

# Time-Varying Optimization of Networked Systems With Human Preferences

Ana M. Ospina , Graduate Student Member, IEEE, Andrea Simonetto , and Emiliano Dall'Anese 

**Abstract**—This article considers a time-varying optimization problem associated with a network of systems, with each of the systems shared by (and affecting) a number of individuals. The objective is to minimize cost functions associated with the individuals' preferences, which are unknown, subject to time-varying constraints that capture physical or operational limits of the network. To this end, this article develops a distributed online optimization algorithm with concurrent learning of the cost functions. The cost functions are learned on-the-fly based on the users' feedback (provided at irregular intervals) by leveraging tools from the shape-constrained Gaussian process. The online algorithm is based on a primal–dual method and acts effectively in a closed-loop fashion where: First, users' feedback is utilized to estimate the cost, and second, measurements from the network are utilized in the algorithmic steps to bypass the need for sensing of (unknown) exogenous inputs of the network. The performance of the algorithm is analyzed in terms of dynamic network regret and constraint violation. Numerical examples are presented in the context of real-time optimization of distributed energy resources.

**Index Terms**—Feedback, Gaussian process (GP), networked systems, regret analysis, time-varying optimization.

## I. INTRODUCTION

WE CONSIDER optimization problems associated with a network of systems, where each of the systems is shared by a number of individuals. Typically, such a multiuser problem includes a cost given by a sum of user-specific functions, and a set of constraints that capture physical or operational limits of the network (and, thus, that couple the users' decisions). For example, multiuser problems were considered in [1] and solved via a primal–dual method based on a regularization of the Lagrangian function, and a distributed

Manuscript received 8 March 2022; revised 11 March 2022 and 5 July 2022; accepted 14 August 2022. Date of publication 1 September 2022; date of current version 13 March 2023. The work of A. Ospina and E. Dall'Anese was supported in part by the National Science Foundation CAREER under Grant 1941896, in part by the NSF ERC ASPIRE and in part by the U.S Department of Energy Project 3.2.6.80. Recommended by Associate Editor A. D. Dominguez-Garcia. (Corresponding author: Ana M. Ospina.)

Ana M. Ospina and Emiliano Dall'Anese are with the Department of Electrical, Computer and Energy Engineering, University of Colorado Boulder, Boulder, CO 80309 USA (e-mail: ana.ospina@colorado.edu; emiliano.dallanese@colorado.edu).

Andrea Simonetto is with the UMA, ENSTA Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France (e-mail: andrea.simonetto@ensta-paris.fr).

Digital Object Identifier 10.1109/TCNS.2022.3203467

resource allocation problem was investigated in [2] (where the communications between users' are susceptible to adversarial attacks). Similar formulations arise in a time-varying setting, where the cost and/or constraints evolve over time to reflect changes in the constraints or problem inputs [3], [4], [5], [6]. Specifically, unconstrained problems with a known time-varying cost are analyzed in [3] via a gradient descent method. A time-varying multiuser problem is presented in [4], where a double regularization both in the primal and in the dual space is employed to increase the convergence rate of the algorithm. In [5], convex optimization problems with known time-varying objective functions are solved using a real-time self-triggered control method. We refer the reader to the survey in [6] for a complete list of references on time-varying convex optimization.

Generally, one prerequisite for solving these problems is that the user-specific functions are known or properly crafted based on synthetic or average models. Instead, we consider the case where synthetic costs are not representative of the preferences of individual users or fail to capture the diversity in their perception of comfort, safety, or dissatisfaction [7], [8], [9]. In this context, a learning method that leverages human feedback is explored in [7], where a robot observes data in the form of state-action and the discrete dynamics are learned during the execution of the algorithm. An example of field research on occupant satisfaction to the indoor temperature via recording of smiley-face polling station is presented in [8]. Moreover, [9] presents an overview of opportunities and risks of data-driven decision-making methods.

We tackle the problem of solving a network optimization problem when the users' cost functions are *unknown* and develop an online algorithm where the cost functions are learned concurrently with the execution of the algorithmic steps. In particular, the learning procedure leverages “users' feedback” and utilizes learning tools from shape-constrained Gaussian processes (GPs) [10]. The algorithm is implemented in a *distributed* fashion; this allows users sharing a system to agree on a solution that minimizes the sum of their (learned) functions, without revealing their preferences or their feedback. In this work, we consider the case where the users report their preferences trustworthy (and we do not consider adversarial behaviors). In addition to the feedback from the users, the algorithm leverages the ideas from, e.g., [11] and [12] and utilizes measurements of network outputs instead of the network model.

The endeavors are motivated by a number of problems arising in power systems [11], [13], [14]; charging of electric vehicles (EV) [15]; and human-aware robot systems [16], just to name a few. While Section II will explain some examples, we stress

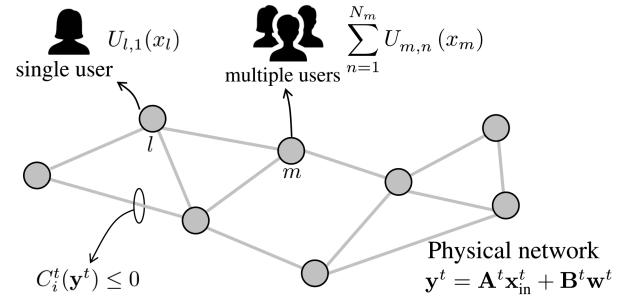
here that an accurate model of the comfort/satisfaction level of users is typically unknown, and it may vary not only across individuals, but also for the same individual [7], [8]. In this context, shape-constrained GP is a powerful tool for nonparametric function estimation when the underlying function is presumed to be convex or monotonic [10]; this implies that each user has a preference for a set of solutions.

*Contributions.* Overall, the main contributions of this article are as follows.

- C1) We show how to estimate the users' functions via GPs and, in particular, how to handle asynchronous and noisy data easily.
- C2) We develop an online algorithm based on a primal–dual method to solve the formulated time-varying network optimization problem; the algorithm is implemented online, and it is modified to accommodate measurements and learned functions.
- C3) We propose a consensus-based formulation that leads to a distributed online algorithm where users minimize the sum of their (learned) functions, without revealing their preferences or their feedback.
- C4) For the performance analysis, we view the algorithm as an inexact online primal–dual method where errors are due to the shape-constrained GP approximation and errors in the computation of the gradient of the GP. We derive bounds for a dynamic extension of the network regret [17], [18], and for the average constraint violation (ACV) (see also, e.g., [19], [20], [21], [22] and references therein for other notions of dynamic regret).
- C5) We showcase the proposed methodology in an example related to control of distributed energy resources (DER) in power grids [14].

*Prior works:* In the context of bandit optimization, a static regret analysis is performed in [19], where the objective is to minimize a sequence of unknown convex cost functions and one has access to functional evaluations. Time-varying cost and constraints are considered in the context of a bandit setting in [23] and online saddle-point algorithms are utilized, where the gradient information is acquired via one-point or multipoint estimates. In [24], a gradient-free approach is presented to solve a multiagent distributed constrained optimization problem where the goal is to cooperatively minimize the sum of time-changing local cost functions subject to time-varying coupled constraints. As an example of works in the context of zeroth-order methods, a multiagent optimization problem is analyzed in [25] where the objective is to minimize the average of the nonconvex local costs that depends on the joint actions of the agents. See also [26] for an overview of zeroth-order algorithms. Our approach is different from zeroth-order or bandit methods [19], [23], [24], [25], [26]; these would require multiple functional evaluations at each step to estimate the gradient of the users' costs—something not feasible for our problem. Our contribution is to use shape-constrained GPs to process feedback that is provided parsimoniously and at irregular intervals. The considered setting is different from that in [27], where GPs are utilized to maximize an unknown function.

A similar problem setup was studied in [28]; however, no constraints are considered (in fact, a gradient tracking scheme



**Fig. 1.** We consider a network of systems, coupled through physical or operational dependencies, where each system is shared by (and affects) a number of users.

is utilized) and the user's cost is assumed to have a known quadratic structure *a priori*. We significantly extend our previous works [29], [30] by considering a distributed setting and constrained problems. Relative to the works [17], [18] on distributed online primal–dual algorithms, we consider problems where the optimal solution set changes in time (and it is not fixed in time) and with both consensus constraints and time-varying constraints that capture operational limits of the network. We also analyze the network regret when the online algorithm is *inexact*. Indeed, we can recover an asymptotic bound of  $\mathcal{O}(T^{\frac{1}{2}})$  as in [17] and [18] from our regret bound when the cost functions are known, the gradients are computed exactly, and the optimal solutions are not time varying (with  $T$  as the number of time steps). Our analysis can also be utilized to extend [25] to time-varying settings. Finally, we extend the findings of the work in [11] and [12] by considering consensus constraints (and accompanying distributed implementations) and by including shape-constrained GPs in the overall algorithmic framework.

The rest of this article is organized as follows. Section II presents the time-varying problem, and Section III outlines the GP-based learning approach and the proposed online algorithm. Section IV presents the performance analysis. The numerical results for a demand response problem are in Section V. Finally, Section VI concludes this article<sup>1</sup>.

## II. PROBLEM STATEMENT

We consider a network of  $M$  systems or devices, with each system interacting with (or shared by) a number of users, as illustrated in Fig. 1. In particular, we denote by  $\mathcal{N}_m = \{1, \dots, N_m\}$  the set of  $N_m \geq 1$  users interacting with the  $m$ th system. The input–output relationship of the network is modeled through the time-varying map

$$\mathbf{y}^t = \mathbf{A}^t \mathbf{x}_{in}^t + \mathbf{B}^t \mathbf{w}^t \quad (1)$$

where  $t$  is the time index;  $\mathbf{x}_{in}^t := [x_1^t, \dots, x_M^t]^\top$ , with  $x_m^t \in \mathbb{R}$  the controllable input of the  $m$ th system;  $\mathbf{y}^t \in \mathbb{R}^Y$  is a vector of

<sup>1</sup>Notation: Upper-case (lower-case) boldface letters will be used for matrices (column vectors);  $(\cdot)^\top$  denotes transposition. For a given column vector  $\mathbf{x} \in \mathbb{R}^n$ ,  $\|\mathbf{x}\| := \sqrt{\mathbf{x}^\top \mathbf{x}}$ ; we denote with  $\mathbf{1}$  and  $\mathbf{0}$  the vectors of all ones and zeros, respectively, where the dimension will be clear from the context.  $\mathcal{O}(\cdot)$  refers to the big-O notation, whereas  $o(\cdot)$  refers to the little-o notation. The max operator is defined as  $[x]^+ := \max\{0, x\}$ . The diameter of  $\mathcal{X}^t$  is defined as  $\text{diam}(\mathcal{X}^t) = \max\{\|\mathbf{x} - \mathbf{x}'\| : \mathbf{x}, \mathbf{x}' \in \mathcal{X}^t\}$ . For a given random variable  $\xi \in \mathbb{R}$ ,  $\mathbb{E}[\xi]$  denotes the expected value of  $\xi$ .

observables or outputs of the network;  $\mathbf{w}^t \in \mathbb{R}^W$  is a vector of *unknown* exogenous inputs; and  $\mathbf{A}^t$  and  $\mathbf{B}^t$  are possibly time-varying matrices (with  $\mathbf{A}^t$  known and  $\mathbf{A}^t \neq \mathbf{0}$ ) of appropriate dimensions. We assume that  $t \in \mathcal{T} := \{k\Delta, k \in \mathbb{N}\}$ , where  $\Delta > 0$  is a given time interval.

Let  $U_{m,n} : \mathbb{R} \rightarrow \mathbb{R}$  be a convex function representing the cost of a user  $n \in \mathcal{N}_m$  associated with the  $m$ th device;<sup>2</sup> this function can capture a sense of dissatisfaction, discomfort, or simply preferences depending on the specific application (a few examples will be provided shortly). The goal is then to compute the sequence of inputs that solves the following time-varying optimization problem [3], [4], [5] associated with the network:

$$\begin{aligned} \min_{\{x_m \in \mathcal{X}_m^t\}_{m=1}^M} \quad & \sum_{m=1}^M \sum_{n=1}^{N_m} U_{m,n}(x_m) \quad (2a) \\ \text{s. to: } & C_i^t(\mathbf{A}^t \mathbf{x}_{\text{in}} + \mathbf{B}^t \mathbf{w}^t) \leq 0, i = 1, \dots, N_C \quad (2b) \end{aligned}$$

where  $\{\mathcal{X}_m^t\}_{m=1}^M$  are time-varying convex and compact sets for the inputs, and  $C_i^t : \mathbb{R}^M \rightarrow \mathbb{R}$  is a time-varying function parameterized by  $\mathbf{w}^t$ . In particular, (2b) models the operational constraints associated with the network output  $\mathbf{y}^t$ . The main goal of (2) is to generate a sequence of inputs that minimizes the cost of the users while respecting pertinent network constraints. To this end, this article focuses on addressing the following four main challenges related to (2).

