

Tracking Moving Agents via Inexact Online Gradient Descent Algorithm

Amrit Singh Bedi, *Student Member, IEEE*, Paban Sarma, and Ketan Rajawat,
Member, IEEE

Abstract

Multi-agent systems are being increasingly deployed in challenging environments for performing complex tasks such as multi-target tracking, search-and-rescue, and intrusion detection. Notwithstanding the computational limitations of individual robots, such systems rely on collaboration to sense and react to the environment. This paper formulates the generic target tracking problem as a time-varying optimization problem and puts forth an inexact online gradient descent method for solving it sequentially. The performance of the proposed algorithm is studied by characterizing its dynamic regret, a notion common to the online learning literature. Building upon the existing results, we provide improved regret rates that not only allow non-strongly convex costs but also explicating the role of the cumulative gradient error. Two distinct classes of problems are considered: one in which the objective function adheres to a quadratic growth condition, and another where the objective function is convex but the variable belongs to a compact domain. For both cases, results are developed while allowing the error to be either adversarial or arising from a white noise process. Further, the generality of the proposed framework is demonstrated by developing online variants of existing stochastic gradient algorithms and interpreting them as special cases of the proposed inexact gradient method. The efficacy of the proposed inexact gradient framework is established on a multi-agent multi-target tracking problem, while its flexibility is exemplified by generating online movie recommendations for Movielens 10M dataset.

Index Terms

Time varying optimization, stochastic optimization, target tracking, gradient descent methods.

The authors are with the department of Electrical Engineering, Indian Institute of Technology Kanpur, Kanpur, Uttar Pradesh, India -208016.

I. INTRODUCTION

Multi-agent systems involve teams of robots capable of accomplishing complex tasks in a coordinated manner [1]–[3]. Fueled by advances in sensing and communications, multi-agent systems are increasingly being used for challenging tasks such as multi-target tracking [4], [5], planetary exploration and mapping [6], search-and-rescue, and intrusion detection [7]. Achieving such team-based goals requires the robotic platforms to not only sense and understand the environment, but also cooperate among themselves through judicious data exchange and fusion. Consequently, resource allocation and optimization becomes an important aspect of the overall motion planning problem. Indeed, the recent trend is to formulate target tracking as a constrained convex optimization problem that must be solved at every time step [8]–[12].

Such time-varying optimization problems have their origins in the control theory literature, where they have been applied to path planning [13] and dynamic parameter tracking problems [14], [15]. Given the limited computational and communications capabilities of the mobile robots, solving the full per-time instant optimization problem before taking the action may not necessarily be viable. Instead, recent works have advocated the use of simpler one-iteration algorithms such as the online interior point, prediction-correction, online ADMM, and gradient descent methods, that have been shown to approach the optimal asymptotically. Leveraging the tools from classical optimization theory, these dynamic optimization algorithms not only admit low-complexity distributed implementations, but are also amenable to analytical performance guarantees.

Online machine learning represents a parallel but closely related development that has been widely applied to solve problems in Big Data [16]. First proposed by [8], the online convex optimization framework models the agents as learners and targets as adversaries. Within this sequential learning paradigm, the learner performs an action and the adversary reveals a corresponding loss function at each time step. The eventual goal of the learner is to minimize the cumulative loss. Recent year have witnessed the development of theoretical guarantees in form of *dynamic regret*, where the performance of the learner is measured against that of an adaptive and time-varying adversary [11], [17], [18].

This paper studies the multi-agent multi-target tracking problem from the lens of online convex optimization. Prompted by the noisy and possibly incorrect target position information available to the agents, we put forth the inexact online gradient descent (IOGD) algorithm. The key

theoretical contribution is the development of improved dynamic regret bounds for non-strongly convex objective functions. Improving the existing results for general convex problems, it is shown that the dynamic regret is bounded by the *path length* of the target. Different from the existing literature, the gradient errors are not required to be unbiased, independent, or identically distributed, and can even be adversarial. The cumulative impact of such errors on the dynamic regret is explicitly studied and characterized. Regret bounds are developed for two classes of problems, namely the ones where the objective function follows the quadratic growth property (P_1), and the ones where the function is convex but with a compact domain (P_2). Interestingly, it is shown that for P_1 with linearly growing path lengths and cumulative errors, target tracking is still possible, albeit with non-vanishing steady state error. Additionally, we provide further motivation for the IOGD algorithm, by developing online variants of the incremental algorithm from [19] and the proximal gradient method [20], that may still be analyzed within the current framework. Finally, the flexibility of the IOGD algorithm is demonstrated by applying it to the multi-agent multi-target tracking problem from [21] and the online matrix completion problem [22].

The rest of the paper is organized as follows. We begin with a brief review of literature on the areas of target tracking and online optimization. Sec.II formulates the problem at hand and provides a succinct comparison of the existing results and those developed in this paper. The required regularity conditions and assumptions, as well as the proposed IOGD algorithm are stated in Sec.II-B and Sec.III. Regret bounds for the two classes of problems are developed in Sec.IV and Sec.V. In the general case when the path lengths or the cumulative errors are not sublinear, bounds on the steady state tracking error are developed in Sec.VI. The IOGD algorithm is also shown to subsume the online variants of two of the existing gradient algorithms in Sec.VII. Finally, numerical tests on real and synthetic data are provided in Sec.VIII.

A. *Related work*

Inexact gradient methods have been widely used to solve a variety of optimization problems, especially in the context of machine learning [19], [23]–[25]. Since calculating an approximate gradient is often cheap, recent years have witnessed the development of several variants, such as the incremental aggregated gradient method [26], stochastic average gradient method [27], [28], stochastic variance reduced gradient method [29], SAGA [30]. For static optimization problems, the inexact gradient methods are known to converge at a linear rate even for non-

strongly convex objectives [31], [32]. It is remarked that the present work considers an online setting characterized by streaming or sequentially arriving data. Therefore, the performance of the proposed algorithm is measured against that of a slowly moving clairvoyant. In contrast, the inexact gradient methods (IGM) developed in [19], [23], [24], [31] are essentially offline algorithms that are meant to be applied when full data is available. The claims in [19], [23], [24], [31] therefore concern the asymptotic behavior of the iterates $\{\mathbf{x}_k\}$ and are quite different from those in the present paper. For instance, the dynamic regret bounds obtained here do not directly follow from the convergence results in these works.

Time-varying optimization problems have classically been studied in the context of optimal control [33], target tracking [34], [35], non-stationary optimization [36], and parametric programming [37]. First order gradient based methods have been advocated as efficient solvers for such problems [8]–[12], [17], [38]–[41], and will be discussed in detail in Sec. II-C. When the dynamics of the target are partially known and the cost function is sufficiently smooth, it may be possible to use second order methods, such as those proposed in [42]–[46]. On the other hand, tracking is possible even for non-differentiable but strongly convex function using the subgradient and alternating directions method of multiplier methods [47]. It is remarked that an underlying assumption in all these papers is that of strong convexity of the cost function. Such an assumption is quite strong and is not satisfied by several problems, such as least squares and logistic regression [31].

Dynamic regret for analyzing tracking problems was first introduced in [8]. As compared to the weaker notion of static regret, the idea here is to compare the performance of the tracker against that of an adaptive and time-varying adversary [11], [12], [17]. Dynamic regret bounds for the gradient descent and related first order methods have been reported in [8], [9], [11], [12], [17], [39], [41]. An even stronger notion of offline regret has recently been introduced in [48]. The present work builds upon the dynamic regret and steady state tracking results reported in [10], [11], [47]. As compared to existing results, the bounds provided here are not only stronger but also require minimal assumptions on the cost function.

Notations: Scalars are denoted by letters in regular font, while vectors (matrices) are denoted by bold face (capital) letters. The (i, j) -th element of a matrix \mathbf{E} is denoted by $[\mathbf{E}]_{ij}$. The all-one vector of size $n \times 1$ is represented by $\mathbf{1}_n$, while \mathbf{I}_n denotes identity matrix of size $n \times n$. The notation $\|\cdot\|$ represents the Euclidean norm. The Kronecker product operator is denoted by \otimes . The expectation operator is symbolized by \mathbb{E} . Finally, $\sigma_{\max}(\mathbf{E})$ and $\sigma_{\min}(\mathbf{E})$ represent the

TABLE I: Summary of related works on time-varying optimization (cf. Sec.III)

References	Loss function	Inexact	Function	Regret rate
[8]	Convex	No	Deterministic	$\mathcal{O}(\sqrt{K}(1 + W_K))$
[9]	Convex	No	Deterministic	$\mathcal{O}(\sqrt{K}(1 + W_K))$
[17]	Convex	No	Stochastic	$\mathcal{O}(K^{2/3}(1 + V_K)^{1/3})$
[17]	Strongly convex	No	Stochastic	$\mathcal{O}(\sqrt{K}(1 + V_K))$
[39]	Convex	No	Deterministic	$\mathcal{O}(\sqrt{D_K + 1} + \min\{\sqrt{(D_K + 1)W_K}, [(D_K + 1)V_K K]^{1/3}\})$
[11]	Strongly convex	No	Deterministic	$\mathcal{O}(1 + W_K)$
[41]	Convex	No	Stochastic	$\mathcal{O}(\sqrt{KW'_K})$
[12]	Convex+QG	No	Deterministic	$\mathcal{O}(1 + W''_K)$
This work	Convex + QG (P ₁)	Yes	Deterministic/Stochastic (A1)	$\mathcal{O}(1 + E_K + W_K)$
This work	Convex (P ₂)	Yes	Deterministic/Stochastic (A1)	$\mathcal{O}(1 + \sqrt{KE_K} + W_K)$
This work	Convex (P ₂)	Yes	Stochastic (A2)	$\mathcal{O}(1 + E_K + W_K)$

maximum and minimum non-zero singular values of matrix \mathbf{E} , respectively.

II. PROBLEM FORMULATION

This section details the target tracking problem and formulates it within the online learning framework. An example is also provided to motivate the setting.

A. System Model

We consider the general problem of tracking a time-varying parameter that evolves according to an unknown dynamic model. The general setting considered here subsumes the target tracking application of interest, where the parameter may represent the location(s) of the target(s) being pursued by one or more agents. As motivated in [11], [18], [47], the parameter at time k can be written as the solution of the following (discrete) time-varying convex optimization problem

$$\mathbf{x}_k^* \in \arg \min_{\mathbf{x} \in \mathcal{X}} f_k(\mathbf{x}) \quad k = 1, 2, \dots \quad (1)$$

where f_k is a smooth convex function and $\mathcal{X} \subset \mathbb{R}^n$ is a convex set. The set notation in (1) emphasizes the fact that in general, the minimizer of f_k may not necessarily be unique. The parameter estimate at time k is denoted by \mathbf{x}_k . The agents do not know the full functional form of f_k but are only revealed an inexact version of the gradient $\tilde{\nabla} f(\mathbf{x}_k) := \nabla f_k(\mathbf{x}_k) + \mathbf{e}_k$ for some $\mathbf{e}_k \in \mathbb{R}^n$. The agents make use of these inexact gradients to improve their estimates of \mathbf{x}_k^* in an online manner.

This paper considers the problem from an online convex optimization perspective, viewing the agents as learners and targets as adversaries. Specifically, at time k , the online learner selects an action $\mathbf{x}_k \in \mathcal{X}$ and incurs a cost $f_k(\mathbf{x}_k)$, where $f_k : \mathbb{R}^n \rightarrow \mathbb{R}$ are smooth convex functions selected by the adversary. In response to the agent's action, the adversary also reveals an inexact gradient $\tilde{\nabla} f_k(\mathbf{x}_k)$ to the learner. The goal of the learner is to minimize its cumulative loss $\sum_{k=1}^K f_k(\mathbf{x}_k)$ over K time slots. Of particular interest is to characterize the so-called *dynamic regret* of the learner, that measures the cumulative mismatch between the learner's action and the optimal action [9], [11], [17]:

$$\mathbf{Reg}_K := \sum_{k=1}^K (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) \quad (2)$$

where \mathbf{x}_k^* is as defined in (1). In order for the tracking to be successful, the dynamic regret must be upper bounded by a sublinear function of K .

B. Parameter variations and error bounds

It is well known that a sublinear dynamic regret may not always be achievable, e.g., if the parameter variations or the gradient errors are too large [12], [17]. The goal of the current paper will therefore be to bound the dynamic regret using functions of the cumulative parameter variations and errors. For the target tracking setting at hand, it makes sense to characterize the parameter variations using the path length, defined as

$$W_K := \sum_{k=2}^K \|\mathbf{x}_k^* - \mathbf{x}_{k-1}^*\| \quad (3)$$

for some sequence of parameter values $\{\mathbf{x}_k^*\}_{k \geq 1}$. More generally there exist a class of related complexity measures that can be used to characterize the parameter variations [11]. Examples include the functional variation $V_K := \sum_{k=2}^K \sup_{\mathbf{x} \in \mathcal{X}} |f_k(\mathbf{x}) - f_{k-1}(\mathbf{x})|$ and the gradient variation

$$D_K := \sum_{k=2}^K \sup_{\mathbf{x} \in \mathcal{X}} \|\nabla f_k(\mathbf{x}) - \nabla f_{k-1}(\mathbf{x})\|^2.$$

The gradient errors \mathbf{e}_k can be modeled either as being deterministic with bounded norms or stochastic with bounded variance. Deterministic errors may be of interest in adversarial settings while stochastic errors are useful for modeling communication and computational noise. In order to unify the subsequent development, a generic stochastic error model is considered that subsumes that deterministic case. Let \mathcal{F}_k denote the sigma field generated by the random sequence $\{\mathbf{e}_\tau\}_{\tau=1}^{k-1}$. The following assumption bounds the second moment of the error.

A1. Error bound: The stochastic sequence $\{\mathbf{e}_k\}_{k=1}^K$ adheres to the following bound on the second moment:

$$\mathbb{E} [\|\mathbf{e}_k\|^2 \mid \mathcal{F}_k] \leq \varepsilon_k^2 + \nu^2 \|\nabla f_k(\mathbf{x}_k)\|^2 \quad (4)$$

with $\varepsilon_k \leq \varepsilon < \infty$ for all $k \geq 1$, and $\nu \geq 0$ and $\varepsilon \geq 0$ are constants.

When the errors are deterministic, (4) is equivalent to a worst-case bound on the error norm. In the general case, the goal will be to establish the sublinearity of the expected dynamic regret $\mathbb{E} [\mathbf{Reg}_K]$. The specific form of the bounds in (A1) is inspired from [49], [50] and allows for errors that are proportional to the gradient norm in addition to an additive term. It is remarked that from Jensen's inequality, (4) implies that $\mathbb{E} [\|\mathbf{e}_k\| \mid \mathcal{F}_k] \leq \varepsilon_k + \nu \|\nabla f_k(\mathbf{x}_k)\|$. The required dynamic regret bounds will be developed in terms of the path length W_K and the cumulative error bound $E_K := \sum_{k=1}^K \varepsilon_k$.

