Team-Optimal Online Estimation of Dynamic Parameters over Distributed Networks

O. Fatih Kilic [a,*], Muhammed O. Sayin [b], Suleyman S. Kozat [c]

[a] *Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78705 USA.*

[b] *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61801 USA.*

[c] *Department of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey.*

Abstract

We study online parameter estimation over a distributed network, where the nodes in the network collaboratively estimate a dynamically evolving parameter using noisy observations. The nodes in the network are equipped with processing and communication capabilities and can share their observations or local estimates with their neighbors. The conventional distributed estimation algorithms cannot perform the team-optimal online estimation in the finite horizon global mean-square error sense (MSE). To this end, we present a team-optimal distributed estimation algorithm through the disclosure of local estimates for tracking an underlying dynamic parameter. We first show that the optimal estimation can be achieved through the diffusion of all the time stamped observations for any arbitrary network and prove that the team optimality through disclosure of local estimates is only possible for certain network topologies. We then derive an iterative algorithm to recursively calculate the combination weights of the disclosed information and construct the team-optimal estimate for each time step. Through series of simulations, we demonstrate the superior performance of the proposed algorithm with respect to the state-of-the-art diffusion distributed estimation algorithms regarding the convergence rate and the finite horizon MSE levels. We also show that while conventional distributed estimation schemes cannot track highly dynamic parameters, through optimal weight and estimate construction, the proposed algorithm presents a stable MSE performance.

Key words: Optimal estimation, distributed network, dynamic parameter, online estimation

---

1 Introduction

Recently, due to advancements in information technologies, distributed learning and estimation techniques have attracted significant attention thanks to their fast convergence and robustness properties for fast streaming data [1–5]. In a distributed estimation framework, we consider a network of agents observing a temporal signal about an underlying state, possibly coming from different spatial sources with different statistics. Each agent in the network is equipped with communication and processing capabilities. The aim of each agent is to estimate the underlying parameter of interest, as an example, by minimizing the expected Euclidean distance between the estimate and the true value of the state (the minimum mean-square estimation (MMSE)). The agents in the network are connected to a set of neighboring nodes and can exchange information, i.e. observations and/or estimates, between them to improve their learning process. To illustrate, assume a network of emission sensors distributed over a greenhouse to monitor the CO_2 levels for a precision agriculture application [6]. Since the agents would collect different observations from different parts of the area, they can cooperate in the network to

* Corresponding author.
Email addresses: `okilic@utexas.edu` (O. Fatih Kilic), `sayin2@illinois.edu` (Muhammed O. Sayin), `kozat@ee.bilkent.edu.tr` (Suleyman S. Kozat).

rapidly learn and track the true CO_2 levels for an enhanced intervention.

In this regard, the distributed learning and estimation has been extensively studied in the signal processing and machine learning literatures [7–15]. However, the classical methods either do not consider the information diffusion scheme among the agents and/or construction of the optimal combination methods to obtain the MMSE performance or are not applicable for real-time applications [11]. To this end, in this paper, we present an approach to obtain a team-optimal distributed online estimation scheme by exploiting the network structure and the information disclosure and combination when the underlying state is non-stationary and time varying.

There exists an extensive research on distributed estimation of a time invariant or a dynamic state parameter, which are mainly studied under centralized and decentralized distributed learning frameworks [7–14, 16, 17]. In the centralized frameworks, all the agents in the network are connected to a fusion center and each agent transmits its information to the center for the construction of the final estimate [7, 16, 17]. Since all the information is collected by a single node such methods do not require any specific information sharing scheme and constructing the global optimal estimate is straightforward. However, this approach has serious disadvantages regarding communication and computation loads on the network, i.e. transmitting all the peripheral information to a single node requires a huge communication bandwidth and processing all the collected information on a single unit requires a significant computational power [7, 10].

In the alternative decentralized frameworks, each agent in the network has a different set of neighboring nodes consisting of spatially close ones and exchange information only with these nodes to overcome the former problems [18]. In these approaches, agents only disclose their local information on the underlying parameter and combine the received information to produce their final estimates. In this framework, the information efficiently propagates through the network to improve the overall performance [19].

In the consensus approach of the decentralized frameworks, all the agents in the network reach to a "consensus" on their estimates after collecting and processing their information locally [13, 14]. However, this approach either requires a use of two time scales to reach to the consensus immediately or decaying learning rate for constructing the consensus among the agents in time [14, 20]. The use of two time scales limits the performance of the network on real-time applications. On the other hand, the use of decaying learning rates hinders the ability of the system to adaptively adjust or learn in time varying environments [11].

In [12–14, 21] and [22], authors present diffusion based approaches for distributed estimation, where the network is able to respond to the fast-streaming data in an online manner by using a single time scale. In the diffusion based strategies, agents process their observations locally and disclose the corresponding estimates to the neighboring nodes and improve their performance through combining the received estimates. In [11], authors prove that the diffusion based approaches outperform single time scale consensus strategies regarding the global MSE performance. However, neither of these methods consider the network topology or information disclosure procedures to obtain a globally optimal solution. On the other hand, in [8, 10], diffusion incremental solutions are shown to reach to the optimal estimate by defining a certain path through the network, which is not practical against the fast streaming data or the dynamic configurations.

In [23], authors presented a novel approach to obtain the team-optimal distributed estimation of a static underlying parameter by exploiting the network structure, and the optimal information disclosure and combination without any incremental path requirements. However, in most of the real-life applications, the underlying parameter is subject to a change, i.e. it evolves in time [24]. Although there exists different studies on the distributed estimation of a dynamic parameter, these algorithms again do not consider the correlation of the disclosed information between the agents in the network due to the dynamic evolution of the underlying parameter. [2, 24, 25]. Hence, these algorithms cannot achieve the team-optimal estimation and the problem requires a different approach than the solutions available in the literature.

To this end, we work on the team-optimal estimation of dynamic parameters over distributed networks. We first use the framework of [26] to establish the model and the problem. Then, we introduce the efficient and optimal distributed learning (EODL) algorithm for the online estimation of dynamic parameters and prove that it is only applicable over certain network topologies. We also show the superior performance of the proposed method compared to the state-of-the-art methods through numerical examples.

We organize the paper as follows. In Section 2, we present the team framework for the dynamic parameter estimation and show that the optimal estimate can be constructed through diffusion of the time stamped information. Then, in Section 3, we prove that the team-optimal estimation through disclosure of local estimates can be achieved only under certain network topologies. Later in Section 4, we provide an iterative algorithm to construct the optimal combination weights and the estimate over such networks. We demonstrate the performance of the proposed algorithm through series of simulations in Section 5 and conclude the paper with final remarks in Section 6.

2 Team Framework for Distributed Estimation

We consider a distributed network with m agents equipped with processing and communication capabilities. We form the network as an undirected graph,

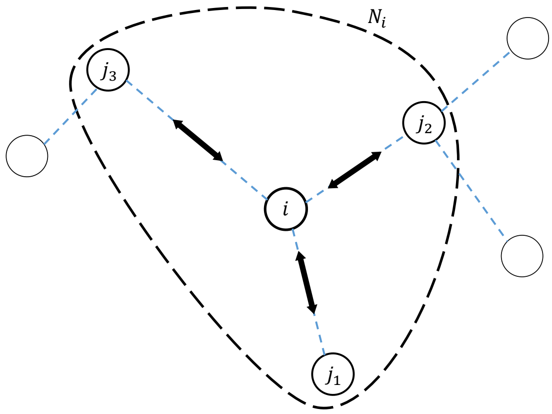

Fig. 1. First order neighborhood of the agent i over a distributed network. The agent i only exchanges information with the nodes in this neighborhood.

where vertices and edges represent the agents and the communication links respectively, as shown in Fig. 1. For each agent i, we denote the set of agents, whose information is available to the agent i after transmission over k communication links (after k-hops) as $\mathbf{N}_i^{(k)}$. We define $\mathbf{N}_i^{(k)}$ as

$$\mathbf{N}_i^{(k)} = \{j_1, \cdots, j_{\pi_i^{(k)}}\}, \tag{1}$$

where $\pi_i^{(k)} = |\mathbf{N}_i^{(k)}|$ is the cardinality of the set $\mathbf{N}_i^{(k)}$. We assume that $\mathbf{N}_i^{(0)} = \{i\}$ and $\mathbf{N}_i^{(k)} = \emptyset$ for $k < 0$. In Fig.1, we demonstrate the first neighborhood of the agent i, where $\mathbf{N}_i = \{j_1, j_2, j_3\}$ and $\pi_i = 3$. We drop the superscript on the first order neighborhood for notational simplicity.