- 1) *Challenge 1:* The variable  $x_m$  is *coupled* across the users  $\mathcal{N}_m$ . Our goal is to solve (2) in a distributed fashion, where users are not required to share information about their costs  $\{U_{m,n}\}$  (and their functional evaluations) with the device or the network operator.
- 2) *Challenge 2:* The costs  $\{U_{m,n}\}$  are *unknown*. One may utilize synthetic cost functions for users' preferences, comfort, or satisfaction based on statistical models; however, these synthetic costs may fail to capture the diversity in the users' perception, preferences, and goals.
- 3) *Challenge 3:* The exogenous inputs  $\mathbf{w}^t$  may be *unknown* or partially known.
- 4) *Challenge 4:* The constraints  $C_i^t$  and the sets  $\{\mathcal{X}_m^t\}$  may change at each time  $t$ ; the interval  $\Delta$  may not be sufficient to solve each instance of the time-varying problem (2) to convergence.

To address the abovementioned challenges 1)–4), this article will develop a *consensus-based online* algorithm where the cost functions of the users are *learned* concurrently with the execution of the algorithm, and *measurements* of  $\mathbf{y}^t$  are utilized in the algorithmic updates.

Before proceeding, we list representative examples of applications in networked systems in which the problem (2) and the accompanying online optimization framework considered in this article are particularly well suited.

*Example 1 (Power grids):* In the context of power grids, (2) may capture a demand response task [13], [14] or a real-time optimal power flow problem [11]. In this case,  $\mathbf{x}_{\text{in}}$  are the

<sup>2</sup>We note that the problem formulation and solution approach proposed in this article can be naturally extended to the case where the number of users interacting with a device changes over time.

power setpoints of DER,  $\mathbf{w}^t$  is a vector of powers consumed by uncontrollable loads, and  $y^t$  may represent the total power as  $y^t = \mathbf{1}^\top \mathbf{x}_{\text{in}} + \mathbf{1}^\top \mathbf{w}^t$  and/or voltage magnitudes (one can also compute the matrix  $\mathbf{A}^t$  based on a linearized ac model). Constraints (2b) may ensure that the net power follows a given automatic gain control or demand response signal; e.g.,  $C_i^t(y) = (y - y_{\text{ref}}^t)^2 - \zeta$ , where  $y_{\text{ref}}^t$  is a given reference signal and  $\zeta > 0$  is an error tolerance. One may also have voltage constraints. The costs  $\{U_{m,n}\}$  model discomfort, e.g., the indoor temperature. The number of users depends on the particular setting, for example, one may expect a large number of users sharing a conference room in an office building. The users that share a device  $m$  agree on the input  $x_m$ ; for example, people sharing a room agree on a temperature and, thus, on the use of an HVAC system.  $\square$

*Example 2 (EV charging):* The formulation (2) may represent the problem of charging a fleet of vehicles at a charging station, the variable  $x_m$  represents the charging rate (or the expected time for full charging) of a vehicle, constraint (2b) captures limits on the total power consumed by the charging station while  $\{U_{m,n}\}$  captures the dissatisfaction for a given expected time of charging completion [15]. If an EV is shared by multiple users,  $x_m$  represents the agreed expected charging time  $x_m$ .  $\square$

Additional examples in transportation systems, communication systems, and human-aware robot systems are not included for space limitations.

We now proceed with the reformulation of (2) into a consensus-based problem. To this end, let  $x_{m,n}$  be an auxiliary variable representing the input for the  $m$ th system preferred by user  $n \in \mathcal{N}_m$ ; with the auxiliary variables in place, stack the optimization variables in the column vector

$$\mathbf{x}^t = [x_1^t, x_{1,1}^t, \dots, x_{1,N_1}^t, \dots, x_M^t, x_{M,1}^t, \dots, x_{M,N_M}^t]^\top.$$

Accordingly, (1) can be rewritten as  $\mathbf{y}^t = \bar{\mathbf{A}}^t \mathbf{x}^t + \mathbf{B}^t \mathbf{w}^t$ , where  $\bar{\mathbf{A}}^t$  is an augmented version of  $\mathbf{A}^t$  with appropriate zero entries.

Then, (2) can be equivalently reformulated as

$$\min_{\{x_m, x_{m,n} \in \mathcal{X}_m^t\}} \sum_{m=1}^M \sum_{n=1}^{N_m} U_{m,n}(x_{m,n}) \quad (3a)$$

$$\text{s.t. : } C_i^t(\bar{\mathbf{A}}^t \mathbf{x}^t + \mathbf{B}^t \mathbf{w}^t) \leq 0, i = 1, \dots, N_C \quad (3b)$$

$$x_m = x_{m,n} \quad \forall n \in \mathcal{N}_m \quad \forall m = 1, \dots, M \quad (3c)$$

where (3c) defines consensus constraints for the  $m$ th system and its users. We note that the choice of posing the consensus constraints as in (3c) is not generic; the particular structure of (3c), which can be modeled as a star graph (with the system as the central node), will facilitate the development of a closed-loop algorithm where  $\{x_m\}$  will be physical inputs [as in (1)] and  $\{x_{m,n}\}$  will be auxiliary control variables.

For future developments, define  $f(\mathbf{x}) := \sum_{m=1}^M \sum_{n=1}^{N_m} U_{m,n}(x_{m,n})$  and  $\mathcal{X}^t := \mathcal{X}_1^t \times \dots \times \mathcal{X}_M^t \subseteq \mathbb{R}^M$ . For each device  $m$ , we define the incidence matrix  $\mathbf{D}_m \in \mathbb{R}^{N_m \times (1+N_m)}$ , which represents the relation between the device and the users sharing that device. For example, if the device 1 is shared by

two users we have that

$$\mathbf{D}_1 = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}.$$

By organizing these incidence matrices in a block diagonal matrix, we construct the augmented incidence matrix  $\mathbf{D} = \text{diag}(\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_M)$ . Then, (3c) can be compactly written as  $\mathbf{D}\mathbf{x} = \mathbf{0}$ , capturing consensus constraints over  $M$  systems.

The next section outlines the proposed algorithm to solve the time-varying network optimization problem (2). We note that in what follows we consider the case where  $N_C = 1$  to simplify exposition and notation; however, the technical arguments of this article straightforwardly extend to problems with multiple constraints on  $\mathbf{y}^t$ .

### III. DISTRIBUTED ONLINE ALGORITHM

In this section, we present our *consensus-based online* algorithm for solving (3). We first explain how to derive a distributed online algorithm based on a primal–dual method, then, we elaborate on how to estimate the cost functions based on feedback received from the users at infrequent times.

#### A. Online Primal–Dual Algorithm

We start by defining the time-varying Lagrangian function associated with (3) as follows:

$$\mathcal{L}^t(\mathbf{x}^t, \nu^t, \lambda^t) := f(\mathbf{x}^t) + \nu^t C^t (\bar{\mathbf{A}}^t \mathbf{x}^t + \mathbf{B}^t \mathbf{w}^t) + (\mathbf{D}\mathbf{x}^t)^\top \lambda^t$$

where  $\nu^t \in \mathbb{R}_+$ ,  $\lambda^t \in \mathbb{R}^N$  are the dual variables associated with the constraint (3b) and the  $M$  consensus constraints (3c), respectively.

Recall that  $t \in \mathcal{T} := \{k\Delta, k \in \mathbb{N}\}$  (and take  $\Delta = 1$  for simplicity of exposition). For a given step size  $\alpha > 0$ , a *model-based* online projected primal–dual algorithm involves the sequential execution of the following steps:

$$\begin{aligned} \mathbf{x}^{t+1} = & \text{proj}_{\mathcal{X}^t} \left\{ \mathbf{x}^t - \alpha (\nabla_{\mathbf{x}} f(\mathbf{x}^t) \right. \\ & \left. + \nu^t (\bar{\mathbf{A}}^t)^\top \nabla_{\mathbf{x}} C^t (\bar{\mathbf{A}}^t \mathbf{x}^t + \mathbf{B}^t \mathbf{w}^t) + \mathbf{D}^\top \lambda^t) \right\} \end{aligned} \quad (4a)$$

$$\nu^{t+1} = \text{proj}_{\Psi^t} \left\{ \nu^t + \alpha C^t (\bar{\mathbf{A}}^t \mathbf{x}^t + \mathbf{B}^t \mathbf{w}^t) \right\} \quad (4b)$$

$$\lambda^{t+1} = \text{proj}_{\Lambda^t} \left\{ \lambda^t + \alpha \mathbf{D}\mathbf{x}^t \right\} \quad (4c)$$

where  $\Psi^t$  is a convex and compact set, and  $\Lambda^t = \Lambda_{1,1}^t \times \dots \times \Lambda_{1,N_1}^t \times \dots \times \Lambda_{M,N_M}^t$ , with  $\Lambda_{m,n}^t$  also a convex and compact set constructed as explained shortly in the Section IV (and in, e.g., [1] and [17]).

We note that the steps (4a) and (4b) require one to know  $\mathbf{w}^t$  and  $\mathbf{B}^t$  (as explained in Challenges 3 and 4). To bypass this hurdle, we adopt the strategy proposed in, e.g., [11] and [12]. To this end, assume that  $\mathbf{x}_{\text{in}}^t$  are implemented as inputs to the systems at time  $t$ , and let  $\hat{\mathbf{y}}^t$  represent a *measurement* of  $\mathbf{y}^t$  collected at time  $t$ . Then,  $\nu^{t+1}$  can be computed as  $\nu^{t+1} = \text{proj}_{\Psi} \{ \nu^t + \alpha C^t(\hat{\mathbf{y}}^t) \}$ ; it is clear that information about  $\mathbf{w}^t$  and  $\mathbf{B}^t$  is not required in this case. We also note that the step (4a) would require the gradient of the functions  $\{U_{m,n}\}$ , which are not known

(see Challenge 2). Let  $\hat{U}_{m,n}^t(x_m^t)$  represent an estimate of the function  $U_{m,n}(x_m^t)$  available at  $x_m^t$ , and let  $g_{m,n}^t$  denote an *estimate* of the gradient (or derivative, in this case) of  $U_{m,n}(x_m^t)$  at time  $t$ . With these definitions, the steps in (4) can be suitably modified to accommodate measurements of  $\mathbf{y}^t$  and estimates of the gradient of the cost function

$$x_m^{t+1} = \text{proj}_{\mathcal{X}_m^t} \left\{ x_m^t - \alpha \left( \nu^t (\mathbf{a}_m^t)^\top \nabla C^t (\hat{\mathbf{y}}^t) + \sum_{n=1}^{N_m} \lambda_{m,n}^t \right) \right\} \quad (5a)$$

$$x_{m,n}^{t+1} = \text{proj}_{\mathcal{X}_m^t} \left\{ x_{m,n}^t - \alpha (g_{m,n}^t - \lambda_{m,n}^t) \right\} \quad \forall n \in \mathcal{N}_m \quad \forall m \quad (5b)$$

$$\nu^{t+1} = \text{proj}_{\Psi^t} \{ \nu^t + \alpha C^t (\hat{\mathbf{y}}^t) \} \quad (5c)$$

$$\lambda_{m,n}^{t+1} = \text{proj}_{\Lambda_{m,n}^t} \{ \lambda_{m,n}^t + \alpha (x_m^t - x_{m,n}^t) \} \quad \forall n \in \mathcal{N}_m \quad \forall m \quad (5d)$$

where  $\mathbf{a}_m^t$  is the  $m$ th column of the matrix  $\mathbf{A}^t$ , and we note that step (5a) is performed in parallel at each system. While measurements of  $\mathbf{y}^t$  are utilized in the primal step (5a) and the dual step (5c), an estimate of the gradient  $g_{m,n}^t$  will be obtained by leveraging feedback from the users as explained in the following section. Subsequently, Section III-C will overview the steps of the proposed algorithm.

#### B. Online Learning via Shape-Constrained GPs

We assume that the feedback of the user comes in the form of a (possibly noisy) functional evaluation, that is, for a given point  $x$ , one may receive from a user  $n$  interacting with the device  $m$  a rating  $z$  given by  $z = U_{m,n}(x) + \epsilon$ . With this model, a popular way to obtain the gradient of  $U_{m,n}$  at a point  $x$  is via zeroth-order methods (see, e.g., [23], [26], and references therein). However, zeroth-order methods are not well suited for our algorithm, because they would require feedback from each user at *each iteration* of the algorithm (either a single functional evaluation or more, depending on the particular method utilized). Instead, we consider a more realistic case where users may provide feedback more parsimoniously and at irregular intervals.

