$

$$\text{s.t. } \mathbf{x}_j^i \in \mathcal{Y}_j^i, \forall\ j \in \mathcal{I}_i,\ \ i = 1, \dots C \tag{2.21a}$$

$$v^{\text{min}}\mathbf{1} - \tilde{\mathbf{v}}^i + \boldsymbol{\gamma}^i = \mathbf{0}, \forall\ i = 1, \dots C \tag{2.21b}$$

$$\tilde{\mathbf{v}}^i + \boldsymbol{\beta}^i - v^{\text{max}}\mathbf{1} = \mathbf{0}, \forall\ i = 1, \dots C \tag{2.21c}$$

$$\mathbf{x}^{j\to i} = \mathbf{M}^{j\to i}(\mathbf{x}^i - \boldsymbol{\ell}^i) + \mathbf{m}^{j\to i} + \sum_{k\in\mathcal{N}_i\setminus\{j\}} \mathbf{M}^{k,j\to i}\mathbf{x}^{j\to i}$$

$$, \forall\, j \in \mathcal{N}_i,\ \ i = 1, \dots, C \tag{2.21d}$$

$$\mathbf{x}^{j\to i} + \mathbf{x}^{i\to j} = \mathbf{0},\ \forall \text{ neighboring areas } (i, j). \tag{2.21e}$$

We can now clearly see a mapping from (P4) to (P2): objective functions are $\Psi(\mathbf{x})$ and $a\|\boldsymbol{\gamma}\|^2+b\|\boldsymbol{\beta}\|^2$(where $a, b > 0$ are small); two blocks of variables are $\{\mathbf{x}^i, \mathbf{x}^{j\to i}\}$ and $\{\boldsymbol{\gamma}^i, \boldsymbol{\beta}^i\}$; constraints are all linear and separable w.r.t each network node. Problem (P4) is time-varying in both objective function and constraint parameters. In order to better illustrate how the proposed algorithm can be applied, we use a 4-cluster network (see Figure 2.3) as an example. First, we substitute $\mathbf{x}^{j\to i}$ in (2.18b) with (2.21d); then, we substitute $\tilde{\mathbf{v}}^i$ in (2.21b)(2.21c) with (2.18b); last, we define the corresponding augmented Lagrangian function as follows:

$$\begin{aligned}
&\mathcal{L}(\mathbf{x}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{\lambda}) = \Psi(\mathbf{x}) + a\|\boldsymbol{\gamma}\|^2 + b\|\boldsymbol{\beta}\|^2 \\
&+\sum_k\sum_{i\in C_k}\frac{\rho}{2}\|\sum_{j\in C_k}(\mathbf{A}_j^i + \mathbf{A}_j^3)\mathbf{x}_j^i + \mathbf{a}_k + \boldsymbol{\beta}^i - v^{\text{max}} + \frac{\boldsymbol{\lambda}_1(1-\gamma)}{\rho}\|^2 \\
&+\sum_k\sum_{i\in C_k}\frac{\rho}{2}\|v^{\text{min}} - \sum_{j\in C_k}(\mathbf{A}_j^i + \mathbf{A}_j^3)\mathbf{x}_j^i - \mathbf{a}_k + \boldsymbol{\gamma}^i + \frac{\boldsymbol{\lambda}_2(1-\gamma)}{\rho}\|^2 \\
&+\sum_k\frac{\rho}{2}\|\sum_i \mathbf{x}^{i\to j} - \sum_{j\in C_k}\mathbf{x}^j + \frac{\boldsymbol{\lambda}_4(1-\gamma)}{\rho}\|^2 \\
&+\frac{\rho}{2}\|\sum_{i\neq j}\mathbf{x}^{i\to j} + \frac{\boldsymbol{\lambda}_3(1-\gamma)}{\rho}\|^2
\end{aligned}$$

The detailed updates follow the same way as (2.10) and from Table I we know that linear convergence to AKKT is guaranteed. To further improve our algorithm for this particular application. We incorporate system measurements in both primal and dual updates in the following way:

$$\sum_{j \in C_k} (\mathbf{A}_j^i + \mathbf{A}_j^3)\mathbf{x}_j^i + \mathbf{a}_k \to \phi(\mathbf{x}), \ \sum_{j \in C_k} \mathbf{x}^j \to \psi(\mathbf{x}), \tag{2.22}$$

where $\phi(\mathbf{x}), \psi(\mathbf{x})$ are measurements. This is beneficial in that: i) A natural distributed computing scheme is achieved while without feedback it is not clear whether the algorithm can be implemented in a distributed way; ii) feedback terms are much less than uncontrollable terms, which essentially shrinks the measuring time; iii) it is easier to satisfy power flow equations with the help of system measurements.

2.5.2 Route Choice in Road Network

This application is about each driver minimizes his/her own traveling time and at the same time not deviate from his/her preferred route. Consider a strongly-connected directed graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, V\}$ represents locations, $\mathcal{E} = \{1, \ldots, E\}$ represents roads. Each driver $i \in \{1, \ldots, N\}$ has an origin O_i and a destination D_i, between which the driver has his/her strategy of which road to transit. We define the strategy as a vector of probabilities $\mathbf{s}_i = [s_i^1; \ldots; s_i^E], s_i^j \in [0, 1]$ for each driver i. In order for all driver to be able to reach destination from origin, the following condition has to hold:

$$\sum_{j \in \to v} s_i^j - \sum_{j \in \leftarrow v} s_i^j = \begin{cases} -1, & \text{if } v = O_i \\ 1, & \text{if } v = D_i \\ 0, & \text{otherwise,} \end{cases} \tag{2.23}$$

where $\to v$ and $\leftarrow v$ represents edges flow into v and out of v, respectively. We define the incidence matrix $\mathbf{M} \in \mathbb{R}^{V \times E}$, where component $\mathbf{M}_{i,j} = 1$ if road j points to i, $\mathbf{M}_{i,j} = -1$ if road j points out of i, $\mathbf{M}_{i,j} = 0$ otherwise. In this way, we can construct a constraint for driver i as $\mathbf{M}\mathbf{s}_i =_{i,i} \in \mathbb{R}^V$. The elements of $_i$ is defined as $^j_i = -1$ when $j = O_i$, $^j_i = 1$ when $j = D_i$, $^j_i = 0$ otherwise. Also we need to acknowledge the fact that there should be a certain limit for number of cars in one road. We model this fact using the following constraint:

$$\frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_i^j \le L^j, \ \forall j = 1, \ldots, E \tag{2.24}$$

where L^j denotes the limit for road j. Recall that we want to minimize driver's traveling time and stick to driver's preferred route. To this end, we introduce the following objective:

$$C(\mathbf{s}_i) = \frac{r}{2}\|\mathbf{s}_i - \hat{\mathbf{s}}_i\|^2 + \sum_{j=1}^{E} T^j(\frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_i^j)\mathbf{s}_i^j, \tag{2.25}$$

where $T^j(\frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_i^j)$ models the traveling time for road j, which is a smoothed version of piecewise linear function and is continuously differentiable; $\hat{\mathbf{s}}_i$ denotes the preferred strategy of driver i. Now adding suitable slack variables z^j to (2.24) and adding a small strongly convex term in the objective we can formulate the following problem:

$$\min_{\mathbf{s}_i,\mathbf{z}:=\{z^j\}\geq 0} \frac{r}{2}\|\mathbf{s}_i - \hat{\mathbf{s}}_i\|^2 + \sum_{j=1}^{E} T^j(\frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_i^j)\mathbf{s}_i^j + c\|\mathbf{z}\|^2 \tag{P5}$$

$$\text{s.t. } \frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_i^j + z^j = L^j,\ j = 1,\dots,E \tag{2.26a}$$

$$\mathbf{M}\mathbf{s}_i =_i,\ i = 1,\dots,N \tag{2.26b}$$

(P5) is a time-varying problem in that each roads' conditions are changing over time, i.e. L^j is a time-varying term, which in turn makes the traveling time function T^j also time-varying; it is also possible that a driver will change his/her location during the trip, so $_i$ is another time-varying term. We can clearly see that (P5) is a special case of our general model (P2) in the following sense: i) Objective functions are strongly convex w.r.t $\mathbf{s}_i, \mathbf{z}$; ii) Constraints (2.26a) and (2.26b) are all linear constraints that can be grouped as the form $\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} = \mathbf{b}$. Therefore, our algorithm can be applied to (P5) with guaranteed linear convergence for each time instance.

2.6 Simulation

In this section, we test our algorithm using the same power systems settings. We consider a similar system as in [4], where a modified IEEE 37-node test feeder is utilized. The network is obtained by considering a single phase equivalent, and by replacing the loads on phase "c" specified in the original dataset with real load data measured from feeders in a neighborhood called Anatolia in California during a week in August 2012. It is assumed that the aggregations of photovoltaic systems are located at nodes 4, 7, 10, 13, 17, 20, 22, 23, 26, 28, 29, 30, 31, 32, 33, 34, 35, and 36. The rating of these inverters are 300kVA for $i = 3$, 350kVA for $i = 15, 16$ and 200kVA for the remaining ones. The objective is set to be

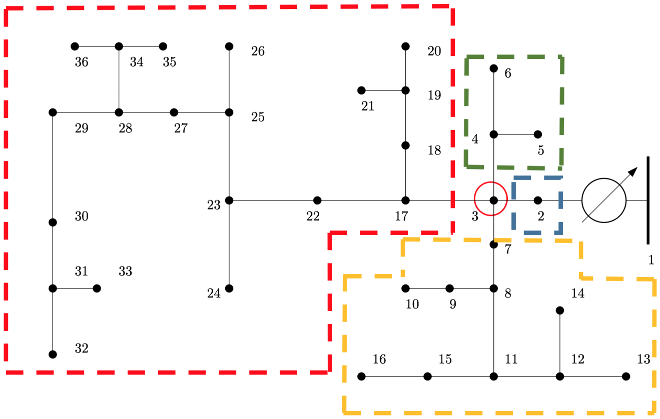


Figure 2.3: power distribution network with 4 clusters, node 3 is a boundary node that belongs to all 4 clusters.

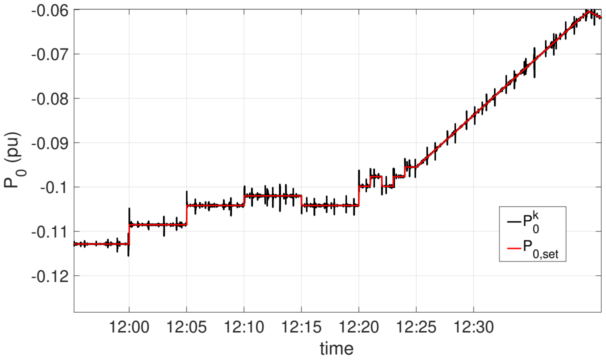


Figure 2.4: Real power at feeder head during 12:00-12:30.

$f^i(\mathbf{x}^i) = c_p(P_{\text{av},i} - P_i)^2 + c_q(Q_i)^2 + \bar{c}_q|Q_i|, g^i(\mathbf{x}^{j\to i}) = 0$ where $P_{\text{av},i}$ is the maximum real power available from the PV system i, and $c_p = 3, c_q = 1, \bar{c}_q = 0.1$. The voltage limits are set to be $V^{\min} = 0.95$pu, $V^{\max} = 1.05$pu. The generation profiles are simulated based on real solar irradiance data and have a granularity of 1 second. First we specify a given trajectory for the power at the common coupling, which is color-coded in red in Fig. 2.4 (negative power indicates reverse power flows). It can be seen that our algorithm is able to regulate P_0^k close to $P_{0,\text{set}}^k$ in real time. Figure 2.5 illustrates the voltage profiles for selected nodes. From 10:00 to 12:00 we observe a few flickers, which is caused by rapid variations of the solar irradiance. Other than that, it can be seen that voltage regulation is enforced and a flat voltage profile is obtained. Note that even there are some big jumps from around 12:00 to 14:00, our algorithm is still able

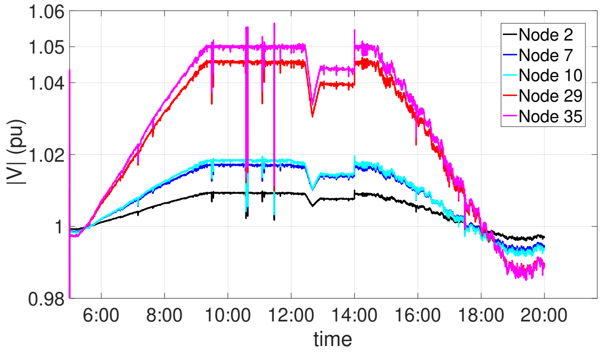


Figure 2.5: Voltage profile achieved (only some nodes are considered for illustration purposes).

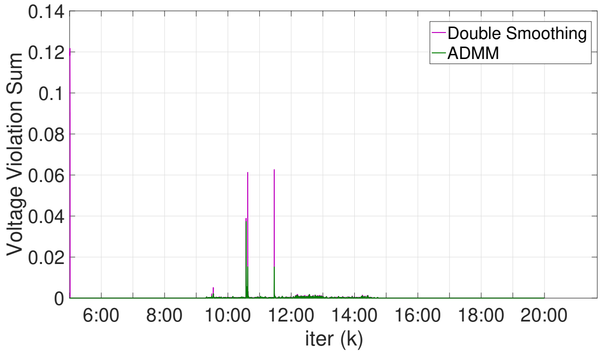
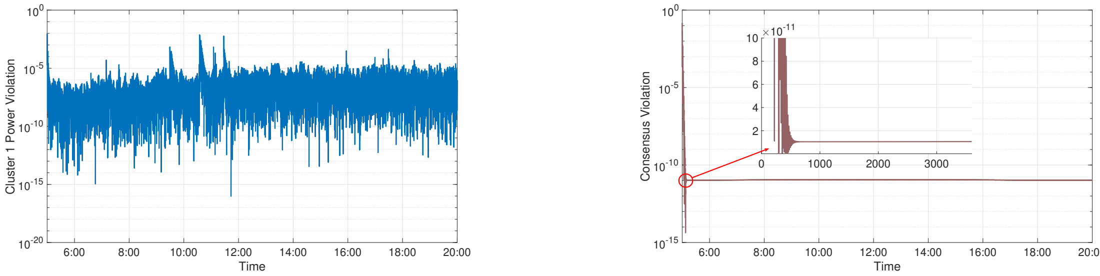


Figure 2.6: Voltage violation across system $\sum\limits_{n\in\mathcal{N}} \big(\max(|V_n^k| - v^{\max}, 0) + \max(v^{\min} - |V_n^k|, 0)\big)$

to track the optimal trajectory. A comparison with double smoothing algorithm [4] is presented in Figure 2.6. The proposed strategy has potentially better voltage regulation ability, especially for extreme cases e.g. the two spikes from 10:00 to 12:00.   We proceed to test in the same setting except we are adding consensus constraints shown in Application 1. In Figure 2.7a we can see that for all 4 clusters, power violation decreases dramatically in first a few minutes and remains at a low level of 10^{-10}. The power consensus violation is shown in Figure 2.7b, where a steep drop at the begging and flat low line after that are observed.

2.7 Conclusion

This paper gives a general online optimization problem formulation and proposes a dynamic algorithm based on alternating direction method of multipliers that can continuously track optimal solution in real



(a) Power violation of Cluster 1: power violation for each cluster is defined as $\| \sum_{i\in\mathcal{N}(j)} \mathbf{x}^{i\to j} - \sum_{j\in C} \mathbf{x}^{j} \|^2$

(b) Consensus violation: $\|\mathbf{x}^{(i\to j)} + \mathbf{x}^{(j\to i)}\|^2$

Figure 2.8: Simulation for multi-area problem

time. The steps of ADMM are proximal gradient steps with modification of adding perturbation to dual variable and incorporating system feedback for certain applications. The resulting algorithm is proved to converge to a neighborhood of optimal solution for each time instance. Numerical results for power systems applications also demonstrate the practicality of the proposed algorithm. Our future research will focus on general online nonconvex optimization problems.

2.8 Proof of Lemma 1

Proof. From the optimality condition of the subproblems (2.10a),(2.10b), one has that:

$$\begin{aligned}&\mathbf{A}^T\boldsymbol{\lambda}^k(1-\beta\gamma)-\beta\mathbf{A}^T(\mathbf{A}\mathbf{x}^k+\mathbf{B}\mathbf{y}^{k+1}-\mathbf{b})-\nabla f_1(\mathbf{x}^k)+\frac{1}{\alpha_1}\mathbf{I}(\mathbf{x}^k-\mathbf{x}^{k+1})\in\partial f_0(\mathbf{x}^{k+1})\\ \Rightarrow&\mathbf{A}^T\boldsymbol{\lambda}^{k+1}+\beta\mathbf{A}^T\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)+\nabla f_1(\mathbf{x}^{k+1})-\nabla f_1(\mathbf{x}^k)+\frac{1}{\alpha_1}\mathbf{I}(\mathbf{x}^k-\mathbf{x}^{k+1})\in\partial f_0(\mathbf{x}^{k+1})+\nabla f_1(\mathbf{x}^{k+1}),\end{aligned}\tag{2.27}$$

$$\begin{aligned}&\mathbf{B}^T\boldsymbol{\lambda}^k(1-\beta\gamma)-\beta\mathbf{B}^T(\mathbf{A}\mathbf{x}^{k+1}+\mathbf{B}\mathbf{y}^{k+1}-\mathbf{b})+\beta\mathbf{B}^T\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)+\beta\mathbf{B}^T\mathbf{B}(\mathbf{y}^{k+1}-\mathbf{y}^k)\\ &+\frac{1}{\alpha_2}\mathbf{I}(\mathbf{y}^k-\mathbf{y}^{k+1})-\nabla g_1(\mathbf{y}^k)\in\partial g_0(\mathbf{y}^{k+1})\\ \Rightarrow&\mathbf{B}^T(\boldsymbol{\lambda}^{k+1}+\beta\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)+\beta\mathbf{B}(\mathbf{y}^{k+1}-\mathbf{y}^k))+\nabla g_1(\mathbf{y}^{k+1})-\nabla g_1(\mathbf{y}^k)+\frac{1}{\alpha_2}\mathbf{I}(\mathbf{y}^k-\mathbf{y}^{k+1})\\ &\in\partial g_0(\mathbf{y}^{k+1})+\nabla g_1(\mathbf{y}^{k+1}).\end{aligned}\tag{2.28}$$

Furthermore, from the dual update, one can obtain:

$$\begin{aligned}&\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k - \beta(\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{b} + \gamma\boldsymbol{\lambda}^k)\\ \Rightarrow &\frac{1}{\beta}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1}) = \gamma\boldsymbol{\lambda}^k + (\mathbf{A}\mathbf{x}^{k+1} + \mathbf{B}\mathbf{y}^{k+1} - \mathbf{b}),\end{aligned} \tag{2.29}$$

and, together with optimality condition, one obtains:

$$\frac{1}{\beta}(\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^{k+1})=\gamma(\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^*)+\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^*)+\mathbf{B}(\mathbf{y}^{k+1}-\mathbf{y}^*) \tag{2.30}$$

Since the functions $f = f_0 + f_1$ and $g = g_0 + g_1$ are strongly convex, we leverage (2.5) and (2.6) and, by plugging the optimality condition (2.27) and (2.28), we have

$$\begin{aligned}&\langle \mathbf{A}^T(\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*) + \beta\mathbf{A}^T\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)+\nabla f_1(\mathbf{x}^{k+1})-\nabla f_1(\mathbf{x}^k)\\ &+\frac{1}{\alpha_1}\mathbf{I}(\mathbf{x}^k - \mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle \geq v_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2\end{aligned} \tag{2.31}$$

$$\begin{aligned}&\langle \mathbf{B}^T(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^* + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k) + \beta\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k))\\ &+ \nabla g_1(\mathbf{y}^{k+1}) - \nabla g_1(\mathbf{y}^k) + \frac{1}{\alpha_2}\mathbf{I}(\mathbf{y}^k - \mathbf{y}^{k+1}), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle\\ &\geq v_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2\end{aligned} \tag{2.32}$$

Next, combine (2.32) and (2.30) to obtain:

$$\begin{aligned}&\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*, \mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^*)\rangle+\langle\beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^*)\rangle\\ &+\langle\nabla f_1(\mathbf{x}^{k+1})-\nabla f_1(\mathbf{x}^k), \mathbf{x}^{k+1}-\mathbf{x}^*\rangle+\langle\frac{1}{\alpha_1}(\mathbf{x}^k-\mathbf{x}^{k+1}), \mathbf{x}^{k+1}-\mathbf{x}^*\rangle\\ &+\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*, \mathbf{B}(\mathbf{y}^{k+1}-\mathbf{y}^*)\rangle+\langle\beta\mathbf{B}^T\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k), \mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\ &+\langle\nabla g_1(\mathbf{y}^{k+1}) - \nabla g_1(\mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle\\ &+\langle\beta\mathbf{B}^T\mathbf{B}(\mathbf{y}^{k+1}-\mathbf{y}^k), \mathbf{y}^{k+1}-\mathbf{y}^*\rangle+\langle\frac{1}{\alpha_2}(\mathbf{y}^k-\mathbf{y}^{k+1}), \mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\ &\geq v_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + v_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 .\end{aligned} \tag{2.33}$$

Define $\Phi = v_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + v_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2$; then, using (2.30) it follows that:

$$
\begin{aligned}
&\langle \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*, \frac{1}{\beta}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1})) - \gamma(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*)\rangle + \beta\langle \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \frac{1}{\beta}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1})) - \gamma(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*)\rangle \\
&+\langle \nabla f_1(\mathbf{x}^{k+1}) - \nabla f_1(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle + \langle \frac{1}{\alpha_1}(\mathbf{x}^k - \mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle + \langle \beta\mathbf{B}^T\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&+\langle \nabla g_1(\mathbf{y}^{k+1}) - \nabla g_1(\mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + \langle \frac{1}{\alpha_2}(\mathbf{y}^k - \mathbf{y}^{k+1}), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \geq \Phi \\
&\Rightarrow \langle \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*, \frac{1}{\beta}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1})\rangle + \langle \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*, \gamma(\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k)\rangle + \beta\langle \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \frac{1}{\beta}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1})\rangle \\
&+\beta\langle \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \gamma(\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k)\rangle + \langle \nabla f_1(\mathbf{x}^{k+1}) - \nabla f_1(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle \\
&+\langle \frac{1}{\alpha_1}(\mathbf{x}^k - \mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle + \langle \beta\mathbf{B}^T\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&+\langle \nabla g_1(\mathbf{y}^{k+1}) - \nabla g_1(\mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + \langle \frac{1}{\alpha_2}(\mathbf{y}^k - \mathbf{y}^{k+1}), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \geq \Phi
\end{aligned}
\tag{2.34}
$$

Define the following quantities:

$$
\mathbf{G} = \begin{pmatrix} \frac{1}{\alpha_1}\mathbf{I} & 0 & 0 \\ 0 & \frac{1}{\alpha_2}\mathbf{I} & 0 \\ 0 & 0 & \frac{1}{\beta}\mathbf{I} \end{pmatrix}, \mathbf{u}^k = \begin{pmatrix} \mathbf{x}^k \\ \mathbf{y}^k \\ \boldsymbol{\lambda}^k \end{pmatrix}, \ \mathbf{u}^* = \begin{pmatrix} \mathbf{x}^* \\ \mathbf{y}^* \\ \boldsymbol{\lambda}^* \end{pmatrix}
$$

so that one can rewrite the inequality above as follows:

$$
\begin{aligned}
&(\mathbf{u}^{k+1} - \mathbf{u}^*)^T\mathbf{G}(\mathbf{u}^k - \mathbf{u}^{k+1}) + \gamma\langle \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k, \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\rangle \\
&+\langle \boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1}, \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)\rangle + \beta\gamma\langle \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^k, \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)\rangle \\
&+\langle \beta\mathbf{B}^T\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&+\langle \mathbf{x}^{k+1} - \mathbf{x}^*, \nabla f_1(x^{k+1}) - \nabla f_1(\mathbf{x}^k)\rangle \\
&+\langle \mathbf{y}^{k+1} - \mathbf{y}^*, \nabla g_1(y^{k+1}) - \nabla g_1(\mathbf{y}^k)\rangle \geq \Phi\,.
\end{aligned}
\tag{2.35}
$$

We then consider the following equality

$$
\|\mathbf{a}-\mathbf{c}\|_{\mathbf{G}}^2 - \|\mathbf{b}-\mathbf{c}\|_{\mathbf{G}}^2 = 2(\mathbf{a}-\mathbf{c})^T\mathbf{G}(\mathbf{a}-\mathbf{b}) - \|\mathbf{a}-\mathbf{b}\|_{\mathbf{G}}^2. \tag{2.36}
$$

and use it in (2.35) to arrive at the following inequality:

$$\begin{aligned}&(\mathbf{u}^{k+1}-\mathbf{u}^*)^T\mathbf{G}(\mathbf{u}^k-\mathbf{u}^{k+1}) \geq \frac{\gamma}{2}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*\|^2\\
&-\frac{\gamma}{2}\|\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^{k+1}\|^2+\frac{\gamma}{2}\|\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^*\|^2+\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k,\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\rangle\\
&+\beta\gamma\langle\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^*,\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\rangle+\langle\mathbf{x}^*-\mathbf{x}^{k+1},\nabla f_1(x^{k+1})-\nabla f_1(\mathbf{x}^k)\rangle\\
&+\langle\mathbf{y}^*-\mathbf{y}^{k+1},\nabla g_1(y^{k+1})-\nabla g_1(\mathbf{y}^k)\rangle\\
&+\langle\beta\mathbf{B}^T\mathbf{B}(\mathbf{y}^{k+1}-\mathbf{y}^k),\mathbf{y}^{k+1}-\mathbf{y}^*\rangle+\Phi\end{aligned}\tag{2.37}$$

Then, we utilize the Cauchy-Schwarz inequality to bound the following terms:

$$\begin{aligned}&\langle\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k),\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\rangle\\
\geq&-\frac{1}{2\rho_1}\|\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\|^2-\frac{\rho_1}{2}\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2,\forall\rho_1>0\\
&\beta\gamma\langle\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k),\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^*\rangle\\
\geq&-\frac{\beta\gamma}{2\rho_2}\|\mathbf{A}(\mathbf{x}^{k+1}-\mathbf{x}^k)\|^2-\frac{\beta\gamma\rho_2}{2}\|\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^*\|^2,\forall\rho_2>0\\
&\langle\beta\mathbf{B}^T\mathbf{B}(\mathbf{y}^k-\mathbf{y}^{k+1}),\mathbf{y}^{k+1}-\mathbf{y}^*\rangle\\
\geq&-\frac{\beta}{\rho_3}\|\mathbf{B}(\mathbf{y}^{k+1}-\mathbf{y}^k)\|^2-\beta\rho_3\|\mathbf{B}(\mathbf{y}^{k+1}-\mathbf{y}^*)\|^2\\
\geq&-\frac{\beta\max\sigma^2(\mathbf{B})}{\rho_3}\|\mathbf{y}^{k+1}-\mathbf{y}^k\|^2-\beta\rho_3\max\sigma^2(\mathbf{B})\|\mathbf{y}^{k+1}-\mathbf{y}^*\|^2\,.\end{aligned}$$

The remaining terms in (2.37) are bounded as follows:

$$\begin{aligned}&\langle\mathbf{x}^*-\mathbf{x}^{k+1},\nabla f_1(x^{k+1})-\nabla f_1(\mathbf{x}^k)\rangle\\
&+\langle\mathbf{y}^*-\mathbf{y}^{k+1},\nabla g_1(y^{k+1})-\nabla g_1(\mathbf{y}^k)\rangle\\
\geq&-\frac{L_f^2}{2\rho_4}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2-\frac{\rho_4}{2}\|\mathbf{x}^{k+1}-\mathbf{x}^*\|^2\\
&-\frac{L_g^2}{2\rho_5}\|\mathbf{y}^{k+1}-\mathbf{y}^k\|^2-\frac{\rho_5}{2}\|\mathbf{y}^{k+1}-\mathbf{y}^*\|^2\end{aligned}\tag{2.38}$$

where we have used the Cauchy-Schwarz inequality and we leveraged the Lipschitz continuity of f_1, g_1. Also, from (2.36), it can be noticed that:

$$\begin{aligned}
&\|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{G}}^2 - \|\mathbf{u}^{k+1} - \mathbf{u}^*\|_{\mathbf{G}}^2 \\
=\ & 2(\mathbf{u}^k - \mathbf{u}^*)^T G(\mathbf{u}^k - \mathbf{u}^{k+1}) - \|\mathbf{u}^k - \mathbf{u}^{k+1}\|_{\mathbf{G}}^2 .
\end{aligned}$$

It therefore follows that:

$$\begin{aligned}
&\|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{G}}^2 - \|\mathbf{u}^{k+1} - \mathbf{u}^*\|_{\mathbf{G}}^2 \\
&\geq \|\mathbf{u}^k - \mathbf{u}^{k+1}\|_{\mathbf{G}}^2 + \gamma\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 \\
&-\gamma\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1}\|^2 + \gamma\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2 \\
&-\frac{\max\sigma^2(\mathbf{A})}{\rho_1}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 - \rho_1\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 \\
&-\frac{\beta\gamma\max\sigma^2(\mathbf{A})}{\rho_2}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 - \beta\gamma\rho_2\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2 \\
&-\frac{L_f^2}{\rho_4}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 - \rho_4\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \\
&-\frac{L_g^2}{\rho_5}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 - \rho_5\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 + \Phi \\
&\frac{\beta\max\sigma^2(\mathbf{B})}{\rho_3}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 - \beta\rho_3\max\sigma^2(\mathbf{B})\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2
\end{aligned}$$

and, rearranging the terms in a suitable way, we arrive at the following inequality:

$$\begin{aligned}
&\|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{G}}^2 - \|\mathbf{u}^{k+1} - \mathbf{u}^*\|_{\mathbf{G}}^2 \\
&\geq (\frac{1}{\alpha_1} - \frac{\max\sigma^2(\mathbf{A})}{\rho_1} - \frac{\beta\gamma\max\sigma^2(\mathbf{A})}{\rho_2} - \frac{L_f^2}{\rho_4})\|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2 \\
&+(\frac{1}{\alpha_2} - \beta\frac{\max\sigma^2(\mathbf{B})}{\rho_3} - \frac{L_g^2}{\rho_5})\|\mathbf{y}^k - \mathbf{y}^{k+1}\|^2 \\
&+(\frac{1}{\beta} - \gamma - \rho_1)\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1}\|^2 + (2v_f - \rho_4)\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \\
&+(2v_g - \rho_5 - \beta\rho_3\max\sigma^2(\mathbf{B}))\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \\
&+\gamma\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 + (\gamma - \beta\gamma\rho_2)\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2 \qquad (2.39)
\end{aligned}$$

Recall that the goal is to prove the following inequality:

$$\|\mathbf{u}^k - \mathbf{u}^*\|_{\mathbf{G}}^2 \geq (1+\delta)\|\mathbf{u}^{k+1} - \mathbf{u}^*\|_{\mathbf{G}}^2, \qquad (2.40)$$

where $\delta > 0$ is a constant. Fro brevity, denote the right-hand-side of (2.39) as C; then it is sufficient to prove that:

$$\begin{aligned} &C \geq \delta\|\mathbf{u}^{k+1} - \mathbf{u}^*\|^2_{\mathbf{G}} \\ = &\frac{\delta}{\alpha_1}\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + \frac{\delta}{\alpha_2}\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 + \frac{\delta}{\beta}\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 \end{aligned}$$

which requires the following to hold true:

$$\begin{aligned} &\frac{1}{\alpha_1} - \frac{\max\sigma^2(\mathbf{A})}{\rho_1} - \frac{\beta\gamma\max\sigma^2(\mathbf{A})}{\rho_2} - \frac{L_f^2}{\rho_4} \geq 0 \\ &\frac{1}{\alpha_2} - \frac{\beta\max\sigma^2(\mathbf{B})}{\rho_3} - \frac{L_g^2}{\rho_5} \geq 0, \ \frac{1}{\beta} - \gamma - \rho_1 \geq 0 \\ &2v_f - \rho_4 - \frac{\delta}{\alpha_1} \geq 0, \ 2v_g - \rho_5 - \beta\rho_3\max\sigma^2(\mathbf{B}) - \frac{\delta}{\alpha_2} \geq 0 \\ &\gamma - \frac{\delta}{\beta} \geq 0, \ \gamma - \beta\gamma\rho_2 \geq 0. \end{aligned}$$

From the inequalities above, one can notice that the constant α_1, α_2 is closely related to various constants as well as the singular values of $\mathbf{A}$ and $\mathbf{B}$, denoted as $\sigma(\mathbf{A})$ and $\sigma(\mathbf{B})$, respectively. Specifically, this leads to the following conditions for the step sizes:

$$\frac{\delta}{2v_f - \rho_4} \leq \alpha_1 \leq \frac{1}{(\frac{1}{\rho_1} + \frac{\beta\gamma}{\rho_2})\max\sigma^2(\mathbf{A}) + \frac{L_f^2}{\rho_4}}, \tag{2.41}$$

$$\frac{\delta}{2v_g - \rho_5 - \beta\rho_3\max\sigma^2(\mathbf{B})} \leq \alpha_2 \leq \frac{1}{\frac{\beta\max\sigma^2(\mathbf{B})}{\rho_3} + \frac{L_g^2}{\rho_5}}. \tag{2.42}$$

To ensure that there exists step sizes α_1, α_2 that satisfy the condition above, one can choose $\rho_1 = 1, \rho_2 = 1, \rho_3 = \frac{v_g}{2\beta\max\sigma^2(\mathbf{B})}, \rho_4 = v_f, \rho_5 = v_g$, and from Assumption 2,3 we know we have all problem dependent parameters uniform bounded. Choosing the biggest step sizes we have:

$$\alpha_1 = \frac{1}{(1 + \beta\gamma)\tilde{\sigma^2}_{\mathbf{A}} + \frac{\tilde{L}_f^2}{\tilde{v}_f}}, \ \alpha_2 = \frac{1}{\frac{2\beta^2\max\tilde{\sigma}_{\mathbf{B}}^4}{\tilde{v}_g} + \frac{\tilde{L}_g^2}{\tilde{v}_g}}$$

with, additionally, δ satisfying the following:

$$\delta \leq \frac{\tilde{v}_f}{(1 + \beta\gamma)\tilde{\sigma}_{\mathbf{A}}^2 + \frac{\tilde{L}_f^2}{\tilde{v}_f}}, \ \delta \leq \frac{\tilde{v}_g}{\frac{4\beta^2\tilde{\sigma}_{\mathbf{B}}^4}{\tilde{v}_g} + \frac{2\tilde{L}_g^2}{\tilde{v}_g}}. \tag{2.43}$$

We already know that $\delta \leq \beta\gamma$; therefore, eventually, one can pick δ as

$$\delta = \min\left(\frac{\tilde{v}_f}{(1+\beta\gamma)\tilde{\sigma}_{\mathbf{A}}^2 + \frac{\tilde{L}_f^2}{\tilde{v}_f}}, \frac{\tilde{v}_g}{\frac{4\beta^2\tilde{\sigma}_{\mathbf{B}}^4}{\tilde{v}_g} + \frac{2\tilde{L}_g^2}{\tilde{v}_g}}, \beta\gamma \right).$$

As for other parameters, it turns out that:

$$\frac{1}{\beta} - \gamma - 1 \geq 0, \; \gamma - \beta\gamma \geq 0 \; \Rightarrow \; \beta\gamma + \beta \leq 1, \; \beta \leq 1.$$

■

CHAPTER 3. ZEROTH ORDER TIME-VARYING OPTIMIZATION

3.1 Introduction

In this chapter, we seek to build a *zeroth-order* dynamic distributed algorithm for *time-varying* optimization problems. Specifically, we consider cases where gradient of objective is computationally expensive or the explicit objective formulation is unknown, and we only have access to function values of the objective, i.e. zeroth-order information. For each time instance, we query the function values from a zeroth-order oracle and construct gradient estimations to feed to our algorithm. By doing this, we avoid expensive gradient evaluation and decrease computation complexity. The resulting algorithm is expected to perform faster (when gradient evaluation is expensive) and have comparable tracking ability as first-order methods (in which exact gradient information is required). We provide a thorough analysis on the convergence of proposed algorithm and some numerical results on a power system model.

