2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Time-Varying Feedback Optimization for Quadratic Programs with Heterogeneous Gradient Step Sizes

Andrey Bernstein*, Joshua Comden*, Yue Chen*, Jing Wang*

***Abstract*— Online feedback-based optimization has become a promising framework for real-time optimization and control of complex engineering systems. This tutorial paper surveys the recent advances in the field as well as provides novel convergence results for primal-dual online algorithms with heterogeneous step sizes for different elements of the gradient. The analysis is performed for quadratic programs and the approach is illustrated on applications for adaptive step-size and model-free online algorithms, in the context of optimal control of modern power systems.**

I. Introduction

This tutorial paper is set in the context of designing real-time optimization and control approaches for efficient operation of complex engineering systems, such as power networks, transportation systems, and communication networks. To illustrate the main concepts, consider an *algebraic* model describing the input-output relation of the system at time t:

$$\boldsymbol{y}(t) = \boldsymbol{h}_t(\boldsymbol{x}(t)) \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ is a vector of system inputs; $\boldsymbol{y}(t) \in \mathbb{R}^m$ is the vector of system outputs; and $\boldsymbol{h}_t(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ is the system model[1] map. The efficient operation of the system is then formalized via a time-varying optimization problem[2]

$$\min_{\boldsymbol{x}\in\mathcal{X}(t), \boldsymbol{y}=\boldsymbol{h}_t(\boldsymbol{x})} f_t(\boldsymbol{y}) \tag{2}$$

where $\mathcal{X}(t)$ is a set representing input constraints (e.g., physical or engineering); and $f_t : \mathbb{R}^m \to \mathbb{R}$ is a function quantifying the system performance.

Despite the success of modern optimization theory, directly applying optimization methods to solve (2) in real time might be impractical for real engineering systems. The major challenges are imprecise knowledge of the system model (including external disturbances) as well as the computational complexity of solving (2) in real time.

Online feedback optimization (OFO) [1]–[8] is an emerging framework that aims at tackling the above challenges. The essential idea of OFO boils down to implementing the optimization algorithms in a feedback loop with the actual system. To illustrate the idea, consider first a standard (feedforward) optimization approach to seek solutions (or stationary points) of (2) via a projected-gradient method that at each iteration $k = 0, 1, ..$ performs

$$\boldsymbol{x}^{(k+1)} = \mathsf{Proj}_{\mathcal{X}(t)}\Big\{\boldsymbol{x}^{(k)} - \alpha(\boldsymbol{J}_t^{(k)})^{\intercal}\nabla_{\boldsymbol{y}} f_t(h_t(\boldsymbol{x}^{(k)}))\Big\}, \tag{3}$$

where $\boldsymbol{J}_t^{(k)} := \boldsymbol{J}_{\boldsymbol{h}_t}(\boldsymbol{x}^{(k)})$ is the Jacobian matrix of $\boldsymbol{h}_t$; $\mathsf{Proj}_{\mathcal{X}}(\boldsymbol{z}) := \arg\min_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{z} - \boldsymbol{x}\|_2$ is the projection operator; and $\alpha > 0$ is the step size. In the OFO framework, (3) is replaced with an *online* update rule

$$\boldsymbol{x}^{(k+1)} = \mathsf{Proj}_{\mathcal{X}^{(k)}}\Big\{\boldsymbol{x}^{(k)} - \alpha(\boldsymbol{J}^{(k)})^{\intercal}\nabla_{\boldsymbol{y}} f^{(k)}(\widehat{\boldsymbol{y}}^{(k)})\Big\}, \tag{4}$$

performed at each discrete time instance t_k (the iteration index is now identified with a time step). In (4), $\mathcal{X}^{(k)} := \mathcal{X}(t_k)$, $\boldsymbol{J}^{(k)} := \boldsymbol{J}_{\boldsymbol{h}_{t_k}}(\boldsymbol{x}^{(k)})$, $f^{(k)} := f_{t_k}$; and $\widehat{\boldsymbol{y}}^{(k)}$ is the measurement of the system output $\boldsymbol{h}_{t_k}(\boldsymbol{x}^{(k)})$ representing the feedback. Note that (4) eliminates the explicit dependence on the system model by the virtue of feedback but still requires the real-time knowledge of the system Jacobian $\boldsymbol{J}^{(k)}$.

The existing OFO methods can be classified according to their application scenarios and implementation details; see, e.g., [8], [9] for a survey. In particular, the following aspects can be used to classify the OFO methods:

1) **Model type: algebraic or dynamic.** This paper focuses on the algebraic model (1). For OFO methods with dynamic system models, e.g., see [5], [8].
2) **Model information.** To implement (4), the gradient of $F^{(k)}(\boldsymbol{x}) := f^{(k)}(\boldsymbol{h}_{t_k}(\boldsymbol{x}))$, or approximation thereof, needs to be computed. There are two main approaches in the literature:
   a) **Jacobian estimation.** The Jacobian matrix $\boldsymbol{J}^{(k)}$ is estimated either based on a partial/approximate system model (e.g., linearization of (1) around a given operating point) [4] or leveraging data-driven/machine-learning-based approaches [10]. Denoting by $\widehat{\boldsymbol{J}}$ the approximate Jacobian, the following proxy is then used in (4) for the gradient of the objective function:

$$\widehat{\nabla} F^{(k)} := \widehat{\boldsymbol{J}}^{\intercal}\nabla_{\boldsymbol{y}} f^{(k)}(\widehat{\boldsymbol{y}}^{(k)}) \tag{5}$$

This work was authored by the National Renewable Energy Laboratory, managed and operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

*A. Bernstein, J. Comden, Y. Chen, and J. Wang are with the Power Systems Engineering Center, National Renewable Energy Laboratory, Golden, CO 80401, USA. andrey.bernstein@nrel.gov, joshua.comden@nrel.gov, yue.chen@nrel.gov, jing.wang@nrel.gov

[1] For example, $\boldsymbol{h}_t(\boldsymbol{x}(t)) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{d}(t))$ where $\boldsymbol{d}(t)$ is a disturbance process.

[2] For simplicity, we assume a stylized optimization where the performance is measured only with respect to the output $\boldsymbol{y}$; a general constrained optimization problem is considered in Section II.

b) **Zero-order approximation.** The gradient of $F^{(k)}$ is directly approximated using one of the many zero-order approximation approaches. For example, in [6], a two-point approximation was used:

$$\widehat{\nabla} F^{(k)} := \frac{1}{2\epsilon}\boldsymbol{\xi}\left[F^{(k)}(\boldsymbol{x}^{(k)}+\epsilon\boldsymbol{\xi}) - F^{(k)}(\boldsymbol{x}^{(k)}-\epsilon\boldsymbol{\xi})\right] \quad (6)$$

where $\boldsymbol{\xi} \in \mathbb{R}^n$ is a perturbation (or exploration) vector, and $\epsilon > 0$ is a (small) scalar. Other approaches include the one-point estimation [11] and the residual one-point estimation [8]. In any case, the obtained proxy for the gradient only requires measurements of the system output feedback.

3) **Algorithm/problem type.** Here the distinction is usually between primal and primal-dual (or dual) methods. Primal methods such as (4) are effective in solving optimization problems where no output constraints are present or when the projection on the constraints can be computed efficiently and centrally [8]. Primal-dual methods can deal efficiently with complicated output constraints with the caveat of only asymptotic constraint satisfaction. Also, primal-dual methods better lend themselves to distributed implementation [4].
4) **Problem time variability.** In static settings, where the objective and constraints in (2) do not vary with time, the proposed algorithms typically are shown to converge to the optimal solution/stationary point, or neighborhood thereof. In dynamic settings, the algorithms are typically designed to track the time-varying optimal solution/stationary point of (2) up to a well-defined error.
5) **Algorithm step size.** Most of the literature, with the exception of [7], [12], focuses on algorithms with a uniform (single) step size parameter; cf. α in (4). This assumption simplifies significantly the convergence analysis, especially in the constrained time-varying case. In practice, however, it is beneficial to have different step sizes for different elements of $\widehat{\nabla} F^{(k)}$. Moreover, adaptive and heterogeneous step sizes have been shown to improve numerical performance of the OFO algorithms on specific test cases [12].