With this setting in mind, we consider utilizing the noisy functional evaluations provided by a user to estimate the function  $U_{m,n}$  via GPs [31]. In particular, we apply a GP with specific constraints on the shape of the function. As explained in [10], shape-constrained GP offers a powerful tool for nonparametric function estimation when the underlying function is presumed to be convex or monotonic. In our case, convexity or monotonicity implies that each user has a preference for a particular solution (or a convex set of solutions), and the dissatisfaction or discomfort increases as the system deviates from such a preferred point. In the following, we first briefly introduce GPs, and then, we provide the main equations to estimate  $U_{m,n}$  via shape-constrained GPs. Since we utilize a GP per function  $U_{m,n}$ , we drop the subscripts  $m$  and  $n$  for simplicity of exposition. We also note that the extension to the multidimensional shape-constrained GP is possible (see, for example, [10]); however, it is left as future work.

A GP is a stochastic process  $U(x)$  and it is specified by its mean function  $\mu(x)$  and its covariance function  $k(x, x')$ ; where  $k(\cdot, \cdot)$  is a given kernel, i.e., for any  $x, x' \in \mathcal{X} \subseteq \mathbb{R}$ ,  $\mu(x) = \mathbb{E}[U(x)]$  and  $k(x, x') = \mathbb{E}[(U(x) - \mu(x))(U(x') - \mu(x'))]$  [31]. Hereafter, we use the short-hand notation  $U(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$  for  $U(x)$  defined as a GP with mean  $\mu(x)$  and covariance  $k(x, x')$ . Let  $\mathbf{x}_p = [x_1 \in \mathcal{X}, \dots, x_p \in \mathcal{X}]^\top$  be the set of  $p$  sample points collected over a given period<sup>3</sup>; let  $z_i = U(x_i) + \epsilon_i$ , with  $\epsilon_i \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$  Gaussian noise, be the noisy measurement at the sample point  $x_i$ ; define further  $\mathbf{z} = [z_1, \dots, z_p]^\top$ . Then, the posterior distribution of  $(U(x)|\mathbf{x}_p, \mathbf{z})$  is a GP with mean  $\mu(x)$ , covariance  $k(x, x')$  and variance  $\sigma^2(x)$  given by [31]

$$\mu(x) = \mathbf{k}(x)^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{z} \quad (6a)$$

$$k(x, x') = k(x, x') - \mathbf{k}(x)^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}(x') \quad (6b)$$

$$\sigma^2(x) = k(x, x) \quad (6c)$$

where  $\mathbf{k}(x) = [k(x_1, x), \dots, k(x_p, x)]^\top$  and  $\mathbf{K}$  is the positive definite kernel matrix  $[k(x, x')]$ . Note that (6) depends on the  $p$  sample points for a specific period of time. Thus, an estimate of the unknown function  $U(x)$  can be written as  $U(x) \sim \mathcal{GP}(\mu(x), k(x, x'))$ . Using a squared exponential (SE) kernel as an example,  $k(x, x')$  is given by  $k(x, x') = \sigma_f^2 e^{-\frac{1}{2\ell^2}(x-x')^2}$  for the univariate input case, where the hyperparameters are the signal variance  $\sigma_f^2$  and the characteristic length scale  $\ell$ .

We now explain how to enforce shape constraints through the use of derivative processes (noting that since differentiation is a linear operator, derivatives of a GP remain a GP [31]). Because the GP obtained from the SE kernel function has derivatives of all orders, the mean and covariance function (jointly with the original process and the second-order derivative process) are [10]

$$\mathbb{E} \left[ \frac{\partial^2 U(x)}{\partial x^2} \right] = \frac{\partial^2 \mu(x)}{\partial x^2} = 0 \quad (7a)$$

$$\begin{aligned} k^{22}(x, x') &:= \text{cov} \left[ \frac{\partial^2 U(x)}{\partial x^2}, \frac{\partial^2 U(x')}{\partial x'^2} \right] = \sigma_f^2 e^{-\frac{1}{2\ell^2}(x-x')^2} \\ &\quad \times \frac{1}{\ell^4} \left( \frac{1}{\ell^4} (x-x')^4 - \frac{1}{\ell^2} 6(x-x')^2 + 3 \right) \end{aligned} \quad (7b)$$

$$\begin{aligned} k^{02}(x, x') &:= \text{cov} \left[ \frac{\partial^2 U(x)}{\partial x^2}, U(x') \right] \\ &= \sigma_f^2 e^{-\frac{1}{2\ell^2}(x-x')^2} \left( \frac{1}{\ell^4} (x-x')^2 - \frac{1}{\ell^2} \right). \end{aligned} \quad (7c)$$

We consider enforcing constraints on the second-order derivative at  $q$  points  $\mathbf{s} := [s_1, \dots, s_q]^\top$  (and these points may be in general different from  $\mathbf{x}_p$ ). Let  $\mathbf{U}(\mathbf{x}_p) = [U(x_1), \dots, U(x_p)]^\top$  and  $\mathbf{U}''(\mathbf{s}) = [U''(s_1), \dots, U''(s_q)]^\top$ , then, the joint distribution

<sup>3</sup>In the context of our work, the vector  $\mathbf{x}_p$  associated with a system  $m$  and a user  $n$  would contain the points  $x_{m,n}^{t_1}, \dots, x_{m,n}^{t_p}$ , where  $\{t_i\}_{i=1}^p \subset \mathcal{T}$  are the times instants where the user provides feedback.

of the GP and its second-order derivative is

$$\begin{bmatrix} \mathbf{U}(\mathbf{x}_p) \\ \mathbf{U}''(\mathbf{s}) \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \mu \mathbf{1}_p \\ \mathbf{0}_q \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{x}_p, \mathbf{x}_p) & \mathbf{K}^{02}(\mathbf{x}_p, \mathbf{s}) \\ \mathbf{K}^{20}(\mathbf{s}, \mathbf{x}_p) & \mathbf{K}^{22}(\mathbf{s}, \mathbf{s}) \end{bmatrix} \right)$$

where

$$\begin{aligned} \mathbf{K}(\mathbf{x}_p, \mathbf{x}_p) &= \mathbf{K}, & \mathbf{K}^{02}(\mathbf{x}_p, \mathbf{s}) &= [k^{02}(x, s)] \\ \mathbf{K}^{20}(\mathbf{s}, \mathbf{x}_p) &= \mathbf{K}^{02}(\mathbf{x}_p, \mathbf{s})^\top, & \mathbf{K}^{22}(\mathbf{s}, \mathbf{s}) &= [k^{22}(s, s')] \end{aligned}$$

Next, assign to  $U(\cdot)$  a GP prior, and consider aiming at an estimated function that is  $L_U$ -smooth and  $\gamma_U$ -strongly convex, for a given  $L_U > 0$  and  $\gamma_U > 0$ . Then, following [10], Lemma 3.1], the joint conditional posterior distribution of  $(\mathbf{U}(x^\circ)|\mathbf{U}''(\mathbf{s}), \mathbf{x}_p, \mathbf{z})$ , for a point  $x^\circ \in \mathbb{R}$ , given the current observations  $\mathbf{z}$ , is a GP with mean, covariance, and standard deviation given by

$$\begin{aligned} \bar{\mu}(x^\circ) &= \mu + B_3(\mathbf{x}_p, x^\circ, \mathbf{s}) B_1(\mathbf{x}_p, \mathbf{s})^{-1} (\mathbf{z} - \mu \mathbf{1}_p) \\ &\quad + (A_2(x^\circ, \mathbf{s}) - B_3(\mathbf{x}_p, x^\circ, \mathbf{s}) B_1(\mathbf{x}_p, \mathbf{s})^{-1} A_1(\mathbf{x}_p, \mathbf{s})) \mathbf{U}''(\mathbf{s}) \end{aligned} \quad (8a)$$

$$\bar{k}(x^\circ, x^\circ) = A(\mathbf{x}_p, x^\circ, \mathbf{s}) \quad (8b)$$

$$\bar{\sigma}(x^\circ) = \sqrt{A(\mathbf{x}_p, x^\circ, \mathbf{s})} \quad (8c)$$

and the posterior distribution of  $(\mathbf{U}''(\mathbf{s})|\mathbf{x}_p, \mathbf{z})$  is given by

$$(\mathbf{U}''(\mathbf{s})|\mathbf{x}_p, \mathbf{z}) \propto \mathcal{N}(\mu(\mathbf{s}), \mathbf{S}(\mathbf{s}, \mathbf{s})) \mathbf{1}_{\{\gamma_U \leq U''(s_i) \leq L_U, i=1, \dots, q\}}$$

where  $\mathbf{1}_{\{\cdot\}}$  is a set indicator function,  $(\mathbf{U}''(\mathbf{s})|\mathbf{x}_p, \mathbf{z})$  is a truncated normal distribution and

$$\mu(\mathbf{s}) = \mathbf{K}^{20}(\mathbf{s}, \mathbf{x}_p) (\sigma^2 \mathbf{I} + \mathbf{K}(\mathbf{x}_p, \mathbf{x}_p))^{-1} (\mathbf{z} - \mu \mathbf{1}_p)$$

$$\mathbf{S}(\mathbf{s}, \mathbf{s}) = \mathbf{K}^{22}(\mathbf{s}, \mathbf{s})$$

$$- \mathbf{K}^{20}(\mathbf{s}, \mathbf{x}_p) (\sigma^2 \mathbf{I} + \mathbf{K}(\mathbf{x}_p, \mathbf{x}_p))^{-1} \mathbf{K}^{02}(\mathbf{x}_p, \mathbf{s})$$

$$A_1(\mathbf{x}_p, \mathbf{d}) = \mathbf{K}^{02}(\mathbf{x}_p, \mathbf{s}) \mathbf{K}^{22}(\mathbf{s}, \mathbf{s})^{-1}$$

$$A_2(x^\circ, \mathbf{s}) = \mathbf{K}^{02}(x^\circ, \mathbf{s}) \mathbf{K}^{22}(\mathbf{s}, \mathbf{s})^{-1}$$

$$B_1(\mathbf{x}_p, \mathbf{s}) = \sigma^2 \mathbf{I} + \mathbf{K}(\mathbf{x}_p, \mathbf{x}_p)$$

$$- \mathbf{K}^{02}(\mathbf{x}_p, \mathbf{s}) \mathbf{K}^{22}(\mathbf{s}, \mathbf{s})^{-1} \mathbf{K}^{20}(\mathbf{s}, \mathbf{x}_p)$$

$$B_2(x^\circ, \mathbf{s}) = \mathbf{K}(x^\circ, \mathbf{x}_p)$$

$$- \mathbf{K}^{02}(x^\circ, \mathbf{s}) \mathbf{K}^{22}(\mathbf{s}, \mathbf{s})^{-1} \mathbf{K}^{20}(\mathbf{s}, x^\circ)$$

$$B_3(\mathbf{x}_p, x^\circ, \mathbf{s}) = \mathbf{K}(x^\circ, \mathbf{x}_p)$$

$$- \mathbf{K}^{02}(x^\circ, \mathbf{s}) \mathbf{K}^{22}(\mathbf{s}, \mathbf{s})^{-1} \mathbf{K}^{20}(\mathbf{s}, \mathbf{x}_p)$$

$$A(\mathbf{x}_p, x^\circ, \mathbf{s}) = B_2(x^\circ, \mathbf{s})$$

$$- B_3(\mathbf{x}_p, x^\circ, \mathbf{s}) B_1(\mathbf{x}_p, \mathbf{s})^{-1} B_3(\mathbf{x}_p, x^\circ, \mathbf{s})^\top$$

with  $\mu$  and  $\sigma^2$  given parameters of the prior. The hyperparameters  $\ell$  and  $\sigma_f^2$  of the GP can be estimated, for example, by using the maximum likelihood estimator [31]. On the other hand,  $L_U$  and  $\gamma_U$  can be estimated via cross validation. The locations of the virtual derivative points are defined beforehand. Note that the shape-constrained GP is a GP whose second-order derivative is constrained on a set of points. When we consider a GP prior

**Algorithm 1:** GP-Based Online Primal–Dual Method.

```

Initialize:  $\mathbf{x}^1, \lambda^1 = \mathbf{0}, \nu^1 = 0$ , set the constant step size  $\alpha$ ;
prior on  $\{\hat{U}_{m,n}^1\}$  if available.
1: for  $t = 1, 2, \dots$  do
2:   Collect measurement of  $\mathbf{y}^t$ 
3:   for each system  $m = 1, 2, \dots, M$  do
4:     Update the input  $x_m^t$  via (5a)
5:     Send  $x_m^t$  to users  $n \in \mathcal{N}_m$ 
6:     Update  $\nu^t$  via (5c)
7:     for each user  $n \in \mathcal{N}_m$  do
8:       if Feedback is given by user  $n$ 
9:         Update  $\hat{U}_{m,n}^t$  via (8a)
10:      else
11:        Keep  $\hat{U}_{m,n}^t = \hat{U}_{m,n}^{t-1}$ 
12:      end if
13:      Estimate  $g_{m,n}^t$  via (9)
14:      Update  $x_{m,n}^t$  via (5b)
15:      Update  $\lambda_{m,n}^t$  via (5d)
16:      Send  $x_{m,n}^t$  to system  $m$ 
17:    end for (users)
18:    System  $m$  updates  $\lambda_{m,n}^t$  via (5d)
19:  end for (systems)
20: end for (time)

```

with a differentiable kernel function, we have the advantage that their derivative processes are also GPs and are jointly Gaussian with the original processes [10].