We choose the random walk for the modeling of the underlying dynamic state since the random walk model is extensively used to model the behavior of highly complex structures from biological systems to social networks [2,24,25]. Hence, the underlying state $x_t \in \mathbb{R}$ evolves according to

$$x_{t+1} = \gamma x_t + w_t, \tag{2}$$

where $\gamma \in \mathbb{R}$ is the expected rate of change. The term $w_t \in \mathbb{R}$ is the state noise and it is an i.i.d. Gaussian random process $\{\mathcal{W}_t\}$ with variance σ_w^2. The initial state is sampled from a Gaussian random variable such that $\mathcal{X}_0 \sim N(0, \sigma_0^2)$ [1].

Each agent in the network observes a noisy version of an underlying dynamic state as

$$y_{i,t} = x_t + n_{i,t} \tag{3}$$

for $i = 1, \cdots, m$ and $n_{i,t} \in \mathbb{R}$ is a white Gaussian process $\{\mathcal{N}_{i,t}\}$ with variance $\sigma_{n_i}^2$. We assume that the observation noise is spatially independent and the variance of the noise signals are known to each agent (if they are not available, then they can be estimated from the observations [27]). Correspondingly, $y_{i,t}$ becomes a realization of a random process $\{\mathcal{Y}_{i,t}\}$, where $\mathcal{Y}_{i,t} = \mathcal{X}_t + \mathcal{N}_{i,t}$. At each instant, an agent receives a local observation and diffused information from the neighboring agents, while it also diffuses information to its neighboring agents.

Obviously, each agent can alone track the underlying state in the MMSE sense under certain regulatory conditions [28]. However, the use of distributed cooperation can greatly enhance the learning rate and the robustness of the system [27]. To this end, we aim to find an optimal estimation strategy regarding the MSE performance for a team of distributed agents. To provide a lower bound on the performance of the team, we first consider a case where the agents in the network disclose the stamped versions, with time and the agent ID, of their observations and the received information. Thus, each agent has access to the observations of all the other agents in the network. However, we note that only the observations from the neighboring agents can be directly received. The observations from the non-neighboring agents can only be accessed after going over certain number of communication links, i.e. the

[1] In this paper, all random variables are represented as uppercase calligraphic letters, i.e. $\mathcal{X}$, and all the realizations of these variables are presented as their lowercase characters, i.e. x. All the vectors are column vectors and denoted by boldface lowercase letters.

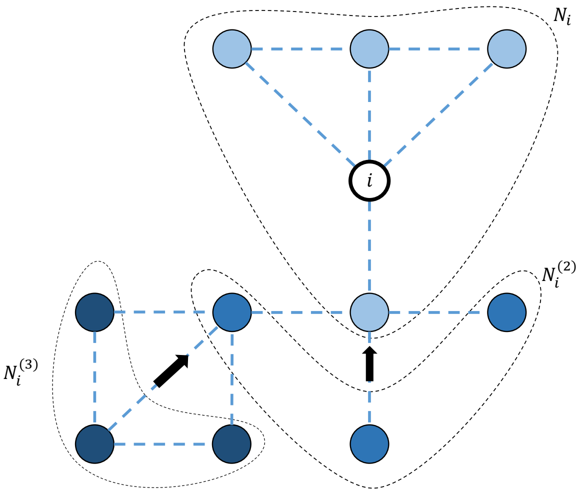

Fig. 2. Information from the agent in $\mathbf{N}_i$ can be directly received by the node i. Information coming from the agents in $\mathbf{N}_i^{(2)}$ and $\mathbf{N}_i^{(3)}$ can be accessed with a certain delay.

information of the agent $j \in \mathbf{N}_i^{(2)}$ can be accessed by the agent i after being transmitted over 2 communications links. We illustrate this behavior of the distributed networks in Fig.2, where we show $\mathbf{N}_i$, $\mathbf{N}_i^{(2)}$ and $\mathbf{N}_i^{(3)}$ for the node i. Only the information from $\mathbf{N}_i$ can be directly received by node i, otherwise the information have to follow the described neighborhood path to reach the node i.

We define the team cost of the network for a time horizon T, when each agent i makes the estimate $\hat{x}_{i,t}$ as

$$\sum_{t=1}^{T}\sum_{i=1}^{m}\mathbb{E}\left|\left|\mathcal{X}_t - \hat{x}_{i,t}\right|\right|^2 .$$

We also emphasize that due to the connected structure of the network, each agent will have access to all the observations in the network, although with certain delay. Therefore, we denote the information aggregated at the agent i at time t as

$$D_{i,t} = \left\{\{y_{i,\tau}\}^{\tau\leq t}, \{y_{j,\tau}\}_{j\in\mathbf{N}_i}^{\tau\leq t-1}, \{y_{j,\tau}\}_{j\in\mathbf{N}_i^{(2)}}^{\tau\leq t-2}, \cdots, \{y_{j,\tau}\}_{j\in\mathbf{N}_i^{(\kappa_i)}}\right\}, \tag{4}$$

where κ_i denotes the communication link delay for the furthest node from the ith node. Note that $\{y_{j,\tau}\}_{j\in\mathbf{N}^{(t_i)}}^{\tau\leq t-t_i}$ is the set of observations received from t_i hop away neighborhood of the agent i, which is explicitly defined as

$$\{y_{j,\tau}\}_{j\in\mathbf{N}^{(t_i)}}^{\tau\leq t-t_i} \triangleq \{y_{j_1,t-t_i},\ldots,y_{j_1,0},\cdots,y_{j_{\pi_i^{(t_i)}},t-t_i},\ldots,y_{j_{\pi_i^{(t_i)}},0}\}. \tag{5}$$

With this aggregated information, we construct the team optimization problem as

$$\begin{aligned}\min_x \sum_{t=1}^{T}\sum_{i=1}^{m}\mathbb{E}\Bigg[||\mathcal{X}_t - x||^2\Bigg|\{\mathcal{Y}_{i,\tau}=y_{i,\tau}\}^{\tau\leq t},\\ \{\mathcal{Y}_{j,\tau}=y_{j,\tau}\}_{j\in\mathbf{N}_i}^{\tau\leq t-1},\cdots,\{\mathcal{Y}_{j,\tau}=y_{j,\tau}\}_{j\in\mathbf{N}_i^{(\kappa_i)}}^{\tau\leq t-\kappa_i}\Bigg],\end{aligned} \tag{6}$$

which corresponds, for each agent, to solving

$$\begin{aligned}\min_x \sum_{t=1}^{T}\mathbb{E}\Bigg[||\mathcal{X}_t - x||^2\Bigg|\{\mathcal{Y}_{i,\tau}=y_{i,\tau}\}^{\tau\leq t},\\ \{\mathcal{Y}_{j,\tau}=y_{j,\tau}\}_{j\in\mathbf{N}_i}^{\tau\leq t-1},\cdots,\{\mathcal{Y}_{j,\tau}=y_{j,\tau}\}_{j\in\mathbf{N}_i^{(\kappa_i)}}^{\tau\leq t-\kappa_i}\Bigg].\end{aligned} \tag{7}$$

The solution to the optimization problem in (7) at each time step t gives the MMSE estimate for the agent i such that

$$\hat{x}_{i,t} = \mathbb{E}\Bigg[\mathcal{X}_t\Bigg|\{\mathcal{Y}_{i,\tau}=y_{i,\tau}\}^{\tau\leq t},\{\mathcal{Y}_{j,\tau}=y_{j,\tau}\}_{j\in\mathbf{N}_i}^{\tau\leq t-1},\cdots,\{\mathcal{Y}_{j,\tau}=y_{j,\tau}\}_{j\in\mathbf{N}_i^{(\kappa_i)}}^{\tau\leq t-\kappa_i}\Bigg]. \tag{8}$$

Therefore, the estimate in (8) produces the team-optimal solution in an MSE sense and creates the lower bound for the team-framework.