3.2 Problem Formulation and Algorithm

To outline ideas, consider the *time-varying* problem

$$\min_{\mathbf{x}\in\mathcal{X}(t_k),\mathbf{y}\in\mathcal{Y}(t_k)} \; C(\mathbf{x},\mathbf{y};t_k) \tag{3.1a}$$

$$\text{s.t.} \;\; \mathbf{A}(t_k)\mathbf{x}+\mathbf{B}(t_k)\mathbf{y}=\mathbf{0}, \tag{3.1b}$$

where $\mathbf{x} = [\mathbf{x}_n]_{n\in\mathcal{N}}, \mathbf{y} = [\mathbf{y}_n]_{n\in\mathcal{N}}$ are the decision variables, $C(\mathbf{x},\mathbf{y};t_k) := \sum_n C_n(\mathbf{x}_n,\mathbf{y}_n;t_k)$ is the summation of cost functions for local nodes. Note that there is only linear equality constraint in (3.1), however, we can always add slack variables to linear inequality constraints to make them equalities. The compact version of augmented Lagrangian function for (3.1) at time t_k is shown as follows:

$$\mathcal{L}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda};t_k) = C(\mathbf{x},\mathbf{y};t_k) + \frac{\rho}{2}\left\|\mathbf{A}(t_k)\mathbf{x}+\mathbf{B}(t_k)\mathbf{y}+\frac{\boldsymbol{\lambda}}{\rho}\right\|^2. \tag{3.2}$$

We assume that the precise expression of the gradient of objective function is not available. Then a stochastic gradient estimate of $C(\mathbf{x},\mathbf{y};t_k)$ can be obtained by *one-point* or *two-point* random function evaluations [50–52]; specifically, with $\mathbf{u}$ being random vector in the unit sphere, $\delta > 0$ a user-defined constant, and N the total number of nodes, the scaled function evaluation at a perturbed point yields an estimate of the gradient:

One-point:

$$\nabla_{\mathbf{x}} C(\mathbf{x},\mathbf{y};t) \approx N\frac{C(\mathbf{x}+\delta,\mathbf{y};t)-C(\mathbf{x},\mathbf{y};t)}{\delta} = E_{\mathbf{u}}\left[\frac{N}{\delta}C(\mathbf{x}+\delta\mathbf{u},\mathbf{y};t)\mathbf{u}\right], \tag{3.3a}$$

$$\nabla_{\mathbf{y}} C(\mathbf{x},\mathbf{y};t) \approx N\frac{C(\mathbf{x},\mathbf{y}+\delta;t)-C(\mathbf{x},\mathbf{y};t)}{\delta} = E_{\mathbf{u}}\left[\frac{N}{\delta}C(\mathbf{x},\mathbf{y}+\delta\mathbf{u};t)\mathbf{u}\right], \tag{3.3b}$$

Two-point:

$$\begin{aligned}\nabla_{\mathbf{x}} C(\mathbf{x},\mathbf{y};t) \approx{}& N\frac{C(\mathbf{x}+\delta,\mathbf{y};t)-C(\mathbf{x}-\delta,\mathbf{y};t)}{2\delta}\\ ={}&E_{\mathbf{u}}\left[\frac{N}{2\delta}(C(\mathbf{x}+\delta\mathbf{u},\mathbf{y};t)-C(\mathbf{x}-\delta\mathbf{u},\mathbf{y};t))\mathbf{u}\right], \end{aligned}\tag{3.4a}$$

$$\begin{aligned}\nabla_{\mathbf{y}} C(\mathbf{x},\mathbf{y};t) \approx{}& N\frac{C(\mathbf{x},\mathbf{y}+\delta;t)-C(\mathbf{x},\mathbf{y}-\delta;t)}{2\delta}\\ ={}&E_{\mathbf{u}}\left[\frac{N}{2\delta}(C(\mathbf{x},\mathbf{y}+\delta\mathbf{u};t)-C(\mathbf{x},\mathbf{y}-\delta\mathbf{u};t))\mathbf{u}\right]. \end{aligned}\tag{3.4b}$$

According to the above expressions, let us define the following quantities: $\hat{C}(\mathbf{x},t), \hat{C}_2(\mathbf{x},t), \hat{C}(\mathbf{y},t), \hat{C}_2(\mathbf{y},t)$ be noisy *measurements* of the cost,

$$\hat{C}_1(\mathbf{x},t) = C(\mathbf{x}(t)+\delta\mathbf{u}(t),\mathbf{y}(t);t), \quad \hat{C}_1(\mathbf{y},t) = C(\mathbf{x}(t),\mathbf{y}(t)+\delta\mathbf{u}(t);t), \tag{3.5a}$$

$$\hat{C}_2(\mathbf{x},t) = C(\mathbf{x}(t)-\delta\mathbf{u}(t),\mathbf{y}(t);t), \quad \hat{C}_2(\mathbf{y},t) = C(\mathbf{x}(t),\mathbf{y}(t)-\delta\mathbf{u}(t);t), \tag{3.5b}$$

We use the following strategies to estimate the gradients:

- *One-point* estimation:

$$\hat{\nabla}_{\mathbf{x}} C(t_k) = \frac{N}{\delta}(\hat{C}_1(\mathbf{x},t_k) - C(\mathbf{x},\mathbf{y},t_k))\mathbf{u}(t_k), \tag{3.6}$$

$$\hat{\nabla}_{\mathbf{y}} C(t_k) = \frac{N}{\delta}(\hat{C}_1(\mathbf{y},t_k) - C(\mathbf{x},\mathbf{y},t_k))\mathbf{u}(t_k) \tag{3.7}$$

- *Two-point* estimation:

$$\hat{\nabla}_{\mathbf{x}} C(t_k) = \frac{N}{2\delta}(\hat{C}(\mathbf{x},t_k) - \hat{C}_2(\mathbf{x},t))\mathbf{u}(t_k), \quad \hat{\nabla}_{\mathbf{y}} C(t_k) = \frac{N}{2\delta}(\hat{C}(\mathbf{y},t_k) - \hat{C}_2(\mathbf{y},t))\mathbf{u}(t_k). \tag{3.8}$$

Using the above quantities, our proposed algorithm involves the following steps:

$$\text{Perform (3.7) or (3.8)} \tag{3.9a}$$

$$\mathbf{x}(t_{k+1}) = \text{proj}_{\mathcal{X}_R(t_k)}\Big\{\mathbf{x}(t_k) - \alpha\big(\hat{\nabla}_{\mathbf{x}}C(t_k)+\rho\mathbf{A}(t_k)^\top\Big[\mathbf{A}(t_k)\mathbf{x}(t_k) + \mathbf{B}(t_k)\mathbf{y}(t_k) + \frac{\boldsymbol{\lambda}(t_k)}{\rho}\Big]\big)\Big\} \tag{3.9b}$$

$$\mathbf{y}(t_{k+1}) = \text{proj}_{\mathcal{Y}_R(t_k)}\Big\{\mathbf{y}(t_k)-\alpha\big(\hat{\nabla}_{\mathbf{y}}C(t_k)+\rho\mathbf{B}(t_k)^\top\Big[\mathbf{A}(t_k)\mathbf{x}(t_{k+1}) + \mathbf{B}(t_k)\mathbf{y}(t_k) + \frac{\boldsymbol{\lambda}(t_k)}{\rho}\Big]\big)\Big\} \tag{3.9c}$$

$$\boldsymbol{\lambda}(t_{k+1}) = \boldsymbol{\lambda}(t_k) + \rho(\mathbf{A}(t_k)\mathbf{x}(t_{k+1}) + \mathbf{B}(t_k)\mathbf{y}(t_{k+1})) \tag{3.9d}$$

where $\mathcal{X}_R(t_k) = (1-R)\mathcal{X} := \{(1-R)x : x \in \mathcal{X}\}, \mathcal{Y}_R(t_k) = (1-R)\mathcal{Y} := \{(1-R)y : y \in \mathcal{Y}\}$ are proper restrictions of $\mathcal{X}(t_k), \mathcal{Y}(t_k)$ to ensure feasibility of the randomized point $\tilde{\mathbf{x}}(t_k), \tilde{\mathbf{y}}(t_k)$ [50,53]. The one-point or two-point estimate can also be replaced by the *multi-point estimate* as follows:

$$\hat{\nabla}_{\mathbf{x}}C(t_k) = \frac{N}{\delta(M-1)}\sum_{m=1}^{M-1}(\hat{C}(\tilde{\mathbf{x}}_m(t_k), t_k) - \hat{C}(\mathbf{x}(t_k), t_k))\mathbf{u}_m(t_k), \tag{3.10a}$$

$$\hat{\nabla}_{\mathbf{y}}C(t_k) = \frac{N}{\delta(M-1)}\sum_{m=1}^{M-1}(\hat{C}(\tilde{\mathbf{y}}_m(t_k), t_k) - \hat{C}(\mathbf{y}(t_k), t_k))\mathbf{u}_m(t_k), \tag{3.10b}$$

for $M-1$ random perturbations $\{\mathbf{u}_m(t_k)\}, \{\mathbf{v}_m(t_k)\}$ [51,52], where

$$\tilde{\mathbf{x}}_m(t_{k+1}) = \mathbf{x}(t_{k+1}) + \delta\mathbf{u}_m(t_{k+1}),$$

$$\tilde{\mathbf{y}}_m(t_{k+1}) = \mathbf{y}(t_{k+1}) + \delta\mathbf{u}_m(t_{k+1}).$$

The corresponding algorithm for multi-point estimate follows the same procedure as (3.9). In the proposed setting, one-point estimates are suitable for fast changing problems. Two-point or multi-point estimates, however, are suitable for relatively slower changing problems in exchange for the advantage of variance reduction in the gradient estimator.

3.3 Numerical Experiment

As a test to verify the feasibility of the proposed approach, we considered a *time-varying* convex optimal power flow problem (see, e.g.,chapter 2 or [54]), where clients in the network only share objective function values to the system upon receiving the power signals as shown in Figure 3.4.

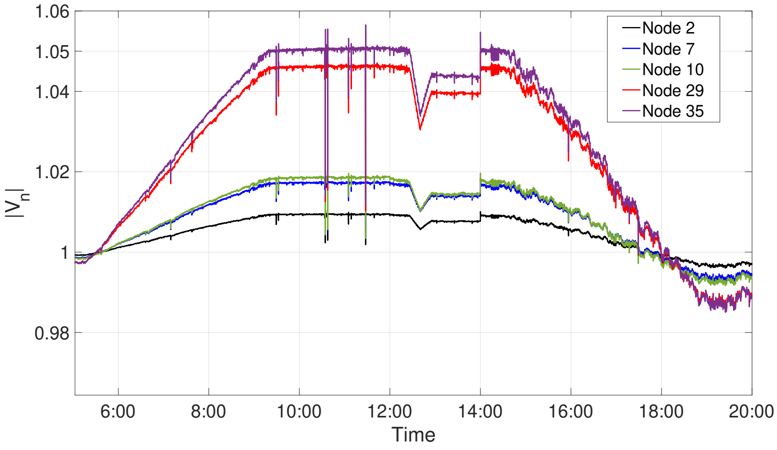


Figure 3.1: Voltage regulation achieved in real time implementation

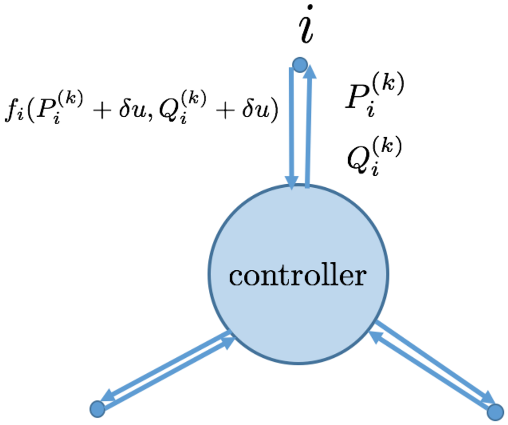


Figure 3.4: Power systems query function value from clients

A 2-point estimation strategy is utilized to approximate the objective gradient. In Figure. 3.1, we can see that zeroth order algorithm is able to regulate voltage within bounds in real time; the performance displayed here is almost as good as first order methods as shown in chapter 2. Note that even there are some large changes in the voltage profile, our algorithm is still able to track the optimal trajectory. In Fig. 3.2, the plot displays the optimal trajectory (color-coded in red) of real power at feeder head, obtained by varying the cost function as well as the the constraints and trajectory obtained by (3.9) (color-coded in green). We can see that our proposed zeroth order algorithm is able to regulate P_0^k close to optimal solution $P_{0,set}$ in real time.

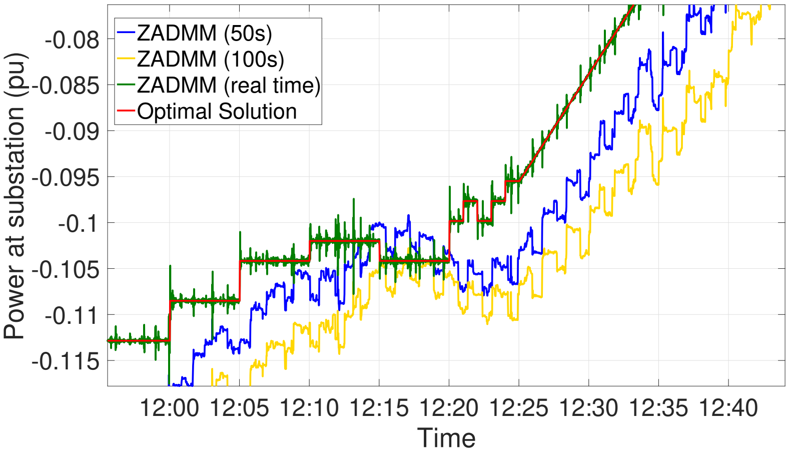


Figure 3.2: Tracking ability achieved through proposed algorithm (3.9)

We also test situations where the algorithm is implemented in a slow pace, i.e. running proposed algorithm for 50 seconds or 100 seconds without updating variables to the system. With this setup we intend to find out whether running proposed algorithm with updated information can cause problems. We can see that the trajectory is still somehow mimicking the optimal solution, but the tracking accuracy drops dramatically, which demonstrates that real time implementation is necessary in this considered application. In Figure. 3.3, we further compare voltage violations between real time implementation and slow pace implementation. The results show that slow pace implementation can cause high voltage violation and eventually damage power system.

3.4 Proof of Zeroth Order Time-Varying ADMM

For notation simplicity, we discard the time-varying notation t_k and use superscript$^{(k)}$. Let the objective function be $C(\mathbf{x}, \mathbf{y}; t_k) = f^{(k)}(\mathbf{x}) + g^{(k)}(\mathbf{y})$, our problem becomes:

$$\min\ f^{(k)}(\mathbf{x}) + g^{(k)}(\mathbf{y}) \tag{3.11a}$$

$$\text{s.t.}\ \ \mathbf{A}^{(k)}\mathbf{x} + \mathbf{B}^{(k)}\mathbf{y} = \mathbf{b}, \tag{3.11b}$$

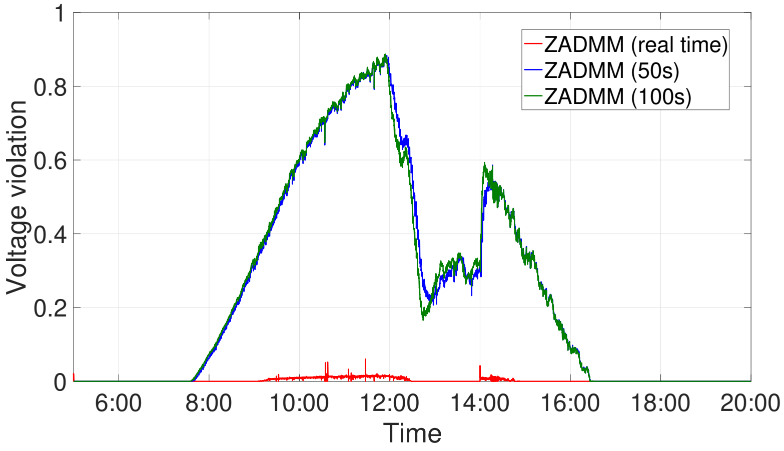


Figure 3.3: Update variables in a slow pace results in failure of voltage regulation

Define the augmented Lagrangian function (compact version) as follows:

$$\mathcal{L}^{(k)}(\mathbf{x},\mathbf{y},\boldsymbol{\lambda}) = f^{(k)}(\mathbf{x}) + g^{(k)}(\mathbf{y}) + \frac{\rho}{2}\left\|\mathbf{A}^{(k)}\mathbf{x} + \mathbf{B}^{(k)}\mathbf{y} - \mathbf{b} - \frac{\boldsymbol{\lambda}}{\rho}\right\|^2.$$

Corresponding algorithm is as follows:

$$\begin{aligned} \mathbf{y}^{(k+1)} &= \arg\min_{\mathbf{y}} \frac{1}{2\alpha}\|\mathbf{y}-\mathbf{y}^{(k)}\|^2 \\ &\quad + \left\langle \hat{\nabla} g^{(k)}(\mathbf{y}^{(k)}) + \rho(\mathbf{B}^{(k)})^T(\mathbf{A}^{(k)}\mathbf{x}^{(k)} + \mathbf{B}^{(k)}\mathbf{y}^{(k)} - \mathbf{b} - \frac{\boldsymbol{\lambda}^{(k)}}{\rho}), \mathbf{y}-\mathbf{y}^{(k)} \right\rangle, \qquad (3.12a) \\ \mathbf{x}^{(k+1)} &= \arg\min_{\mathbf{x}} \frac{1}{2\alpha}\|\mathbf{x}-\mathbf{x}^{(k)}\|^2 \\ &\quad + \left\langle \hat{\nabla} f^{(k)}(\mathbf{x}^{(k)}) + \rho(\mathbf{A}^{(k)})^T(\mathbf{A}^{(k)}\mathbf{x}^{(k)} + \mathbf{B}^{(k)}\mathbf{y}^{(k+1)} - \mathbf{b} - \frac{\boldsymbol{\lambda}^{(k)}}{\rho}), \mathbf{x}-\mathbf{x}^{(k)} \right\rangle. \qquad (3.12b) \\ \boldsymbol{\lambda}^{(k+1)} &= \boldsymbol{\lambda}^{(k)} - \rho(\mathbf{A}^{(k)}\mathbf{x}^{(k+1)} + \mathbf{B}^{(k)}\mathbf{y}^{(k+1)} - \mathbf{b}) \qquad (3.12c) \end{aligned}$$

where $\hat{\nabla} f^{(k)}, \hat{\nabla} g^{(k)}$ are zeroth order gradient estimation of $f^{(k)}, g^{(k)}$ in the form of (3.7)(3.8) or (3.10), depending on what estimate strategy we are using. For ease of readability, we first present the main theoretical result, specifying sufficient conditions under which the desired tracking capability can be obtained. Subsequently, we will then verify the conditions.

Assumption 5. *$f^{(k)}, g^{(k)}$ are strongly convex, smooth functions for each time instance k.*

Assumption 6. *Let* $\mathbf{w}^{*,(k)} = \{\mathbf{x}^{*,(k)}, \mathbf{y}^{*,(k)}, \boldsymbol{\lambda}^{*,(k)}\}$ *be the optimal solution of* (3.11) *at time* k*. Successive difference of optimal solutions of* (3.11) *is bounded, i.e.*

$$\|\mathbf{w}^{*,(k+1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}} \leq \sigma_{\mathbf{w}} \tag{3.13}$$

where $\mathbf{G}$ *is some positive definite matrix.*

Before we go to the main result, we have following properties of zeroth order estimation (see [55]):

Remark:

- Let $\hat{\nabla} f^{(k)}(\mathbf{x}^{(k)}), \hat{\nabla} g^{(k)}(\mathbf{y}^{(k)})$ be the zeroth order estimation, taking expectation w.r.t random vector $\mathbf{u}^{(k)}$ conditioned on $\mathbf{x}^{(k)}$ or $\mathbf{y}^{(k)}$ we have

$$\mathbb{E}_{\mathbf{u}^{(k)}}(\hat{\nabla} f^{(k)}(\mathbf{x}^{(k)})) = \nabla\phi_{f^{(k)}}(\mathbf{x}^{(k)}), \mathbb{E}_{\mathbf{u}^{(k)}}(\hat{\nabla} g^{(k)}(\mathbf{y}^{(k)})) = \nabla\phi_{g^{(k)}}(\mathbf{y}^{(k)}) \tag{3.14}$$

  where $\phi_{f^{(k)}}(\mathbf{x}^{(k)}) := \mathbb{E}_{\mathbf{v}}(f^{(k)}(\mathbf{x}^{(k)} + \delta\mathbf{v})), \phi_{g^{(k)}}(\mathbf{y}^{(k)}) := \mathbb{E}_{\mathbf{v}}(g^{(k)}(\mathbf{y}^{(k)} + \delta\mathbf{v}))$ are the smoothed version of $f^{(k)}(\mathbf{x}^{(k)}), g^{(k)}(\mathbf{y}^{(k)})$, $\mathbf{v}$ is drawn from a unit ball.

- f, g, ϕ_f, ϕ_g all have Lipschitz gradient (with Lipschitz constant $\hat{L}_f, \hat{L}_g, L_f, L_g$, respectively), i.e.

$$\begin{aligned}
\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| &\leq \hat{L}_f\|\mathbf{x}_1 - \mathbf{x}_2\|, \ \forall \mathbf{x}_1, \mathbf{x}_2 \\
\|\nabla g(\mathbf{y}_1) - \nabla g(\mathbf{y}_2)\| &\leq \hat{L}_g\|\mathbf{y}_1 - \mathbf{y}_2\|, \ \forall \mathbf{y}_1, \mathbf{y}_2 \\
\|\nabla \phi_f(\mathbf{x}_1) - \nabla \phi_f(\mathbf{x}_2)\| &\leq L_f\|\mathbf{x}_1 - \mathbf{x}_2\|, \ \forall \mathbf{x}_1, \mathbf{x}_2 \\
\|\nabla \phi_g(\mathbf{y}_1) - \nabla \phi_g(\mathbf{y}_2)\| &\leq L_g\|\mathbf{y}_1 - \mathbf{y}_2\|, \ \forall \mathbf{y}_1, \mathbf{y}_2
\end{aligned}$$

- We have the following bounds regarding the differences between gradient estimations $\hat{\nabla} f^{(k)}(\mathbf{x}^{(k)})$, $\hat{\nabla} g^{(k)}(\mathbf{y}^{(k)})$ and gradients of smoothed functions $\nabla\phi_{f^{(k)}}(\mathbf{x}^{(k)})$, $\nabla\phi_{g^{(k)}}(\mathbf{y}^{(k)})$

$$\mathbb{E}_{\mathbf{u}^{(k)}}(\|\hat{\nabla} f^{(k)}(\mathbf{x}^{(k)}) - \nabla\phi_{f^{(k)}}(\mathbf{x}^{(k)})\|) \leq \sigma_f \tag{3.15a}$$

$$\mathbb{E}_{\mathbf{u}^{(k)}}(\|\hat{\nabla} g^{(k)}(\mathbf{y}^{(k)}) - \nabla\phi_{g^{(k)}}(\mathbf{y}^{(k)})\|) \leq \sigma_g \tag{3.15b}$$

  where σ_f, σ_g are positive constants.

- We have the following bounds regarding the differences between $\nabla f(\mathbf{x}), \nabla g(\mathbf{y})$ and $\nabla \phi_f(\mathbf{x}), \nabla \phi_g(\mathbf{y})$

$$\|\nabla f(\mathbf{x}) - \nabla \phi_f(\mathbf{x})\| \leq \frac{\delta^2}{2} L_f N \triangleq \bar{\sigma}_f \tag{3.16a}$$

$$\|\nabla g(\mathbf{y}) - \nabla \phi_g(\mathbf{y})\| \leq \frac{\delta^2}{2} L_g N \triangleq \bar{\sigma}_g \tag{3.16b}$$

Theorem 2. *At each time instance k, suppose that Assumption 5 holds; let $\mathbf{w}^{(k)} = \{\mathbf{x}^{(k)}, \mathbf{y}^{(k)}, \boldsymbol{\lambda}^{(k)}\}$ be iterates generated by our zeroth order algorithm (3.12) and $\mathbf{w}^{*,(k)} = \{\mathbf{x}^{*,(k)}, \mathbf{y}^{*,(k)}, \boldsymbol{\lambda}^{*,(k)}\}$ be the optimal solution of (3.11) at time k; If the following is true for some positive definite matrix $\mathbf{G}$ and $\psi = \mathcal{F}(\sigma_f, \sigma_g, \bar{\sigma}_f, \bar{\sigma}_g, R) \geq 0$,*

$$\mathbb{E}(\|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) \leq r\mathbb{E}(\|\mathbf{w}^{(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) + r\psi, \tag{3.17}$$

where $0 < r < 1$ is a constant, then it holds that:

$$\begin{aligned}\lim_{k\to\infty} \sup \mathbb{E}(\|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) &\leq \lim_{k\to\infty} \left(\frac{r(1-r^k)}{1-r}\sigma_{\mathbf{w}} + r^k \mathbb{E}(\|\mathbf{w}^{(0)} - \mathbf{w}^{*,(0)}\|_{\mathbf{G}}) + \frac{r(1-r^k)}{1-r}\psi \right) \\ &\leq \frac{r(\sigma_{\mathbf{w}} + \psi)}{1-r},\end{aligned} \tag{3.18}$$

where $\sigma_{\mathbf{w}}$ is some positive constant.

Clearly, this theorem critically depends on the sufficient condition (3.17), which indicates that the iterates generated by the algorithm exhibit are contracting updates, but with a bias term. In fact, (3.17) can be regarded as linear convergence of $\mathbf{w}^{(k)}$ for each fixed time instance k (i.e. static case) with bias. Next, we discuss conditions under which (3.17) holds true. Also, notice that there is no term related to initialization error in (3.18), which indicates that no matter how far away the iterate is from the optimal solution, our proposed algorithm is able to steer it to track optimal solutions.

3.4.1 Linear Convergence for Static Case

Proposition 1. *For fixed time instance k, let $\{\mathbf{w}^k\}$ be the sequence generated by zeroth order algorithm (3.12), $\mathbf{w}^*$ be the optimal solution of problem (3.11), we have the following linear convergence with bias:*

$$\mathbb{E}(\|\mathbf{w}^k - \mathbf{w}^*\|_{\mathbf{G}}^2) \geq (1+\gamma)\mathbb{E}(\|\mathbf{w}^{k+1} - \mathbf{w}^*\|_{\mathbf{G}}^2) - \psi, \tag{3.19}$$

where $\psi \geq 0$ *is a constant,* $0 < \gamma < 1$

Proof. Optimality condition for the problem (3.11) for a fixed time instance is as follows:

$$\mathbf{A}^T\boldsymbol{\lambda}^* = \nabla f(\mathbf{x}^*) \tag{3.20a}$$

$$\mathbf{B}^T\boldsymbol{\lambda}^* = \nabla g(\mathbf{y}^*) \tag{3.20b}$$

$$\mathbf{A}\mathbf{x}^*{+}\mathbf{B}\mathbf{y}^* - \mathbf{b} = 0 \tag{3.20c}$$

Optimality condition for the updates are as follows:

$$\begin{aligned}&\mathbf{B}^T(\boldsymbol{\lambda}^{k+1} + \rho\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k) + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k))\\ &+ \frac{1}{\alpha_2}(\mathbf{y}^k - \mathbf{y}^{k+1}) + \frac{1}{\alpha_2}(\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}) = \hat{\nabla} g(\mathbf{y}^k) &(3.21a)\\ &\mathbf{A}^T(\boldsymbol{\lambda}^{k+1} + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)) + \frac{1}{\alpha_1}(\mathbf{x}^k - \mathbf{x}^{k+1}) + \frac{1}{\alpha_1}(\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}) = \hat{\nabla} f(\mathbf{x}^k) &(3.21b)\end{aligned}$$

where $\hat{\nabla} g(\mathbf{y}^k), \hat{\nabla} f(\mathbf{x}^k)$ are zeroth order gradient estimations; $\bar{\mathbf{x}}^{(k+1)}, \bar{\mathbf{y}}^{(k+1)}$ are iterates before projection. Define the gradient of smoothed version of f, g as ϕ_f, ϕ_g , we make the following changes

$$\begin{aligned}&\mathbf{B}^T(\boldsymbol{\lambda}^{k+1} + \rho\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k) + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)) + \frac{1}{\alpha_2}(\mathbf{y}^k - \mathbf{y}^{k+1}) + \frac{1}{\alpha_2}(\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1})\\ &+ \nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1}) + \nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k) + \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k) = \nabla g(\mathbf{y}^{k+1}) &(3.22a)\\ &\mathbf{A}^T(\boldsymbol{\lambda}^{k+1} + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)) + \frac{1}{\alpha_1}(\mathbf{x}^k - \mathbf{x}^{k+1}) + \frac{1}{\alpha_1}(\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1})\\ &+ \nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1}) + \nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k) + \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k) = \nabla f(\mathbf{x}^{k+1}) &(3.22b)\end{aligned}$$

We know $f(\mathbf{x})$ and $g(\mathbf{y})$ are strongly convex, then we have

$$\langle \mathbf{x}_1 - \mathbf{x}_2, \nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\rangle \geq v_f\|\mathbf{x}_1 - \mathbf{x}_2\|^2$$

$$\langle \mathbf{y}_1 - \mathbf{y}_2, \nabla g(\mathbf{y}_1) - \nabla g(\mathbf{y}_2)\rangle \geq v_g\|\mathbf{y}_1 - \mathbf{y}_2\|^2$$

Let $\mathbf{x}_1 = \mathbf{x}^{k+1}, \mathbf{x}_2 = \mathbf{x}^*, \mathbf{y}_1 = \mathbf{y}^{k+1}, \mathbf{y}_2 = \mathbf{y}^*$ and use (3.20) we have

$$\begin{aligned}
&\langle \mathbf{B}^T(\boldsymbol{\lambda}^{k+1} + \beta\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k) + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)) + \frac{1}{\alpha_2}(\mathbf{y}^k - \mathbf{y}^{k+1}) + \frac{1}{\alpha_2}(\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}) - \mathbf{B}^T\boldsymbol{\lambda}^* \\
&+ \nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1}) + \nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k) \\
&+ \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \geq v_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \qquad (3.23) \\
&\langle \mathbf{A}^T(\boldsymbol{\lambda}^{k+1} + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)) + \frac{1}{\alpha_1}(\mathbf{x}^k - \mathbf{x}^{k+1}) + \frac{1}{\alpha_1}(\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}) + \nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1}) \\
&+ \nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k) + \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k) - \mathbf{A}^T\boldsymbol{\lambda}^*, \mathbf{x}^{k+1} - \mathbf{x}^*\rangle \geq v_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \qquad (3.24)
\end{aligned}$$

Summing up (3.23),(3.24) we have

$$\begin{aligned}
&\langle \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^* + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^*)\rangle + \langle\beta\langle\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^*)\rangle \\
&+ \frac{1}{\alpha_2}\langle\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + \frac{1}{\alpha_2}\langle\mathbf{y}^k - \mathbf{y}^{k+1}, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + \langle\nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1}), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&+ \langle\nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + \langle\nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&+ \langle\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^* + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^*)\rangle + \frac{1}{\alpha_1}\langle\mathbf{x}^k - \mathbf{x}^{k+1}, \mathbf{x}^{k+1} - \mathbf{x}^*\rangle \\
&+ \frac{1}{\alpha_1}\langle\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}, \mathbf{x}^{k+1} - \mathbf{x}^*\rangle + \langle\nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle \\
&+ \langle\nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle + \langle\nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle \\
&\geq v_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + v_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \qquad (3.25)
\end{aligned}$$

Also we know from the optimality condition (3.20c) and dual updates that

$$\frac{1}{\beta}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1}) = \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^*) + \mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^*) \qquad (3.26)$$

Plugging (3.26) back to (3.25) we have

$$\begin{aligned}
&\langle \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^* + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \frac{1}{\beta}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1})\rangle + \langle\beta\langle\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k), \mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^*))\rangle \\
&+ \frac{1}{\alpha_2}\langle\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + \frac{1}{\alpha_2}\langle\mathbf{y}^k - \mathbf{y}^{k+1}, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&+ \langle\nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1}), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + \langle\nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle \\
&+ \langle\nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^{k+1} - \mathbf{y}^*\rangle + \frac{1}{\alpha_1}\langle\mathbf{x}^k - \mathbf{x}^{k+1}, \mathbf{x}^{k+1} - \mathbf{x}^*\rangle \\
&+ \langle\nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1}), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle + \frac{1}{\alpha_1}\langle\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}, \mathbf{x}^{k+1} - \mathbf{x}^*\rangle \\
&+ \langle\nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle + \langle\nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^{k+1} - \mathbf{x}^*\rangle \\
&\geq v_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + v_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \qquad (3.27)
\end{aligned}$$

We know that

$$\begin{aligned}
\frac{1}{\alpha_2}\langle\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}, \mathbf{y}^{k+1} - \mathbf{y}^*\rangle &= \frac{1}{\alpha_2}\langle\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}, \mathbf{y}^{k+1} - \tilde{\mathbf{y}}^*\rangle + \frac{1}{\alpha_2}\langle\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}, \tilde{\mathbf{y}}^* - \mathbf{y}^*\rangle \\
\frac{1}{\alpha_1}\langle\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}, \mathbf{x}^{k+1} - \mathbf{x}^*\rangle &= \frac{1}{\alpha_1}\langle\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}, \mathbf{x}^{k+1} - \tilde{\mathbf{x}}^*\rangle + \frac{1}{\alpha_1}\langle\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}, \tilde{\mathbf{x}}^* - \mathbf{x}^*\rangle
\end{aligned}$$

where $\tilde{\mathbf{x}}^*, \tilde{\mathbf{y}}^*$ are projections of $\mathbf{x}^*, \mathbf{y}^*$ onto $\mathcal{X}_R, \mathcal{Y}_R$. Note that if $\mathbf{x}^*, \mathbf{y}^*$ are inside $\mathcal{X}_R, \mathcal{Y}_R$, $\tilde{\mathbf{x}}^* = \mathbf{x}^*, \tilde{\mathbf{y}}^* = \mathbf{y}^*$. Since $\mathbf{x}^{(k+1)}, \tilde{\mathbf{x}}^* \in \mathcal{X}_R$ and $\mathbf{y}^{(k+1)}, \tilde{\mathbf{y}}^* \in \mathcal{Y}_R$, we have

$$\begin{aligned}
\langle\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}, \tilde{\mathbf{x}}^* - \mathbf{x}^{k+1}\rangle &\geq 0, \\
\langle\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}, \tilde{\mathbf{y}}^* - \mathbf{y}^{k+1}\rangle &\geq 0.
\end{aligned}$$

Also we have

$$\begin{aligned}
\langle\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}, \tilde{\mathbf{y}}^* - \mathbf{y}^*\rangle &\geq -\frac{1}{2\tau}\|\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}\|^2 - \frac{\tau}{2}\|\mathbf{y}^* - \tilde{\mathbf{y}}^*\|^2 \\
\langle\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}, \tilde{\mathbf{x}}^* - \mathbf{x}^*\rangle &\geq -\frac{1}{2\tau}\|\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}\|^2 - \frac{\tau}{2}\|\mathbf{x}^* - \tilde{\mathbf{x}}^*\|^2
\end{aligned}$$

We know that $\|\mathbf{y}^{k+1} - \bar{\mathbf{y}}^{k+1}\|^2 \leq \epsilon^2,\ \|\mathbf{x}^{k+1} - \bar{\mathbf{x}}^{k+1}\|^2 \leq \epsilon^2$ (see [56]) and $\|\tilde{\mathbf{y}}^* - \mathbf{y}^*\|^2 \leq R^2,\ \|\tilde{\mathbf{x}}^* - \mathbf{x}^*\|^2 \leq R^2$.

