2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

An Online Optimization Approach for Multi-Agent Tracking of Dynamic Parameters in the Presence of Adversarial Noise

Shahin Shahrampour and Ali Jadbabaie

Abstract— **This paper addresses tracking of a moving target in a multi-agent network. The target follows a linear dynamics corrupted by an adversarial noise, i.e., the noise is not generated from a statistical distribution. The location of the target at each time induces a global time-varying loss function, and the global loss is a sum of local losses, each of which is associated to one agent. Agents noisy observations could be nonlinear. We formulate this problem as a distributed online optimization where agents communicate with each other to track the minimizer of the global loss. We then propose a decentralized version of the Mirror Descent algorithm and provide the non-asymptotic analysis of the problem. Using the notion of dynamic regret, we measure the performance of our algorithm versus its offline counterpart in the centralized setting. We prove that the bound on dynamic regret scales inversely in the network spectral gap, and it represents the adversarial noise causing deviation with respect to the linear dynamics. Our result subsumes a number of results in the distributed optimization literature. Finally, in a numerical experiment, we verify that our algorithm can be simply implemented for multi-agent tracking with nonlinear observations.**

I. Introduction

Distributed estimation, detection, and tracking is ubiquitous in engineering applications ranging from sensor and robotic networks to social networks, and it has received a lot of attention for many years [1]–[5]. In these scenarios, the task is to estimate the value of a parameter which may or may not be dynamic. A group of agents aim to accomplish this task as a team. Each individual agent only partially observes the parameter, but the global spread of observations in the network allows agents to estimate the parameter collaboratively. This would require agents to aggregate local information, and many methods use consensus protocols as a critical component [6]. It is well-known that when agents' observations are linear with respect to the parameter, the tracking problem is equivalent to minimizing a *global* quadratic loss, written as a sum of *local* quadratic losses (see e.g. [7]). However, in general, the global loss can be more complicated, resulting in nonlinear observations.

In real-world applications, the parameter of interest is often time-varying. Therefore, regardless of the structure of the loss, the dynamic nature of the problem brings forward two issues: *(i)* The local losses are observed in an *online* or *sequential* fashion, i.e., the local losses are disclosed to agents only after they form their estimates at each round, and they are not aware of future loss functions. Therefore, the problem must be solved in an online setting. *(ii)* The online algorithm should mimic the performance of its offline counterpart in which the losses are known a priori. The gap between the two is often called *regret*. Tracking the minimizer of the global loss over time introduces the notion of *dynamic* regret [8]. This framework has been studied in centralized online optimization [8]–[12], where the hardness of the problem is captured via the variation in the minimizer sequence.

To address these issues in this paper, we adopt an online optimization approach to formulate distributed tracking. We consider tracking of a dynamic parameter or a moving target in a *network* of agents. The dynamics of the target is linear and known to agents, but the target deviates from this dynamics due to an *unstructured* or *adversarial* disturbance or noise. In other words, the noise is not necessarily generated from a statistical distribution, or it can be highly correlated to its past values over time. At each time instance, the target induces a *global* convex loss whose minimizer coincides with the target location. The global loss is a sum of *local* losses, where each local loss is associated to a specific agent. Agents exchange noisy local gradients according to a communication protocol to track the moving target.

Our problem setup is reminiscent of a distributed Kalman [13]. However, we differentiate the two as follows: *(i)* We do not assume that the target is driven by a Gaussian noise. Nor do we assume that this noise has a statistical distribution. Instead, we consider an adversarial-noise model with *unknown* structure. *(ii)* Agents observations are not necessarily linear; in fact, the observations are noisy local gradients that are non-linear when the loss is not quadratic. Furthermore, our focus is on the *finite-time* analysis rather than asymptotic results.

We propose a decentralized version of the Mirror Descent algorithm, developed by Nemirovksi and Yudin [14]. Using the notion of Bregman divergence in lieu of Euclidean distance for projection, Mirror Descent has been shown to be a powerful tool in large-scale optimization. Our algorithm consists of three interleaved updates: *(i)* each agent follows the noisy local gradient while staying close to previous estimates in the local neighborhood; *(ii)* agents take into account the dynamics of the moving target; *(iii)* agents average their estimates in their local neighborhood in a consensus step.

We then use a *dynamic* notion of regret to measure the difference between our online decentralized algorithm and its offline centralized version. We establish a regret bound that scales inversely in the spectral gap of the network, and it represents the adversarial noise causing deviation with respect to the linear dynamics. We further show that from optimization perspective our result subsumes two important classes of decentralized optimization in the literature: *(i)* decentralized optimization of *time-invariant* losses, and *(ii)* decentralized optimization of time-variant losses for *fixed* targets. This generalization is achieved by allowing the loss function and the target value to vary simultaneously. We also provide a numerical experiment to show that our algorithm can be simply implemented to work with nonlinear observations in multi-agent tracking.