Remark 2.1 *The presented case provides a lower bound on the error performance of the team in an MSE sense through the disclosure of the time stamped observations. This scheme requires excessive amount of storage on the nodes and the communication load for the network, especially for larger networks. Note that the reduced storage and the communication load are essential for the* *applicability of the distributed networks to real life problems [29,30]. Therefore, we develop team-optimal estimation strategies for the distributed networks that achieves the error performance lower bound of* (8)*, albeit the nodes only store and diffuse their current local estimates. However, in the next section, we show that such an error performance with the disclosure of local estimates can only be achieved over certain network topologies.*

3 Optimal Estimation with the Disclosure of Local Estimates

In this section, we show that the team optimal estimation lower bound for dynamic parameters can be achieved over tree-networks through disclosure of local estimates and such performance cannot be achieved over cyclic networks [26].

We define the tree-networks as graph structures, where the vertices are connected with undirected edges without any cycles as shown in Fig.3. We also note that for any arbitrary network topology, a minimum spanning tree of the network can be constructed by eliminating the cycles [31–34].

Using the tree structure of the network, we partition the set of information coming from a particular neighborhood. For the tree networks, a neighboring set for the agent i can be expressed as

$$\mathbf{N}_i^{(k)} = \bigcup_{j \in \mathbf{N}_i} (\mathbf{N}_i^{(k)} \cap \mathbf{N}_j^{(k-1)})$$

and again due to the network structure, the intersecting sets are disjoint such that

$$(\mathbf{N}_i^{(k)} \cap \mathbf{N}_{j_1}^{(k-1)}) \bigcap (\mathbf{N}_i^{(k)} \cap \mathbf{N}_{j_2}^{(k-1)}) = \emptyset$$

for all $j_1, j_2 \in \mathbf{N}_i$ and $j_1 \neq j_2$. Therefore, we partition the information received

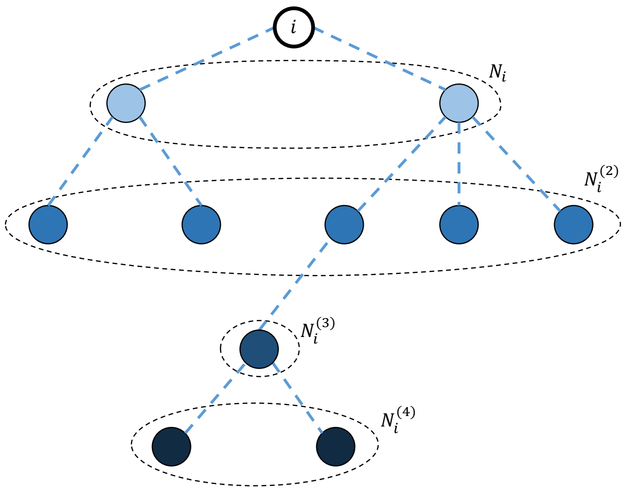

Fig. 3. Structure of a depth-4 tree network with the corresponding neighborhoods of the agent i. Note that we eliminated the cyclic connections from Fig.2 to avoid multipath information diffusion and obtain team-optimal estimation.

at the agent i after k-hops as

$$\{y_{j,\tau}\}_{j\in\mathbf{N}_i^{(k)}}^{\tau\leq t-k} = \{\{y_{j,\tau}\}_{j\in\mathbf{N}_i^{(k)}\cap\mathbf{N}_{j_1}^{(t-k)}}^{\tau\leq k-1}, \cdots, \{y_{j,\tau}\}_{j\in\mathbf{N}_i^{(k)}\cap\mathbf{N}_{j_{\pi_i}}^{(k-1)}}^{\tau\leq t-k}\}.$$

Using this partitioning method, we define the set of new measurements coming from agent j to i at time $t=2$ as

$$z_{j\to i,2} \triangleq \left\{\{y_{k,\tau}\}_{k\in\mathbf{N}_i\cap\mathbf{N}_j^{(0)}}, \{y_{k,\tau}\}_{k\in\mathbf{N}_i^{(2)}\cap\mathbf{N}_j^{(1)}}\right\}. \tag{9}$$

Note that the expression in (9) can also be written as

$$z_{j\to i,2} = D_{j,2}/\{y_{j,1}, y_{i,1}\}, \tag{10}$$

where $y_{j,1} = D_{j,1}$ and $y_{i,1} = D_{i,1} = z_{i\to j,1}$. Thus we can generalize the new information expression for any time t as

$$z_{j\to i,t} = D_{j,t}/\{D_{j,t-1} \cup z_{i\to j,t-1}\}, \tag{11}$$

Using (11), we write all the information aggregated at the agent i as

$$D_{i,t} = \{y_{i,t}, z_{j_1 \to i,t-1}, \cdots, z_{j_{\pi_i} \to i,t-1}, D_{i,t-1}\}, \tag{12}$$

where $z_{j \to i,t}$ is constructible from $D_{i,\tau}$ and $D_{j,\tau}$ for $\tau \le t$ as we show using (10) and (11). Therefore, using (12), we construct the optimal estimate again with an abuse of notation as

$$\hat{x}_{i,t} = \mathbb{E}\left[\mathcal{X}_t \middle| y_{i,t}, z_{j_1 \to i,t-1}, \cdots, z_{j_{\pi_i} \to i,t-1}, D_{i,t-1}\right]. \tag{13}$$

Considering $z_{j \to i,t}$ is constructible from $D_{i,\tau}$ and $D_{j,\tau}$ for $\tau \le t$, we write the optimal estimate in (13) as

$$\begin{aligned}\hat{x}_{i,t} &= \mathbb{E}\left[\mathcal{X}_t \middle| \{y_i, \tau\}^{\tau \le t}, \{D_{j,\tau}\}_{j \in \mathbf{N}_i}^{\tau \le t-1}\right] \\ &= \mathbb{E}\left[\mathcal{X}_t \middle| y_{i,t}, D_{i,t-1}, \{D_{j,t-1}\}_{j \in \mathbf{N}_i}\right]\end{aligned}$$

and since $\hat{x}_{j,t-1} = \mathbb{E}[\mathcal{X}_{t-1} | D_{j,t-1}]$, we obtain

$$\begin{aligned}\hat{x}_{i,t} &= \mathbb{E}\left[\mathcal{X}_t \middle| y_{i,t}, \mathbb{E}[\mathcal{X} | D_{i,t-1}], \{\mathbb{E}[\mathcal{X} | D_{j,t-1}]\}_{j \in \mathbf{N}_i}\right] \\ &= \mathbb{E}\left[\mathcal{X}_t \middle| y_{i,t}, \hat{x}_{i,t-1}, \{\hat{x}_{j,t-1}\}_{j \in \mathbf{N}_i}\right].\end{aligned} \tag{14}$$

Hence, we conclude that we can construct the optimal estimate through disclosure of local estimates over the tree-networks.

In the following, we introduce the efficient and the optimal distributed online learning algorithm for dynamic state estimation. We propose a method that iteratively constructs the team-optimal estimate in (14) for dynamic parameters and achieves the error lower bound in (6).

4 Efficient and Optimal Distributed Online Learning

In Section 3, we show that over a tree network, the team-optimal estimate can be constructed using the disclosure of local estimates as

$$\hat{x}_{i,t} = \mathbb{E}\left[\mathcal{X}_t \middle| y_{i,t}, \hat{x}_{i,t-1}, \{\hat{x}_{j,t-1}\}_{j\in\mathbf{N}_i}\right].$$

Each local estimate $\hat{x}_{i,t}$ is linear in previous estimates $\hat{x}_{i,t-1}$ and $\{\hat{x}_{j,t-1}\}_{j\in\mathbf{N}_i}$. Therefore, instead of disclosing the local estimates, we constrain each agent to disclose the information that was not included in the old estimates. Then each agent extracts only the innovation terms, i.e. the new information in the disclosed data that the agent has not received before. Although this operation imposes more computational load on the agents, it significantly reduces the communication load on the network, which is more essential for highly-connected larger networks that require more power for the transmission of information [29].