Define $\mathbf{G} = \begin{pmatrix} \frac{1}{\alpha_1}\mathbf{I} & & \\ & \frac{1}{\alpha_2}\mathbf{I} - \beta\mathbf{B}^T\mathbf{B} & \\ & & \frac{1}{\beta}\mathbf{I} \end{pmatrix}$, $\mathbf{w} = (\mathbf{x};\mathbf{y};\boldsymbol{\lambda})$ we can derive

$$\begin{aligned}
&(\mathbf{w}^k - \mathbf{w}^{k+1})^T\mathbf{G}(\mathbf{w}^{k+1} - \mathbf{w}^*) \geq \langle \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\rangle + \langle \nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1}), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle \\
&+ \langle \nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle + \langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle \\
&+ \langle \nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1}), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle + \langle \nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle \\
&+ \langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle + v_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + v_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \\
&- \frac{\epsilon^2}{2\alpha_1\tau} - \frac{\epsilon^2}{2\alpha_2\tau} - \frac{\tau R^2}{2\alpha_1} - \frac{\tau R^2}{2\alpha_2}
\end{aligned}$$

$\Rightarrow$

$$\begin{aligned}
&(\mathbf{w}^k - \mathbf{w}^{k+1})^T\mathbf{G}(\mathbf{w}^k - \mathbf{w}^*) \geq \|\mathbf{w}^k - \mathbf{w}^{k+1}\|_{\mathbf{G}}^2 + \langle \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\rangle \\
&+ \langle \nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1}), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle + \langle \nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle \\
&+ \langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle + \langle \nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1}), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle \\
&+ \langle \nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle + \langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle \\
&+ v_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + v_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 - \frac{\epsilon^2}{2\alpha_1\tau} - \frac{\epsilon^2}{2\alpha_2\tau} - \frac{\tau R^2}{2\alpha_1} - \frac{\tau R^2}{2\alpha_2}
\end{aligned} \tag{3.28}$$

Recall the following equality

$$\|\mathbf{w}^k - \mathbf{w}^*\|_{\mathbf{G}}^2 - \|\mathbf{w}^{k+1} - \mathbf{w}^*\|_{\mathbf{G}}^2 = 2(\mathbf{w}^k - \mathbf{w}^{k+1})^T\mathbf{G}(\mathbf{w}^k - \mathbf{w}^*) - \|\mathbf{w}^k - \mathbf{w}^{k+1}\|_{\mathbf{G}}^2 \tag{3.29}$$

Plugging back to (3.28) we have

$$\begin{aligned}
&\|\mathbf{w}^k - \mathbf{w}^*\|_{\mathbf{G}}^2 - \|\mathbf{w}^{k+1} - \mathbf{w}^*\|_{\mathbf{G}}^2 \geq \|\mathbf{w}^k - \mathbf{w}^{k+1}\|_{\mathbf{G}}^2 + 2\langle \mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k), \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\rangle \\
&+ 2\langle \nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1}), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle + 2\langle \nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle \\
&+ 2\langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle + 2\langle \nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1}), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle \\
&+ 2\langle \nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle + 2\langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle \\
&+ 2v_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + 2v_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \\
&- \frac{\epsilon^2}{\alpha_1\tau} - \frac{\epsilon^2}{\alpha_2\tau} - \frac{\tau R^2}{\alpha_1} - \frac{\tau R^2}{\alpha_2}
\end{aligned} \tag{3.30}$$

For the cross terms in (3.30) we do the following:

$$
\begin{aligned}
&\langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^{k+1}\rangle = \langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^k\rangle + \langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^k - \mathbf{y}^{k+1}\rangle \\
&\geq \langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^k\rangle - \frac{\eta}{2}\|\nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k)\|^2 - \frac{1}{2\eta}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 \\
&\langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^{k+1}\rangle = \langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^k\rangle + \langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^k - \mathbf{x}^{k+1}\rangle \\
&\geq \langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^k\rangle - \frac{\eta}{2}\|\nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k)\|^2 - \frac{1}{2\eta}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2
\end{aligned}
$$

Utilize Cauchy-Schwartz inequality to (3.30) we have

$$
\begin{aligned}
&\|\mathbf{w}^k - \mathbf{w}^*\|_{\mathbf{G}}^2 - \|\mathbf{w}^{k+1} - \mathbf{w}^*\|_{\mathbf{G}}^2 \geq \|\mathbf{w}^k - \mathbf{w}^{k+1}\|_{\mathbf{G}}^2 - \rho_1\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 - \frac{1}{\rho_1}\|\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)\|^2 \\
&- \rho_5\|\nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1})\|^2 - \frac{1}{\rho_5}\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 - \rho_2\|\nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k)\|^2 - \frac{1}{\rho_2}\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \\
&+ \langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^k\rangle - \eta\|\nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k)\|^2 - \frac{1}{\eta}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 \\
&+ \langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^k\rangle - \eta\|f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x})^k\|^2 - \frac{1}{\eta}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \\
&- \rho_6\|\nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1})\|^2 - \frac{1}{\rho_6}\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 - \rho_4\|\nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k)\|^2 \\
&- \frac{1}{\rho_4}\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + 2\upsilon_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2) + 2\upsilon_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 - \frac{\epsilon^2}{\alpha_1\tau} - \frac{\epsilon^2}{\alpha_2\tau} - \frac{\tau R^2}{\alpha_1} - \frac{\tau R^2}{\alpha_2} \qquad (3.31)
\end{aligned}
$$

We know ϕ_g, ϕ_f have Lipschitz gradients, thus we have

$$
\begin{aligned}
&\|\mathbf{w}^k - \mathbf{w}^*\|_{\mathbf{G}}^2 - \|\mathbf{w}^{k+1} - \mathbf{w}^*\|_{\mathbf{G}}^2 \geq \|\mathbf{w}^k - \mathbf{w}^{k+1}\|_{\mathbf{G}}^2 - \rho_1\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 - \frac{1}{\rho_1}\|\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)\|^2 \\
&- \rho_5\|\nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1})\|^2 - \frac{1}{\rho_5}\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 - \rho_2 L_g^2\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 - \frac{1}{\rho_2}\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \\
&+ \langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^k\rangle - \eta\|\nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k)\|^2 - \frac{1}{\eta}\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2 \\
&+ \langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^k\rangle - \eta\|f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x})^k\|^2 - \frac{1}{\eta}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \\
&- \rho_6\|\nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1})\|^2 - \frac{1}{\rho_6}\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 - \rho_4 L_f^2\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 - \frac{1}{\rho_4}\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 \\
&+ 2\upsilon_f\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2) + 2\upsilon_g\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 - \frac{\epsilon^2}{\alpha_1\tau} - \frac{\epsilon^2}{\alpha_2\tau} - \frac{\tau R^2}{\alpha_1} - \frac{\tau R^2}{\alpha_2} \qquad (3.32)
\end{aligned}
$$

Take expectation to both side of (3.30) w.r.t random vector $\mathbf{u}$ conditioned on $(\mathbf{x}^k; \mathbf{y}^k)$ (here we write $\mathbb{E}$ instead of $\mathbb{E}_{\mathbf{u}^k}$ for notation simplicity) , we have

$$\mathbb{E}(\langle \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k), \mathbf{y}^* - \mathbf{y}^k \rangle) = \langle \nabla\phi_g(\mathbf{y}^k) - \mathbb{E}(\hat{\nabla} g(\mathbf{y}^k)), \mathbf{y}^* - \mathbf{y}^k \rangle = 0$$

$$\mathbb{E}(\langle \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k), \mathbf{x}^* - \mathbf{x}^k \rangle) = \langle \nabla\phi_f(\mathbf{x}^k) - \mathbb{E}(\hat{\nabla} f(\mathbf{x}^k)), \mathbf{x}^* - \mathbf{x}^k \rangle = 0$$

where we have utilized the fact that $\mathbb{E}(\hat{\nabla} f(\mathbf{x}^k)) = \nabla\phi_f(\mathbf{x}^k), \mathbb{E}(\hat{\nabla} g(\mathbf{y}^k)) = \nabla\phi_g(\mathbf{y}^k)$. Now (3.32) becomes

$$\begin{aligned}
&\mathbb{E}(\|\mathbf{w}^k - \mathbf{w}^*\|_{\mathbf{G}}^2) - \mathbb{E}(\|\mathbf{w}^{k+1} - \mathbf{w}^*\|_{\mathbf{G}}^2) \geq \mathbb{E}(\|\mathbf{w}^k - \mathbf{w}^{k+1}\|_{\mathbf{G}}^2) - \rho_1 \mathbb{E}(\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2) \\
&- \frac{1}{\rho_1}\mathbb{E}(\|\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k))\|^2 - L_g^2\rho_2\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2) - \frac{1}{\rho_2}\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2) \\
&- \eta\mathbb{E}(\|\nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k)\|^2) - \frac{1}{\eta}\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2) - \rho_5\mathbb{E}(\|\nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1})\|^2) \\
&- \frac{1}{\rho_5}\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2) - \rho_6\mathbb{E}(\|\nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1})\|^2) - \frac{1}{\rho_6}\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2) \\
&- L_f^2\rho_4\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2) - \frac{1}{\rho_4}\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2) - \eta\mathbb{E}(\|f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x})^k\|^2) - \frac{1}{\eta}\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2) \\
&+ 2v_f\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2) + 2v_g\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2) - \frac{\epsilon^2}{\alpha_1\tau} - \frac{\epsilon^2}{\alpha_2\tau} - \frac{\tau R^2}{\alpha_1} - \frac{\tau R^2}{\alpha_2}
\end{aligned} \tag{3.33}$$

We know that $\mathbb{E}\|\nabla\phi_f(\mathbf{x}^{k+1}) - \hat{\nabla} f(\mathbf{x}^{k+1})\|^2 \leq \sigma_f^2, \mathbb{E}\|\nabla\phi_g(\mathbf{y}^{k+1}) - \hat{\nabla} g(\mathbf{y}^{k+1})\|^2 \leq \sigma_g^2$, where σ_f, σ_g are maximal expected deviation of the gradient estimate (see e.g. [52]). Also we know that the difference between smoothed version gradient and exact gradient is bounded, i.e. (3.16). Now (3.33) becomes

$$\begin{aligned}
&\mathbb{E}(\|\mathbf{w}^k - \mathbf{w}^*\|_{\mathbf{G}}^2) - \mathbb{E}(\|\mathbf{w}^{k+1} - \mathbf{w}^*\|_{\mathbf{G}}^2) \geq \mathbb{E}(\|\mathbf{w}^k - \mathbf{w}^{k+1}\|_{\mathbf{G}}^2) - \rho_1 \mathbb{E}(\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2) \\
&- \frac{1}{\rho_1}\mathbb{E}(\|\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k))\|^2 - L_g^2\rho_2\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2) - (\frac{1}{\rho_2} + \frac{1}{\rho_5})\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2) - \eta\sigma_g^2 - \rho_5\bar{\sigma}_g^2 \\
&- \frac{1}{\eta}\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2) - L_f^2\rho_4\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2) - (\frac{1}{\rho_4} + \frac{1}{\rho_6})\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2) - \eta\sigma_f^2 - \rho_6\bar{\sigma}_f^2 \\
&- \frac{1}{\eta}\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2) + 2v_f\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2) + 2v_g\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2) \\
&- \frac{\epsilon^2}{\alpha_1\tau} - \frac{\epsilon^2}{\alpha_2\tau} - \frac{\tau R^2}{\alpha_1} - \frac{\tau R^2}{\alpha_2}
\end{aligned} \tag{3.34}$$

$$\begin{aligned}
&\geq (\frac{1}{\beta} - \rho_1)\mathbb{E}(\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2) + (\frac{1}{\alpha_1} - \frac{1}{\rho_1}\mathbf{A}^T\mathbf{A} - L_f^2\rho_4\mathbf{I} - \frac{1}{\eta}\mathbf{I})\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2) \\
&- \eta\sigma_g^2 - \eta\sigma_f^2 - \rho_5\bar{\sigma}_g^2 - \rho_6\bar{\sigma}_f^2 + (\frac{1}{\alpha_2} - \beta\mathbf{B}^T\mathbf{B} - L_g^2\rho_2\mathbf{I} - \frac{1}{\eta}\mathbf{I})\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^k\|^2) \\
&+ (2v_g - \frac{1}{\rho_2})\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2) + (2v_f - \frac{1}{\rho_4})\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2) - \frac{\epsilon^2}{\alpha_1\tau} - \frac{\epsilon^2}{\alpha_2\tau} - \frac{\tau R^2}{\alpha_1} - \frac{\tau R^2}{\alpha_2}
\end{aligned} \tag{3.35}$$

Recall that our goal is to prove

$$\mathbb{E}(\|\mathbf{w}^k - \mathbf{w}^*\|^2_{\mathbf{G}}) \geq (1+\gamma)\mathbb{E}(\|\mathbf{w}^{k+1} - \mathbf{w}^*\|^2_{\mathbf{G}}) - \psi \tag{3.36}$$

Therefore, we need to make sure

$$C \geq \gamma\mathbb{E}(\|\mathbf{w}^{k+1} - \mathbf{w}^*\|^2_{\mathbf{G}}) - \psi, \tag{3.37}$$

where C is the right hand side of (3.35). We can see that there's no $\mathbb{E}(\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2)$ term in (3.35), using similar idea as in [15] we can bound this term with other terms. First from (3.21a),(3.21b) we know that

$$\begin{aligned}
&\mathbf{B}^T(\boldsymbol{\lambda}^{k+1} + \rho\mathbf{B}(\mathbf{y}^{k+1} - \mathbf{y}^k) + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)) + \frac{1}{\alpha}(\mathbf{y}^k - \mathbf{y}^{k+1}) \\
&+ \nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1}) + \nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k) + \nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k) = \nabla g(\mathbf{y}^{k+1}) \qquad (3.38) \\
&\mathbf{A}^T(\boldsymbol{\lambda}^{k+1} + \beta\mathbf{A}(\mathbf{x}^{k+1} - \mathbf{x}^k)) + \frac{1}{\alpha}(\mathbf{x}^k - \mathbf{x}^{k+1}) \\
&+ \nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1}) + \nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k) + \nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k) = \nabla f(\mathbf{x}^{k+1}) \qquad (3.39)
\end{aligned}$$

Using optimality condition $\mathbf{A}^T\boldsymbol{\lambda}^* = \nabla f(\mathbf{x}^*), \mathbf{B}^T\boldsymbol{\lambda}^* = \nabla g(\mathbf{y}^*)$ and Lipschitz continuity of $\nabla f(\mathbf{x}), \nabla g(\mathbf{y})$,

$$\begin{aligned}
&\|\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^*)\|^2 + \|\nabla g(\mathbf{y}^{k+1}) - \nabla g(\mathbf{y}^*)\|^2 \\
&= \| \begin{pmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{pmatrix} (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) + \begin{pmatrix} \beta\mathbf{A}^T\mathbf{A} - \frac{1}{\alpha_1}\mathbf{I} \\ \beta\mathbf{B}^T\mathbf{A} \end{pmatrix} (\mathbf{x}^{k+1} - \mathbf{x}^k) + \begin{pmatrix} \mathbf{0} \\ \beta\mathbf{B}^T\mathbf{B} - \frac{1}{\alpha_2}\mathbf{I} \end{pmatrix} (\mathbf{y}^{k+1} - \mathbf{y}^k) \\
&+ \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix} (\nabla\phi_g(\mathbf{y}^k) - \hat{\nabla} g(\mathbf{y}^k)) + \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix} (\nabla\phi_g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^k)) + \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix} (\nabla g(\mathbf{y}^{k+1}) - \nabla\phi_g(\mathbf{y}^{k+1})) \\
&+ \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix} (\nabla\phi_f(\mathbf{x}^k) - \hat{\nabla} f(\mathbf{x}^k)) + \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix} (\nabla\phi_f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^k)) + \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix} (\nabla f(\mathbf{x}^{k+1}) - \nabla\phi_f(\mathbf{x}^{k+1}))\|^2 \\
&\leq \hat{L}_f^2\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2 + \hat{L}_g^2\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2 \qquad (3.40)
\end{aligned}$$

$\mathbf{A}, \mathbf{B}$ are not necessarily full row rank matrices, so without loss of generality, assume the first r rows of $[\mathbf{A}, \mathbf{B}]$ (denoted as $[\mathbf{A}_r, \mathbf{B}_r]$) are linear independent, we have

$$[\mathbf{A}, \mathbf{B}] = \begin{bmatrix} \mathbf{I} \\ \mathbf{L} \end{bmatrix} [\mathbf{A}_r, \mathbf{B}_r]$$

If the initial $\boldsymbol{\lambda}^0$ is in the range space of $[\mathbf{A}, \mathbf{B}]$ then $\boldsymbol{\lambda}^{k+1}$ always stays in the range space of $[\mathbf{A}, \mathbf{B}]$, it follows that

$$\boldsymbol{\lambda}^{k+1} = \begin{bmatrix} \mathbf{I} \\ \mathbf{L} \end{bmatrix} \boldsymbol{\lambda}_r^{k+1}, \ \boldsymbol{\lambda}^* = \begin{bmatrix} \mathbf{I} \\ \mathbf{L} \end{bmatrix} \boldsymbol{\lambda}_r^*$$

Thus we have

$$\begin{aligned} \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix} (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) &= \begin{bmatrix} \mathbf{A}_r^T \\ \mathbf{B}_r^T \end{bmatrix} (\mathbf{I} + \mathbf{L}^T\mathbf{L})(\boldsymbol{\lambda}_r^{k+1} - \boldsymbol{\lambda}_r^*) \\ \Rightarrow \quad \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2 \leq \bar{c} &\left\| \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix} (\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*) \right\|^2 \end{aligned}$$

where $\mathbf{I} \in \mathbb{R}^{r\times r}$ is the identity matrix and $\mathbf{L} \in \mathbb{R}^{(m-r)\times r}$, $\mathbf{E} = (\mathbf{I}+\mathbf{L}^T\mathbf{L})[\mathbf{A}_r, \mathbf{B}_r]$ and $\bar{c} = \lambda_{min}^{-1}(\mathbf{E}\mathbf{E}^T)\|\mathbf{I}+\mathbf{L}^T\mathbf{L}\| > 0$. Use the following inequalities to the LHS of (3.40)

$$\begin{aligned} &\|p+q\|^2 \geq (1-\frac{1}{\mu})\|p\|^2 + (1-\mu)\|q\|^2 \\ &\|p+q\|^2 \leq (1+\frac{1}{\mu})\|p\|^2 + (1+\mu)\|q\|^2, \forall \mu > 0 \end{aligned}$$

and taking expectation to both hand side of (3.40) we have

$$\begin{aligned} \mathbb{E}(\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\|^2) \leq & c_1\mathbb{E}(\|\mathbf{x}^k - \mathbf{x}^{k+1}\|^2) + c_2\mathbb{E}(\|\mathbf{y}^k - \mathbf{y}^{k+1}\|^2) + c_3\mathbb{E}(\|\mathbf{x}^{k+1} - \mathbf{x}^*\|^2) \\ & + c_4\mathbb{E}(\|\mathbf{y}^{k+1} - \mathbf{y}^*\|^2) + c_5\sigma_f^2 + c_6\sigma_g^2 + c_7\bar{\sigma}_f^2 + c_8\bar{\sigma}_g^2 \end{aligned} \tag{3.41}$$

where

$$\begin{aligned} &c_1 = \mu_1(1+\frac{1}{\mu_2})\|[\beta\mathbf{A}^T\mathbf{A} - \frac{1}{\alpha_1}\mathbf{I}, \beta\mathbf{A}^T\mathbf{B}]\|^2\bar{c} + \mu_1\bar{c}\prod_{i=2}^{7}(1+\mu_i)(1+\frac{1}{\mu_8})L_f^2, \\ &c_2 = \mu_1(1+\mu_2)(1+\frac{1}{\mu_3})\|\beta\mathbf{B}^T\mathbf{B} - \frac{1}{\alpha_2}\mathbf{I}\|^2\bar{c} + \mu_1\bar{c}\prod_{i=2}^{4}(1+\mu_i)(1+\frac{1}{\mu_5})L_g^2, \\ &c_3 = (1-\frac{1}{\mu_1})^{-1}L_f^2\bar{c}, c_4 = (1-\frac{1}{\mu_1})^{-1}L_g^2\bar{c}, \\ &c_5 = \mu_1\prod_{i=2}^{6}(1+\frac{1}{\mu_7})\bar{c}, \quad c_6 = \mu_1(1+\mu_2)(1+\mu_3)(1+\frac{1}{\mu_4})\bar{c} \\ &c_7 = \mu_1\prod_{i=2}^{8}(1+\mu_i)\bar{c}, \quad c_8 = \mu_1\prod_{i=2}^{5}(1+\frac{1}{\mu_6})\bar{c} \end{aligned}$$

Plugging (3.41), (3.35) to (3.37) we need the following to be true

$$\begin{aligned}
&(\frac{1}{\beta}-\rho_1)\mathbb{E}(\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2)+(\frac{1}{\alpha_1}-\frac{1}{\rho_1}\mathbf{A}^T\mathbf{A}-L_f^2\rho_4\mathbf{I}-\frac{1}{\eta}\mathbf{I})\mathbb{E}(\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2)\\
&-\eta\sigma_g^2-\eta\sigma_f^2-\rho_5\bar{\sigma}_g^2-\rho_6\bar{\sigma}_f^2+(\frac{1}{\alpha_2}-\beta\mathbf{B}^T\mathbf{B}-L_g^2\rho_2\mathbf{I}-\frac{1}{\eta}\mathbf{I})\mathbb{E}(\|\mathbf{y}^{k+1}-\mathbf{y}^k\|^2)\\
&+(2v_g-\frac{1}{\rho_2})\mathbb{E}(\|\mathbf{y}^{k+1}-\mathbf{y}^*\|^2)+(2v_f-\frac{1}{\rho_4})\mathbb{E}(\|\mathbf{x}^{k+1}-\mathbf{x}^*\|^2)\\
&-\frac{\epsilon^2}{\alpha_1\tau}-\frac{\epsilon^2}{\alpha_2\tau}-\frac{\tau R^2}{\alpha_1}-\frac{\tau R^2}{\alpha_2}\geq\gamma\mathbb{E}(\|\mathbf{w}^{k+1}-\mathbf{w}^*\|_{\mathbf{G}}^2)-\psi
\end{aligned}\tag{3.42}$$

Rearrange terms we get

$$\begin{aligned}
&(\frac{1}{\beta}-\rho_1)\mathbb{E}(\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\|^2)+(\frac{1}{\alpha_1}-\frac{1}{\rho_1}\mathbf{A}^T\mathbf{A}-L_f^2\rho_4\mathbf{I}-\frac{1}{\eta}\mathbf{I}-c_1\mathbf{I})\mathbb{E}(\|\mathbf{x}^{k+1}-\mathbf{x}^k\|^2)\\
&-(c_6+\eta)\sigma_g^2-(c_5+\eta)\sigma_f^2+(\frac{1}{\alpha_2}-\beta\mathbf{B}^T\mathbf{B}-L_g^2\rho_2\mathbf{I}-\frac{1}{\eta}\mathbf{I}-c_2\mathbf{I})\mathbb{E}(\|\mathbf{y}^{k+1}-\mathbf{y}^k\|^2)\\
&+(2v_g\mathbf{I}-\frac{1}{\rho_2}\mathbf{I}-c_4\mathbf{I}-\frac{\gamma}{\alpha_2}\mathbf{I}+\gamma\beta\mathbf{B}^T\mathbf{B})\mathbb{E}(\|\mathbf{y}^{k+1}-\mathbf{y}^*\|^2)+(1-\frac{\gamma}{\beta})\mathbb{E}(\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*\|^2)-c_7\bar{\sigma}_f^2-c_8\bar{\sigma}_g^2\\
&+(2v_f-\frac{1}{\rho_4}-c_3-\frac{\gamma}{\alpha_1})\mathbb{E}(\|\mathbf{x}^{k+1}-\mathbf{x}^*\|^2)-\frac{\epsilon^2}{\alpha_1\tau}-\frac{\epsilon^2}{\alpha_2\tau}-\frac{\tau R^2}{\alpha_1}-\frac{\tau R^2}{\alpha_2}\geq-\psi
\end{aligned}\tag{3.43}$$

Define

$$\psi=\frac{\epsilon^2}{\alpha_1\tau}+\frac{\epsilon^2}{\alpha_2\tau}+\frac{\tau R^2}{\alpha_1}+\frac{\tau R^2}{\alpha_2}+(c_6+\eta)\sigma_g^2+(c_5+\eta)\sigma_f^2+c_7\bar{\sigma}_f^2+c_8\bar{\sigma}_g^2\geq 0,\tag{3.44}$$

then we know if the following is true, we will have linear convergence with bias ψ:

$$\begin{aligned}
&\frac{1}{\beta}-\rho_1\geq 0,\ \frac{1}{\alpha_1}\mathbf{I}-\frac{1}{\rho_1}\mathbf{A}^T\mathbf{A}-L_f^2\rho_4\mathbf{I}-c_1\mathbf{I}-c_5\mathbf{I}-L_f^2\rho_5\mathbf{I}\succeq\mathbf{0},\\
&\frac{1}{\alpha_2}\mathbf{I}-\beta\mathbf{B}^T\mathbf{B}-L_g^2\rho_2\mathbf{I}-c_2\mathbf{I}-c_6\mathbf{I}-L_g^2\rho_3\mathbf{I}\succeq\mathbf{0},\\
&2v_g\mathbf{I}-\frac{1}{\rho_2}\mathbf{I}-\frac{1}{\rho_3}\mathbf{I}-c_4\mathbf{I}-\frac{\gamma}{\alpha_2}\mathbf{I}+\gamma\beta\mathbf{B}^T\mathbf{B}\succeq\mathbf{0},\\
&2v_f-\frac{1}{\rho_4}-\frac{1}{\rho_5}-c_3-\frac{\gamma}{\alpha_1}\geq 0,\ 1-\frac{\gamma}{\beta}\geq 0
\end{aligned}$$

To summarize, we have

$$\begin{aligned}
\alpha_1&=\frac{1}{\frac{1}{\rho_1}\max\sigma^2(\mathbf{A})+L_f^2(\rho_4+\rho_5)+c_1+c_5},\ \alpha_2=\frac{1}{\beta\max\sigma^2(\mathbf{B})+L_g^2(\rho_2+\rho_3)+c_2+c_6}\\
\gamma&=\min(\frac{2v_f-\frac{1}{\rho_4}-\frac{1}{\rho_5}-c_3}{\frac{1}{\rho_1}\max\sigma^2(\mathbf{A})+L_f^2(\rho_4+\rho_5)+c_1+c_5},\frac{2v_g-\frac{1}{\rho_2}-\frac{1}{\rho_3}-c_4}{L_g^2(\rho_2+\rho_3)+c_2+c_6},\frac{1}{\beta})
\end{aligned}$$

Then we can have the following result

$$\mathbb{E}(\|\mathbf{w}^k - \mathbf{w}^*\|^2) \geq (1+\gamma)\mathbb{E}(\|\mathbf{w}^{k+1} - \mathbf{w}^*\|^2) - \psi \tag{3.45}$$

■

3.4.2 Proof of Theorem 1

Proof. Now that we have (3.17), we have

$$\mathbb{E}(\|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) \leq r\mathbb{E}(\|\mathbf{w}^{(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) + r\psi,$$

where $0 < r = \frac{1}{1+\gamma} < 1$. Based on this and triangle inequality we have

$$\begin{aligned}
&\mathbb{E}(\|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) \leq r\mathbb{E}(\|\mathbf{w}^{(k-1)} - \mathbf{w}^{*,(k-1)} + \mathbf{w}^{*,(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) + r\psi \\
&\leq r\mathbb{E}(\|\mathbf{w}^{(k-1)} - \mathbf{w}^{*,(k-1)}\|_{\mathbf{G}}) + r\mathbb{E}(\|\mathbf{w}^{*,(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) + r\psi \\
&\leq r(r\mathbb{E}(\|\mathbf{w}^{(k-2)} - \mathbf{w}^{*,(k-2)}\|_{\mathbf{G}}) + r\mathbb{E}(\|\mathbf{w}^{*,(k-2)} - \mathbf{w}^{*,(k-1)}\|_{\mathbf{G}}) + r\psi) \\
&+ r\mathbb{E}(\|\mathbf{w}^{*,(k-1)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) + r\psi \\
&\cdots \\
&\leq r^k\mathbb{E}(\|\mathbf{w}^{(0)} - \mathbf{w}^{*,(0)}\|_{\mathbf{G}}) + \sum_{i=1}^{k} r^{k-i+1}\mathbb{E}(\|\mathbf{w}^{*,(i-1)} - \mathbf{w}^{*,(i)}\|_{\mathbf{G}}) + \sum_{i=1}^{k} r^{k-i+1}\psi.
\end{aligned}$$

From out assumption we know $\mathbb{E}(\|\mathbf{w}^{*,(i-1)} - \mathbf{w}^{*,(i)}\|_{\mathbf{G}}) \leq \sigma_{\mathbf{w}}$ Taking $k \to +\infty$, we can derive

$$\begin{aligned}
&\lim_{k\to\infty} \mathbb{E}(\|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) \\
&\leq \lim_{k\to\infty} \left(\frac{r(1-r^k)}{1-r}\sigma_{\mathbf{w}} + r^k\mathbb{E}(\|\mathbf{w}^{(0)} - \mathbf{w}^{*,(0)}\|_{\mathbf{G}}) + \frac{r(1-r^k)}{1-r}\psi \right) \\
&\Rightarrow \limsup_{k\to\infty} \mathbb{E}(\|\mathbf{w}^{(k)} - \mathbf{w}^{*,(k)}\|_{\mathbf{G}}) \leq \frac{r}{1-r}(\sigma_{\mathbf{w}} + \psi).
\end{aligned}$$

The desired result is obtained. ■

CHAPTER 4. DISTRIBUTED CONTROLLERS SEEKING AC OPTIMAL POWER FLOW SOLUTIONS USING ADMM

4.1 Introduction

This chapter focuses on power distribution systems with inverter-interfaced renewable energy sources (RESs), and develops a distributed control framework to steer the RES output powers to solutions of AC optimal power flow (OPF) problems. This problem is a special case of time-varying optimization, where random error is added to static case problem. The design of distributed control algorithm is based on suitable linear approximation of the AC power-flow equations, and leverages the so-called alternating direction method of multipliers (ADMM). Convergence of the RES-inverter output powers to solutions of the approximate AC OPF problem is established under suitable conditions on the mismatches between the commanded setpoints and actual RES output powers. Overall, since the proposed scheme can be cast as an ADMM with inexact primal and dual updates, the convergence results can be applied to more general distributed optimization settings. The overarching objective of this chapter is to leverage the flexibility offered by power-electronics-interfaced RESs to address reliability and power-quality concerns that emerge from reverse power flows and renewable generation volatility [57, 58]. Similar to e.g., [59–62], the general control strategy involves a continuous update of the RES setpoints based on current output powers and given OPF objectives (e.g., ensuring voltage regulation, minimization of power losses, as well as maximization of economic benefits to utility and end users).

