# Online Distributed Estimation via Adaptive Sensor Networks

Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi

***Abstract*—The paper presents an online distributed estimation scheme over adaptive sensor networks. The objective of the algorithm is consistent with distributed least squares without prior assumptions on the uncertainties in the operating environment of the sensors or the quality of sensor observations. Specifically, it is assumed that the observation process is time-varying due to the sensor's susceptibility to unknown errors. Furthermore, there is no probabilistic assumption made on the additive measurement noise. Inspired by recent advances in distributed convex optimization, we propose an online distributed algorithm based on a dual subgradient averaging for the solution of the corresponding estimation problem. Moreover, we examine the situation where the algorithm adapts the weights of the communication links in the network due to uncertainty on the reliability of neighboring sensors. An upper bound on the regret of the algorithm as a function of the underlying network topology is then discussed, followed by simulation results for a few representative classes of sensor networks.**

***Index Terms*—Sensor networks; distributed estimation; network topology design; online optimization**

## I. INTRODUCTION

The past decade has seen successful applications of sensor networks in many practical areas ranging from environmental monitoring, robotics, target recognition, air traffic control, to industrial and manufacturing automation. Based on data fusion techniques, the data from multiple sensors are integrated to achieve an accurate and consistent representation of an unknown state compared with the estimate obtained via a single, independent sensor. By increasing the size and complexity of sensor networks, decentralized estimation schemes are desired for reducing data transmission rates and ensuring robustness in the presence of local failures. In recent years, the area of distributed estimation and filtering in sensor networks has received extensive attention [1], [2], [3], [4], [5] and the adaptation and monitoring of their underlying networks is of increasing interest [6], [7], [8]. These estimators are particularly relevant when there is a lack of access to centralized information by each individual sensor. In this direction, a number of algorithms based on distributed least squares have been developed based on consensus [9], [5], [10], [11] and diffusion [12], [13] strategies. The original implementations of consensus in distributed estimators [14] require two time scales; each sensor performs a local estimation and then through consensus iterations the sensors agree on the desired estimation. In the diffusion-based distributed estimation, information is processed locally at the nodes and then diffused in real-time across the network. Since sensors are susceptible to errors, the observation process can be time varying. As compared with consensus-based distributed algorithms, Tu and Sayed [15] have shown that diffusion-based distributed estimators generally have superior performance. In diffusion approach, the errors are generally modeled as white noise with a time varying covariance [16], [17], [12], [18], [19] and the distributed algorithms are examined from a mean-square-error perspective. In [20], Zhou *et al.* have assumed that the observation mode switches stochastically between two given modes while estimating a scalar signal in the presence of Gaussian white noise for the measurement error. However, in situations where the observation process can change in an unpredictable manner in a dynamic uncertain environment and no meaningful assumption on the statistical properties of data is available a priori, a more robust strategy is deemed necessary.

Given that the least squares problem is convex, it is natural that distributed convex optimization can effectively be used to derive distributed estimation algorithms. There has grown an extensive literature on distributed convex optimization recently. A class of such distributed algorithms are based on the subgradient method [21], [22], [23], [24]. In these works a convex cost function is assumed to be known while the topology of network is allowed to vary. However, uncertainties in the environment and measurement scheme can in fact affect the underlying convex cost functions, making the optimization approach unsuitable for this class of problems. One approach to improve the robustness of algorithms for convex optimization is via stochastic methods [25], [26], [27], where the probability distribution of uncertain variable is known a priori. One such approach has been pursued by Duchi *et al.* [24] who solved this problem using a stochastic subgradient method where the distribution of subgradients are known a priori.

Despite its many successes, stochastic optimization-based methods do not explicitly address the dynamic aspect of the estimation problem in an uncertain dynamic environment. Online learning is an extension of stochastic optimization where the uncertainty in the system is demonstrated by an arbitrarily varying cost function. In particular, the cost function is assumed to be unknown to the decision-maker, even without a probabilistic assumption, at the time the relevant decision is made. Such learning algorithms have had a significant impact on modern machine learning [28], [29], [30]. One standard metric to measure the performance

A preliminary version of this work will appear in the *IEEE Conference on Decision and Control, 2013.*

The research of the authors was supported by the ONR grant N00014-12-1-1002 and AFOSR grant FA9550-12-1-0203-DEF. The authors are with the Department of Aeronautics and Astronautics, University of Washington, WA 98105. Emails: {saghar, airliec, mesbahi}@uw.edu.

of these online algorithms is called *regret*. Regret measures the difference between the incurred cost and the cost of the *best fixed decision in hindsight*. An online algorithm is then declared "good" when, on average, its regret approaches zero.

Distributed online optimization and its applications in multi-agent systems has not been studied at large by the systems and control community. Yan *et al.* in [31] introduced a decentralized online optimization based on a subgradient method in which the agents are interacting over a weighted strongly connected directed graph. Considering an undirected path graph with a fixed-radius neighborhood information structure, Raginsky *et al.* [32] proposed an online algorithm for distributed optimization based on sequential updates, proving a regret bound of $O(\sqrt{T})$. In [33], we proposed an extension to the work of Duchi *et al.* [24] on distributed optimization with convergence rate of $O(\sqrt{T}\log T)$ to an online setting. In addition, an improved regret bound of $O(\sqrt{T})$ , which also highlighted the regret's dependence on the connectivity of the underlying network, was derived for strongly connected networks. We note that the aforementioned works did not exploit adaptive network topologies to improve the performance of the corresponding distributed algorithm. In the existing systems and control literature, certain metrics for networked systems have been used for designing adaptive mechanisms for networks based on the centralized [34], [35], [36] and distributed [37], [38] strategies. Network topology design is also favorable in the area of sensor networks and distributed estimation since the communication links among sensors are subject to failures, as well as power and data rate constraints [6], [7], [8]. Kar and Moura [7] have looked at a Bernoulli random network to improve the convergence rate of consensus based distributed estimation. In addition, Chapman *et al.* [39] proposed an online distributed algorithm for re-weighting the network edges in order to dampen the effect of external disturbances on the system.

In this paper, an online adaptive algorithm for distributed estimation over sensor networks operating in an uncertain environment is proposed. The main assumption for this online algorithm is that the local estimation error and its subgradient is observable at each sensor, and can be shared with the neighboring sensors in the network. Specifically, the contribution of this paper is two fold. First, we discuss the applicability and implications of the Online Distributed Dual Averaging (ODD) algorithm [33] for distributed estimation over sensor networks. The proposed algorithm is inspired by the Distributed Dual Averaging (DDA) algorithm, discussed in [24], that is extended to an online setting capturing the uncertainties in sensor measurements and unavailability of reliable statistics on the noise characteristics. Second, an online distributed topology design method based on the online weighted majority approach [40], [41] is embedded in the ODD algorithm allowing the weights on the network's edges to adaptively change in order to optimize the information diffusion in the network. We then proceed to derive regret bounds that highlight the link between the adaptive network topology and the ODD algorithm and can thus be used to design networks with good regret performance.

The organization of the paper is as follows. The notation and background on graphs and regret are reviewed in §II. In §III, the distributed estimation problem over sensor networks operating in an uncertain environment is formulated as an online distributed convex optimization problem. This is then followed by the description for the ODD algorithm and online topology design in §IV where the convergence analysis of the proposed algorithm is then discussed. In §V we present simulation results for some representative distributed sensor networks, demonstrating the viability of the online approach in distributed estimation. Finally, §VI provides our concluding remarks and future directions for utilizing the online framework for system and control problems.