We denote the innovation term extracted at the agent i from the data disclosed by the agent j at time t as $z_{j\to i,t-1}$. With this definition, we define the random vector collecting the previous estimate and the aggregated information on the agent i at time t as

$$\boldsymbol{d}_{i,t} = \begin{bmatrix} \mathcal{Y}_{i,t} & \hat{\mathcal{X}}_{i,t-1} & \mathcal{Z}_{j_1\to i,t-1} & \cdots & \mathcal{Z}_{j_{\pi_i}\to i,t-1} \end{bmatrix}^T, \tag{15}$$

so that we find the optimal estimate of the state with realizations of the elements in $\boldsymbol{d}_{i,t}$ as

$$\hat{x}_{i,t} = \mathbb{E}\left[\mathcal{X}_t \middle| \mathcal{Y}_{i,t} = y_{i,t}, \hat{\mathcal{X}}_{i,t-1} = \hat{x}_{i,t-1}, \{\mathcal{Z}_{j\to i,t-1} = z_{j\to i,t-1}\}_{j\in\mathbf{N}_i}\right].$$

Due to the state-space model defined in (2) and (3), all the parameters in (15) are jointly Gaussian. Hence, for the estimation of the next state at the agent

i, we have

$$\hat{x}_{i,t} = \alpha_{i,t}\hat{x}_{i,t-1} + \beta_{i,t}y_{i,t} + \sum_{j\in\mathbf{N}_i} c_{i,j}^{(j)} z_{j\to i,t-1}. \tag{16}$$

Using the estimation equation in (16), the information disclosed by the agent j at time t is given by

$$\begin{aligned} z_{j,t} &= \hat{x}_{j,t} - \alpha_{j,t}\hat{x}_{j,t-1} \\ &= \beta_{j,t}y_{j,t} + \sum_{k\in\mathbf{N}_j} c_{j,t}^{(k)} z_{k\to j,t-1}. \end{aligned} \tag{17}$$

Hence, we extract the innovation from the disclosed information on the agent i as

$$\begin{aligned} z_{j\to i,t} &= z_{j,t} - c_{j,t}^{(i)} z_{i\to j,t-1} \\ &= z_{j,t} - c_{j,t}^{(i)} z_{i,t-1} + c_{j,t}^{(i)} c_{i,t-1}^{(j)} z_{j\to i,t-2}. \end{aligned} \tag{18}$$

Remark 4.1 *Some of the previously diffused information are received after certain delays over the network due to multi-hops. Therefore, some of the received information will be the noisy versions of the previous instances of the underlying state. Due to the random walk model in* (2)*, the state noise on these previous instances will become correlated with the more recent observations. Hence, this situation requires a significantly more detailed approach than the existing methods [23].*

In order to calculate the parameters in the estimation recursion (16), we first need to calculate the auto-correlation matrix of $\boldsymbol{d}_{i,t}$ and the cross-correlation vector with the underlying state $\mathcal{X}_t$, where we define them as $\boldsymbol{\Sigma}_{dd_i,t}$ and $\boldsymbol{\Sigma}_{xd_i,t}$ respectively. We first calculate the terms of $\boldsymbol{\Sigma}_{xd_i,t}$ starting with

$$\begin{aligned} \mathbb{E}[\mathcal{X}_t\mathcal{Y}_{i,t}] &= \mathbb{E}[\mathcal{X}_t(\mathcal{X}_t + \mathcal{N}_{i,t})] = \mathbb{E}[\mathcal{X}_t^2] \\ &= \gamma^2\mathbb{E}[\mathcal{X}_{t-1}^2] + \sigma_w^2. \end{aligned} \tag{19}$$

Then, we calculate

$$
\begin{aligned}
\mathbb{E}[\mathcal{X}_t \hat{\mathcal{X}}_{i,t-1}] &= \mathbb{E}[\mathcal{X}_t(\alpha_{i,t-1}\hat{\mathcal{X}}_{i,t-2} + \beta_{i,t-1}\mathcal{Y}_{i,t-1} + \sum_{j\in \mathbf{N}_i} c_{i,t-1}^{(j)} \mathcal{Z}_{j\to i,t-2})] \\
&= \alpha_{i,t-1}\mathbb{E}[\mathcal{X}_t\hat{\mathcal{X}}_{i,t-2} + \beta_{i,t-1}\mathbb{E}[\mathcal{X}_t\mathcal{Y}_{i,t-1}]] + \sum_{j\in \mathbf{N}_i} c_{i,t-1}^{(j)}\mathbb{E}[\mathcal{X}_t\mathcal{Z}_{j\to i,t-2})] \\
&= \gamma^2\alpha_{i,t-1}\mathbb{E}[\mathcal{X}_{t-2}\hat{\mathcal{X}}_{i,t-2}] + \gamma\beta_{i,t-1}\mathbb{E}[\mathcal{X}_{t-1}^2] + \sum_{j\in \mathbf{N}_i} c_{i,t-1}^{(j)}\mathbb{E}[\mathcal{X}_t\mathcal{Z}_{j\to i,t-2}].
\end{aligned} \tag{20}
$$

In order to calculate (20), we also need to calculate $\mathbb{E}[\mathcal{X}_t\mathcal{Z}_{j\to i,t-2}]$. For that, we first introduce

$$
\boldsymbol{h}_{i,0} = \gamma \begin{bmatrix} \beta_{j_1,0} \\ \vdots \\ \beta_{j_{\pi_i},0} \end{bmatrix} \mathbb{E}[\mathcal{X}_0^2].
$$

Then, with this initialization, for any time t, we find

$$
\boldsymbol{h}_{i,t} = \gamma \begin{bmatrix} \beta_{j_1,t}\mathbb{E}[\mathcal{X}_t^2] + \boldsymbol{c}_{j_1,t}^T\boldsymbol{h}_{j_1,t-1} \\ \vdots \\ \beta_{j_{\pi_i},t}\mathbb{E}[\mathcal{X}_t^2] + \boldsymbol{c}_{j_{\pi_i},t}^T\boldsymbol{h}_{j_{\pi_i},t-1} \end{bmatrix} - \gamma \begin{bmatrix} c_{j_1,t}^{(i)} \\ \vdots \\ c_{j_{\pi_i},t}^{(i)} \end{bmatrix} \odot \begin{bmatrix} h_{j_1,t-1}^{(i)} \\ \vdots \\ h_{j_{\pi_i},t-1}^{(i)} \end{bmatrix},
$$

where $\boldsymbol{c}_{j_1,t} = \left[c_{j_1,1}^{(k_1)} \dots c_{j_1,1}^{(k_{\pi_{j_1}})}\right]^T, \quad k \in \mathbf{N}_{j_1}$. Note that $\boldsymbol{h}_{i,t}$ can also be expressed as

$$
\boldsymbol{h}_{i,t-1} = \begin{bmatrix} E[\mathcal{Z}_{j_1\to i,t-1}\mathcal{X}_t] \\ \vdots \\ E[\mathcal{Z}_{j_{\pi_i}\to i,t-1}\mathcal{X}_t] \end{bmatrix}.
$$

Using this notation, we obtain

$$\begin{aligned}\mathbb{E}[\mathcal{X}_t \mathcal{Z}_{j\to i,t-2}] &= \gamma \mathbb{E}[\mathcal{X}_{t-1}\mathcal{Z}_{j\to i,t-2}] \\ &= \gamma h_{i,t-1}^{(j)}.\end{aligned}$$

Therefore, we can finalize the calculation of $\mathbb{E}[\mathcal{X}_t \hat{\mathcal{X}}_{t-1}]$ as

$$\mathbb{E}[\mathcal{X}_t \hat{\mathcal{X}}_{t-1}] = \gamma^2 \alpha_{i,t-1}\mathbb{E}[\mathcal{X}_{t-2}\hat{\mathcal{X}}_{t-2}] + \gamma\beta_{i,t-1}\mathbb{E}[\mathcal{X}_{t-1}^2] + \gamma \boldsymbol{c}_{i,t-1}^T \boldsymbol{h}_{i,t-1}.$$