A particular case of interest is when the gradient errors constitute a white noise process as specified in the following assumption.

A2. White noise: The zero-mean error sequence $\{\mathbf{e}_k\}_{k=1}^K$ is independent identically distributed, i.e.,

$$\mathbb{E} [\mathbf{e}_k \mid \mathcal{F}_k] = \mathbf{0}. \quad (5)$$

Assumption (A2) may be applicable, for instance, when the gradient errors arise from communication errors. The requirement in (5) is more restrictive than that in (4), but also results in improved regret bounds.

The dynamic regret bounds will only be meaningful when the quantities W_K and E_K are sublinear in K . In the context of target tracking, a sublinear W_K is obtained, for instance, if the speed of the target goes to zero as K increases. Such situations are commonly encountered for targets with finite total operational energy. Likewise, E_K may be sublinear in scenarios where the gradient error goes to zero with K . Again, such situations may arise if the noise can be filtered out with increasingly higher accuracy. Finally, for the case when the W_K and E_K are linear functions of K , it makes sense to instead characterize the tracking performance $\|\mathbf{x}_k - \mathbf{x}_k^*\|$ of the proposed algorithm. Different from regret, the goal here is to simply ensure that the asymptotic tracking performance is not too large; see Sec. VI.

C. Related work and problem statement

Before detailing the problem statement, the existing literature on dynamic regret bounds is briefly reviewed. In [8], an online gradient descent algorithm is considered and shown to achieve $\mathcal{O}(\sqrt{K}(1 + W_K))$ regret bound under diminishing step size. Similar regret bounds are shown to exist for the online mirror descent algorithm in [9]. The online gradient descent with stochastic error is considered in [17] and regret bounds for both convex and strongly convex case are provided as $\mathcal{O}(K^{2/3}(1 + V_K)^{1/3})$ and $\mathcal{O}(\sqrt{K}(1 + V_K))$, respectively. It is remarked that these results cannot be practically applied to tracking problems since they assume a priori knowledge of the functional variation V_K . A regret bound in terms of all the three measures W_K , V_K and D_K is obtained in [39] for optimistic mirror descent algorithm. In succession to these works, authors in [11], [12] achieved the $\mathcal{O}(1 + W_K)$ dynamic regret bound for online gradient descent algorithm under constant step size and strongly convex functions. The case of non-strongly convex functions has been recently considered in [12] and a regret bound of $\mathcal{O}(1 + W_K'')$ is obtained for deterministic time-varying problems. The present results generalize those in [12] to stochastic problems with adversarial gradient errors, unbounded domain \mathcal{X} , and a stronger path length definition (3). Finally, the case of noisy gradient is also considered in [41] for general convex functions and dynamic regret bounds are obtained while assuming the knowledge of K . The results in the present work improve upon those in [41], in addition to allowing gradient errors that are not necessarily independent or identically distributed (cf. **(A1)**). Moreover, the impact of gradient errors on the regret is explicitly specified, allowing us to develop IOGD variants that cannot be analyzed as special cases of the noisy OGD proposed in [41]. Table I summarizes the regret bounds obtained in the existing literature. Next, we briefly remark on the path length definitions used in [12], [41].

Remark 1. The path length definition used in (3) applies to an arbitrary sequence of true parameter values $\{\mathbf{x}_k^*\}$ and does not depend on \mathcal{X}_k^* . Consequently, the path length definition in (3) is stronger than those used in [12], [41]. In particular, the definitions in [12], [41] take the following form

$$W_K' := \max_{\{u_k \in \mathcal{X}_k^*\}_{k=2}^K} \sum_{k=2}^K \|\mathbf{u}_k - \mathbf{u}_{k-1}\| \quad (6)$$

$$W_K'' := \sum_{k=2}^K \max_{\mathbf{u} \in \mathcal{X}} \|P_k(\mathbf{u}) - P_{k-1}(\mathbf{u})\|, \quad (7)$$

respectively, where $P_k(\mathbf{u}) := \arg \min_{\mathbf{y} \in \mathcal{X}_k^*} \|\mathbf{y} - \mathbf{u}\|$. Observe that in both cases, the path length could become unbounded if the set \mathcal{X}_k^* is not compact, as is the case in P_1 . For example, the path lengths W'_K and W''_K turn out to be infinite for the least squares example discussed in Sec. VIII-A. The use of an arbitrary trajectory in (3) allows us handle such unbounded sets while also ensuring that $W_K \leq W'_K$ and $W_K \leq W''_K$ for problems where \mathcal{X}_k^* is compact.

The present work considers the online inexact gradient descent method and develops various regret and tracking bounds for the case when the cost function f_k is not strongly convex. The bounds developed here are efficient, and match the existing bounds in the cases when the gradient is exact. Keeping the tracking application in mind, the algorithms developed here will not require any prior knowledge of K . To this end, we consider two broad classes of non-strongly convex functions.

P₁. Special structure: *The function f_k takes the form $f_k(\mathbf{x}) = g_k(\mathbf{E}\mathbf{x})$ where \mathbf{E} is an arbitrary $m \times n$ matrix and the function g_k is strongly convex with parameter $\mu > 0$.*

The functional form required in P_1 is significantly more general than the strong convexity condition for f_k . Since g_k is strongly convex, there exists a unique $\mathbf{u}^* = \arg \min_{\mathbf{y} \in \mathcal{X}} g_k(\mathbf{y})$. Consequently, the set of minimizers of f_k is given by $\mathcal{X}_k^* := \{\mathbf{x}_k^* \mid \mathbf{E}\mathbf{x}_k^* = \mathbf{u}^*\}$. The key property of f_k that will subsequently be utilized is the so-called quadratic growth condition

$$f_k(\mathbf{x}) - f_k^* \geq \frac{\mu \sigma_{\min}(\mathbf{E})}{2} \|\mathbf{x} - P_k(\mathbf{x})\|^2 \quad (8)$$

where $f_k^* := f_k(\mathbf{x}_k^*)$ for $\mathbf{x}_k^* \in \mathcal{X}_k^*$, $P_k(\mathbf{x}) := \arg \min_{\mathbf{y} \in \mathcal{X}_k^*} \|\mathbf{y} - \mathbf{x}\|$ and $\sigma_{\min}(\mathbf{E})$ denotes the smallest non-zero singular value of \mathbf{E} . The quadratic growth condition for convex problems is weaker than the standard assumption of strong convexity [51]. For instance, P_1 is applicable to common machine learning and signal processing problems such as least squares, support vector machines, and logistic regression, none of which are strongly convex [31].

P₂. Bounded domain: *For each $k \geq 1$, f_k is convex with the set of minimizers denoted by \mathcal{X}_k^* and the set \mathcal{X} is compact with diameter $R < \infty$.*

It is remarked that the bounds developed for P_2 are directly proportional to the diameter R , and may therefore be quite loose. Before concluding, a remark on the possible generalization of P_1 is due.

Remark 2. The results developed here can be extended to the generalized version of P_1 where $f_k(\mathbf{x}) = g_k(\mathbf{h}(\mathbf{x}))$ satisfies the QG condition, \mathbf{h} is a smooth function, and g_k is strictly convex. Alternatively, it is also possible to consider convex QG functions f_k that have a unique minimum.

The extensions can be incorporated via minor modifications in the proofs, but are not pursued here in order to keep the exposition succinct.

III. PROPOSED ALGORITHM AND ASSUMPTIONS

The online gradient descent algorithm has been widely used to solve online learning problems owing to its flexibility and simplicity [8], [9], [11], [17], [39]. This work considers the inexact online gradient descent (IOGD) method that takes the form:

$$\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}[\mathbf{x}_k - \alpha(\nabla f_k(\mathbf{x}_k) + \mathbf{e}_k)] \quad (9)$$

where $\mathcal{P}_{\mathcal{X}}(\cdot)$ denotes the projection onto the set \mathcal{X} and is defined as $\mathcal{P}_{\mathcal{X}}(\mathbf{u}) := \arg \min_{\mathbf{y} \in \mathcal{X}} \|\mathbf{y} - \mathbf{u}\|$. The IOGD method has also been applied to static and online problems [31]. The IOGD method is also closely related to the incremental and variance reduced variants of the gradient descent algorithm. The full algorithm is summarized in Algorithm 1.

Algorithm 1 IOGD: Inexact Online Gradient Descent

- 1: **Initialize** \mathbf{x}_1
 - 2: **for** $k = 1, 2, \dots$ **do**
 - 3: **Perform** action \mathbf{x}_k
 - 4: **Observe** inexact gradient $\nabla f_k(\mathbf{x}_k) + \mathbf{e}_k$ at \mathbf{x}_k
 - 5: **Compute** next action $\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}}[\mathbf{x}_k - \alpha(\nabla f_k(\mathbf{x}_k) + \mathbf{e}_k)]$
 - 6: **end for**
-

In addition to Assumptions (A1) and (A2) stated in Sec. II, the subsequent analysis will also require the following regularity conditions.

A3. Lipschitz continuity: The function ∇f_k is Lipschitz continuous with parameter L :

$$\|\nabla f_k(\mathbf{u}) - \nabla f_k(\mathbf{v})\| \leq L \|\mathbf{u} - \mathbf{v}\| \quad (10)$$

for all $k \geq 1$ and $\mathbf{u}, \mathbf{v} \in \mathcal{X}$.

A4. Vanishing gradient: The optimum \mathbf{x}_k^* lies in the relative interior of the set \mathcal{X} , that is, $\nabla f_k(\mathbf{x}_k^*) = 0$ for all $\mathbf{x}_k^* \in \mathcal{X}_k^*$.

A5. Bounded Variation: For a given \mathbf{x}_k^* , there exists some $\sigma > 0$ such that $\|\mathbf{x}_{k+1}^* - \mathbf{x}_k^*\| \leq \sigma$ for all $k \in \mathbb{N}$.

Of these, both (A3) and (A4) are standard and apply to large class of online learning problems. Likewise, the requirement in (A5) imposes a limit on the maximum velocity of the target and is

therefore applicable to most target tracking problems. The bounded variation condition is also satisfied, for instance, if W_K is sublinear or linear and the target motion is not too ‘jumpy’.

The subsequent two sections develop the dynamic regret bounds stated in Table I. As stated earlier, these bounds are meaningful when the variations W_K and error bounds E_K are sublinear in K . Development of the tracking error bounds for linear W_K and E_K is deferred to Sec. VI.

IV. REGRET BOUNDS FOR P_1

This section develops the dynamic regret bounds for P_1 for the case when W_K and E_K are sublinear in K . As discussed earlier, a simple example is that of a target that eventually stops, either upon exhausting its energy or upon reaching its destination. For such targets, there exists $K_0 < \infty$ such that $\mathbf{x}_{k+1}^* = \mathbf{x}_k^*$ for all $k \geq K_0$, making W_K constant for $K \geq K_0$. Likewise, the term ε_k in the gradient error bound may decrease over time resulting in a sublinear E_K . Interestingly however, the gradient error is allowed to be proportional to the gradient norm, and the factor ν need not be diminishing with k .

The section proceeds by developing regret bounds for the general case in **(A1)**, while the special case when the gradient errors constitute a white noise process as in **(A2)** is treated towards the end. The mathematical analysis for these two cases is unified using an indicator variable 1_d , that takes the value 1 when **(A1)** is in effect and zero when both **(A1)** and **(A2)** are in effect (cf. Appendix B). The overall proof involves four key steps, namely, (a) bounding the quantity $\mathbb{E} [\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|]$ in terms of the gradient error and $\text{dist}(\mathbf{x}_k, \mathcal{X}_k^*) := \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|$; (b) bounding the cumulative sum of $\text{dist}(\mathbf{x}_k, \mathcal{X}_k^*)$; (c) establishing a bound on the average gradient norm $\mathbb{E} [\|\nabla f_k(\mathbf{x}_k)\|]$; and finally, (d) using the gradient norm bound to obtain the required bound on $\mathbb{E} [\text{Reg}_K]$. The following intermediate lemma develop the bounds required in the first three steps. The proofs of these results are deferred to Appendix B.

Lemma 1. *Under **(A1)**, and **(A3)**-**(A5)**, the sequence $\{\mathbf{x}_k\}$ for all $k \in \mathbb{N}$ generated by IOGD algorithm satisfies the following bounds for any sequence $\{\mathbf{x}_k^*\}$ with sublinear path length W_K :*

$$\mathbb{E} [\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|] \leq \ell \mathbb{E} [\text{dist}(\mathbf{x}_k, \mathcal{X}_k^*)] + \zeta \varepsilon_k \quad (11)$$

$$\sum_{k=1}^K \mathbb{E} [\text{dist}(\mathbf{x}_k, \mathcal{X}_k^*)] \leq \frac{\|\mathbf{x}_1 - \mathbf{x}_1^*\| + W_K + \zeta E_K}{\chi - \ell} \quad (12)$$

$$\mathbb{E} [\|\nabla f_k(\mathbf{x}_k)\|] \leq G := L \frac{\zeta \varepsilon + \sigma}{\chi} + L \text{dist}(\mathbf{x}_1, \mathcal{X}_1^*) \quad (13)$$

where, ℓ^2 is obtained from (77) using $1_d = 1$ as

$$\ell^2 := 1 - \mu\alpha(1 - \alpha L(1 + \nu)^2) + 2\nu\alpha L$$

and ζ is obtained from (79) using $1_d = 1$ as

$$\zeta := \frac{\alpha(1 + \alpha L)}{\ell}. \quad (14)$$

The step size α is chosen such that

$$\frac{\mu - 2\nu L - \varpi}{2\mu L(1 + \nu)^2} < \alpha < \frac{\min\{\mu - 2\nu L + \varpi, 2\mu\}}{2\mu L(1 + \nu)^2} \quad (15)$$

where $\varpi^2 = (\mu - 2\nu L)^2 - 4\mu L(1 + \nu)^2(1 - \chi^2)$.

It is remarked that an appropriate value of α satisfying the conditions required in Lemma 1 always exists if $\mu - 2\nu L > 2(1 + \nu)\sqrt{\mu L(1 - \chi^2)}$. For example, in the case when \mathbf{E} is an identity matrix, the condition becomes $\mu > 2\nu L$, while in the case when $\nu = 0$, the condition translates to $\mu > 4L(1 - \chi^2)$. Interestingly, the selection of α does not require the prior information about K . As also detailed in [11], [52], target tracking algorithms are generally required to run indefinitely or until the some stopping criteria such as the ‘distance to target is sufficiently small’ is met. In either case, the number of iterations K is generally not known or fixed in advance.

