Partial Diffusion Kalman Filtering for Distributed State Estimation in Multiagent Networks

Vahid Vahidpour, Amir Rastegarnia, *Member, IEEE*, Azam Khalili, and Saeid Sanei, *Senior Member, IEEE*

***Abstract*—Many problems in multiagent networks can be solved through distributed learning (state estimation) of linear dynamical systems. In this paper, we develop a partial-diffusion Kalman filtering (PDKF) algorithm, as a fully distributed solution for state estimation in the multiagent networks with limited communication resources. In the PDKF algorithm, every agent (node) is allowed to share only a subset of its intermediate estimate vectors with its neighbors at each iteration, reducing the amount of internode communications. We analyze the stability of the PDKF algorithm and show that the algorithm is stable and convergent in both mean and mean-square senses. We also derive a closed-form expression for the steady-state mean-square deviation criterion. Furthermore, we show theoretically and by numerical examples that the PDKF algorithm provides a trade-off between the estimation performance and the communication cost that is extremely profitable.**

***Index Terms*—Diffusion strategy, distributed estimation, Kalman filtering, partial update, state estimation.**

I. Introduction

Distributed learning of linear dynamical systems is required in many applications related to multiagent networks, machine learning, big data, sensor networks, and smart grid [1]. We consider a multiagent network where the agents (nodes) collaborate to estimate the state of a linear dynamic system. In previous studies, some decentralized Kalman filtering algorithms such as parallel information filter [2], distributed Kalman filter with embedded consensus filter [3]–[7], and distributed Kalman filter with weighted averaging [8], [9] have been developed for distributed state estimation. Here, we focus on diffusion Kalman filtering (DKF) algorithm, originally reported in [10], which relies on the diffusion adaptive network strategy. This is because diffusion adaptive networks deliver better performance for distributed optimization and learning problems, compared with those given by the consensus strategies [11]. In the DKF algorithm, the nodes communicate only with their immediate neighbors, and the information is diffused across the entire network through a sequence of Kalman iterations and data aggregation. Unfortunately, the successful implementation of such a strategy requires considerable information exchange among the nodes, which may affect their performance in practical applications where the communication resource is restricted. Therefore, lowering the amount of information exchange among the neighboring nodes, while keeping the benefits of cooperation, is of practical importance.

When the observed data by agents follow a *linear regression model*, several methods have been reported to reduce the communication cost without any significant degradation of estimation in diffusion algorithms, such as reducing the dimension of estimates [12]–[14], selecting a subset of entries of the estimates [15]–[18], and set-membership filtering [19]–[21] or partial updating [22]–[26]. Regarding the linear dynamic model, algorithms whose ultimate goal is to reduce the communication cost have also been reported in the literatures [27]–[32]. In [33]–[35], some distributed Kalman filters have been reported to estimate a common state in a sensor network. However, none of these algorithms rely on the adaptive diffusion strategy.

Solutions to distributed learning systems suffer from lack of continuous (full) communications between the nodes. Therefore, the objective of this paper is to enable distributed learning, including state estimation, under limited communication resources. Thus, in this brief, we propose a PDKF algorithm as a fully distributed and reduced-communication solution for state estimation in the multiagent networks. In the PDKF algorithm, each agent uses an entry selection mechanism to exchange and diffuse only a part of its estimate state vector with its direct neighbors at each time update. Using the diffusion adaptation alongside partial updating enables the proposed algorithm to provide a trade-off between estimation performance and communication cost, which is required in many practical applications.

The contributions of this brief are summarized as follows.

1) A PDKF algorithm is proposed to solve the distributed state estimation problem, which is amenable to implementation in resource-constrained multiagent networks. The provided theoretical analysis and numerical simulations show that the PDKF algorithm exhibits an acceptable estimation performance, while the utilization of communication resources is kept low.
2) The stability of PDKF algorithm in both mean and mean-square senses under certain statistical conditions is investigated.
3) A closed-form expression for mean-square deviation (MSD) is derived, which explains how the PDKF algorithm performs in the steady state. The provided numerical tests validate the theoretical derivations.
4) Theoretically, and by numerical examples, it is shown that the PDKF algorithm provides a trade-off between estimation performance and communication cost.