Additionally, we define the cross correlation term between the state and the estimate as

$$\begin{aligned}\tilde{\sigma}_{i,t}^2 &\triangleq \mathbb{E}[\mathcal{X}_t \hat{\mathcal{X}}_{i,t}] \\ &= \gamma\alpha_{i,t}\tilde{\sigma}_{i,t-1}^2 + \beta_{i,t}\mathbb{E}[\mathcal{X}_t^2] + \boldsymbol{c}_{i,t}^T \boldsymbol{h}_{i,t}\end{aligned}$$

and the variance for the underlying state as

$$\begin{aligned}\sigma_t^2 &\triangleq \mathbb{E}[\mathcal{X}_t^2] \\ &= \gamma^2\sigma_{t-1}^2 + \sigma_w^2,\end{aligned}$$

which concludes our calculation for the terms in $\boldsymbol{\Sigma}_{xd_i,t}$ such that

$$\begin{aligned}\boldsymbol{\Sigma}_{xd_i,t} &= \begin{bmatrix}\mathbb{E}[\mathcal{X}_t\mathcal{Y}_{i,t}] \ \mathbb{E}[\mathcal{X}_t\hat{\mathcal{X}}_{i,t-1}]\mathbb{E}[\mathcal{X}_t\mathcal{Z}_{j_1\to i,t-1}] \cdots \mathbb{E}[\mathcal{X}_t\mathcal{Z}_{j_{\pi_i}\to i,t-1}]\end{bmatrix}^T \\ &= \begin{bmatrix}\gamma^2\sigma_{t-1}^2 + \sigma_w^2 & \gamma\tilde{\sigma}_{i,t-1}^2 & \boldsymbol{h}_{i,t-1}^T\end{bmatrix}^T.\end{aligned}$$

Next, we calculate the terms of $\boldsymbol{\Sigma}_{dd_i,t}$. First, we have

$$\begin{aligned}\mathbb{E}[\mathcal{Y}_{i,t}^2] &= \mathbb{E}[(\mathcal{X}_t + \mathcal{N}_{i,t})^2] \\ &= \sigma_t^2 + \sigma_{n_i}^2.\end{aligned} \tag{21}$$

Then, for the term $\mathbb{E}[\hat{\mathcal{X}}_{i,t-1}\mathcal{Y}_{i,t}]$ we get

$$\begin{aligned}\mathbb{E}[\hat{\mathcal{X}}_{i,t-1}\mathcal{Y}_{i,t}] &= \mathbb{E}[\hat{\mathcal{X}}_{i,t-1}\mathcal{X}_t] \\ &= \gamma\tilde{\sigma}^2_{i,t-1}\end{aligned} \tag{22}$$

and note that we already found that $\mathbb{E}[\mathcal{Y}_{i,t}\mathcal{Z}_{j\to i,t-1}] = \boldsymbol{h}^{(j)}_{i,t-1}$. We then calculate the terms that include the random variable corresponding to the estimate of the previous state. We begin with defining

$$\begin{aligned}\hat{\sigma}^2_{i,t-1} &\triangleq \mathbb{E}[\hat{\mathcal{X}}^2_{i,t-1}] \\ &= \mathbb{E}\Bigg[\Big(\alpha_{i,t-}\hat{\mathcal{X}}_{i,t-2} + \beta_{i,t-2}\mathcal{Y}_{i,t-1} + \sum_{j\in\mathbf{N}_i} c^{(j)}_{i,t-1}\mathcal{Z}_{j\to i,t-2}\Big)^2\Bigg] \\ &= \alpha^2_{i,t-1}\underbrace{\mathbb{E}[\hat{\mathcal{X}}^2_{i,t-2}]}_{\hat{\sigma}^2_{i,t-2}} + 2\alpha_{i,t-1}\beta_{i,t-2}\underbrace{\mathbb{E}[\hat{\mathcal{X}}_{i,t-2}y_{i,t-1}]}_{\gamma\tilde{\sigma}^2_{i,t-2}} \\ &\quad + 2\alpha_{i,t-1}\sum_{j\in N_i} c^{(j)}_{i,t-1}E[\hat{\mathcal{X}}_{i,t-2}\mathcal{Z}_{j\to i,t-2}] + \beta^2_{i,t-2}\underbrace{\mathbb{E}[\mathcal{Y}^2_{i,t-1}]}_{\sigma^2_{t-1}+\sigma^2_{n_i}} \\ &\quad + 2\beta_{i,t-2}\underbrace{\sum_{j\in\mathbf{N}_i} c^{(j)}_{i,t-1}\mathbb{E}[\mathcal{Y}_{i,t-1}\mathcal{Z}_{j\to i,t-2}]}_{\gamma\boldsymbol{c}^T_{i,t-1}\boldsymbol{h}_{i,t-2}} + \mathbb{E}\Bigg[\Big(\sum_{j\in\mathbf{N}_i} c^{(j)}_{i,t-1}\mathcal{Z}_{j\to i,t-2}\Big)^2\Bigg]\end{aligned} \tag{23}$$

and $\hat{\sigma}^2_{i,0} = \beta^2_{i,0}(\sigma^2_0 + \sigma^2_{n_i})$. We need to calculate $\mathbb{E}[\hat{\mathcal{X}}_{i,t-2}\mathcal{Z}_{j\to i,t-2}]$ in order to complete the calculation of (23). For that, we introduce a more compact form of the term $\mathcal{Z}_{j\to i,t}$ as

$$
\begin{aligned}
\mathcal{Z}_{j\to i,t} =& \beta_{j,t}\mathcal{Y}_{j,t} + \sum_{k\in\mathbf{N}_j,k\neq i} c_{j,t}^{(k)}\Big(\beta_{k,t-1}\mathcal{Y}_{k,t-1} + \sum_{l\in\mathbf{N}_k,l\neq j} c_{k,t-1}^{(l)}\Big(\beta_{l,t-2}\mathcal{Y}_{l,t-2} + \sum\cdots\Big)\Big) \\
=& \Bigg[\overbrace{\beta_{j,t} + \frac{1}{\gamma}\sum_{k\in\mathbf{N}_j,k\neq i} c_{j,t}^{(k)}\Big(\beta_{k,t-1} + \frac{1}{\gamma}\sum_{l\in\mathbf{N}_k,l\neq j} c_{k,t-1}^{(l)}\Big(\beta_{l,t-2} + \frac{1}{\gamma}\sum\cdots\Big)\Big)}^{g_{i,t}^{(j)}}\Bigg]\mathcal{X}_t \\
&- \Bigg[\frac{1}{\gamma}\sum_{k\in\mathbf{N}_j,k\neq i} c_{j,t}^{(k)}\Big(\beta_{k,t-1} + \frac{1}{\gamma}\sum_{l\in\mathbf{N}_k,l\neq j} c_{k,t-1}^{(l)}\Big(\beta_{l,t-2} + \frac{1}{\gamma}\sum\cdots\Big)\Big)\Bigg]\mathcal{W}_{t-1} \\
&- \Bigg[\frac{1}{\gamma}\sum_{k\in N_j,k\neq i}\sum_{l\in N_k,l\neq j} c_{j,t}^{(k)}c_{k,t-1}^{(l)}\Big(\beta_{l,t-2} + \frac{1}{\gamma}\sum\cdots\Big)\Bigg]\mathcal{W}_{t-2} - \cdots - \Big[\cdots\Big]\mathcal{W}_{t-\kappa_i+1} + \text{(i.n.t.)},
\end{aligned}
\tag{24}
$$

where κ_i is the number of hops from the furthest agent and (i.n.t.) is the abbreviation of *independent noise terms*. We point out that the term $g_{i,t}^{(j)}$ can be calculated in a recursive form as in (24). Hence, using (24), we write the term $\mathcal{Z}_{j\to i,t}$ as

$$
\begin{aligned}
\mathcal{Z}_{j\to i,t} = g_{i,t}^{(j)}\mathcal{X}_t - (g_{i,t}^{(j)} - \beta_{j,t})\mathcal{W}_{t-1} - \gamma\Big(g_{i,t}^{(j)} - \beta_{j,t} - \frac{1}{\gamma}\sum_{k\in N_j,k\neq i} c_{j,t}^{(k)}\beta_{k,t-1}\Big)\mathcal{W}_{t-2} \\
- \cdots - \Big(\cdots\Big)\mathcal{W}_{t-\kappa_i+1} + \text{(i.n.t.)}
\end{aligned}
\tag{25}
$$

and we obtain

$$
\mathbb{E}[\hat{\mathcal{X}}_{i,t}\mathcal{Z}_{j\to i,t}] = g_{i,t}^{(j)}\mathbb{E}[\hat{\mathcal{X}}_{i,t}\mathcal{X}_t] - (g_{i,t}^{(j)} - \beta_{j,t})\mathbb{E}[\hat{\mathcal{X}}_{i,t}\mathcal{W}_{t-1}] - \cdots - \Big(\cdots\Big)\mathbb{E}[\hat{\mathcal{X}}_{i,t}\mathcal{W}_{t-\kappa_i+1}].
$$