In this context, shape-constrained GPs guarantee that the posterior mean function  $\bar{\mu}(x^\circ)$  is practically<sup>4</sup> smooth and strongly convex [10]. We will now incorporate the estimates of the users' function in the algorithm as explained next.

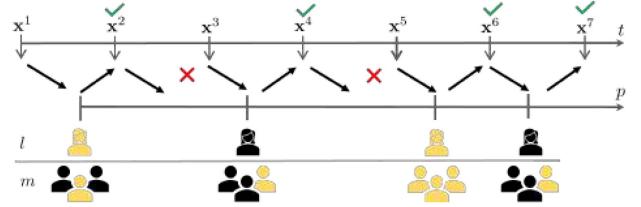
**Remark 1:** One may want to select a dense set of points  $s$  that ensures a uniform covering of the domain. However, it is important to note that the number of points  $q$  influences the dimensions of the matrices  $\mathbf{K}^{20}$ ,  $\mathbf{K}^{02}$ , and  $\mathbf{K}^{22}$ ; therefore,  $q$  should be selected to ensure a sufficiently dense covering, but based on computational complexity considerations.

**Remark 2:** In this article, we use the SE kernel. In this case, two hyperparameters are required for  $U_{m,n} : \ell$  and  $\sigma_f^2$ . These two hyperparameters are learned by maximizing the likelihood [31]. Note that the ability to impose shape constraints on the posterior GP is not affected by the kernel function or the hyperparameters, see [10].

### C. Distributed GP-Based Online Algorithm

Using the shape-constrained GPs machinery described in the previous section, the idea is to utilize the posterior mean (8a) as a surrogate for  $U_{m,n}$ . In particular, for a given user  $n \in \mathcal{N}_m$ , the posterior mean (8a) is computed based on the  $p_{m,n}^t$  functional evaluations  $\{U_{m,n}(x_m^{t_i}) + \epsilon_i\}_{i=1}^{p_{m,n}^t}$ ,  $t_i \in \{1, \dots, t\}$ , provided by

<sup>4</sup>With a limited number of enforcing points  $s$ , the mean has the functional properties we require up to a very (almost negligible) small error, which is smaller and smaller increasing the number of data points  $\mathbf{x}_p$ . The error will be incorporated into the gradient estimation inaccuracies.



**Fig. 2.** Feedback (color coded in yellow) from users may come sporadically, and each user may provide feedback at different times (green ticks indicate that a functional evaluation is received, red crosses indicate that a functional evaluation is not received). The functions  $\{U_{m,n}\}$  are estimated concurrently with the execution of Algorithm 1.

the user up to time  $t$ , that is, we set  $\hat{U}_{m,n}^t(x) = \bar{\mu}_{p_{m,n}^t}(x)$ . As pictorially illustrated in Fig. 2, functional evaluations are provided sporadically by each user, and each user may provide feedback at different times. Furthermore, feedback is not required at each step of the online algorithm as in zeroth-order methods.

Once the posterior mean is available, an estimate of the gradient can be obtained via, e.g., multivariable zeroth-order methods when  $\mathbf{x}_m^t \in \mathbb{R}^n$ , or via finite difference when  $x_m^t \in \mathbb{R}$ . As an example of the latter, one way to obtain  $g_{m,n}^t$  is

$$g_{m,n}^t = \frac{\hat{U}_{m,n}^t(x_m^t + \delta/2) - \hat{U}_{m,n}^t(x_m^t - \delta/2)}{\delta} \quad (9)$$

with  $\delta$  a preselected parameter. The overall algorithm with feedback from both, users and the network, is tabulated as Algorithm 1. In terms of operation, the updates (5a) and (5c) are implemented at each system  $m$ , with  $\{x_m^t\}$  being physical inputs for the system; on the other hand, the update (5b) is implemented at the user's side. The updates (5d) are implemented by both users and systems, and users and systems are required to exchange the local variables  $x_m^t$  and  $x_{m,n}^t$ .

## IV. PERFORMANCE ANALYSIS

In this section, we investigate the convergence of the online algorithm presented in Section III. To this end, the following standard assumptions are presumed.

**Assumption 1:** The set  $\mathcal{X}^t$  is convex and compact for all  $t$ .

**Assumption 2:** The functions  $\mathbf{x} \mapsto f(\mathbf{x})$  and  $\mathbf{x} \mapsto C^t(\bar{\mathbf{A}}^t \mathbf{x} + \mathbf{B}^t \mathbf{w}^t)$  are convex and continuously differentiable for all  $\{x_m, x_{m,n} \in \mathcal{X}_m^t\}$ .

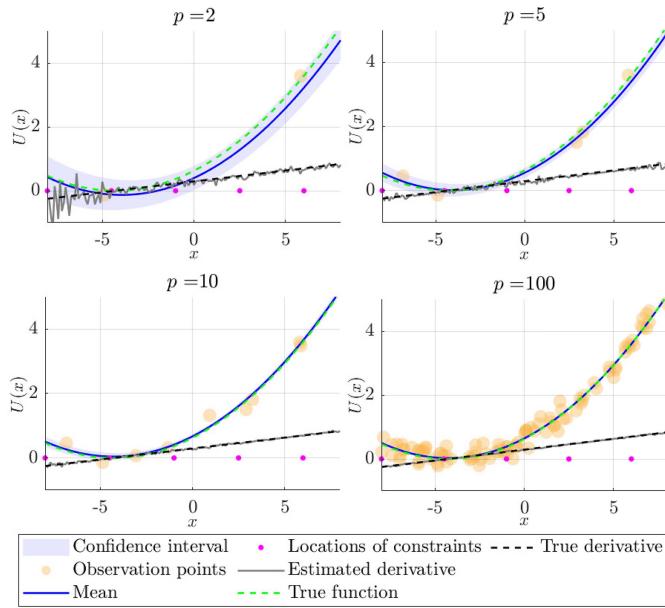
The previous assumptions imply that  $f$  and  $C^t$  are Lipschitz continuous, their gradients are also bounded by their Lipschitz constant, and they are uniformly bounded for all  $\mathbf{x}^t \in \mathcal{X}^t$  and for all  $t$ . Let  $L$  be the Lipschitz constant of  $f$ . The next assumption pertains to each user–system star network.

**Assumption 3:** Each system  $m$  is connected with its users via a star network [induced by (3c)]. The largest singular value of the incidence matrix  $\mathbf{D}$  is  $\Omega$ .

Regarding Assumption 3, each system–user network has a diameter  $h_m \leq 2$ .

**Assumption 4:** Slater's constraint qualification holds  $\forall t$ .

**Assumption 5 (see [17]):** The convex set  $\Lambda^t$  is included in a 2-norm ball of radius  $B_\lambda$ , where  $B_\lambda \geq N_{\max} h_{\max} LM + 1$ ,  $h_{\max} = \max_{i=1, \dots, M} \{h_i\}$ , and  $N_{\max} = \max_{i=1, \dots, M} \{N_i\}$  for all  $t$ .



**Fig. 3.** Example of the derivative estimation using shape-constrained GPs with hyperparameters  $\sigma_f = 1$  and  $\ell = 10$ . As expected, the estimated derivative error decreases with the increase of the number of observations  $p$ . In this case, shape-constrained GP outperforms GP for low numbers of samples (see also Section V).

**Assumption 6 (see [1]):** The set  $\Psi^t$  for the dual variable  $\nu$  is convex and compact for all  $t$ .

We note that, under the Slater's constraint qualification, the dual variables  $\nu^t$  are bounded, as shown in [1], Sec. 3.1]. Additional comments on Assumptions 5–6 will be provided in the next section.

We tackle the analysis by viewing our algorithm as an online primal–dual with *inexact* gradient information [32]. In particular, the approximate gradient (obtained based on a finite difference method based on the estimated functions  $\{\hat{U}_{m,n}^t\}$ ) can be expressed as  $\mathbf{g}^t = \nabla f(\mathbf{x}^t) + \mathbf{e}^t$ , where  $\nabla f(\mathbf{x}^t)$  is the true gradient and  $\mathbf{e}^t$  is a stochastic vector. Before stating appropriate assumptions, define the filtration  $\mathcal{F}_t = \{\mathbf{e}^1, \dots, \mathbf{e}^{t-1}\}$ . The following assumption is made.

**Assumption 7:**  $\exists \bar{E}^t < \infty$  such that  $\mathbb{E}[\|\mathbf{e}^t\|^2 | \mathcal{F}_t] \leq \bar{E}^t \quad \forall t$ .

Note that the error  $\mathbf{e}^t$  is not assumed to have zero mean, since the GP may introduce a bias in the estimated gradient. Assumption 7 presumes that the expected value of  $\|\mathbf{e}^t\|^2$  is bounded; this is consistent with standard assumptions in the context of inexact gradient methods and stochastic gradient methods (see, e.g., [32] and [33]). We also note that  $\bar{E}^t$  can be taken to be the Bayes errors for regression with GPs; in fact, Wang and Berger [10] showed that the error of the GP regression is higher than the error incurred by the shape-constrained GP regression. We point out that Assumption 7 is also supported by empirical evidence; an example for the shape-constrained GP is provided in Fig. 3, which illustrates the estimated function and the estimated derivative for different numbers of noisy functional evaluations. With these standard assumptions in place, the next section will provide performance bounds for the proposed algorithm.

### A. Dynamic Regret

We characterize the performance of Algorithm 1 using the so-called dynamic regret (see, e.g., [17], [18], [21], [22], and references therein). In particular, given the distributed nature of our algorithm, we consider an extension of the network regret [17], [18] to a time-varying setting. To this end, define the *dynamic regret* per user  $j \in \mathcal{N}_m$  interacting with the system  $m$ , with respect to an optimal solution of (3) for the system  $m$  at time  $t$ , i.e.,  $x_m^{t*}$ , as

$$\text{Reg}_T^{m,j} := \sum_{t=1}^T \sum_{i=1}^{N_m} U_{m,i}(x_{m,j}^t) - \sum_{t=1}^T \sum_{i=1}^{N_m} U_{m,i}(x_m^{t*}). \quad (10)$$

Then, based on (10), we define the *dynamic network regret* for the system  $m$  as [17], [18]

$$\begin{aligned} \text{Reg}_T^m &= \frac{1}{N_m} \sum_{j=1}^{N_m} \text{Reg}_T^{m,j} \\ &= \frac{1}{N_m} \sum_{t=1}^T \sum_{i,j=1}^{N_m} U_{m,i}(x_{m,j}^t) - \sum_{t=1}^T \sum_{i=1}^{N_m} U_{m,i}(x_m^{t*}). \end{aligned} \quad (11)$$

By using (11), the dynamic *global network regret* after  $T$  iterations can be defined as

$$\text{Reg}_T := \sum_{m=1}^M \text{Reg}_T^m. \quad (12)$$

To analyze the global network regret, let  $\mathbf{x}^{t*}$  be an optimal solution of (3) at time  $t$ , and consider the following standard definitions of the path length, which captures the drift of the optimal solutions over  $T$  time steps

$$\Phi^T := \sum_{t=1}^T \|\mathbf{x}^{t*} - \mathbf{x}^{t+1*}\|, \quad \Upsilon^T := \sum_{t=1}^T \|\mathbf{x}^{t*} - \mathbf{x}^{t+1*}\|^2. \quad (13)$$

Furthermore, consider the following definitions for the cumulative expected error in the gradient:

$$\xi^T := \sum_{t=1}^T \mathbb{E}[\|\mathbf{e}^t\| | \mathcal{F}_t], \quad \Xi^T := \sum_{t=1}^T \bar{E}^t. \quad (14)$$

With these metrics in place, we start by bounding the norm of the true gradient of the time-varying Lagrangian with respect to the primal variable for any  $\mathbf{x} \in \mathcal{X}^t$ ,  $\nu \in \Psi^t$ , and  $\lambda \in \Lambda^t$ . Let  $J := \sup_{t \in \mathcal{T}} \max_{\mathbf{x} \in \mathcal{X}^t} \|\nabla C^t(\mathbf{y}^t(\mathbf{x}))\|$  by Assumptions 1 and 2,  $B_\lambda := \sup_{t \in \mathcal{T}} \max_{\lambda \in \Lambda^t} \|\lambda\|$  and  $B_\nu := \sup_{t \in \mathcal{T}} \max_{\nu \in \Psi^t} |\nu|$  under Assumptions 5 and 6, then we have

$$\begin{aligned} \|\nabla_{\mathbf{x}} \mathcal{L}^t(\mathbf{x}, \nu, \lambda)\| &\leq \|\nabla f(\mathbf{x})\| + \nu \|\nabla C^t(\mathbf{y}^t(\mathbf{x}))\| + \|\mathbf{D}^\top \lambda\| \\ &\leq L + B_\nu J + \Omega B_\lambda := \Gamma_x \end{aligned} \quad (15)$$

where the last inequality holds by using the triangle and Cauchy–Schwarz inequality and Assumption 3. The term  $\Gamma_x = L + B_\nu J + \Omega B_\lambda$  is an upper bound for the gradient of the Lagrangian with respect to  $\mathbf{x}$ .