Prior works in context include [63], wherein feedback control architectures that seek Karush-Kuhn-Tucker (KKT) optimality conditions for economic dispatch in transmission systems are developed, and [64], where a heuristic comprising continuous-time dual ascent and discrete-time reference-signal updates is proposed; local stability of the resultant closed-loop system is also established in [64]. A feedback control algorithm for a finite-horizon economic dispatch problem for distributed energy resources is also considered in [65]. Focusing on AC OPF models, a continuous-time saddle-point-flow method is utilized in [66];

however, stability analysis is available only for specific optimization settings. A reactive power control strategy is proposed in [67] for single-phase distribution systems with a tree topology based on the so-called extremum-seeking control method. Stochastic dual-subgradient solvers are developed in [68] to achieve the solutions of ergodic OPF formulations, based on exact and approximate grid models. An online AC OPF algorithm is proposed in [60] for distribution systems with a tree topology. A controller for a number of resources in general microgrid and distribution-system settings is developed in [61, 62], based on gradient-steering algorithms; the algorithm in [61, 62] is composable in the sense that subsystems can be aggregated into virtual devices that hide their internal complexity, it accounts for errors in the implementable power setpoints, and the average setpoints are provably convergent (on average) to the minimum of the considered control objective. A dual-subgradient method is utilized in [59] to develop feedback controllers that drive the RES output powers to solutions of convex surrogates of the AC OPF; convergence results are available for diminishing stepsize rules in the dual subgradient. A feedback control strategy is proposed in [12] to track solutions of time-varying OPF solutions based on primal-dual methods applied to a modified Lagrangian function.

A key contribution of the present paper consists in leveraging the so-called Alternating Direction Method of Multipliers (ADMM) [69] to develop distributed controllers that pursue solutions of the AC OPF problem. The choice of ADMM is motivated by its favorable scalability with respect to the system size as well as the superior convergence properties compared to subgradient methods [70, 71]. For instance, while convergence results are available for the control scheme in [59] only for for diminishing stepsize rules, the ADMM-based framework proposed here allows one to utilize a constant stepsize, which is desirable for practical implementations. Q-linear convergence is achieved in the gradient-based method proposed in [12], but at the cost of perturbing the optimal solution of the underlying AC OPF. To facilitate the design of computationally affordable ADMM-based controllers, the paper leverages appropriate linear approximations of the AC power-flow equations [49, 72–75]. Based on this linear approximation, two distinct control strategies are developed to trade off convergence speed for computational complexity: in the first strategy, the update of the optimization variables that are proxies for voltage magnitudes is performed by solving a linearly-constrained quadratic program, whereas a simpler projected gradient step is involved in the second setting. In both cases, convergence of the RES-inverter output powers is established under suitable conditions on the stepsize and responsiveness of the RES inverters to power commands. The algorithms afford a distributed solution where both the distribution system operator (DSO) and RES-owners pursue given performance objectives, while ensuring that system operational constraints are observed.

The resultant control framework is close in spirit to the feedback-control strategies proposed in, e.g., [59, 60, 62], where RES setpoints are continuously updated based on current output powers, given OPF objectives, as well as relevant voltage constraints; however, compared to [62] and [60], the proposed framework does not resort to relaxations (e.g., barrier functions) to enforce voltage limits. Further, while [60] is applicable to single-phase radial systems, the method proposed here is applicable to multi-phase settings. Compared to [59], the proposed method requires less stringent assumptions on the mismatch between commanded setpoints and current system outputs and offers improved convergence properties.

Overall, the paper offers the following contributions:

- Online algorithms that pursue solutions of AC OPF problems are designed by leveraging (and suitably adapting) the ADMM;
- Two different algorithmic solutions are proposed to trade-off convergence for computational complexity;
- Convergence of ADMM with inexact primal and dual updates is established. To the best of our knowledge, this is a unique contribution in the broader optimization literature.

Some preliminary results were presented in [76].

4.2 Problem Formulation

4.2.1 Notation

Throughout the paper, $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ denote the real and imaginary parts of a complex number, respectively; for given vector $\mathbf{x}$, $\mathrm{diag}(\mathbf{x})$ denotes a diagonal matrix with diagonal entries composed of the components of $\mathbf{x}$; $j := \sqrt{-1}$. Notation $\|\mathbf{x}\|$ denotes the ℓ_2 norm of $\mathbf{x}$. For column vectors $\mathbf{x}, \mathbf{y}$,

$[\mathbf{x}; \mathbf{y}] := [\mathbf{x}^\top, \mathbf{y}^\top]^\top$; For a given matrix $\mathbf{X}$, vector $\mathbf{X}(i)$ denotes the ith row of $\mathbf{X}$. For given matrix $\mathbf{D}$ and vector $\mathbf{z}$, $\|\mathbf{z}\|_{\mathbf{D}}^2 = \mathbf{z}^\top \mathbf{D}\mathbf{z}$.

4.2.2 Problem setup

Consider modeling the dynamics of the output-powers of the RES inverters through the following general dynamical model [59, 77, 78]:

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}_i(\mathbf{x}_i(t), \mathbf{u}_i(t)), \tag{4.1a}$$

$$\mathbf{y}_i(t) = \mathbf{r}_i(\mathbf{x}_i(t)), \tag{4.1b}$$

where:

- $\mathbf{x}_i(t) := [P_i(t), Q_i(t)]^\mathsf{T}$, with $P_i(t)$ and $Q_i(t)$ denoting the active and reactive output powers (averaged over one AC cycle) of the RES inverter i;
- $\mathbf{u}_i(t) := [\bar{P}_i(t), \bar{Q}_i(t)]^\mathsf{T}$ collects the *commanded* active and reactive powers (i.e., power setpoints);
- $\mathbf{f}_i : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}^2$ and $\mathbf{r}_i : \mathbb{R}^2 \to \mathbb{R}^2$ are arbitrary (non)linear functions; and,
- $\mathbf{y}_i(t)$ is a measurement of $\mathbf{x}_i(t)$ collected at time t.

These dynamics capture the behavior of primal-level controllers embedded into the RES inverters [78]. For a given power setpoint, the following is assumed regarding the regulation capabilities of the primal-level controllers [64, 77]:

Assumption 7. *For a given power setpoint* $\mathbf{u}_i$*,* (4.1) *is asymptotically stable and the equilibrium point* $\mathbf{x}_i$ *satisfies:*

$$\mathbf{0} = \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i), \quad \mathbf{u}_i = \mathbf{r}_i(\mathbf{x}_i). \tag{4.2}$$

This assumption captures the operation of existing devices, where the primary-controllers are designed so that the output powers are regulated to the commanded powers $\mathbf{x}_i$, provided the commanded powers are feasible [78].

Regarding the electrical system, consider a distribution network with $N + 1$ nodes collected in the set $\mathcal{N} = \{0\} \cup \mathcal{N}_D \cup \mathcal{N}_O$, where 0 denotes the secondary of the step-down transformer, and $\mathcal{N}_D, \mathcal{N}_O$ denote the set of locations with and without RESs, respectively. Let $\mathbf{Y}_{\text{net}} \in \mathbb{C}^{(N+1)\times(N+1)}$ denote the network admittance matrix, which is formed according to the system topology and π-equivalent model of the distribution lines. Define the vector $\mathbf{i} := [I_1, \ldots, I_N]^\top \in \mathbb{C}^N$, where I_n denotes the phasor of the current injected at node n. Let $\mathbf{v} := [V_1, \ldots, V_N]^\top \in \mathbb{C}^N$, where $V_i = |V_i|\angle\theta_i \in \mathbb{C}$ denotes the voltage phasor at node i, where $V_0 e^{j\theta_0}$ is the slack-bus voltage with V_0 denoting the voltage magnitude. Let $\bar{P}_i + j\bar{Q}_i$ denote the setpoints of RES $i \in \mathcal{N}_D$, and define $\mathbf{u}_i := [\bar{P}_i, \bar{Q}_i]^\top$ for brevity. Similarly, let $P_{l,i} + jQ_{l,i}$ denote the non-controllable complex load at node $i \in \mathcal{N}$ and $\mathbf{d}_i := [P_{l,i}, Q_{l,i}]^\top$. Based on Kirchhoff's Current Law and Ohm's Law, we can establish the following linear relationship:

$$\begin{bmatrix} I_0 \\ \mathbf{i} \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{y} & \bar{\mathbf{y}}^\top \\ \bar{\mathbf{y}} & \mathbf{Y} \end{bmatrix}}_{\mathbf{Y}_{\text{net}}} \begin{bmatrix} V_0 e^{j\theta_0} \\ \mathbf{v} \end{bmatrix}, \tag{4.3}$$

where $\mathbf{Y}_{\text{net}}$ is partitioned as $\bar{\mathbf{y}} \in \mathbb{C}^N$, $\mathbf{Y} \in \mathbb{C}^{N\times N}$, and $\tilde{y} \in \mathbb{C}\backslash\{0\}$.

Consider then the following prototypical OPF formulation to optimize the steady-state operation of the distribution network:

$$\min_{\mathbf{v},\mathbf{i},\mathbf{u}_i} H(\mathbf{v}) + \sum_{i\in\mathcal{N}_D} G_i(\mathbf{u}_i) \tag{OPF}$$

$$\text{s.t.} \quad \mathbf{i} = \mathbf{Y}\mathbf{v} + \bar{\mathbf{y}} V_0 e^{j\theta_0}, \tag{4.4a}$$

$$V_i I_i^* = \bar{P}_i - P_{l,i} + j(\bar{Q}_i - Q_{l,i}), \ \forall i \in \mathcal{N}_D \tag{4.4b}$$

$$V_n I_n^* = -P_{l,n} - jQ_{l,n}, \qquad \forall n \in \mathcal{N}_O \tag{4.4c}$$

$$V^{\min} \leq |V_i| \leq V^{\max}, \qquad \forall i \in \mathcal{N} \tag{4.4d}$$

$$\mathbf{u}_i = [\bar{P}_i, \bar{Q}_i]^\top \in \mathcal{Y}_i \qquad \forall i \in \mathcal{N}, \tag{4.4e}$$

where $P_{l,i} + jQ_{l,i}$ denotes the loads at node i; (4.4b) and (4.4c) describe power-balance equations for nodes with and without RES inverters, respectively; $V^{\min}$ and $V^{\max}$ are prescribed voltage limits (e.g., ANSI C84.1 limits); the function $H(\mathbf{v}) : \mathbb{C}^N \to \mathbb{R}$ captures network-oriented performance objectives;

$G_i(\mathbf{u}_i) : \mathbb{R}^2 \rightarrow \mathbb{R}$ models optimization objectives at the RES side (e.g., minimization of real power curtailed and reactive power provisioning). Finally, the set $\mathcal{Y}_i \subset \mathbb{R}^2$ models hardware and operational constraints of the inverter i. For example, for photovoltaic (PV) systems, $\mathcal{Y}_i$ takes the following form:

$$\mathcal{Y}_i := \{(\bar{P}_i, \bar{Q}_i) : P_i^{\text{min}} \leq \bar{P}_i \leq P_i^{\text{av}}, \bar{P}_i^2 + \bar{Q}_i^2 \leq S_i^2\} \tag{4.5}$$

where $P_i^{\text{av}} \geq 0$ denotes the available real power, and S_i is the inverter capacity.

Problem (4.4) defines the optimal operating setpoints $\mathbf{u}_i = [\bar{P}_i, \bar{Q}_i]^\mathsf{T}$ of the RES inverter i in terms of commanded inputs and, based on Assumption 1, of the steady-state output powers. However, problem (4.4) is a nonconvex and NP hard problem in general [79]. Recently, convex relaxation methods have been explored to solve the OPF with reduced computational burden, while possibly retaining globally optimal solutions [80]. In this paper, to facilitate the design of low-complexity controllers that can be implemented on microcontrollers that accompany power-electronics interfaces of gateways and inverters, this paper leverages suitable linear approximations of (4.4) [49, 74, 75]. In particular, a linearization approach proposed in [75] is utilized, which is briefly discussed in the next section.

Remark. For ease of exposition, the problem formulation is tailored to the case where one RES is connected at each of the nodes $\mathcal{N}_D$; however, the proposed algorithm can be utilized in settings where RES aggregations are present at (some of) the nodes.

4.2.3 Leveraging approximate linear models

By plugging (4.4a) into (4.4b)-(4.4c), the power-balance equations can be rewritten as:

$$\mathbf{s} = \text{diag}(\mathbf{v})\mathbf{i}^* = \text{diag}(\mathbf{v})(\mathbf{Y}^*\mathbf{v}^* + \bar{\mathbf{y}}^* V_0 e^{-j\theta_0}), \tag{4.6}$$

where $\mathbf{s}$ is a vector collecting the net complex power injections throughout the network. Consider then re-writing the voltages $\mathbf{v}$ satisfying the nonlinear power-balance equations (4.6) as $\mathbf{v} = \mathbf{v}_{\text{nom}} + \mathbf{v}_d$, where $\mathbf{v}_{\text{nom}} = |\mathbf{v}_{\text{nom}}| \angle \boldsymbol{\theta}_{\text{nom}} \in \mathbb{C}^N$ is a predefined nominal voltage profile and $\mathbf{v}_d$ captures deviations around $\mathbf{v}_{\text{nom}}$. Similar to [75], consider further setting $\mathbf{v}_{\text{nom}}$ as $\mathbf{v}_{\text{nom}} = -\mathbf{Y}^{-1}\bar{\mathbf{y}} V_0 e^{j\theta_0}$, which corresponds to the voltage across the network with zero current injections (however, other linearization points can be utilized).

Then, by plugging $\mathbf{v}_{\text{nom}}$ into (4.6) and neglecting the second-order terms (in $\mathbf{v}_d$), we obtain the following expression:

$$\mathbf{v}_d = \mathbf{Y}^{-1}\text{diag}\left(\frac{1}{\mathbf{v}^*_{\text{nom}}}\right)\mathbf{s}^*. \tag{4.7}$$

After expanding (4.7), one can readily derive expressions for the real and the imaginary parts of $\mathbf{v}_d$ separately; however, the resulting expression will couple the components of $\mathbf{p}$ and $\mathbf{q}$, thus challenging the design of computationally-affordable distributed algorithms. To bypass this hurdle, consider rearranging terms to arrive at the following equivalent expression:

$$\text{diag}(\mathbf{v}^*_{\text{nom}})\mathbf{Y}\mathbf{v}_d = \mathbf{s}^*. \tag{4.8}$$

Define $\mathbf{Y} := \mathbf{G} + j\mathbf{B}$, where $\mathbf{G} \in \mathbb{R}^{N\times N}$ is the conductance matrix and $\mathbf{B} \in \mathbb{R}^{N\times N}$ is the susceptance matrix. Further, let $\mathbf{M} := \text{diag}(|\mathbf{v}_{\text{nom}}|\cos\boldsymbol{\theta}_{\text{nom}})$ and $\mathbf{N} := \text{diag}(|\mathbf{v}_{\text{nom}}|\sin\boldsymbol{\theta}_{\text{nom}})$. By expanding (4.8), the following expressions can be obtained:

$$(\mathbf{MG} + \mathbf{NB})\text{Re}(\mathbf{v}_d) - (\mathbf{MB} - \mathbf{NG})\text{Im}(\mathbf{v}_d) = \mathbf{p} \tag{4.9a}$$

$$-(\mathbf{MG} + \mathbf{NB})\text{Im}(\mathbf{v}_d) - (\mathbf{MB} - \mathbf{NG})\text{Re}(\mathbf{v}_d) = \mathbf{q} \tag{4.9b}$$

where the components of vectors $\mathbf{p}$ and $\mathbf{q}$ are defined as: $p_i = \bar{P}_i - P_{l,i}$ and $q_i = \bar{Q}_i - Q_{l,i}$ for $i \in \mathcal{N}_D$; whereas, $p_i = -P_{l,i}$ and $q_i = -Q_{l,i}$ for $i \in \mathcal{N}_O$. Clearly the expression for $\mathbf{p}$ and $\mathbf{q}$ are decoupled. For notational simplicity, define the vector $\boldsymbol{\Delta} := [\text{Re}(\mathbf{v}_d); \text{Im}(\mathbf{v}_d)] \in \mathbb{R}^{2N}$.

Based on these definitions, and noticing that $|\mathbf{v}_{\text{nom}}| + \text{Re}\{\mathbf{v}_d\}$ serves as a first-order approximation to the voltage magnitudes across the distribution network whenever the entries of $\mathbf{v}_{\text{nom}}$ dominate $\mathbf{v}_d$, a convex surrogate of the OPF problem can be formulated as:

$$\min_{\Delta,\mathbf{u}_i} H(\boldsymbol{\Delta}) + \sum_{i\in\mathcal{N}_D} G_i(\mathbf{u}_i) \tag{OPF-2}$$

$$\text{s.t.} \quad \mathbf{C}(i)\boldsymbol{\Delta} - \bar{P}_i + P_{l,i} = 0, \quad i \in \mathcal{N}\backslash\{0\} \tag{4.10a}$$

$$\mathbf{D}(i)\boldsymbol{\Delta} - \bar{Q}_i + Q_{l,i} = 0, \quad i \in \mathcal{N}\backslash\{0\} \tag{4.10b}$$

$$\boldsymbol{\Delta} \in \mathcal{V},\ \mathbf{u}_i = [\bar{P}_i, \bar{Q}_i]^\top \in \mathcal{Y}_i.$$

where $\bar{P}_i = \bar{Q}_i = 0$ for nodes $i \in \mathcal{N}_O$ and:

$$\mathbf{C} := \Big(\mathbf{MG} + \mathbf{NB}, -\mathbf{MB} + \mathbf{NG}\Big) \in \mathbb{R}^{N \times 2N} \tag{4.11a}$$

$$\mathbf{D} := \Big(-\mathbf{MB} + \mathbf{NG}, -\mathbf{MG} - \mathbf{NB}\Big) \in \mathbb{R}^{N \times 2N}. \tag{4.11b}$$

The set $\mathcal{V}$ is designed to enforce voltage regulation as [74]:

$$\mathcal{V} := \{\boldsymbol{\Delta} \mid V^{\min} - |\mathbf{v}_{\text{nom,i}}| \leq \Delta_i \leq V^{\max} - |\mathbf{v}_{\text{nom,i}}|, \\ i = 1, \dots, N\}.$$

For notational simplicity, denote $\boldsymbol{\Phi}_i = [\mathbf{C}(i); \mathbf{D}(i)] \in \mathbb{R}^{2 \times 2N}$, $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1; \cdots; \boldsymbol{\Phi}_N] \in \mathbb{R}^{2N \times 2N}$, and $\mathbf{d}_i = [P_{l,i}, Q_{l,i}]^\mathsf{T}$. Then, (4.10) can be rewritten in the following compact form:

$$\min_{\Delta, \mathbf{u}_i} H(\boldsymbol{\Delta}) + \sum_{i \in \mathcal{N}_D} G_i(\mathbf{u}_i) \tag{OPF-3}$$

$$\text{s.t.} \quad \boldsymbol{\Phi}_i \boldsymbol{\Delta} - \mathbf{u}_i + \mathbf{d}_i = \mathbf{0}, i \in \mathcal{N} \backslash \{0\}, \tag{4.12a}$$

$$\boldsymbol{\Delta} \in \mathcal{V}, \; \mathbf{u}_i \in \mathcal{Y}_i. \tag{4.12b}$$

4.3 Design of Online OPF Solvers

The objective is to design a distributed control scheme that steers the RES-inverter setpoints $\{\mathbf{u}_i \in \mathcal{Y}_i\}_{i=1}^N$ (and, thus, the output powers $\{\mathbf{y}_i(t)\}_{i=1}^N$) to the solution of the OPF problem (4.12). A brief overview of ADMM-based algorithms is outlined next; the ADMM-based control architecture is then discussed in Section 4.3.2.

4.3.1 Open-loop ADMM-based distributed optimization

Consider the following augmented Lagrangian function associated with problem (4.12):

$$\mathcal{L}(\boldsymbol{\Delta}, \{\mathbf{u}_i\}, \{\boldsymbol{\lambda}_i\}) := H(\boldsymbol{\Delta}) + \sum_{i \in \mathcal{N}_D} G_i(\mathbf{u}_i) + \frac{\rho}{2} \sum_{i \in \mathcal{N} \backslash \{0\}} \left\| \boldsymbol{\Phi}_i \boldsymbol{\Delta} - \mathbf{u}_i + \mathbf{d}_i + \frac{\boldsymbol{\lambda}_i}{\rho} \right\|^2, \tag{4.13}$$

where $\boldsymbol{\lambda}_i \in \mathbb{R}^N$ is the Lagrangian multiplier associated with the linear constraint (4.12a), $\rho > 0$ is a design parameter, and $\mathbf{u}_i = \mathbf{0}$ for $i \in \mathcal{N}_O$. ADMM involves an iterative procedure where the following steps are performed at each iteration k:

$$\boldsymbol{\Delta}^k = \arg\min_{\boldsymbol{\Delta}\in\mathcal{V}} H(\boldsymbol{\Delta}) + \sum_{i\in\mathcal{N}\backslash\{0\}} \frac{\rho}{2}\left\|\boldsymbol{\Phi}_i\boldsymbol{\Delta} - \mathbf{u}_i^{k-1} + \mathbf{d}_i - \boldsymbol{\lambda}_i^{k-1}/\rho\right\|^2, \tag{4.14a}$$

$$\boldsymbol{\lambda}_i^k = \boldsymbol{\lambda}_i^{k-1} - \rho(\boldsymbol{\Phi}_i\boldsymbol{\Delta}^k - \mathbf{u}_i^{k-1} + \mathbf{d}_i), \tag{4.14b}$$

$$\mathbf{u}_i^k = \arg\min_{\mathbf{u}_i\in\mathcal{Y}_i} G_i(\mathbf{u}_i) + \frac{\rho}{2}\left\|\boldsymbol{\Phi}_i\boldsymbol{\Delta}^k - \mathbf{u}_i + \mathbf{d}_i - \boldsymbol{\lambda}_i^k/\rho\right\|^2. \tag{4.14c}$$

One way to reduce the computational complexity associated with the update of the voltage-related vector $\boldsymbol{\Delta}^k$ is to consider solving the following quadratic approximation:

$$\boldsymbol{\Delta}^k = \arg\min_{\boldsymbol{\Delta}\in\mathcal{V}} \langle \mathbf{g}^{k-1}, \boldsymbol{\Delta}-\boldsymbol{\Delta}^{k-1}\rangle + \frac{L}{2}\left\|\boldsymbol{\Delta}-\boldsymbol{\Delta}^{k-1}\right\|^2, \tag{4.15}$$

where $L > 0$ is a design parameter, and $\mathbf{g}^{k-1}$ denotes the gradient of the augmented Lagrangian function with respect to $\boldsymbol{\Delta}$; particularly, $\mathbf{g}^{k-1}$ is given by:

$$\mathbf{g}^{k-1} = \nabla H(\boldsymbol{\Delta}^{k-1}) + \sum_{i\in\mathcal{N}_D} \boldsymbol{\Phi}_i^\top\left(\boldsymbol{\Phi}_i\boldsymbol{\Delta}^{k-1} - \mathbf{u}_i^{k-1} + \mathbf{d}_i + \boldsymbol{\lambda}_i^{k-1}/\rho\right). \tag{4.16}$$

It can be verified that the optimal solution of (4.15) amounts to a projected-gradient step in the following form:

$$\boldsymbol{\Delta}^k = \mathcal{P}_{\mathcal{V}}(\boldsymbol{\Delta}^{k-1} - \frac{1}{L}\mathbf{g}^{k-1}), \tag{4.17}$$

where $\mathcal{P}_{\mathcal{V}}$ denotes the projection operation onto the set $\mathcal{V}$.

The steps described above lead to a distributed procedure that is provably convergent to a solution of (4.12); the distributed algorithm is tabulated as Algorithm 1.

Algorithm 1 ADMM-based algorithm

```
1: Perform (4.14a) with two options:
       • Option 1: Solve (4.14a).
       • Option 2: Perform (4.17).
2: Perform (4.14b).
3: Perform (4.14c).
```

However, one drawback of Algorithm 1 is that the setpoints $\mathbf{u}_i$ can be commanded to the RES inverters only *upon convergence*. On the other hand, sending the setpoints to the RES inverters at each intermediate

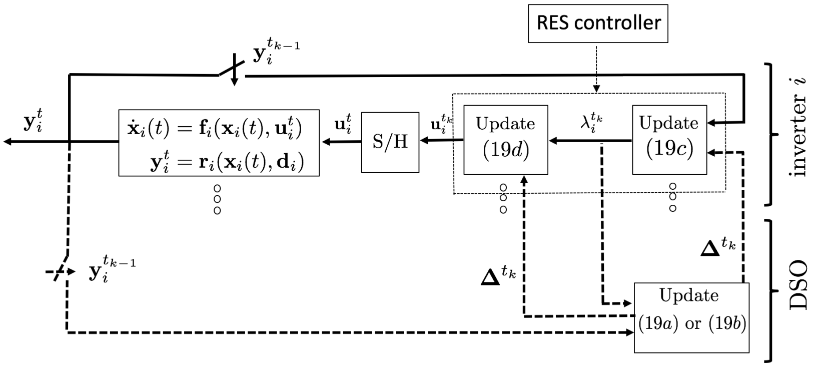


Figure 4.1: Illustration of the distributed framework (see also Algorithm 2). Steps. (4.19c) and (4.19d) constitute the RES controller and they generate a discrete time signal $\mathbf{u}_i^{t_k}$, which is applied to the inverter by utilizing a sample-and-hold unit. The inverter output is then sampled and utilized to update the control signals.

step k leads to an *open-loop* procedure where no actionable feedback from the electrical system is utilized; for instance, steps 2 and 3 of Algorithm 1 would utilize the commanded inputs $\{\mathbf{u}_i^k\}_{i \in \mathcal{N}_D}$, which may not necessarily coincide with the actual outputs powers of the RES inverters (commanded setpoints and output powers coincide only after a given settling time of the primary controllers of the inverters). To capture non-idealities of existing devices (which may not respond quickly to changes in the setpoints) as well as discrepancies between the input setpoints and the power outputs due to faulty estimations of the maximum available powers from the RESs, the next section will develop a control scheme that dynamically update the setpoints of the devices based on current system outputs and problem parameters. The setting is close in spirit to the feedback-control strategies proposed in e.g., [59, 60, 62]. Compared to [62] and [60], the proposed framework does not resort to barrier-type functions to enforce voltage limits and is applicable to multi-phase settings; the contribution over [59] consists in considering less stringent assumptions on the mismatch between the commanded setpoints and current system outputs, and improved convergence properties.

4.3.2 From open-loop optimization to feedback-control

Similar to, e.g., [59, 60, 62], consider updates performed at discrete time instants $t \in \{t_k, k \in \mathbb{N}\}$. At time t_k, let $\mathbf{u}^{t_k} = \{\mathbf{u}_i^{t_k}\}_{i\in\mathcal{N}\setminus\{0\}}$, $\boldsymbol{\Delta}^{t_k}$ and $\boldsymbol{\lambda}^{t_k} := \{\boldsymbol{\lambda}_i^{t_k}\}_{i\in\mathcal{N}\setminus\{0\}}$ denote the primal and dual variables, respectively. At time t_{k-1}, the RES outputs are sampled as [cf. Fig. 4.1]:

$$\mathbf{y}_i^{t_{k-1}} = \mathbf{r}_i(\mathbf{x}_i(t_{k-1}), \mathbf{d}_i), \forall\, i \in \mathcal{N}_D. \tag{4.18}$$

The measured output powers are then utilized to update the voltage-related vector $\boldsymbol{\Delta}$ and the dual variables as follows:

$$\texttt{Option 1}:\boldsymbol{\Delta}^{t_k} = \arg\min_{\boldsymbol{\Delta}\in\mathcal{V}}\; H(\boldsymbol{\Delta}) + \sum_{i\in\mathcal{N}\setminus\{0\}} \frac{\rho}{2}\left\|\boldsymbol{\Phi}_i\boldsymbol{\Delta} - \mathbf{y}_i^{t_{k-1}} + \mathbf{d}_i - \frac{\boldsymbol{\lambda}_i^{t_{k-1}}}{\rho}\right\|^2, \tag{4.19a}$$

$$\texttt{Option 2}:\boldsymbol{\Delta}^{t_k} = \mathcal{P}_{\mathcal{V}}\left(\boldsymbol{\Delta}^{t_{k-1}} - \frac{1}{L}\mathbf{g}^{t_{k-1}}\right), \tag{4.19b}$$

$$\boldsymbol{\lambda}_i^{t_k} = \boldsymbol{\lambda}_i^{t_{k-1}} - \rho(\boldsymbol{\Phi}_i\boldsymbol{\Delta}^{t_k} - \mathbf{y}_i^{t_{k-1}} + \mathbf{d}_i). \tag{4.19c}$$

$$\mathbf{u}_i^{t_k} = \arg\min_{\mathbf{u}_i\in\mathcal{Y}_i}\; G_i(\mathbf{u}_i) + \frac{\rho}{2}\left\|\boldsymbol{\Phi}_i\boldsymbol{\Delta}^{t_k} - \mathbf{u}_i + \mathbf{d}_i - \frac{\boldsymbol{\lambda}_i^{t_k}}{\rho}\right\|^2. \tag{4.19d}$$

Note that (4.19d) produces the commands to the RES inverter (4.1) [cf. Fig. 4.1]. Different from (4.17), in Option 2 the gradient vector $\mathbf{g}^{t_{k-1}}$ is evaluated at $\mathbf{y}_i^{t_{k-1}}$, i.e.,

$$\mathbf{g}^{t_{k-1}} = \nabla H(\boldsymbol{\Delta}^{t_{k-1}}) + \sum_{i\in\mathcal{N}_D} \boldsymbol{\Phi}_i^\top\left(\boldsymbol{\Phi}_i\boldsymbol{\Delta}^{t_{k-1}} - \mathbf{y}_i^{t_{k-1}} + \mathbf{d}_i + \boldsymbol{\lambda}_i^{k-1}/\rho\right).$$

In summary, the controllers perform the steps tabulated as Algorithm 2.

Algorithm 2 ADMM-based OPF Controllers

```
1: At time $t_{k-1}$, RES outputs are sampled as (4.18).
2: Perform (4.19a) or (4.19b).
3: Perform (4.19c) using the sampled RES outputs.
4: Perform (4.19d) and apply the resulting signal to (4.1a) during $(t_{k-1}, t_k]$, i.e. $\mathbf{u}_i(t) = \mathbf{u}^{t_k}, t \in (t_{k-1}, t_k]$.
5: When $t = t_k$ go to step 1.
```

The algorithm (4.19) affords a distributed implementation. With reference to the illustrative diagram in Fig. 4.1, one possible distributed solution involves the following steps:

i) update (4.19a) or (4.19b) can be performed at the DSO, after receiving $\mathbf{y}_i^{t_{k-1}}$ and $\boldsymbol{\lambda}_i^{t_{k-1}}$ from each RES i;

ii) the DSO subsequently broadcasts to the RESs the vector Δ^{t_k};

iii) updates (4.19c) and (4.19d) are then performed locally at each individual RES $i \in \mathcal{N} \backslash \{0\}$. These steps are computationally light and, when $G_i(\mathbf{u}_i)$ is linear or quadratic and $\mathcal{Y}_i$ is given as in (4.5), $\mathbf{u}_i^k$ admits a closed-form solution; see e.g., [12, Appendix B].

It is worth reiterating that the key differences compared to the open-loop optimization strategy (4.14) are: 1) the setpoints are commanded to the RES inverters at each time instant t_k (whereas Algorithm 1 produces setpoints only upon convergence of the ADMM); and, ii) measurements of the RES-inverter output-powers are used in the updates. The (continuous-time) reference signals $\{\mathbf{u}_i(t)\}_{i \in \mathcal{N}_D}$ produced by the controller have step changes at instants $\{t_k, k \in \mathbb{N}\}$ and are left-continuous functions that take the constant values $\{\mathbf{u}_i^{t_k}\}_{i \in \mathcal{N}_D}$ over the time interval $(t_{k-1}, t_k]$. When the interval $(t_{k-1}, t_k]$ is longer than the settling time of (4.1), then the RES output powers converge to the intermediate setpoints $\{\mathbf{u}_i^{t_k}\}_{i=1}^N$ at each iteration; that is, $\lim_{t \to t_k^-} \|\mathbf{y}_i^t - \mathbf{u}_i^{t_k}\| = 0$. Hence, (4.14) and (4.19) coincide, and the well-known convergence claims for the ADMM naturally apply to the present setup [69]. However, in case of slow-responding inverters, or, when the updates (4.19) can be performed faster than the systems' settling times, then one has that the inverter outputs may not coincide with the commanded setpoints; particularly, define the *error term* $\boldsymbol{\eta}_i^{t_k} = \mathbf{u}_i^{t_k} - \mathbf{y}_i^{t_k}$, $i \in \mathcal{N}_D$ to quantify this discrepancy. In the following, convergence of the RES output powers when $\boldsymbol{\eta}_i^{t_k} \neq \mathbf{0}$ is analyzed.

4.3.3 Convergence Analysis

Algorithm 2 can be interpreted as a variation of the ADMM with inexact primal and dual updates. To the best of our knowledge, convergence of the ADMM in this setting is not available in the prior literature. This paper considers the following two types of updates:

1. Exact minimization in the primal steps using RES output $\{\mathbf{y}_i^{t_k}\}$ [i.e., `option 1`];

2. Gradient steps are performed in the primal steps using RES output $\{\mathbf{y}_i^{t_k}\}$ [i.e., `option 2`].

In the remainder of this section, convergence of Option 2 is studied; in fact, Option 1 can be analyzed using similar techniques, but with considerably simpler steps. To simplify the notation, the superscript t_k is hereafter dropped.