The state noise $\mathcal{W}_t$ is independent from previous states and we express the term $\mathcal{X}_{i,t}$ as

$$\begin{aligned}\hat{\mathcal{X}}_{i,t} = \beta_{i,t}\mathcal{W}_{t-1} &+ \Big(\alpha_{i,t}\beta_{i,t-1} + \sum_{j\in N_i} c_{i,t}^{(j)}\beta_{j,t-1}\Big)\mathcal{W}_{t-2} \\ &+ \Big(\alpha_{i,t}\alpha_{i,t-1}\beta_{i,t-2} + \alpha_{i,t}\sum_{j\in N_i} c_{i,t}^{(j)}\beta_{j,t-2} + \sum_{j\in N_i}\sum_{k\in N_j,k\neq i} c_{i,t}^{(j)}c_{j,t-1}^{(k)}\beta_{k,t-2}\Big)\mathcal{W}_{t-3} \\ &+ \cdots + \Big(\cdots\Big)\mathcal{W}_{t-\kappa_i+1} + \text{i.n.t.}.\end{aligned} \tag{26}$$

Therefore, we conclude that

$$\mathbb{E}[\hat{\mathcal{X}}_{i,t}\mathcal{Z}_{j\to i,t}] = g_{i,t}^{(j)}\mathbb{E}[\hat{\mathcal{X}}_{i,t}\mathcal{X}_t] + A\sigma_w^2,$$

where A is calculated according to the recursions in (25) and (26).

Finally, we calculate the remaining terms of $\mathbf{\Sigma}_{dd_i,t}$ as

$$\begin{aligned}&\mathbb{E}\Big[\Big(\mathcal{Z}_{j\to i,t-1}\Big)^2\Big] = (g_{i,t-1}^{(j)^2})\sigma_{t-1}^2 \\ &+ \Big[\Big((g_{i,t-1}^{(j)} - \beta_{j,t-1})^2 - \Big(g_{i,t-1}^{(j)} - \beta_{j,t-1} - \frac{1}{\gamma}\sum_{k\in N_j,k\neq i} c_{j,t-1}^{(k)}\beta_{k,t-2}\Big)\Big)^2 - B_{j,4}^2 - B_{j,5}^2\cdots - B_{j,\kappa_i}^2\Big]\sigma_w^2 \\ &+ (\beta_{j,t-1})^2\sigma_{n_j} + \sum_{k\in N_j,k\neq i}(c_{j,t-1}^{(k)}\beta_{k,t-2})\sigma_{n_k}^2 + \sum_{k\in N_j,k\neq i}\sum_{l\in N_k,l\neq j}(c_{j,t-1}^{(k)}c_{k,t-2}^{(l)}\beta_{k,t-2}\beta_{l,t-3})^2\sigma_{n_l}^2 + \cdots \\ &+ \sum_{k\in N_j,k\neq i}\sum_{l\in N_k,l\neq j}\cdots\sum_{r\in N_i^{(\kappa_i-3)}}\sum_{s\in N_i^{(\kappa_i-2)}}\sum_{m\in N_i^{(\kappa_i-1)},m\neq r}(c_{j,t-1}^{(k)}c_{k,t-2}^{(l)}\cdots c_{s,t-\kappa_i}^{(m)}\beta_{k,t-2}\beta_{l,t-3}\cdots\beta_{m,t-\kappa_i})^2\sigma_m^2\end{aligned} \tag{27}$$

and

$$\begin{aligned}\mathbb{E}\Big[\mathcal{Z}_{j_o\to i,t-1}\mathcal{Z}_{j_p\to i,t-1}\Big] &= g_{i,t-1}^{j_o}g_{i,t-1}^{j_p}\sigma_{t-1}^2 + \Big[(g_{i,t-1}^{(j_o)} - \beta_{j_o,t-1})(g_{i,t-1}^{(j_p)} - \beta_{j_p,t-1}) \\ &- \Big(g_{i,t-1}^{(j_o)} - \beta_{j_o,t-1} - \frac{1}{\gamma}\sum_{k\in N_{j_o},k\neq i} c_{j_o,t-1}^{(k)}\beta_{k,t-2}\Big)\Big(g_{i,t-1}^{(j_p)} - \beta_{j_p,t-1} - \frac{1}{\gamma}\sum_{k\in N_{j_p},k\neq i} c_{j_p,t-1}^{(k)}\beta_{k,t-2}\Big) \\ &- B_{j_o,4}B_{j_p,4} - \cdots - B_{j_o,\kappa_i}B_{j_p,\kappa_i}\Big]\sigma_w^2,\end{aligned} \tag{28}$$

where $o, p \in \{1, \ldots, \pi_i\}, o \neq p$ and $B_{j,t}$ for $t = 4, \ldots, \kappa - i$ represents the remaining recursive terms derived in (25).

Next, we recursively calculate the parameters in (16). We define the vector containing the parameters as

$$\boldsymbol{P} \triangleq \left[\tilde{\alpha}_{i,t} \;\; \beta_{i,t} \;\; c_{i,t}^{(j_1)} \;\; \cdots \;\; c_{i,t}^{(j_{\pi_i})}\right]^T.$$

In (13), all the conditioned parameters are jointly Gaussian with the state. Therefore, we calculate the parameter vector as $\boldsymbol{P} = \boldsymbol{\Sigma}_{dd_i,t}^{-1}\boldsymbol{\Sigma}_{xd_i,t}$. The estimation and the variance recursions are given by

$$\hat{x}_{i,t} = \gamma\hat{x}_{i,t-1} + \beta_{i,t}\Big(y_{i,t} - \gamma\hat{x}_{i,t-1}\Big) + \sum_{j \in N_i} c_{i,t}^{(j)}\Big(z_{j\to i,t-1} - g_{i,t}^{(j)}\hat{x}_{i,t-1}\Big),$$
$$\hat{\sigma}_{i,t}^2 = \gamma^2\hat{\sigma}_{i,t-1}^2 + \sigma_w^2 - \boldsymbol{\Sigma}_{xd_i,t}^T\boldsymbol{\Sigma}_{dd_i,t}^{-1}\boldsymbol{\Sigma}_{xd_i,t}.$$

Hence, for the parameter $\alpha_{i,t}$ in (16), we have

$$\alpha_{i,t} = \gamma - \gamma\beta_{i,t} - \sum_{j \in N_i} c_{i,t}^{(j)} g_{i,t}^{(j)},$$

which finalizes the efficient and optimal distributed online learning algorithm. We give the detailed pseudo-code of the overall algorithm in Algorithm 1.

In the following, we provide numerical examples to evaluate the performance of our algorithm against several other distributed estimation algorithms under different scenarios.