Notation: We adopt small boldface letters for vectors and bold capital letters for matrices. The notation diag$\{\cdot,\cdot\}$ is a block matrix where the diagonal terms are given as the arguments of diag$\{\cdot\}$. The notation col$\{p, q\}$ denotes a column vector with entries p and q stacked on top of each other. We use $\|\mathbf{x}\|^2_{\boldsymbol{\Sigma}} = \mathbf{x}^T\boldsymbol{\Sigma}\mathbf{x}$ for the weighted square norm of $\mathbf{x}$. We use $\otimes$ and $\mathbb{1}$ to denote the Kronecker product and a column vector with unity entries, respectively. We use the notation (p mod q) as the modulus operator, which returns the remainder after division of p by q.

II. Preliminaries and Problem Formulation

A. Network Model and System Description

Consider a network with N agents represented by the set $\mathcal{N} = \{1, 2, \ldots, N\}$. Only single-hop communications are allowed, i.e., agent k can only communicate with the agents in its neighborhood $\mathcal{N}_k$. The neighborhood of agent k is defined as the set of

Manuscript received February 21, 2018; revised October 20, 2018, January 3, 2019, and January 9, 2019; accepted February 9, 2019. *(Corresponding author: Amir Rastegarnia.)*

V. Vahidpour, A. Rastegarnia, and A. Khalili are with the Department of Electrical Engineering, Malayer University, Malayer 65719-95863, Iran (e-mail: v.vahidpour@ieee.org; rastegarnia@malayeru.ac.ir; khalili@malayeru.ac.ir).

S. Sanei is with the School of Science and Technology, Nottingham Trent University, Nottingham NG1 4FQ, U.K. (e-mail: saeid.sanei@ntu.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2019.2899052

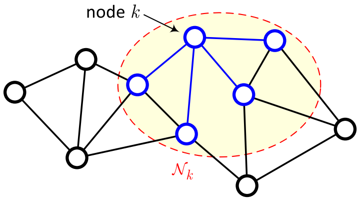

Fig. 1. Typical connected network with N agents (nodes).

all neighbors of agent k including itself (see Fig. 1). The network topology is a connected graph, i.e., there exists at least one path between every pair of agents. Note that the connectivity condition ensures that the information from an arbitrary agent can diffuse through the entire network [36].

At time instant i, each node k measures $\mathbf{y}_{k,i} \in \mathbb{R}^P$ of the true state $\mathbf{x}_i \in \mathbb{R}^M$ of a linear dynamic system as

$$\mathbf{y}_{k,i} = \mathbf{H}_{k,i}\mathbf{x}_i + \mathbf{v}_{k,i} \tag{1}$$

where $\mathbf{H}_{k,i} \in \mathbb{R}^{P\times M}$ and $\mathbf{v}_{k,i}$ denote, respectively, the local observation matrix and observation noise vector. The state vector $\mathbf{x}_i$ evolves according to

$$\mathbf{x}_{i+1} = \mathbf{F}_i\mathbf{x}_i + \mathbf{G}_i\mathbf{n}_i \tag{2}$$

where $\mathbf{F}_i \in \mathbb{R}^{M\times M}$, $\mathbf{G}_i \in \mathbb{R}^{M\times M}$, and $\mathbf{n}_i \in \mathbb{R}^M$ denote, respectively, the model matrix, the state noise matrix, and the state noise vector. For every agent, the objective is to estimate $\mathbf{x}_i$ by collaborating with other agents. For the state-space model described by (1) and (2), it is customary to make the following set of assumptions.

Assumption 1:

1) The measurement noises $\mathbf{v}_{k,i}$ are zero mean and spatially and temporally uncorrelated with covariance matrices given by $\mathbb{E}[\mathbf{v}_{k,i}\mathbf{v}_{l,j}^T] = \delta_{ij}\delta_{kl}\mathbf{R}_{k,i}$, where $\mathbf{R}_{k,i} > 0$ (δ_{ij} is the Kronecker delta.)
2) The state noise vectors $\{\mathbf{n}_i\}$ are zero mean and temporally uncorrelated with covariance matrices $\mathbb{E}[\mathbf{n}_i\mathbf{n}_j^T] = \delta_{ij}\mathbf{Q}_i$.
3) The noise signals $\mathbf{v}_{k,i}$ and $\mathbf{n}_i$ are uncorrelated for all i.
4) The initial state $\mathbf{x}_0$ is zero mean with covariance matrix $\mathbb{E}[\mathbf{x}_0\mathbf{x}_0^T] = \mathbf{\Pi}_0 > 0$ and uncorrelated with the noise signals $\mathbf{v}_{k,i}$ and $\mathbf{n}_i$ for all i and k.