Contributions. In addition to surveying the recent advances of OFO methods, this paper aims at addressing the last challenge above, namely providing convergence guarantees for time-varying OFO methods with heterogeneous step sizes. Specifically, we consider the online OFO algorithm of the form

$$\boldsymbol{x}^{(k+1)} = \mathsf{Proj}_{\mathcal{X}^{(k)}}\left\{\boldsymbol{x}^{(k)} - \alpha\boldsymbol{\Gamma}^{(k)}\widehat{\nabla} F^{(k)}\right\}, \quad (7)$$

where $\boldsymbol{\Gamma}^{(k)} = \mathsf{diag}(\boldsymbol{\gamma}^{(k)})$ is a diagonal positive definite matrix. Most of our analysis focuses on quadratic programs (QPs) and linear algebraic system models due to reasons that will become apparent next.

While the prototypical problem (2) and algorithm (7) are used here to present the main concepts, the paper considers constrained problems and online primal-dual solution methods. Finally, to illustrate the performance of the developed methods, we apply them to the problem of optimal operation of power distribution network with distributed energy resources.

Organization: The rest of the paper is organized as follows. In Section II, we describe the essentials of the OFO framework. In Section III, we introduce the algorithms with heterogeneous step sizes and provide conditions for convergence. In Section IV, we showcase two applications of heterogeneous step-size OFO in the context of power systems optimization. Finally, we conclude in Section V.

II. Time-Varying OFO Framework

A. Target Optimization Problem

Consider a network of N systems, whose objectives are defined via the following optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{y}\in\mathbb{R}^m} \quad f_0^{(k)}(\boldsymbol{y}) + \sum_{i=1}^{N} f_i^{(k)}(\boldsymbol{x}_i) \quad (8a)$$

$$\text{subject to}: \quad \boldsymbol{x}_i \in \mathcal{X}_i^{(k)}, \, i = 1, \dots, N \quad (8b)$$

$$\boldsymbol{y} = \boldsymbol{h}^{(k)}(\boldsymbol{x}) \quad (8c)$$

$$g_j^{(k)}(\boldsymbol{y}) \leq 0, \, j = 1, \dots, M \quad (8d)$$

where $\mathcal{X}_i^{(k)} \subset \mathbb{R}^{n_i}$, with $\sum_{i=1}^N n_i = n$, are convex sets defining input constraints for system i; and $\boldsymbol{y} = \boldsymbol{h}^{(k)}(\boldsymbol{x})$ is the algebraic model for the networked system output. This paper mostly focuses on the linear (or linearized) case

$$\boldsymbol{h}^{(k)}(\boldsymbol{x}) := \boldsymbol{C}\boldsymbol{x} + \boldsymbol{U}\boldsymbol{w}^{(k)} \in \mathbb{R}^m \quad (9)$$

where $\boldsymbol{C} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{U} \in \mathbb{R}^{m\times w}$ are given model parameters, and $\boldsymbol{w}^{(k)} \in \mathbb{R}^w$ is a vector of time-varying exogenous (uncontrollable) inputs. For example, in the power systems case considered in Section IV, $\boldsymbol{h}^{(k)}(\boldsymbol{x})$ is the linear power-flow model.

In (8), $f_0^{(k)}(\boldsymbol{h}^{(k)}(\boldsymbol{x}))$ is a convex function (in terms of $\boldsymbol{x}$) defining the objective in terms of the outputs; and $f_i^{(k)}(\boldsymbol{x}_i)$ are convex functions defining the objectives of subsystems in terms of their input. Finally, $g_j^{(k)}(\boldsymbol{h}^{(k)}(\boldsymbol{x}))$ are convex (in terms of $\boldsymbol{x}$) output-constraints functions.

To simplify notation, we let:

$$\boldsymbol{g}^{(k)}(\boldsymbol{h}^{(k)}(\boldsymbol{x})) := \left[g_1^{(k)}(\boldsymbol{h}^{(k)}(\boldsymbol{x})), \dots, g_M^{(k)}(\boldsymbol{h}^{(k)}(\boldsymbol{x}))\right]^\intercal$$

$$\phi^{(k)}(\boldsymbol{x}) := \sum_i f_i^{(k)}(\boldsymbol{x}_i)$$

$$f^{(k)}(\boldsymbol{x}) := \phi^{(k)}(\boldsymbol{x}) + f_0^{(k)}(\boldsymbol{h}^{(k)}(\boldsymbol{x})).$$

Let $\boldsymbol{\lambda} \in \mathbb{R}_+^M$ denote the vector of dual variables associated with (8d), and

$$\mathcal{L}^{(k)}(\boldsymbol{x}, \boldsymbol{\lambda}) := f^{(k)}(\boldsymbol{x}) + \boldsymbol{\lambda}^\intercal \boldsymbol{g}^{(k)}(\boldsymbol{h}^{(k)}(\boldsymbol{x})) \quad (10)$$

denote the associated Lagrangian function. The *regularized* Lagrangian is given by [13]:

$$\mathcal{L}_{p,d}^{(k)}(\boldsymbol{x}, \boldsymbol{\lambda}) := \mathcal{L}^{(k)}(\boldsymbol{x}, \boldsymbol{\lambda}) + \frac{p}{2}\|\boldsymbol{x}\|_2^2 - \frac{d}{2}\|\boldsymbol{\lambda}\|_2^2 \quad (11)$$

for some $p > 0$ and $d > 0$.

Consider then the saddle-point problem:

$$\max_{\boldsymbol{\lambda} \in \mathcal{D}^{(k)}} \min_{\boldsymbol{x} \in \mathcal{X}^{(k)}} \mathcal{L}_{p,d}^{(k)}(\boldsymbol{x}, \boldsymbol{\lambda}) \qquad k \in \mathbb{N} \tag{12}$$

where $\mathcal{X}^{(k)} := \mathcal{X}_1^{(k)} \times \ldots \times \mathcal{X}_N^{(k)}$ and $\mathcal{D}^{(k)}$ is a convex and compact set that contains the optimal dual variable of (12) [4]. We interpret the optimal trajectory $\boldsymbol{z}^{(*,k)} := \{\boldsymbol{x}^{(*,k)}, \boldsymbol{\lambda}^{(*,k)}\}_{k \in \mathbb{N}}$ of (12) as the *reference* trajectory which the online algorithms are designed to track. Since $\mathcal{L}_{p,d}^{(k)}(\boldsymbol{x}, \boldsymbol{\lambda})$ is strongly convex in $\boldsymbol{x}$ and strongly concave in $\boldsymbol{\lambda}$ by design, the optimizer $\boldsymbol{z}^{(*,k)}$ of (12) is unique. However, it typically would be different from any saddle point of the Lagrangian $\mathcal{L}^{(k)}(\boldsymbol{x}, \boldsymbol{\lambda})$ (and therefore from any solution of (8)); the difference can be bounded as, e.g., in [13], as a function of the regularization parameters p and d.

The following assumptions are typically imposed[3] on the different elements of (8) to establish convergence/tracking guarantees.

Assumption 1. Slater's condition holds at each time step k.

Assumption 2. The functions $f_0^{(k)}(\boldsymbol{y})$ and $f_i^{(k)}(\boldsymbol{x}_i)$ are convex and continuously differentiable. The maps $\nabla\phi^{(k)}(\boldsymbol{x})$, $\nabla f_0^{(k)}(\boldsymbol{y})$, and $\nabla^2 f_0^{(k)}(\boldsymbol{y})$ are Lipschitz continuous.

Assumption 3. For each $j = 1, \ldots, M$, the function $g_j^{(k)}(\boldsymbol{y})$ is convex and continuously differentiable. Moreover, $\nabla g_j^{(k)}(\boldsymbol{y})$ and $\nabla^2 g_j^{(k)}(\boldsymbol{y})$ are Lipschitz continuous.