## II. Background and Preliminaries

We provide a brief background on constructs that will be used in this paper. For the column vector $v \in \mathbb{R}^p$, $v_i$ or $[v]_i$ denotes the $i$th element, $e_i$ denotes the column vector which contains all zero entries except $[e_i]_i = 1$. The vector of all ones will be denoted by $\mathbf{1}$. For matrix $M \in \mathbb{R}^{p\times q}$, $[M]_{ij}$ denotes the element in its $i$th row and $j$th column. A row stochastic matrix $P$ is a non-negative matrix where $\sum_{j=1}^{n} P_{ij} = 1$ for all $i$. A time varying matrix $P(t)$ is also presented by $P^t$ and a sequence of time varying matrices is presented by $P^{(t,0)} = P^t P^{t-1} \cdots P^0$. For any positive integer $n$, the set $\{1, 2, ..., n\}$ is denoted by $[n]$. The inner product of two vectors $\theta$ and $\phi$ is represented by $\langle\theta, \phi\rangle$. The 2-norm is signified by $||.||_2$ and the dual norm of a vector $\theta$ is defined as $||\theta||_* = \sup_{||\phi||=1} \langle\theta, \phi\rangle$. A function $f : \Theta \to R$ is called $L$-Lipschitz continuous if there exists a positive constant $L$ for which

$$|f(\theta) - f(\phi)| \leq L\|\theta - \phi\| \text{ for all } \theta, \phi \in \Theta. \qquad (1)$$

### A. Graphs

A succinct way to represent the interactions of dynamic agents, e.g., sensors, over a network is through a graph. A weighted graph $\mathcal{G} = (V, E, W)$ is defined by a node set $V$ with cardinality $n$, the number of nodes in the graph which represent the agents in the network, and an edge set $E$ comprising of pairs of nodes which represent the agents interactions, i.e., agent $i$ affects agent $j$'s dynamics if there is an edge from $i$ to $j$, i.e., $(i, j) \in E$. In addition, a function $W : E \to R$ is given that associates a weight $w_{ji} \in W$ to every edge $(i, j) \in E$. The adjacency matrix $A(\mathcal{G})$ is a matrix representation of $\mathcal{G}$ with $[A(\mathcal{G})]_{ji} = w_{ji}$ for $(i, j) \in E$ and $[A(\mathcal{G})]_{ji} = 0$, otherwise. A graph $\mathcal{G}$ is strongly connected if between every pair of distinct vertices there exists a directed path. For a graph $\mathcal{G}$, $d_i$ is the weighted in-degree of $i$, defined as $d_i = \sum_{\{j|(j,i)\in E\}} w_{ij}$. In addition, $L(\mathcal{G}) = \Delta(\mathcal{G}) - A(\mathcal{G})$ is called the graph Laplacian where $\Delta(\mathcal{G})$ is the diagonal matrix of $d_i$'s. Based on the construction of weighted directed graph Laplacian, every graph $\mathcal{G}$ has a right eigenvector of $\mathbf{1}$ associated with eigenvalue $\lambda = 0$ [10].

There are many families of graphs that are often used to model networks of practical interest. In this paper, we use path graph and random graphs for some of our simulations. Specifically, Erdős-Rényi random graphs with edge probability $p$ are constructed by having an edge $(i, j) \in E$ in the graph with probability $p$ for all possible edges. A random tree is a particular realization of a random graph that is minimally connected and a random $k$-regular graph is a random graph in which $d_i = k$ for all vertices $i \in V$. In the path graph, edge $(i, j) \in E$ if and only if $|i - j| = 1$.

### B. Regret

Regret is one measure of the performance of learning algorithms. In the online optimization setting, an algorithm is used to generate a sequence of decisions $\{\hat{\theta}(t)\}_{t=1}^{T}$. The number of iterations is denoted by $T$ which is unknown to the online player. At iteration $t$, after committing to $\hat{\theta}(t)$, a previously unknown convex cost function $f_t$ is revealed, and a loss $f_t(\hat{\theta}(t))$ is incurred. The goal of the online algorithm is to ensure that the time average of the difference between the total cost and the cost of the best fixed single decision $\theta^*$ is small. The difference between these two costs over $t = 1, 2, ..., T$, iterations is called the regret of the online algorithm, i.e.,

$$R_T = \sum_{t=1}^{T} \left( f_t(\hat{\theta}(t)) - f_t(\theta^*) \right). \quad (2)$$

An algorithm performs well if its regret is sublinear as a function of $T$, i.e. $\lim_{T\to\infty} R_T/T = 0$. This implies that on average, the algorithm performs as well as the best fixed strategy in hindsight independent of the adversary's moves and environmental uncertainties. Further discussion on online algorithms and their regret can be found in [42], [43], [44].

The general definition of regret is presented in (2) for a single decision-making unit. In order to to analyze the performance of *distributed online algorithms* two variations of the notion of regret are introduced. First is the regret due to sensor $i$'s estimation

$$R_T(\theta^*, \hat{\theta}_i) = \sum_{t=1}^{T} \left( f_t(\hat{\theta}_i(t)) - f_t(\theta^*) \right), \quad (3)$$

which is the cumulative penalty sensor $i$ pays because of its decisions on the global cost sequence $\{f_t\}$. Second is the regret based on the *running average* of the estimates $\{\hat{\theta}_i(t)\}_{t=1}^{T}$,

$$R_T(\theta^*, \tilde{\theta}_i) = \sum_{t=1}^{T} \left( f_t(\tilde{\theta}_i(t)) - f_t(\theta^*) \right), \quad (4)$$

where $\tilde{\theta}_i(T) = \frac{1}{T} \sum_{t=1}^{T} \hat{\theta}_i(t)$.

## III. Problem Statement

Adopting a least squared point of view, a model for estimation over a distributed sensor network is presented in this section. The distributed sensor network aims to estimate a random vector $\theta \in \Theta = \left\{\theta \in \mathbb{R}^d | \, \|\theta\|_2 \le \theta_{\max}\right\}$. Note that $\Theta$ is a closed convex set containing the origin. The observation vector $z_{i,t} : \mathbb{R}^d \to \mathbb{R}^{p_i}$ represents the $i$th sensor measurement at time $t$ which is uncertain and time-varying due to the sensor's susceptibility to unknown environmental factors such as jamming. The sensor is assumed (not necessarily accurately) to have a linear model of the form $h_i(\theta) = H_i\theta$ where $H_i \in \mathbb{R}^{p_i \times d}$ is the observation matrix of sensor $i$ and $\|H_i\|_1 \le h_{\max}$ for all sensors $i$. Consider now the interconnection topology between the sensors defined via a graph $\mathcal{G} = (V, E, W)$, where the set of $n$ sensors are represented by $V$. The presence of an edge $(j, i) \in E$ indicates an information flow from sensor $j$ to sensor $i$. The set of agents that are communicating with agent $i$ is defined as the neighborhood set $N(i) = \{j \in V | (j, i) \in E\}$. Figure 1 graphically summarizes the problem setup. The objective is to find the argument $\hat{\theta}$ that minimizes the cost function

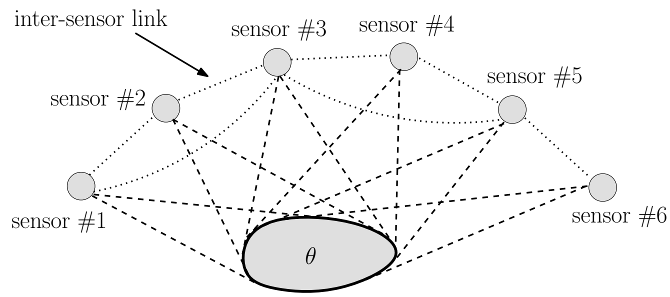


Figure 1. A graphical representation of a distributed sensor network.

$$f_t(\hat{\theta}) = \frac{1}{n} \sum_{i=1}^{n} f_{i,t}(\hat{\theta}) \quad \text{subject to } \hat{\theta} \in \Theta, \quad (5)$$

where

$$f_{i,t}(\hat{\theta}) = \frac{1}{2} \left\| z_{i,t} - H_i\hat{\theta} \right\|_2^2 \quad (6)$$

is a convex cost function associated with sensor $i \in [n]$. It is assumed that the value of this local cost at time $t$ is only revealed to the sensor after $\hat{\theta}(t)$ has been computed, that is, the local error functions are allowed to change over time in an unpredictable manner due to modeling errors and uncertainties in the environment. The (sub)gradient of the local estimation error (6),

$$\partial f_{i,t}(\hat{\theta}) = H_i^T \left( z_{i,t}(\theta) - H_i\hat{\theta} \right), \quad (7)$$

is also assumed to be known to the sensor and its neighbors. We note that the cumulative cost at time $T$ is defined as $f(\hat{\theta}) = \sum_{t=1}^{T} f_t(\hat{\theta})$.