Among the bounds developed in Lemma 1, the gradient bound in (13) is particularly interesting. Different from the vast majority of literature on online convex optimization, the gradient bound allows us to forgo the assumption on the boundedness of the set \mathcal{X} [8], [9], [17]. Further, such a result is useful even when the set \mathcal{X} is compact but has a large diameter R , for the dynamic regret results presented here will not depend on R . Having stated the intermediate lemma, the main result of this section can finally be stated as the following theorem.

Theorem 1. *Under (A1), (A3)-(A5), and α satisfying (15), the IOGD iterates result in the following dynamic regret bound*

$$\mathbb{E}[\mathbf{Reg}_K] \leq \mathcal{O}(1 + E_K + W_K). \quad (16)$$

Proof: The proof follows from the first order convexity condition for f_k and the use of the bounds in (12) and (13). Given a sequence $\{\mathbf{x}_k^*\}$ with sublinear path length W_K , it holds that

$$\begin{aligned} \sum_{k=1}^K (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) &= \sum_{k=1}^K (f_k(\mathbf{x}_k) - f_k(P_k(\mathbf{x}_k))) \\ &\leq \sum_{k=1}^K \langle \nabla f_k(\mathbf{x}_k), \mathbf{x}_k - P_k(\mathbf{x}_k) \rangle \end{aligned} \quad (17)$$

$$\leq \sum_{k=1}^K \|\nabla f_k(\mathbf{x}_k)\| \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| \quad (18)$$

where we have used the first order convexity condition and the Cauchy-Schwarz inequality. Taking expectation and using the bounds in (12) and (13), it follows that

$$\begin{aligned} \mathbb{E}[\mathbf{Reg}_K] &\leq \left(L \frac{\zeta\varepsilon + \sigma}{\chi - \ell} + L \text{dist}(\mathbf{x}_1, \mathcal{X}_1^*) \right) \times \\ &\quad \left(\frac{\|\mathbf{x}_1 - \mathbf{x}_1^*\| + W_K + \zeta E_K}{\chi - \ell} \right) \end{aligned} \quad (19)$$

$$\leq \mathcal{O}(1 + W_K + E_K) \quad (20)$$

which is the required result. \blacksquare

The behavior of regret bound for the IOGD algorithm is governed by the target trajectory $\{\mathbf{x}_k^*\}$ and the error sequence e_k . The results provided here subsume the OGD results in [11] where f_k is strongly convex and the exact gradient is available, i.e., $E_K = 0$. Interestingly, this theorem establishes that an $\mathcal{O}(1 + W_K)$ is still obtainable for P_1 with inexact gradients as long as $E_K \leq \mathcal{O}(W_K)$.

Finally, we provide the dynamic regret bounds for the case when the gradient errors follow a white noise process as in (A2).

Corollary 1. Under (A1)-(A5) and α satisfying (15), the dynamic regret for the IOGD iterates is bounded as follows

$$\begin{aligned} \mathbb{E}[\mathbf{Reg}_K] &\leq \left(L \frac{\alpha\varepsilon + \sigma}{\chi - \ell} \right) \left(\frac{\|\mathbf{x}_1 - \mathbf{x}_1^*\| + W_K + \alpha E_K}{\chi - \ell} \right) \\ &= \mathcal{O}(1 + W_K + E_K) \end{aligned} \quad (21)$$

where ℓ is obtained from (77) with $1_d = 0$ given by $\ell := \sqrt{1 - \mu\alpha(1 - \alpha L(1 + \nu^2))}$ and $\ell < \chi$. The value of α is chosen such that

$$\frac{\mu - \varpi}{2\mu L(1 + \nu^2)} < \alpha < \frac{\min\{\mu + \varpi, 2\mu\}}{2\mu L(1 + \nu^2)}$$

where $\varpi^2 = (\mu)^2 - 4\mu L(1 + \nu^2)(1 - \chi^2)$.

Proof: As shown in Appendix B, the intermediate results for (A1)-(A2) take the form

$$\mathbb{E} [\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|] \leq \ell \mathbb{E} [\text{dist}(\mathbf{x}_k, \mathcal{X}_k^*)] + \alpha \varepsilon_k \quad (22)$$

$$\sum_{k=1}^K \mathbb{E} [\text{dist}(\mathbf{x}_k, \mathcal{X}_k^*)] \leq \frac{\|\mathbf{x}_1 - \mathbf{x}_1^*\| + W_K + \alpha E_K}{\chi - \ell} \quad (23)$$

$$\mathbb{E} [\|\nabla f_k(\mathbf{x}_k)\|] \leq L \frac{\alpha \varepsilon + \sigma}{\chi - \ell r} + L \text{dist}(\mathbf{x}_1, \mathcal{X}_1^*) \quad (24)$$

where ℓ is as defined in Corollary 1. The required dynamic regret bounds can be obtained by proceeding along the lines of proof of Theorem 1. \blacksquare

It is remarked that valid values of α exist if $\mu > 4L(1 - \chi^2)(1 + \nu^2)$. It can be observed that the additional assumption in (A2) does not provide any improvement in the asymptotic regret rate, but only in the associated constants.

V. REGRET BOUND FOR P_2

Having established the regret rates for P_1 with a special QG structure, we now move to the more general problem in P_2 . However, for this case, it would not be possible to develop the results for generic stochastic errors following (A1). Instead, the results will be presented here for two specific scenarios, namely (a) gradient errors following (A1) with $\nu = 0$, and (b) gradient errors following (A1)-(A2) but possibly non-zero value of ν . We begin with stating the following intermediate lemma whose proof is deferred to Appendix C.

Lemma 2. *Under (A1) with $\nu = 0$, (A3)-(A4), and for a sequence $\{\mathbf{x}_k^*\}$ with sublinear path length W_K , it holds for P_2 that the IOGD iterates satisfy*

$$\mathbb{E} [\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|] \leq \mathbb{E} [\|\mathbf{x}_k - \mathbf{x}_k^*\|] + \frac{\xi}{R} (\mathbb{E} [f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)]) + s_k$$

where $\xi := 2\alpha(1 - 2\alpha L)$ and $s_k := \sqrt{2\alpha^2 \varepsilon_k^2 + 2\alpha \varepsilon_k R}$ (cf. (103) and (104)) under $1_d = 1$. The step size α is chosen as $0 < \alpha < 1/2L$.

Lemma 2 leads directly to the required dynamic regret bounds for P_2 under (A1) with $\nu = 0$.

Theorem 2. *Under (A1) with $\nu = 0$, (A3)-(A4), and for a sequence $\{\mathbf{x}_k^*\}$ with sublinear path length W_K , it holds for P_2 that the IOGD iterates adhere to the following dynamic regret rate*

$$\mathbb{E} [\mathbf{Reg}_K] \leq \mathcal{O}(1 + \sqrt{KE_K} + W_K). \quad (25)$$

with step size selected as $0 < \alpha < 1/2L$.

Proof: Using triangle inequality and result of lemma 2 for $k \geq 1$, we have that

$$\begin{aligned} \mathbb{E} [\|\mathbf{x}_{k+1} - \mathbf{x}_{k+1}^*\|] &\leq \mathbb{E} [\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|] + \|\mathbf{x}_{k+1}^* - \mathbf{x}_k^*\| \\ &\leq \mathbb{E} [\|\mathbf{x}_k - \mathbf{x}_k^*\|] - \frac{\xi}{R} \mathbb{E} [f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)] \\ &\quad + s_k + \|\mathbf{x}_{k+1}^* - \mathbf{x}_k^*\|. \end{aligned} \quad (26)$$

It is now possible to apply the relationship in (26) recursively for $k = 1, \dots, K$, so as to obtain

$$\begin{aligned} \mathbb{E} [\|\mathbf{x}_{K+1} - \mathbf{x}_{K+1}^*\|] &\leq \|\mathbf{x}_1 - \mathbf{x}_1^*\| - \frac{\xi}{R} \sum_{k=1}^K \mathbb{E} [f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)] \\ &\quad + \sum_k s_k + W_K. \end{aligned} \quad (27)$$

Since the left hand side is positive by definition, it follows that

$$\frac{\xi}{R} \mathbb{E} \left[\sum_{k=1}^K (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) \right] \leq \|\mathbf{x}_1 - \mathbf{x}_1^*\| + S_K + W_K \quad (28)$$

where we have that

$$S_K = \sum_{k=1}^K \sqrt{2\alpha^2 \varepsilon_k^2 + 2\alpha \varepsilon_k R} \quad (29)$$

$$\leq \sum_{k=1}^K \sqrt{2\alpha R \varepsilon_k} + \sqrt{2\alpha} \varepsilon_k \quad (30)$$

$$\leq \sqrt{2\alpha R} \left(K \sum_{k=1}^K \varepsilon_k \right)^{1/2} + \sqrt{2\alpha} E_K \quad (31)$$

$$= \sqrt{2\alpha R K E_K} + \sqrt{2\alpha} E_K. \quad (32)$$

Substituting the result in (32) into (28) yields the required result

$$\begin{aligned} \mathbb{E} [\mathbf{Reg}_K] &\leq \frac{R}{\xi} \left(\|\mathbf{x}_1 - \mathbf{x}_1^*\| + \sqrt{2\alpha R K E_K} + \sqrt{2\alpha} E_K + W_K \right) \\ &\leq \mathcal{O}(1 + \sqrt{K E_K} + W_K) \end{aligned} \quad (33)$$

where the inequality in (33) follows since $\sqrt{K E_K} > E_K$ whenever E_K is sublinear. \blacksquare

The results can be improved for the case when the gradient errors follow a white noise process. Specifically, the following corollary holds as shown in Appendix C.

Corollary 2. Under (A1)-(A4), and for a sequence $\{\mathbf{x}_k^*\}$ with sublinear path length W_K , it holds for P_2 that the IOGD iterates satisfy

$$\mathbb{E} [\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|] \leq \mathbb{E} [\|\mathbf{x}_k - \mathbf{x}_k^*\|] + \frac{\xi}{R} (\mathbb{E} [f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)]) + \alpha \varepsilon_k \quad (34)$$

where $\xi := 2\alpha(1 - \alpha L(1 + \nu^2))$ since $1_d = 0$ (cf. (103)). For this case, the dynamic regret is bounded as

$$\mathbb{E} [\mathbf{Reg}_K] \leq \mathcal{O}(1 + E_K + W_K). \quad (35)$$

for $0 < \alpha < \frac{1}{L(1+\nu^2)}$.

Proof: The proof of the bound in (34) is provided in Appendix C. The required regret bound follows along the lines of the proof of Theorem 2. ■

VI. TRACKING PERFORMANCE

Departing from the regret analysis pursued in Sections IV and V, we now consider the asymptotic tracking performance for the scenarios when W_K or E_K are not sublinear in K . As a practical example, consider a target that continues to move with a constant velocity in an adversarial manner. Likewise, we may consider the scenario when errors are unpredictably random with constant variance. More generally, this section will only assume that the variations and the gradient errors terms are bounded as follows

$$\|\mathbf{x}_{k+1}^* - \mathbf{x}_k^*\| \leq \sigma \quad \text{and} \quad \varepsilon_k \leq \varepsilon \quad (36)$$

where realistically, the terms σ and ε should not be too large. In both cases, it may not be possible for the agents to reduce their distance from the target beyond a certain value [47], [52].

The goal of this section will be to characterize the tracking performance $\mathbb{E} [\|\mathbf{x}_k - \mathbf{x}_k^*\|]$. Towards this end, only the problem P_1 will be considered. As before, we begin with analyzing the general case when (A1) is in effect, and specialize the results later to the case when both (A1)-(A2) are in effect. The main result is provided in the subsequent lemma, whose proof is provided in Appendix B.

Lemma 3. Under (A1) and (A3)-(A5), and sequences $\{\mathbf{x}_k^*, \mathbf{e}_k\}$ satisfying (36), the IOGD iterates for \mathcal{P}_1 adhere to the following bound

$$\begin{aligned} \mathbb{E} [\text{dist}(\mathbf{x}_{k+1}, \mathcal{X}_{k+1}^*)] &\leq (\ell/\chi)^k \text{dist}(\mathbf{x}_1, \mathcal{X}_1^*) \\ &\quad + \left[\frac{1 - (\ell/\chi)^k}{1 - (\ell/\chi)} \right] \frac{\sigma + \zeta\epsilon}{\chi}. \end{aligned} \quad (37)$$

for all $k \geq 1$. Here, the constants χ , ℓ , and ζ are as defined in Lemma 1 since $1_d = 1$ for this case and α is chosen as in (15).

Lemma 3 establishes that the tracking error decreases exponentially with k , and is ultimately bounded by a steady state value of $\frac{\sigma + \zeta\epsilon}{\chi - \ell}$. Alternatively, if it holds that $\Delta_1 := \text{dist}(\mathbf{x}_1, \mathcal{X}_1^*) > \epsilon$, then the number of iterations required for the tracking error is to satisfy

$$\mathbb{E} [\text{dist}(\mathbf{x}_{k+1}, \mathcal{X}_{k+1}^*)] \leq \epsilon + \frac{\sigma + \zeta\epsilon}{\chi}, \quad (38)$$

is $\mathcal{O}(\log(\lceil 1/\epsilon \rceil))$.

For the case when errors constitute a white noise process, the following corollary provides the required result.

Corollary 3. Under (A1)-(A5), and sequences $\{\mathbf{x}_k^*, \mathbf{e}_k\}$ satisfying (36), the IOGD iterates for \mathcal{P}_1 adhere to the following bound

$$\begin{aligned} \mathbb{E} [\text{dist}(\mathbf{x}_{k+1}, \mathcal{X}_{k+1}^*)] &\leq (\ell/\chi)^k \text{dist}(\mathbf{x}_1, \mathcal{X}_1^*) \\ &\quad + \left[\frac{1 - (\ell/\chi)^k}{1 - (\ell/\chi)} \right] \frac{\sigma + \alpha\epsilon}{\chi}. \end{aligned} \quad (39)$$

for all $k \geq 1$ and ℓ is as defined in Corollary 1 and α should be chosen in same manner as in Corollary 1.

The proof of Corollary 3 follows from the analysis provided in Appendix B. Likewise, the iteration complexity of the tracking error for this case is also $\mathcal{O}(\log(\lceil 1/\epsilon \rceil))$.

VII. IMPLICATIONS OF PROPOSED RESULTS

This section provides two online algorithms that can be interpreted as variants of the proposed IOGD algorithm.