Similarly, we bound the norm of the gradient of the Lagrangian with respect to the dual variables  $\lambda$  and  $\nu$  for any

$\mathbf{x} \in \mathcal{X}^t$ ,  $\nu \in \Psi^t$  and  $\lambda \in \Lambda^t$ . Let  $\kappa = [\nu, \lambda^\top]^\top$ ,  $|C^t(\mathbf{y}^t(\mathbf{x}))| \leq H < \infty$  for all  $t$  by Assumptions 1 and 2, and  $B_x := \sup_{t \in \mathcal{T}} \max_{\mathbf{x} \in \mathcal{X}^t} \|\mathbf{x}\|$  under Assumption 1. Then,  $\|\nabla_{\kappa} \mathcal{L}^t(\mathbf{x}, \kappa)\|^2$  can be bounded by

$$\begin{aligned} \|\nabla_{\kappa} \mathcal{L}^t(\mathbf{x}, \kappa)\|^2 &= \left\| \begin{array}{c} \nabla_{\lambda} \mathcal{L}^t(\mathbf{x}, \nu, \lambda) \\ \nabla_{\nu} \mathcal{L}^t(\mathbf{x}, \nu, \lambda) \end{array} \right\|^2 \\ &\leq \Omega^2 B_x^2 + H^2 := \Gamma_{\kappa} \end{aligned} \quad (16)$$

where we used the Cauchy–Schwarz inequality, Assumptions 1 and 3. Furthermore, let  $\text{diam}(\mathcal{X}^t) \leq D_x^t$  where  $D_x^t \leq D_x < \infty$  by Assumption 1. The term  $\Gamma_{\kappa} = \Omega^2 B_x^2 + H^2$  is an upper bound for the gradient of the Lagrangian with respect to  $\kappa$ .

With these notations in place, we are now ready to state the main result for the dynamic regret.

**Theorem 1:** Let Assumptions 1–7 hold. Set  $\lambda^1 = \mathbf{0}$ ,  $\nu^1 = 0$ . Then, the dynamic network regret after  $T$  iterations is upper bounded by

$$\begin{aligned} \mathbb{E}[\text{Reg}_T] &\leq \frac{1}{2\alpha} (\|\mathbf{x}^1 - \mathbf{x}^{1*}\|^2 + B_{\lambda}^2 + B_{\nu}^2) + \frac{\alpha}{2} T (\Gamma_x^2 + \Gamma_{\kappa}) \\ &\quad + \frac{\alpha}{2} \Xi^T + \frac{1}{2} \xi^T (2B_x + \alpha \Gamma_x) + \frac{1}{2\alpha} \Upsilon^T + \frac{1}{\alpha} D_x \Phi^T. \end{aligned} \quad (17)$$

The proof of this theorem will be provided in Section IV-C. Theorem 1 asserts that the bound on the expected dynamic network regret  $\mathbb{E}[\text{Reg}_T]$  after  $T$  time steps depends on the temporal variability of the solutions (through the terms  $\frac{1}{2\alpha} \Upsilon^T$  and  $\frac{1}{\alpha} D_x \Phi^T$ ) and the errors associated with the estimation of the gradient [through the terms  $\frac{\alpha}{2} \Xi^T$  and  $\frac{1}{2} \xi^T (2B_x + \alpha \Gamma_x)$ ]. The remaining terms are in line with standard regret results (e.g., [17]). In particular, we can observe the following.

- 1) *Persistent variations and gradient errors:* Suppose that  $\Upsilon^T$  and  $\Phi^T$  grow as  $\mathcal{O}(T)$ , modeling a persistent temporal variation of the constraints and/or the vector  $\mathbf{w}^t$ ; suppose further that  $\Xi^T$  and  $\xi^T$  grow as  $\mathcal{O}(T)$ , modeling a persistent error in the gradient  $\mathbf{g}^t$ . Then,  $\mathbb{E}[\text{Reg}_T]/T$  behaves as  $\mathcal{O}(1)$ .
- 2) *Vanishing gradient errors:* If  $\Xi^T$  and  $\xi^T$  grow sublinearly, i.e., as  $o(T)$ , then the asymptotic average regret is in general  $\frac{1}{T} \mathbb{E}[\text{Reg}_T] = \mathcal{O}(1 + T^{-1} \Upsilon^T + T^{-1} \Phi^T)$ . Note that if the algorithm is executed over a finite interval of  $T$  steps and the step size is chosen as  $\alpha = \frac{1}{\sqrt{T}}$ , then,  $\mathbb{E}[\text{Reg}_T]/T$  behaves as  $\frac{1}{T} \mathbb{E}[\text{Reg}_T] = \mathcal{O}(T^{-\frac{1}{2}} + T^{-\frac{1}{2}} \Upsilon^T + T^{-\frac{1}{2}} \Phi^T)$ .
- 3) *Vanishing variations and errors:* If  $\Upsilon^T$ ,  $\Phi^T$ ,  $\Xi^T$ , and  $\xi^T$  all grow as  $o(T)$ , then, we recover the asymptotic result of [17].

**Remark 3:** Regarding the errors in the gradient, if the function  $U_{m,n}$  is strongly convex, then the estimation error yielded by shape-constrained GPs will decrease with the increase of the number of functional evaluations, see the illustrative example in Fig. 3 and the discussion in [10]. However, zeroth-order and finite-different methods will still generate an error in the estimation of the gradient, and thus,  $\Xi^T$  and  $\xi^T$  would generally increase as  $\mathcal{O}(T)$ . To obtain a trend as  $o(T)$ , one has to increase the accuracy in the gradient estimation to make  $\mathbf{e}^t$  arbitrarily small.  $\square$

**Remark 4:** Assumption 4 guarantees boundedness of the dual variable  $\nu^t$ , at each  $t$ , as shown in [1], Sec. 3.1]. On the other hand, we note that Assumption 5 does not provide an analytical expression for an upper bound on the radius of  $\Lambda^t$ . This is one limitation of the technical findings of this article. However, we note that in our algorithm, we project the dual variables  $\lambda^t$  onto  $\Lambda^t$ ; if  $\Lambda^t$  is not “large enough” to contain the optimal multipliers  $\lambda^{t*}$ , then one would incur a consensus error due to clipping of the multipliers. This is in the spirit of a relaxation of the constraints with dual smoothing; see, for example, [24], where a regularization term is added to the Lagrange function to avoid the growth of the dual multiplier. In a practical implementation of the algorithm, clipping leads to an increased consensus error, in the case of persistent and large consensus errors, the radius of  $\Lambda^t$  can be increased. A detailed analysis of the boundedness of this set is a future work direction.

## B. Average Constraint Violation

We now consider a bound on the violation of the constraint on the network output  $\mathbf{y}^t$ . To this end, we consider the so-called ACV, which we define here as

$$\text{ACV}_T := \sum_{t=1}^T [C^t(\mathbf{y}^t(\mathbf{x}^t))]^+. \quad (18)$$

We note that some prior works in context (e.g., [19] and [34]) consider the modified definition  $\left[ \sum_{t=1}^T C^t(\mathbf{y}^t(\mathbf{x}^t)) \right]^+$ , which may lead to looser bounds.

**Theorem 2:** Let Assumptions 1–7 hold. Then, the ACV incurred by the algorithm (5) can be bounded as follows:

$$\begin{aligned} \mathbb{E}[\text{ACV}_T] &\leq B_{\nu}^{-1} T (D_x L + B_{\lambda} \Omega B_x) \\ &\quad + B_{\nu}^{-1} \left( \frac{\alpha}{2} \Xi^T + \xi^T (2B_x + \alpha \Gamma_x) + \frac{1}{2\alpha} \Upsilon^T + \frac{1}{\alpha} D_x \Phi^T \right) \\ &\quad + B_{\nu}^{-1} T \left( \frac{1}{\alpha} (4B_x^2 + B_{\nu}^2) + \frac{\alpha}{2} (\Gamma_x^2 + H^2) \right). \end{aligned} \quad (19)$$

The second and third terms of (19) exhibit the same asymptotic behavior of the dynamic regret under persistent variations and gradient errors, and with vanishing variations and errors. In the case of vanishing variations and gradient errors, the bottlenecks are the two terms  $T D_x L$  and  $T B_{\lambda} \Omega B_x$ , which make  $\mathbb{E}[\text{ACV}_T]/T$  behave as  $\mathcal{O}(1)$ .

## C. Proofs of the Results

To streamline exposition, define  $\phi^t := \|\mathbf{x}^{t*} - \mathbf{x}^{t+1*}\|$  and  $\bar{e}^t := \mathbb{E}[\|\mathbf{e}^t\| \mid \mathcal{F}_t]$ . To derive our main results, we will utilize the following lemma.

**Lemma 1:** Let Assumptions 1–7 hold, and set  $\lambda^1 = \mathbf{0}$ ,  $\nu^1 = 0$ . Then, for any  $\kappa \in \Psi^t \times \Lambda^t$  the following holds:

$$\begin{aligned} \mathbb{E} \left[ \sum_{t=1}^T (\mathcal{L}^t(\mathbf{x}^t, \kappa) - \mathcal{L}^t(\mathbf{x}^{t*}, \kappa^t)) \right] &\leq \frac{1}{2\alpha} (\|\mathbf{x}^1 - \mathbf{x}^{1*}\|^2 + \|\kappa\|^2) + \frac{\alpha}{2} T (\Gamma_x^2 + \Gamma_{\kappa}) \\ &\quad + \frac{\alpha}{2} \Xi^T + \xi^T (2B_x + \alpha \Gamma_x) + \frac{1}{2\alpha} \Upsilon^T + \frac{1}{\alpha} D_x \Phi^T. \end{aligned} \quad (20)$$

**Proof:** Note first that

$$\|\tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda}) - \nabla_{\mathbf{x}} \mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda})\| = \|\mathbf{e}^t\| \quad (21)$$

where  $\tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda})$  is the inexact gradient of the Lagrangian function (where we utilize  $\mathbf{g}^t$  instead of  $\nabla f(\mathbf{x}^t)$ ). Recall that  $\mathbf{x}^{t*}$  denotes an optimal solution at time  $t$ . Using the primal update, adding and subtracting  $\mathbf{x}^{t*}$ , we get:

$$\begin{aligned} \|\mathbf{x}^{t+1} - \mathbf{x}^{t+1*}\|^2 &= \|\mathbf{x}^{t+1} - \mathbf{x}^{t*} + \mathbf{x}^{t*} - \mathbf{x}^{t+1*}\|^2 \\ &= \|\mathbf{x}^{t+1} - \mathbf{x}^{t*}\|^2 + 2(\mathbf{x}^{t+1} - \mathbf{x}^{t*})^\top (\mathbf{x}^{t*} - \mathbf{x}^{t+1*}) \\ &\quad + \|\mathbf{x}^{t*} - \mathbf{x}^{t+1*}\|^2 \\ &= \left\| \text{proj}_{\mathcal{X}^t} \left\{ \mathbf{x}^t - \alpha \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t) \right\} - \mathbf{x}^{t*} \right\|^2 + (\phi^t)^2 \\ &\quad + 2(\mathbf{x}^{t+1} - \mathbf{x}^{(*,t)})^\top (\mathbf{x}^{t*} - \mathbf{x}^{t+1*}) \\ &\leq \|\mathbf{x}^t - \alpha \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t) - \mathbf{x}^{t*}\|^2 + (\phi^t)^2 \\ &\quad + 2\|\mathbf{x}^{t+1} - \mathbf{x}^{t*}\| \|\mathbf{x}^{t*} - \mathbf{x}^{t+1*}\| \\ &\leq \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t)\|^2 + (\phi^t)^2 + 2D_x \phi^t \end{aligned} \quad (22)$$

where (22) holds by the nonexpansiveness property of the projection operator, the Cauchy–Schwarz inequality, and the definition of  $\phi^t$ .