In an offline setting, for all $t \in [T]$, each sensor $i$ has a noisy observation $z_{i,t} = H_i\theta + v_{i,t}$, where $v_{i,t}$ is generally assumed to be (independent) white noise. In this case, the centralized time-averaged optimal estimate for (5) is

$$\theta^* = \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{i=1}^{n} H_i^T \Sigma_{i,t}^{-1} H_i \right)^{-1} \left( \sum_{i=1}^{n} H_i^T \Sigma_{i,t}^{-1} z_{i,t} \right), \quad (8)$$

where $\Sigma_{i,t}$ is the covariance of the error observed by sensor $i$ at time $t$ [14]. For the case where $\theta \in \mathbb{R}$, $\Sigma_{i,t} = I$, and $H_i = 1$, the optimal estimate is $\theta^* = \frac{1}{nT} \sum_{i=1}^{n} \sum_{t=1}^{T} z_{i,t}$. However, this approach to estimation problems is not suitable in scenarios where the noise characteristics are not known. For example when a wireless sensor network is employed in an unknown and dynamic environment, the measurement signal can be blocked or diminished due to obstructions such as walls, furniture, trees, or buildings. This is known as the shadowing effect and usually modeled as a function of the environment in which the network is deployed. Another example is jamming of one or more sensors in the network. When the sensor resolution and noise characteristics are not known ahead of time, the online network topology algorithm eliminates the information from jammed sensors. An online framework is particularly suitable for such estimation problems without using an assumption or knowledge of the statistical properties of the data. In the proposed distributed estimation algorithm, at time step $t$, each sensor $i$ estimates $\hat{\theta}_i \in \Theta$ based on the local information available to it and then an "oracle" announces the cost $f_t(\hat{\theta}_i)$.

## IV. Solution Strategy

In order to solve the optimization problem proposed in §III in an online setting, we adapt Nesterov's dual averaging algorithms [45] and our preliminary results on the Online Distributed Dual Averaging (ODD) algorithm [33], that in turn is inspired by [24]. The ODD algorithm sequentially updates the state estimate $\hat{\theta}_i(t)$ and a "working" variable $y_i(t)$ for each agent $i$. The update itself is based on a provided local gradient of the loss $f_{t,i}(\hat{\theta}_i(t))$ denoted as $g_i(t)$. The centralized form of the dual averaging algorithm appears as a gradient decent method followed by a projection step onto the constraint set $\Theta$, specifically,

$$y(t+1) = y(t) + g(t),$$

where $g(t) = \nabla f_t(\hat{\theta}(t))$. Then

$$\hat{\theta}(t+1) = \Pi^{\psi}_{\Theta}\left(y(t+1), \alpha(t)\right),$$

where $\Pi^{\psi}_{\Theta}(\cdot)$ is a regularized projection onto $\Theta$, to be formally defined shortly.

We now consider distributed decision processes on a large number of agents cooperating to optimize a global objective function

$$f_t(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^{n} f_{t,i}(\hat{\theta}) \quad \text{subject to} \;\; \hat{\theta} \in \Theta, \tag{9}$$

where $f_{t,i} : \mathbb{R}^d \to \mathbb{R}$ is a convex cost function associated with agent $i \in [n]$, assumed to be revealed to the agent only after the agent commits to the decision $\hat{\theta}(t)$.

### A. Online Distributed Optimization via Dual Averaging

The Online Distributed Dual Averaging (ODD) algorithm is presented in Algorithm 1. The projection function used in this algorithm is defined as

$$\Pi^{\psi}_{\Theta}(y(t), \alpha(t)) = \arg\min_{\hat{\theta}\in\Theta}\left\{\langle y(t), \hat{\theta}\rangle + \frac{1}{\alpha(t)}\psi(\hat{\theta})\right\}, \tag{10}$$

where $\alpha(t)$ is a non-increasing sequence of positive functions and $\psi(\hat{\theta}) : \Theta \to \mathbb{R}$ is a proximal function. The standard dual averaging algorithm uses proximal function $\psi(\hat{\theta})$ to avoid wide oscillations in the projection step. Without loss of generality, $\psi$ is assumed to be strongly convex with respect to $\|.\|$, $\psi \geq 0$, and $\psi(0) = 0$.

**Algorithm 1:** Online Distributed Dual Averaging (ODD)

```
1 for t = 1 to T do
2     Adversary reveals f_t(t) = {f_{t,i}(t); for all i = 1, ..., n}
3     Compute subgradient g_i(t) ∈ ∂f_{t,i}(θ̂_{t,i})
4     for Each Agent i do
5         y_i(t+1) = Σ_{j∈N(i)} P_{j,i}(t) y_j(t) + g_i(t)
6         θ̂_i(t+1) = Π^ψ_X (y_i(t+1), α(t))
7         θ̃_i(t+1) = 1/(t+1) Σ_{s=1}^{t+1} θ̂_i(s)
8     end
9 end
```

The distributed algorithm can be considered as an approximated gradient descent using $g_1(t)$, $g_2(t), \ldots, g_n(t)$ instead of $g(t)$. The approximation is attained locally by an agent $i$ via a convex combination of information provided by its neighbors. This operation can be represented compactly as a stochastic matrix $P \in \mathbb{R}^{n\times n}$ which preserves the zero structure of the Laplacian matrix $L(\mathcal{G})$. It is clear that for all agents to have access to each cost function $f_{t,i}$ there must be a path from every agent $i$ to every other agent. Consequently, a minimum requirement on the underlying network is that it must be strongly connected. The following section provides a method to construct a row stochastic matrix $P$ of the required form that is associated with an weighted directed graph.

### B. Distributed Network Topology Design via an Online Allocation Algorithm

In this section, we propose an adaptation scheme for the sensor network topology in order to improve the information diffusion in line 5 of Algorithm 1 such that the communication matrix $P$ preserves its zero structure. Moreover, it is assumed that the matrix $P$ is a row stochastic matrix with positive diagonal elements, i.e., $[P]_{ii} > 0$. In this proposed distributed algorithm, each sensor $i \in [n]$ estimates its loss function via a convex combination of loss functions available to it by its neighboring sensors. This convex combination is specified by weights $w_{ij}$'s and $w_{ii}$ associated to each edge $(j,i) \in E$ where $j \in N(i)$ and sensor $i$, respectively. The edge re-weighting problem parallels the Weighted Majority (WM) algorithm [40]. The operational context of the WM algorithm is the presence of $|N(i)|+1$ experts, and at each time-step $t \in [T]$ there is an associated binary cost $f_{t,j}$ assigining a loss value to expert $j \in \{N(i), i\}$. In each time-step, we then pick a probability distribution $\mathbf{q}_t$ over the $|N(i)|+1$ experts in order to minimize $l_{t,i} = \sum_{j\in\{N(i),i\}} q_j f_{t,j}$. The regret for the WM algorithm is then defined as

$$L_T = \sum_{t=1}^{T}\left\{\sum_{j\in\{N(i),i\}} q_j f_{t,j}\right\} - \min_{j\in\{N(i),i\}}\sum_{t=1}^{T} f_{t,j}, \tag{11}$$

where the best fixed strategy is the performance of the single best expert $j \in \{N(i), i\}$ in hindsight.