Related Literature on Decentralized Optimization: In [15], decentralized mirror descent has been developed for time-invariant functions in the case that agents receive the gradients with a delay. Moreover, Rabbat in [16] proposes a decentralized mirror descent for stochastic composite optimization problems and provide guarantees for strongly convex regularizers. Duchi et al. [17] study

This work was supported by ONR BRC Program on Decentralized, Online Optimization.

Shahin Shahrampour is with the Department of Electrical Engineering at Harvard University, Cambridge, MA 02138 USA. (e-mail: shahin@seas.harvard.edu).

Ali Jadbabaie is with the Institute for Data, Systems, and Society at Massachusetts Institute of Technology, Cambridge, MA 02139 USA. (email: jadbabai@mit.edu).

978-1-5090-5992-8/$31.00 ©2017 AACC

dual averaging for distributed optimization, and the extension of dual averaging to online distributed optimization is considered in [18]. Mateos-Núnez and Cortés [19] consider online optimization using subgradient descent of local functions, where the graph structure is time-varying. In [20], a decentralized variant of Nesterov's primal-dual algorithm is proposed for online optimization. In [21], distributed online optimization is studied for strongly convex objective functions over time-varying networks. Our setup follows the work of [22] on decentralized online mirror descent, but we extend the results to *high probability* bounds on the dynamic regret.

$[n]$	The set $\{1, 2, ..., n\}$ for any integer n
$x^\top$	Transpose of the vector x
$x(k)$	The k-th element of vector x
I_n	Identity matrix of size n
Δ_d	The d-dimensional probability simplex
$\langle \cdot, \cdot \rangle$	Standard inner product operator
$\|\cdot\|_p$	p-norm operator
$\|\cdot\|_*$	The dual norm of $\|\cdot\|$
$\sigma_i(W)$	The i-th largest eigenvalue of W in magnitude

II. Problem Formulation and Algorithm

A. Dynamical Model and Optimization Perspective

Consider a d-dimensional moving target $x_t^\star$ following the linear dynamics A for a *finite* time T as

$$x_{t+1}^\star = Ax_t^\star + v_t, \qquad t \in [T] \tag{1}$$

where $A \in \mathbb{R}^{d\times d}$ is *known*, and $v_t \in \mathbb{R}^d$ is an *adversarial* noise, i.e., the sequence $\{v_t\}_{t=1}^T$ is neither generated according to a statistical distribution, nor it is independent over time. Our goal is to track $x_t^\star$, and regardless of the observation model, a distribution-dependent mechanism, such as Kalman or particle filter, cannot solve the problem since the noise does not assume a statistical distribution.

In the centralized version of the tracking problem above, the observations of $x_t^\star$ are realized through a *time-varying*, *global* loss function. That is, consider the tracking problem above as an optimization, where $x_t^\star$ is the minimizer of the global loss at time t. Let $\mathcal{X}$ be a convex, compact set, and represent the global loss by $f_t : \mathcal{X} \to \mathbb{R}$ at time t. As the global loss varies over time, the goal is to track the minimizer of $f_t(\cdot)$, which is $x_t^\star$. The *offline* and *centralized* version of our problem can be viewed as follows

$$\begin{aligned} \underset{x_1,\dots,x_T}{\text{minimize}} \quad & \sum_{t=1}^{T} f_t(x_t) \\ \text{subject to} \quad & x_t \in \mathcal{X},\ t \in [T]. \end{aligned} \tag{2}$$

We are interested to solve the problem above in an *online* and *decentralized* fashion. In particular, the global function at time t is a sum of n *local* functions as

$$f_t(x) := \frac{1}{n}\sum_{i=1}^{n} f_{i,t}(x), \tag{3}$$

where $f_{i,t} : \mathcal{X} \to \mathbb{R}$ is a local convex function on $\mathcal{X}$ for all $i \in [n]$. We consider a network of n agents facing two challenges when solving problem (2): *(i)* agent $j \in [n]$ receives information only about $f_{j,t}(\cdot)$ and does not observe the global loss function $f_t(\cdot)$, which is common to decentralized schemes; *(ii)* The functions are revealed to agents sequentially along the time horizon, i.e., at any time instance s, agent j has observed $f_{j,t}(\cdot)$ for $t < s$, whereas the agent does not know $f_{j,t}(\cdot)$ for $s \le t \le T$, which is common to online settings.

The agents interact with each another, and their relationship is captured via an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [n]$ denotes the set of nodes, and $\mathcal{E}$ is the set of edges. Each agent i assigns a positive weight $[W]_{ij}$ for the information received from agent $j \neq i$, and the set of neighbors of agent i is defined as $\mathcal{N}_i := \{j : [W]_{ij} > 0\}$.