Assumption 4. There exists a constant e_f such that for all k and $\boldsymbol{x}, \boldsymbol{y}$:

$$\begin{aligned}
&\|\nabla f_0^{(k)}(\boldsymbol{y}) - \nabla f_0^{(k-1)}(\boldsymbol{y})\| \le e_f, \\
&\|\nabla \phi^{(k)}(\boldsymbol{x}) - \nabla \phi^{(k-1)}(\boldsymbol{x})\| \le e_f, \\
&\|\nabla g_j^{(k)}(\boldsymbol{y}) - \nabla g_j^{(k-1)}(\boldsymbol{y})\| \le e_f, \quad j = 1, \ldots, M.
\end{aligned}$$

B. Feedback Primal-Dual Methods

To track the reference trajectory $\{\boldsymbol{z}^{(*,k)}\}$, a feedback-based primal-dual method is typically used in the OFO framework [1], [4]:

$$\boldsymbol{x}^{(k+1)} = \mathsf{Proj}_{\mathcal{X}^{(k)}} \left\{ \boldsymbol{x}^{(k)} - \alpha \widehat{\nabla}_{\boldsymbol{x}} \mathcal{L}_{p,d}^{(k)} \right\} \tag{13a}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathsf{Proj}_{\mathcal{D}^{(k)}} \left\{ \boldsymbol{\lambda}^{(k)} + \alpha \widehat{\nabla}_{\lambda} \mathcal{L}_{p,d}^{(k)} \right\} \tag{13b}$$

where $\alpha > 0$ is a constant step size, and $\widehat{\nabla}_{\boldsymbol{x}} \mathcal{L}_{p,d}^{(k)}$ and $\widehat{\nabla}_{\lambda} \mathcal{L}_{p,d}^{(k)}$ are feedback-based proxies for the gradients of the regularized Lagrangian (cf. (5) and (6)). In particular, if the system Jacobian matrix $\boldsymbol{C}$ (or estimation thereof) is used, we have that

$$\begin{aligned}
\widehat{\nabla}_{\boldsymbol{x}} \mathcal{L}_{p,d}^{(k)} = {} & \nabla_{\boldsymbol{x}} \phi^{(k)}(\boldsymbol{x}^{(k)}) + \boldsymbol{C}^\intercal \nabla_{\boldsymbol{y}} f_0^{(k)}(\widehat{\boldsymbol{y}}^{(k)}) \\
& + \sum_{j=1}^{M} \lambda_j^{(k)} \boldsymbol{C}^\intercal \nabla g_j^{(k)}(\widehat{\boldsymbol{y}}^{(k)}) + p\boldsymbol{x}^{(k)}
\end{aligned} \tag{14a}$$

$$\widehat{\nabla}_{\lambda} \mathcal{L}_{p,d}^{(k)} = \boldsymbol{g}^{(k)}(\widehat{\boldsymbol{y}}^{(k)}) - d\boldsymbol{\lambda}^{(k)} \tag{14b}$$

where $\widehat{\boldsymbol{y}}^{(k)}$ is a measurement of $\boldsymbol{h}^{(k)}(\boldsymbol{x}^{(k)})$. Alternatively, a zero-order proxy can be used for $\widehat{\nabla}_{\boldsymbol{x}} \mathcal{L}_{p,d}^{(k)}$ instead of (14a). For example, using a two-point gradient estimation (6), we have [6]:

$$\begin{aligned}
\widehat{\nabla}_{\boldsymbol{x}} \mathcal{L}_{p,d}^{(k)} := {} & \nabla_{\boldsymbol{x}} \phi^{(k)}(\boldsymbol{x}^{(k)}) + p\boldsymbol{x}^{(k)} \\
& + \frac{1}{2\epsilon} \boldsymbol{\xi}^{(k)} \left[f_0^{(k)}(\widehat{\boldsymbol{y}}_+^{(k)}) - f_0^{(k)}(\widehat{\boldsymbol{y}}_-^{(k)}) \right] \\
& + \frac{1}{2\epsilon} \boldsymbol{\xi}^{(k)} (\boldsymbol{\lambda}^{(k)})^\intercal \left[\boldsymbol{g}^{(k)}(\widehat{\boldsymbol{y}}_+^{(k)}) - \boldsymbol{g}^{(k)}(\widehat{\boldsymbol{y}}_-^{(k)}) \right],
\end{aligned} \tag{15}$$

where $\{\boldsymbol{\xi}^{(k)}\}$ is an exploration process (either deterministic [11] or stochastic [14]); $\epsilon > 0$ is a (small) scalar controlling the magnitude of exploration; and $\widehat{\boldsymbol{y}}_\pm^{(k)}$ are the measurements of the system output at $\boldsymbol{x}_\pm^{(k)} := \boldsymbol{x}^{(k)} \pm \epsilon \boldsymbol{\xi}^{(k)}$.

In this paper, we are focusing on the *distributable* case that is quantified by the following assumption.

Assumption 5. The target optimization problem is distributable in the sense that the associated Lagrangian function, and regularizations and approximations thereof, are decomposable across different subsystems, i.e.:

$$\nabla_{\boldsymbol{x}_i} \mathcal{L}_{p,d}^{(k)}(\boldsymbol{x}, \boldsymbol{\lambda}) = \nabla_{\boldsymbol{x}_i} \mathcal{L}_{i,p,d}^{(k)}(\boldsymbol{x}_i, \boldsymbol{\lambda})$$

for all i.

Under Assumption 5, the primal step of the OFO algorithm (13a) can be written as

$$\boldsymbol{x}_i^{(k+1)} = \mathsf{Proj}_{\mathcal{X}_i^{(k)}} \left\{ \boldsymbol{x}_i^{(k)} - \alpha \widehat{\nabla}_{\boldsymbol{x}_i} \mathcal{L}_{p,d}^{(k)} \right\}, \quad i = 1, \ldots, N. \tag{16}$$

Note that (16) and (13b) can be implemented in a distributed, gather-and-broadcast approach, wherein (13b) is computed centrally and (16) is computed locally at each subsystem; see, e.g., [15] for details.

C. Key Elements of Convergence Analysis for Time-Varying OFO

In this section, we present typical results for tracking the reference trajectory $\{\boldsymbol{z}^{(*,k)}\}$ and discuss the key elements in the their proofs. Let e_y denote the worst-case error in the output feedback measurement. Then the following result holds [4], [6].

Theorem 1. *Let $\boldsymbol{z}^{(k)} := \{\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}^{(k)}\}_{k \in \mathbb{N}}$. Under Assumptions 1-4, there exists $\bar{\alpha} > 0$ such that if the step size α satisfies $0 < \alpha < \bar{\alpha}$, the sequence $\{\boldsymbol{z}^{(k)}\}$ converges Q-linearly to $\{\boldsymbol{z}^{(*,k)}\}$ up to an asymptotic error bound given by:*

$$\limsup_{k \to \infty} \|\boldsymbol{z}^{(k)} - \boldsymbol{z}^{(*,k)}\|_2 \le \frac{\alpha \epsilon_\phi + \sigma}{1 - c} \tag{17}$$

where $c < 1$, $\sigma := \max_k \|\boldsymbol{z}^{(*,k+1)} - \boldsymbol{z}^{(*,k)}\|_2$, and $\epsilon_\phi = O(e_y)$ when the estimated gradient (14) *is used; or $\epsilon_\phi = O(\alpha + \epsilon^2 + e_f + e_y)$ when the estimated gradient* (15) *is used.*

To prove such a theorem, standard time-varying fixed-point iteration analysis is typically used; see, e.g., [16]. To

[3] We list these assumptions here for completeness; not all the assumptions are necessary for specific convergence results.

this end, the following two properties of the primal-dual operator

$$\Phi^{(k)}(\boldsymbol{z}) := \begin{bmatrix} \nabla_{\boldsymbol{x}} \mathcal{L}_{p,d}^{(k)}(\boldsymbol{z}) \\ -\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{p,d}^{(k)}(\boldsymbol{z}) \end{bmatrix}, \quad \boldsymbol{z} := (\boldsymbol{x}, \boldsymbol{\lambda}) \tag{18}$$

are established.