5 Simulations

In this section, we study the performance of the proposed algorithm under different scenarios. For the network structure, we consider a depth 2 tree-network. Each agent i observes a noise corrupted version $y_t \in \mathbb{R}$ of an under-

Algorithm 1 The Efficient and Optimal Distributed Online Learning Algorithm (EODL)

for $i = 1$ to m **do**
  $\hat{x}_{i,0} = \bar{x}$
  $\hat{\sigma}^2_{i,0} = \sigma^2_0$
end for
for $t \geq 1$ **do**
  for $i = 1$ to m **do**
    Receive $\{z_{j,t-1}\}_{j \in \mathbf{N}_i}$
    Extract Innovation
    $z_{j \to i,t-1} = z_{j,t-1} - c^{(i)}_{j,t-1} + c^{(i)}_{j,t-1} c^{(j)}_{i,t-2} z_{j \to i,t-3}$
    Calculate $\mathbf{\Sigma}_{xd_i,t}$,$\mathbf{\Sigma}_{dd_i,t}$
    Find Parameters:
    $\boldsymbol{P} \triangleq [\tilde{\alpha}_{i,t}\ \beta_{i,t}\ c^{(j_1)}_{i,t}\ \cdots\ c^{(j_{\pi_i})}_{i,t}]^T$
    $\boldsymbol{P} \leftarrow \mathbf{\Sigma}^{-1}_{dd_i,t} \mathbf{\Sigma}_{xd_i,t}$
    $\alpha_{i,t} = \gamma - \gamma\beta_{i,t} - \sum_{j \in N_i} c^{(j)}_{i,t} g^{(j)}_{i,t}$
    Update:
      $\hat{x}_{i,t} = \alpha_{i,t}\hat{x}_{i,t-1} + \beta_{i,t} y_{i,t}$
        $+ \sum_{j \in N_i} c^{(j)}_{i,t} z_{j \to i,t-1}$
      $\hat{\sigma}^2_{i,t} = \gamma^2 \hat{\sigma}^2_{i,t-1} + \sigma^2_w - \mathbf{\Sigma}^T_{xd_i,t} \mathbf{\Sigma}^{-1}_{dd_i,t} \mathbf{\Sigma}_{xd_i,t}$
  end for
end for

lying state $x_t \in \mathbb{R}$, where it evolves according to the random walk model in (2) with $\gamma = 0.98$. The state noise w_t is driven by a Gaussian process, with zero mean and variance $\sigma^2_w = 0.025$. The observation noise is also zero-mean white Gaussian random process.

We use the terminal cost function for measuring the team performance of different algorithms. The terminal cost is a function of the time horizon T and defined as [26]

$$J(T) = \sum_{i=1}^{m} \mathbb{E} \left|\left|\mathcal{X}_T - \hat{x}_{i,T}\right|\right|^2, \tag{29}$$

which represents the impact of the final estimate on the horizon. We use ensemble average over 200 experiments in order to approximate the cost measure in (29).

We compare the performance of the proposed algorithm with the diffusion least mean squares (D-LMS), the diffusion recursive least squares (D-RLS) algorithm and the diffusion implementation of the Kalman filtering algorithm (D-Kalman) under different settings [10, 12, 22]. We also use a distributed consensus algorithm in our comparison framework [35]. We implement the diffusion based distributed algorithms with the adapt-then-combine (ATC) technique, where each agent first makes an estimate based on its local observation and discloses its estimate [35]. Then, the agents decide on their final estimate by combining the local and the received estimates. For the combination step, we use the Metropolis rule, where the combination weight $\lambda_{i,j}$ for the estimate coming to the agent i from the agent j is calculated as

$$\lambda_{i,j} = \begin{cases} \frac{1}{\max(\mathbf{N}_i, \mathbf{N}_j)} & \text{if } i \neq j \text{ are linked,} \\ 0 & \text{for } i \text{ and } j \text{ not linked,} \\ 1 - \sum_{j \in \mathcal{N}_i \setminus i} \lambda_{i,j} & \text{for } i = j. \end{cases}$$

We set the learning rates of the diffusion LMS and the consensus algorithm to $\mu = 0.2$. We select this learning rate so that these algorithms do not follow the observation noise and capture the underlying parameter. Also, we set the memory parameter of the diffusion RLS algorithm to $\eta = 0.3$. Note that we set the memory parameter of RLS algorithm to a relatively small value in order to put more emphasis on the recent observations and estimates. We choose this parameter so that the diffusion RLS algorithm converges fast, but still be able to track the underlying dynamic parameter.

In Fig.4, we compare the algorithms under a space-invariant noise over the network, where each agent experiences the same level of disturbance. We select the random walk and observation model parameter so that each agent experiences 0.5dB signal-to-noise ratio (SNR). We observe that the proposed algorithm (EODL) achieves a superior performance regarding the global finite horizon MSE measure and the convergence rate compared to the other dis-

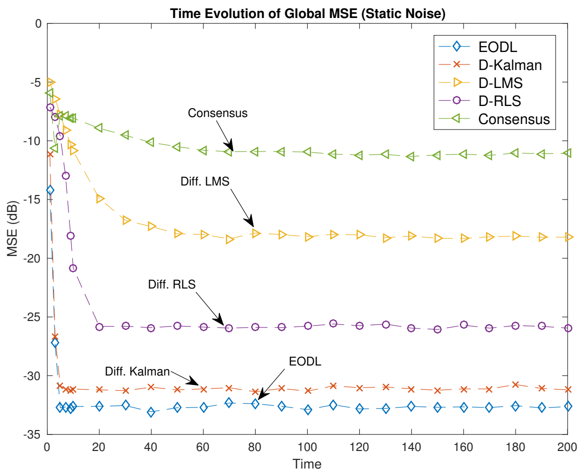

Fig. 4. Comparison of the global MSE of the algorithms under space-invariant noise with $\gamma = 0.98$.

tributed estimation algorithms. The consensus algorithm performs the worst since it has a decaying learning rate as the nodes reach to a consensus in time and the network loses its adaptation capabilities against a dynamic parameter.

In another scenario, we evaluate the performance of the algorithms under a space-variant noise statistics over the network. For this case, we randomly sample the standard deviation of the observation noise of the each agent from a folded Gaussian distribution so that signal-to-noise ratio of the network will be around 0.5dB. In this case randomness is involved and some of the agents will experience higher(lower) SNR levels. In Fig.5, we compare the algorithms for the space-variant noise case. Note that the algorithms perform better than the space-invariant noise statistics case. This is because, in this case, some of the agents experience smaller noise levels while some others experience higher, but through the communication between the agents, they all benefit from the estimates of the agents having better observation channels. We also emphasize that the EODL algorithm even performs better in this case since it utilizes the optimal information disclosure and the estimate construction. Furthermore, we observe a similar performance between the compared algorithms, where the EODL algorithm achieves a superior performance regarding the global

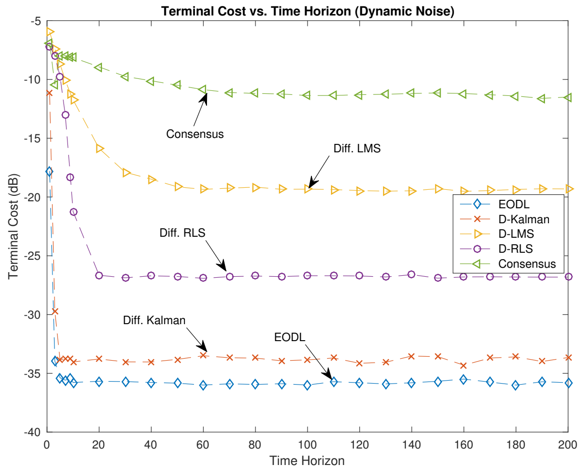

Fig. 5. Comparison of the global MSE of the algorithms under space-variant noise with $\gamma = 0.98$.

MSE measure and the convergence rate compared to the other distributed estimation schemes.

We also investigate the effect of the random walk parameter γ with a simulation under the space-variant noise framework, thus we select $\gamma = 1$ for another set of simulations. We emphasize that in this case the random walk model diverges, however, as we will observe, our algorithm provides a bounded estimation MSE. In Fig.6, we present the results for the case of no cooperation between the nodes and for the case the cooperation occurs. We observe that the D-LMS, D-RLS and the consensus networks become unstable, even if individual nodes are stable and able to track the underlying parameter. Only the D-Kalman and the EODL networks are able to achieve the convergence for $\gamma = 1$ case. We also emphasize that, with the EODL algorithm, we produce the optimal parameters and the combination weights for the network in contrast to the D-LMS, D-RLS and consensus algorithms, where we need to select their parameters beforehand. Therefore, the proposed algorithm overcomes the issue of parameter selection and provides more stable solution for this scenario.