While the problem framework is reminiscent of a distributed Kalman [13], there are fundamental distinctions in our setup: *(i)* The adversarial noise v_t is neither Gaussian nor of known statistical distribution. It can be thought as a noise with *unknown* structure, which represents the deviation from the dynamics[1]. *(ii)* Agents observations are not necessarily linear; in fact, the observations are local gradients of $\{f_{i,t}(\cdot)\}_{t=1}^T$ and are non-linear when the objective is not quadratic. The other implicit distinction in this work is our focus on *finite-time* analysis rather than asymptotic results.

From optimization perspective, our framework subsumes two important classes of decentralized optimization in the literature:

1) Existing methods often consider *time-invariant* objectives (see e.g. [15], [17], [24]). This is simply the special case where $f_t(x) = f(x)$ and $x_t = x$ in (2).
2) Online algorithms deal with *time-varying* functions, but often the network's objective is to minimize the temporal average of $\{f_t(x)\}_{t=1}^T$ over a fixed variable x (see e.g. [18], [19]). This can be captured by our setup when $x_t = x$ in (2).

However, in the tracking problem, *functions* and *comparator variables* evolve simultaneously, i.e., the variables $\{x_t\}_{t=1}^T$ are not constrained to be fixed in (2). Recall that $x_t^\star := \text{argmin}_{x\in\mathcal{X}} f_t(x)$ is the minimizer of the global loss function at time t. Then, the solution to problem (2) is simply $\sum_{t=1}^T f_t(x_t^\star)$. Denote by $\mathbf{x}_{i,t}$ the estimate of agent i for $x_t^\star$ at time t. To exhibit the online nature of problem (2), we reformulate it using the notion of *dynamic* regret as follows

$$\boldsymbol{Reg}_T^d = \frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{T} f_t(\mathbf{x}_{i,t}) - \sum_{t=1}^{T} f_t(x_t^\star). \tag{4}$$

Then, the objective is to minimize the dynamic regret above which measures the gap between the online algorithm and its offline version. Our performance bound shall exhibit the impact of system noise, i.e., we want to prove a regret bound in terms of

$$\|v_t\| = \|x_{t+1}^\star - Ax_t^\star\|, \tag{5}$$

which represents the deviation of the moving target with respect to dynamics A. Note that generalizing the results to the linear time-variant dynamics is straightforward, i.e., when A is replaced by A_t in (1).

B. Technical Assumptions

To solve the multi-agent online optimization (4), we propose to decentralize the Mirror Descent algorithm [14]. Mirror Descent has been shown to be a powerful method in large-scale optimization by using Bregman divergence in lieu of Euclidean distance in the projection step. Before defining Bregman divergence and elaborating the algorithm, we start by stating a couple of standard assumptions on loss functions and agents communication.

Assumption 1: For any $i \in [n]$, the function $f_{i,t}(\cdot)$ is Lipschitz continuous on $\mathcal{X}$ with a uniform constant L. That is,

$$|f_{i,t}(x) - f_{i,t}(y)| \le L\,\|x - y\|,$$

[1] In online optimization, the focus is not on distribution of data. Instead, data is thought to be generated arbitrarily, and its effect is observed through the loss functions [23].

for any $x, y \in \mathcal{X}$.

Assumption 2: The network is connected[2], i.e., there exists a path from any agent $i \in [n]$ to any agent $j \in [n]$. Also, the matrix W is symmetric and doubly stochastic with positive diagonal. That is,

$$\sum_{i=1}^{n} [W]_{ij} = \sum_{j=1}^{n} [W]_{ij} = 1.$$

The connectivity constraint in Assumption 2 guarantees the information flow in the network.

We now outline the notion of Bregman divergence, which is critical in the development of Mirror Descent. Consider a compact, convex set $\mathcal{X}$, and let $\mathcal{R} : \mathcal{X} \to \mathbb{R}$ denote a 1-strongly convex function on $\mathcal{X}$ with respect to a norm $\|\cdot\|$. That is,

$$\mathcal{R}(x) \geq \mathcal{R}(y) + \langle \nabla\mathcal{R}(y), x - y\rangle + \frac{1}{2}\|x - y\|^2 .$$

for any $x, y \in \mathcal{X}$. Then, the Bregman divergence $\mathcal{D}_{\mathcal{R}}(\cdot,\cdot)$ with respect to the function $\mathcal{R}(\cdot)$ is defined as follows:

$$\mathcal{D}_{\mathcal{R}}(x, y) := \mathcal{R}(x) - \mathcal{R}(y) - \langle x - y, \nabla\mathcal{R}(y)\rangle .$$

The definition of the Bregman divergence and the strong convexity of $\mathcal{R}(\cdot)$ imply that

$$\mathcal{D}_{\mathcal{R}}(x, y) \geq \frac{1}{2}\|x - y\|^2 , \tag{6}$$

for any $x, y \in \mathcal{X}$. Two famous examples of Bregman divergence are the Euclidean distance and the Kullback-Leibler (KL) divergence generated from $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$ and $\mathcal{R}(x) = \sum_{i=1}^{d} x(i)\log x(i) - x(i)$, respectively.