1) *Incremental OGD with increasing sample size*: Consider a scenario where the target trajectory can be expressed as the solution to the following composite function optimization problem

$$\mathbf{x}_k^* = \arg \min_{\mathbf{x} \in \mathcal{X}} f_k(\mathbf{x}_k) := \frac{1}{N} \sum_{i=1}^N f_k^i(\mathbf{x}) \quad k = 1, 2, \dots \quad (40)$$

where the objective function adheres to either P_1 or P_2 . Inspired by the sampling-based gradient methods [19], consider the following online algorithm:

$$\mathbf{x}_{k+1} = \mathcal{P}_{\mathcal{X}} \left(\mathbf{x}_k - \frac{\alpha}{N_k} \sum_{i \in \mathcal{N}_k} \nabla f_k^i(\mathbf{x}_k) \right) \quad (41)$$

where $N_k = |\mathcal{N}_k|$ and \mathcal{N}_k is a random subset of $\{1, \dots, N\}$. In order to view (41) as an IOGD variant, observe that the gradient error is given by $\mathbf{e}_k := \frac{1}{N_k} \sum_{i \in \mathcal{N}_k} \nabla f_k^i(\mathbf{x}_k) - \nabla f_k(\mathbf{x}_k)$. If the subsets \mathcal{N}_k are formed by sampling the functions f_k^i uniformly without replacement and in an i.i.d. fashion, it holds that [Sec. 2.8] [53],

$$\mathbb{E} [\|\mathbf{e}_k\|^2 \mid \mathcal{F}_k] = \frac{N - N_k}{NN_k} \Lambda^2 \quad (42)$$

where Λ^2 is a bound on the sample variance of the gradients $\{\nabla f_k^i(\mathbf{x}_k)\}_{i=1}^N$, i.e.,

$$\frac{1}{N-1} \sum_{i=1}^N \|\nabla f_k^i(\mathbf{x}) - \nabla f_k(\mathbf{x})\| \leq \Lambda^2 \quad \mathbf{x} \in \mathcal{X} \quad (43)$$

Defining $\varepsilon_k := \sqrt{(1/N_k - 1/N)}$, it can be seen that E_K is sublinear, for instance when $N_k = Nk^\rho / (N + k^\rho)$ with $\rho > 0$.

2) *Proximal OGD methods for composite minimization*: Next, consider a scenario where the target trajectory is given by the solution of the following composite function minimization problem:

$$\mathbf{x}_t^* = \arg \min_{\mathbf{x}} f_k(\mathbf{x}) := g_k(\mathbf{x}) + h_k(\mathbf{x}) \quad (44)$$

where h_k is a differentiable regularization function. We propose the proximal OGD algorithm for solving (44) that takes the form

$$\mathbf{x}_{k+1} = \text{prox}_{\alpha h_k}(\mathbf{x}_k - \alpha \nabla g_k(\mathbf{x}_k)) \quad (45)$$

where the proximal function is defined as $\text{prox}_{\alpha h_k}(\mathbf{z}) = \arg \min_{\mathbf{x}} \alpha h_k(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|^2$. Equivalently, it is possible to write the update in (45) as

$$\begin{aligned} \nabla_{\mathbf{x}} \left(\alpha h_k(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{x}_k + \alpha \nabla g_k(\mathbf{x}_k)\|^2 \right) &= 0 \\ \Rightarrow \mathbf{x}_{k+1} &= \mathbf{x}_k - \alpha \nabla g_k(\mathbf{x}_k) - \alpha \nabla h_k(\mathbf{x}_{k+1}) \end{aligned} \quad (46)$$

Recall that the IOGD algorithm for (44) takes the form $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha(\nabla g_k(\mathbf{x}) + \nabla f(\mathbf{x}) + \mathbf{e}_k)$. Therefore, the error term becomes:

$$\begin{aligned} \|\mathbf{e}_k\| &= \|\nabla h_k(\mathbf{x}_k) - \nabla h_k(\mathbf{x}_{k+1})\| \leq L \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \\ &\leq \alpha L \|\nabla g_k(\mathbf{x}_k) + \nabla h_k(\mathbf{x}) + \mathbf{e}_k\| \\ &\leq \frac{\alpha L}{1 - \alpha L} \|\nabla f_k(\mathbf{x}_k)\| \end{aligned} \quad (47)$$

where L is the Lipschitz constant of ∇h_k . Consequently, as long as $\nu = \frac{\alpha L}{1 - \alpha L}$ is sufficiently small and subject to the assumptions (A1)-(A5), the proximal OGD algorithm adheres to the $\mathcal{O}(1 + W_K)$ bound developed earlier. In other words, the inexact gradient used in (46) does not have any effect on the dynamic regret rate.

VIII. NUMERICAL TESTS

This section provides detailed numerical tests that demonstrate the usefulness and applicability of the proposed IOGD framework to three different applications.

A. Time varying parameter estimation

Consider a network of N nodes placed over a planar region $\mathcal{A} \subseteq \mathbb{R}^2$. The nodes seek to cooperatively estimate (track) a time-varying parameter $\mathbf{x}_k^* \in \mathbb{R}^n$ using observations $\mathbf{b}_{i,k} \in \mathbb{R}^m$ made at each node i and time k . The measurements follow the linear model $\mathbf{b}_{i,k} = \mathbf{E}_i \mathbf{x}_k^* + \mathbf{n}_{i,k}$ where \mathbf{E}_i is a node-dependent measurement matrix and the noise $\mathbf{n}_{i,k} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ is independent identically distributed across nodes and time. At time k , the least squares estimate for \mathbf{x}_k is given by

$$\mathbf{x}_k^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{N} \sum_{i=1}^N \|\mathbf{E}_i \mathbf{x}_k - \mathbf{b}_{i,k}\|_2^2. \quad (48)$$

Here, the measurement matrices $\{\mathbf{E}_i\}_{i=1}^N$ may not necessarily be full rank and therefore the optimum \mathbf{x}_k^* may not be unique. In this case, the path length in (3) is well-defined for a given trajectory $\{\mathbf{x}_k^*\}$, while the path lengths in (6) and (7) turn out to be infinite.

Towards estimating \mathbf{x}_k in an efficient manner, we make use of the incremental OGD algorithm proposed in (41). Letting \mathcal{N}_k be a random subset of $\{1, \dots, N\}$, the incremental OGD updates take the form:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\alpha}{N_k} \sum_{i \in \mathcal{N}_k} \nabla f_k^i(\mathbf{x}_k) \quad (49)$$

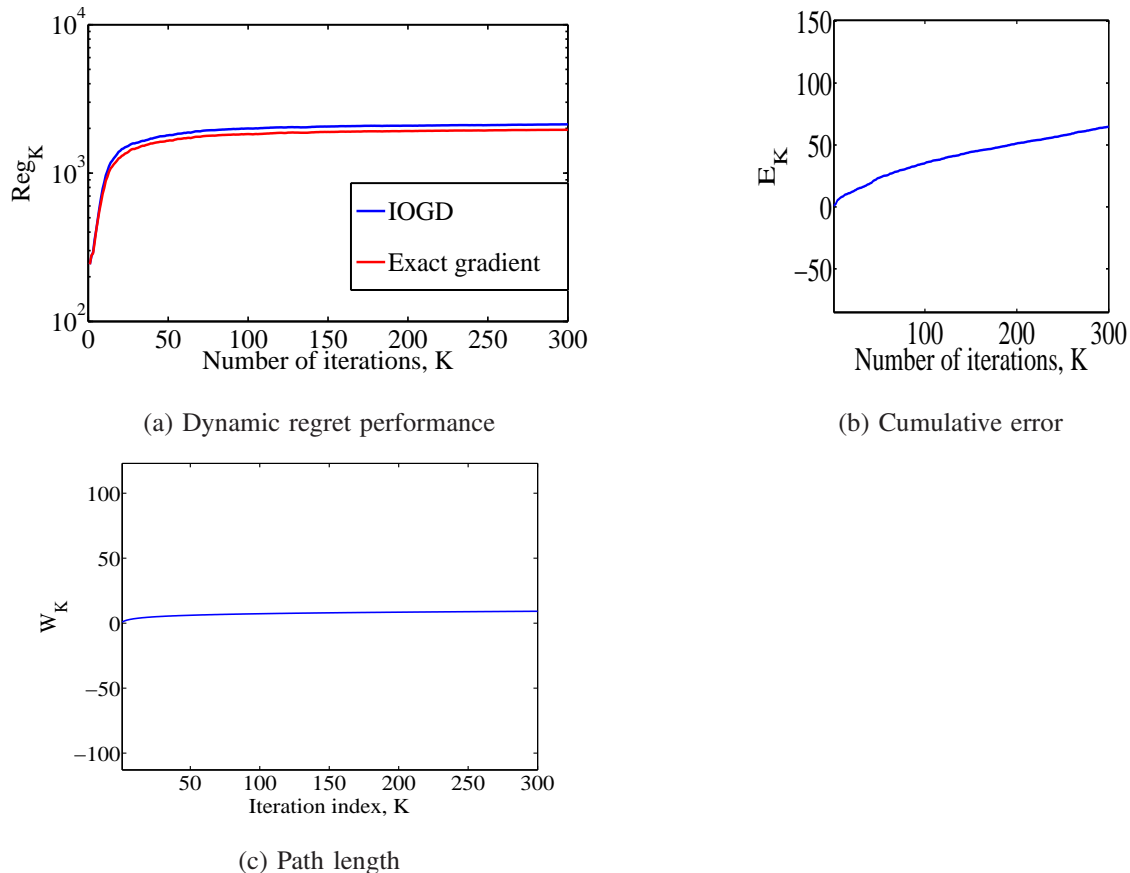


Fig. 1: Time varying parameter estimation for $m = 2$ and $n = 3$, $N = 100$. Fig. 1(a) plots the dynamic regret, Fig. 1(b) plots the cumulative error E_K , and Fig. 1(c) shows the sublinear behavior of path length W_K as a function of the total number of iterations K .

where $f_k^i(\mathbf{x}_k) := \|\mathbf{E}_i \mathbf{x}_k - \mathbf{b}_{i,k}\|_2^2$ and $N_k := |\mathcal{N}_k|$. As detailed in Sec. VII, the incremental OGD is a special case of the proposed IOGD algorithm if N_k increases to N as k goes from 1 to K . Observe that the gradient error for this case is not independent or identically distributed. Therefore, for this case, assumptions **(A1)** and **(A3)-(A5)** are satisfied and the results of Sec. IV are applicable.

In order to validate the performance of the proposed algorithm, we consider a network with $N = 100$ nodes and for $m = 2$ and $n = 3$. The subset \mathcal{N}_k is selected uniformly without replacement from $\{1, \dots, N\}$ and following the rule $N_k = Nk/(N + k)$ as stated in Sec. VII. The step size is set to $\alpha = 0.1$ for all iterations and irrespective of K . The algorithm is run for different values of K and the resulting dynamic regret, error bound, and the path length are shown in Figs. 1(a), 1(b), and 1(c). For the purposes of comparison, the regret obtained when

full data is used (i.e. $N_k = N$ for all $k \geq 1$) is also shown (exact gradient). It can be seen that although proposed incremental IOGD does not make use of the full data, its regret is not too far from that of the classical OGD algorithm. Further, as expected from the analytical results, the dynamic regret is clearly sublinear.

B. Multi-target tracking

This section develops a low-complexity online multi-target tracking algorithm inspired from the convex optimization-based target tracking framework developed in [4]. Specifically, a team of n agents at locations $\{\mathbf{x}_k^i\}_{i=1}^n$ is tasked with tracking a set of m targets at locations $\{\mathbf{y}_k^j\}_{j=1}^m$. The discretized problem is formulated as the following convex optimization that must be solved for each $k \geq 1$ [4, Thm. 3.8.1]:

$$\{\mathbf{x}_{k+1}^i\} = \arg \min_{\{\mathbf{x}^i \in \mathbb{R}^p\}} \sum_{i=1}^n \psi_k^i(\mathbf{x}^i) \quad (50a)$$

$$\text{s.t. } \|\mathbf{x}^i - \mathbf{x}_k^i\|^2 \leq v \quad ; \quad i = 1, \dots, n \quad (50b)$$

$$\sum_{i=1}^n w_k^{ij} \|\mathbf{x}^i - \mathbf{y}_k^j\|^2 \leq \eta \quad ; \quad j = 1, \dots, m \quad (50c)$$

where $\psi_k^i(\cdot)$ is a time-varying cost function and v is the square of the maximum distance that an agent may cover within a single time slot. For this paper, the following cost function is used:

$$\psi_k^i(\{\mathbf{x}^i\}) = \frac{1}{2} \sum_{\ell \neq i} \|\mathbf{x}^i - \mathbf{x}^\ell\|^2 + \gamma \|\mathbf{x}^i - \mathbf{x}_k^i\|^2 \quad (51)$$

where $\gamma > 0$ is a regularization parameter. The objective function encourages agent i to remain close to the other agents. At the same time, the regularization term forces the agents to not move around unnecessarily. The constraint in (50c) is the linearized version of the original constraint obtained from using the process described in [4, Chap. 3]. A sigmoidal weight function is utilized that takes the form:

$$w_k^{ij} = \left(1 + e^{-\vartheta(\epsilon - \|\mathbf{x}_k^i - \mathbf{y}_k^j\|)}\right)^{-1} \quad (52)$$

where ϑ and ϵ are positive parameters. Observe that the weights are small for agent-target pairs that are far from each other. In other words, the constraint in (50c) encourages the set of agents tracking a target j to stay close to it. The weights are normalized so that $\sum_{i=1}^n w_k^{ij} = 1$ in order to ensure that each target is tracked by at least one agent. Finally, it is remarked that the agents

may only know the estimated target locations $\{\hat{\mathbf{y}}_k^j\}_{j=1}^m$ instead of the true locations required for (50c).

Different from [21], where interior point methods are proposed towards solving (50) at each time instant, the goal here is to track the targets at a significantly low complexity. To this end, we propose an IOGD-based low-complexity algorithm capable of tracking high speed targets in large multi-agent networks. Since the constrained optimization problem in (50) is not of the form required in (1), the IOGD algorithm will instead be applied in the dual domain. It can be verified that (50) satisfies the Slater's conditions, and therefore has zero duality gap.