The general form of the WM algorithm is presented in [41] as the Online Allocation (OA) algorithm that is applicable to any bounded loss function over general decision and outcome spaces. Based on the OA algorithm, the regret for each agent $i \in [n]$ is bounded as

$$L_T \leq M \left( \sqrt{2T \ln \left( |N_i| + 1 \right)} + \ln \left( |N_i| + 1 \right) \right), \qquad (12)$$

where $M$ is the upper bound on the loss function $f_{t,j}$ for all $j \in \{N(i), i\}$. Since the regret in (12) is sub-linear over time, the weight allocation will perform as well as the best strategy in hindsight. A Distributed Online Allocation (DOA) algorithm is proposed based on the OA algorithm where each agent $i \in [n]$ specifies the weights $w_{ij}$'s associated to each edge $(j, i) \in E$ as well as the weight on the self-loop. The DOA algorithm is presented in Algorithm 2 and is embedded in the Algorithm 1 at each iteration.

**Algorithm 2:** Distributed Online Allocation (DOA)

1 Choose $\beta \in [0, 1]$ and initial weight vector $\mathbf{w}_i(t) \in (0, 1]^{|N(i)|+1}$ for all $i \in [n]$, such that $\sum_{j \in \{N(i), i\}} w_{ij}(t) = 1$
2 **for** $t = 1$ *to* $T$ **do**
3     **for** *Each Agent* $i$ **do**
4         $\mathbf{q}(t) = \frac{\mathbf{w}_i(t)}{\sum_{j \in \{N(i), i\}} w_{ij}(t)}$
5         Adversary reveals $f_t(t) = \{f_{t,j}(t);$ for all $j \in \{N(i), i\}\}$
6         Suffer loss $l_{t,i} = \sum_{j \in \{N(i), i\}} q_j(t) f_{t,j}(t)$
7         **for** *Each Agent* $j \in \{N(i), i\}$ **do**
8             $w_{ij}(t+1) = w_{ij}(t) \beta^{f_{t,j}}$
9         **end**
10     **end**
11 **end**

In sensor networks, each agent decides on the weights associated with the information received by its neighboring sensors based on the neighbor local loss function. Intuitively, the algorithm places more weight on the link associated with the neighboring sensor that has a higher confidence in its estimate. The positive diagonal entries represent the self confidence of each sensor and is updated based on the sensor's estimation error or local loss. In addition, the non-diagonal non-zero elements in each row of communication matrix $P$ is specified by line 4 in Algorithm 2:

$$P_{i,j}(t) = \begin{cases} q_j(t) & \text{for } j \in \{N(i), i\} \\ 0 & \text{otherwise} \end{cases}. \qquad (13)$$

Since for each agent $i$, $\mathbf{q}(t)$ is a probability distribution, the communication matrix $P(t)$ will be row stochastic at every time step and the weighted graph Laplacian can then be formed as

$$L(\mathcal{G}(t)) = I - P(t). \qquad (14)$$

Note that since the graph is strongly connected, the communication matrix $P$ is 1-irreducible as stated in Corollary 4 in [46] and given positive diagonal elements, it is indecomposable and aperiodic. These properties of $P(t)$ will be employed in the convergence analysis of the ODD algorithm.

### C. Convergence Analysis

Before proceeding to the convergence analysis of the online distributed algorithm, a few preliminary remarks and definitions are in order. A convex function $f_{t,i}$ on a compact domain is uniformly $L$-Lipschitz with respect to $\|.\|$. In order to take advantage of the properties of the standard weighted dual averaging in our regret analysis, the sequences $\bar{y}(t)$ and $\bar{g}(t)$ are defined as

$$\bar{y}(t) = \sum_{i=1}^{n} \pi_i y_i(t), \quad \bar{g}(t) = \sum_{i=1}^{n} \pi_i g_i(t), \qquad (15)$$

signifying the (network-level) weighted average of dual variables and subgradients in the ODD algorithm, respectively. In addition, the weighting factors $\pi_i \geq 0$ for all $i \in [n]$ comprise a probability distribution [47], implying that

$$\pi_j = \sum_{i=1}^{n} \pi_i P_{ij}^t \text{ for all } t \in [T]. \qquad (16)$$

Therefore, based on (15) and (16),

$$\begin{aligned} \bar{y}(t+1) &= \sum_{i=1}^{n} \pi_i \left\{ \sum_{j=1}^{n} P_{ij}^t y_j(t) + g_i(t) \right\} \\ &= \sum_{j=1}^{n} y_j(t) \sum_{i=1}^{n} \pi_i P_{ij}^t + \bar{g}(t) \\ &= \sum_{j=1}^{n} y_j(t) \pi_j + \bar{g}(t). \end{aligned}$$

Thus, the following update rule is introduced

$$\bar{y}(t+1) = \bar{y}(t) + \bar{g}(t), \qquad (17)$$

which is analogous to the standard dual averaging algorithm where the primal variable is updated as

$$\phi(t+1) = \Pi_{\mathcal{X}}^{\psi}(\bar{y}(t+1), \alpha(t)). \qquad (18)$$

The performance analysis of the online distributed estimation and topology design can now be stated.

**Theorem 1.** *Given the sequence of $\hat{\theta}_i(t)$ and $y_i(t)$ generated by lines* 5 *and* 6 *in* Algorithm 1*, for all $i \in [n]$ with $\psi(\theta^*) \leq R^2$ and $\alpha(t) = k/\sqrt{t}$, we have*

$$R_T(\theta^*, \hat{\theta}_i) \leq \left( \frac{R^2}{k} + kL^2 \left( \frac{6n}{1-\gamma} + 6n\nu + 1 \right) \right) \sqrt{T}, \qquad (19)$$

*where $\gamma$ is a function of the ergodicity of the of communication matrix (see* (38)*) while $\nu$ is a measure of network connectivity and is bounded by the diameter of the graph $\mathcal{G}^t$ (see also Proposition 6). In addition, $k$ is an arbitrary constant.*

*Proof:* Consider an arbitrary fixed decision $\theta^* \in \Theta$ and a sequence $\phi(t)$ generated by (18). From the $L$-Lipschitz continuity of $f_{t,i}$'s and the definition of regret in (3), the regret is bounded as

$$R_T(\theta^*, \hat{\theta}_i) \leq \sum_{t=1}^{T} \left( f_t(\phi(t)) - f_t(\theta^*) + L\|\hat{\theta}_i(t) - \phi(t)\| \right). \tag{20}$$

Note that we can rephrase the first term on the right hand side of (20) as

$$\begin{aligned} f_t(\phi(t)) - f_t(\theta^*) = &\left( \frac{1}{n} \sum_{i=1}^{n} f_{t,i}(\hat{\theta}_i(t)) - f_t(\theta^*) \right) \\ &+ \left( \frac{1}{n} \sum_{i=1}^{n} \left[ f_{t,i}(\phi(t)) - f_{t,i}(\hat{\theta}_i(t)) \right] \right). \end{aligned} \tag{21}$$

In addition, based on the convexity of $f_{t,i}$'s, we have

$$\begin{aligned} &\sum_{t=1}^{T} \left( \frac{1}{n} \sum_{i=1}^{n} f_{t,i}(\hat{\theta}_i(t)) - f_t(\theta^*) \right) \leq \\ &\sum_{t=1}^{T} \left( \frac{1}{n} \sum_{i=1}^{n} \left\langle g_i(t), \hat{\theta}_i(t) - \theta^* \right\rangle \right), \end{aligned} \tag{22}$$

where $g_i(t) \in \partial f_{t,i}(\hat{\theta}_i(t))$ is the subgradient of $f_{t,i}$ at $\hat{\theta}_i(t)$. Thereby, we can express the regret bound based on (21), (22), and the $L$-Lipschitz continuity of $f_{t,i}$'s as:[1]

$$\begin{aligned} R_T(\theta^*, \hat{\theta}_i) \leq &\sum_{t=1}^{T} (\frac{1}{n} \sum_{i=1}^{n} \left\langle g_i(t), \hat{\theta}_i(t) - \theta^* \right\rangle \\ &+ \frac{L}{n} \sum_{i=1}^{n} \|\hat{\theta}_i(t) - \phi(t)\| + L\|\hat{\theta}_i(t) - \phi(t)\|). \end{aligned} \tag{23}$$