Assumption 3: Let x and $\{y_i\}_{i=1}^{n}$ be vectors in $\mathbb{R}^d$. We assume that the Bregman divergence satisfies the separate convexity in the following sense

$$\mathcal{D}_{\mathcal{R}}(x, \sum_{i=1}^{n} \alpha(i) y_i) \leq \sum_{i=1}^{n} \alpha(i)\mathcal{D}_{\mathcal{R}}(x, y_i),$$

where $\alpha \in \Delta_n$ is on the n-dimensional simplex.
The assumption is satisfied for commonly used cases of Bregman divergence. For instance, the Euclidean distance evidently respects the condition. The KL-divergence also satisfies the constraint, and we refer the reader to Theorem 6.4. in [25] for the proof.

Assumption 4: The Bregman divergence satisfies a Lipschitz condition of the form

$$|\mathcal{D}_{\mathcal{R}}(x, z) - \mathcal{D}_{\mathcal{R}}(y, z)| \leq K\|x - y\|,$$

for all $x, y, z \in \mathcal{X}$.
When the function $\mathcal{R}$ is Lipschitz on $\mathcal{X}$, the Lipschitz condition on the Bregman divergence is automatically satisfied. Again, for the Euclidean distance the assumption evidently holds. In the particular case of KL divergence, the condition can be achieved via mixing a uniform distribution to avoid the boundary (see e.g. [11] for more comments on the assumption).

Assumption 5: The dynamics A is assumed to be *non-expansive*. That is, the condition

$$\mathcal{D}_{\mathcal{R}}\big(Ax, Ay\big) \leq \mathcal{D}_{\mathcal{R}}\big(x, y\big),$$

holds for all $x, y \in \mathcal{X}$, and $\|A\| \leq 1$.
The assumption postulates a natural constraint on the dynamics A: it does not allow the effect of a poor estimation (at one step) to be amplified as the algorithm moves forward.

C. Decentralized Tracking via Online Mirror Descent

We now propose our algorithm to solve the problem formulated in terms of dynamic regret in (4). In our setting, agents observations are gradients of the local losses. However, common in distributed state estimation and tracking, these observations are noisy. Hence, denoting the local gradient of agent i at time t by $\nabla_{i,t} := \nabla f_{i,t}(\mathbf{x}_{i,t})$, the agent only receives $\boldsymbol{\nabla}_{i,t}$ representing the stochastic gradient. The stochastic oracle that provides noisy gradients satisfies the following constraints[3]

$$\mathbb{E}\left[\boldsymbol{\nabla}_{i,t}\middle|\mathcal{F}_{t-1}\right] = \nabla_{i,t} \qquad \|\boldsymbol{\nabla}_{i,t}\|_* \leq L, \tag{7}$$

where $\mathcal{F}_t$ is the σ-field containing all information prior to the outset of round $t+1$. A commonly used model to generate stochastic gradients satisfying (7) is an additive, bounded, zero-mean noise. Agents then track the moving target using a decentralized variant of Mirror Descent as follows[4]

$$\hat{\mathbf{x}}_{i,t+1} = \text{argmin}_{x\in\mathcal{X}}\left\{\eta_t \langle x, \boldsymbol{\nabla}_{i,t}\rangle + \mathcal{D}_{\mathcal{R}}(x, \mathbf{y}_{i,t})\right\}, \tag{8a}$$

$$\mathbf{x}_{i,t} = A\hat{\mathbf{x}}_{i,t}, \qquad \text{and} \qquad \mathbf{y}_{i,t} = \sum_{j=1}^{n} [W]_{ij}\mathbf{x}_{j,t}, \tag{8b}$$

where $\{\eta_t\}_{t=1}^{T}$ is the step-size sequence, and $A \in \mathbb{R}^{d\times d}$ is the given dynamics in (1) which is *common* knowledge. In these updates, $\mathbf{x}_{i,t} \in \mathbb{R}^d$ represents the estimate of agent i of the moving target $x_t^\star$ at time t. The step-size sequence should be tuned for different cases, but it is generally non-increasing and positive.

The update (8a) allows an agent to follow the noisy local gradient while keeping the estimate close to those of the local neighborhood. This closeness occurs by minimizing the Bregman divergence. On the other hand, the first update in (8b) takes into account the dynamics of the moving target, and the second update in (8b) is the *consensus* term averaging the estimates in the local neighborhood.

III. Theoretical Results

In this section, we state our theoretical result on the *non-asymptotic* performance of the decentralized online mirror descent for tracking dynamic parameters. Theorem 1 proves a bound on the dynamic regret, which captures the deviation of the moving target from the dynamics A (tracking error), the decentralization cost (network error), and the impact of stochastic gradients (stochastic error). We show that this theorem recovers previous rates on decentralized optimization once the tracking error is removed. Also, it recovers previous rates on centralized online optimization in dynamic setting when the network error is eliminated. The proof can be found in the Appendix of [26].