Define $\eta^{t_k} := \|\mathbf{u}^{t_k} - \mathbf{y}^{t_k}\|$, and consider the following assumption.

Assumption 8. *The gradient stepsize $\frac{1}{L} > 0$ satisfies the following property:*

$$(L - \gamma)\mathbf{I}_{2N} - \rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi} \succcurlyeq 0, \tag{4.20}$$

where $\mathbf{I}_{2N}$ is the $2N \times 2N$ identity matrix, and γ denotes the Lipschitz constant of $\nabla H(\boldsymbol{\Delta})$, i.e. $\|\nabla H(\mathbf{x}) - \nabla H(\mathbf{y})\| \geq \gamma\|\mathbf{x} - \mathbf{y}\|$, for all $\mathbf{x}, \mathbf{y} \in$ dom H.

Further, assume that

$$\sum_{k=1}^{\infty} \eta^{t_k} < \infty.$$

Assumption 8 asserts that, for given loading and ambient conditions, the discrepancy between the commanded inputs and the output powers should diminish as the system reaches the AC OPF solution. From Assumption 2, it is clear that L has to be greater than or equal to the largest eigenvalue of the Hessian matrix of the augmented Lagrangian function (for a fixed ρ). In fact, from Assumption 8 it follows that:

$$L\mathbf{I}_{2N} \succcurlyeq \gamma\mathbf{I}_{2N} + \rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi}.$$

Since γ is the Lipschitz constant of $\nabla H(\boldsymbol{\Delta})$, it follows that $\gamma\mathbf{I}_{2N} \succcurlyeq \nabla^2 H(\boldsymbol{\Delta})$, and hence:

$$L\mathbf{I}_{2N} \succcurlyeq \nabla^2 H(\boldsymbol{\Delta}) + \rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi},$$

where the right-hand-side is the Hessian matrix of the augmented Lagrangian function.

To facilitate analysis, define the vectors $\mathbf{w}^{t_k} := [\mathbf{u}^{t_k}; \boldsymbol{\Delta}^{t_k}; \boldsymbol{\lambda}^{t_k}]$ and $\hat{\mathbf{w}}^{t_k} := [\hat{\mathbf{u}}^{t_k}; \hat{\boldsymbol{\Delta}}^{t_k}; \hat{\boldsymbol{\lambda}}^{t_k}]$, where $\mathbf{w}^{t_k}$ is the update generated by (4.19) (with possibly nonzero error terms $\boldsymbol{\eta}_i^{t_k} = \mathbf{u}_i^{t_k} - \mathbf{y}_i^{t_k}$), and $\hat{\mathbf{w}}^{t_k}$ is generated by the same iteration, but with zero error (i.e., $\mathbf{0} = \mathbf{u}_i^{t_k} - \mathbf{y}_i^{t_k}$). Henceforth, $\hat{\mathbf{w}}^{t_k}$ is referred to as the "error-free" iterates. Let W^* be the optimal set of (4.12), which is nonempty, closed, and convex.

The ultimate goal of this analysis is to prove the convergence of iterates $\mathbf{w}^{t_k}$. For the purpose of readibility, we first state the main result of this paper in the following theorem:

Theorem 3. *Suppose that Assumption 4 and 8 hold true. Then the sequence* $\mathbf{w}^{t_k}$ *generated by* (4.19b)-(4.19d) *converges to some* $\mathbf{w}^{\infty} \in W^*$*, where* $\mathbf{w}^{\infty}$ *is a cluster point of the sequence* $\{\mathbf{w}^{t_k}\}$.

To prove this result, we start from the optimality condition. Denote the set of optimality conditions of $\mathbf{w}$ as

$$D(\mathbf{w}) := \begin{pmatrix} \mathbf{u} - \mathcal{P}_{\mathcal{Y}}\{\mathbf{u} - [\nabla G(\mathbf{u}) + \boldsymbol{\lambda}]\} \\ \boldsymbol{\Delta} - \mathcal{P}_{\mathcal{V}}\{\boldsymbol{\Delta} - [\nabla H(\boldsymbol{\Delta}) - \boldsymbol{\Phi}^{\top}\boldsymbol{\lambda}]\} \\ \boldsymbol{\Phi}\boldsymbol{\Delta} - \mathbf{u} + \mathbf{d} \end{pmatrix},$$

$$\text{dist}(\mathbf{w}, W^*) := \min\{\|\mathbf{w} - \mathbf{z}^*\| \mid \mathbf{z}^* \in W^*\}.$$

Then it is easy to verify that the following holds:

$$\text{dist}(\mathbf{w}, W^*) = 0 \Leftrightarrow D(\mathbf{w}) = 0. \tag{4.21}$$

Lemma 2. *There exists a constant* $\tau > 0$*, such that*

$$\left\|D(\hat{\mathbf{w}}^{t_k})\right\|^2 \leq \tau \cdot \left\|\boldsymbol{\Phi}\boldsymbol{\Delta}^{t_{k-1}} - \hat{\mathbf{u}}^{t_k} + \mathbf{d}\right\|^2_{\rho\mathbf{I}}, \; \forall\, k \geq 1. \tag{4.22}$$

To show convergence, the right-hand-side of (4.22) needs to approach 0 when $k \to \infty$; this result is provided by the following two lemmas.

Lemma 3. *Let* $\mathbf{w}^* := [\mathbf{u}^*; \boldsymbol{\Delta}^*; \boldsymbol{\lambda}^*]$ *be an optimal solution of* (4.12)*. Then the following inequality holds*

$$\begin{aligned} \left\|\hat{\mathbf{w}}^{t_k} - \mathbf{w}^*\right\|^2_{\tilde{\mathbf{H}}} \leq & \left\|\mathbf{w}^{t_{k-1}} - \mathbf{w}^*\right\|^2_{\tilde{\mathbf{H}}} - \left\|\boldsymbol{\Phi}\boldsymbol{\Delta}^{t_{k-1}} - \hat{\mathbf{u}}^{t_k} + \mathbf{d}\right\|^2_{\rho\mathbf{I}} \\ & - \left\|\hat{\boldsymbol{\Delta}}^{t_k} - \boldsymbol{\Delta}^{t_{k-1}}\right\|^2_{\boldsymbol{\Psi}}, \end{aligned} \tag{4.23}$$

where $\tilde{\mathbf{H}} := \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & L\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{1}{\rho}\mathbf{I} \end{pmatrix}$ *and* $\boldsymbol{\Psi} := (L - \gamma)\mathbf{I} - \rho\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}$.

Lemma 3 establishes a relationship between the exact and inexact updates in terms of the distance to an optimal solution.

Lemma 4. *Let* $\mathbf{w}^{t_k} = [\mathbf{u}^{t_k}; \boldsymbol{\Delta}^{t_k}; \boldsymbol{\lambda}^{t_k}]$ *and* $\hat{\mathbf{w}}^{t_k} = [\hat{\mathbf{u}}^{t_k}; \hat{\boldsymbol{\Delta}}^{t_k}; \hat{\boldsymbol{\lambda}}^{t_k}]$ *be the inexact and exact iterates, respectively. We have the following limit*

$$\lim_{k\to\infty} \left\|\boldsymbol{\Phi}\boldsymbol{\Delta}^{t_{k-1}} - \hat{\mathbf{u}}^{t_k} + \mathbf{d}\right\|_{\rho\mathbf{I}}^2 = 0. \tag{4.24}$$

The proofs of Theorem 3 and Lemma 3 are presented in the Appendix.

4.3.4 Relaxing the requirements on the RES outputs

In this section, Assumption 4 is relaxed to consider cases where the error sequence $\{\eta^{t_k}\}$ is no longer diminishing; this case captures scenarios where 1) the primary controllers have a steady-state regulation error (i.e., $\mathbf{y}^{t_k} \neq \mathbf{u}^{t_k}$) and/or 2) irradiance and load conditions are very fast changing.

To facilitate the design of distributed controllers in this setting, key is to consider a modified version of (4.12) where the subproblem solved to update $\boldsymbol{\Delta}$ is unconstrained (this particular problem structure will ensure convergence of the algorithm developed in this section). For notational simplicity, let $\text{LB}(j) \leq 0$ and $\text{UB}(j) \geq 0$ be the lower and upper bounds for Δ_j, respectively; that is:

$$\text{LB}(j) \leq \Delta_j \leq \text{UB}(j), \quad j = 1, \ldots, 2N.$$

From (4.12), it follows that $\boldsymbol{\Phi}\boldsymbol{\Delta} + \mathbf{d} = \mathbf{u}$, the following approximation can be utilized to express the voltage-regulation constraints as linear functions of $\mathbf{u}_i$:

$$\sum_{j=1}^{2N} |\boldsymbol{\Phi}_{i,j}| \cdot \text{LB}(j) + \mathbf{d}_i \leq \mathbf{u}_i \leq \sum_{j=1}^{2N} |\boldsymbol{\Phi}_{i,j}| \cdot \text{UB}(j) + \mathbf{d}_i. \tag{4.25}$$

We refer to the feasible set defined by (4.25) as $\mathcal{V}'$. Using (4.25), the approximate OPF problem becomes:

$$\min_{\Delta, \mathbf{u}_i} H(\boldsymbol{\Delta}) + \sum_{i \in \mathcal{N}_D} G_i(\mathbf{u}_i) \tag{4.26a}$$

$$\text{s.t.} \quad \boldsymbol{\Phi}_i \boldsymbol{\Delta} - \mathbf{u}_i + \mathbf{d}_i = \mathbf{0}, i \in \mathcal{N} \backslash \{0\}, \tag{4.26b}$$

$$\mathbf{u}_i \in \mathcal{Y}_i \cap \mathcal{V}'. \tag{4.26c}$$

The algorithm (4.19) can be slightly modified to accommodate (4.26); particularly, the projection onto $\mathcal{V}$ in (4.19a) should be removed and a projection onto $\mathcal{V}'$ should be added in (4.19d). The resultant algorithm can be used to solve (4.26). We refer to this algorithm as Algorithm 3.

Consider then the following assumption.

Assumption 9. *Functions $H(\mathbf{\Delta})$ and $G(\mathbf{u})$ are strongly convex, $\mathbf{\Phi} := [\mathbf{\Phi}_1; \cdots; \mathbf{\Phi}_N] \in \mathbb{R}^{2N \times 2N}$ is full rank, and ∇H is Lipschitz continuous.*

Based on this assumption, one can leverage the results of [70, Theorem 3.4] to obtain the following.

Corollary 1. *If Assumption 9 holds, the iterates $\mathbf{w}^{t_k}$ generated by Algorithm 3 to solve the approximated problem (4.26) and "error-free" iterates $\hat{\mathbf{w}}^{t_k}$ satisfy the following inequality:*

$$\begin{aligned}\left\|\hat{\mathbf{w}}^{t_k} - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 \le (1-\delta)\left\|\mathbf{w}^{t_{k-1}} - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 \\ - \left\|\hat{\mathbf{w}}^{t_k} - \mathbf{w}^{t_{k-1}}\right\|_{\tilde{\mathbf{H}}}^2,\end{aligned} \tag{4.27}$$

where $\delta \in (0, 1)$ is some positive constant.

Corollary 1 can be proved by following steps similar to [70, Theorem 3.4]. The main convergence results are established next.

Theorem 4. *Suppose that Assumption 9 holds and that there exists a constant ϵ such that*

$$\left\|\mathbf{y}^{t_k} - \mathbf{u}^{t_k}\right\| \le \epsilon. \tag{4.28}$$

Then, the sequence $\mathbf{w}^{t_k}$ generated by (4.19) to solve problem (4.26) satisfies $\left\|\mathbf{w}^{t_k} - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 \le (\beta + \xi)^2$, as $k \to \infty$, where

$$\xi = \epsilon\sqrt{(1-\delta)(1+\theta) \times \frac{r}{1-r}}, \quad \beta = \epsilon\sqrt{L\|\mathbf{\Phi}^\top\|^2 + \rho},$$

are both constants and $\theta = \frac{1-\delta}{r-1+\delta} > 0,\ 0 < r < 1$ is some constant.

The theorem asserts that, if $\left\|\mathbf{y}^{t_k} - \mathbf{u}^{t_k}\right\| \le \epsilon$ for all k (which reflects the actual operation of some existing inverters), then the algorithm will converge to a ball centered around the optimal solution set of (4.26).

4.4 Numerical experiment

Numerical results are provided to corroborate the analytical findings and demonstrate the efficacy of the proposed method. Consider the modified version of the IEEE 37-node test feeder taken from [59]; see also Fig. 4.2. In the OPF problem, the voltage limits are set to $V^{\min} = 0.95$ pu and $V^{\max} = 1.05$pu, whereas $V_0 = 1 + \mathrm{j}0$ pu. With reference to the node numbering utilized in [59], assume that there are 6 PV systems located at nodes $4, 11, 22, 26, 29$, and 32, and assume that the primary controllers of the PV systems are modeled as a first-order system [78]. The following ratings are assumed: $\{S_i\}_{i\in\mathcal{N}_D} = \{100, 240, 100, 200, 240, 160\}$ kVA; Further, $\theta = \frac{\pi}{2}$, $P_i^{\min} = 0$, and the objective functions are set to:

$$H(\boldsymbol{\Delta}) = 10 \times \sum_{i=1}^{N} (\boldsymbol{\Delta}(i) - 1)^2, \tag{4.29}$$

$$\begin{aligned} G_i(P_i, Q_i) =& a_i(P_i^{\mathrm{av}} - P_i)^2 + b_i(P_i^{\mathrm{av}} - P_i) \\ & + c_i Q_i^2 + d_i|Q_i|, \end{aligned} \tag{4.30}$$

where $H(\boldsymbol{\Delta})$ promotes a flat voltage profile, and $G_i(P_i, Q_i)$ penalizes real power curtailment and limits the amount of reactive power provided. As an example, the coefficients in (4.30) are chosen as $a_i = 1, b_i = 10, c_i = 0.01, d_i = 0.01$ for $i = 1, \ldots, 4$ and $a_i = 1, b_i = 10, c_i = 0.03, d_i = 0.03$ for $i = 5, 6$. For the ADMM-type methods, the following quantities are utilized to measure the optimality of the solutions [14]:

$$\|r_p^k\| = \|\mathbf{C}\boldsymbol{\Delta}^{t_k} - \mathbf{p}^{t_k} + \mathbf{p}_l\|, \; \|r_q^k\| = \|\mathbf{D}\boldsymbol{\Delta}^{t_k} - \mathbf{q}^{t_k} + \mathbf{q}_l\|$$
$$\|s_p^k\| = \|\mathbf{C}(\boldsymbol{\Delta}^{t_k} - \boldsymbol{\Delta}^{t_{k-1}})\|, \; \|s_q^k\| = \|\mathbf{D}(\boldsymbol{\Delta}^{t_k} - \boldsymbol{\Delta}^{t_{k-1}})\|.$$

and, for given load and ambient conditions, the algorithm terminates when quantities above are smaller than 5×10^{-4}; for the dual-subgradient method, only the first 300 iterations are plotted.

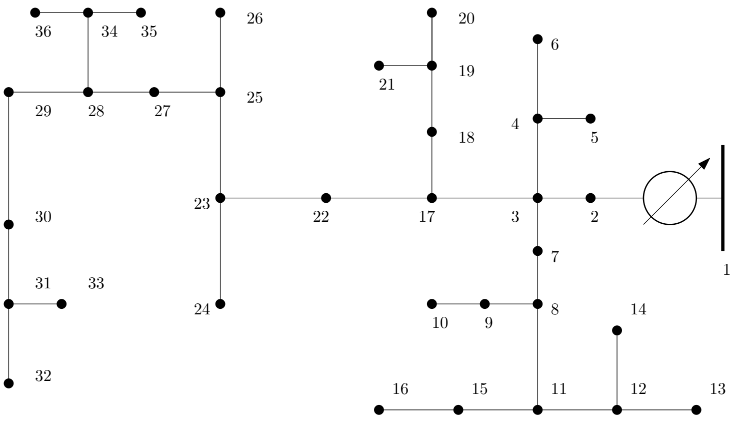
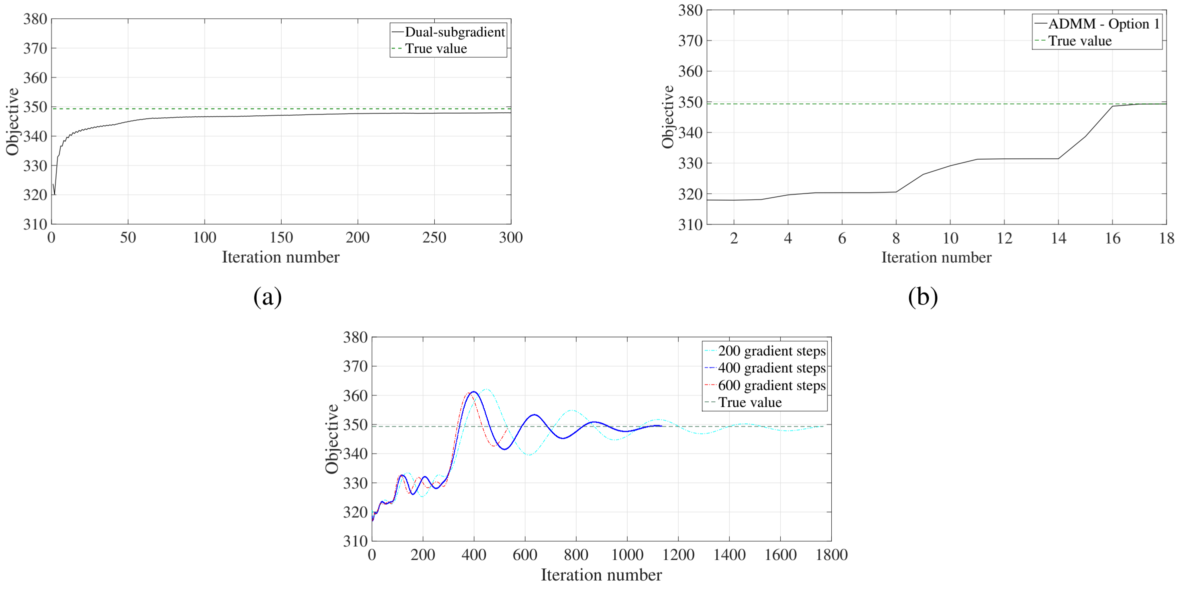


Figure 4.2: IEEE 37-node feeder.



(a)

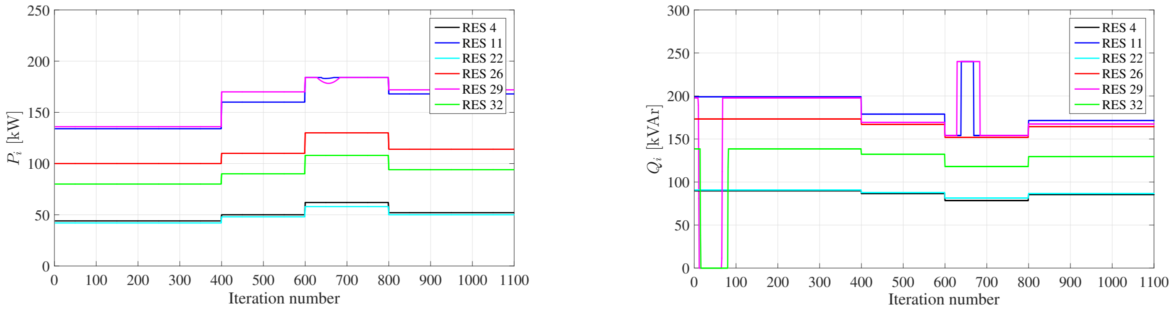
(b)

(c)

Figure 4.4: Convergence of the dual-subgradient method as well as the ADMM-based algorithm. For the latter, both Option 1 and Option 2 are tested. A first-order system is used to emulate the behavior of the RES system. As a benchmark, CVX [1] is utilized to obtain the optimal solution of (4.12). Figure 4.3c illustrates the trade-off between the total number of iterations and the number of gradient steps used for each iteration

Fig. 4.4 shows that the ADMM-based algorithm (with either Option 1 or Option 2) converges to the optimal objective value. Dual-subgradient methods (e.g., [59]) are also convergent, but they require a significantly higher number of iterations. Each iterations of the dual-subgradient method and of the ADMM-Option 1 take a similar computational time since they both solve the $\boldsymbol{\Delta}$-subproblem exactly. Notice that compared to Option 1, Option 2 requires more iterations to converge; however, each iteration is computationally lighter for Option 2. This sets a natural trade-off between convergence and computational complexity. To further highlight this point, Figure 4.3c compares a few different scenarios in which *multiple* gradient steps (4.19b) are performed in each iteration. Clearly, the higher is the amount of gradient steps performed in each iteration, the fewer is the total iterations are required.

Next, adaptability of the proposed ADMM-based strategy to changing irradiance conditions is tested; particularly, assume the following changes in the available powers $\{P_i^{\text{av}}\}$ of the PV systems:



(a) (b)

Figure 4.6: Tracking performance of the proposed algorithm in case of changing operational conditions. Changes in the solar irradiance are presumed at $k = 400, 600,\ 800$.

$$P^{\text{av}}(k) = [44, 134, 42, 100, 136, 80]^\top \text{kW}, k \in [1, 400]$$

$$P^{\text{av}}(k) = [50, 160, 48, 110, 170, 90]^\top \text{kW}, k \in [400, 600]$$

$$P^{\text{av}}(k) = [62, 184, 58, 135, 184, 108]^\top \text{kW}, k \in [600, 800]$$

$$P^{\text{av}}(k) = [52, 168, 50, 114, 172, 94]^\top \text{kW}, k \in [800, 1200].$$

Note that the changes are presupposed at iterations 400, 600 and 800. It can be seen from Fig. 4.6 that the inverter outputs $\mathbf{y}_i[t_k] = [P_i(t), Q_i(t)]^\top$ quickly converge to the new optimal setpoints within each interval.

To assess whether the proposed scheme enforces voltage regulation, consider a setting where the PV capacities $\{S_i\}_{i \in \mathcal{N}_D}$ and available powers P^{av} are five time higher than the initial setting. In this case, the feeder would incur overvoltage conditions when the PV systems operate at the business-as-usual setpoint $(P^{\text{av}}, 0)$. Instead, Fig. 4.7 demonstrates that the ADMM-based controller maintains the voltage magnitude within the limits.

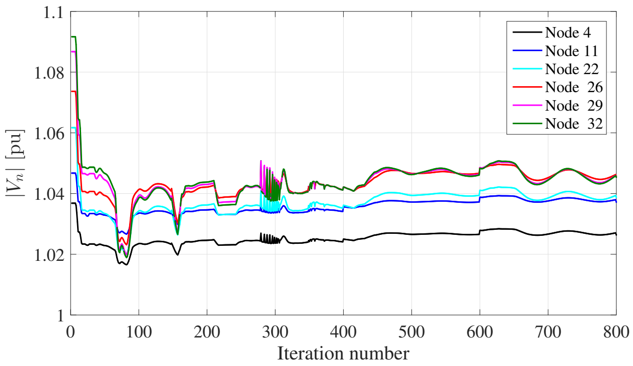


Figure 4.7: Example of trajectory for the voltage magnitude when the PV systems are controlled using the proposed ADMM-based algorithm.

Finally, the ADMM-based Algorithm 3 is tested in the presence of constant error; particularly, it is assumed that $(\eta^k)^2 = \|\mathbf{y}_i^{t_k} - \mathbf{u}_i^{t_k}\|^2 = 0.002$. The following objective functions are utilized:

$$H(\boldsymbol{\Delta}) = 10 \times \sum_{i=1}^{2N} (\boldsymbol{\Delta}(i) - 1)^2, \tag{4.31}$$

$$\begin{aligned} G_i(P_i, Q_i) =& a_i(P_i^{\text{av}} - P_i)^2 + b_i(P_i^{\text{av}} - P_i) \\ &+ c_i Q_i^2 + d_i |Q_i|. \end{aligned} \tag{4.32}$$

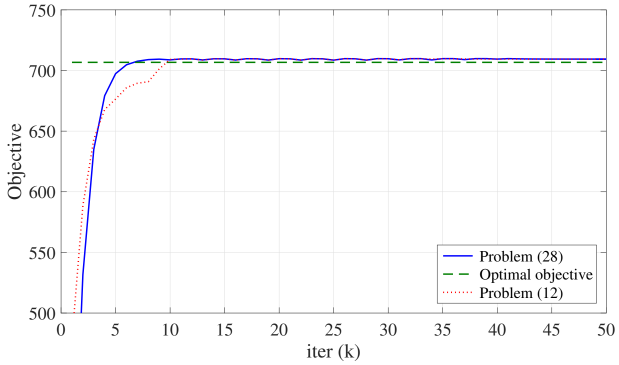
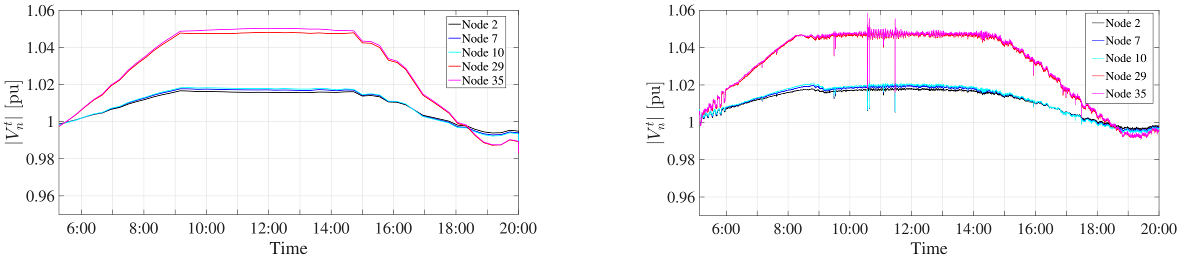


Figure 4.8: Convergence of ADMM with constant error in RES outputs. The two curves are generated by running (4.19) to solve problem (4.12), and by running Algorithm 3 to solve problem (4.26), respectively.



(a)

(b)

Figure 4.10: Voltage profiles when $\mathbf{d}_i$ is changing every second.

where the parameters a_i, b_i, c_i, and b_i are set as in the previous experiments. In Figure 4.8 it can be seen that the approximation error is negligible and, as established in Theorem 4, the algorithm converges to a neighborhood of the optimal value. In the figure, the trajectory ”solving problem (12)” is used for comparison purposes, and it is generated by running (4.19) to solve the original problem (4.12).

The performance of ADMM-based algorithms depends on the tuning parameter ρ. For `Option` 1, we use the adaptive stepsize strategy explained in [14] to improve the convergence. In the `Option` 2), ρ is chosen empirically and it is set to 10^2.

The theoretical results outlined in the paper are applicable to the case where the non-controllable loads $\mathbf{d}_i$ are slow time-varying or constant. While extending the theoretical claims to the case of time-varying loads, constraints, and cost functions is the subject of future endeavors, in this section we provide some numerical results to show how the proposed ADMM-based algorithm can cope with time-varying problem parameters. To this end, we consider the simulation setting utilized in [81], where the the loads $\mathbf{d}_i$ and the maximum active powers available from the PV inverters are changing on a second basis; see [81] for a complete description of the dataset. Figure 4.9a reports the evolution of the voltage magnitude over time when CVX [1] is utilized to solve problem (4.12); to obtain reasonable simulation times, (4.12) was solved with CVX only every 1000 seconds. As for Algorithm 3, three iterations are performed every second. Figure 4.9b illustrates the trajectory of the voltage magnitudes; it can be seen that the ADMM-based method can successfully enforce voltage regulation and tracks the benchmark trajectories.

4.5 Proof of Lemma 1

Proof. Define first the following:

$$D(\hat{\mathbf{w}}^k) = \begin{pmatrix} \hat{\mathbf{u}}^k - \mathcal{P}_{\mathcal{Y}}\{\hat{\mathbf{u}}^k - [\nabla G(\hat{\mathbf{u}}^k) + \hat{\boldsymbol{\lambda}}^k]\} \\ \hat{\boldsymbol{\Delta}}^k - \mathcal{P}_{\mathcal{V}}\{\hat{\boldsymbol{\Delta}}^k - [\nabla H(\hat{\boldsymbol{\Delta}}^k) - \boldsymbol{\Phi}^\top \hat{\boldsymbol{\lambda}}^k]\} \\ \boldsymbol{\Phi}\hat{\boldsymbol{\Delta}}^k - \hat{\mathbf{u}}^k + \mathbf{d} \end{pmatrix}$$

and notice that the optimality condition on $\hat{\boldsymbol{\Delta}}^k$ and $\hat{\mathbf{u}}^k$ imply the following:

$$\begin{aligned} \hat{\boldsymbol{\Delta}}^k &= \mathcal{P}_{\mathcal{V}}\{\hat{\boldsymbol{\Delta}}^k - [\nabla H(\hat{\boldsymbol{\Delta}}^k) - \boldsymbol{\Phi}^\top \hat{\boldsymbol{\lambda}}^{k-1} \\ &\quad + \rho \boldsymbol{\Phi}^\top (\boldsymbol{\Phi}\hat{\boldsymbol{\Delta}}^k - \hat{\mathbf{u}}^k + \mathbf{d})]\} \\ &= \mathcal{P}_{\mathcal{V}}\{\hat{\boldsymbol{\Delta}}^k - [\nabla H(\hat{\boldsymbol{\Delta}}^k) - \boldsymbol{\Phi}^\top \hat{\boldsymbol{\lambda}}^k]\} \\ \hat{\mathbf{u}}^k &= \mathcal{P}_{\mathcal{Y}}\{\hat{\mathbf{u}}^k - [\nabla G(\hat{\mathbf{u}}^k) + \hat{\boldsymbol{\lambda}}^k + \rho \boldsymbol{\Phi}(\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k)]\}. \end{aligned}$$

Combining the above equalities and using nonexpansive property of the projection operator, it follows that:

$$\begin{aligned} \left\| D(\hat{\mathbf{w}}^k) \right\| &\leq \left\| \begin{matrix} \rho \boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}) \\ \mathbf{0} \\ \boldsymbol{\Phi}\boldsymbol{\Delta}^{k-1} - \hat{\mathbf{u}}^k + \mathbf{d} + \boldsymbol{\Phi}\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Phi}\boldsymbol{\Delta}^{k-1} \end{matrix} \right\| \\ &\leq (\rho + 1) \left\| \boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}) \right\| + \left\| \boldsymbol{\Phi}\boldsymbol{\Delta}^{k-1} - \hat{\mathbf{u}}^k + \mathbf{d} \right\| \end{aligned}$$

Thus, one can conclude that:

$$\left\| D(\hat{\mathbf{w}}^k) \right\|^2 \leq \tau \left\| \boldsymbol{\Phi}\boldsymbol{\Delta}^{k-1} - \hat{\mathbf{u}}^k + \mathbf{d} \right\|_{\rho \mathbf{I}}^2,$$

where $\tau > 0$ is some constant. ■

4.6 Proof of Lemma 2

Proof. Based on the following equality

$$\|\mathbf{a} + \mathbf{b}\|^2 = \|\mathbf{a}\|^2 - \|\mathbf{b}\|^2 + 2(\mathbf{a} + \mathbf{b})^\top \mathbf{b} \tag{4.33}$$

we have that:

$$\begin{aligned}
&\left\|\hat{\mathbf{w}}^k - \mathbf{w}^*\right\|^2_{\tilde{\mathbf{H}}} = \left\|\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^*\right\|^2_{L\mathbf{I}} + \left\|\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^*\right\|^2_{\frac{1}{\rho}\mathbf{I}} \\
=&\left\|\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1} + \boldsymbol{\Delta}^{k-1} - \boldsymbol{\Delta}^*\right\|^2_{L\mathbf{I}} + \left\|\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1} + \boldsymbol{\lambda}^{k-1} - \boldsymbol{\lambda}^*\right\|^2_{\frac{1}{\rho}\mathbf{I}} \\
\overset{(4.33)}{=}&\left\|\boldsymbol{\Delta}^{k-1} - \boldsymbol{\Delta}^*\right\|^2_{L\mathbf{I}} + \left\|\boldsymbol{\lambda}^{k-1} - \boldsymbol{\lambda}^*\right\|^2_{\frac{1}{\rho}\mathbf{I}} - \left\|\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}\right\|^2_{L\mathbf{I}} \\
&- \left\|\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1}\right\|^2_{\frac{1}{\rho}\mathbf{I}} + 2L(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^*)^\top(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}) \\
&+ \frac{2}{\rho}(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^*)^\top(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1}). \qquad (4.34)
\end{aligned}$$

We then leverage the convergence results for the standard ADMM, and utilize the optimality condition for $\boldsymbol{\Delta}^*$ and $\hat{\boldsymbol{\Delta}}^k$ as well as the convexity of $H(\cdot)$ and $G(\cdot)$ to bound the cross term in (4.34). For $\forall\boldsymbol{\Delta} \in \mathcal{V}$ and $\forall\mathbf{u} \in \mathcal{Y}$:

$$\begin{aligned}
&H(\boldsymbol{\Delta}) - H(\hat{\boldsymbol{\Delta}}^k) - (\boldsymbol{\Delta} - \hat{\boldsymbol{\Delta}}^k)^\top(\boldsymbol{\Phi}^\top\hat{\boldsymbol{\lambda}}^k + \rho\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Phi}\boldsymbol{\Delta}^{k-1})) \\
&\geq L(\boldsymbol{\Delta} - \hat{\boldsymbol{\Delta}}^k)^\top(\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k) - \frac{\gamma}{2}\left\|\boldsymbol{\Delta}^k - \hat{\boldsymbol{\Delta}}^k\right\|^2 \\
&H(\boldsymbol{\Delta}) - H(\boldsymbol{\Delta}^*) - (\boldsymbol{\Delta} - \boldsymbol{\Delta}^*)^\top(\boldsymbol{\Phi}^\top\boldsymbol{\lambda}^*) \geq 0 \\
&G(\mathbf{u}) - G(\hat{\mathbf{u}}^k) + (\mathbf{u} - \hat{\mathbf{u}}^k)^\top\left[\hat{\boldsymbol{\lambda}}^k - \rho(\boldsymbol{\Phi}\boldsymbol{\Delta}^{k-1} - \boldsymbol{\Phi}\hat{\boldsymbol{\Delta}}^k)\right] \geq 0 \\
&G(\mathbf{u}) - G(\mathbf{u}^*) + (\mathbf{u} - \mathbf{u}^*)^\top\boldsymbol{\lambda}^* \geq 0.
\end{aligned}$$