By adding and subtracting  $\alpha \nabla_{\mathbf{x}} \mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t)$  [to simplify notation we use  $\nabla_{\mathbf{x}} \mathcal{L}^t = \nabla_{\mathbf{x}} \mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t)$ ], and expanding the first term of the right-hand side (RHS) of (22), we get

$$\begin{aligned} \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t\|^2 &= \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t + \alpha (\nabla_{\mathbf{x}} \mathcal{L}^t - \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t)\|^2 \\ &= \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t\|^2 + \alpha^2 \|\nabla_{\mathbf{x}} \mathcal{L}^t - \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t\|^2 \\ &\quad + 2\alpha (\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t)^\top (\nabla_{\mathbf{x}} \mathcal{L}^t - \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t) \\ &\leq \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t\|^2 + \alpha^2 \|\nabla_{\mathbf{x}} \mathcal{L}^t - \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t\|^2 \\ &\quad + 2\alpha \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t\| \|\nabla_{\mathbf{x}} \mathcal{L}^t - \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t\|. \end{aligned} \quad (23)$$

By using (21), we have that (23) can be bounded as

$$\begin{aligned} \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t\|^2 &\leq \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t\|^2 + \alpha^2 \|\mathbf{e}^t\|^2 \\ &\quad + 2\alpha \|\mathbf{e}^t\| \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t\| \\ &\leq \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t\|^2 + \alpha^2 \|\mathbf{e}^t\|^2 \\ &\quad + 2\alpha \|\mathbf{e}^t\| (\|\mathbf{x}^t - \mathbf{x}^{t*}\| + \alpha \|\nabla_{\mathbf{x}} \mathcal{L}^t\|) \\ &\leq \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t\|^2 + \alpha^2 \|\mathbf{e}^t\|^2 + 2\alpha \|\mathbf{e}^t\| (2B_x + \alpha \Gamma_x) \end{aligned} \quad (24)$$

where the last inequality is derived using (15) and the triangle inequality. By expanding the first term on the RHS of (24), we get

$$\begin{aligned} \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \nabla_{\mathbf{x}} \mathcal{L}^t\|^2 &= \|\mathbf{x}^t - \mathbf{x}^{t*}\|^2 + \alpha^2 \|\nabla_{\mathbf{x}} \mathcal{L}^t\|^2 - 2\alpha (\mathbf{x}^t - \mathbf{x}^{t*})^\top \nabla_{\mathbf{x}} \mathcal{L}^t \\ &\leq \|\mathbf{x}^t - \mathbf{x}^{t*}\|^2 + \alpha^2 \Gamma_x^2 \\ &\quad - 2\alpha [\mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t) - \mathcal{L}^t(\mathbf{x}^{t*}, \nu^t, \boldsymbol{\lambda}^t)]. \end{aligned} \quad (25)$$

Since the function  $f(\mathbf{x})$  is convex, the time-varying Lagrangian is convex in  $\mathbf{x}$ . Thus, from the first-order characterization of convexity (25) holds.

Using (25) the bound (24), one has that

$$\begin{aligned} \|\mathbf{x}^t - \mathbf{x}^{t*} - \alpha \tilde{\nabla}_{\mathbf{x}} \mathcal{L}^t\|^2 &\leq \|\mathbf{x}^t - \mathbf{x}^{t*}\|^2 + \alpha^2 \Gamma_x^2 + \alpha^2 \|\mathbf{e}^t\|^2 + 2\alpha \|\mathbf{e}^t\| (2B_x + \alpha \Gamma_x) \\ &\quad - 2\alpha [\mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t) - \mathcal{L}^t(\mathbf{x}^{t*}, \nu^t, \boldsymbol{\lambda}^t)]. \end{aligned} \quad (26)$$

By using (26) in (22), we have

$$\begin{aligned} \|\mathbf{x}^{t+1} - \mathbf{x}^{t+1*}\|^2 &\leq \|\mathbf{x}^t - \mathbf{x}^{t*}\|^2 + \alpha^2 \Gamma_x^2 + \alpha^2 \|\mathbf{e}^t\|^2 + 2\alpha \|\mathbf{e}^t\| (2B_x + \alpha \Gamma_x) \\ &\quad - 2\alpha [\mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t) - \mathcal{L}^t(\mathbf{x}^{t*}, \nu^t, \boldsymbol{\lambda}^t)] + (\phi^t)^2 + 2D_x \phi^t. \end{aligned} \quad (27)$$

It thus follows that

$$\begin{aligned} \mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t) - \mathcal{L}^t(\mathbf{x}^{t*}, \nu^t, \boldsymbol{\lambda}^t) &\leq \frac{1}{2\alpha} (\|\mathbf{x}^t - \mathbf{x}^{t*}\|^2 - \|\mathbf{x}^{t+1} - \mathbf{x}^{t+1*}\|^2) + \frac{\alpha}{2} \Gamma_x^2 \\ &\quad + \frac{\alpha}{2} \|\mathbf{e}^t\|^2 + \|\mathbf{e}^t\| (2B_x + \alpha \Gamma_x) + \frac{1}{2\alpha} (\phi^t)^2 + \frac{1}{\alpha} D_x \phi^t. \end{aligned} \quad (28)$$

Now, we proceed with a similar analysis for the distance between the updates of the dual variables  $\boldsymbol{\lambda}^{t+1}, \nu^{t+1}$  and an arbitrary dual variable, i.e., for any  $\boldsymbol{\lambda} \in \Lambda^t$  and any  $\nu \in \Psi^t$ . We start with the following:

$$\begin{aligned} \|\boldsymbol{\kappa}^{t+1} - \boldsymbol{\kappa}\|^2 &\leq \|\boldsymbol{\kappa}^t + \alpha \nabla_{\boldsymbol{\kappa}} \mathcal{L}^t - \boldsymbol{\kappa}\|^2 \\ &\leq \|\boldsymbol{\kappa}^t - \boldsymbol{\kappa}\|^2 + \alpha^2 \|\nabla_{\boldsymbol{\kappa}} \mathcal{L}^t\|^2 + 2\alpha \nabla_{\boldsymbol{\kappa}} \mathcal{L}^{t\top} (\boldsymbol{\kappa}^t - \boldsymbol{\kappa}) \\ &\leq \|\boldsymbol{\kappa}^t - \boldsymbol{\kappa}\|^2 + \alpha^2 \Gamma_{\boldsymbol{\kappa}} + 2\alpha \nabla_{\boldsymbol{\kappa}} \mathcal{L}^{t\top} (\boldsymbol{\kappa}^t - \boldsymbol{\kappa}) \end{aligned} \quad (29)$$

where  $\nabla_{\boldsymbol{\kappa}} \mathcal{L}^t = \nabla_{\boldsymbol{\kappa}} \mathcal{L}^t(\mathbf{x}^t, \boldsymbol{\kappa}^t)$  for brevity.

The time-varying Lagrangian is concave in  $\boldsymbol{\lambda}$  and  $\nu$ , i.e., any  $\boldsymbol{\kappa}$ , which implies that the time-varying Lagrangian difference for a fixed  $\mathbf{x}^t \in \mathcal{X}^t$  satisfies

$$\mathcal{L}^t(\mathbf{x}^t, \boldsymbol{\kappa}^t) - \mathcal{L}^t(\mathbf{x}^t, \boldsymbol{\kappa}) \geq (\boldsymbol{\kappa}^t - \boldsymbol{\kappa})^\top \nabla_{\boldsymbol{\kappa}} \mathcal{L}^t(\mathbf{x}^t, \boldsymbol{\kappa}). \quad (30)$$

Substituting (30) into (29) for any  $\boldsymbol{\kappa} \in \Psi^t \times \Lambda^t$ , we get

$$\begin{aligned} \mathcal{L}^t(\mathbf{x}^t, \boldsymbol{\kappa}^t) - \mathcal{L}^t(\mathbf{x}^t, \boldsymbol{\kappa}) &\geq \frac{1}{2\alpha} (\|\boldsymbol{\kappa}^{t+1} - \boldsymbol{\kappa}\|^2 - \|\boldsymbol{\kappa}^t - \boldsymbol{\kappa}\|^2) - \frac{\alpha}{2} \Gamma_{\boldsymbol{\kappa}}. \end{aligned} \quad (31)$$

By subtracting the results in (28) and (31), we have

$$\begin{aligned} \mathcal{L}^t(\mathbf{x}^t, \boldsymbol{\kappa}) - \mathcal{L}^t(\mathbf{x}^{t*}, \boldsymbol{\kappa}^t) &\leq \frac{1}{2\alpha} (\|\mathbf{x}^t - \mathbf{x}^{t*}\|^2 - \|\mathbf{x}^{t+1} - \mathbf{x}^{t+1*}\|^2) \end{aligned}$$

$$+\frac{1}{2\alpha}(\|\kappa^t - \kappa\|^2 - \|\kappa^{t+1} - \kappa\|^2) + \|\mathbf{e}^t\|(2B_x + \alpha\Gamma_x) \\ + \frac{\alpha}{2}\|\mathbf{e}^t\|^2 + \frac{1}{2\alpha}(\phi^t)^2 + \frac{1}{\alpha}D_x\phi^t + \frac{\alpha}{2}(\Gamma_x^2 + \Gamma_\kappa). \quad (32)$$

Assume that  $\kappa^1 = \mathbf{0}$ . Then, summing (32) over time, using the telescopic property for the first two terms on the RHS, and by taking the expectation of both sides, the result follows.  $\blacksquare$

Based on Lemma 1, in the following, we provide the proof for Theorem 1.

**1) Proof for Theorem 1:** First, we can rewrite  $\mathcal{L}^t(\mathbf{x}^t, \kappa) - \mathcal{L}^t(\mathbf{x}^{t*}, \kappa^t)$  as

$$\mathcal{L}^t(\mathbf{x}^t, \kappa) - \mathcal{L}^t(\mathbf{x}^{t*}, \kappa^t) = f(\mathbf{x}^t) - f(\mathbf{x}^{t*}) \\ + \underbrace{\kappa^\top \begin{bmatrix} C^t(\mathbf{y}^t(\mathbf{x}^t)) \\ \mathbf{D}\mathbf{x}^t \end{bmatrix} - \kappa^{t\top} \begin{bmatrix} C^t(\mathbf{y}^t(\mathbf{x}^{t*})) \\ \mathbf{D}\mathbf{x}^{t*} \end{bmatrix}}_{:=\beta^t} \quad (33)$$

where we recall that  $\mathbf{D}\mathbf{x}^{t*} = \mathbf{0}$  and  $C^t(\mathbf{y}^t(\mathbf{x}^{t*})) \leq 0$ .

By using the definition of the cost function  $f(\mathbf{x})$  and summing (33) over time, we have

$$\sum_{t=1}^T (\mathcal{L}^t(\mathbf{x}^t, \kappa) - \mathcal{L}^t(\mathbf{x}^{t*}, \kappa^t)) = \sum_{t=1}^T \beta^t \\ + \sum_{t=1}^T \left( \sum_{m=1}^M \sum_{n=1}^{N_m} U_{m,n}(x_{m,n}^t) - \sum_{m=1}^M \sum_{n=1}^{N_m} U_{m,n}(x_m^{t*}) \right). \quad (34)$$

Taking the expected value and using Lemma 1 in (34), we have

$$\mathbb{E} \left[ \sum_{t=1}^T \left( \sum_{m=1}^M \sum_{n=1}^{N_m} U_{m,n}(x_{m,n}^t) - \sum_{m=1}^M \sum_{n=1}^{N_m} U_{m,n}(x_m^{t*}) \right) + \sum_{t=1}^T \beta^t \right] \\ \leq \frac{1}{2\alpha} (\|\mathbf{x}^1 - \mathbf{x}^{1*}\|^2 + \|\kappa\|^2) + \frac{\alpha}{2} T (\Gamma_x^2 + \Gamma_\kappa) \\ + \frac{\alpha}{2} \Xi^T + \xi^T (2B_x + \alpha\Gamma_x) + \frac{1}{2\alpha} \Upsilon^T + \frac{1}{\alpha} D_x \Phi^T. \quad (35)$$

In order to bound the global network regret, we add and subtract the RHS of (34) in (12), to obtain

Reg<sub>T</sub>

$$= \sum_{t=1}^T \left[ \sum_{m=1}^M \left( \frac{1}{N_m} \sum_{i,j=1}^{N_m} U_{m,i}(x_{m,j}^t) - \sum_{n=1}^{N_m} U_{m,n}(x_{m,n}^t) \right) - \beta^t \right] \\ + \sum_{t=1}^T \left[ \left( \sum_{m=1}^M \sum_{n=1}^{N_m} U_{m,n}(x_{m,n}^t) - \sum_{m=1}^M \sum_{n=1}^{N_m} U_{m,n}(x_m^{t*}) \right) + \beta^t \right]. \quad (36)$$

The expected value of the last term of (36) is bounded by (35). Next, focus on the first term in (36). Assumption 2 implies that

$$\sum_{i,j=1}^{N_m} U_{m,i}(x_{m,j}^t) - \sum_{n=1}^{N_m} U_{m,n}(x_{m,n}^t) \\ \leq \sum_{i,j=1}^{N_m} L_{m,i} |x_{m,j}^t - x_{m,i}^t| \quad (37)$$

where  $L_{m,i} \leq L_m \leq L$  per each network composed of a system and the users interacting with such a system. Maximizing over the RHS of (37) we can get an expression for the worst case disagreement per system  $m$  as

$$\sum_{i,j=1}^{N_m} L_{m,i} |x_{m,j}^t - x_{m,i}^t| \leq N_m^2 L_m h_m \|\mathbf{D}_m \mathbf{x}_m^t\| \quad (38)$$

where  $\max_{i,j} |x_{m,j}^t - x_{m,i}^t|$  can be upper bounded using Assumption 3, and  $\mathbf{x}_m := [x_m, x_{m,1}, \dots, x_{m,N_m}] \forall m$ .