Theorem 1: Consider a moving target $x_t^\star \in \mathbb{R}^d$ with the dynamical model of (1). *Further consider the distributed, online tracking problem formulated in* (4), *where $\mathbf{x}_{i,t}$ denotes the local estimate of agent $i \in [n]$ of the moving target $x_t^\star$ at time $t \in [T]$. Let the local estimates be generated by updates* (8a)-(8b), *where the stochastic gradients satisfy the condition* (7). *Given Assumptions [1-5], the dynamic regret can be bounded as*

$$\boldsymbol{Reg}_T^d \leq \mathrm{E}_{\mathtt{Track}} + \mathrm{E}_{\mathtt{Net}} + \mathrm{E}_{\mathtt{Stoch}},$$

[2] The setup is generalizable to when network connectivity changes over time, and the communication matrix is time-varying.

[3] For simplicity, we use one constant L to bound gradients as well as the stochastic gradients.

[4] We set $\mathbf{x}_{i,t}$ to be the vector of all zeros to initialize the algorithm. In general, any initialization could work for the algorithm.

with probability at least $1-\delta$, *where*

$$\mathtt{E_{Track}} := \frac{2R^2}{\eta_{T+1}} + \sum_{t=1}^{T} \frac{K}{\eta_{t+1}} \|x^\star_{t+1} - Ax^\star_t\| + L^2 \sum_{t=1}^{T} \frac{\eta_t}{2}$$

$$\mathtt{E_{Net}} := 4L^2\sqrt{n}\sum_{t=1}^{T}\sum_{\tau=0}^{t-1} \eta_\tau \sigma_2^{t-\tau-1}(W)$$

$$\mathtt{E_{Stoch}} := 8LR\sqrt{-T\log\delta},$$

and $R^2 := \sup_{x,y\in\mathcal{X}} \mathcal{D}_{\mathcal{R}}(x,y)$.

In view of (1), the dynamical model of the target is described with the noise v_t. The term $\mathtt{E_{Track}}$ shows the dependence of performance bound to noise by aggregating the errors $\|x^\star_{t+1} - Ax^\star_t\| = \|v_t\|$ over time. Also, $\mathtt{E_{Net}}$ and $\mathtt{E_{Stoch}}$ are the errors related to network and stochastic gradients, respectively.

In Section II, we discussed that our setup generalizes some of the previous results. It is now important to see that this generalization is valid in the sense that our result can recover those special cases:

▷ When the global loss $f_t(x) = f(x)$ is time-invariant, the target $\{x^\star_t\}_{t=1}^T$ is fixed, i.e., the dynamics $A = I_d$ and $v_t = \mathbb{0}$ in (1). In this case in Theorem 1, the term involving $\|x^\star_{t+1} - Ax^\star_t\|$ in $\mathtt{E_{Track}}$ is equal to zero, and we can use the step-size sequence $\eta = \sqrt{(1-\sigma_2(W))/T}$ to recover the result of comparable algorithms, such as Theorem 4 in [17] on distributed dual averaging.

▷ The same argument holds when the global loss is time-variant, but the target is fixed. This setup is studied, for instance, in [18] via distributed online dual averaging with exact gradients. Disregarding $\mathtt{E_{Stoch}}$ in our bound due to stochastic gradients, since $\|x^\star_{t+1} - Ax^\star_t\| = 0$ again, we recover Corollary 3 in [18].

▷ When the graph is complete, $\sigma_2(W) = 0$ and hence $\mathtt{E_{Net}} = 0$. We then recover the results of [9] on centralized online learning (for linear dynamics) with exact gradients once we remove $\mathtt{E_{Stoch}}$ due to stochastic gradients.

IV. Numerical Experiment: Tracking Maneuvering Targets

In Mirror Descent algorithm, one has freedom over the selection of the Bregman divergence. A particularly well-known type of Bregman is the Euclidean distance, commonly used in state estimation and tracking dynamic parameters. We focus on this scenario in this section to provide the numerical experiments for our method.

We consider a slowly maneuvering target in the $2D$ plane and assume that each position component of the target evolves independently according to a near constant velocity model [27]. The state of the target at each time consists of four components: horizontal position, vertical position, horizontal velocity, and vertical velocity. We represent the state at time t by $x^\star_t \in \mathbb{R}^4$, and therefore, the state space model takes the form

$$x^\star_{t+1} = Ax^\star_t + v_t,$$

where $v_t \in \mathbb{R}^4$ is the system noise, and using $\otimes$ for Kronecker product, A can be written as