To this end, associate dual variables $\{\lambda^i\}_{i=1}^n$ and $\{\nu^j\}_{j=1}^m$ with (50b) and (50c), respectively. Collecting the primal and dual variables $\{\lambda^i\}$, $\{\nu^j\}$, and $\{\mathbf{x}^i\}$ into vectors $\boldsymbol{\lambda}$, $\boldsymbol{\nu}$, and \mathbf{x} respectively, the Lagrangian can be written as

$$L_k(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \sum_{i=1}^n \sum_{\ell=i+1}^n \|\mathbf{x}^i - \mathbf{x}^\ell\|^2 + \gamma \|\mathbf{x} - \mathbf{x}_k\|^2 \quad (53)$$

$$+ \sum_{i=1}^n \lambda^i (\|\mathbf{x}^i - \mathbf{x}_k^i\|^2 - v) + \sum_{j=1}^m \nu^j \left[\sum_{i=1}^n w_k^{ij} \|\mathbf{x}^i - \mathbf{y}_k^j\|^2 - \eta \right].$$

Thus the dual function can be written as

$$\varrho_k(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \arg \min_{\mathbf{x}} L_k(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \quad (54)$$

Since the dual function is always concave, the proposed IOGD algorithm can be utilized to maximize ϱ_k in an online fashion. In particular, for a given $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$, the gradient of the dual function is given by

$$\frac{\partial L_k(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})}{\partial \lambda^i} = \|\mathbf{x}^i(\boldsymbol{\lambda}, \boldsymbol{\nu}) - \mathbf{x}_k^i\|^2 - v \quad (55)$$

$$\frac{\partial L_k(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})}{\partial \nu^j} = \sum_{i=1}^n w_k^{ij} \|\mathbf{x}^i(\boldsymbol{\lambda}, \boldsymbol{\nu}) - \mathbf{y}_k^j\|^2 - \eta \quad (56)$$

where $\{\mathbf{x}^i(\boldsymbol{\lambda}, \boldsymbol{\nu})\}_{i=1}^n = \arg \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$. In order to evaluate the minimization, observe that the condition $\nabla_{\mathbf{x}^i} L_k(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = 0$ is equivalent to

$$(n + 2\gamma + 2\lambda^i + 2 \sum_{j=1}^m \nu^j w_k^{ij}) \mathbf{x}^i - \sum_{\ell=1}^n \mathbf{x}^\ell$$

$$= 2(\gamma + \lambda^i) \mathbf{x}_k^i + 2 \sum_{j=1}^m \nu^j w_k^{ij} \mathbf{y}_k^j \quad (57)$$

for $i = 1, \dots, n$. Let \mathbf{D}_k be a diagonal matrix with (i, i) -th element $[\mathbf{D}_k]_{ii} = n + 2\gamma + 2\lambda^i + 2 \sum_{j=1}^m \nu^j w_k^{ij}$ and let the right-hand side of (57) be denoted by the i -th element of the vector

φ_k . Then the solution to (57) can be written as $\mathbf{x} = (\mathbf{D}_k - \mathbf{1}\mathbf{1}^T)^{-1}\varphi_k$ where $\mathbf{1}$ is an all-one vector of appropriate dimension. Observe further that the solution is unique for all λ and ν since $L_k(\mathbf{x}, \lambda, \nu)$ is strongly convex in \mathbf{x} . Using the matrix inversion lemma, it is possible to calculate \mathbf{x} with $\mathcal{O}(n)$ complexity as follows:

$$\mathbf{x}^i = [\mathbf{D}_k]_{ii}^{-1}[\varphi_k]_i + \frac{[\mathbf{D}_k]_{ii}^{-1}}{1 - \sum_{\ell=1}^n [\mathbf{D}_k]_{\ell\ell}^{-1}} \left(\sum_{\ell=1}^n [\mathbf{D}_k]_{\ell\ell}^{-1}[\varphi_k]_{\ell} \right) \quad (58)$$

Since the target location is not known exactly, the estimated target location must be used in (56), making the gradient inexact. The agent location is determined from the current primal iterate \mathbf{x}_k which serves as a proxy for the primal solution \mathbf{x}_k^* at time k . Consequently, the results of Sec. VI will be applicable here. The full IOGD-based multi-target tracking algorithm is summarized in Algorithm 2.

Algorithm 2 IOGD-based multi-target tracking

1: **Initialize** \mathbf{x}_1 , λ_1 , and ν_1 , and step sizes α_λ , and α_ν

Repeat for $k = 1, 2, \dots$,

2: Compute weights $\{w_k^{ij}\}$ from (52)

3: **Calculate** the next location as

$$\mathbf{x}_{k+1} = \arg \min_{\mathbf{x}} L_k(\mathbf{x}, \lambda_k, \nu_k)$$

4: Update for all agents and targets:

$$\lambda_{k+1}^i = \left[\lambda_k^i - \alpha_\lambda (\|\mathbf{x}_{k+1}^i - \mathbf{x}_k^i\|^2 - v) \right]_+ \quad (59)$$

$$\nu_{k+1}^j = \left[\nu_k^j - \alpha_\nu \left(\sum_{i=1}^n w_k^{ij} \|\mathbf{x}_{k+1}^i - \hat{\mathbf{y}}_k^j\|^2 - \eta \right) \right]_+ \quad (60)$$

The performance of the proposed multi-target tracking algorithm is studied on a number of simulated planar environments. The agent velocities are restricted to 0.2 m/s and a target is assumed covered if it is within $\eta = 0.54$ m from at least one agent. Consider first a simple scenario consisting of three targets ($m = 3$) and three agents $n = 3$. The targets are co-located at time $k = 1$ and start moving away from each other along the paths shown in Fig. 2. It can be seen that the proposed algorithm works as expected, and the agent team splits up in order to track the three targets. On the other hand, the algorithm in [21] does not necessarily exhibit such a behavior and requires careful parameter tuning so as to allow tracking with reasonable accuracy;

see Fig. 2. Indeed, since [21] entailed solving a constrained convex optimization problem at every time instant, it was observed that unless the parameters are not selected carefully, the problem could become infeasible. It was however possible to circumvent this behavior to a certain extent by explicitly adding noise to the output of the optimization problem. In contrast, no such issue was present in the proposed IOGD algorithm, whose performance was quite robust to the choice of parameters. It is observed from the figure that the trajectory of the solution of (50) using [21] is a bit away from the target trajectory because the formulation in (50) does not incentivize the agents to come too close to target. On the other hand, the IOGD algorithm over-tracks to a certain extent, thereby coming close to the target but at the same time yielding a higher objective value.

Fig. 3 shows the error performance $\|\mathbf{x}_k - \mathbf{x}_k^*\|$ for the IOGD algorithm. As expected from the results of Sec. VI, the tracking error remains bounded. It is remarked that for tracking error is small initially while the agents are close to each other and increases when they split up to track the diverging targets.

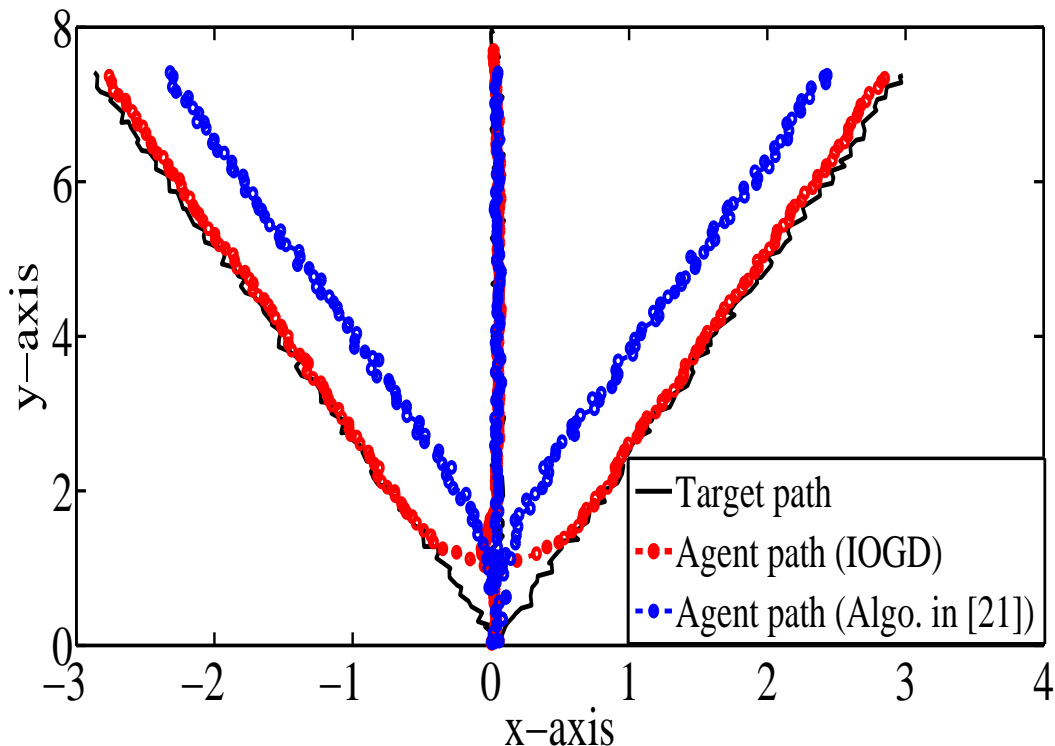


Fig. 2: Tracking performance comparison for $m = 3$ and $n = 3$.

Next, we consider a large scale system with $m = 10$ targets and $n = 50$ agents. As expected,

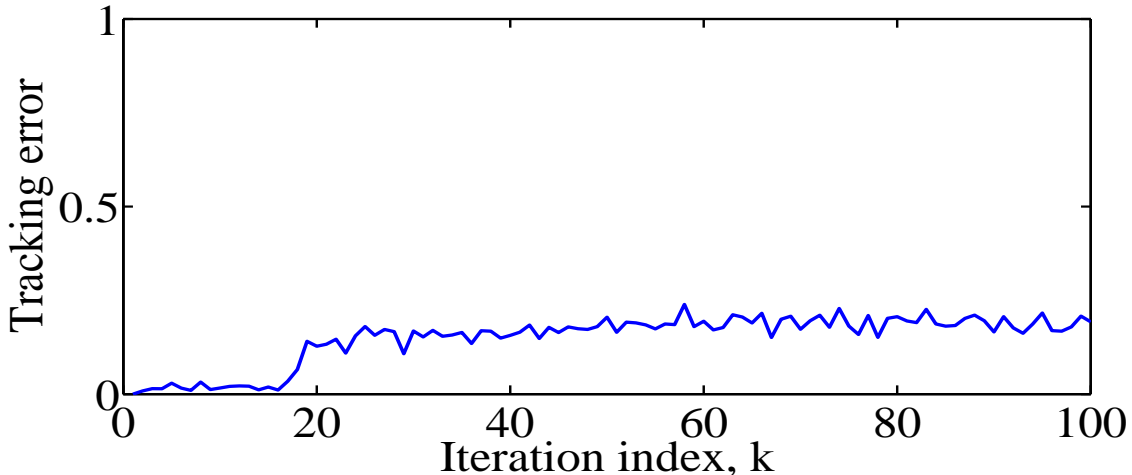


Fig. 3: Tracking error performance of IOGD algorithm.

the IOGD algorithm is capable of tracking most of the targets at low complexity; see Figs. 4 and 5. As with the smaller system considered earlier, the splitting of the agent teams is observed in Fig. 4d. The splitting behavior is also evident from the supplementary video included with this paper¹. It is important to emphasize that the tracking performance of the IOGD is at par with the convex optimization approach of [21]. In contrast, solving a general convex optimization problem as required in [21] incurs a complexity of at least $\mathcal{O}(n^3)$ as opposed to the $\mathcal{O}(n)$ complexity incurred in the calculation of the inexact gradient in (59)-(60). For the sake of comparison, both algorithms were implemented in MATLAB and their run-times measured on an Intel Xeon E3-1226 3.30GHz CPU machine. The resulting per-iteration run-time for the proposed algorithm was 49 ms, as compared to that of 974 ms required by [21].

C. Tracking movie ratings

This section considers the problem of tracking movie ratings in recommender systems. Traditionally, the problem has been posed as that of completing a low-rank matrix and solved using a variety of offline or incremental algorithms. In many applications however, the dataset grows over time, necessitating a real-time recommendation engine that predicts the missing ratings in an online manner; see e.g. [22]. This section advocates an IOGD-inspired algorithm that can predict and track missing ratings at low complexity.

¹<https://www.youtube.com/watch?v=bVto6LItehM>

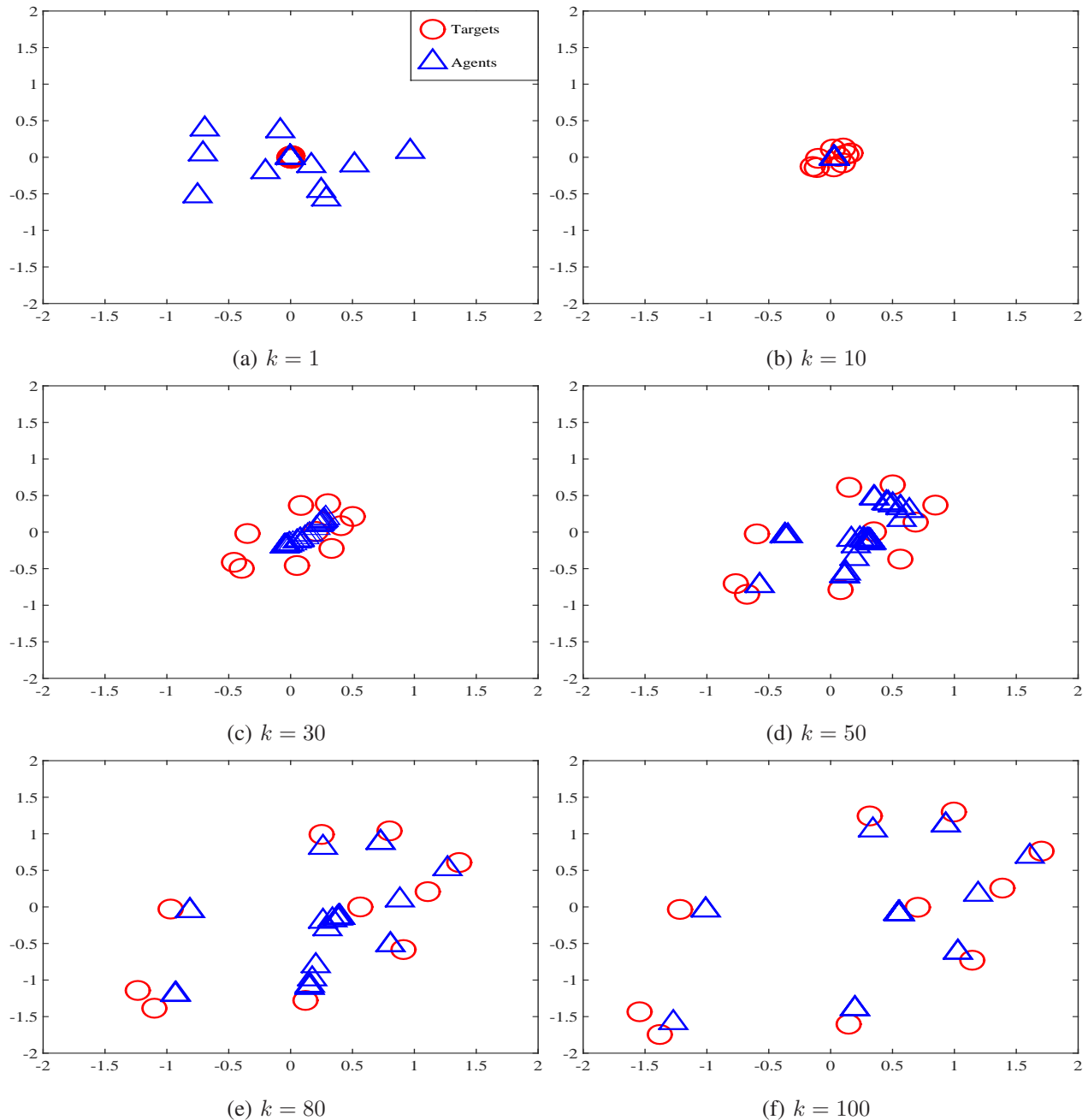


Fig. 4: Target tracking for $m = 10$ and $n = 50$.