1) Fixed-Point of the Algorithm is the Saddle Point: Consider the fixed-point equation associated with the standard primal-dual algorithm applied to $\mathcal{L}_{p,d}^{(k)}$:

$$\boldsymbol{z} = \mathsf{Proj}_{\mathcal{X}^{(k)} \times \mathcal{D}^{(k)}} \Big\{ \boldsymbol{z} - \alpha \Phi^{(k)}(\boldsymbol{z}) \Big\}. \tag{19}$$

It is well-known [1], [4] that under the assumptions in this paper, the unique fixed point of (19) coincides with the unique saddle point of (12).

2) Strong Monotonicity of $\Phi^{(k)}$: Under the assumptions in this paper, the operator $\Phi^{(k)}$ is strongly monotone [1], [4]. Namely, there exists $\eta > 0$ such that

$$\Big(\Phi^{(k)}(\boldsymbol{x}_1) - \Phi^{(k)}(\boldsymbol{x}_2) \Big)^{\mathsf{T}} (\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq \eta \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2 \tag{20}$$

for all $\boldsymbol{x}_1, \boldsymbol{x}_2$ and all k.

III. Algorithms with Heterogeneous Step Sizes

In this section, we illustrate the fact that even in the simple case of (primal) projected gradient method, heterogeneous step sizes might lead to convergence to a suboptimal point. We then discuss the conditions under which convergence to the optimum can be guaranteed and extend the results to primal-dual methods. Most of the analysis in this section is restricted to target optimization problems (8) which can be written in the QP form[4]:

$$\min_{\boldsymbol{x} \in \mathcal{X}} \; f(\boldsymbol{x}) \tag{21a}$$

$$\text{subject to}: \; \boldsymbol{g}(\boldsymbol{x}) \leq 0 \tag{21b}$$

where

$$f(\boldsymbol{x}) = \frac{1}{2} \boldsymbol{x}^{\mathsf{T}} \boldsymbol{A} \boldsymbol{x} + \boldsymbol{b}^{\mathsf{T}} \boldsymbol{x} + c, \quad \boldsymbol{A} \succcurlyeq 0 \tag{22}$$

and

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{d}. \tag{23}$$

We note that QPs capture a large set of practical problems that are of interest in our application domains. For example, the linearized optimal power flow problem discussed in Section IV is a QP. Extension of the results to more general problems is left for future research.

A. Projected Gradient Method

We first consider the case where the constraint (21b) is absent and analyze the convergence of the projected gradient algorithm

$$\boldsymbol{x}^{(k+1)} = \mathsf{Proj}_{\mathcal{X}} \Big\{ \boldsymbol{x}^{(k)} - \alpha \boldsymbol{\Gamma}^{(k)} \nabla f(\boldsymbol{x}^{(k)}) \Big\}, \tag{24}$$

where $\boldsymbol{\Gamma}^{(k)}$ is a diagonal positive definite matrix of step sizes and $\alpha > 0$ is the step-size scaling factor. We next discuss the two key properties for convergence analysis presented in Section II-C.

1) Optimality of Fixed Points: Note that without the projection, this algorithm converges to a solution of (21) under standard conditions on the scaling factor α (e.g., using the analysis in [16]). However, this is not the case with the projection in general. Specifically, the set of fixed points of

$$\boldsymbol{x} = \mathsf{Proj}_{\mathcal{X}} \Big\{ \boldsymbol{x} - \alpha \boldsymbol{\Gamma}^{(k)} \nabla f(\boldsymbol{x}) \Big\}, \tag{25}$$

does not necessarily coincide with the set of optimal points, as the following example shows.

Example 1. Suppose that $\mathcal{X} := \Big\{ \boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{1}^{\mathsf{T}} \boldsymbol{x} \geq 8 \Big\}$, $\boldsymbol{A} := \boldsymbol{I}$, $\boldsymbol{b} := \boldsymbol{0}$, and $c := 0$. If $\boldsymbol{\Gamma}^{(k)}(\boldsymbol{x}) := \boldsymbol{I}$, then the fixed point is $[4 \;\; 4]^{\mathsf{T}}$, and is on the boundary of $\mathcal{X}$ with the gradient of the objective function being $[4 \;\; 4]^{\mathsf{T}}$. It also happens to be optimal with the objective value of 16. However, if instead $\boldsymbol{\Gamma}^{(k)}$ is slightly perturbed away from $\boldsymbol{I}$, i.e., $\boldsymbol{\Gamma}^{(k)}(\boldsymbol{x}) := \mathsf{diag}([\frac{3}{4} \;\; \frac{5}{4}]^{\mathsf{T}})$, then the fixed point is $[5 \;\; 3]^{\mathsf{T}}$, and is on the boundary of $\mathcal{X}$ with the gradient of the objective function being $[5 \;\; 3]^{\mathsf{T}}$. Also, it is not optimal with the objective value of 17. □

To tackle this, we slightly redefine the projected gradient algorithm as follows:

$$\boldsymbol{x}^{(k+1)} = \mathsf{Proj}_{\mathcal{X}} \Big\{ \boldsymbol{x}^{(k)} - \alpha \boldsymbol{\Gamma}^{(k)}(\boldsymbol{x}^{(k)}) \nabla f(\boldsymbol{x}^{(k)}) \Big\}, \tag{26}$$

where $\boldsymbol{\Gamma}^{(k)}(\boldsymbol{x})$ is now defined via:

$$\boldsymbol{\Gamma}^{(k)}(\boldsymbol{x}) := \begin{cases} \mathsf{diag}(\boldsymbol{\gamma}^{(k)}) & \text{if } \boldsymbol{x} - \alpha \mathsf{diag}(\boldsymbol{\gamma}^{(k)}) \nabla f(\boldsymbol{x}) \in \mathcal{X}, \\ \boldsymbol{I} & \text{otherwise.} \end{cases} \tag{27}$$

We then have the following.

Proposition 1. *For each k, the set of fixed points of*

$$\boldsymbol{x} = \mathsf{Proj}_{\mathcal{X}} \Big\{ \boldsymbol{x} - \alpha \boldsymbol{\Gamma}^{(k)}(\boldsymbol{x}) \nabla f(\boldsymbol{x}) \Big\} \tag{28}$$

coincides with the set of fixed points of

$$\boldsymbol{x} = \mathsf{Proj}_{\mathcal{X}} \Big\{ \boldsymbol{x} - \alpha \nabla f(\boldsymbol{x}) \Big\}. \tag{29}$$

Therefore, the set of fixed points of (28) *coincides with the set of the optimal solutions to* $\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$.

The proof leverages the following Lemma that follows from the standard arguments of optimiality of convex problems.

Lemma 1. *If* $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$ *with* $\nabla f(\boldsymbol{x}^*) \neq \boldsymbol{0}$, *then:*

$$\boldsymbol{x}^* - \alpha \boldsymbol{\Gamma} \nabla f(\boldsymbol{x}^*) \notin \mathcal{X} \tag{30}$$

for any diagonal positive definite matrix $\boldsymbol{\Gamma}$ *and any* $\alpha > 0$.

Proof of Proposition 1. Let $\boldsymbol{x}^*$ satisfy (28). If $\boldsymbol{x}^* - \alpha \mathsf{diag}(\boldsymbol{\gamma}_k) \nabla f(\boldsymbol{x}^*) \in \mathcal{X}$ then $\boldsymbol{x}^* = \boldsymbol{x}^* - \alpha \mathsf{diag}(\boldsymbol{\gamma}_k) \nabla f(\boldsymbol{x}^*)$

[4] In this section, we omit the time variability of the optimization problem for brevity and focus on static problems; however, the general framework applies to time-varying problems as mentioned in Section III-C.

and $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$. If $\boldsymbol{x}^* - \alpha \mathsf{diag}(\boldsymbol{\gamma}_k)\nabla f(\boldsymbol{x}^*) \notin \mathcal{X}$, then $\boldsymbol{x}^* = \mathsf{Proj}_{\mathcal{X}}\Big\{\boldsymbol{x}^* - \alpha\nabla f(\boldsymbol{x}^*)\Big\}$. Therefore, $\boldsymbol{x}^*$ satisfies (29).