$$A = I_2 \otimes \begin{bmatrix} 1 & \epsilon \\ 0 & 1 \end{bmatrix},$$

with ϵ being the sampling interval[5]. The goal is to cooperatively track $x^\star_t$ in a network of agents. This problem has been studied in the context of distributed Kalman filtering [13], [28], state estimation [29]–[31], and particle filtering [32]–[34]. However, in contrast to Kalman filtering, we do not assume that the system noise v_t is Gaussian. Also, as opposed to particle filtering, we do not receive a large number of samples (particles) per iteration since our setup is online, i.e., agents only observe one sample per time. Furthermore, we do not assume a statistical distribution on v_t in our analysis, which differentiates our framework from state estimation. We adopt a model-free approach where the noise can be adversarial (deterministic), stochastic with dependence over time, or of some complex structure. We generate the noise as follows. At each time t we draw a sample $\nu_t \in \mathbb{R}^4$ from a zero-mean Gaussian distribution with covariance matrix Σ as follows

$$\Sigma = \sigma_\nu^2 I_2 \otimes \begin{bmatrix} \epsilon^3/3 & \epsilon^2/2 \\ \epsilon^2/2 & \epsilon \end{bmatrix},$$

for the sampling interval $\epsilon = 0.1$ seconds which amounts to frequency $10\ Hz$. Then, we let the system noise be $v_t = \nu_t \|\nu_t\|_\infty$. Though ν_t is generated from Gaussian distribution, the mismatch noise v_t is non-Gaussian and can have a complicated distribution. The constant σ_ν^2 takes different values in each experiment, and we describe this choice later.

We consider a sensor network of $n = 25$ agents located on a 5×5 grid. Agents aim to track the moving target $x^\star_t$ collaboratively. Agents observe a noisy version of the target through a *local* loss function, and these observations are nonlinear. In particular, let the quantity $\mathbf{z}_{i,t}$ be a noisy version of one coordinate of $x^\star_t$ as follows

$$\mathbf{z}_{i,t} = \mathbb{e}_{k_i}^\top x^\star_t + \mathbf{w}_{i,t},$$

where $\mathbf{w}_{i,t} \in \mathbb{R}$ denotes a random noise, and $\mathbb{e}_k$ is the k-th unit vector in the standard basis of $\mathbb{R}^4$ for $k \in \{1,2,3,4\}$. We partition the agents into four groups, and for each group we select one specific k_i from the set $\{1,2,3,4\}$. The random noise $\mathbf{w}_{i,t}$ satisfies the standard assumption of being zero-mean and finite-variance. Again, to show that our results are not dependent on Gaussian noise, we generate $\mathbf{w}_{i,t}$ independently from a uniform distribution on $[-1,1]$.

Then, at time t the *local* loss for agent i takes the form

$$f_{i,t}(x) := \frac{1}{4}\mathbb{E}\left[\left(\mathbf{z}_{i,t} - \mathbb{e}_{k_i}^\top x\right)^4 |\mathcal{F}_{t-1}, x^\star_t\right],$$

resulting in the *global* loss

$$f_t(x) := \frac{1}{4n}\sum_{i=1}^{n}\mathbb{E}\left[\left(\mathbf{z}_{i,t} - \mathbb{e}_{k_i}^\top x\right)^4 |\mathcal{F}_{t-1}, x^\star_t\right],$$

where $\mathcal{F}_{t-1}$ is the σ-field containing all information in $\{\mathbf{w}_{i,s}\}_{s=1}^{t-1}$. It is straightforward to see that $x^\star_t$ is the minimizer of the global loss. Observation of agent i at time t is the stochastic gradient of the local loss

$$\nabla f_{i,t}(x) = \left(\mathbf{z}_{i,t} - \mathbb{e}_{k_i}^\top x\right)^3 \mathbb{e}_{k_i}.$$

We derive an explicit update to form an estimate $\mathbf{x}_{i,t}$ of $x^\star_t$. We use Euclidean distance as the Bregman divergence in updates (8a)-(8b) to get[6]

$$\mathbf{x}_{i,t} = \sum_{j=1}^{n}[W]_{ij}A\mathbf{x}_{j,t-1} + \eta_t A\mathbb{e}_{k_i}\left(\mathbf{z}_{i,t-1} - \mathbb{e}_{k_i}^\top \mathbf{x}_{i,t-1}\right)^3,$$

[5] The sampling interval of ϵ (seconds) is equivalent to the sampling rate of $1/\epsilon$ (Hz).

[6] We assume that the state of the target remains in a convex, compact set, and the updates can keep the estimate in the set without the projection step. This assumption can be satisfied in the finite-time domain.