Let $\mathbf{M}_k \in \mathbb{R}^{K \times L}$ be the measurement matrix containing user ratings at locations specified in the binary matrix \mathbf{J}_k , i.e., $[\mathbf{J}_k]_{ij} = 1$ if $[\mathbf{M}_k]_{ij}$ contains a rating and zero otherwise. The goal is to generate a complete ratings matrix \mathbf{X}_k , which is the solution to the following time-varying

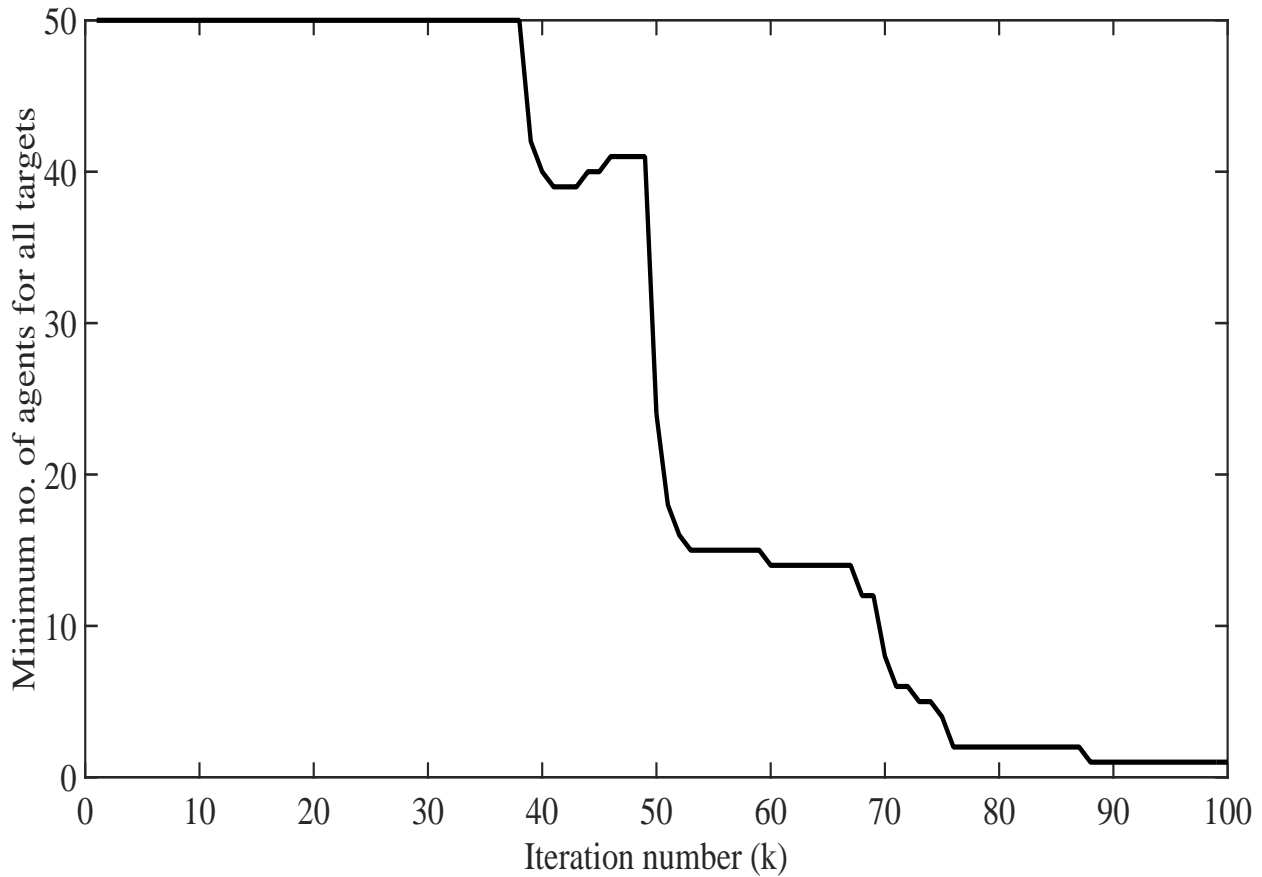


Fig. 5: Number of agents covering each target ($m = 10$, $n = 50$).

optimization problem:

$$\mathbf{X}_k^* = \min_{\mathbf{X} \in \mathcal{X}} \frac{1}{2} \|\mathbf{J}_k \odot (\mathbf{M}_k - \mathbf{X})\|_F^2 + \lambda \|\mathbf{X}\|_* . \quad (61)$$

Following the notation of [54, Sec. III-D], \odot denotes the Hadamard product while $\|\mathbf{X}\|_*$ denotes the nuclear norm of \mathbf{X} . In order to obtain \mathbf{X}_k in an online fashion, we utilize the proximal OGD algorithm introduced in (45), that takes the form:

$$\mathbf{X}_{k+1} = D_\lambda (\mathbf{X}_k + \mu \mathbf{J}_k \odot (\mathbf{M}_k - \mathbf{X})) \quad (62)$$

where, $D_\lambda(\cdot)$ is the singular value thresholding operator. Given the singular value decomposition of a matrix $\mathbf{Y} = \mathbf{U} \text{diag}(\sigma_1, \dots, \sigma_r) \mathbf{V}^T$, the singular value thresholding operation is defined as

$$D_\lambda(\mathbf{Y}) = \mathbf{U} \text{diag}([\sigma_1 - \lambda]_+, \dots, [\sigma_r - \lambda]_+) \mathbf{V}^T . \quad (63)$$

It is remarked that although the proximal OGD in Sec. VII is motivated as an IOGD variant for differentiable regularizers, it is applied and will be shown to work even for the non-differentiable

regularizer $\|\mathbf{X}\|_*$. The other assumptions, namely **(A1)**, **(A4)**, and **(A5)** are however satisfied for this case.

Towards testing the performance of the proposed algorithm, we consider the Movielens 10M dataset, consisting of about 10 million ratings. The time-stamp associated with each rating is utilized to obtain 40 matrices $\{\mathbf{M}_k\}_{k=1}^{40}$, each consisting of new ratings added over the last 30 days. Fig. 6 shows the evolution of the root mean-square error over time for the step size parameter $\alpha = 1$. Interestingly, the regularized mean square error (RMSE) decays quickly over time and attains a value competitive with the benchmark [22]. Indeed, the proximal algorithm in (59) can be viewed as a generalization of the Soft-Impute method for the dynamic matrix completion problem. The tracking MSE for a class of related adaptive matrix completion algorithms is provided in [54].

IX. CONCLUSION AND FUTURE WORK

This paper considered the problem of tracking time-varying and possibly adversarial targets. An inexact gradient descent method is proposed, that is applicable to non-strongly convex objective functions. The performance of the proposed framework is analyzed by developing bounds on its dynamic regret in terms of the target path length and the cumulative gradient error. Two distinct cases are considered: (a) case P_1 , where the cost function adheres to the quadratic growth condition; and (b) case P_2 where the cost function is convex but the optimization variable belongs to a compact domain. For both cases, it is established that the dynamic regret is sublinear if the target path length and the cumulative gradient errors are sublinear. Steady state tracking errors are developed for the scenarios where these quantities are not sublinear. Further, the proposed framework is utilized to develop online variants of the incremental gradient and the proximal gradient descent algorithms. Tests on the multi-target tracking and movie recommendation problems showcase the efficacy and low-complexity of the proposed algorithm.

While the present work broadens the applicability of OGD-like methods to time-varying non-strongly convex functions, it also motivates some interesting research directions. To begin with, the assumptions made here still preclude a large class of machine learning problems with non-differentiable objective functions. More generally, it may be possible to relax **(A4)** and characterize the dynamic regret for general time-varying and constrained convex optimization problems. In practical multi-agent systems, it may also be desirable that the nodes minimize their communication overhead and use distributed OGD variants such as the cyclic incremental

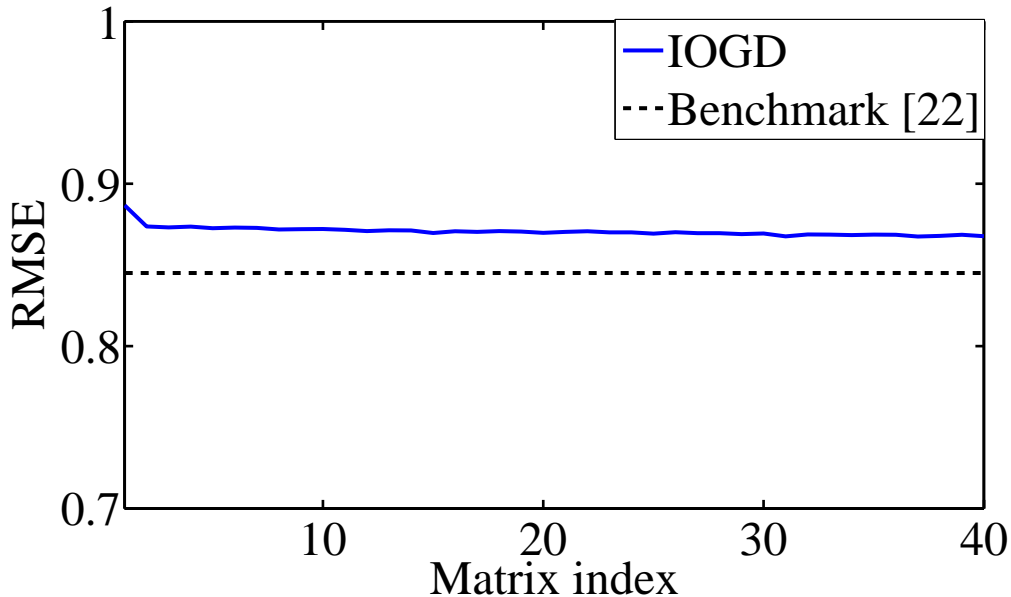


Fig. 6: Tracking RMSE for the MovieLens Dataset

gradient descent method. Next, the existing literature on online learning relies heavily on characterizing the performance of the algorithms using the dynamic regret as defined in (2). In the context of target tracking however, it may be useful to consider a more appropriate exponentially weighted dynamic regret, defined as $\mathbf{Reg}^\omega := \sum_{k=1}^K \omega^{K-k} (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*))$ for $0 < \omega < 1$. By assigning higher weights to the decisions made in the recent past, such a definition is apposite for tracking problems with shorter memory. Such an exponential window function is also reminiscent of the classical adaptive filtering algorithms such as the recursive least squares algorithm.

APPENDIX A

PRELIMINARIES

To begin with, several preliminary results required for establishing various bounds are derived. Observe that for P_1 , it follows from the convexity and quadratic growth property of f_k that for all $\mathbf{x} \in \mathcal{X}$,

$$\begin{aligned} \nabla f_k(\mathbf{x})^T (\mathbf{x} - P_k(\mathbf{x})) &\geq f_k(\mathbf{x}) - f_k(P_k(\mathbf{x})) \\ &\geq \frac{\mu}{2} \|\mathbf{x} - P_k(x)\|^2. \end{aligned} \quad (64)$$

Likewise, the following gradient bounds follow from **(A3)**-**(A4)**:

$$\begin{aligned} \|\nabla f_k(\mathbf{x}_k)\| &= \|\nabla f_k(\mathbf{x}_k) - \nabla f_k(P_k(\mathbf{x}_k))\| \\ &\leq L \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| = L \operatorname{dist}(\mathbf{x}_k, \mathcal{X}_k^*) \end{aligned} \quad (65)$$

$$\|\nabla f_k(\mathbf{x}_k)\|^2 \leq 2L(f_k(\mathbf{x}_k) - f_k^*). \quad (66)$$

Note that $f_k^* = f_k(P_k(\mathbf{x}_k))$ and this term will be used interchangeably throughout the analysis. The following inequality will be utilized towards obtaining the bounds in Sec. V. For positive scalars u , v , and w that satisfy $u^2 > v$, it holds that

$$\begin{aligned} \sqrt{u^2 - v + w^2} &\leq \sqrt{u^2 - v + v^2/4u^2 + w^2} \\ &= \sqrt{u^2 \left(1 - \frac{v}{2u^2}\right)^2 + w^2} \\ &\leq u \left(1 - \frac{v}{2u^2}\right) + w \\ &= u - \frac{v}{2u} + w. \end{aligned} \quad (67)$$

APPENDIX B

PROOF OF LEMMA 1, COROLLARY 1, AND LEMMA 3

Proof of (11): The proof begins by expanding the expression for $\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|^2$ in terms of $\operatorname{dist}(\mathbf{x}_k, \mathcal{X}_k^*)$. Specifically, since $\mathcal{P}_{\mathcal{X}}(P_k(\mathbf{x}_k)) = P_k(\mathbf{x}_k)$, it follows from the non-expansiveness property of the projection $\mathcal{P}_{\mathcal{X}}(\cdot)$ that,

$$\begin{aligned} \|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|^2 &\leq \|\mathbf{x}_k - \alpha(\nabla f_k(\mathbf{x}_k) + \mathbf{e}_k) - P_k(\mathbf{x}_k)\|^2 \\ &= \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 - 2\alpha \nabla f_k(\mathbf{x}_k)^T (\mathbf{x}_k - P_k(\mathbf{x}_k)) \\ &\quad + \alpha^2 \|\nabla f_k(\mathbf{x}_k)\|^2 + \alpha^2 \|\mathbf{e}_k\|^2 \\ &\quad - 2\alpha \mathbf{e}_k^T (\mathbf{x}_k - P_k(\mathbf{x}_k)) + 2\alpha^2 \mathbf{e}_k^T \nabla f_k(\mathbf{x}_k) \end{aligned} \quad (68)$$

$$\begin{aligned} &\leq \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 - 2\alpha(f_k(\mathbf{x}_k) - f_k^*) \\ &\quad + \alpha^2 \|\mathbf{e}_k\|^2 + \alpha^2 \|\nabla f_k(\mathbf{x}_k)\|^2 \\ &\quad - 2\alpha \mathbf{e}_k^T (\mathbf{x}_k - P_k(\mathbf{x}_k)) + 2\alpha^2 \mathbf{e}_k^T \nabla f_k(\mathbf{x}_k) \end{aligned} \quad (69)$$

where the inequality in (69) follows from (64).