The first term on the right had side of (23) can be expanded as

$$\begin{aligned} &\sum_{t=1}^{T} \left( \frac{1}{n} \sum_{i=1}^{n} \left\langle g_i(t), \hat{\theta}_i(t) - \theta^* \right\rangle \right) \\ &= \sum_{t=1}^{T} (\frac{1}{n} \sum_{i=1}^{n} \left\langle g_i(t), \hat{\theta}_i(t) - \phi(t) \right\rangle \\ &+ \frac{1}{n} \sum_{i=1}^{n} \langle g_i(t), \phi(t) - \theta^* \rangle). \end{aligned} \tag{24}$$

Now, we need to bound the terms on the right hand side of (24). The first term is bounded based on the convexity and $L$-Lipschitz conditions on $f_{t,i}$.[2] In other words,

$$\left\langle g_i(t), \hat{\theta}_i(t) - \phi(t) \right\rangle \leq L\|\hat{\theta}_i(t) - \phi(t)\|. \tag{25}$$

[1]Note that the $L$-Lipschitz continuity of $f_{t,i}$'s implies $f_{t,i}(\phi(t)) - f_{t,i}(\hat{\theta}_i(t)) \leq L\|\hat{\theta}_i(t) - \phi(t)\|$.

[2]Note that convexity of $f_{t,i}$ implies $\left\langle g_i(t), \phi(t) - \hat{\theta}_i(t) \right\rangle \leq f_{t,i}(\phi(t)) - f_{t,i}(\hat{\theta}_i(t))$ . Therefore, based on $L$-Lipschitz continuity of $f_{t,i}$'s, we have $||g_i||_* \leq L$ and we can deduce (25).

Since $\hat{\theta}_i(t)$ and $\phi(t)$ are the projections of $y_i(t)$ and $\bar{y}(t)$ respectively, the Lipschitz continuity of $\Pi_\Theta^\psi(., \alpha)$ presented in Lemma 3 imposes a bound on $\|\hat{\theta}_i(t) - \phi(t)\|$ as

$$\|\hat{\theta}_i(t) - \phi(t)\| \leq \alpha(t)\|\bar{y}(t) - y_i(t)\|_*, \tag{26}$$

where $||.||_*$ is the dual norm. Therefore, using the bound in Lemma 4 and noting that $\|g_i(t)\|_* \leq L$, we can write (24) as

$$\begin{aligned} &\sum_{t=1}^{T} \left( \frac{1}{n} \sum_{i=1}^{n} \left\langle g_i(t), \hat{\theta}_i(t) - \theta^* \right\rangle \right) \\ &\leq \frac{L}{n} \sum_{t=1}^{T} \sum_{i=1}^{n} \alpha(t)\|\bar{y}(t) - y_i(t)\|_* \\ &+ \frac{L^2}{2} \sum_{t=2}^{T} \alpha(t-1) + \frac{1}{\alpha(T)} \psi(\theta^*). \end{aligned} \tag{27}$$

Thus, (23), (26), and (27) implies

$$\begin{aligned} R_T(\theta^*, \hat{\theta}_i) \leq &\frac{L^2}{2} \sum_{t=2}^{T} \alpha(t-1) + \frac{1}{\alpha(T)} \psi(\theta^\star) \\ + L \sum_{t=1}^{T} \alpha(t) &\left( \|\bar{y}(t) - y_i(t)\|_* + \frac{2}{n} \sum_{i=1}^{n} \|\bar{y}(t) - y_i(t)\|_* \right). \end{aligned} \tag{28}$$

Lemma 5 imposes an upper bound on the last term on the right hand side of (28). Thus, using (40) the regret is further bounded as

$$\begin{aligned} R_T(\theta^*, \hat{\theta}_i) \leq &\frac{L^2}{2} \sum_{t=1}^{T-1} \alpha(t) + \frac{1}{\alpha(T)} \psi(\theta^\star) \\ &+ 3L^2 \left( \frac{n}{1-\gamma} + n\nu + 2(1-n) \right) \sum_{t=1}^{T} \alpha(t). \end{aligned} \tag{29}$$

The integral test on $\alpha(t) = k/\sqrt{t}$ provides a bound[3] on the first and last terms in (29). Therefore, based on the bound $\psi(\theta^*) \leq R^2$, the statement of the theorem follows. ■

The significance of Theorem 1 is its validation of a "good" performance through sublinear regret as well as highlighting the importance of the underlying network topology through the parameters $\gamma$ and $\nu$. In particular, $\gamma$ is proportional to the the ergodic coefficient $\tau(P^t)$ of the communication matrix $P^t$ as formed by Algorithm 2. The ergodic coefficient bounds the second largest eigenvalue of $P^t$, presented by $\lambda_2(P^t)$ as $|\lambda_2(P^t)| \leq \tau(P^t) < 1$. Thus, based on (14), $1 - \lambda_2(P^t) = \lambda_{n-1}(\mathcal{G}^t)$ where $\lambda_{n-1}(\mathcal{G}^t)$ is the second smallest eigenvalue of the weighted graph Laplacian $L(\mathcal{G}^t)$ and a well known measure of network connectivity. Consequently, high network connectivity promotes good performance of the algorithm.

Next we present the regret analysis for the (temporal) running average estimates at each sensor exhibiting a similar dependence on the parameters of the network connectivity.

[3]Note that $\sum_{t=1}^{T} \frac{k}{\sqrt{t}} \leq 2k\sqrt{T} - k$.

**Corollary 2.** *Given the sequence $\tilde{\theta}_i(t)$ generated by line 7 in Algorithm 1 for all $i \in [n]$ with $\psi(\theta^*) \le R^2$ and $\alpha(t) = k/\sqrt{t}$, we have*

$$R_T(\theta^*, \tilde{\theta}_i) \le 2\left(\frac{R^2}{k} + kL^2\left(\frac{6n}{1-\gamma} + 6n\nu + 1\right)\right)\sqrt{T}.$$

*Proof:* Since the cost function $f_t(\hat{\theta}(t))$ is convex, $f_t(\tilde{\theta}_i(t)) \le \frac{1}{t}\sum_{s=1}^{t} f_t(\hat{\theta}_i(s))$. Therefore, we have

$$f_t(\tilde{\theta}_i(t)) - f_t(\theta^*) \le \frac{1}{t} R_t(\theta^*, \hat{\theta}_i). \tag{30}$$

Thus, the running average regret is bounded by the regret $R_t(\theta^*, \hat{\theta}_i)$ as

$$R_T(\theta^*, \tilde{\theta}_i) \le \sum_{t=1}^{T}\left(\frac{1}{t} R_t(\theta^*, \hat{\theta}_i)\right). \tag{31}$$

The regret bound is given in (19) which implies

$$R_T(\theta^*, \tilde{\theta}_i) \le \left(\frac{R^2}{k} + kL^2\left(\frac{6n}{1-\gamma} + 6n\nu + 1\right)\right) \times \left(\sum_{t=1}^{T}\frac{1}{\sqrt{t}}\right). \tag{32}$$

The corollary now follows from the integral test on the right hand side of (32). ■

## V. Simulation Results

The bounds presented in Theorem 1 apply after selecting $\psi(\hat{\theta}) = \frac{1}{2}\|\hat{\theta}\|_2^2$ and the parameter $\alpha(t)$ accordingly. In order to find the constants $R$ and $L$ featured in the result, we note that for $\hat{\theta} \in \Theta$, $\psi(\hat{\theta}) \le \frac{1}{2}\theta_{\max}^2$, and thus $R \le \frac{1}{\sqrt{2}}\theta_{\max}$. In this example, we assume that the observation for agent $i$ at time $t$ is of the form $z_{i,t} = a_t\theta + b_t$ for some $a \in (0, a_{\max})$ and $b \in (-b_{\max}, b_{\max})$. Therefore,

$$\sup_{\theta \in \chi} \|z_{t,i}(\theta)\|_2 \le a_{\max}\theta_{\max} + b_{\max}.$$