Back to the first term in (36), we have that

$$\sum_{t=1}^T \left[ \sum_{m=1}^M \left( \frac{1}{N_m} \sum_{i,j=1}^{N_m} U_{m,i}(x_{m,j}^t) - \sum_{n=1}^{N_m} U_{m,n}(x_{m,n}^t) \right) - \beta^t \right] \\ = \sum_{t=1}^T \left[ \sum_{m=1}^M \frac{1}{N_m} \sum_{i,j=1}^{N_m} (U_{m,i}(x_{m,j}^t) - U_{m,n}(x_{m,n}^t)) - \beta^t \right] \\ \leq \sum_{t=1}^T \sum_{m=1}^M \frac{1}{N_m} (N_m^2 L_m h_m \|\mathbf{D}_m \mathbf{x}_m^t\|) - \sum_{t=1}^T \beta^t \\ \leq \sum_{t=1}^T N_{\max} h_{\max} L M \|\mathbf{D}\mathbf{x}^t\| - \sum_{t=1}^T \beta^t. \quad (39)$$

By the fact that at optimality  $\lambda^\top \mathbf{D}\mathbf{x}^{t*} = 0$  and  $\nu^t C^t(\mathbf{y}^t(\mathbf{x}^{t*})) \leq 0$ , we can rewrite the RHS of (39) as

$$\sum_{t=1}^T N_{\max} h_{\max} L M \|\mathbf{D}\mathbf{x}^t\| - \sum_{t=1}^T \kappa^\top \begin{bmatrix} C^t(\mathbf{y}^t(\mathbf{x}^t)) \\ \mathbf{D}\mathbf{x}^t \end{bmatrix} \\ = \sum_{t=1}^T \left( N_{\max} h_{\max} L M \frac{\mathbf{D}\mathbf{x}^t}{\|\mathbf{D}\mathbf{x}^t\|} - \lambda \right)^\top \mathbf{D}\mathbf{x}^t \\ - \sum_{t=1}^T \nu C^t(\mathbf{y}^t(\mathbf{x}^t)). \quad (40)$$

By construction, we can choose a feasible dual variable  $\bar{\lambda}$  using the compactness of  $\mathcal{X}^t$  and the boundedness of  $\|\mathbf{D}\mathbf{x}^t\|$ , as in [17], so that the first term in (40) equals 0. Similarly, we can set  $\bar{\nu} = 0$ . With this choice for  $\bar{\kappa} = [\bar{\nu}, \bar{\lambda}^\top]^\top$ , the expression on (40) is equal to 0. Then, the expected value of the regret defined in (12) is bounded, as shown in (17).  $\blacksquare$

**2) Proof for Theorem 2:** We start by bounding the distance between the update of the dual variable  $\nu^{t+1}$  and an arbitrary dual variable, i.e., for any  $\nu \in \Psi^t$

$$|\nu^{t+1} - \nu|^2 \leq |\nu^t + \alpha \nabla_\nu \mathcal{L}^t - \nu|^2 \\ \leq |\nu^t - \nu|^2 + \alpha^2 H^2 + 2\alpha \nabla_\nu \mathcal{L}^t(\nu^t - \nu) \quad (41)$$

where  $\nabla_\nu \mathcal{L}^t = \nabla_\nu \mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda}) = C^t(\mathbf{y}^t(\mathbf{x}^t))$  for any feasible  $\boldsymbol{\lambda}$ . Reorganizing (41) for any  $\nu \in \Psi^t$ , we get

$$(\nu^t - \nu) \nabla_\nu \mathcal{L}^t \geq \frac{1}{2\alpha} (|\nu^{t+1} - \nu|^2 - |\nu^t - \nu|^2) - \frac{\alpha}{2} H^2. \quad (42)$$

The time-varying Lagrangian differences for a fixed  $\mathbf{x}^t \in \mathcal{X}^t$  and  $\boldsymbol{\lambda}^t \in \Lambda^t$  satisfies

$$\mathcal{L}^t(\mathbf{x}^t, \nu^t, \boldsymbol{\lambda}^t) - \mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda}^t) = (\nu^t - \nu) \nabla_\nu \mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda}). \quad (43)$$

Replacing (43) in (42), and subtracting the results in (28), for any  $\nu \in \Psi^t$ , we get

$$\begin{aligned} & \mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda}^t) - \mathcal{L}^t(\mathbf{x}^{t*}, \nu^t, \boldsymbol{\lambda}^t) \\ & \leq \frac{1}{2\alpha} (\|\mathbf{x}^t - \mathbf{x}^{t*}\|^2 - \|\mathbf{x}^{t+1} - \mathbf{x}^{t+1*}\|^2) \\ & + \frac{1}{2\alpha} (|\nu^t - \nu|^2 - |\nu^{t+1} - \nu|^2) + \|\mathbf{e}^t\| (2B_x + \alpha\Gamma_x) \\ & + \frac{\alpha}{2} \|\mathbf{e}^t\|^2 + \frac{1}{2\alpha} (\phi^t)^2 + \frac{1}{\alpha} D_x \phi^t + \frac{\alpha}{2} (\Gamma_x^2 + H^2). \end{aligned} \quad (44)$$

Since  $\boldsymbol{\lambda}^{t\top} \mathbf{D}\mathbf{x}^{t*} = 0$ , we can also express  $\mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda}^t) - \mathcal{L}^t(\mathbf{x}^{t*}, \nu^t, \boldsymbol{\lambda}^t)$  as

$$\begin{aligned} & \mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda}^t) - \mathcal{L}^t(\mathbf{x}^{t*}, \nu^t, \boldsymbol{\lambda}^t) = (f(\mathbf{x}^t) - f(\mathbf{x}^{t*})) \\ & + \nu C^t(\mathbf{y}^t(\mathbf{x}^t)) + \boldsymbol{\lambda}^{t\top} \mathbf{D}\mathbf{x}^t - \nu^t C^t(\mathbf{y}^t(\mathbf{x}^{t*})). \end{aligned} \quad (45)$$

Reorganizing terms in (45), and by the fact that  $\nu^t C^t(\mathbf{y}^t(\mathbf{x}^{t*})) \leq 0$ , we have

$$\begin{aligned} \nu C^t(\mathbf{y}^t(\mathbf{x}^t)) & \leq (\mathcal{L}^t(\mathbf{x}^t, \nu, \boldsymbol{\lambda}^t) - \mathcal{L}^t(\mathbf{x}^{t*}, \nu^t, \boldsymbol{\lambda}^t)) \\ & - (f(\mathbf{x}^t) - f(\mathbf{x}^{t*})) - \boldsymbol{\lambda}^{t\top} \mathbf{D}\mathbf{x}^t. \end{aligned} \quad (46)$$

On the other hand, since the cost function is convex, we have

$$f(\mathbf{x}^t) - f(\mathbf{x}^{t*}) \leq \|\mathbf{x}^t - \mathbf{x}^{t*}\| \|\nabla f(\mathbf{x}^{t*})\| \leq D_x L \quad (47)$$

and because  $\mathcal{X}^t$  and  $\Lambda^t$  are compact sets uniformly in time, we further get

$$|\boldsymbol{\lambda}^{t\top} \mathbf{D}\mathbf{x}^t| \leq \|\boldsymbol{\lambda}^t\| \|\mathbf{D}\| \|\mathbf{x}^t\| \leq B_\lambda \Omega B_x. \quad (48)$$

By using (44), (47), and (48) in (46), and reorganizing terms

$$\begin{aligned} C^t(\mathbf{y}^t(\mathbf{x}^t)) & \leq \nu^{-1} \left( \frac{1}{\alpha} (4B_x^2 + B_\nu^2) + D_x L + B_\lambda \Omega B_x \right. \\ & \left. + \frac{\alpha}{2} (\Gamma_x^2 + H^2) + \|\mathbf{e}^t\| (2B_x + \alpha\Gamma_x) + \frac{\alpha}{2} \|\mathbf{e}^t\|^2 \right. \\ & \left. + \frac{1}{2\alpha} (\phi^t)^2 + \frac{1}{\alpha} D_x \phi^t \right). \end{aligned} \quad (49)$$

We now take the max operator, sum over time, and take the expectation of both sides of (49), then

$$\begin{aligned} \mathbb{E} \left[ \sum_{t=1}^T [C^t(\mathbf{y}^t(\mathbf{x}^t))]^+ \right] & \leq T \nu^{-1} \left( \frac{1}{\alpha} (4B_x^2 + B_\nu^2) + D_x L \right. \\ & \left. + B_\lambda \Omega B_x + \frac{\alpha}{2} (\Gamma_x^2 + H^2) \right) + \nu^{-1} (\xi^T (2B_x + \alpha\Gamma_x)) \end{aligned}$$

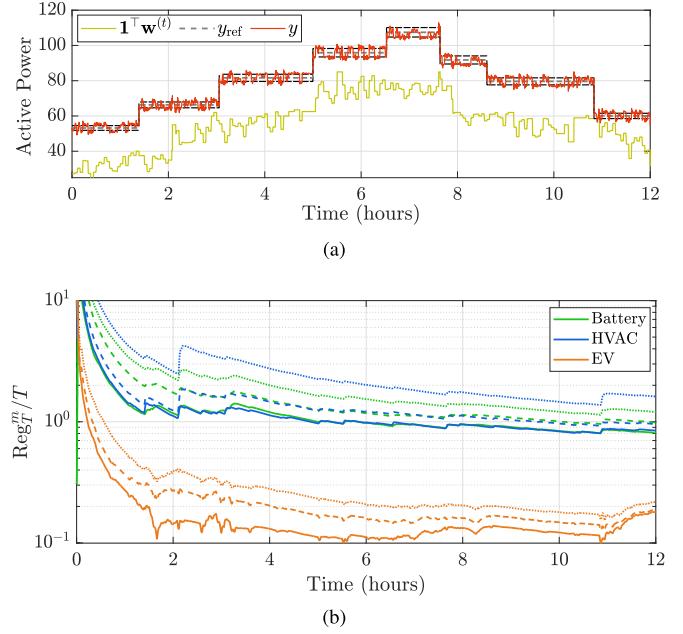


Fig. 4. (a) Tracking of  $y_{\text{ref}}$  by the GP-based online primal-dual method and (b) local network regret (solid line corresponds to true  $\{U_{m,n}\}$ , dashed line to estimate  $\{\hat{U}_{m,n}\}$  via shape-constrained GPs, and dotted line to estimate  $\{\hat{U}_{m,n}\}$  via GPs).