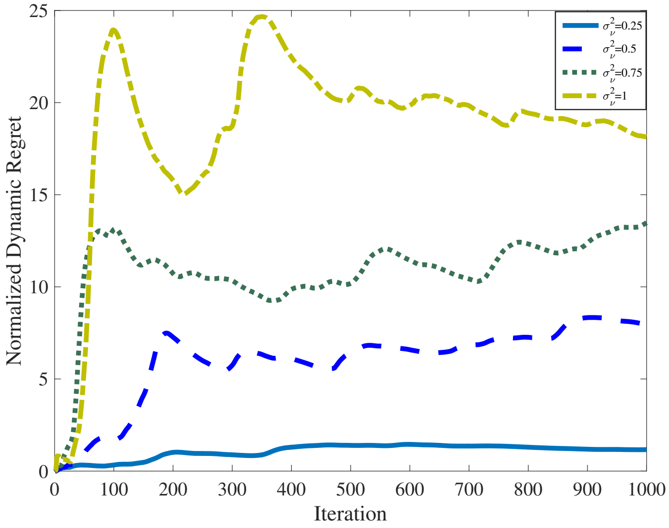

Fig. 1. The plot of dynamic regret versus iterations. Naturally, when σ_v^2 is smaller, the innovation noise added to the dynamics is smaller with high probability, and the network incurs a lower dynamic regret. In this plot, the dynamic regret is normalized by iterations, so the y-axis is $\boldsymbol{Reg}_T^d/T$.

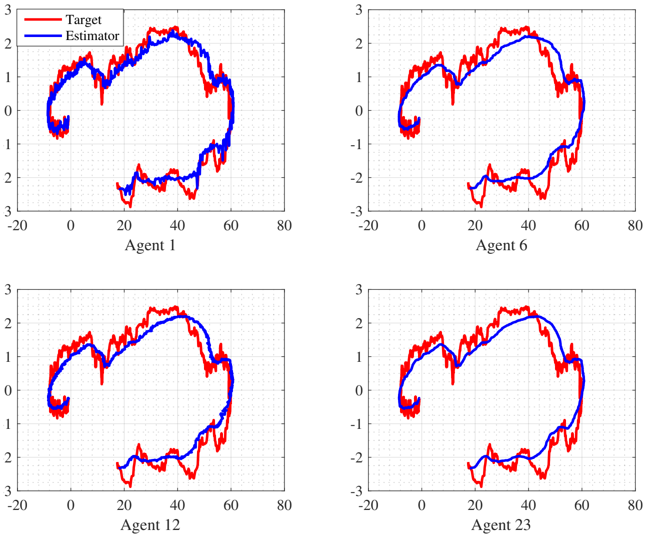

Fig. 2. The trajectory of $x_t^\star$ over $T = 1000$ iterations is shown in red. We also depict the trajectory of the estimator $\mathbf{x}_{i,t}$ (shown in blue) for $i \in \{1, 6, 12, 23\}$ and observe that it closely follows $x_t^\star$ in every case.

and tune the step size to $\eta_t = \eta = 0.1$. The update is akin to *consensus+innovation* updates in the literature (see e.g. [2], [7]) though we recall that the observation is nonlinear, and the system noise v_t is arbitrary.

It is proved in [7] that in decentralized tracking, the dynamic regret can be presented in terms of the tracking error $\mathbf{x}_{i,t} - x_t^\star$ when the local losses are quadratic. More specifically, the expected dynamic regret averages the tracking error over space and time (when normalized by T). While here we deal with polynomial loss of power four, the connection between tracking error and dynamic regret still holds true. Therefore, using the result of Theorem 1 we can expect that once the parameter does not deviate too much from the dynamics, i.e., when $\sum_{t=1}^T \|v_t\|$ is small, the bound on the dynamic regret as well as the collective tracking error is small.

We show this intuitive idea by setting σ_ν^2 to different vaues. Larger values for σ_ν^2 are expected to cause more deviations from the dynamics A and larger dynamic regret (worse performance). In Fig. 1, we plot the normalized dynamic regret for $\sigma_\nu^2 \in \{0.25, 0.5, 0.75, 1\}$. Note that for each value of σ_ν^2, we run the experiment only once to investigate the high probability bound in Theorem 1. As expected, the performance improves once σ_v^2 tends to smaller values.

We next restrict our attention to the case that $\sigma_v^2 = 0.5$. For one run of this case, we provide a snapshot of the target trajectory (in red) in Fig. 2 and plot the estimator trajectory (in blue) for agents $i \in \{1, 6, 12, 23\}$. Fig. 2 suggests that agents' estimators closely follow the trajectory of the moving target with high probability.

V. Conclusion

In this paper, we addressed tracking of a moving target in a network of agents. The target follows a linear dynamics which is common knowledge to agents, but it deviates from this dynamics due to an additive noise of an unknown structure. We formulated the problem as an online optimization of a global time-varying loss in a distributed fashion. The global loss at each time is a sum of a finite number of local losses, and each agent in the network holds a private copy of one local loss. Agents are unaware of the future loss functions as the local losses only become available to them sequentially. They exchange noisy local gradients with each other to track the value of the target.