Now let $\boldsymbol{x}^*$ satisfy (29). If $\nabla f(\boldsymbol{x}^*) = \mathbf{0}$, $\boldsymbol{x}^*$ clearly satisfies (28). If $\nabla f(\boldsymbol{x}^*) \neq \mathbf{0}$, then using Lemma 1, we have that $\boldsymbol{x}^* - \alpha \mathsf{diag}(\boldsymbol{\gamma}^{(k)})\nabla f(\boldsymbol{x}^*) \notin \mathcal{X}$, and $\boldsymbol{x}^*$ satisfies (28) with $\boldsymbol{\Gamma}^{(k)}(\boldsymbol{x}^*) = \boldsymbol{I}$. □

2) Strong Monotonicity: Consider the operator

$$\Phi(\boldsymbol{x}) := \boldsymbol{\Gamma}\nabla f(\boldsymbol{x}). \tag{31}$$

The following example shows that $\Phi(\boldsymbol{x})$ is not necessarily monotone even in the case of quadratic and strongly convex objective (22).

Example 2. Suppose that $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{A}\boldsymbol{x}$ for some positive definite and symmetric matrix $\boldsymbol{A}$. Clearly, f is strongly convex and we have

$$\Phi(\boldsymbol{x}) = \boldsymbol{\Gamma}\boldsymbol{A}\boldsymbol{x}.$$

To check if $\Phi(\boldsymbol{x})$ is strongly monotone, we need to establish at least the monotonicity of $\Phi(\boldsymbol{x})$ in the sense that for any $\boldsymbol{x}_1, \boldsymbol{x}_2$,

$$(\Phi(\boldsymbol{x}_1) - \Phi(\boldsymbol{x}_2))^\mathsf{T}(\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq 0$$

which is equivalent to

$$(\boldsymbol{x}_1 - \boldsymbol{x}_2)^\mathsf{T}(\boldsymbol{A}\boldsymbol{\Gamma})(\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq 0.$$

The latter holds true if and only if $0.5(\boldsymbol{\Gamma}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{\Gamma})$ is positive semidefinite. However, this is not the case in general. For example, consider

$$\boldsymbol{A} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \boldsymbol{\Gamma} = \begin{bmatrix} \delta & 0 \\ 0 & 1 \end{bmatrix}.$$

If $\delta = 20$, we have that

$$\frac{1}{2}\left(\boldsymbol{\Gamma}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{\Gamma}\right) = \begin{bmatrix} 40 & -10.5 \\ -10.5 & 2 \end{bmatrix}$$

with eigenvalues 42.7 and -0.7. In this case, the more unbalanced $\boldsymbol{\Gamma}$ is (namely, the larger δ is), the smaller is the second eigenvalue. Specifically, for $\delta \geq 14$, the matrix $0.5(\boldsymbol{\Gamma}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{\Gamma})$ becomes indefinite. □

To enforce strong monotonicity of $\Phi(x)$, we use the regularization approach similar to (11). In particular, we consider the following regularized objective function:

$$\widetilde{f}(\boldsymbol{x}) := f(\boldsymbol{x}) + \frac{p}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{\Gamma}^{-1}\boldsymbol{x} \tag{32}$$

for some $p > 0$. Consider the operator

$$\begin{aligned} \widetilde{\Phi}(\boldsymbol{x}) &:= \boldsymbol{\Gamma}\nabla\widetilde{f}(\boldsymbol{x}) && (33) \\ &= (\boldsymbol{\Gamma}\boldsymbol{A} + p\boldsymbol{I})\boldsymbol{x} + \boldsymbol{\Gamma}\boldsymbol{b}. && (34) \end{aligned}$$

We then have the following result to esnure strong monotonicty of $\widetilde{\Phi}$.

Proposition 2. *Let*

$$\boldsymbol{M} := \frac{1}{2}\Big(\boldsymbol{\Gamma}\boldsymbol{A} + \boldsymbol{A}\boldsymbol{\Gamma}\Big) \tag{35}$$

and let $\lambda_{\min}$ *be the smallest eigenvalue of* $\boldsymbol{M}$*. If* $p > \max\{0, -\lambda_{\min}\}$*, then* $\widetilde{\Phi}$ *is strongly monotone with parameter* $\eta := p + \min\{0, \lambda_{\min}\}$*.*

Proof. We have that

$$\begin{aligned} &\left(\widetilde{\Phi}(\boldsymbol{x}_1) - \widetilde{\Phi}(\boldsymbol{x}_2)\right)^\mathsf{T}(\boldsymbol{x}_1 - \boldsymbol{x}_2) \\ &= \left[\left(\boldsymbol{\Gamma}\boldsymbol{A} + p\boldsymbol{I}\right)(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right]^\mathsf{T}(\boldsymbol{x}_1 - \boldsymbol{x}_2) \\ &= (\boldsymbol{x}_1 - \boldsymbol{x}_2)^\mathsf{T}\left(\boldsymbol{A}\boldsymbol{\Gamma} + p\boldsymbol{I}\right)(\boldsymbol{x}_1 - \boldsymbol{x}_2) \\ &= (\boldsymbol{x}_1 - \boldsymbol{x}_2)^\mathsf{T}\left(\boldsymbol{M} + p\boldsymbol{I}\right)(\boldsymbol{x}_1 - \boldsymbol{x}_2). \end{aligned} \tag{36}$$

If $p > \max\{0, -\lambda_{\min}\}$, the matrix $\boldsymbol{M} + p\boldsymbol{I}$ is positive definite, and specifically

$$\boldsymbol{M} + p\boldsymbol{I} - \eta\boldsymbol{I} \succcurlyeq 0$$

for η define above. Hence

$$(\boldsymbol{x}_1 - \boldsymbol{x}_2)^\mathsf{T}\left(\boldsymbol{M} + p\boldsymbol{I} - \eta\boldsymbol{I}\right)(\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq 0$$

implying that

$$(\boldsymbol{x}_1 - \boldsymbol{x}_2)^\mathsf{T}\left(\boldsymbol{M} + p\boldsymbol{I}\right)(\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq \eta\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2. \tag{37}$$

Plugging (37) in (36) completes the proof. □

B. Primal-Dual Method

Next we extend the results from the previous section to a primal-dual algorithm for solving (21). To this end, let

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^\mathsf{T}\boldsymbol{g}(\boldsymbol{x}) \tag{38}$$

denote the Lagrangian associated with (21) and assume that the problem is distributable (Assumption 5). Similarly to (32), we consider the regularized Lagrangian

$$\mathcal{L}_p(\boldsymbol{x}, \boldsymbol{\lambda}) := \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) + \frac{p}{2}\boldsymbol{x}^\mathsf{T}\boldsymbol{\Gamma}_x^{-1}\boldsymbol{x} - \frac{p}{2}\boldsymbol{\lambda}^\mathsf{T}\boldsymbol{\Gamma}_\lambda^{-1}\boldsymbol{\lambda}. \tag{39}$$

and the corresponding saddle-point problem. Note that, compared to (11), we use the same parameter p for both primal and dual part for simplicity. We next provide conditions under which the saddle point of $\mathcal{L}_p(\boldsymbol{x}, \boldsymbol{\lambda})$ coincides with the solution of the corresponding fixed-point equation as well as conditions for strong monotonicity of the primal-dual operator associated with $\mathcal{L}_p(\boldsymbol{x}, \boldsymbol{\lambda})$.

1) Optimality of Fixed Points: Consider the projected primal-dual algorithm

$$\boldsymbol{x}_i^{(k+1)} = \mathsf{Proj}_{\mathcal{X}_i}\Big\{\boldsymbol{x}_i^{(k)} - \alpha_i\boldsymbol{\Gamma}_{x,i}^{(k)}(\boldsymbol{x}_i^{(k)})\nabla_{\boldsymbol{x}_i}\mathcal{L}_{i,p}(\boldsymbol{x}_i^{(k)}, \boldsymbol{\lambda}^{(k)})\Big\}, \tag{40a}$$

$$\boldsymbol{\lambda}^{(k+1)} = \mathsf{Proj}_{\mathcal{D}}\Big\{\boldsymbol{\lambda}^{(k)} + \alpha\boldsymbol{\Gamma}_\lambda^{(k)}(\boldsymbol{\lambda}^{(k)})\nabla_\lambda\mathcal{L}_p(\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}^{(k)})\Big\}, \tag{40b}$$

where $\boldsymbol{\Gamma}_{x,i}^{(k)}$ and $\boldsymbol{\Gamma}_\lambda^{(k)}$ are defined similalry to (27).