The next step is different, depending on the gradient error model in effect. In the general case (cf. **(A1)**), the last two terms in (69) can be bounded from the use of the Cauchy-Schwarz inequality as

$$\begin{aligned} & -2\alpha \mathbf{e}_k^T (\mathbf{x}_k - P_k(\mathbf{x}_k)) + 2\alpha^2 \mathbf{e}_k^T \nabla f_k(\mathbf{x}_k) \\ & \leq 2\alpha \|\mathbf{e}_k\| (\|\mathbf{x}_k - P_k(\mathbf{x}_k)\| + \alpha \|\nabla f_k(\mathbf{x}_k)\|). \end{aligned} \quad (70)$$

Taking the conditional expectation $\mathbb{E}_k[\cdot] := \mathbb{E}[\cdot | \mathcal{F}_k]$, we obtain

$$\begin{aligned} & \mathbb{E}_k [-2\alpha \mathbf{e}_k^T (\mathbf{x}_k - P_k(\mathbf{x}_k)) + 2\alpha^2 \mathbf{e}_k^T \nabla f_k(\mathbf{x}_k)] \\ & \leq 2\alpha(\varepsilon_k + \nu \|\nabla f_k(\mathbf{x}_k)\|) (\|\mathbf{x}_k - P_k(\mathbf{x}_k)\| + \alpha \|\nabla f_k(\mathbf{x}_k)\|). \end{aligned} \quad (71)$$

On the other hand, when the errors are zero-mean and i.i.d. (cf. **(A2)**), we have that \mathbf{x}_k is independent of \mathbf{e}_k , implying that

$$\mathbb{E}_k [-2\alpha \mathbf{e}_k^T (\mathbf{x}_k - P_k(\mathbf{x}_k)) + 2\alpha^2 \mathbf{e}_k^T \nabla f_k(\mathbf{x}_k)] = 0 \quad (72)$$

The inequalities in (71)-(72) can be encoded into a single inequality by making use of an indicator variable 1_d that takes the value 1 when **(A1)** is in effect and zero when both **(A1)** and **(A2)** are in effect. Taking conditional expectation in (69), we have from (71)-(72) that

$$\begin{aligned} & \mathbb{E}_k [\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|^2] \\ & \leq \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 - 2\alpha(f_k(\mathbf{x}_k) - f_k^*) \\ & \quad + 2\alpha 1_d (\varepsilon_k + \nu \|\nabla f_k(\mathbf{x}_k)\|) (\|\mathbf{x}_k - P_k(\mathbf{x}_k)\| + \alpha \|\nabla f_k(\mathbf{x}_k)\|) \\ & \quad + \alpha^2 \|\nabla f_k(\mathbf{x}_k)\|^2 + \alpha^2 \mathbb{E}_k [\|\mathbf{e}_k\|^2] \end{aligned} \quad (73)$$

$$\begin{aligned} & \leq \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 - 2\alpha(f_k(\mathbf{x}_k) - f_k^*) \\ & \quad + 2\alpha 1_d \left(\varepsilon_k \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| + \nu \|\nabla f_k(\mathbf{x}_k)\| \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| \right. \\ & \quad \left. + \alpha \varepsilon_k \|\nabla f_k(\mathbf{x}_k)\| + \nu \alpha \|\nabla f_k(\mathbf{x}_k)\|^2 \right) \\ & \quad + \alpha^2 (1 + \nu^2) \|\nabla f_k(\mathbf{x}_k)\|^2 + \alpha^2 \varepsilon_k^2 \end{aligned} \quad (74)$$

where we have used the error bound in (A1). Next using the inequality $2uv \leq u^2 + v^2$, and the inequalities in (65) and (66), we obtain

$$\begin{aligned}
\mathbb{E}_k [\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|^2] &\leq \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 \\
&\quad - 2\alpha(1 - L\alpha(1 + \nu^2) - 2\alpha\nu L1_d)(f_k(\mathbf{x}_k) - f_k^*) \\
&\quad + 2\alpha 1_d \left(\varepsilon_k(1 + \alpha L) \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| + \nu L \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 \right) \\
&\quad + \alpha^2 \varepsilon_k^2 \\
&= (1 + 2\alpha\nu L1_d) \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 + \alpha^2 \varepsilon_k^2 \\
&\quad - 2\alpha(1 - L\alpha(1 + \nu^2 + 2\nu 1_d))(f_k(\mathbf{x}_k) - f_k^*) \\
&\quad + 2\alpha 1_d \varepsilon_k(1 + \alpha L) \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|
\end{aligned} \tag{75}$$

where for the second term on the right of (75) to be positive, it is required that $\alpha L(1 + \nu^2 + 2\nu 1_d) < 1$. Next, the QG property of f_k implies that

$$\begin{aligned}
\mathbb{E}_k [\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|^2] &\leq \ell^2 \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 \\
&\quad + 2\alpha 1_d \varepsilon_k(1 + \alpha L) \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| + \alpha^2 \varepsilon_k^2
\end{aligned} \tag{76}$$

where

$$\ell^2 := 1 - \mu\alpha(1 - L\alpha(1 + \nu^2 + 2\nu 1_d)) + 2\alpha\nu L1_d < 1. \tag{77}$$

Note that it is always possible to find an appropriate α if $\mu > 2\nu L$. The use of Jensen's inequality yields

$$\begin{aligned}
\mathbb{E}_k [\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|] &\leq \sqrt{\mathbb{E}_k [\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|^2]} \\
&\leq \left(\ell^2 \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 + \alpha^2 \varepsilon_k^2 \right. \\
&\quad \left. + 2\alpha 1_d \varepsilon_k(1 + \alpha L) \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| \right)^{1/2}.
\end{aligned} \tag{78}$$

Observe that for the case when $1_d = 0$, the right hand side is upper bounded by $\ell \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| + \alpha\varepsilon_k$ from the triangle inequality. In order to obtain a compact expression for the general case, define

$$\begin{aligned}
\zeta &:= \begin{cases} \alpha & 1_d = 0 \\ \frac{\alpha(1 + \alpha L)}{\ell} & 1_d = 1 \end{cases} \\
&= \alpha + 1_d \alpha(1 + \alpha L - \ell)/\ell
\end{aligned} \tag{79}$$

and observe that $\zeta > \alpha$ since $\ell < 1$. Therefore, it follows from (78) that

$$\begin{aligned} \mathbb{E}_k [\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\|] &\leq \left(\ell^2 \|\mathbf{x}_k - P_k(\mathbf{x}_k)\|^2 + \zeta^2 \varepsilon_k^2 \right. \\ &\quad \left. + 2\alpha 1_d \varepsilon_k (1 + \alpha L) \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| \right)^{1/2} \end{aligned} \quad (80)$$

$$\leq \ell \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| + \zeta \varepsilon_k \quad (81)$$

which is the required expression. \blacksquare

Proof of (12): This proof will make use of the following bounds, that hold for any $\mathbf{x}^* \in \mathcal{X}^*$,

$$\|\mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{x}^*\| \geq \sigma_{\min} \|\mathbf{x} - P(\mathbf{x})\| \quad (82a)$$

$$\|\mathbf{E}\mathbf{x} - \mathbf{E}\mathbf{y}\| \leq \sigma_{\max} \|\mathbf{x} - \mathbf{y}\| \quad (82b)$$

where $\sigma_{\max} := \sigma_{\max}(\mathbf{E})$ and $\sigma_{\min} := \sigma_{\min}(\mathbf{E})$.

Using the lower bound in (82a) and from the triangle inequality, it holds for any sequence $\{\mathbf{x}_k^*\}_{k=1}^K$ that

$$\sum_{k=1}^K \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| \leq \frac{1}{\sigma_{\min}} \sum_{k=1}^K \|\mathbf{E}\mathbf{x}_k - \mathbf{E}\mathbf{x}_k^*\| \quad (83)$$

$$\begin{aligned} &\leq \frac{1}{\sigma_{\min}} \|\mathbf{E}\mathbf{x}_1 - \mathbf{E}\mathbf{x}_1^*\| \\ &\quad + \frac{1}{\sigma_{\min}} \sum_{k=1}^{K-1} (\|\mathbf{E}\mathbf{x}_{k+1} - \mathbf{E}\mathbf{x}_k^*\| + \|\mathbf{E}\mathbf{x}_{k+1}^* - \mathbf{E}\mathbf{x}_k^*\|). \end{aligned} \quad (84)$$

Next, since all points in \mathcal{X}_k^* map to a unique point \mathbf{u}_k^* , it holds that $\mathbf{E}\mathbf{x}_k^* = \mathbf{E}P_k(\mathbf{x}_k)$. Therefore, the upper bound in (82b) implies that

$$\begin{aligned} \sum_{k=1}^K \|\mathbf{x}_k - P_k(\mathbf{x}_k)\| &\leq \frac{\sigma_{\max}}{\sigma_{\min}} \|\mathbf{x}_1 - \mathbf{x}_1^*\| \\ &\quad + \frac{\sigma_{\max}}{\sigma_{\min}} \sum_{k=1}^{K-1} (\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\| + \|\mathbf{x}_{k+1}^* - \mathbf{x}_k^*\|). \end{aligned} \quad (85)$$

While the result in (85) holds for any arbitrary $\{\mathbf{x}_k^*\}$, of particular interest is a sequence with sublinear path length, as defined in (3). Multiplying both sides of (85) by $\chi = \frac{\sigma_{\min}}{\sigma_{\max}}$, taking expectations, and substituting the result from (81), we obtain

$$\begin{aligned} \chi \sum_{k=1}^K \mathbb{E} [\|\mathbf{x}_k - P_k(\mathbf{x}_k)\|] &\leq \|\mathbf{x}_1 - \mathbf{x}_1^*\| + W_K \\ &\quad + \ell \sum_{k=1}^K \mathbb{E} [\|\mathbf{x}_k - P_k(\mathbf{x}_k)\|] + \zeta \sum_{k=1}^K \varepsilon_k. \end{aligned} \quad (86)$$

Since $\ell < \chi$, it therefore holds that

$$\sum_{k=1}^K \mathbb{E} [\|\mathbf{x}_k - P_k(\mathbf{x}_k)\|] \leq \frac{\|\mathbf{x}_1 - \mathbf{x}_1^*\| + W_K + \zeta E_K}{\chi - \ell}. \quad (87)$$

■

Proof of Lemma 3 and equation (13): Using the bounds in (82) and the triangle inequality, it follows that

$$\|\mathbf{x}_{k+1} - P_{k+1}(\mathbf{x}_{k+1})\| \leq \frac{1}{\sigma_{\min}} \|\mathbf{E}\mathbf{x}_{k+1} - \mathbf{E}\mathbf{x}_{k+1}^*\| \quad (88)$$

$$\leq \frac{1}{\sigma_{\min}} (\|\mathbf{E}\mathbf{x}_{k+1} - \mathbf{E}\mathbf{x}_k^*\| + \|\mathbf{E}\mathbf{x}_{k+1}^* - \mathbf{E}\mathbf{x}_k^*\|) \quad (89)$$

$$= \frac{1}{\sigma_{\min}} (\|\mathbf{E}\mathbf{x}_{k+1} - \mathbf{E}P_k(\mathbf{x}_k)\| + \|\mathbf{E}\mathbf{x}_{k+1}^* - \mathbf{E}\mathbf{x}_k^*\|) \quad (90)$$

$$\leq \frac{1}{\chi} (\|\mathbf{x}_{k+1} - P_k(\mathbf{x}_k)\| + \sigma) \quad (91)$$

where the equality in (90) holds since $\mathbf{E}\mathbf{x}_k^* = \mathbf{E}P_k(\mathbf{x}_k)$ and the last inequality makes use of the bounded variation property in (A5). Taking expectation, substituting the result in (81), and using the bounds in (36), we obtain

$$\mathbb{E} [\|\mathbf{x}_{k+1} - P_{k+1}(\mathbf{x}_{k+1})\|] \leq \ell/\chi \mathbb{E} [\|\mathbf{x}_k - P_k(\mathbf{x}_k)\|] + \frac{\zeta\varepsilon + \sigma}{\chi} \quad (92)$$

Recursive use of (92) yields

$$\begin{aligned} \mathbb{E} [\text{dist}(\mathbf{x}_{k+1}, \mathcal{X}_{k+1}^*)] &\leq (\ell/\chi)^k \text{dist}(\mathbf{x}_1, \mathcal{X}_1^*) \\ &\quad + \left[\frac{(1 - (\ell/\chi)^k)}{(1 - \ell/\chi)} \right] \frac{\zeta\varepsilon + \sigma}{\chi}. \end{aligned} \quad (93)$$

which is the required result in the statement of Lemma 3. The result in (13) can be obtained by using the upper bound $\frac{\ell}{\chi} < 1$ in (93), resulting in

$$\mathbb{E} [\text{dist}(\mathbf{x}_{k+1}, \mathcal{X}_{k+1}^*)] \leq \text{dist}(\mathbf{x}_1, \mathcal{X}_1^*) + \frac{\zeta\varepsilon + \sigma}{\chi} \quad (94)$$

which, along with (65), yields the required result. ■

APPENDIX C

PROOF OF LEMMA. 2 AND COROLLARY 2

As earlier, the proofs of Lemma 2 and Corollary 2 will be developed in a unified manner. To this end, we will again make use of the indicator function 1_d that takes values 0 or 1, depending on the assumption under effect. In particular, the following two cases are considered: (a) the

general case under **(A1)** with $\nu = 0$ for which $1_d = 1$ (Lemma 2); and (b) the white noise error case under **(A1)**-**(A2)** for which $1_d = 0$ Corollary 2.