Our proposed algorithm for this setup can be cast as a decentralized version of Mirror Descent. We however incorporated two more steps to include agents interactions and dynamics of the target. We used a notion of network dynamic regret to measure the performance of our algorithm versus its offline counterpart. We established that the regret bound scales inversely in the spectral gap of the network and captures the deviation of the target with respect to the dynamics. Our results generalized a number of results in online and offline distributed optimization. Also, numerical experiments verified the applicability of our algorithm to multi-agent tracking with nonlinear observations. Future directions include studying the algorithm in the case that several observations are available per round, i.e., when agents can receive multiple noisy gradients per time. The method can be useful in the sensor networks where each sensor can have multiple measurements from different sources.

References

[1] F. Bullo, J. Cortes, and S. Martinez, *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[2] S. Kar, J. M. Moura, and K. Ramanan, "Distributed parameter estimation in sensor networks: Nonlinear observation models and imperfect communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3575–3605, 2012.

[3] S. Shahrampour, S. Rakhlin, and A. Jadbabaie, "Online learning of dynamic parameters in social networks," in *Advances in Neural Information Processing Systems*, 2013.

[4] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-time analysis and impact of network topology," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3256–3268, 2016.

[5] A. Nedić, A. Olshevsky, and C. A. Uribe, "Network independent rates in distributed learning," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 1072–1077.

[6] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[7] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed estimation of dynamic parameters: Regret analysis," in *American Control Conference (ACC)*, July 2016, pp. 1066–1071.

[8] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," *International Conference on Machine Learning (ICML)*, 2003.

[9] E. C. Hall and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, 2015.

[10] O. Besbes, Y. Gur, and A. Zeevi, "Non-stationary stochastic optimization," *Operations Research*, vol. 63, no. 5, pp. 1227–1244, 2015.

[11] A. Jadbabaie, A. Rakhlin, S. Shahrampour, and K. Sridharan, "Online optimization: Competing with dynamic comparators," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015, pp. 398–406.

[12] A. Mokhtari, S. Shahrampour, A. Jadbabaie, and A. Ribeiro, "Online optimization in dynamic environments: Improved regret rates for strongly convex problems," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2016, pp. 7195–7201.

[13] R. Olfati-Saber, "Distributed kalman filtering for sensor networks," in *IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.

[14] D. Yudin and A. Nemirovskii, "Problem complexity and method efficiency in optimization," 1983.

[15] J. Li, G. Chen, Z. Dong, and Z. Wu, "Distributed mirror descent method for multi-agent optimization with delay," *Neurocomputing*, vol. 177, pp. 643–650, 2016.

[16] M. Rabbat, "Multi-agent mirror descent for decentralized stochastic optimization," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop on*. IEEE, 2015, pp. 517–520.

[17] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, 2012.

[18] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed optimization via dual averaging," in *IEEE Conference on Decision and Control (CDC)*, 2013, pp. 1484–1489.

[19] D. Mateos-Núnez and J. Cortés, "Distributed online convex optimization over jointly connected digraphs," *IEEE Transactions on Network Science and Engineering*, vol. 1, no. 1, pp. 23–37, 2014.

[20] A. Nedić, S. Lee, and M. Raginsky, "Decentralized online optimization with global objectives and local communication," in *IEEE American Control Conference (ACC)*, 2015, pp. 4497–4503.

[21] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Transactions on Control of Network Systems*, 2015.

[22] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *arXiv preprint arXiv:1609.02845*, 2016.

[23] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2011.

[24] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[25] H. H. Bauschke and J. M. Borwein, "Joint and separate convexity of the bregman distance," *Studies in Computational Mathematics*, vol. 8, pp. 23–36, 2001.

[26] S. Shahrampour and A. Jadbabaie, "An online optimization approach for multi-agent tracking of dynamic parameters in the presence of adversarial noise," *arXiv preprint arXiv:1702.06219*, 2017.

[27] Y. Bar-Shalom, *Tracking and data association*. Academic Press Professional, Inc., 1987.

[28] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed kalman filtering and smoothing," *IEEE Transactions on automatic control*, vol. 55, no. 9, pp. 2069–2084, 2010.

[29] U. Khan, S. Kar, A. Jadbabaie, J. M. Moura *et al.*, "On connectivity, observability, and stability in distributed estimation," in *IEEE Conference on Decision and Control (CDC)*, 2010, pp. 6639–6644.

[30] S. Das and J. M. Moura, "Distributed state estimation in multi-agent networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 4246–4250.

[31] D. Han, Y. Mo, J. Wu, S. Weerakkody, B. Sinopoli, and L. Shi, "Stochastic event-triggered sensor schedule for remote state estimation," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2661–2675, 2015.

[32] D. Gu, "Distributed particle filter for target tracking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3856–3861.

[33] O. Hlinka, O. Sluciak, F. Hlawatsch, P. M. Djuric, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4334–4349, 2012.

[34] J. Li and A. Nehorai, "Distributed particle filtering via optimal fusion of gaussian mixtures," in *Information Fusion (Fusion), 2015 18th International Conference on*. IEEE, 2015, pp. 1182–1189.