Further, the function $f_{i,t}$ is Lipschitz as it is convex on a compact domain and the Lipschitz constant can be found by observing that

$$\begin{aligned} &\left| f_{i,k}(\hat{\theta}) - f_{i,k}(\phi)\right| \\ &\le \frac{1}{2}\left|\left(\hat{\theta}-\phi\right)^T H_i^T H_i \left(\hat{\theta}-\phi\right)\right| + \left|z^T H_i\left(\hat{\theta}-\phi\right)\right| \\ &\le \left(\frac{1}{2}\|H_i\|_F^2\left\|\hat{\theta}-\phi\right\|_2 + \|z_{i,t}\|_2\|H_i\|_F\right)\left\|\hat{\theta}-\phi\right\|_2 \end{aligned}$$

and thus $L = \left(\frac{1}{2}\theta_{\max}h_{\max} + a_{\max}\theta_{\max} + b_{\max}\right)h_{\max}$. Hence $R_T(\theta^*, \hat{\theta}_i)/T \to 0$ and the algorithm performs as well as best fixed estimate $\theta^*$ in hindsight (8) "on average". For the case where $\theta_t = \theta_{t+1}$ for $t = 1, 2, \ldots, T$, $\theta^*$ is the optimal estimate.

The ODD and DOA algorithms have been implemented on the described distributed sensor setup (without dual averaging) for $n = 100$ sensors. The objective has been to estimate a scalar $\theta \in \left(-\frac{1}{2}, \frac{1}{2}\right)$ with a fixed $H_i \in \left(0, \frac{1}{4}\right)$ for each agent; hence $\sup_i |H_i| = \frac{1}{4}$. In this example, we have assumed $a \in (0,1)$, $b \in \left(-\frac{1}{4}, \frac{1}{4}\right)$, $\beta = 0.9$, and $k = \frac{1}{4}$. Thus, $d = 1$, $\Theta = \left(-\frac{1}{2}, \frac{1}{2}\right)$, $h_{\max} = \frac{1}{4}$ , $\theta_{\max} = \frac{1}{2}$, $R = \frac{1}{2\sqrt{2}}$, and $L = \frac{13}{64}$.

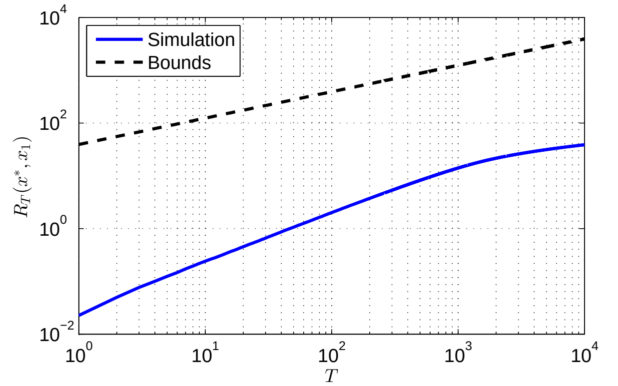


Figure 2. Accuracy of the bounds in (19) where $\mathcal{G}$ is a 100 node random directed graph with edge probability $p = 0.08$, $\nu = 5$, and $\gamma = 0.2034$.

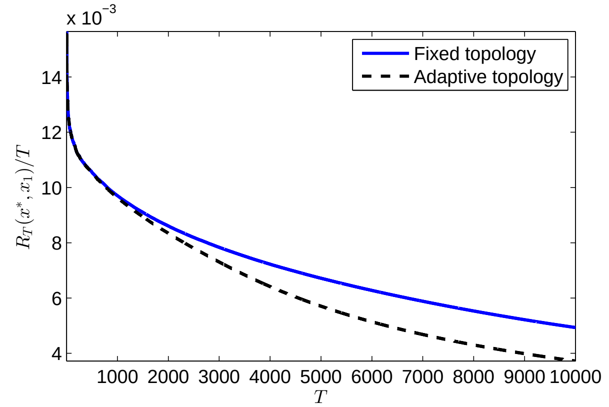


Figure 3. Regret performance over fixed and adaptive network topologies where $\mathcal{G}$ is a 100 node random 4-regular graph and 25 sensors are assumed to have been jammed. For the jammed sensors, $b_t = b_{\max}$ and $a_t = H_i$.

The ODD and DOA algorithms were also applied to random sensor network with edge probability $p = 0.08$. Figure 2 shows a good agreement of the theoretical regret bound (19) and simulation results, indicating that $R_T(x^*, x_1) = O(\sqrt{T})$. The improved performance of the adaptive network topology has been emphasized in Figure 3 in the context of a jamming scenario, where a number of sensors in the random regular network are assumed to have been jammed. This figure also demonstrates that the adaptive sensor network has a better regret performance as compared with the fixed topology sensor network.

In addition, the performance of the proposed adaptive online distributed estimation in the presence of various noise types is presented in Figure 4. These simulation results indicate that $R_T(\theta^*, \hat{\theta}_1) = O(\sqrt{T})$ for all noise types considered without a prior assumption on the noise characteristics.

Furthermore, the role of network connectivity in the performance of the algorithm has been emphasized in Figure 5 for various classes of network topologies, directly correlated to the network connectivity measure $\gamma$. This result can directly

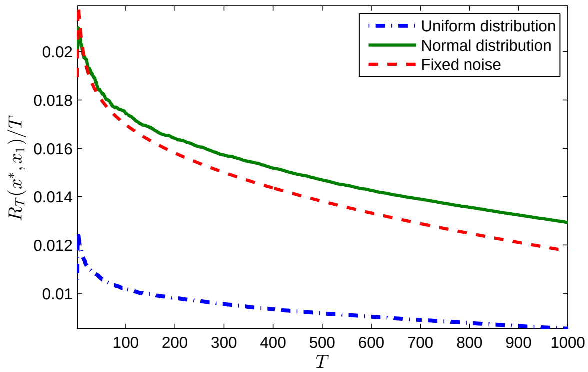


Figure 4. Regret performance for three different observation noise characteristics, where $\mathcal{G}$ is a 100 node random 4-regular graph. The noise signals have been generated from distributions with mean $-b_{\max}$ and standard deviation $b_{\max}$.

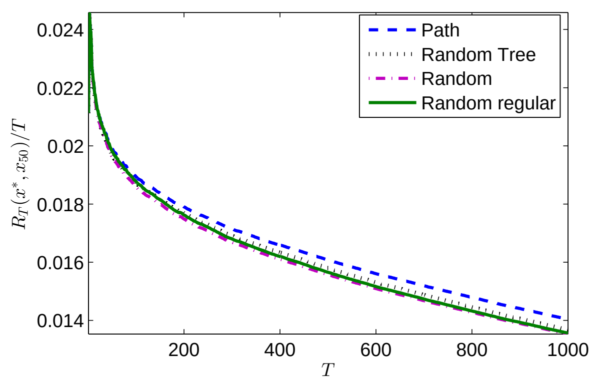


Figure 5. The performance of the online distributed estimation algorithm on four different 100 node graphs with $\gamma = \{0.8999,\ 0.8998,\ 0.7939,\ 0.4110\}$, for the path, random tree, random $k$-regular with $k = 4$, and random graph with egde probability $p = 0.08$, respectively, in an increasing order of performance.

be applied to the designing effective sensor network topologies that operate in highly uncertain environments. Suitable metrics for such a topology design include $\lambda_2(P(\mathcal{G}^0))$ that predictively scales with $n$, such as random regular graphs and expander graphs [48].