B. Diffusion Kalman Filter Algorithm

As mentioned earlier, the DKF algorithm is an effective tool to solve the distributed state estimation problems. Before proceeding further, we define $\hat{\mathbf{x}}_{k,i|j}$ as the local estimator of $\mathbf{x}_i$ that node k computes at time i based on the local observations and information up to and including time j. We further use $\tilde{\mathbf{x}}_{k,i|j} = \mathbf{x}_i - \hat{\mathbf{x}}_{k,i|j}$ to denote the estimation error at node k and $\mathbf{P}_{k,i|j}$ to denote the covariance matrix of $\tilde{\mathbf{x}}_{k,i|j}$. Then, the DKF algorithm in its time and measurement update form is given in Algorithm 1. The algorithm starts with $\hat{\mathbf{x}}_{k,0|-1} = \mathbf{0}$ and $\mathbf{P}_{k,0|-1} = \mathbf{\Pi}_0$, where $\mathbf{P}_{k,0|-1} \in \mathbb{R}^{M\times M}$. The symbol ← denotes a sequential assignment. The algorithm consists of two steps: the *incremental update* and *diffusion update*. In the incremental update step, first, node k communicates with its neighbors and incorporates the real-time measurement information denoted by $\{\mathbf{y}_{l,i}, \mathbf{H}_{l,i}, \mathbf{R}_{l,i}\}$ into $\{\boldsymbol{\psi}_{k,i}, \mathbf{P}_{k,i}\}$. Then, each node performs KF using the available data to obtain the intermediate estimates $\boldsymbol{\psi}_{k,i}$ as (3). In the diffusion step, the nodes share intermediate estimates $\boldsymbol{\psi}_{k,i}$ and then compute a convex combination of intermediate estimates to obtain the local estimate $\hat{\mathbf{x}}_{k,i|i}$ as (4). The scalars $\{c_{lk}\}$ are nonnegative coefficients that satisfy

$$\sum_{l=1}^{N} c_{lk} = 1, \quad c_{lk} = 0 \text{ if } l \notin \mathcal{N}_k, \ \forall l, k. \tag{5}$$

Algorithm 1 DKF [10]

```
Initialization: $\hat{\mathbf{x}}_{k,0|-1} = 0$ and $\mathbf{P}_{k,0|-1} = \mathbf{\Pi}_0$
For every time instant $i$, every node $k$ computes
Step1: Incremental Update
$\boldsymbol{\psi}_{k,i} \leftarrow \hat{\mathbf{x}}_{k,i|i-1}$
$\mathbf{P}_{k,i} \leftarrow \mathbf{P}_{k,i|i-1}$
for $l \in \mathcal{N}_k$ do
    $\mathbf{R}_{e,i} \leftarrow \mathbf{R}_{l,i} + \mathbf{H}_{l,i}\mathbf{P}_{k,i}\mathbf{H}_{l,i}^T$
    $\boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k,i} + \mathbf{P}_{k,i}\mathbf{H}_{l,i}^T\mathbf{R}_{e,i}^{-1}[\mathbf{y}_{l,i} - \mathbf{H}_{l,i}\boldsymbol{\psi}_{k,i}]$
    $\mathbf{P}_{k,i} \leftarrow \mathbf{P}_{k,i} - \mathbf{P}_{k,i}\mathbf{H}_{l,i}^T\mathbf{R}_{e,i}^{-1}\mathbf{H}_{l,i}\mathbf{P}_{k,i}$    (3)
end for
Step2: Diffusion Update
    $\hat{\mathbf{x}}_{k,i|i} \leftarrow \sum_{l\in\mathcal{N}_k} c_{lk}\boldsymbol{\psi}_{l,i}$
    $\mathbf{P}_{k,i|i} \leftarrow \mathbf{P}_{k,i}$
    $\hat{\mathbf{x}}_{k,i+1|i} = \mathbf{F}_i\hat{\mathbf{x}}_{k,i|i}$
    $\mathbf{P}_{k,i+1|i} = \mathbf{F}_i\mathbf{P}_{k,i|i}\mathbf{F}_i^T + \mathbf{G}_i\mathbf{Q}_i\mathbf{G}_i^T$    (4)
```

The coefficients $\{c_{lk}\}$ are free-weighting parameters and their selection influences the algorithm performance. If the coefficients form an $N \times N$ matrix $\mathbf{C} \triangleq [c_{lk}]$, named combination matrix, the condition above conveys that all the columns of $\mathbf{C}$ add up to unity, i.e., $\mathbf{C}^T\mathbb{1} = \mathbb{1}$. This means that $\mathbf{C}$ is a left stochastic matrix, i.e., the magnitude of any of eigenvalues of $\mathbf{C}$ is bounded by one.