Using the identification $\boldsymbol{\Delta} = \boldsymbol{\Delta}^*$ and $\boldsymbol{\Delta} = \hat{\boldsymbol{\Delta}}^k$ for the optimality condition of $\boldsymbol{\Delta}^*$ and $\hat{\boldsymbol{\Delta}}^k$, respectively, and doing the same for $\mathbf{u}^*$ and $\hat{\mathbf{u}}^k$, one can obtain the following inequalities:

$$\begin{aligned}
&(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^*)^\top(\boldsymbol{\Phi}^\top(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^*) + \rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1})) \\
\geq& L(\boldsymbol{\Delta}^* - \hat{\boldsymbol{\Delta}}^k)^\top(\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k) - \frac{\gamma}{2}\left\|\boldsymbol{\Delta}^k - \hat{\boldsymbol{\Delta}}^k\right\|^2 \qquad (4.35) \\
&(\mathbf{u}^* - \hat{\mathbf{u}}^k)^\top(\boldsymbol{\lambda}^* - \hat{\boldsymbol{\lambda}}^k) \\
\leq& \rho(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))^\top(\mathbf{u}^* - \hat{\mathbf{u}}^k). \qquad (4.36)
\end{aligned}$$

thus, adding up (4.35)-(4.36), one has that:

$$\begin{aligned}
&(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^*)^\top(\boldsymbol{\Phi}\boldsymbol{\Delta}^* - \boldsymbol{\Phi}\hat{\boldsymbol{\Delta}}^k + \hat{\mathbf{u}}^k - \mathbf{u}^*)\\
&+ (\boldsymbol{\Delta}^* - \hat{\boldsymbol{\Delta}}^k)^\top(\rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))\\
\leq&\frac{\gamma}{2}\left\|\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k\right\|^2 - L(\boldsymbol{\Delta}^* - \hat{\boldsymbol{\Delta}}^k)^\top(\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k)\\
&+ \rho(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}))^\top(\mathbf{u}^* - \hat{\mathbf{u}}^k).
\end{aligned}$$

Using the dual update of (4.19) in the above inequality, one obtains:

$$\begin{aligned}
&\frac{1}{\rho}(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1})(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^*)\\
\leq&\frac{\gamma}{2}\left\|\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k\right\|^2 + \rho(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))^\top(\mathbf{u}^* - \hat{\mathbf{u}}^{k+1})\\
&+ (\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^*)^\top(L\mathbf{I} - \rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi})(\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k).
\end{aligned}$$

Now we can bound the cross term of (4.34) as follows:

$$\begin{aligned}
&2L(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^*)^\top(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}) + \frac{2}{\rho}(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^*)^\top(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1})\\
\leq&2L(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^*)^\top(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}) + \gamma\left\|\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k\right\|^2\\
&+ (\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^*)^\top(2L\mathbf{I} - 2\rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi})(\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k)\\
&+ 2\rho(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))^\top(\mathbf{u}^* - \hat{\mathbf{u}}^k)\\
=&\gamma\left\|\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k\right\|^2 + (\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^*)^\top(2\rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi})(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1})\\
&+ 2\rho(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))^\top(\mathbf{u}^* - \hat{\mathbf{u}}^k)\\
=&\gamma\left\|\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k\right\|^2 + 2\rho(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))^\top(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^*)\\
&+ \mathbf{u}^* - \hat{\mathbf{u}}^k)\\
=&\gamma\left\|\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k\right\|^2 + 2\rho(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))^\top(\boldsymbol{\Phi}\hat{\boldsymbol{\Delta}}^k - \hat{\mathbf{u}}^k + \mathbf{d})\\
=&\gamma\left\|\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k\right\|^2 - 2(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))^\top(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1}).
\end{aligned} \tag{4.37}$$

Combine (4.34)-(4.37) and define $\boldsymbol{\Psi} := (L-\gamma)\mathbf{I} - \rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$. Then, $\left\|\hat{\mathbf{w}}^k - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2$ can be bounded as shown next:

$$\begin{aligned}
\left\|\hat{\mathbf{w}}^k - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 \le{}& \left\|\mathbf{w}^{k-1} - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 - \left\|\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}\right\|_{L\mathbf{I}}^2 \\
&- \left\|\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1}\right\|_{\frac{1}{\rho}\mathbf{I}}^2 + \gamma\left\|\boldsymbol{\Delta}^{k-1} - \hat{\boldsymbol{\Delta}}^k\right\|^2 \\
&- 2(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))^\top(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1}) \\
={}& \left\|\mathbf{w}^{k-1} - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 - \left\|\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}\right\|_{\rho\boldsymbol{\Phi}^\top\boldsymbol{\Phi}}^2 - \left\|\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1}\right\|_{\frac{1}{\rho}\mathbf{I}}^2 \\
&- 2(\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}))^\top(\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1}) - \left\|\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}\right\|_{\boldsymbol{\Psi}}^2 \\
={}& \left\|\mathbf{w}^{k-1} - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 - \left\|\hat{\boldsymbol{\lambda}}^k - \boldsymbol{\lambda}^{k-1} + \rho\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1})\right\|_{\frac{1}{\rho}\mathbf{I}}^2 \\
&- \left\|\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}\right\|_{\boldsymbol{\Psi}}^2 \\
={}& \left\|\mathbf{w}^{k-1} - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 - \left\|\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}\right\|_{\boldsymbol{\Psi}}^2 \\
&- \left\|-\rho(\boldsymbol{\Phi}\hat{\boldsymbol{\Delta}}^k - \hat{\mathbf{u}}^k + \mathbf{d}) + \rho\boldsymbol{\Phi}(\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1})\right\|_{\frac{1}{\rho}\mathbf{I}}^2 \\
={}& \left\|\mathbf{w}^{k-1} - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 - \left\|\boldsymbol{\Phi}\boldsymbol{\Delta}^{k-1} - \hat{\mathbf{u}}^k + \mathbf{d}\right\|_{\rho\mathbf{I}}^2 \\
&- \left\|\hat{\boldsymbol{\Delta}}^k - \boldsymbol{\Delta}^{k-1}\right\|_{\boldsymbol{\Psi}}^2
\end{aligned} \tag{4.38}$$

■

4.7 Proof of Lemma 3

Proof. It can be readily shown that

$$\begin{aligned}
\|\mathbf{w}^{t_k} - \mathbf{w}^*\|_{\tilde{\mathbf{H}}}^2 &= \|\mathbf{w}^{t_k} - \hat{\mathbf{w}}^{t_k} + \hat{\mathbf{w}}^{t_k} - \mathbf{w}^*\|_{\tilde{\mathbf{H}}}^2 \\
&= \|\mathbf{w}^{t_k} - \hat{\mathbf{w}}^{t_k}\|_{\tilde{\mathbf{H}}}^2 + \|\hat{\mathbf{w}}^{t_k} - \mathbf{w}^*\|_{\tilde{\mathbf{H}}}^2 \\
&+ 2\|\mathbf{w}^{t_k} - \hat{\mathbf{w}}^{t_k}\|_{\tilde{\mathbf{H}}} \cdot \|\hat{\mathbf{w}}^{t_k} - \mathbf{w}^*\|_{\tilde{\mathbf{H}}}.
\end{aligned} \tag{4.39}$$

On the other hand, from Lemma 3 it follows that

$$\begin{aligned}\left\|\mathbf{w}^{t_k}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}} &\leq \left\|\hat{\mathbf{w}}^{t_k}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}+\|\mathbf{w}^{t_k}-\hat{\mathbf{w}}^{t_k}\|_{\tilde{\mathbf{H}}}\\ &\leq \left\|\mathbf{w}^{t_{k-1}}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}+\|\mathbf{w}^{t_k}-\hat{\mathbf{w}}^{t_k}\|_{\tilde{\mathbf{H}}}.\end{aligned}$$

According to Hölder's inequality and the fact that there are errors both in Δ and λ updates, we have

$$\begin{aligned}\|\mathbf{w}^{t_k}-\hat{\mathbf{w}}^{t_k}\|_{\tilde{\mathbf{H}}}^2 &= \left\|\begin{pmatrix}0\\ \mathbf{\Delta}^{t_k}-\hat{\mathbf{\Delta}}^{t_k}\\ \lambda^{t_k}-\hat{\lambda}^{t_k}\end{pmatrix}\right\|_{\tilde{\mathbf{H}}}^2 = \left\|\begin{pmatrix}0\\ \mathbf{\Phi}^\top(\mathbf{u}^{t_k}-\mathbf{y}^{t_k})\\ \rho(\mathbf{u}^{t_k}-\mathbf{y}^{t_k})\end{pmatrix}\right\|_{\tilde{\mathbf{H}}}^2\\ &\leq L(\eta^{t_k})^2\|\mathbf{\Phi}^\top\|^2+\rho(\eta^{t_k})^2. \end{aligned}\tag{4.40}$$

Combining the above two inequalities we can derive

$$\left\|\mathbf{w}^{t_k}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}} \leq \left\|\mathbf{w}^{t_{k-1}}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}+\eta^{t_k}\sqrt{L\|\mathbf{\Phi}^\top\|^2+\rho}. \tag{4.41}$$

Summing both sides over k, we obtain

$$\left\|\mathbf{w}^{t_k}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}} \leq \sum_{i=1}^{k}\sigma\eta^{t_i} \tag{4.42}$$

where $\sigma := \sqrt{L\|\mathbf{\Phi}^\top\|^2+\rho}$. The above inequality implies that if $\sum_{k=1}^{\infty}\eta^{t_k} < +\infty$, then $\|\hat{\mathbf{w}}^{t_k}-\mathbf{w}^*\|_{\tilde{\mathbf{H}}} \leq c$, where c is some constant. Consequently, combining (4.39) and (4.42) one can obtain the following:

$$\left\|\mathbf{w}^{t_k}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 \leq \left\|\hat{\mathbf{w}}^{t_k}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2+(\sigma\eta^{t_k})^2+2\sigma\eta^{t_k}c. \tag{4.43}$$

Combining (4.43) with Lemma 3 and Assumption 4, it follows that:

$$\begin{aligned}\left\|\mathbf{w}^{t_k}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 &\leq \left\|\hat{\mathbf{w}}^{t_k}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2+(\sigma\eta^{k})^2+2\sigma\eta^{t_k}c\\ &\leq \left\|\mathbf{w}^{t_{k-1}}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2-\left\|\mathbf{\Phi}\mathbf{\Delta}^{t_{k-1}}-\hat{\mathbf{u}}^{t_k}+\mathbf{d}\right\|_{\rho\mathbf{I}}^2\\ &+(\sigma\eta^{t_k})^2+2\sigma\eta^{t_k}c.\end{aligned}\tag{4.44}$$

Summing (4.44) from 1 to k, we obtain:

$$\begin{aligned}\left\|\mathbf{w}^{t_k}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 \leq \left\|\mathbf{w}^{0}-\mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 &- \sum_{i=1}^{k}\left\|\mathbf{\Phi}\mathbf{\Delta}^{t_{i-1}}-\hat{\mathbf{u}}^{t_i}+\mathbf{d}\right\|_{\rho\mathbf{I}}^2\\ &+\sum_{i=1}^{k}(\sigma\eta^{t_i})^2+2\sum_{i=1}^{k}\sigma\eta^{t_i}c.\end{aligned}\tag{4.45}$$

Further, letting $k \to \infty$ for (4.45), it is clear that $\left\|\mathbf{w}^{t_\infty}- \mathbf{w}^*\right\|^2_{\tilde{\mathbf{H}}}$ and $\left\|\mathbf{w}^0-\mathbf{w}^*\right\|^2_{\tilde{\mathbf{H}}}$ are finite. On the other hand, with Assumption 2 we know that $\sum_{i=1}^{\infty}(\sigma\eta^{t_i})^2$ and $2\sum_{i=1}^{\infty}\sigma\eta^{t_i}c$ are also finite. Then one can show that

$$\sum_{i=1}^{+\infty}\left\|\mathbf{\Phi}\mathbf{\Delta}^{t_{i-1}} - \hat{\mathbf{u}}^{t_i} + \mathbf{d}\right\|^2_{\rho\mathbf{I}} < +\infty,$$

which leads to the following result:

$$\lim_{k\to\infty}\left\|\mathbf{\Phi}\mathbf{\Delta}^{t_{k-1}} - \hat{\mathbf{u}}^{t_k} + \mathbf{d}\right\|^2_{\rho\mathbf{I}} = 0. \tag{4.46}$$

■

4.8 Proof of Theorem 1

Proof. From Lemma 2 and (4.46), it follows that

$$\lim_{k\to\infty} D(\hat{\mathbf{w}}^k) = 0\,.$$

On the other hand, from Assumption 4 and (4.45), one has that $\{\mathbf{w}^k\}$ is bounded and so is $\{\hat{\mathbf{w}}^k\}$. Then, there exists a closed and bounded set S such that $\{\hat{\mathbf{w}}^k\} \subset S$, $\lim_{k\to\infty}\|\hat{\mathbf{w}}^k - \mathbf{w}^k\| = 0$. It remains to show that $\lim_{k\to\infty}\text{dist}(\hat{\mathbf{w}}^k, W^*) = 0$.

Suppose that $\lim_{k\to\infty}\text{dist}(\hat{\mathbf{w}}^k, W^*) \neq 0$. Then, there exists a $\delta > 0$ such that $\lim\sup_{k\to\infty}\text{dist}(\hat{\mathbf{w}}^k, W^*) = \delta > 0$. Further,

$$\{\hat{\mathbf{w}}^k\} \subset S \cap \{\mathbf{z}|\text{dist}(\mathbf{z}, W^*) \geq \frac{\delta}{2}\} \triangleq S_1,$$

and, since $S_1 \cap W^* \neq \emptyset$, then $D(\mathbf{z}) \neq 0$ for many $\mathbf{z} \in S_1$; that is, $\min_{\mathbf{z}\in S}\{\|D(\mathbf{z})\|^2\} = \epsilon > 0$. This contradicts the fact that $\{\hat{\mathbf{w}}^k\} \subset S$ and $\lim_{k\to\infty}\|D(\hat{\mathbf{w}}^k)\| = 0$.

Since $\lim_{k\to\infty}\text{dist}(\mathbf{w}^k, W^*) = 0$, every subsequence of $\{\mathbf{w}^k\}$ converges to an optimal solution. Without loss of generality, let $\mathbf{w}^\infty$ be a cluster point of $\{\mathbf{w}^k\}$, and $\{\mathbf{w}^{k_j}\}$ be the subsequence of $\{\mathbf{w}^k\}$, which converges to $\mathbf{w}^\infty$. For $\mathbf{w}^\infty \in W^*$ and for all $\epsilon > 0$, there exists an integer l such that:

$$\left\|\mathbf{w}^{k_l} - \mathbf{w}^\infty\right\|^2_{\tilde{\mathbf{H}}} < \frac{\epsilon^2}{3},$$

$$\sum_{i=k_l}^{\infty}\sigma\eta_i < \frac{\epsilon^2}{6c}, \quad \sum_{i=k_l}^{\infty}(\sigma\eta_i)^2 < \frac{\epsilon^2}{3}.$$

By (4.44), we have that $\forall k \geq k_l + 1$

$$\begin{aligned}\left\|\mathbf{w}^k - \mathbf{w}^\infty\right\|_{\tilde{\mathbf{H}}}^2 \leq \left\|\mathbf{w}^{k_l} - \mathbf{w}^\infty\right\|_{\tilde{\mathbf{H}}}^2 &+ \sum_{i=k_l}^{k-1} (\sigma\eta_i)^2 + \sum_{i=k_l}^{k-1} 2c\sigma\eta_i \\ &- \sum_{i=k_l}^{k-1} \left\|\mathbf{\Phi}\mathbf{\Delta}^{i-1} - \hat{\mathbf{u}}^i + \mathbf{d}\right\|_{\rho\mathbf{I}}^2 \\ &\leq \frac{\epsilon^2}{3} + \frac{\epsilon^2}{3} + \frac{\epsilon^2}{3} = \epsilon^2\end{aligned}$$

Hence, $\|\mathbf{w}^k - \mathbf{w}^\infty\|_{\tilde{\mathbf{H}}} \to 0$, i.e. $\mathbf{w}^k \to \mathbf{w}^\infty$. ■

4.9 Proof of Theorem 2

Proof. Given underlying assumptions, one can always find a $\theta = \frac{1-\delta}{r-1+\delta} > 0$ such that

$$(1-\delta)(1+\frac{1}{\theta}) = r < 1,$$

where r is some constant. Since $\left\|\mathbf{y}^k - \mathbf{u}^k\right\| \leq \epsilon$, then it holds that

$$\left\|\mathbf{w}^k - \hat{\mathbf{w}}^k\right\|_{\tilde{\mathbf{H}}} \leq \epsilon\sqrt{L\|\mathbf{\Phi}^\top\|^2 + \rho}. \tag{4.47}$$

For simplicity, define $\beta = \epsilon\sqrt{L\|\mathbf{\Phi}^\top\|^2 + \rho}$. From Corollary 1, we have the following inequality:

$$\begin{aligned}\left\|\hat{\mathbf{w}}^k - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 &\leq (1-\delta)(1+\theta)\left\|\mathbf{w}^{k-1} - \hat{\mathbf{w}}^{k-1}\right\|_{\tilde{\mathbf{H}}}^2 \\ &+ (1-\delta)(1+\frac{1}{\theta})\left\|\hat{\mathbf{w}}^{k-1} - \mathbf{w}^*\right\|_{\tilde{\mathbf{H}}}^2 - \left\|\hat{\mathbf{w}}^k - \mathbf{w}^{k-1}\right\|_{\tilde{\mathbf{H}}}^2\end{aligned}$$

and, by applying the same iteration to $\|\hat{\mathbf{w}}^{k-1} - \mathbf{w}^*\|_{\tilde{\mathbf{H}}}^2$, we obtain:

$$
\begin{aligned}
\left\|\hat{\mathbf{w}}^{k}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}}^{2} &\leq(1-\delta)(1+\theta) \beta^{2}+r\left\|\hat{\mathbf{w}}^{k-1}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}}^{2} \\
&-\left\|\hat{\mathbf{w}}^{k}-\mathbf{w}^{k-1}\right\|_{\tilde{\mathbf{H}}}^{2} \\
&\leq(1-\delta)(1+\theta) \beta^{2}+r[(1-\delta)(1+\theta) \beta^{2} \\
&+r\left\|\hat{\mathbf{w}}^{k-2}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}}^{2}-\left\|\hat{\mathbf{w}}^{k-1}-\mathbf{w}^{k-2}\right\|_{\tilde{\mathbf{H}}}^{2}] \\
&-\left\|\hat{\mathbf{w}}^{k}-\mathbf{w}^{k-1}\right\|_{\tilde{\mathbf{H}}}^{2} \\
\ldots & \\
&\leq(1-\delta)(1+\theta) \beta^{2} \sum_{i=0}^{k-1} r^{i} \\
&+r^{k}\left\|\hat{\mathbf{w}}^{k-2}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}}^{2} .
\end{aligned}
$$

Letting $k \rightarrow \infty$, one has that $\lim\limits_{k \rightarrow \infty} \sum\limits_{i=0}^{k-1} r^{i}=\frac{r}{1-r}$ is a constant and

$$
r^{k}\left\|\hat{\mathbf{w}}^{k-2}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}}^{2} \rightarrow 0 .
$$

It thus follow that:

$$
\left\|\hat{\mathbf{w}}^{\infty}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}}^{2} \leq(1-\delta)(1+\theta) \beta^{2} \times \frac{r}{1-r}
$$

Let $\xi^{2}=(1-\delta)(1+\theta) \beta^{2} \times \frac{r}{1-r}$. Since $\|\hat{\mathbf{w}}^{k}-\mathbf{w}^{k}\|_{\tilde{\mathbf{H}}}^{2} \leq \beta^{2}$ is a constant, it follows that

$$
\begin{aligned}
\left\|\mathbf{w}^{\infty}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}}^{2}= & \left\|\mathbf{w}^{\infty}-\hat{\mathbf{w}}^{\infty}+\hat{\mathbf{w}}^{\infty}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}}^{2} \\
= & \left\|\mathbf{w}^{\infty}-\hat{\mathbf{w}}^{\infty}\right\|_{\tilde{\mathbf{H}}}^{2}+\left\|\hat{\mathbf{w}}^{\infty}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}}^{2} \\
& +2\left\|\mathbf{w}^{\infty}-\hat{\mathbf{w}}^{\infty}\right\|_{\tilde{\mathbf{H}}} \cdot\left\|\hat{\mathbf{w}}^{\infty}-\mathbf{w}^{*}\right\|_{\tilde{\mathbf{H}}} \\
\leq & \beta^{2}+\xi^{2}+2 \beta \xi=(\beta+\xi)^{2}
\end{aligned}
$$

■

CHAPTER 5. DSPD: A DOUBLE STOCHASTIC PRIMAL-DUAL ALGORITHM FOR TRAINING TASK

5.1 Introduction

In this chapter, we introduce a novel stochastic primal-dual method for training neural networks. Previous chapters all focus on the tracking ability of our time-varying algorithm, where problem parameters are changing in real time and the goal is to track multiple optimal solutions. In this chapter we formulate the training problem into a related time-varying optimization problem, where each time we sample part of the data as problem parameters. The goal, however, is to learn one solution that can fit all problem parameters. This is a fundamental difference of this chapter.

Deep neural networks (DNNs) have been successfully implemented in many applications in recent years [26]. Owning to the availability of powerful CPU and GPUs, we are able to utilize big datasets to train DNNs as function approximators. A classic feed-forward deep neural network is a hierarchical mapping from inputs to outputs, where each layer consists of linear operators and nonlinear activation functions. Given enough neurons, such a nested system can have arbitrary accuracy in function approximation [82]. However, the nested structure also poses significant challenges in training as it involves minimizing a highly non-convex and possibly nonsmooth loss function. Multiple local optima, saddle points or even flat regions are likely to exist and jeopardize the whole training process [83, 84].

Related Works: Recently, a new line of work focuses on designing training schemes that decouple the connections between DNN layers by introducing auxiliary variables. It is shown that these schemes can be used together with the techniques such as normalization and residual nets to alleviate the vanishing gradient problem. Reference [32] proposes to decouple the layers using auxiliary variables. This resulting equality constrained problem is then solved by penalizing the constraints to the objective and performing alternating updating between variables. The work [33] follows similar ideas and proposes to use alternating direction method of multipliers (ADMM) to solve the resulting equality constrained problem. The authors show that

ADMM has a better scalability than SGD in terms of parallelizing over multiple CPU cores. However, the algorithm suffers from significant drawback, in that the final optimized variables can be inconsistent because the relaxed constraints are never enforced; Further, it is a *batch* algorithm which requires the access of all data points each time, and there is no theoretical guarantee that the algorithm is convergent. It has been stated in [33] that whether these efficient primal-dual training scheme are convergent has been an open problem. Another ADMM-based algorithm is proposed in [34] to train DNNs for supervised learning of hash codes, where only empirical convergence is discussed.

Besides primal-dual type methods, researchers have also discovered that block coordinate descent (BCD) can further boost the performance of SGD. In [35, 85] BCD type methods are shown to be superior over SGD-based algorithms under certain settings. A proximal BCD algorithm was proposed in [86], and global convergence results are provided by assuming certain Kurdyka-Łojasiewicz (KL) property. All the aforementioned works are *batch* algorithms, meaning each gradient evaluation step requires one pass to the entire dataset, which is time consuming, memory inefficient, thus are not realistic in practice. To the best of our knowledge, there is no stochastic primal-dual algorithm or stochastic coordinate descent algorithm that have been developed to train neural networks in the literature. Convergence of such stochastic algorithm also remains unknown. In this paper, we will address these issues.

Our Contributions: Considering all the works mentioned above, we address the following research question: Can we rigorously develop algorithms that *do not require backpropagation*, and are able to effectively and systematically access the gradient information for deeper layers, while still being able to leverage key features of existing training schemes such as stochastic access of the data, incremental updates of the variables? Towards this end, the main contributions of this paper are listed below:

• We propose a novel training framework called double stochastic primal-dual (DSPD) method. The updates of this algorithm are computationally cheap due to the fact that it is based on stochastically chosen subsets of parameters and data points.

• Under suitable constraint qualifications (which are commonly used in optimization literature), we prove that the proposed algorithm can converge to the set of stationary solutions with probability 1;

• We conduct experiments to demonstrate that our stochastic algorithm is able to train neural networks.

Note that by no means we are claiming our algorithm is superior over SGD-based methods or we can solve vanishing gradient problem. In fact, SGD has been improved so many times over the years that its performance is much better than the original version. Nevertheless the SGD family has its own limitation such as in dealing with vanishing gradient, and in exploring network structures. This work provides a new perspective as to how to remove one key requirement for SGD training – the need to perform back propagation.

5.2 Problem Formulation

Consider a feed-forward fully connected neural network with $L+1$ layers, for each layer l we define weight and bias as $\boldsymbol{W}_l$ and $\boldsymbol{b}_l$, respectively. Define the entire dataset as ϕ, assume that we have n data points, and for each data point $i \in \phi$ the input for layer l is $\boldsymbol{W}_l \boldsymbol{x}_{l-1,i} + \boldsymbol{b}_l$, while the output is each layer as $\boldsymbol{x}_{l,i} = \sigma(\boldsymbol{W}_l \boldsymbol{x}_{l-1,i} + \boldsymbol{b}_l), l = 1, \ldots, L$, where $\sigma(\cdot)$ is some activation function. Suppose $\boldsymbol{d}_i$ represents the label of data point i for all $i \in \phi$. Let us define $\boldsymbol{W} := \{\boldsymbol{W}_l\}_{l=1}^{L}$. The goal is to find parameter $\boldsymbol{W}$ that optimizes an empirical loss function denoted by $\ell(\cdot)$ over training data sets:

$$\min_{\boldsymbol{W}} \sum_{i=1}^{n} \ell(\boldsymbol{x}_{L,i}(\{\boldsymbol{W}, \boldsymbol{b}\}), \boldsymbol{d}_i). \tag{5.1}$$

Gradient-based methods solve (5.1) by updating all weights jointly through one stochastic gradient step. Due to the coupling of weights from different layers, gradients from shallower layers are computed based on products of weight matrices and derivatives of activation functions from deeper layers. If at some points the eigenvalues of weight matrices are very small or derivatives of activation functions are close to 0, the resulting gradients will become small therefore “non-informative”, hence the name “vanishing” gradient. This phenomenon becomes even worse when the network involves more hidden layers, and consequently the training can be very slow.

Since vanishing gradient comes from the nested structure of DNN, one would find it natural to consider decomposing the connections between layers. Following [33], we introduce auxiliary variables $\boldsymbol{y}_{l,i}$ for each layer l and data point i. See Figure 5.1 for network structure and explanation. Problem (5.1) can be

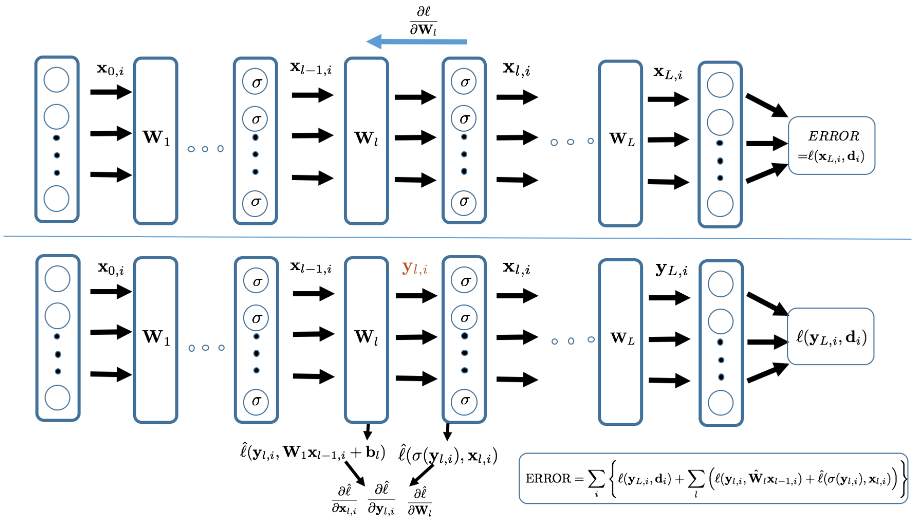

$\frac{\partial \ell}{\partial \mathbf{W}_l}$
$\mathbf{x}_{0,i}$
$\mathbf{W}_1$
$\mathbf{x}_{l-1,i}$
$\mathbf{W}_l$
$\mathbf{x}_{l,i}$
$\mathbf{W}_L$
$\mathbf{x}_{L,i}$
σ
ERROR $=\ell(\mathbf{x}_{L,i}, \mathbf{d}_i)$
$\mathbf{y}_{l,i}$
$\mathbf{y}_{L,i}$
$\ell(\mathbf{y}_{L,i}, \mathbf{d}_i)$
$\hat{\ell}(\mathbf{y}_{l,i}, \mathbf{W}_1\mathbf{x}_{l-1,i} + \mathbf{b}_l)$
$\hat{\ell}(\sigma(\mathbf{y}_{l,i}), \mathbf{x}_{l,i})$
$\frac{\partial \hat{\ell}}{\partial \mathbf{x}_{l,i}}$ $\frac{\partial \hat{\ell}}{\partial \mathbf{y}_{l,i}}$ $\frac{\partial \hat{\ell}}{\partial \mathbf{W}_l}$
$\text{ERROR} = \sum_i \left\{ \ell(\mathbf{y}_{L,i}, \mathbf{d}_i) + \sum_l \left(\ell(\mathbf{y}_{l,i}, \hat{\mathbf{W}}_l\mathbf{x}_{l-1,i}) + \hat{\ell}(\sigma(\mathbf{y}_{l,i}), \mathbf{x}_{l,i}) \right) \right\}$

Figure 5.1: SGD methods backprop the error from deep layers to shallow layers in the form of gradients of a single loss function $\ell(\cdot)$ and then update weights jointly. We introduce auxiliary variables $\mathbf{y}_l$ to decompose the coupling term $\boldsymbol{x}_l = \sigma(\boldsymbol{W}_l\boldsymbol{x}_{l-1} + \boldsymbol{b}_l)$ into $\boldsymbol{x}_l = f(\boldsymbol{y}_l)$ and $\boldsymbol{y}_l = \boldsymbol{W}_l\boldsymbol{x}_{l-1} + \boldsymbol{b}_l$. After relaxing these terms and penalizing them to the objective, the gradients of all variables can be directly accessed without using backprop.

reformulated as the following equality constrained optimization problem:

$$\min_{\boldsymbol{x}_{l,i}, \boldsymbol{y}_{l,i}, \boldsymbol{W}_l, \boldsymbol{b}_l} \sum_{i=1}^{n} \ell(\boldsymbol{y}_{L,i}, \boldsymbol{d}_i) \tag{5.2}$$

$$\text{s.t. } \boldsymbol{y}_{l,i} = \boldsymbol{W}_l\boldsymbol{x}_{l-1,i} + \boldsymbol{b}_l, \ \forall i \in \phi, l = 1, 2, \ldots, L;$$

$$\boldsymbol{x}_{l,i} = \sigma(\boldsymbol{y}_{l.i}), \qquad \forall i \in \phi, l = 1, 2, \ldots, L-1,$$

where we do not have activation function for last layer. Instead of only optimizing over weight matrices, we treat $\boldsymbol{x}_{l,i}, \boldsymbol{y}_{l,i}, \boldsymbol{W}_l, \boldsymbol{b}_l$, $l = 1, \cdots, L$, $i \in \phi$ as variables. In the following section we introduce the proposed algorithm to efficiently optimize (5.2).

5.3 Stochastic Primal-Dual Decomposition Algorithm

First we provide a brief introduction for a number of key ingredients of the proposed algorithm.

Stochastic Variance-Reduced Methods: In SGD the cost of evaluating the gradient is significantly cheaper than GD algorithm. However, at the same time the estimation might have a high variance. Variance-reduced methods utilize past stochastic gradient information to reduce the variance of the gradient estimation while still maintaining cheap gradient evaluation [87–89]. It is proved that this algorithm converges to the set of first-order stationary solutions under specific conditions [90–92].

Stochastic Block Coordinate Descent: For high-dimensional problems, it is often expensive to compute full gradients of all variables in a single iteration. Therefore it is usually computationally beneficial to stochastically pick a subset of variables to update [93].

Double-Stochastic Block Coordinate Descent: A natural algorithm to deal with the problem when both data size and variable number are large is to combine variance reduction and BCD together; for example see the S2CD algorithm proposed for convex problems [94].