## VI. Conclusion

This paper studies the problem of decentralized estimation in sensor networks based on an online diffusion strategy operating in highly uncertain environments. An online distributed algorithm has been presented that evolves distributively using only local information available to the sensors. Our analysis provided a sublinear regret bound of $O(\sqrt{T})$ for this distributed online estimation algorithm. In addition, the distributed online topology design has been proposed as a weight allocation strategy that, on average, performs as well as the best strategy for information diffusion in hindsight. The convergence analysis of the distributed estimation algorithm highlighted the role of two measures of network connectivity on regret performance. The first measure relates to the second smallest eigenvalue of the graph Laplacian and the second measure to the diameter of the network.

In order to justify the suitability of the online setting for sensor networks, the proposed algorithm was applied to a distributed sensor estimation problem. The estimates were acquired in real-time and coupled with sensors' susceptibility to unknown errors and jamming. The simulation results indicate that the sensor network can provide an estimate that on the average performs as well as the best case fixed solution in hindsight. In addition, we explored the proposed online distributed estimation algorithm for various classes of sensor networks and highlighted the role of network connectivity on the network-level regret.

This work can be extended in several directions. One such extension, which is the subject of our future work, involves examining online distributed filtering. More generally, the online approach can be adopted for a host of network dynamic systems that operate in highly unstructured environments, requiring that a learning algorithm is embedded in the network-level decision-making process.

## VII. Appendix

We note that Lemmas 3 and 4 can be found in [24], and are presented here without proof.

**Lemma 3.** *For any $u, v \in \mathbb{R}^m$, and under the conditions stated for proximal function $\psi$ and step size $\alpha(t)$, we have $\|\Pi_{\mathcal{X}}^{\psi}(u,\alpha) - \Pi_{\mathcal{X}}^{\psi}(v,\alpha)\| \le \alpha\|u - v\|_*$.*

**Lemma 4.** *For any positive and non-increasing sequence $\alpha(t)$ and $\theta^* \in \Theta$,*

$$\sum_{t=1}^{T}\langle \overline{g}(t), \phi(t) - \theta^*(t)\rangle \le \frac{1}{2}\sum_{t=1}^{T}\alpha(t-1)\|\overline{g}(t)\|_\star^2 + \frac{1}{\alpha(T)}\psi(\theta^*),$$

*where the sequence $\phi(t)$ is generated by* (18).

The following result presents a bound on $\|\bar{y}(t) - y_i(t)\|_*$ proportional to the error incurred in the decentralized update step of Algorithm 1.

**Lemma 5.** *For sequences $y_i(t)$ and $\bar{y}(t)$ generated by line 5 of Algorithm 1 and* (17)*, respectively, we have, $\|\bar{y}(t) - y_i(t)\|_* \le nL\left(\frac{2\gamma-1}{1-\gamma} + \nu + 2\right)$ for all $i \in [n]$, where $\gamma$ and $\nu$ are measures of network connectivity.*

*Proof:* If we write the update in line 5 for all $i \in [n]$, by induction through $s$ steps we have,

$$y_i(t) = \sum_{j=1}^{n} P_{ij}^{(t-1,t-s)} y_j(t-s) + \sum_{k=t-s}^{t-2}\sum_{j=1}^{n} P_{ij}^{(t-1,k+1)} g_j(k) + g_i(t-1). \tag{33}$$

Since $\bar{y}(t)$ evolves as in (17), by setting $s = t$ in (33) and assuming $y_i(0) = 0$, we get,

$$\bar{y}(t) - y_i(t) = \sum_{k=0}^{t-2}\left(\sum_{j=1}^{n}\left(\pi_j - P_{ij}^{(t-1,k+1)}\right) g_j(k)\right) + \bar{y}(t-1) - g_i(t-1). \tag{34}$$

Thus, the dual norm of (34) is bounded as

$$\|\bar{y}(t)-y_i(t)\|_* \leq \| \sum_{k=0}^{t-2}\left(\sum_{j=1}^{n}\left(\pi_j - P_{ij}^{(t-1,k+1)}\right) g_j(k)\right)\|_* + \|\bar{g}(t-1)-g_i(t-1)\|_*, \quad (35)$$

and the right hand side of (35) can be bounded by

$$\|\bar{y}(t)-y_i(t)\|_* \leq \sum_{k=0}^{t-2}\sum_{j=1}^{n}\left|P_{ij}^{(t-1,k+1)}-\pi_j\right| \|g_j(k)\|_* + \|\bar{g}(t-1)-g_i(t-1)\|_*. \quad (36)$$

Since $\|g_i(t)\|_\star \leq L$,

$$\|\bar{y}(t)-y_i(t)\|_* \leq L\sum_{k=0}^{t-2}\sum_{j=1}^{n}\left|P_{ij}^{(t-1,k+1)}-\pi_j\right| + 2L. \quad (37)$$

In addition, since the graph is assumed to be strongly connected, the communication matrices are stochastic, indecomposable and aperiodic. Therefore, based on the weak ergodicity of inhomogeneous Markov chains, $P^{(t-1,0)}$ converges exponentially to a rank-one matrix of the form $\mathbf{1}\pi^T$ as $t \to \infty$, and the vector $\pi$ belongs to $\{\pi \in \mathbb{R}^n | \pi \geq 0, \sum_{i=1}^n \pi_i = 1\}$. The rate of convergence of this Markov chain is a function of its ergodic coefficient. The ergodic coefficient for a stochastic matrix $Q \in \mathbb{R}^{n\times n}$ is given by

$$\tau(Q) = 1 - \min_{i,j\in[n]}\sum_{k=1}^{n}\min\{q_{ik}, q_{jk}\}.$$

Based on Lemma 4 in [49], there exists an integer $\nu \geq 1$ for which the matrix $P^{(\nu,0)}$ is scrambling[4] and thus $\tau(P^{(\nu,0)}) < 1$ [50]. In addition, by defining

$$\gamma = \max\left\{\tau(P^{(\nu,0)}) < 1 \,|\, \nu \geq 1\right\}, \quad (38)$$

where the maximization is over all realization of the sequence $P^{(\nu,0)}$. Then, the rate of convergence of the product of stochastic matrices, based on Theorem 1 of [47], can be expressed as[5]

$$\left|P_{ij}^{(t-1,0)} - \pi_j\right| \leq \gamma^{\lfloor \frac{t}{\nu} \rfloor}. \quad (39)$$

Since $\gamma < 1$, (37) is bounded by

$$\|\bar{y}(t)-y_i(t)\|_* \leq nL\sum_{k=1}^{t-1}\gamma^k + nL(\nu-1) + 2L \leq nL\left(\frac{1}{1-\gamma}+\nu-2\right)+2L. \quad (40)$$

■

[4]Note that a matrix $Q$ is scrambling if and only if for every pair of rows $i_1$ and $i_2$, there exists a column $k$ such that $[Q]_{i_1 k} > 0$ and $[Q]_{i_2 k} > 0$.

[5]Note that $\lfloor x \rfloor$ is the largest integer less than or equal to $x$.

Based on the sub-multiplicative property of the ergodic coefficient, we have

$$\tau(P^{(\nu,0)}) \leq \tau(P^{\nu})\tau(P^{\nu-1})\cdots\tau(P^0) \;\text{ for } \nu > 0,$$

which implies $\gamma \leq \tau(P^{\nu})\tau(P^{\nu-1})\cdots\tau(P^0)$. The following proposition provides an upper bound on $\nu$ as in (40).