Proposition 3. *The unique fixed point associated with* (40) *coincides with the unique saddle point of* $\mathcal{L}_p$*.*

The proof is similar to that of Proposition 1.

2) Strong Monotonicity: Let $\boldsymbol{z} := (\boldsymbol{x}^\intercal, \boldsymbol{\lambda}^\intercal)^\intercal$ and

$$\widetilde{\Phi}(\boldsymbol{z}) := \boldsymbol{\Gamma} \begin{bmatrix} \nabla_{\boldsymbol{x}} \mathcal{L}_p(\boldsymbol{z}) \\ -\nabla_{\boldsymbol{\lambda}} \mathcal{L}_p(\boldsymbol{z}) \end{bmatrix} \tag{41}$$

$$= \boldsymbol{\Gamma} \begin{bmatrix} \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} + \boldsymbol{D}^\intercal \boldsymbol{\lambda} + p\boldsymbol{\Gamma}_x^{-1}\boldsymbol{x} \\ -\boldsymbol{D}\boldsymbol{x} - \boldsymbol{d} + p\boldsymbol{\Gamma}_\lambda^{-1}\boldsymbol{\lambda} \end{bmatrix} \tag{42}$$

$$= \boldsymbol{\Gamma} \left(\begin{bmatrix} \boldsymbol{A} & \boldsymbol{D}^\intercal \\ -\boldsymbol{D} & \boldsymbol{0} \end{bmatrix} \boldsymbol{z} + p\boldsymbol{\Gamma}^{-1}\boldsymbol{z} + \begin{bmatrix} \boldsymbol{b} \\ -\boldsymbol{d} \end{bmatrix} \right) \tag{43}$$

$$= (\boldsymbol{\Gamma}\boldsymbol{W} + p\boldsymbol{I})\boldsymbol{z} + \boldsymbol{\Gamma}\boldsymbol{w}, \tag{44}$$

where we have defined

$$\boldsymbol{W} := \begin{bmatrix} \boldsymbol{A} & \boldsymbol{D}^\intercal \\ -\boldsymbol{D} & \boldsymbol{0} \end{bmatrix}, \quad \boldsymbol{w} := \begin{bmatrix} \boldsymbol{b} \\ -\boldsymbol{d} \end{bmatrix}. \tag{45}$$

Proposition 4. *Let*

$$\boldsymbol{V} := \frac{1}{2}\Big(\boldsymbol{\Gamma}\boldsymbol{W} + \boldsymbol{W}^\intercal\boldsymbol{\Gamma}\Big) \tag{46}$$

and let $\lambda_{\min}$ *be the smallest eigenvalue of* $\boldsymbol{V}$. *If* $p > \max\{0, -\lambda_{\min}\}$, *then* $\widetilde{\Phi}$ *is strongly monotone with parameter* $\eta := p + \min\{0, \lambda_{\min}\}$.

The proof of Proposition 4 follows that of Proposition 2.

C. Convergence Analysis of Time-Varying OFO Algorithms

Given the above results, a similar tracking result to that of Theorem 1 can be established. In particular, the tracking is now with respect to the modified reference trajectory $\{\boldsymbol{z}^{(*,k)}\}$ which are the saddle points of the regularized Lagrangian (39). The analysis is based on the standard time-varying fixed point convergence analysis [16] and is omitted here for brevity.

IV. Application to Power Systems

We next show two applications in the context of power systems, where heterogeneous step sizes naturally arise. The target optimization problem (8) is the linearized optimal power flow (OPF) problem as, e.g., in [15]. The OFO framework (13b) (16) is set up in a hierarchical structure (see Figure 1) where a coordinator collects the measurements of the network outputs $\widehat{\boldsymbol{y}}^{(k)}$, keeps track of the output constraints (8d), calculates their associated dual variables $\boldsymbol{\lambda}^{(k)}$, and broadcasts them to the N local controllers. In turn, each local controller i computes power injection set points $\boldsymbol{x}_i^{(k)}$ for distributed energy resource (DER) i based on its cost function $f_i^{(k)}$, its feasible set $\mathcal{X}_i^{(k)} \subset \mathbb{R}^2$, and the dual variables received from the coordinator.

There are two types of output constraints (8d) in the OPF problem associated with two different grid services:

- **Virtual power plant (VPP).** Vector $\boldsymbol{y}$ contains the aggregate power flow at the distribution feeder head (substation), and a constraint is added to enforce tracking of a given aggregate power setpoint.
- **Voltage regulation.** Vector $\boldsymbol{y}$ contains voltage magnitudes at different buses of the distribution network, and constraints are added to keep voltage magnitudes between prescribed bounds.

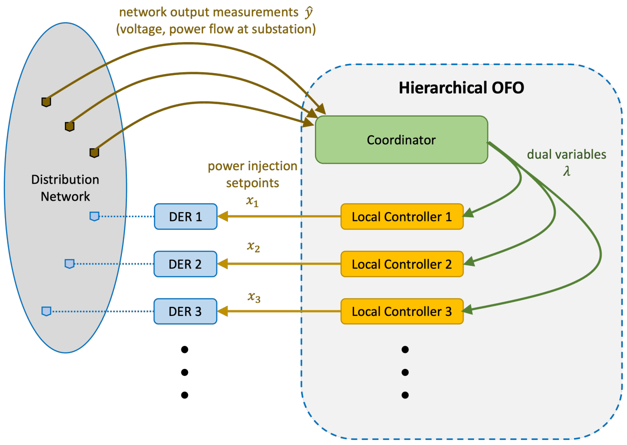


Fig. 1: Hierarchical control architecture of DERs in a power distribution network using OFO.

An estimate of the Jacobian matrix $\boldsymbol{C}$ in (9) is determined by linearizing the power flow equations as, e.g., in [17]. The cost functions $f_i^{(k)}$ of DERs are modeled with a quadratic function, quantifying the distance from a prescribed setpoint. For example, for a solar panel, the cost can be measured as the squared amount of active power curtailed from the maximum available power being generated from the panel. For an energy storage device or electric vehicle, it can be the squared difference between the injected active power and power to get to a desired state of charge. The feasible set $\mathcal{X}_i^{(k)}$ depends on the state of the DER. More details of DER models can be found in [15].

A. Adaptive Step Sizes

One of the challenges to implement OFO on a real distribution system is that the step sizes need to be decided which heavily determine its performance. If it were necessary to choose a common time-invariant step size, then a centralized operator would need to measure the performance of all the systems and constraints while finding a common step size that strikes a balance between them. Allowing each system and constraint to have its own time-varying step size opens up the possibility that each system and constraint can automatically find their own step sizes based on local conditions. Specifically, *adaptive* step sizes approach is proposed in [12], [18], as summarized next.

The adaptivity of the step sizes is based on two intuitive observations: (i) if the primal/dual variables are moving in the same direction for a period of time, then it likely indicates that they are heading towards a similar target point that is far away and speeding up the rate to get there would benefit the system; (ii) if the variables are oscillating, then it likely indicates that they are continually overshooting a target point and should dampen their action speeds. Additionally, each system of primal variables and grouping of dual variables, based on similarities between associated constraints, should independently adapt their step sizes.