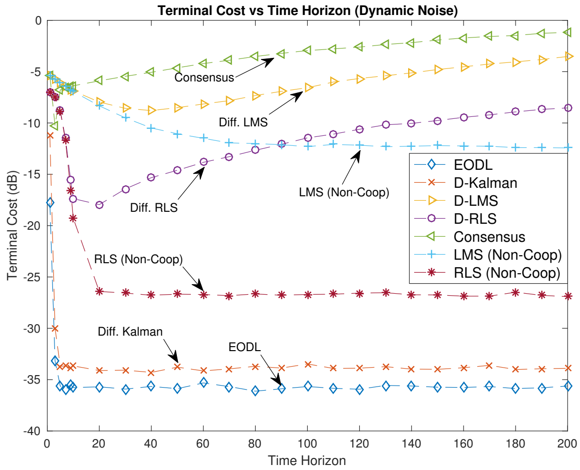

Fig. 6. Comparison of the global MSE of the algorithms under space-variant noise with $\gamma = 1$. Even if D-LMS and D-RLS are stable when there is no cooperation among nodes, the networks are diverging when they cooperate

6 Conclusion

In this paper, we introduce a novel approach for the distributed estimation of dynamically changing parameters. We first construct a framework for the estimation of dynamic parameters by a team of distributed agents. Here, we provide a lower bound on the estimation error of the team of agents in the MSE sense. We prove that the lower bound can be achieved for any arbitrary network when the agents disclose the stamped observations. We also show that this method imposes huge communication loads and requires excessive storage on the agents. Therefore, we introduced an efficient method where the agents only disclose the "new information" they have collected. We prove that the error lower bound in this case can only be achieved over certain network topologies. We introduce an algorithm to recursively extract the innovations from the disclosed information and construct the optimal estimates. Through series of simulations over different scenarios, we illustrate the significant performance improvements introduced by our algorithm with respect to the state of the art methods.

Acknowledgements

This work is supported by TUBITAK Contract No 117E153.

References

[1] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, Z. J. Towfic, Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior, IEEE Signal Processing Magazine 30 (3) (2013) 155–171.

[2] S. Shahrampour, S. Rakhlin, A. Jadbabaie, Online learning of dynamic parameters in social networks, in: Advances in Neural Information Processing Systems, 2013.

[3] Z. Zheng, Z. Liu, M. Huang, Diffusion least mean square/fourth algorithm for distributed estimation, Signal Processing 134 (2017) 268–274.

[4] S. Xu, K. Doğançay, H. Hmam, Distributed pseudolinear estimation and uav path optimization for 3d aoa target tracking, Signal Processing 133 (2017) 64–78.

[5] A. Bertrand, M. Moonen, Distributed signal estimation in sensor networks where nodes have different interests, Signal Processing 92 (7) (2012) 1679–1690.

[6] D. D. Chaudhary, S. P. Nayse, L. M. Waghmare, Application of wireless sensor networks for greenhouse parameter control in precision agriculture, International Journal of Wireless & Mobile Networks (IJWMN) Vol 3 (1) (2011) 140–149.

[7] D. Estrin, L. Girod, G. Pottie, M. Srivastava, Instrumenting the world with wireless sensor networks, in: Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on, Vol. 4, IEEE, 2001, pp. 2033–2036.

[8] C. G. Lopes, A. H. Sayed, Diffusion least-mean squares over adaptive networks: Formulation and performance analysis, IEEE Transactions on Signal Processing 56 (7) (2008) 3122–3136.

[9] N. Takahashi, I. Yamada, A. H. Sayed, Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis, IEEE Transactions on Signal Processing 58 (9) (2010) 4795–4810.

[10] F. S. Cattivelli, C. G. Lopes, A. H. Sayed, Diffusion recursive least-squares for distributed estimation over adaptive networks, IEEE Transactions on Signal Processing 56 (5) (2008) 1865–1877.

[11] S. Y. Tu, A. H. Sayed, Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks, IEEE Transactions on Signal Processing 60 (12) (2012) 6217–6234.

[12] F. S. Cattivelli, C. G. Lopes, A. H. Sayed, Diffusion strategies for distributed kalman filtering: formulation and performance analysis, Proc. Cognitive Information Processing (2008) 36–41.

[13] G. Soatti, M. Nicoli, S. Savazzi, U. Spagnolini, Consensus-based algorithms for distributed network-state estimation and localization, IEEE Transactions on Signal and Information Processing over Networks 3 (2) (2017) 430–444.

[14] I. D. Schizas, G. Mateos, G. B. Giannakis, Distributed lms for consensus-based in-network adaptive processing, IEEE Transactions on Signal Processing 57 (6) (2009) 2365–2382.

[15] V. Matta, P. Braca, S. Marano, A. H. Sayed, Distributed detection over adaptive networks: Refined asymptotics and the role of connectivity, IEEE Transactions on Signal and Information Processing over Networks 2 (4) (2016) 442–460.

[16] S. Joshi, S. Boyd, Sensor selection via convex optimization, IEEE Transactions on Signal Processing 57 (2) (2009) 451–462.

[17] M. Shamaiah, S. Banerjee, H. Vikalo, Greedy sensor selection under channel uncertainty, IEEE Wireless Communications Letters 1 (4) (2012) 376–379.

[18] H. Durrant-Whyte, M. Stevens, E. Nettleton, Data fusion in decentralised sensing networks, in: 4th International Conference on Information Fusion, 2001, pp. 302–307.

[19] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, Z. J. Towfic, Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior, IEEE Signal Processing Magazine 30 (3) (2013) 155–171.

[20] L. Xiao, S. Boyd, S. Lall, A scheme for robust distributed sensor fusion based on average consensus, in: Proceedings of the 4th international symposium on Information processing in sensor networks, IEEE Press, 2005, p. 9.

[21] J. Fernandez-Bes, L. A. Azpicueta-Ruiz, J. Arenas-García, M. T. M. Silva, Distributed estimation in diffusion networks using affine least-squares combiners, Digital Signal Processing 36 (2015) 1–14.

[22] F. S. Cattivelli, A. H. Sayed, Diffusion lms strategies for distributed estimation, IEEE Transactions on Signal Processing 58 (3) (2010) 1035–1048.

[23] M. O. Sayin, N. D. Vanli, I. Delibalta, S. S. Kozat, Optimal and efficient distributed online learning for big data, in: Big Data (BigData Congress), 2015 IEEE International Congress on, IEEE, 2015, pp. 126–133.

[24] D. Acemoglu, A. Nedic, A. Ozdaglar, Convergence of rule-of-thumb learning rules in social networks, in: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, IEEE, 2008, pp. 1714–1720.

[25] R. Frongillo, G. Schoenebeck, O. Tamuz, Social learning in a changing world, Internet and Network Economics (2011) 146–157.

[26] M. O. Sayin, S. S. Kozat, T. Başar, Team-optimal distributed mmse estimation in general and tree networks, Digital Signal Processing 64 (2017) 83–95.

[27] A. H. Sayed, Fundamentals of adaptive filtering, John Wiley & Sons, 2003.

[28] B. D. O. Anderson, J. B. Moore, Optimal filtering, Englewood Cliffs 21 (1979) 22–95.

[29] M. O. Sayin, S. S. Kozat, Compressive diffusion strategies over distributed networks for reduced communication load, IEEE Transactions on Signal Processing 62 (20) (2014) 5308–5323.

[30] M. O. Sayin, S. S. Kozat, Single bit and reduced dimension diffusion strategies over distributed networks, IEEE Signal Processing Letters 20 (10) (2013) 976–979.

[31] P. Humblet, A distributed algorithm for minimum weight directed spanning trees, IEEE Transactions on Communications 31 (6) (1983) 756–762.

[32] M. Khan, G. Pandurangan, V. S. A. Kumar, Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks, IEEE Transactions on Parallel and Distributed Systems 20 (1) (2009) 124–139.

[33] D. Peleg, Distributed computing: a locality-sensitive approach, SIAM, 2000.

[34] M. Elkin, An unconditional lower bound on the time-approximation trade-off for the distributed minimum spanning tree problem, SIAM Journal on Computing 36 (2) (2006) 433–456.

[35] C. G. Lopes, A. H. Sayed, Diffusion least-mean squares over adaptive networks, in: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, Vol. 3, IEEE, 2007, pp. III–917.