Our Proposed Algorithm. Now we present our algorithm for the reformulated problem (5.2). First let us define the augmented Lagrangian (AL) function as follows:

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}_{l,i}, \boldsymbol{y}_{l,i}, \{\boldsymbol{W}_l; \boldsymbol{b}_l\}; \boldsymbol{\lambda}_{l,i}, \gamma_{l,i}) = &\sum_{i=1}^{n} \ell(\boldsymbol{y}_{L,i}, \boldsymbol{d}_i) + \sum_{i=1}^{n}\sum_{l=1}^{L} \frac{\rho_{l,i}}{2}\left\|\boldsymbol{y}_{l,i} - \boldsymbol{W}_l\boldsymbol{x}_{l-1,i} - \boldsymbol{b}_l + \frac{\boldsymbol{\lambda}_{l,i}}{\rho_{l,i}}\right\|^2 \\ &+ \sum_{i=1}^{n}\sum_{l=1}^{L-1}\frac{\beta_{l,i}}{2}\left\|\boldsymbol{x}_{l,i} - \sigma(\boldsymbol{y}_{l,i}) + \frac{\gamma_{l,i}}{\beta_{l,i}}\right\|^2 \end{aligned} \tag{5.3}$$

where $\boldsymbol{\lambda}_{l,i}$ and $\gamma_{l,i}$ are Lagrangian multipliers and $\rho_{l,i} > 0$ and $\beta_{l,i} > 0$ are penalty parameters. For notation simplicity, we define $\hat{\boldsymbol{W}}_l = \{\boldsymbol{W}_l; \boldsymbol{b}_l\}$ and $\hat{\boldsymbol{x}}_{l-1,i} = \{\boldsymbol{x}_{l-1,i}; 1\}$ and

$$q_i(\hat{\boldsymbol{W}}_l) = \sum_{l=1}^{L} \frac{\rho_{l,i}}{2}\left\|\boldsymbol{y}_{l,i} - \hat{\boldsymbol{W}}_l\hat{\boldsymbol{x}}_{l-1,i} + \frac{\boldsymbol{\lambda}_{l,i}}{\rho_{l,i}}\right\|^2 .$$

Our algorithm contains two loops. The inner loop minimizes the AL function with respect to the primal variables i.e. $(\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{W}})$ using certain variance reduced double-stochastic BCD algorithm. Notice that this primal problem is in a finite-sum format over data point and is nonconvex, therefore it is quite challenging. There have been some efforts to tackle this problem in nonconvex setting. See for example [90–92]. However, all of these methods need to update all coordinates in each iteration, where this is computationally very prohibitive in the case of problem (5.3) with too many parameters. Our algorithm randomly picks one data point and one coordinate, and update using the stochastic variance-reduced algorithm, while keeping other variables fixed. For notation simplicity, we group $\{\boldsymbol{x}_{l,i}, \boldsymbol{y}_{l,i}, \hat{\boldsymbol{W}}_l\}$ over coordinates as $\{\boldsymbol{x}_i, \boldsymbol{y}_i, \hat{\boldsymbol{W}}\}$. In order to track the summation of gradients we define an intermediate variable $\hat{\boldsymbol{V}}_j$ for $j = 1, \dots, n$ (one for each data point). We make the following assignments: $\hat{\boldsymbol{V}}_j^k = \hat{\boldsymbol{W}}^k$ if data point j is selected at iteration k, otherwise $\hat{\boldsymbol{V}}_j^k = \hat{\boldsymbol{V}}_j^{k-1}$ for $j \in \{1, \dots, n\}$. Up till now we have introduced one iteration of primal update. Unlike ADMM we choose to perform primal updates for several iterations until it reaches some accuracy, which we define as $\mathcal{S}$. See Algorithm 3 for detailed steps. A specific stopping criteria is given in the convergence analysis section.

Algorithm 3 Double-stochastic-BCD($\{\boldsymbol{x}^k, \boldsymbol{y}^k, \hat{\boldsymbol{W}}^k\}$)

```
Initialize $k = 0, \frac{\partial q_i(\hat{\boldsymbol{W}}_l^0)}{\partial \hat{\boldsymbol{W}}_l^0} = 0, i = 1, \dots, n, l = 1, \dots, L$
while stopping criteria $\mathcal{S}$ is not satisfied do
    randomly pick $j \in \{1, \dots, n\}$ with probability $p_j$
    randomly pick $p \in \{1, 2, 3\}$
    if $p = 1$ then
        $\hat{\boldsymbol{W}}^{k+1} = \hat{\boldsymbol{W}}^k - \beta \left( \sum_i \frac{\partial q_i(\hat{\boldsymbol{V}}_i^{k-1})}{\partial \hat{\boldsymbol{W}}^{k-1}} - \frac{\partial q_j(\hat{\boldsymbol{V}}_i^{k-1})}{\partial \hat{\boldsymbol{W}}^{k-1}} + \frac{\partial q_j(\hat{\boldsymbol{W}}^k)}{\partial \hat{\boldsymbol{W}}^k} \right)$
        $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k, \quad \boldsymbol{y}^{k+1} = \boldsymbol{y}^k, \quad \hat{\boldsymbol{V}}_j^k = \hat{\boldsymbol{W}}^k, \quad \hat{\boldsymbol{V}}_i^k = \hat{\boldsymbol{V}}_i^{k-1}$, for $i \neq j$
    end if
    if $p = 2$ then
        $\boldsymbol{x}_j^{k+1} = \boldsymbol{x}_j^k - \beta \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}_j^k}, \quad \boldsymbol{x}_{j_r}^{k+1} = \boldsymbol{x}_{j_r}^k, j\ _r \neq j$
        $\boldsymbol{y}^{k+1} = \boldsymbol{y}^k, \hat{\boldsymbol{W}}^{k+1} = \hat{\boldsymbol{W}}^k, \hat{\boldsymbol{V}}_i^k = \hat{\boldsymbol{V}}_i^{k-1}$, for $i \in \{1, \dots, n\}$
    end if
    if $p = 3$ then
        $\boldsymbol{y}_j^{k+1} = \boldsymbol{y}_j^k - \beta \frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_j^k}, \boldsymbol{y}_{j_r}^{k+1} = \boldsymbol{y}_{j_r}^k, j_r \neq j$
        $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k, \hat{\boldsymbol{W}}^{k+1} = \hat{\boldsymbol{W}}^k, \hat{\boldsymbol{V}}_i^k = \hat{\boldsymbol{V}}_i^{k-1}$, for $i \in \{1, \dots, n\}$
    end if
    $k = k + 1$
end while
return $\{\boldsymbol{x}^k, \boldsymbol{y}^k, \hat{\boldsymbol{W}}^k\}$
```

Next we present the outer loop which performs either augmented Lagrangian method or penalty method, depending on the size of the constraint violation. For simplicity we denote equality constraints in problem (5.2) as $h(\boldsymbol{x}_{l,i}, \boldsymbol{y}_{l,i}, \hat{\boldsymbol{W}}_l) = \mathbf{0}$. The algorithm starts outer loop by checking the following condition:

$$||h(\boldsymbol{x}_{l,i}^r, \boldsymbol{y}_{l,i}^r, \hat{\boldsymbol{W}}_l^r)||_\infty \leq \delta_r, \tag{5.4}$$

where $\delta_r > 0$ is a positive constant which measures the equality constraint violation and $\delta_r \to 0$ as $r \to \infty$. If (5.4) is satisfied, we update the dual variables $(\boldsymbol{\lambda}_{l,i}, \gamma_{l,i})$ by performing a dual ascent step while keeping

Algorithm 4 Stochastic Primal-Dual Decomposition (DSPD)

```
Initialize $\boldsymbol{x}^0_{l,j}, \boldsymbol{y}^0_{l,j}, \hat{\boldsymbol{W}}^0_l, \boldsymbol{\lambda}^0_{l,j}, \gamma^0_{l,j}, \alpha > 1$
repeat
  $\{\boldsymbol{x}^r, \boldsymbol{y}^r, \hat{\boldsymbol{W}}^r\} = \texttt{Double-stochastic-BCD}(\{\boldsymbol{x}^{r-1}, \boldsymbol{y}^{r-1}, \hat{\boldsymbol{W}}^{r-1}\})$
  for $i = 1, \dots, n$ do
    for $l = 1, \dots, L$ do
      if $\|h(\boldsymbol{x}^r_{l,i}, \boldsymbol{y}^r_{l,i}, \hat{\boldsymbol{W}}^r_l)\|_\infty \le \delta_r$ then
        $\begin{pmatrix} \boldsymbol{\lambda}^{r+1}_{l,i} \\ \gamma^{r+1}_{l,i} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\lambda}^{r}_{l,i} \\ \gamma^{r}_{l,i} \end{pmatrix} + \begin{pmatrix} \rho_{l,i} \\ \beta_{l,i} \end{pmatrix} h(\boldsymbol{x}^r_{l,i}, \boldsymbol{y}^r_{l,i}, \hat{\boldsymbol{W}}^r_l)$
      else
        $\rho_{l,i} = \rho_{l,i} * \alpha, \beta_{l,i} = \beta_{l,i} * \alpha$
      end if
    end for
  end for
  $r = r + 1$
until some termination criterion is met
```

$(\rho_{l,i}, \beta_{l,i})$ unchanged. Otherwise, we increase the penalty parameter $(\rho_{l,i}, \beta_{l,i})$ while the dual variables $(\boldsymbol{\lambda}_{l,i}, \gamma_{l,i})$ remain fixed. See Algorithm 4 for detail updates.

It is worth mentioning that if performing the dual ascent update without increasing the penalty, the method becomes the classical augmented Lagrangian method, and it may not converge for non-convex problems [95]. On the other hand, is it also known that using penalty method alone could be very inefficient as it requires the penalty parameters to go to infinity [95]. Therefore, we propose to switch between dual ascent step and penalty method is trying to find a proper penalty parameter to ensure augmented Lagrangian method is convergent.

5.4 Convergence Analysis

For simplicity of presentation, we consider the following general problem:

$$\min_{\boldsymbol{z}} \sum_i g_i(\boldsymbol{z}), \quad \text{s.t.} \quad h(\boldsymbol{z}) = \boldsymbol{0}, \tag{5.5}$$

where $g_i(\boldsymbol{z})$'s are continuously differentiable; $h(\boldsymbol{z}) = \{h_i(\boldsymbol{z}_i)\}_{i \in \phi}$; $\boldsymbol{z}$ has m block variables. Problem (5.2) is a special case of (5.5) in the following sense: i) First, the variables $\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{W}}$ can be viewed as three blocks consisting of $\boldsymbol{z}$; ii) For a fixed i, we can specialize $g_i(\boldsymbol{z})$ as $\ell_i(\boldsymbol{W}, \boldsymbol{x}_i, \boldsymbol{y}_i)$ which is the component loss function in (5.2). It is important to note that in our problem (5.2), each $\boldsymbol{x}_i, \boldsymbol{y}_i$ is only contained in the ith loss function, while problem (5.5) is slightly more general since we do not explicitly write out such a dependency. iii) Each $h_i(\boldsymbol{z}_i)$ corresponds to equality constraints for each data point i. Next, we define the augmented Lagrangian function for (5.5) as follows:

$$\mathcal{L}(\boldsymbol{z};\boldsymbol{\mu}) = \sum_{i=1}^{n} g_i(\boldsymbol{z}) + \sum_{i=1}^{n} \frac{\kappa_i}{2} \left\| h_i(\boldsymbol{z}) + \frac{\boldsymbol{\mu}_i}{\kappa_i} \right\|^2, \tag{5.6}$$

where the $\boldsymbol{\mu}_i$ is the dual variable associated with the constraint $h_i(\boldsymbol{z}) = \mathbf{0}$ and $\kappa_i > 0$ is the penalty parameter. In the primal problem of DSPD the dual variable $\boldsymbol{\mu}$ is held fixed. So for further simplicity let us define $f_i(\boldsymbol{z}) := g_i(\boldsymbol{z}) + \frac{\kappa_i}{2} \left\| h_i(\boldsymbol{z}) + \frac{\boldsymbol{\mu}_i^r}{\kappa_i} \right\|^2$. Then the primal problem is indeed minimizing the finite-sum function $f(\boldsymbol{z}) := \sum_{i=1}^{n} f_i(\boldsymbol{z})$ over variable $\boldsymbol{z}$. Furthermore, let us define the following gradients:

$$\boldsymbol{\Phi}(\boldsymbol{z}) = \sum_{i=1}^{n} \nabla f_i(\boldsymbol{z}), \;\; \boldsymbol{\Psi}(\boldsymbol{z},\boldsymbol{\mu}) = \begin{pmatrix} \boldsymbol{\Phi}(\boldsymbol{z}) \\ h(\boldsymbol{z}) \end{pmatrix}, \tag{5.7}$$

where $\boldsymbol{\Phi}(\boldsymbol{z})$ is gradient for the primal problem and $\boldsymbol{\Psi}(\boldsymbol{z},\boldsymbol{\mu})$ is gradient for the AL function (5.6). Also we define an intermediate variable $\boldsymbol{y}_{i,j}$ as (5.17), then Algorithm 1 can be compactly and equivalently written as follows.

Algorithm 5 Double-stochastic-BCD($\boldsymbol{z}^k$)

```
Initialize $k = 0, \frac{\partial f_i(\boldsymbol{z}^0)}{\partial \hat{\boldsymbol{W}}_l^0} = 0, i = 1, \ldots, n, l = 1, \ldots, L$
while stopping criteria $\mathcal{S}$ is not satisfied do
    randomly pick $i_k \in \{1, \ldots, n\}$ with probability $p_i$
    randomly pick $j_k \in \{1, \ldots, m\}$
    $\boldsymbol{z}_j^{k+1} = \begin{cases} \boldsymbol{z}_j^k - \beta\Big( \sum_{i=1}^{n} \frac{\partial f_i(\boldsymbol{y}_{ij}^{k-1})}{\partial \boldsymbol{z}_j} + \frac{1}{p_{i_k}} \big( \frac{\partial f_{i_k}(\boldsymbol{z}^k)}{\partial \boldsymbol{z}_j} - \frac{\partial f_{i_k}(\boldsymbol{y}_{i_k j}^{k-1})}{\partial \boldsymbol{z}_j} \big) \Big), \; j = j_k \\ \boldsymbol{z}_j^k, \text{ o.w.,} \end{cases}$
    $k = k + 1$
end while
return $\boldsymbol{z}^k$
```

Before we go into details of convergence analysis, we make the following assumptions:

Assumptions A

A1. The gradient $\nabla f_i(\boldsymbol{z})$ is Lipschitz continuous, i.e. there exists $L_i > 0$ such that

$$\|\nabla f_i(\boldsymbol{z_1}) - \nabla f_i(\boldsymbol{z_2})\| \le L_i \|\boldsymbol{z_1} - \boldsymbol{z_2}\|, \ \forall \ \boldsymbol{z_1}, \boldsymbol{z_2} \in \mathrm{dom}(f_i), \ \forall \ i. \tag{5.8}$$

A2. For all $i = 1, 2, \cdots, n$, and for all $j = 1, 2, \cdots, m$, partial gradient of function f_i with respect to coordinate j is Lipschitz continuous i.e. there exists $L_{ij} > 0$ such that

$$\left\| \frac{\partial f_i(\boldsymbol{z_1})}{\partial \boldsymbol{z_j}} - \frac{\partial f_i(\boldsymbol{z_2})}{\partial \boldsymbol{z_j}} \right\| \le L_{ij} \|\boldsymbol{z_1} - \boldsymbol{z_2}\|, \ \forall \ \boldsymbol{z_1}, \boldsymbol{z_2} \in \mathrm{dom}(f_i). \tag{5.9}$$

A3. The function $f(\boldsymbol{z}) := \sum_{i=1}^{n} f_i(\boldsymbol{z})$ is lower bounded.

We also assume certain constraint qualification condition known as the *Robinson's condition* [96]:

Definition 1. *Rewrite the constraints in* (5.5) *as* $h(\boldsymbol{z}) \in Z_0$*, where* $Z_0 = \{0\}$*. Robinson's condition is satisfied at* $\bar{z}$ *for problem* (5.5) *if the following holds*

$$\{\nabla h(\bar{z})\boldsymbol{d} : \boldsymbol{d} \in T_{Z_0}(h(\bar{z}))\} = \mathbb{R}^n, \tag{5.10}$$

where $T_{Z_0}(\boldsymbol{z})$ *denotes the tangent cone of* Z_0 *at* $h(\boldsymbol{z})$*,* n *is the cardinality of* ϕ*.*

From [96] it is known that *Robinson's condition* is equivalent to metric regularity, which states that distance of perturbed point $\hat{\boldsymbol{z}}$ to a solution $\bar{\boldsymbol{z}}$ of the perturbed system is proportional to the violation of constraints. From this intuition we can see that the switching condition between AL method and penalty method makes perfect sense. As long as we can have constraint violation small, we should be able to achieve a solution that is close to stationary point. It is worth mentioning that using regularity conditions such as Robinson' condition has been standard in classical nonlinear nonconvex optimization literature, for example see the existing works [96–101]. Such condition also reduces to the well-known Mangasarian-Fromovitz constraint qualification (MFCQ) and Slater conditions under appropriate problem settings [96].

Let us associate a new parameter $\eta_i > 0$ to ith component for all $i = 1, 2, \cdots, n$ and define the stepsize $\beta := \frac{1}{\sum_{i=1}^{n} \eta_i}$. In the next lemma we prove that the gradient of primal problem converges to zero almost surely under specific condition on η_i.

Lemma 5. *Suppose Assumptions A hold true,* η_i *is satisfied in* $5\eta_i - 4L_i - \sqrt{L_i^2 + \frac{32c}{p_i}} > 0$*, where* $c := \frac{1}{m}\sum_{j=1}^{m}(1 + \frac{2m}{p_i})L_{ij}^2$*, and* p_i *is the probability of picking* $i \in \{1, 2, , \cdots, n\}$*. Let* $\{\boldsymbol{z}^r\}$ *be the sequence generated by Algorithm 4, we have*

$$\lim_{r\to\infty} \|\boldsymbol{\Phi}(\boldsymbol{z}^r)\| = 0 \textit{ with probability 1} \tag{5.11}$$

Lemma 5 gives us guarantees that primal problem can converge to stationary solution almost surely. Further, we can show that the primal problem also achieves a sublinear rate of convergence.

Theorem 5. *Suppose all the conditions in Lemma 5 are satisfied. Let the primal step runs for* R *iterations, and* u *is a uniformly randomly number chosen from* $\{1, 2, \cdots, R\}$*. Define* Q *as a potential function of* $\sum_i f_i(\boldsymbol{z})$ *with formulation in (5.23). Then we have*

$$\mathbb{E}\|\nabla f(\boldsymbol{z}^u)\|^2 \leq \frac{16m^2}{\beta}\frac{\mathbb{E}[Q^1 - Q^{R+1}]}{R}. \tag{5.12}$$

Finally, we are ready to present the overall convergence analysis. To this end, let us introduce the termination condition for the inner loop:

$$\mathcal{S} : \|\boldsymbol{\Phi}(\boldsymbol{z}^r)\| \leq \epsilon_r, \epsilon_r \to 0 \text{ as } r \to \infty \tag{5.13}$$

This means that the inner loop is performed with increasing accuracy as the iteration continues. In practice this is easy to achieve since as the algorithm proceeds, the problem will be close to some stationary solution already, therefore it does not take long to reach small ϵ_r.

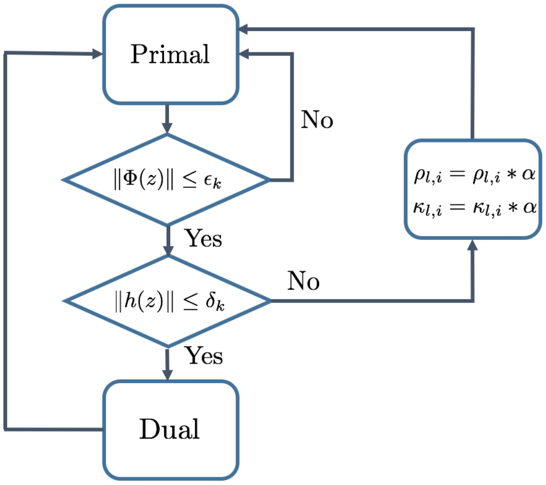

Primal
No
$\rho_{l,i} = \rho_{l,i} * \alpha$
$\kappa_{l,i} = \kappa_{l,i} * \alpha$
$\|\Phi(z)\| \leq \epsilon_k$
Yes
No
$\|h(z)\| \leq \delta_k$
Yes
Dual

Figure 5.2: Conditions on primal and dual updates ensure convergence to stationary solution.

Remark. We can see from the analysis that conditions imposed on primal and dual updates are the key to ensure convergence of our algorithm, see Figure 5.2. As long as we can keep primal gradient and constraint violation very small. We can achieve stationary solution with probability 1. With that, we give our main convergence result in the following theorem.

Theorem 6. *Let $\{\boldsymbol{z}^r, \boldsymbol{\mu}^r\}$ be the sequence generated by Algorithm 4 for problem (5.5), where $\boldsymbol{\mu}^r = \{\boldsymbol{\mu}_i^r\}$ are the Lagrangian multipliers associated with the equality constraints. Assume termination condition (5.13) is satisfied and suppose that $\boldsymbol{z}^*$ is a limit point of $\{\boldsymbol{z}^r\}$ and at the limit point $\boldsymbol{z}^*$ Robinson's condition holds true. Then we have the following result:*

$$\lim_{r\to\infty} \|\boldsymbol{\Psi}(\boldsymbol{z}^r, \boldsymbol{\mu}^r)\| = 0, \quad \text{with probability 1} \tag{5.14}$$

Proof. Without loss of generality, we assume $\{\boldsymbol{z}^r\}$ converges to $\boldsymbol{z}^*$ with probability 1, otherwise we can restrict to a convergent subsequence of $\{\boldsymbol{z}^r\}$. From Lemma 5 we have

$$\begin{aligned}&\lim_{r\to\infty} \|\boldsymbol{\Phi}(\boldsymbol{z}^r)\| \\ &= \lim_{r\to\infty} \left\| \sum_i (\nabla g_i(\boldsymbol{z}^r) + \boldsymbol{\mu}_i^r \nabla h_i(\boldsymbol{z}^r) + \kappa_i \nabla h_i(\boldsymbol{z}^r)^T h_i(\boldsymbol{z}^r)) \right\| = 0 \text{ with probability 1}\end{aligned} \tag{5.15}$$

Define $\hat{\boldsymbol{\mu}}_i^r = \boldsymbol{\mu}_i^k + \kappa_i h_i(\boldsymbol{z}^r)$, $\hat{\boldsymbol{\mu}}^r = \{\hat{\boldsymbol{\mu}}_i^r\}$, then we have

$$\lim_{r\to\infty} \|\nabla g(\boldsymbol{z}^r) + \nabla h(\boldsymbol{z}^r)^T(\hat{\boldsymbol{\mu}}^r)\| = \mathbf{0} \text{ with probability } 1. \tag{5.16}$$

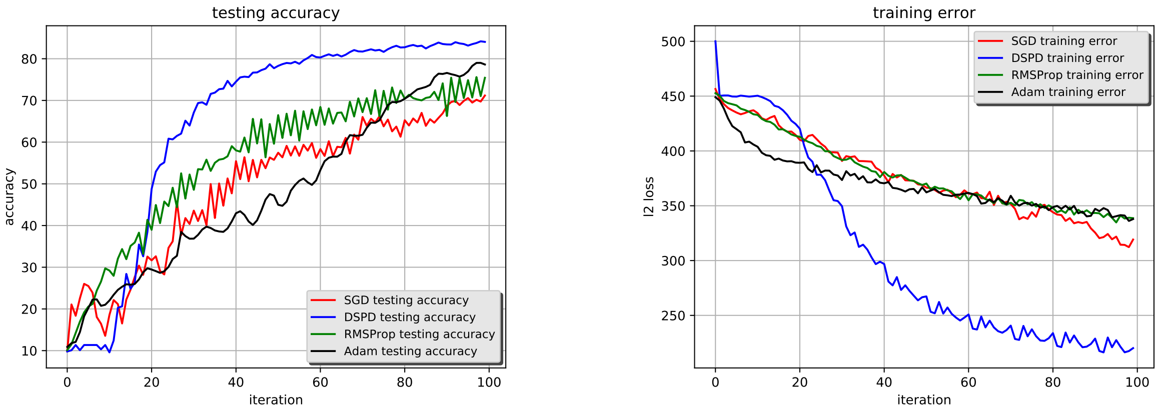
Note that $\|\boldsymbol{\Psi}(\boldsymbol{z}^r, \boldsymbol{\mu}^r)\| \leq \|\boldsymbol{\Phi}(\boldsymbol{z}^r)\| + \|h(\boldsymbol{z}^r)\|$, so in order to prove (5.14), we just need to show the dual step gradient $\lim_{r\to\infty} \|h(\boldsymbol{z}^r)\| = 0$ with probability 1.

We start by proving $\hat{\boldsymbol{\mu}}^r$ is a bounded sequence. Assume to the contrary that $\hat{\boldsymbol{\mu}}^r$ is unbounded, define $\bar{\boldsymbol{\mu}}^r = \frac{\hat{\boldsymbol{\mu}}^r}{\|\hat{\boldsymbol{\mu}}^r\|}$, then $\bar{\boldsymbol{\mu}}^r$ is bounded. Therefore there exists a convergent subsequence $\{\bar{\boldsymbol{\mu}}^{r_k}\}$ with $\bar{\boldsymbol{\mu}}^{r_k} \to \bar{\boldsymbol{\mu}}$ as $k \to \infty$. Since $g(\boldsymbol{z})$ is continuously differentiable, then $\nabla g(\boldsymbol{z})$ is bounded. Divide both sides of (5.16) by $\|\hat{\boldsymbol{\mu}}^r\|$ we have for sufficiently large k that $\|\nabla h(\boldsymbol{z}^*)^T(\bar{\boldsymbol{\mu}})\| = 0$ with probability 1. Also from Robinson's condition we know there exists some $\boldsymbol{z}$ and $c > 0$ such that $-\bar{\boldsymbol{\mu}} = c\nabla h(\boldsymbol{z}^*)(\boldsymbol{z} - \boldsymbol{z}^*)$. Combining together implies that $\|\bar{\boldsymbol{\mu}}\| = \mathbf{0}$ with probability 1, contradicting to the fact that $\|\bar{\boldsymbol{\mu}}\| = 1$. Therefore $\{\hat{\boldsymbol{\mu}}^r\}$ is bounded. From the definition of $\hat{\boldsymbol{\mu}}^r$ there are 2 possible cases corresponding to 2 scenarios in outer loop: 1) $\hat{\boldsymbol{\mu}}^r - \boldsymbol{\mu}^r \to 0$ with κ bounded; 2) $\hat{\boldsymbol{\mu}}^r$ and $\boldsymbol{\mu}^r$ are both bounded with $\kappa \to \infty$. Hence we must have $\|h(\boldsymbol{z}^r)\| \to 0$ with probability 1, together with (5.7) , (5.11) we complete the proof. ■

5.5 Numerical Experiments

We conduct numerical experiments to demonstrate the efficiency of our algorithm, particularly in the early stage of training. We implement the experiments on MNIST dataset [102] and the detailed settings are as follows: 1) we use 784-784-784-10 fully connected feedforward neural network; 2) penalty parameters are initialized as 0.001 and δ_r is set to be 0.9×previous constraint violation; 3) $\ell(\cdot)$ is set to be l_2 loss. i.e. $\frac{1}{2}\sum_i \|\boldsymbol{y}_{L,i} - \boldsymbol{d}_i\|^2$; 4) activation is set to be hyperbolic tangent function (tanh); 5) DSPD is implemented using Python 3.6.3 without optimizing the code on a CPU, while SGD-based methods are implemented using Tensorflow training function with GPU support and the step size is optimized by hand to get the best result.

In the first example we compare our proposed DSPD algorithm with SGD-based algorithms, vanilla SGD, RMSProp, and Adam. Each time a minibatch of size 1000 data points is sampled. To make it a fair comparison, one pass of all the data through primal updates of DSPD is counted as one iteration. We can see from Figure 5.4, that DSPD is able to outperform SGD-based methods in the early stage.



(a)

(b)

Figure 5.4: Comparison of training error and testing accuracy between DSPD and gradient-based methods

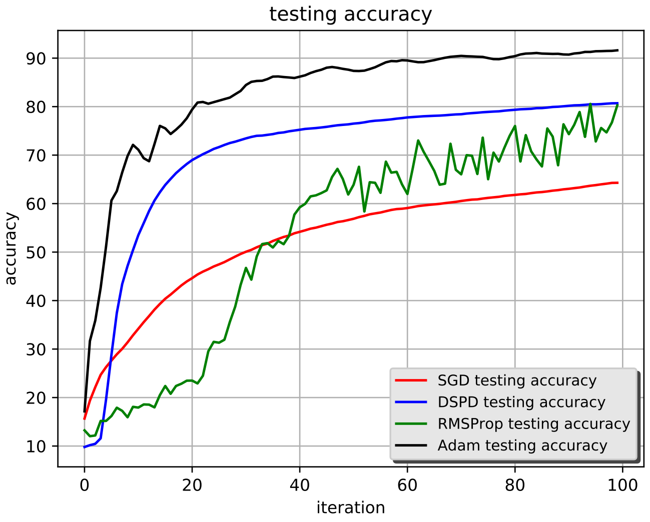


Figure 5.5: For limited data, DSPD is able to extract more information than vanilla SGD.

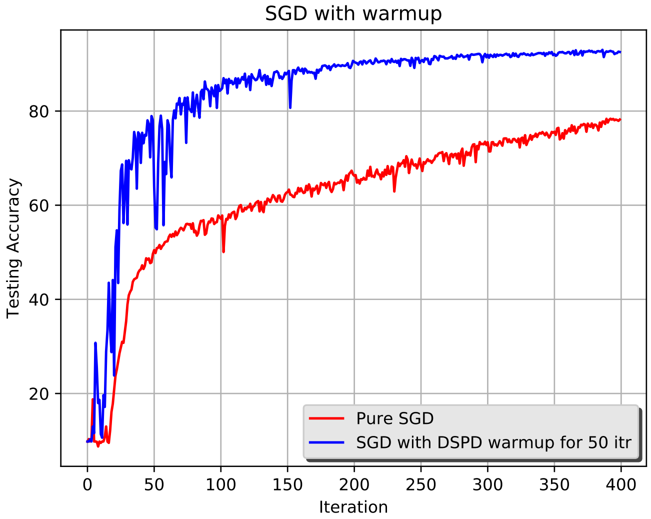


Figure 5.6: Running DSPD for 50 iterations can give SGD accuracy boost, achieving 92% accuracy within 400 iterations.

In the second example, we focus on a small portion of MNIST dataset (1000 data points) and run DSPD and several gradient-based methods in a batch way. We can see from Figure 5.5, that DSPD algorithm can achieve a decent accuracy using limited data, better than SGD and RMSProp, slightly worse than Adam. Recall that all the rest of the gradient-based methods (besides SGD) are using techniques such as momentum, adaptive learning rate etc., while our algorithm is the basic version.

In light of the early advantage of DSPD, we continue implementation using DSPD as initialization for SGD. This time we use a minibatch of 100 data points and the same neural network. From Figure 5.6 we can see that, running 50 iterations DSPD can help boost SGD convergence speed dramatically. The is because the decomposition of layers as well as having a loss function for each layer makes the algorithm able to see gradients of each layer explicitly at early stage, hence more information is extracted using DSPD.

5.6 Proof of Lemma 5

First let us derive the primal update for general problem $\min_{z\in\mathbb{R}^m} \ f(z) = \sum_{i=1}^n f_i(z)$. At each iteration we randomly pick a data index and variable index. Now let us define m new intermediate variables $y_{ij} \in \mathbb{R}^m$

for each i as follows

$$y_{ij}^r := \begin{cases} z^r & (i,j) = (i_r, j_r); \\ y_{ij}^{r-1} & \text{o.w.} \end{cases} \tag{5.17}$$

Let us associate a new parameter η_i to ith component and define $\beta := \frac{1}{\sum_{i=1}^n \eta_i}$. The primal update is presented as the following for all coordinates $j = 1, 2, \cdots, m$:

$$[z^{r+1}]_j = \begin{cases} [z^r]_j - \beta\Big(\sum_{i=1}^n \frac{\partial f_i(y_{ij}^{r-1})}{\partial z_j} + \frac{1}{p_{i_r}}\big(\frac{\partial f_{i_r}(z^r)}{\partial z_j} - \frac{\partial f_{i_r}(y_{i_r j}^{r-1})}{\partial z_j}\big)\Big), \; j = j_r \\ [z^r]_j, \; \text{o.w.}, \end{cases} \tag{5.18}$$

where p_{i_r} is the probability of picking i_r from $\{1, 2, \cdots, n\}$. For further simplicity, let us define

$$v_{i_r j_r}^r := \sum_{i=1}^n \frac{\partial f_i(y_{ij_r}^{r-1})}{\partial z_{j_r}} + \frac{1}{p_{i_r}}\big(\frac{\partial f_{i_r}(z^r)}{\partial z_{j_r}} - \frac{\partial f_{i_r}(y_{i_r j_r}^{r-1})}{\partial z_{j_r}}\big). \tag{5.19}$$

Also, let us define vector $e^r \in \mathbb{R}^m$ as $[e^r]_{j_r} = v_{i_r j_r}^r$ and $[e^r]_j = 0$ for $j \neq j_r$. From here we have the following compact update rule:

$$z^{r+1} = z^r - \beta e^r. \tag{5.20}$$

Taking expectation over i_r and j_r one can simply check that the following holds true

$$\mathbb{E}[z^{r+1} - z^r] = -\frac{\beta}{m}\nabla f(z^r). \tag{5.21}$$

Further from this and utilizing the fact that $\mathbb{E}\|x - \mathbb{E}[x]\|^2 \leq \mathbb{E}[x^2]$ for a random variable x we obtain the following:

$$\mathbb{E}\|z^{r+1} - z^r + \frac{\beta}{m}\nabla f(z^r)\|^2 \leq \mathbb{E}\|z^{r+1} - z^r\|^2. \tag{5.22}$$

Define the filtration $\mathcal{F}^r$ as the σ-field generated by $\{i_t, j_t\}_{t=1}^{r-1}$. Now let us define the following function:

$$Q^r := \sum_{i=1}^n \left[f_i(z^r) + \frac{1}{m}\sum_{j=1}^m \frac{3}{p_i \eta_i}\left\| \frac{\partial f_i(y_{ij}^{r-1})}{\partial z_j} - \frac{\partial f_i(z^r)}{\partial z_j} \right\|^2 \right]. \tag{5.23}$$

The potential function to measure the progress of the algorithm is defined as $\mathbb{E}_{\mathcal{F}^r}[Q^r]$.