Let Ω be the set of primal and dual variables grouped for adapting step sizes, i.e., $\Omega := \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{\lambda}_{\text{volt}}, \boldsymbol{\lambda}_{\text{VPP}}\}$

where $\boldsymbol{\lambda}_{\text{volt}}$ are the dual variables associated with voltage regulation and $\boldsymbol{\lambda}_{\text{VPP}}$ are the dual variables associated with VPP. The gradient of the Lagrangian with respect to a specific set of variables $\omega \in \Omega$ gives the direction that the algorithm is heading in, and thus comparing the gradients of the Lagrangian between consecutive iterations indicates a change or continuity in directions. We use the cosine similarity metric to measure how much the directions change for $\omega \in \Omega$ at iteration k:

$$s_\omega^{(k)} := \frac{\nabla_\omega \mathcal{L}_p(\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}^{(k)})^\intercal \nabla_\omega \mathcal{L}_p(\boldsymbol{x}^{(k-1)}, \boldsymbol{\lambda}^{(k-1)})}{\|\nabla_\omega \mathcal{L}_p(\boldsymbol{x}^{(k)}, \boldsymbol{\lambda}^{(k)})\| \|\nabla_\omega \mathcal{L}_p(\boldsymbol{x}^{(k-1)}, \boldsymbol{\lambda}^{(k-1)})\|} \tag{47}$$

where $s_\omega^{(k)}$ is between -1 and 1. A value closer to 1 indicates that the directions are similar; whereas, a value closer to -1 indicates that the directions are opposite.

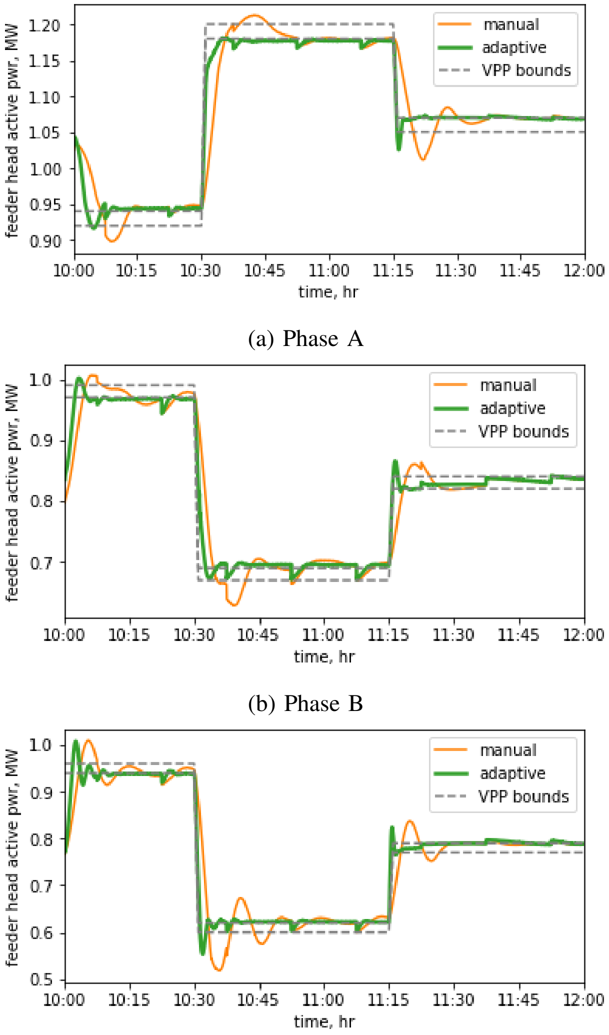
The cosine similarity at an iteration is used in a threshold rule to decide whether to proportionally increase, decrease or keep the step size the same:

$$\gamma_\omega^{(k)} := \begin{cases} \overline{\kappa}_\omega \gamma_\omega^{(k-1)} & \text{if } s_\omega^{(k)} > \overline{s}_\omega \\ \underline{\kappa}_\omega \gamma_\omega^{(k-1)} & \text{if } s_\omega^{(k)} < \underline{s}_\omega \\ \gamma_\omega^{(k-1)} & \text{otherwise} \end{cases} \tag{48}$$

where $\overline{\kappa}_\omega > 1$ is the increase factor, $\underline{\kappa}_\omega < 1$ is the decrease factor, and $(\overline{s}_\omega, \underline{s}_\omega)$ are the thresholds. The settings of the factors depend on the application, but typically it is good for $(\overline{\kappa}_\omega - 1) < (1 - \underline{\kappa}_\omega)$ so that the speed up is gradual and the slow down to avoid oscillations is quick. The stepsizes are then used in (40), with $\boldsymbol{\Gamma}_{x,i}^{(k)} := \mathsf{diag}(\gamma_{\boldsymbol{x}_i}^{(k)})$ and $\boldsymbol{\Gamma}_{\lambda}^{(k)} := \mathsf{diag}(\boldsymbol{\lambda}_{\text{volt}}^{(k)}, \boldsymbol{\lambda}_{\text{VPP}}^{(k)})$.

As a demonstration, we apply OFO with adaptive step sizes to a three-phase power distribution system (see [12] for more details and results). The following parameters for the adaptive step size rule are given the common settings for all sets of variables $\omega \in \Omega$: $\underline{s}_\omega := 0$, $\overline{s}_\omega := 0.9$, and $\overline{\kappa}_\omega := 1.005$. The parameters for $\underline{\kappa}_\omega$ are differentiated for the dual variables: $\underline{\kappa}_{\boldsymbol{\lambda}_{\text{volt}}} := 0.995$ and $\underline{\kappa}_{\boldsymbol{\lambda}_{\text{VPP}}} := 0.5$ so that step size for VPP decreases much faster than that for voltage regulation when oscillations are observed. This makes the voltage regulation constraints have a stronger adherence than the VPP constraints. Finally, the DER step size parameter $\underline{\kappa}_{\boldsymbol{x}_i} : \forall i \in \{1, \ldots, N\}$ is set to 0.95.

In the scenario we showcase here, the VPP setpoint makes two step changes and then the OFO controller needs to coordinate the DERs to change the feeder head power while keeping the voltage magnitudes within bounds. We compare using the best manually tuned common step size with step sizes from the rule in (47) and (48). Figure 2 shows how well the feeder head power follows the VPP bounds. The trajectories using the adaptive step sizes follows the bounds much faster and with less oscillations than those using the manually tuned step size. Figure 3 shows how well the voltage magnitudes are being kept within the bounds. The voltage magnitude trajectories using the adaptive step sizes has almost no violation and very little oscillations as compared those using the manually tuned step size.



(a) Phase A

(b) Phase B

(c) Phase C

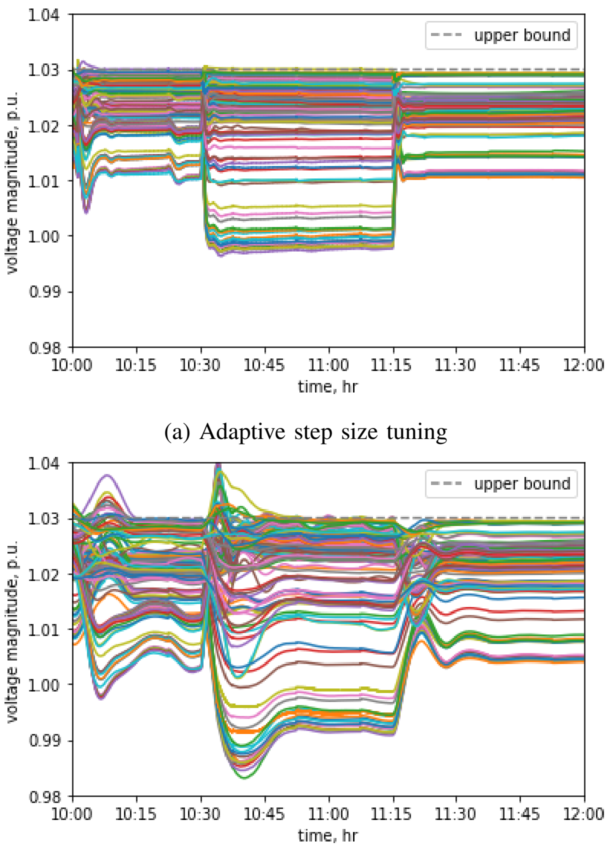
Fig. 2: VPP set point tracking: manually tuned versus adaptively tuned step sizes.

Finally, we show in Figure 4 how the step sizes change over time for the DERs and the grid services. It is easy to see that the heterogeneity of the step sizes and of the directions they go when the VPP bounds change are what enhances OFO to perform better than when using a manually tuned common time-invariant step size.