**Proposition 6.** *Consider a set $\mathcal{P}$ of stochastic matrices with positive diagonal elements, representing the underlying topology of the network $\mathcal{G} = (V, E)$. Suppose that any two matrices $P^{k_1} \in \mathcal{P}$ and $P^{k_2} \in \mathcal{P}$ are of the same type, i.e., $P^{k_1} \sim P^{k_2}$.[6] Then, there exists an integer $\nu$ such that*

$$1 \leq \nu \leq \min_i \max_j \{\mathit{dist}(j,i) |\, \mathit{for\ all}\ i,j \in [n]\}, \quad (41)$$

*where any sequence $Q = P^{(m+\nu-2,m)}$ of matrices in $\mathcal{P}$ is not scrambling, while $P^{m+\nu-1}Q$ is scrambling. Moreover, the dist$(i,j)$ is the minimum length of any directed path from node $i$ to node $j$ while the length of each edge $\{j,i\} \in E$ is 1.*

*Proof:* Consider $Q_1 = P^m$ and $Q_2 = P^{m+1}P^m$. Thus every entry of $Q_2$ is represented by

$$[Q_2]_{ij} = \sum_{k=1}^{n}\left[P^{m+1}\right]_{ik}\left[P^m\right]_{kj}.$$

Since $[P^m]_{ii} > 0$ for all $i \in [n]$ and integer $m \geq 1$, the entry $[Q_2]_{ij}$ is positive if $(j,i) \in E$, $(i,j) \in E$, or there exists a node $k \in [n]$ in the directed path from node $j$ to node $i$ with dist$(j,i) = 2$. Thus, the corresponding zero entry of $Q_1$ that has one of the aforementioned properties will be positive in $Q_2$. By induction, it follows that the entry of $[Q_\nu]_{ij}$ will be positive if $(j,i) \in E$, $(i,j) \in E$, or there exists a node $k \in [n]$ in the directed path from node $j$ to node $i$ with dist$(j,i) = \nu_i$. Therefore, for each row $i$ of $Q_\nu$, all entries will be positive when

$$\nu_i = \max_j \{\text{dist}(j,i) |\text{ for all } j \in [n]\}.$$

When every element of any row of the sequence $Q = P^{(m+\nu-2,m)}$ of matrices in $\mathcal{P}$ is positive, the matrix $Q$ is scrambling and $\nu$ satisfies the bound (41). ■

A similar observation for the adjacency matrix of $\mathcal{G} = (V, E)$ can be found in the algebraic graph theory literature such as [51].

## References


[1] A. J. Hashemipour, HR, Roy, Sumit, Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. I, pp. 88–94, 1988.

[2] B. Rao and H. Durrant-Whyte, "Fully decentralised algorithm for multisensor Kalman filtering," *IEE Proceedings D: Control Theory and Applications*, vol. 138, no. 5, p. 413, 1991.

[3] V. Delouille, R. Neelamani, and R. Baraniuk, "Robust distributed estimation in sensor networks using the embedded polygons algorithm," *International Symposium on Information Processing in Sensor Networks*, p. 405, 2004.


[6]Note that matrices $A$ and $B$ are of the same type if they have zero elements and positives elements in the same place.

[4] D. Spanos, R. Olfati-Saber, and R. Murray, "Approximate distributed kalman filtering in sensor networks with quantifiable performance," *International Symposium on Information Processing in Sensor Networks*, pp. 133–139, 2005.

[5] A. Das and M. Mesbahi, "Distributed linear parameter estimation over wireless sensor networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 4, pp. 1293–1306, 2009.

[6] S. Aldosari and J. Moura, "Topology of sensor networks in distributed detection," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1061–1064, 2006.

[7] S. Kar and J. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3315–3326, 2008.

[8] A. Laszka, L. Buttyán, and D. Szeszlér, "Designing robust network topologies for wireless sensor networks in adversarial environments," *Pervasive and Mobile Computing*, vol. 9, no. 4, pp. 546–563, 2013.

[9] S. S. Stanković, M. S. Stanković, and D. M. Stipanović, "Consensus based overlapping decentralized estimation with missing observations and communication faults," *Automatica*, vol. 45, no. 6, pp. 1397–1406, 2009.

[10] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. NJ: Princeton University Press, 2010.

[11] G. B. G. Schizas, ID, Gonzalo Mateos, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[12] F. Cattivelli and A. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[13] S. Kar and J. M. F. Moura, "Convergence Rate Analysis of Distributed Gossip (Linear Parameter) Estimation: Fundamental Limits and Trade-offs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 674–690, 2011.

[14] L. Xiao, S. Boyd, and S. Lall, "A space-time diffusion scheme for peer-to-peer least-squares estimation," *International Conference on Information Processing in Sensor Networks*, 2006.

[15] S. Tu and A. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, 2012.

[16] F. Cattivelli, C. Lopes, and A. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, 2008.

[17] L. Li and J. Chambers, "Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology," *Statistical Signal Processing*, vol. 1, no. 5, pp. 757–760, 2009.

[18] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive Robust Distributed Learning in Diffusion Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, 2011.

[19] X. Zhao and A. Sayed, "Performance limits for distributed estimation over LMS adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5107–5124, 2012.

[20] Q. Zhou, S. Kar, L. Huie, and S. Cui, "Distributed estimation in sensor networks with imperfect model information: An adaptive learning-based approach," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3109–3112, 2012.

[21] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control,*, vol. 54, pp. 48–61, 2009.

[22] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control,*, pp. 1291–1306, 2011.

[23] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, pp. 221–229, 2013.

[24] 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.

[25] S. Ram, A. Nedic, and V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," *SIAM Journal on Optimization*, pp. 1–30, 2009.

[26] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, 2010.

[27] A. Agarwal and J. Duchi, "Distributed delayed stochastic optimization," *IEEE Conference on Decision and Control*, pp. 5451–5452, 2012.

[28] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," *International Conference on Machine Learning*, pp. 421–422, 2003.

[29] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, pp. 169–192, 2007.

[30] L. Xiao, "Dual Averaging Methods for Regularized Stochastic Learning and Online Optimization," *Journal of Machine Learning Research*, vol. 11, pp. 2543–2596, 2010.

[31] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 99, p. 1, 2012.

[32] M. Raginsky, N. Kiarashi, and R. Willett, "Decentralized Online Convex Programming with Local Information," *American Control Conference*, pp. 5363–5369, 2011.

[33] S. Hosseini, A. Chapman, and M. Mesbahi, "Online Distributed Optimization via Dual Averaging," *IEEE Conference on Decision and Control*, 2013.

[34] A. Ghosh and S. Boyd, "Growing well-connected graphs," *IEEE Conference on Decision and Control*, pp. 6605–6611, 2006.

[35] D. Zelazo and M. Mesbahi, "Edge agreement: graph-theoretic performance bounds and passivity analysis," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 544–555, 2011.

[36] Y. Wan, S. Roy, and A. Saberi, "Network design problems for controlling virus spread," *IEEE Conference on Decision and Control*, pp. 3925–3932, 2007.

[37] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[38] A. Chapman and M. Mesbahi, "Semi-autonomous consensus: network measures and adaptive trees," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 19–31, 2013.

[39] A. Chapman, E. Schoof, and M. Mesbahi, "Distributed Online Topology Design for Disturbance Rejection," *IEEE Conference on Decision and Control*, 2013.

[40] N. Littlestone and M. Warmuth, "The weighted majority algorithm," *Information and Computation*, vol. 108, pp. 212–261, 1994.

[41] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[42] S. Shalev-Shwartz, "Online learning and online convex optimization," *Foundations and Trends in Machine Learning*, vol. 4, pp. 107–194, 2012.

[43] S. Bubeck, "Introduction to Online Optimization," *Lecture Notes*, 2011.

[44] E. Hazan, "The Convex Optimization Approach to Regret Minimization," *Optimization for machine learning*, pp. 287–294, 2011.

[45] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical Programming*, vol. 120, pp. 221–259, 2007.

[46] C. W. Wu, "On bounds of extremal eigenvalues of irreducible and m-reducible matrices," *Linear Algebra and its Applications*, vol. 402, pp. 29–45, 2005.

[47] J. Anthonisse and H. Tijms, "Exponential convergence of products of stochastic matrices," *Journal of Mathematical Analysis and Applications*, vol. 59, no. 2, pp. 360–364, 1977.

[48] B. Bollobás, *Modern Graph Theory*. New York: Springer, 1998.

[49] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices," *Proceedings of the American Mathematical Society*, vol. 14, no. 5, pp. 733–737, 1963.

[50] I. Ipsen and T. Selee, "Ergodicity coefficients defined by vector norms," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 1, pp. 153–200, 2011.

[51] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer, 2001.