Proof of Lemma 2: Given a sequence $\{\mathbf{x}_k^*\}$ satisfying (3), and using the first order convexity property of f_k , we obtain

$$\begin{aligned} & \|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|^2 \\ &= \|\mathbf{x}_k - \mathbf{x}_k^*\|^2 - 2\alpha \nabla f_k(\mathbf{x}_k)^T (\mathbf{x}_k - \mathbf{x}_k^*) + \alpha^2 \|\nabla f_k(\mathbf{x}_k)\|^2 \\ &\quad - 2\alpha \mathbf{e}_k^T (\mathbf{x}_k - \mathbf{x}_k^*) + \alpha^2 \|\mathbf{e}_k\|^2 + 2\alpha^2 \mathbf{e}_k^T \nabla f_k(\mathbf{x}_k)^T \end{aligned} \quad (95)$$

$$\begin{aligned} &\leq \|\mathbf{x}_k - \mathbf{x}_k^*\|^2 - 2\alpha (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) + \alpha^2 \|\mathbf{e}_k\|^2 \\ &\quad + 2\alpha^2 L (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) + 2\alpha^2 \mathbf{e}_k^T \nabla f_k(\mathbf{x}_k) \\ &\quad - 2\alpha \mathbf{e}_k^T (\mathbf{x}_k - \mathbf{x}_k^*) \end{aligned} \quad (96)$$

Next, we take conditional expectation given \mathcal{F}_k in (96), and consider the two cases separately. In the general case under **(A1)** with $\nu = 0$, the last two terms can be bounded as in (71) to yield

$$\begin{aligned} & \mathbb{E}_k [2\alpha^2 \mathbf{e}_k^T \nabla f_k(\mathbf{x}_k) - 2\alpha \mathbf{e}_k^T (\mathbf{x}_k - \mathbf{x}_k^*)] \\ & \leq 2\alpha \mathbb{E}_k [\|\mathbf{e}_k\|] (\alpha \|\nabla f_k(\mathbf{x}_k)\| + \|\mathbf{x}_k - \mathbf{x}_k^*\|) \end{aligned} \quad (97)$$

$$\leq 2\alpha \varepsilon_k (\alpha \|\nabla f_k(\mathbf{x}_k)\| + \|\mathbf{x}_k - \mathbf{x}_k^*\|) \quad (98)$$

$$\leq \alpha^2 (\varepsilon_k^2 + \|\nabla f_k(\mathbf{x}_k)\|^2) + 2\alpha \varepsilon_k R \quad (99)$$

where recall that $R := \text{diam}(\mathcal{X})$. Inequality in (99) follows from $2uv \leq u^2 + v^2$. Finally, it follows from (66) that

$$\begin{aligned} & \mathbb{E}_k [2\alpha^2 \mathbf{e}_k^T \nabla f_k(\mathbf{x}_k) - 2\alpha \mathbf{e}_k^T (\mathbf{x}_k - \mathbf{x}_k^*)] \\ & \leq \alpha^2 \varepsilon_k^2 + 2\alpha^2 L (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) + 2\alpha \varepsilon_k R. \end{aligned} \quad (100)$$

In the second case, when gradient errors are zero mean and i.i.d., it can be seen from (97) that the right-hand side is zero. Combining the two cases within (96), it follows that

$$\begin{aligned} & \mathbb{E}_k [\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|^2] \leq \|\mathbf{x}_k - \mathbf{x}_k^*\|^2 \\ & \quad - 2\alpha (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) + \alpha^2 \mathbb{E}_k [\|\mathbf{e}_k\|^2] + \alpha^2 \|\nabla f_k(\mathbf{x}_k)\|^2 \\ & \quad + 1_d (\alpha^2 \varepsilon_k^2 + 2\alpha^2 L (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) + 2\alpha \varepsilon_k R) \end{aligned} \quad (101)$$

$$\begin{aligned} & \leq \|\mathbf{x}_k - \mathbf{x}_k^*\|^2 + \alpha^2 \varepsilon_k^2 (1_d + 1) + 2\alpha \varepsilon_k R 1_d \\ & \quad - 2\alpha (1 - \alpha L (1 + \nu^2 + 1_d (1 - \nu^2))) (f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) \end{aligned} \quad (102)$$

where we have again used (66) and the bounds in (A1) and for the two cases. It is remarked that the last term in (102) is negative for $\alpha L < 1/(1 + \nu^2 + 1_d(1 - \nu^2))$. Henceforth, denote

$$\xi := 2\alpha(1 - \alpha L(1 + \nu^2 + 1_d(1 - \nu^2))) \quad (103)$$

$$s_k^2 := \alpha^2 \varepsilon_k^2 (1_d + 1) + 2\alpha \varepsilon_k R 1_d. \quad (104)$$

Using (67), it follows that

$$\begin{aligned} \left[\mathbb{E}_k [\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|^2] \right]^{1/2} &\leq \|\mathbf{x}_k - \mathbf{x}_k^*\| - \frac{\xi(f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*))}{\|\mathbf{x}_k - \mathbf{x}_k^*\|} + s_k \\ &\leq \|\mathbf{x}_k - \mathbf{x}_k^*\| - \frac{\xi}{R}(f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)) + s_k. \end{aligned} \quad (105)$$

Taking full expectation and using the Jensen's inequality, it follows that

$$\begin{aligned} \mathbb{E} [\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|] &\leq \mathbb{E} \left[\left[\mathbb{E}_k [\|\mathbf{x}_{k+1} - \mathbf{x}_k^*\|^2] \right]^{1/2} \right] \\ &\leq \mathbb{E} [\|\mathbf{x}_k - \mathbf{x}_k^*\|] - \frac{\xi}{R} \mathbb{E} [f_k(\mathbf{x}_k) - f_k(\mathbf{x}_k^*)] + s_k \end{aligned} \quad (106)$$

which is the required result. ■

REFERENCES

- [1] S. Dubowsky, K. Iagnemma, S. Liberatore, D. Lambeth, J. Plante, and P. Boston, "A concept mission: Microbots for large-scale planetary surface and subsurface exploration," in *AIP Conf. Proc.*, vol. 746, no. 1, 2005, pp. 1449–1458.
- [2] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Auton. Robots*, vol. 8, no. 3, pp. 345–383, 2000.
- [3] J. Ota, "Multi-agent robot systems as distributed autonomous systems," *Advanced engineering informatics*, vol. 20, no. 1, pp. 59–70, 2006.
- [4] J. C. Derenick, *A Convex Optimization Framework for Multi-agent Motion Planning*. ProQuest, 2009.
- [5] S.-Y. Tu and A. H. Sayed, "On the influence of informed agents on learning and adaptation over networks," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1339–1356, 2013.
- [6] A. Trafton, "Team envisions exploring mars with mini probes (mit news office)." [Online]. Available: <http://web.mit.edu/newsoffice/2006/microbots.html>.
- [7] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Adv. Neural Inf. Process. Syst.*, 2007, pp. 1–8.
- [8] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. of the 20th Int. Conf. on Machine Learning*, 2003, pp. 928–936.
- [9] 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, 2015.
- [10] A. Simonetto, A. Koppel, A. Mokhtari, G. Leus, and A. Ribeiro, "Decentralized prediction-correction methods for networked time-varying convex optimization," *IEEE Trans. on Autom. Control*, 2017.
- [11] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *55th IEEE Conf. on Decision and Control (CDC)*, 2016, pp. 7195–7201.

- [12] L. Zhang, T. Yang, J. Yi, R. Jin, and Z.-H. Zhou, “Improved dynamic regret for non-degeneracy functions,” *arXiv preprint arXiv:1608.03933*, 2016.
- [13] C. Castillo, W. Moreno, and K. Valavanis, “Unmanned helicopter waypoint trajectory tracking using model predictive control,” in *IEEE Mediterranean Conference on Control & Automation (MED’07)*, 2007, pp. 1–8.
- [14] R. Abdolee, B. Champagne, and A. H. Sayed, “Estimation of space-time varying parameters using a diffusion lms algorithm,” *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 403–418, 2014.
- [15] K. Zhou and S. I. Roumeliotis, “Multirobot active target tracking with combinations of relative observations,” *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 678–695, 2011.
- [16] D. Perera, J. Kay, I. Koprinska, K. Yacef, and O. R. Zaïane, “Clustering and sequential pattern mining of online collaborative learning data,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 6, pp. 759–772, 2009.
- [17] O. Besbes, Y. Gur, and A. Zeevi, “Non-stationary stochastic optimization,” *Operations Research*, vol. 63, no. 5, pp. 1227–1244, 2015.
- [18] S. Shahrampour and A. Jadbabaie, “Distributed online optimization in dynamic environments using mirror descent,” *arXiv preprint arXiv:1609.02845*, 2016.
- [19] M. P. Friedlander and M. Schmidt, “Hybrid deterministic-stochastic methods for data fitting,” *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. A1380–A1405, 2012.
- [20] M. Schmidt, N. L. Roux, and F. R. Bach, “Convergence rates of inexact proximal-gradient methods for convex optimization,” in *Adv. Neural Inf. Process. Syst.*, 2011, pp. 1458–1466.
- [21] J. Derenick, J. Spletzer, and A. Hsieh, “An optimal approach to collaborative target tracking with performance guarantees,” *J. Intell. and Robotic Syst.*, vol. 56, no. 1-2, pp. 47–67, 2009.
- [22] C. Dhanjal, R. Gaudel, and S. Cléménçon, “Online matrix completion through nuclear norm regularisation,” in *Proc. of the SIAM Int. Conf. on Data Mining*. SIAM, 2014, pp. 623–631.
- [23] M. Schmidt, N. L. Roux, and F. R. Bach, “Convergence rates of inexact proximal-gradient methods for convex optimization,” in *Adv. Neural Inf. Process. Syst.*, 2011, pp. 1458–1466.
- [24] H. Zhang and L. Cheng, “Restricted strong convexity and its applications to convergence analysis of gradient-type methods in convex optimization,” *Optimization Letters*, vol. 9, no. 5, pp. 961–979, 2015.
- [25] G. Lan, *Convex optimization under inexact first-order information*. Georgia Institute of Technology, 2009.
- [26] D. Blatt, A. O. Hero, and H. Gauchman, “A convergent incremental gradient method with a constant step size,” *SIAM J. on Opt.*, vol. 18, no. 1, pp. 29–51, 2007.
- [27] N. L. Roux, M. Schmidt, and F. R. Bach, “A stochastic gradient method with an exponential convergence rate for finite training sets,” in *Adv. Neural Inf. Process. Syst.*, 2012, pp. 2663–2671.
- [28] A. Nedic and A. E. Ozdaglar, “Approximate primal solutions and rate analysis for dual subgradient methods,” *SIAM J. Opt.*, vol. 19, pp. 1757–1780, 2009.
- [29] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in *Adv. Neural Inf. Process. Syst.*, 2013, pp. 315–323.
- [30] A. Defazio, F. Bach, and S. Lacoste-Julien, “SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives,” in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 1646–1654.
- [31] A. M.-C. So, “Non-asymptotic convergence analysis of inexact gradient methods for machine learning without strong convexity,” *arXiv preprint arXiv:1309.0113*, 2013.
- [32] Z.-Q. Luo and P. Tseng, “On the linear convergence of descent methods for convex essentially smooth minimization,” *SIAM J. Control and Opt.*, vol. 30, no. 2, pp. 408–425, 1992.

- [33] J.-H. Hours and C. N. Jones, “A parametric nonconvex decomposition algorithm for real-time and distributed NMPC,” *IEEE Trans. Automa. Control*, vol. 61, no. 2, pp. 287–302, 2016.
- [34] A. L. Dontchev, M. Krastanov, R. T. Rockafellar, and V. M. Veliov, “An Euler–Newton continuation method for tracking solution trajectories of parametric variational inequalities,” *SIAM J. Control and Opt.*, vol. 51, no. 3, pp. 1823–1840, 2013.
- [35] V. M. Zavala and M. Anitescu, “Real-time nonlinear optimization as a generalized equation,” *SIAM J. on Cont.and Opt.*, vol. 48, no. 8, pp. 5444–5467, 2010.
- [36] A. Y. Popkov, “Gradient methods for nonstationary unconstrained optimization problems,” *Autom. Remote Control*, vol. 66, no. 6, pp. 883–891, 2005.
- [37] A. L. Dontchev and R. T. Rockafellar, “Implicit functions and solution mappings,” *Springer Monogr. Math.*, 2009.
- [38] C. Wilson, V. Veeravalli, and A. Nedić, “Dynamic stochastic optimization,” in *53rd IEEE Conf. on Decision and Control*, 2014, pp. 173–178.
- [39] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, “Online optimization: Competing with dynamic comparators,” in *Artificial Intelligence and Statistics*, 2015, pp. 398–406.
- [40] A. Simonetto, “Time-varying convex optimization via time-varying averaged operators,” *arXiv preprint arXiv:1704.07338*, 2017.
- [41] 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 *International Conference on Machine Learning*, 2016, pp. 449–457.
- [42] Q. T. Dinh, C. Savorgnan, and M. Diehl, “Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization,” *SIAM J. on Opt.*, vol. 22, no. 4, pp. 1258–1284, 2012.
- [43] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, “A class of prediction-correction methods for time-varying convex optimization,” *arXiv preprint arXiv:1509.05196*, 2015.
- [44] A. Koppel, A. Simonetto, A. Mokhtari, G. Leus, and A. Ribeiro, “Target tracking with dynamic convex optimization,” in *Proceedings of the IEEE Global Conference on Signal and Information Processing (to appear)*, 2015.
- [45] A. Simonetto, A. Koppel, A. Mokhtari, G. Leus, and A. Ribeiro, “Prediction-correction methods for time-varying convex optimization,” in *Proceedings of the Asilomar Conference on Signals, Systems, and Computers (submitted)*, (Pacific Grove, USA), 2015.
- [46] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, “Prediction-correction interior-point method for time-varying convex optimization,” *arXiv preprint arXiv:1608.07544*, 2016.
- [47] A. Simonetto and G. Leus, “On non-differentiable time-varying optimization,” in *IEEE 6th Int. Workshop on Computational Advances in Multi-Sensor Adaptive Process.*, 2015, pp. 505–508.
- [48] T. Chen, Q. Ling, and G. B. Giannakis, “An online convex optimization approach to dynamic network resource allocation,” *arXiv preprint arXiv:1701.03974*, 2017.
- [49] D. P. Bertsekas and J. N. Tsitsiklis, “Gradient convergence in gradient methods with errors,” *SIAM J. on Opt.*, vol. 10, no. 3, pp. 627–642, 2000.
- [50] L. Rosasco, S. Villa, and B. C. Vũ, “Convergence of stochastic proximal gradient algorithm,” *arXiv preprint arXiv:1403.5074*, 2014.
- [51] H. Karimi, J. Nutini, and M. Schmidt, “Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 795–811.
- [52] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, “A class of prediction-correction methods for time-varying convex optimization.” *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4576–4591, 2016.
- [53] S. Lohr, *Sampling: design and analysis*. Nelson Education, 2009.

- [54] R. Tripathi, B. Mohan, and K. Rajawat, "Adaptive low-rank matrix completion," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3603–3616, July 2017.