Step 1. In this step we show that the potential function is decreasing under particular parameter selections. Let us define $c := \frac{1}{m}\sum_{j=1}^{m}(1+\frac{2m}{p_i})L_{ij}^2$, and pick parameter η_i large enough such that

$$\eta_i > \frac{4L_i + \sqrt{L_i^2 + \frac{32c}{p_i}}}{5}. \tag{5.24}$$

Then the following descent estimate holds true:

$$\begin{aligned}\mathbb{E}_{\mathcal{F}^r}[Q^r - Q^{r-1} \mid \mathcal{F}^{r-1}] \le & -\frac{\sum_{i=1}^n \eta_i}{8}\mathbb{E}_{i_{r-1}}[\|z^r - z^{r-1}\|^2 \mid \mathcal{F}^{r-1}] \\ & - \sum_{i=1}^{n}\frac{1}{2\eta_i}\sum_{j=1}^{m}\left\|\frac{\partial f_i(z^{r-1})}{\partial z_j} - \frac{\partial f_i(y_{ij}^{r-2})}{\partial z_j}\right\|^2.\end{aligned} \tag{5.25}$$

To show (5.25) using the definition of potential function Q^r, we have:

$$\begin{aligned}&\mathbb{E}_{\mathcal{F}^r}[Q^r - Q^{r-1} \mid \mathcal{F}^{r-1}] = \mathbb{E}_{\mathcal{F}^r}\left[\sum_{i=1}^{n}\big(f_i(z^r) - f_i(z^{r-1})\big) \mid \mathcal{F}^{r-1}\right] \\ &+ \mathbb{E}_{\mathcal{F}^r}\left[\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\frac{3}{p_i\eta_i}\left(\left\|\frac{\partial f_i(y_{ij}^{r-1})}{\partial z_j} - \frac{\partial f_i(z^r)}{\partial z_j}\right\|^2 - \left\|\frac{\partial f_i(y_{ij}^{r-2})}{\partial z_j} - \frac{\partial f_i(z^{r-1})}{\partial z_j}\right\|^2\right) \mid \mathcal{F}^{r-1}\right].\end{aligned} \tag{5.26}$$

The first term in (5.26) can be bounded as follows

$$\begin{aligned}&\mathbb{E}_{\mathcal{F}^r}\left[\sum_{i=1}^{n}\big(f_i(z^r) - f_i(z^{r-1})\big) \mid \mathcal{F}^{r-1}\right] \\ &\overset{\text{(i)}}{\le} \mathbb{E}_{\mathcal{F}^r}\left[\sum_{i=1}^{n}\langle\frac{\partial f_i(z^{r-1})}{\partial z_{j_{r-1}}}, [z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\rangle + \frac{\sum_{i=1}^n L_i}{2}\|[z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\|^2 \mid \mathcal{F}^{r-1}\right] \\ &= \mathbb{E}_{\mathcal{F}^r}\left[\left\langle\sum_{i=1}^{n}\frac{\partial f_i(z^{r-1})}{\partial z_{j_{r-1}}} + \frac{1}{\beta}([z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}), [z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\right\rangle \mid \mathcal{F}^{r-1}\right] \\ &\quad - \left(\frac{1}{\beta} - \frac{\sum_{i=1}^n L_i}{2}\right)\mathbb{E}_{\mathcal{F}^r}\left[\|[z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\|^2 \mid \mathcal{F}^{r-1}\right] \\ &\overset{(5.20)}{=} \mathbb{E}_{\mathcal{F}^r}\left[\left\langle\sum_{i=1}^{n}\frac{\partial f_i(z^{r-1})}{\partial z_{j_{r-1}}} - v_{i_{r-1}j_{r-1}}^{r-1}, [z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\right\rangle \mid \mathcal{F}^{r-1}\right] \\ &\quad - \left(\frac{1}{\beta} - \frac{\sum_{i=1}^n L_i}{2}\right)\mathbb{E}_{\mathcal{F}^r}\left[\|[z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\|^2 \mid \mathcal{F}^{r-1}\right] \\ &\overset{\text{(ii)}}{\le} \frac{1}{2\ell_1}\mathbb{E}_{\mathcal{F}^r}\left[\left\|\sum_{i=1}^{n}\frac{\partial f_i(z^{r-1})}{\partial z_{j_{r-1}}} - v_{i_{r-1}j_{r-1}}^{r-1}\right\|^2 \mid \mathcal{F}^{r-1}\right] + \frac{\ell_1}{2}\mathbb{E}_{\mathcal{F}^r}\left[\|[z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\|^2 \mid \mathcal{F}^{r-1}\right] \\ &\quad - \left(\frac{1}{\beta} - \frac{\sum_{i=1}^n L_i}{2}\right)\mathbb{E}_{\mathcal{F}^r}\left[\|[z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\|^2 \mid \mathcal{F}^{r-1}\right]\end{aligned} \tag{5.27}$$

where in (i) we have used the Lipschitz continuity of the gradients of f_i's together with the fact that $[z^r]_j - [z^{r-1}]_j = 0$ when $j \neq j_{r-1}$. In (ii) we have applied the Young's inequality for some $\ell_1 > 0$.

Choosing $\ell_1 = \frac{1}{2\beta}$, overall we have the following bound for the first term in (5.26):

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{n}\left(f_i(z^r) - f_i(z^{r-1})\right) \mid \mathcal{F}^{r-1}\right] \qquad (5.28)\\
&\le \sum_{i=1}^{n}\frac{\beta}{p_i}\left\|\frac{\partial f_i(z^{r-1})}{\partial z_{j_{r-1}}} - \frac{\partial f_i(y^{r-2}_{ij_{r-1}})}{\partial z_{j_{r-1}}}\right\|^2 - \left(\frac{3}{4\beta} - \frac{\sum_{i=1}^{n} L_i}{2}\right)\mathbb{E}_{\mathcal{F}^r}\left[\|[z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\|^2 \mid \mathcal{F}^{r-1}\right]\\
&\le \sum_{i=1}^{n}\frac{\beta}{p_i}\left\|\frac{\partial f_i(z^{r-1})}{\partial z_{j_{r-1}}} - \frac{\partial f_i(y^{r-2}_{ij_{r-1}})}{\partial z_{j_{r-1}}}\right\|^2 - \left(\frac{3}{4\beta} - \frac{\sum_{i=1}^{n} L_i}{2}\right)\mathbb{E}_{\mathcal{F}^r}\left[\frac{1}{m}\|z^r - z^{r-1}\|^2 \mid \mathcal{F}^{r-1}\right]
\end{aligned}$$

We bound the second term in (5.26) in the following way:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{F}^r}\left[\left\|\frac{\partial f_i(y^{r-1}_{ij})}{\partial z_j} - \frac{\partial f_i(z^r)}{\partial z_j}\right\|^2 \mid \mathcal{F}^{r-1}\right]\\
&= \mathbb{E}_{\mathcal{F}^r}\left[\left\|\frac{\partial f_i(y^{r-1}_{ij})}{\partial z_j} - \frac{\partial f_i(z^r)}{\partial z_j} + \frac{\partial f_i(z^{r-1})}{\partial z_j} - \frac{\partial f_i(z^{r-1})}{\partial z_j}\right\|^2 \mid \mathcal{F}^{r-1}\right]\\
&\overset{\text{(i)}}{\le} (1+\xi_{ij})\mathbb{E}_{\mathcal{F}^r}\left[\left\|\frac{\partial f_i(z^r)}{\partial z_j} - \frac{\partial f_i(z^{r-1})}{\partial z_j}\right\|^2 \mid \mathcal{F}^{r-1}\right]\\
&+ \left(1 + \frac{1}{\xi_{ij}}\right)\mathbb{E}_{\mathcal{F}^r}\left[\left\|\frac{\partial f_i(y^{r-1}_{ij})}{\partial z_j} - \frac{\partial f_i(z^{r-1})}{\partial z_j}\right\|^2 \mid \mathcal{F}^{r-1}\right]\\
&\overset{\text{(ii)}}{=} (1+\xi_{ij})L_{ij}^2\mathbb{E}_{\mathcal{F}^r}\left[\|[z^r]_{j_{r-1}} - [z^{r-1}]_{j_{r-1}}\|^2 \mid \mathcal{F}^{r-1}\right]\\
&+ (1 - \frac{p_i}{m})\left(1 + \frac{1}{\xi_{ij}}\right)\left\|\frac{\partial f_i(y^{r-2}_{ij})}{\partial z_j} - \frac{\partial f_i(z^{r-1})}{\partial z_j}\right\|^2 \qquad (5.29)
\end{aligned}$$

where in (i) we use Young's inequity for constant $\xi_{ij} > 0$. The equality (ii) is true because the randomness of y^{r-1}_{ij} comes from i_{r-1},and j_{r-1}. Also, i_r and j_r are independent random variables so for each i and j there is a probability $p_i.\frac{1}{m}$ such that jth block of z_i is updated. Therefore, we have

$$\frac{\partial f_i(y^{r-1}_{ij})}{\partial z_j} = \begin{cases} \frac{\partial f_i(z^{r-1})}{\partial z_j}, & \text{with probability } p_i.\frac{1}{m}\\ \frac{\partial f_i(y^{r-2}_{ij})}{\partial z_j}, & \text{with probability } 1 - p_i.\frac{1}{m} \end{cases} \qquad (5.30)$$

Applying (5.29), the second part of (5.26) can be bounded as

$$\begin{aligned}&\mathbb{E}_{\mathcal{F}^r}\left[\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m}\frac{3}{p_i\eta_i}\left(\left\|\frac{\partial f_i(y_{ij}^{r-1})}{\partial z_j}-\frac{\partial f_i(z^r)}{\partial z_j}\right\|^2-\left\|\frac{\partial f_i(y_{ij}^{r-2})}{\partial z_j}-\frac{\partial f_i(z^{r-1})}{\partial z_j}\right\|^2\right)\mid\mathcal{F}^{r-1}\right]\\&\leq\sum_{i=1}^{n}\frac{3}{p_i\eta_i}\frac{1}{m}\sum_{j=1}^{m}(1+\xi_{ij})L_{ij}^2\mathbb{E}_{\mathcal{F}^r}\left[\|z^r-z^{r-1}\|^2\mid\mathcal{F}^{r-1}\right]\\&+\sum_{i=1}^{n}\frac{3}{p_i\eta_i}\left(\sum_{j=1}^{m}(1-\frac{p_i}{m})(1+\frac{1}{\xi_{ij}})-1\right)\left\|\frac{\partial f_i(y_{ij}^{r-2})}{\partial z_j}-\frac{\partial f_i(z^{r-1})}{\partial z_j}\right\|^2.\end{aligned}\tag{5.31}$$

Combining (5.28) and (5.31) eventually we have

$$\begin{aligned}&\mathbb{E}[Q^r-Q^{r-1}\mid\mathcal{F}^r]\\&\leq\sum_{i=1}^{n}\left\{\frac{\beta}{p_i}+\frac{3}{p_i\eta_i}\left(\sum_{j=1}^{m}(1-\frac{p_i}{m})(1+\frac{1}{\xi_{ij}})-1\right)\right\}\left\|\frac{\partial f_i(y_{ij}^{r-2})}{\partial z_j}-\frac{\partial f_i(z^{r-1})}{\partial z_j}\right\|^2\\&+\left\{-\frac{3}{4\beta}+\frac{\sum_{i=1}^n L_i}{2}+\sum_{i=1}^{n}\frac{3}{p_i\eta_i}\frac{1}{m}\sum_{j=1}^{m}(1+\xi_{ij})L_{ij}^2\right\}\mathbb{E}_{\mathcal{F}^r}\left[\|z^r-z^{r-1}\|^2\mid\mathcal{F}^{r-1}\right].\end{aligned}\tag{5.32}$$

Let us define $\{\tilde{c}_i\}$ and $\hat{c}$ as the following:

$$\begin{aligned}\tilde{c}_i&=\frac{\beta}{p_i}+\frac{3}{p_i\eta_i}\left(\sum_{j=1}^{m}(1-\frac{p_i}{m})(1+\frac{1}{\xi_{ij}})-1\right)\\\hat{c}&=-\frac{3}{4\beta}+\frac{\sum_{i=1}^n L_i}{2}+\sum_{i=1}^{n}\frac{3}{p_i\eta_i}\frac{1}{m}\sum_{j=1}^{m}(1+\xi_{ij})L_{ij}^2.\end{aligned}$$

In order to prove the lemma it is enough to show that $\tilde{c}_i<-\frac{1}{2\eta_i}\ \forall\, i$, and $\hat{c}<-\sum_{i=1}^n\frac{\eta_i}{8}$. Let us pick

$$\xi_{ij}=\frac{2m}{p_i},\ p_i=\frac{\eta_i}{\sum_{i=1}^n\eta_i}.\tag{5.33}$$

Recall that $\beta=\frac{1}{\sum_{i=1}^n\eta_i}$. These values yield the following:

$$\tilde{c}_i=\frac{1}{\eta_i}-\frac{3}{\eta_i}\left(\frac{1}{2}+\frac{p_i}{m}\right)\leq-\frac{1}{2\eta_i}.$$

Next we show $\hat{c}\leq-\sum_{i=1}^n\frac{\eta_i}{8}$. Setting $c:=\frac{1}{m}\sum_{j=1}^m(1+\xi_{ij})L_{ij}^2$, it is sufficient to pick η_i such that

$$\eta_i>\frac{4L_i+\sqrt{L_i^2+\frac{32c}{p_i}}}{5}.\tag{5.34}$$

Step 2. First, using the fact that $f(z)$ is lower bounded [cf. Assumption A3], it is easy to check that $\{Q^r\}$ is a bounded sequence. Let us denote its lower bound by Q. From equation (5.32), we conclude that $\{Q^r - \text{Q}\}$ is a nonnegative supermartingale. Therefore, we can apply the Supermartingale Convergence Theorem proposed in [103, Proposition 4.2] and consequently we conclude that $\{Q^r\}$ converges w.p.1, and that

$$\mathbb{E}_{\mathcal{F}^r}\left[\|z^r - z^{r-1}\|^2 \mid \mathcal{F}^{r-1}\right] \to 0, \text{and} \quad \left\|\frac{\partial f_i(z^{r-1})}{\partial x_j} - \frac{\partial f_i(y_{ij}^{r-2})}{\partial x_j}\right\|^2 \to 0, \quad \forall\, i, j. \tag{5.35}$$

Using the definition of expectation we have that

$$\mathbb{E}_{\mathcal{F}^r}\|z^r - z^{r-1}\|^2 = \mathbb{E}_{\mathcal{F}^{r-1}}\left[\mathbb{E}_{\mathcal{F}^r}\left[\|z^r - z^{r-1}\|^2 \mid \mathcal{F}^{r-1}\right]\right]. \tag{5.36}$$

From the first expression of (5.35) together with equation (5.36) we conclude that

$$\mathbb{E}\|z^r - z^{r-1}\|^2 = \mathbb{E}_{\mathcal{F}^r}\|z^r - z^{r-1}\|^2 \to 0 \quad \text{w.p.1} \tag{5.37}$$

Combining this equation with (5.21) we conclude that $\|\nabla f(z^r)\| \to 0$ w.p.1. This completes the proof.

5.6.1 Proof of Theorem 5

We show that the stationarity gap vanishes in a sublinear manner as the algorithm proceeds. For this gap we have the following:

$$\begin{aligned}
\mathbb{E}\|\nabla f(z^r)\|^2 &= \frac{m^2}{\beta^2}\mathbb{E}\left\|\frac{-\beta}{m}\nabla f(z^r)\right\|^2 \\
&= \frac{m^2}{\beta^2}\mathbb{E}\left\|\frac{-\beta}{m}\nabla f(z^r) - (z^{r+1} - z^r) + (z^{r+1} - z^r)\right\|^2 \\
&\leq \frac{m^2}{\beta^2}\mathbb{E}\left\|\frac{-\beta}{m}\nabla f(z^r) - (z^{r+1} - z^r)\right\|^2 + \frac{m^2}{\beta^2}\mathbb{E}\|z^{r+1} - z^r\|^2 \\
&\overset{(5.22)}{\leq} \frac{2m^2}{\beta^2}\mathbb{E}\|z^{r+1} - z^r\|^2.
\end{aligned} \tag{5.38}$$

From the above equation one can conclude that

$$\mathbb{E}\|\nabla f(z^r)\|^2 \leq \frac{2m^2}{\beta^2}\mathbb{E}\|z^{r+1} - z^r\|^2 + \frac{8m^2}{\beta}\sum_{i=1}^{n}\frac{1}{\eta_i}\sum_{j=1}^{m}\left\|\frac{\partial f_i(z^{r-1})}{\partial z_j} - \frac{\partial f_i(y_{ij}^{r-2})}{\partial z_j}\right\|^2. \tag{5.39}$$

Further, from equation (5.25) we have

$$\frac{16m^2\mathbb{E}[Q^r - Q^{r+1}]}{\beta} \geq \frac{2m^2}{\beta^2}\mathbb{E}\|z^r - z^{r-1}\|^2 + \frac{8m^2}{\beta}\sum_{i=1}^{n}\frac{1}{\eta_i}\sum_{j=1}^{m}\left\|\frac{\partial f_i(z^{r-1})}{\partial z_j} - \frac{\partial f_i(y_{ij}^{r-2})}{\partial z_j}\right\|^2. \tag{5.40}$$

Combining two equations (5.39) and (5.40) we reach the following relation:

$$\mathbb{E}\|\nabla f(z^r)\|^2 \leq \frac{16m^2}{\beta}\mathbb{E}[Q^r - Q^{r+1}]. \tag{5.41}$$

Taking sum over both sides for $r = 1, \cdots, R$, (R is the total number of primal iterations) we obtain:

$$\sum_{r=1}^{R}\mathbb{E}\|\nabla f(z^r)\|^2 \leq \frac{16m^2}{\beta}\mathbb{E}[Q^1 - Q^{R+1}].$$

Now instead of taking the z^R, we pick a random number u uniformly from $\{1, 2, \cdots, R\}$ uniformly, and consider z^u as the primal solution. Using the definition of z^u, we have

$$\mathbb{E}\|\nabla f(z^u)\|^2 = \mathbb{E}_{\mathcal{F}^r}\left[\mathbb{E}_u[\|\nabla f(z^r)\|^2 \mid \mathcal{F}^r]\right] = \frac{1}{R}\sum_{r=1}^{R}\mathbb{E}_{\mathcal{F}^r}\|\nabla f(z^r)\|^2.$$

Therefore, we can finally conclude that:

$$\mathbb{E}\|\nabla f(z^u)\|^2 \leq \frac{16m^2}{\beta}\frac{\mathbb{E}[Q^1 - Q^{R+1}]}{R}. \tag{5.42}$$

The proof is complete.

5.7 Conclusion

In this chapter we propose a double stochastic primal-dual training framework. Using auxiliary variables we decouple the connections between layers and formulate an equality constrained optimization problem. We have shown that the proposed algorithm is able to compute stationary solution almost surely. Moreover, we demonstrate with simulation that our algorithm is able to train neural networks without using backpropagation. An early advantage over SGD is observed, based on which we further develop a warm-up strategy to boost SGD convergence speed by first running a few iterations of DSPD. The simulation result demonstrates the efficacy of such strategy.

CHAPTER 6. GENERAL CONCLUSION

In this dissertation, we present two lines of time-varying optimization framework. First line of work can be characterized as real-time optimization, where problem parameters are changing in real time. We propose dynamic algorithms based on alternating direction method of multipliers (ADMM). In the case where first order information is available, we design a perturbed ADMM algorithm that allows us to balance between convergence speed and solution accuracy. In the case where only objective function values are available, we design a zeroth order ADMM algorithm to solve the time-varying problem with just two-point estimation of original gradient. Both cases are proved to have the tracking ability. Specifically, we prove that our algorithms are able to converge to a neighbourhood of the optimal solution for each time instance. The neighbourhood radius is quantified as a function of optimal drift and changes in problem parameters. We apply our algorithms to power flow control problem and utilize real world data to demonstrate the efficacy of proposed algorithms. The real time setting can further be extended to the cases where static optimization has random error in updates. This is, again, applied to the power system problem, where a feedback controller for inverter-interfaced renewable energy sources (RESs) systems that drives the outputs of RESs to the optimal solution of convex surrogates of the AC OPF problem is developed.

The second line of work can be characterized as stochastic optimization where we aim at finding one solution that is close to all optimal solutions for each time instance (or each data point). Specifically, we consider the problem of training a feed-forward neural network. We apply the idea of splitting layers by adding auxiliary variables and construct an equality constrained optimization problem. Each time we randomly sample a mini-batch of the full dataset and utilize a double stochastic primal-dual decomposition algorithm to solve this problem. To the best of our knowledge, this is the first stochastic version primal dual training method. Convergence to stationary solution is established by designing specific stopping criteria for primal, dual updates and assuming Robinson's constraint qualification. Simulation results show that our proposed algorithm is able to fully explore hidden layer information and network structure, which in return gives an early advantage over vanilla SGD.

In the future, we want to further improve our time-varying framework in following directions:

Tracking task

- Explore more applications that can benefit from our real-time optimization work.
- Focus on theories of real-time nonconvex optimization problems.

Training task

- Incorporate various acceleration techniques to improve DSPD convergence speed.
- Explore systematic ways to reduce computation complexity of DSPD.

BIBLIOGRAPHY

[1] M. Grant, S. Boyd, and Y. Ye, “CVX: Matlab software for disciplined convex programming,” 2008.

[2] A. Simonetto and G. Leus, “Distributed asynchronous time-varying constrained optimization,” in *48th Asilomar Conference on Signals, Systems and Computers*, Nov. 2014, pp. 2142–2146.

[3] A. Bernstein, E. Dall’Anese, and A. Simonetto, “Online optimization with feedback,” *arXiv preprint arXiv:1804.05159*, 2018.

[4] E. Dall’Anese and A. Simonetto, “Optimal power flow pursuit,” *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 942–952, 2018.

[5] S. Rahili and W. Ren, “Distributed continuous-time convex optimization with time-varying cost functions,” *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1590–1605, 2017.

[6] M. Fazlyab, C. Nowzari, G. J. Pappas, A. Ribeiro, and V. M. Preciado, “Self-triggered time-varying convex optimization,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 3090–3097.

[7] M. J. Neely and H. Yu, “Online convex optimization with time-varying constraints,” *arXiv preprint arXiv:1702.04783*, 2017.

[8] S. Kar and J. M. Moura, “Gossip and distributed kalman filtering: Weak consensus under weak detectability,” *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1766–1784, 2011.

[9] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[10] A. Cherukuri, B. Gharesifard, and J. Cortes, “Saddle-point dynamics: conditions for asymptotic stability of saddle points,” *SIAM Journal on Control and Optimization*, vol. 55, no. 1, pp. 486–511, 2017.

[11] Y. Tang, K. Dvijotham, and S. Low, “Real-time optimal power flow,” *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2963–2973, 2017.

[12] E. Dall’Anese and A. Simonetto, “Optimal power flow pursuit,” *arXiv preprint arXiv:1601.07263*, 2016.

[13] M. Colombino, E. Dall’Anese, and A. Bernstein, “Online optimization as a feedback controller: Stability and tracking,” *arXiv preprint arXiv:1805.09877*, 2018.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[15] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, vol. 66, no. 3, pp. 889–916, 2016.

[16] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *Mathematical Programming*, vol. 162, no. 1-2, pp. 165–199, 2017.

[17] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[19] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[20] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International Conference on Machine Learning*, 2013, pp. 1139–1147.

[21] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[22] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[23] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, pp. 26–31, 2012.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] T. Rögnvaldsson, "On langevin updating in multilayer perceptrons," *Neural Computation*, vol. 6, no. 5, pp. 916–926, 1994.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[27] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Springer, 1998, pp. 9–50.

[28] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[29] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *arXiv preprint arXiv:1312.6120*, 2013.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[32] M. Carreira-Perpinan and W. Wang, “Distributed optimization of deeply nested systems,” in *Artificial Intelligence and Statistics*, 2014, pp. 10–19.

[33] G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel, and T. Goldstein, “Training neural networks without gradients: A scalable admm approach,” in *International Conference on Machine Learning*, 2016, pp. 2722–2731.

[34] Z. Zhang, Y. Chen, and V. Saligrama, “Efficient training of very deep neural networks for supervised hashing,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1487–1495.

[35] Z. Zhang and M. Brand, “Convergent block coordinate descent for training tikhonov regularized deep neural networks,” in *Advances in Neural Information Processing Systems*, 2017, pp. 1719–1728.

[36] Q. Ling and A. Ribeiro, “Decentralized dynamic optimization through the alternating direction method of multipliers.” *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1185–1197, 2014.

[37] X. Cao and K. Liu, “Dynamic sharing through the admm,” *arXiv preprint arXiv:1702.03874*, 2017.

[38] Y. Zhang, E. Dall’Anese, and M. Hong, “Dynamic admm for real-time optimal power flow,” in *Signal and Information Processing (GlobalSIP), 2017 IEEE Global Conference on*. IEEE, 2017, pp. 1085–1089.

[39] D. Hajinezhad and M. Hong, “Perturbed proximal primal dual algorithm for nonconvex nonsmooth optimization.”

[40] J. Koshal, A. Nedić, and U. V. Shanbhag, “Multiuser optimization: distributed algorithms and error analysis,” *SIAM Journal on Optimization*, vol. 21, no. 3, pp. 1046–1081, 2011.

[41] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[42] C. Daskalakis and I. Panageas, “The limit points of (optimistic) gradient descent in min-max optimization,” *arXiv preprint arXiv:1807.03907*, 2018.

[43] J. Dutta, K. Deb, R. Tulshyan, and R. Arora, “Approximate kkt points and a proximity measure for termination,” *Journal of Global Optimization*, vol. 56, no. 4, pp. 1463–1499, 2013.

[44] R. Andreani, G. Haeser, and J. M. Martínez, “On sequential optimality conditions for smooth constrained optimization,” *Optimization*, vol. 60, no. 5, pp. 627–641, 2011.

[45] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, “Projected gradient descent on riemannian manifolds with applications to online power system optimization,” in *Proc. of 54th Annual Allerton Conference on Communication, Control, and Computing*, Sep. 2016.

[46] B. Kroposki, E. Dall’Anese, A. Bernstein, Y. Zhang, and B.-M. Hodge, “Autonomous energy grids,” in *Hawaii International Conference on System Sciences*, Jan. 2018.

[47] D. Paccagnan, B. Gentile, F. Parise, M. Kamgarpour, and J. Lygeros, "Nash and wardrop equilibria in aggregative games with coupling constraints," 2017, [Online] Available at: https://arxiv.org/abs/1702.08789.

[48] A. Bernstein and E. Dall'Anese, "Linear power-flow models in multiphase distribution networks," in *The 7th IEEE Intl. Conference on Innovative Smart Grid Technologies*, Sep. 2017.

[49] S. Bolognani and F. Dörfler, "Fast power system analysis via implicit linearization of the power flow manifold," in *Proceedings of 53rd Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Oct. 2015.

[50] A. Flaxman, A. Kalai, and H. McMahan, "Online convex optimization in the bandit setting: gradient descent without a gradient," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2005, pp. 385–394.

[51] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multipoint bandit feedback." in *COLT*, 2010, pp. 28–40.

[52] D. Hajinezhad, M. Hong, and A. Garcia, "Zeroth order nonconvex multi-agent optimization over networks," *arXiv preprint arXiv:1710.09997*, 2017.

[53] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic iot management," *IEEE Internet of Things Journal*, 2018.

[54] Y. Zhang, E. Dall'Anese, and M. Hong, "Dynamic ADMM for real-time optimal power flow," in *IEEE Global Conference on Signal and Information Processing*, Nov 2017.

[55] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, pp. 1–40, 2011.

[56] A. Ajalloeian, A. Simonetto, and E. Dall'Anese, "Inexact online proximal-gradient method for time-varying convex optimization," *arXiv preprint arXiv:1910.02018*, 2019.

[57] Y. Liu, J. Bebic, B. Kroposki, J. de Bedout, and W. Ren, "Distribution system voltage performance analysis for high-penetration PV," in *IEEE Energy 2030 Conference*, Nov. 2008.

[58] A. Woyte, V. Van Thong, R. Belmans, and J. Nijs, "Voltage fluctuations on distribution level introduced by photovoltaic systems," *IEEE Transactions on Energy Conversion*, vol. 21, no. 1, pp. 202–209, 2006.

[59] E. Dall'Anese, S. V. Dhople, and G. B. Giannakis, "Photovoltaic inverter controller seeking AC optimal power flow solutions," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2809–2823, 2016.

[60] L. Gan and S. H. Low, "An online gradient algorithm for optimal power flow in radial networks," *IEEE Journal on Selected Areas in Communications*, 2016, to appear.

[61] A. Bernstein, L. Reyes-Chamorro, J. Le Boudec, and M. Paolone, "A composable method for real-time control of active distribution networks with explicit power setpoints. part I: Framework," *Electric Power Systems Research*, vol. 125, pp. 254 – 264, 2015.

[62] A. Bernstein, N. J. Bouman, and J.-Y. Le Boudec, "Design of resource agents with guaranteed tracking properties for real-time control of electrical grids," 2015, [Online] Available at: http://arxiv.org/abs/1511.08628.

[63] A. Jokić, M. Lazar, and P. Van den Bosch, "Real-time control of power systems using nodal prices," *International Journal of Electrical Power & Energy Systems*, vol. 31, no. 9, pp. 522–530, 2009.

[64] K. Hirata, J. P. Hespanha, and K. Uchida, "Real-time pricing leading to optimal operation under distributed decision makings," in *Proceedings of American Control Conference*, Portland, OR, June 2014.

[65] A. Cherukuri and J. Cortes, "Distributed coordination of ders with storage for dynamic economic dispatch," 2016, [Online] Available at: https://arxiv.org/abs/1605.00721.

[66] X. Ma and N. Elia, "A distributed continuous-time gradient dynamics approach for the active power loss minimizations," in *Proceedings of 51st Annual Allerton Conference on Communication, Control, and Computing*, UIUC, IL, USA, Oct. 2013.

[67] D. B. Arnold, M. Negrete-Pincetic, M. D. Sankur, D. M. Auslander, and D. S. Callaway, "Model-free optimal control of VAR resources in distribution systems: An extremum seeking approach," *IEEE Transactions on Power Systems*, 2015.

[68] G. Wang, V. Kekatos, A. J. Conejo, and G. B. Giannakis, "Ergodic energy management leveraging resource variability in distribution grids," *IEEE Transactions on Power Systems*, 2016.

[69] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

[70] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," *Journal of Scientific Computing*, pp. 1–28, 2012.

[71] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *arXiv preprint arXiv:1208.3922*, 2012.

[72] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, pp. 1401–1407, Apr. 1989.

[73] K. Christakou, J.-Y. Le Boudec, M. Paolone, and D.-C. Tomozei, "Efficient Computation of Sensitivity Coefficients of Node Voltages and Line Currents in Unbalanced Radial Electrical Distribution Networks," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 741–750, 2013.

[74] S. Guggilam, E. Dall'Anese, Y. Chen, S. Dhople, and G. B. Giannakis, "Scalable optimization methods for distribution networks with high PV integration," *IEEE Transactions on Smart Grid*, 2016, to appear.

[75] S. V. Dhople, S. S. Guggilam, and Y. C. Chen, "Linear approximations to ac power flow in rectangular coordinates," in *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*. IEEE, 2015, pp. 211–217.

[76] Y. Zhang, M. Hong, E. Dall'Anese, S. Dhople, and X. Zu, "Regulation of renewable energy sources to optimal power flow solutions using ADMM," in *American Control Conference*, May 2017.

[77] A. Jokić, M. Lazar, and P. P. Van den Bosch, "On constrained steady-state regulation: dynamic KKT controllers," *IEEE Transactions Automatic Control*, vol. 54, no. 9, pp. 2250–2254, Sep. 2009.

[78] A. Yazdani and R. Iravani, *Voltage-sourced converters in power systems: modeling, control, and applications*. John Wiley & Sons, 2010.

[79] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[80] S. H. Low, "Convex relaxation of optimal power flow part i: Formulations and equivalence," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 15–27, March 2014.

[81] E. Dall'Anese and S. Simonetto, "Optimal power flow pursuit," *IEEE Transactions on Smart Grid.*, 2016, [Online] Available at: http://arxiv.org/abs/1601.07263.

[82] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of Electronic Imaging*, vol. 16, no. 4, p. 049901, 2007.

[83] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in Neural Information Processing Systems*, 2014, pp. 2933–2941.

[84] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[85] J. Zeng, T. T.-K. Lau, S. Lin, and Y. Yao, "Block coordinate descent for deep learning: Unified convergence guarantees," *arXiv preprint arXiv:1803.00225*, 2018.

[86] T. T.-K. Lau, J. Zeng, B. Wu, and Y. Yao, "A proximal block coordinate descent algorithm for deep neural network training," *arXiv preprint arXiv:1803.09082*, 2018.

[87] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *The Proceeding of NIPS*, 2014.

[88] M. Schmidt, N. L. Roux, and F. Bach., "Minimizing finite sums with the stochastic average gradient," 2013, technical report, INRIA.

[89] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *the Proceedings of the Neural Information Processing (NIPS)*, 2013.

[90] D. Hajinezhad, M. Hong, T. Zhao, and Z. Wang, "NESTT: A nonconvex primal-dual splitting method for distributed and stochastic optimization," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 3215–3223.

[91] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola, "Stochastic variance reduction for nonconvex optimization," in *International Conference on Machine Learning*, 2016, pp. 314–323.

[92] Z. Allen-Zhu and E. Hazan, "Variance Reduction for Faster Non-Convex Optimization," in *Proceedings of the 33rd International Conference on Machine Learning*, ser. ICML, 2016.

[93] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[94] J. Konečnỳ, Z. Qu, and P. Richtárik, "Semi-stochastic coordinate descent," *Optimization Methods and Software*, vol. 32, no. 5, pp. 993–1005, 2017.

[95] D. Bertsekas, *Nonlinear Programming*. Athena Scientific Belmont, 1999.

[96] A. P. Ruszczyński and A. Ruszczynski, *Nonlinear Optimization*. Princeton university press, 2006, vol. 13.

[97] Y. Zhang and Z. Lu, "Penalty decomposition methods for rank minimization," in *Advances in Neural Information Processing Systems*, 2011, pp. 46–54.

[98] Z. Lu and Y. Zhang, "Sparse approximation via penalty decomposition methods," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2448–2478, 2013.

[99] A. F. Izmailov and M. V. Solodov, "Optimality conditions for irregular inequality-constrained problems," *SIAM Journal on Control and Optimization*, vol. 40, no. 4, pp. 1280–1295, 2002.

[100] Q. Shi, M. Hong, X. Fu, and T.-H. Chang, "Penalty dual decomposition method for nonsmooth nonconvex optimization," *arXiv preprint arXiv:1712.04767*, 2017.

[101] D. Hajinezhad and M. Hong, "Perturbed proximal primal dual algorithm for nonconvex nonsmooth optimization," *technical report*, 2018.

[102] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[103] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods, 2nd ed.* Belmont, MA: Athena Scientific, 1997.