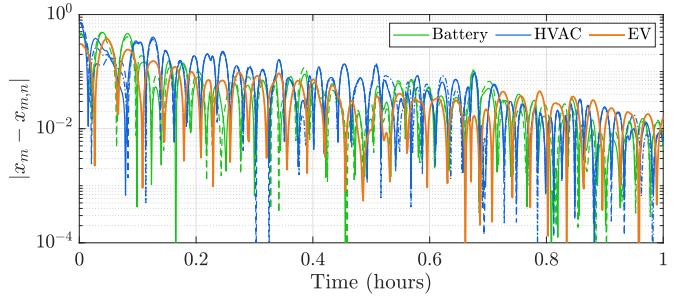


Fig. 5. Disagreement between system and users (zoom for the first hour of simulation where each user is represented by a different line style). The primal variables  $x_m$  and  $x_{m,n}$  are normalized such that  $10^{-2}$  corresponds to 1% error.

$$+ \nu^{-1} \left( \frac{\alpha}{2} \Xi^T + \frac{1}{2\alpha} \Upsilon^T + \frac{1}{\alpha} D_x \Phi^T \right) \forall \nu \in (0, B_\nu].$$

The previous inequality implies that the tightest result happens at  $\nu = B_\nu$ , then (19) holds.  $\blacksquare$

## V. NUMERICAL RESULTS

In this section, the proposed algorithm is numerically evaluated on a problem related to real-time management of DERs at the power distribution level. In particular, we consider the aggregation of controllable DERs as well as uncontrollable loads connected to a distribution transformer. DERs include HVAC systems, EV, and energy storage systems. We consider  $M = 3$  DERs shared among  $N = 6$  users, specifically,  $N_1 = 2$ ,  $N_2 = 3$ ,  $N_3 = 1$ . The goal is to minimize the discomfort or dissatisfaction for each user while enabling the aggregation of DERs to actively emulate a virtual power plant where the

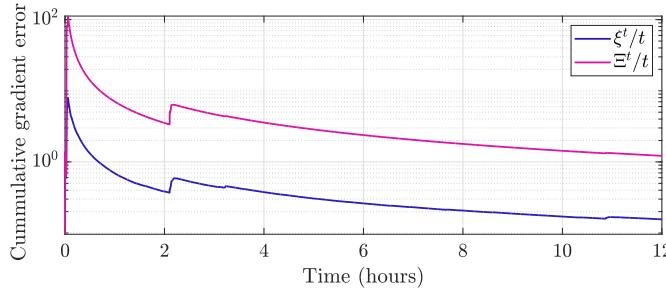


Fig. 6. Behavior of  $\xi^t$  and  $\Xi^t$  over time.

total power  $y^t = \mathbf{1}^\top \mathbf{x}_{\text{in}}^t + \mathbf{1}^\top \mathbf{w}^t$  follows a given automatic gain control or demand response reference signal  $y_{\text{ref}}^t$ . The operational sets for the devices are battery  $\mathcal{X}_1 = [-8, 8]$  kW, HVAC  $\mathcal{X}_2 = [0, 10]$  kW, and EV  $\mathcal{X}_3 = [2, 30]$  kW. We note that, in the context of inverter-interfaced DERs, the set of feasible active power setpoints is typically convex. The discomfort functions  $\{U_{m,n}\}$  are assumed to be quadratic, the minimum of each of the functions is inside the set constraints  $\mathcal{X}_m$ , and it corresponds to a preferred setting of the user. For example, for EVs they represent a preferred charging rate, for HVAC systems, they represent a preferred temperature setpoint (converted into a preferred power setpoint) [14].

The time-varying constraint is  $C^t(\mathbf{x}_{\text{in}}^t) = \frac{\beta}{2}(\mathbf{1}^\top \mathbf{x}_{\text{in}}^t + \mathbf{1}^\top \mathbf{w}^t - y_{\text{ref}}^t)^2 - \zeta^t$ ; where  $\zeta^t$  is a given tolerance, which is set to 5% of the value of  $y_{\text{ref}}^t$ . The power of the uncontrollable loads is taken from the Anatolia dataset (National Renewable Energy Laboratory, Tech. Rep. NREL/TP-5500-56610) and has a granularity of 1 s (see Fig. 4). In this case, the evaluation of the gradient of  $C^t$  requires measurements of the total power at each step  $t$ . We evaluate the performance of the online algorithm at a period of time equivalent to 12 h, each step of Algorithm 1 is performed every 5 s (except for the HVAC systems, which are updated at a slower rate). The priors  $\{\hat{U}_{m,n}\}$  are determined from some noisy measurements ( $\sigma = 1.5$ ) and the discomfort function is updated through the user's feedback every 30 min. In this case, the parameter  $L_U$  and  $\gamma_U$  are defined beforehand; however, these values can be estimated via cross-validation. In Fig. 4(a), it can be seen that the trajectory  $y$  is within 5% of the reference setpoint  $y_{\text{ref}}$  most of the time, despite the variation of the uncontrollable loads.

The local network regret is presented in Fig. 4(b). It can be seen that the jumps in the dynamic regret correspond to instants where the reference  $y_{\text{ref}}$  changes abruptly. The regret of the proposed algorithm is, as one would expect, higher than the clairvoyant case where the functions  $\{U_{m,n}\}$  are known. However, the difference diminishes over time, as the estimation error decreases. Furthermore, Fig. 5 shows the behavior of the users' disagreement during the first hour of the simulation. In addition, Fig. 6 shows numerically how the metrics in (14), related to the accuracy of the gradient estimation of the cost function, decrease over time. Finally, Fig. 7 illustrates the sensitivity of the proposed method to the choice of the parameter  $\delta$  in (9), we also provide the trajectory for the online algorithm with perfect knowledge of the users' function (Online PD) for comparison purposes.

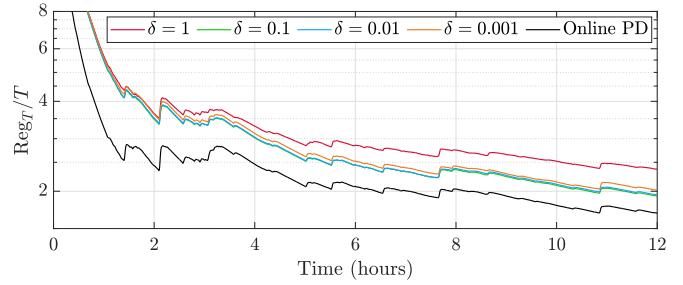


Fig. 7. Global network regret for different values of  $\delta$ .

## VI. CONCLUSION

In this work, we presented an online consensus-based algorithm to solve a time-varying optimization problem associated with a network of systems shared by multiple users. The cost functions that represent the individuals' preferences were learned concurrently with the execution of the algorithm via shape-constrained GPs using users' feedback. We developed an online algorithm based on a primal–dual method, properly modified to accommodate feedback from the users and measurements from the network. We analyzed the performance via dynamic network regret and ACV, and we showed that the bounds of these metrics depend on the temporal variability of the optimal solution set, and the errors associated with the estimation of the gradients. Numerical results were presented in the context of real-time management of DER. Future efforts include investigating the convergence of multitime-scale algorithms, and the case where the users' cost functions are time-varying to reflect preferences that change over time.

## REFERENCES

- [1] J. Koshal, A. Nedić, and U. V. Shanbhag, "Multiuser optimization: Distributed algorithms and error analysis," *SIAM J. Optim.*, vol. 21, no. 3, pp. 1046–1081, 2011.
- [2] B. Turan, C. A. Uribe, H. T. Wai, and M. Alizadeh, "Resilient primal–dual optimization algorithms for distributed resource allocation," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 1, pp. 282–294, Mar. 2021.
- [3] A. Y. Popkov, "Gradient methods for nonstationary unconstrained optimization problems," *Autom. Remote Control*, vol. 66, no. 6, pp. 883–891, 2005.
- [4] A. Simonetto and G. Leus, "Double smoothing for time-varying distributed multiuser optimization," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2014, pp. 852–856.
- [5] M. Fazlyab, C. Nowzari, G. J. Pappas, A. Ribeiro, and V. M. Preciado, "Self-triggered time-varying convex optimization," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 3090–3097.
- [6] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proc. IEEE*, vol. 108, no. 11, pp. 2032–2048, Nov. 2020.
- [7] A. Bajcsy, A. Siththaranjan, C. J. Tomlin, and A. D. Dragan, "Analyzing human models that adapt online," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 2754–2760.
- [8] N. Lassen, F. Goia, S. Schiavon, and J. Pantelic, "Field investigations of a smiley-face polling station for recording occupant satisfaction with indoor climate," *Building Environ.*, vol. 185, 2020, Art. no. 107266.
- [9] B. Lepri, J. Staiano, D. Sangokoya, E. Letouzé, and N. Oliver, "The tyranny of data? The bright and dark sides of data-driven decision-making for social good," in *Transparent Data Mining for Big and Small Data*. Cham, Switzerland: Springer, 2017, pp. 3–24.
- [10] X. Wang and J. O. Berger, "Estimating shape constrained functions using Gaussian processes," *SIAM/ASA J. Uncertainty Quantification*, vol. 4, no. 1, pp. 1–25, 2016.

[11] S. Bolognani and S. Zampieri, "A distributed control strategy for reactive power compensation in smart microgrids," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2818–2833, Nov. 2013.

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

[13] Q. Wang, M. Liu, and J. L. Mathieu, "Adaptive demand response: Online learning of restless and controlled bandits," in *Proc. IEEE Int. Conf. Smart Grid Commun.*, 2014, pp. 752–757.

[14] A. Lesage-Landry and D. S. Callaway, "Dynamic and distributed online convex optimization for demand response of commercial buildings," *IEEE Control Syst. Lett.*, vol. 4, no. 3, pp. 632–637, Jul. 2020.

[15] T. Chen, B. Zhang, H. Pourbabak, A. Kavousi-Fard, and W. Su, "Optimal routing and charging of an electric vehicle fleet for high-efficiency dynamic transit systems," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3563–3572, Jul. 2018.

[16] X. Luo, Y. Zhang, and M. M. Zavlanos, "Socially-aware robot planning via bandit human feedback," in *Proc. ACM/IEEE 11th Int. Conf. Cyber-Phys. Syst.*, 2020, pp. 216–225.

[17] A. Koppel, F. Jakubeic, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5149–5164, Oct. 2015.

[18] S. Lee and M. M. Zavlanos, "Distributed primal-dual methods for online constrained optimization," in *Proc. Amer. Control Conf.*, 2016, pp. 7171–7176.

[19] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic IoT management," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 1276–1286, Feb. 2019.

[20] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 647–662, Jun. 2015.

[21] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Trans. Autom. Control*, vol. 63, no. 3, pp. 714–725, Mar. 2018.

[22] T. Yang, L. Zhang, R. Jin, and J. Yi, "Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 449–457.

[23] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2005, pp. 385–394.

[24] J. Li, C. Gu, Z. Wu, and T. Huang, "Online learning algorithm for distributed convex optimization with time-varying coupled constraints and bandit feedback," *IEEE Trans. Cybern.*, vol. 52, no. 2, pp. 1009–1020, Feb. 2022.

[25] Y. Tang, Z. Ren, and N. Li, "Zeroth-order feedback optimization for cooperative multi-agent systems," in *Proc. IEEE 59th Conf. Decis. Control*, 2020, pp. 3649–3656.

[26] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principles, recent advances, and applications," *IEEE Signal Process. Mag.*, vol. 37, no. 5, pp. 43–54, Sep. 2020.

[27] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012.

[28] I. Notarnicola, A. Simonetto, F. Farina, and G. Notarstefano, "Distributed personalized gradient tracking with convex parametric models," *IEEE Trans. Autom. Control*, to be published, doi: [10.1109/TAC.2022.3147007](https://doi.org/10.1109/TAC.2022.3147007).

[29] A. Simonetto, E. Dall'Anese, J. Monteil, and A. Bernstein, "Personalized optimization with user's feedback," *Automatica*, vol. 131, 2021, Art. no. 109767.

[30] A. M. Ospina, A. Simonetto, and E. Dall'Anese, "Personalized demand response via shape-constrained online learning," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids*, 2020, pp. 1–6.

[31] C. E. Rasmussen, "Gaussian processes for machine learning," in *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[32] D. P. Bertsekas and J. N. Tsitsiklis, "Gradient convergence in gradient methods with errors," *SIAM J. Optim.*, vol. 10, no. 3, pp. 627–642, 2000.

[33] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik, "SGD: General analysis and improved rates," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5200–5209.

[34] X. Yi, X. Li, L. Xie, and K. H. Johansson, "Distributed online convex optimization with time-varying coupled inequality constraints," *IEEE Trans. Signal Process.*, vol. 68, pp. 731–746, 2020.



**Ana M. Ospina** (Graduate Student Member, IEEE) received the B.S. degree in electronic and electrical engineering and the M.S. degree in electrical engineering from the University of Los Andes, Bogotá, Colombia, in 2011 and 2014, respectively. She is currently working toward the Ph.D. degree in systems and controls with the Department of Electrical, Computer and Energy Engineering, University of Colorado Boulder, Boulder, CO, USA.

Her research interests include online optimization and learning, with applications in power and energy systems.



**Andrea Simonetto** received the Ph.D. degree in systems and control from the Delft University of Technology, Delft, The Netherlands, in 2012.

He is currently a Professor with ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France. He was with IBM Research until 2021 and joined ENSTA Paris in 2021. His research interests include optimization, control, and signal processing, with applications in smart energy, transportation, and personalized health.



**Emiliano Dall'Anese** received the Ph.D. degree in information engineering from the Department of Information Engineering, University of Padova, Padua, Italy, in 2011.

He is currently an Assistant Professor with the Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, Boulder, CO, USA, where he is also an affiliated Faculty with the Department of Applied Mathematics.

Dr. Dall'Anese was the recipient of the National Science Foundation CAREER Award in 2020 and the IEEE Power and Energy Society Prize Paper Award in 2021.