B. Zero-Order OFO

We next briefly mention the second application where heterogeneous step sizes naturally arise, which is the OFO with zero-order gradient estimation, as, e.g., in (6) (15). To illustrate the idea, we present the results from [6] regarding the approximate gradient (6). Specifically, under the assumptions in [6], the zero-order projected gradient descent is approximated via the following *averaged* iteration

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &= \mathsf{Proj}_{\mathcal{X}^{(k)}} \Big\{ \boldsymbol{x}^{(k)} \\ &\qquad - \alpha(\Gamma^{(k)} \nabla F^{(k)}(\boldsymbol{x}^{(k)}) + O(\alpha + \epsilon^2)) \Big\} \end{aligned} \tag{49}$$



(a) Adaptive step size tuning

(b) Manual step size tuning

Fig. 3: Voltages magnitudes when the VPP set points have step changes.

where

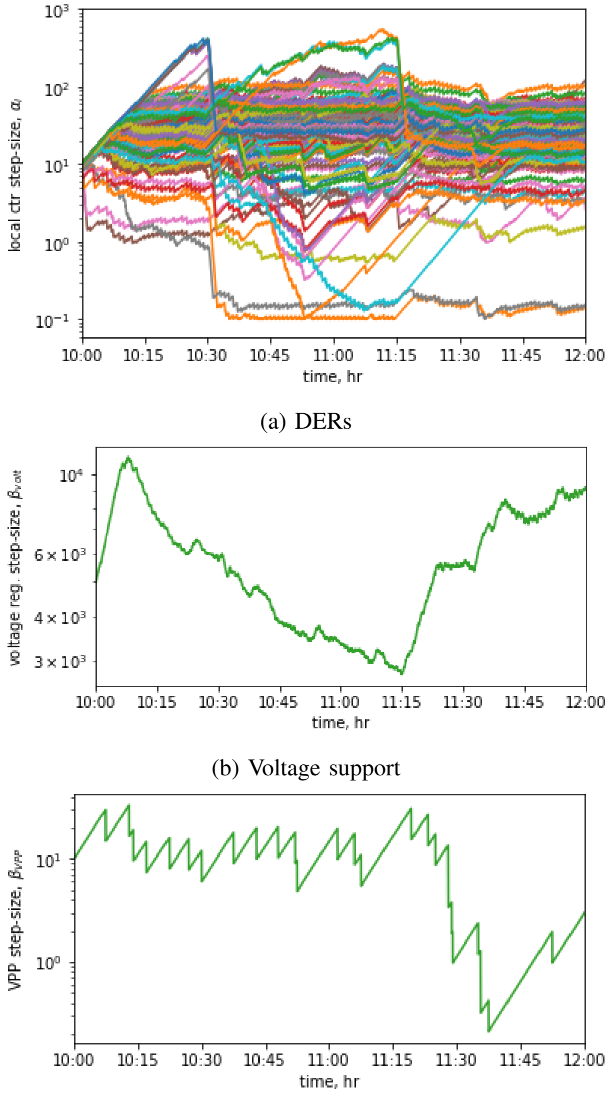
$$\Gamma^{(k)} := \int_{kT}^{k(T+1)} \boldsymbol{\xi}(\tau)\boldsymbol{\xi}(\tau)^{\mathsf{T}}\, d\tau. \tag{50}$$

In (50), T is the averaging period, and $\boldsymbol{\xi}(t)$ is a continuous-time exploration signal. For example, if $\boldsymbol{\xi}(t)$ is chosen as a set of sinusoidal signals with distinct periods and amplitudes, and T is chosen as a common integer multiplier of the periods, the matrix $\Gamma^{(k)}$ will be diagonal; see [6] for details. Therefore, iteration (49) becomes an approximate projected gradient algorithm with heterogeneous step sizes and the analysis in this paper applies.

V. Conclusion

In this paper, we presented an overview of the OFO framework and the recent advances on the topic. We also provided new theoretical insights on primal-dual OFO algorithms with heterogeneous step sizes for different elements of the gradient, and illustrated two applications in the power systems control context.

Most of the new analysis is restricted to quadratic programs, and the extension of the approach to general optimization problems remains an open research topic. The proposed regularization approach is effective in establishing strong monotonicity but leads to an analysis with respect to perturbed optimal saddle points; establishing tracking properties by bypassing the regularization (and perhaps strong monotonicity property altogether) is an open problem. Finally, an interesting direction is to consider the case where the matrix $\Gamma^{(k)}$ is a general positive definite, or even positive semi-definite matrix. The semi-definite case is of particular interest in the context of exact (rather than average) analysis of zero-order methods [6] and of asynchronous feedback gradient methods [19].



(a) DERs

(b) Voltage support

(c) VPP

Fig. 4: Step sizes of the DERs and the grid services during step changes in the VPP set points.

References

[1] E. Dall'Anese and A. Simonetto, "Optimal power flow pursuit," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 942–952, March 2018.

[2] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Projected gradient descent on riemannian manifolds with applications to online power system optimization," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2016, pp. 225–232.

[3] A. Hauswirth, I. Subotić, S. Bolognani, G. Hug, and F. Dörfler, "Time-varying projected dynamical systems with applications to feedback optimization of power systems," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 3258–3263.

[4] A. Bernstein, E. Dall'Anese, and A. Simonetto, "Online primal-dual methods with measurement feedback for time-varying convex optimization," *IEEE Transactions on Signal Processing*, vol. 67, no. 8, pp. 1978–1991, April 2019.

[5] M. Colombino, E. Dall'Anese, and A. Bernstein, "Online optimization as a feedback controller: Stability and tracking," *IEEE Transactions on Control of Network Systems*, pp. 1–1, 2019.

[6] Y. Chen, A. Bernstein, A. Devraj, and S. Meyn, "Model-free primal-dual methods for network optimization with application to real-time optimal power flow," in *2020 American Control Conference (ACC)*, 2020, pp. 3140–3147.

[7] V. Häberle, A. Hauswirth, L. Ortmann, S. Bolognani, and F. Dörfler, "Non-convex feedback optimization with input and output constraints," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 343–348, 2021.

[8] Z. He, S. Bolognani, J. He, F. Dörfler, and X. Guan, "Model-free nonlinear feedback optimization," 2022.

[9] S. Zhan, I. Dukovska, W. van den Akker, A. van der Molen, J. Morren, N. Paterakis, and J. Slootweg, "Review of recent developments in technical control approaches for voltage and congestion management in distribution networks," in *IEEE PowerTech Belgrade 2023*. Institute of Electrical and Electronics Engineers, 2023.

[10] M. Nonhoff and M. A. Müller, "Online convex optimization for data-driven control of dynamical systems," *IEEE Open Journal of Control Systems*, vol. 1, pp. 180–193, 2022.

[11] A. Bernstein, Y. Chen, M. Colombino, E. Dall'Anese, P. Mehta, and S. Meyn, "Quasi-stochastic approximation and off-policy reinforcement learning," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 5244–5251.

[12] J. Comden, J. Wang, and A. Bernstein, "Adaptive primal-dual control for distributed energy resource management," *arXiv preprint arXiv:2304.12478*, 2023.

[13] J. Koshal, A. Nedić, and U. Y. Shanbhag, "Multiuser optimization: Distributed algorithms and error analysis," *SIAM J. on Optimization*, vol. 21, no. 3, pp. 1046–1081, 2011.

[14] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, March 1992.

[15] A. Bernstein and E. Dall'Anese, "Real-time feedback-based optimization of distribution grids: A unified approach," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 1197–1209, Sep. 2019.

[16] A. Bernstein and E. Dall'Anese, "Asynchronous and distributed tracking of time-varying fixed points," in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 3236–3243.

[17] A. Bernstein and E. Dall'Anese, "Linear power-flow models in multiphase distribution networks," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE, 2017, pp. 1–6.

[18] T. Yokota and H. Hontani, "An efficient method for adapting step-size parameters of primal-dual hybrid gradient method in application to total variation regularization," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 973–979.

[19] B. K. Poolla, G. Cavraro, and A. Bernstein, "State estimation for distribution networks with asynchronous sensors using stochastic descent," in *2022 IEEE Power Energy Society General Meeting (PESGM)*, 2022, pp. 1–5.