III. PDKF Algorithm Derivation

In the DKF given in Algorithm 1, the nodes exchange local data $\{\mathbf{y}_{l,i}, \mathbf{H}_{l,i}, \mathbf{R}_{l,i}\}$, $l \in \mathcal{N}_k$, to calculate the intermediate estimates $\boldsymbol{\psi}_{k,i}$. Clearly, the implementation of such strategy requires considerable communication resources. In the proposed algorithm, the reduction in communication complexity is achieved by the following solutions.

1) Unlike DKF algorithm, the nodes do not exchange the local data $\{\mathbf{y}_{l,i}, \mathbf{H}_{l,i}, \mathbf{R}_{l,i}\}$, $l \in \mathcal{N}_k$ with their neighbors in the incremental step and the algorithm solely relies on the transmission selected entries of $\boldsymbol{\psi}_{l,i}$. Therefore, in the proposed PDKF algorithm, the incremental step (3) changes to the following *Adaptation Phase*:

$$\begin{cases} \mathbf{R}_{e,i} = \mathbf{R}_{k,i} + \mathbf{H}_{k,i}\mathbf{P}_{k,i|i}\mathbf{H}_{k,i}^T \\ \boldsymbol{\psi}_{k,i} = \boldsymbol{\psi}_{k,i} + \mathbf{P}_{k,i|i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}[\mathbf{y}_{k,i} - \mathbf{H}_{k,i}\boldsymbol{\psi}_{k,i}] \\ \mathbf{P}_{k,i} = \mathbf{P}_{k,i|i} - \mathbf{P}_{k,i|i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}\mathbf{H}_{k,i}\mathbf{P}_{k,i|i} \end{cases} \tag{6}$$

.

2) At time instant i, each node k exchanges and diffuses only L out of M, $0 \le L \le M$, entries of the intermediate state estimate vector. According to this scheme, the selection process can be implemented using a diagonal selection matrix, $\mathbf{T}_{k,i} \in \mathbb{R}^{M\times M}$. Multiplication of $\boldsymbol{\psi}_{k,i}$ by $\mathbf{T}_{k,i}$ that has L ones and $M-L$ zeros on its diagonal replaces its nonselected entries with zero. The positions of ones on the diagonal of $\mathbf{T}_{k,i}$ determine the entries of node k that are selected to diffuse at time i. Note that the integer L is fixed and prespecified.

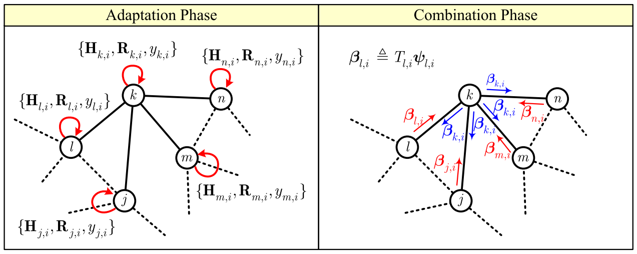
Adaptation Phase
Combination Phase
$\{\mathbf{H}_{k,i}, \mathbf{R}_{k,i}, y_{k,i}\}$
$\{\mathbf{H}_{n,i}, \mathbf{R}_{n,i}, y_{n,i}\}$
$\{\mathbf{H}_{l,i}, \mathbf{R}_{l,i}, y_{l,i}\}$
$\{\mathbf{H}_{m,i}, \mathbf{R}_{m,i}, y_{m,i}\}$
$\{\mathbf{H}_{j,i}, \mathbf{R}_{j,i}, y_{j,i}\}$
k
n
l
m
j
$\boldsymbol{\beta}_{l,i} \triangleq T_{l,i}\boldsymbol{\psi}_{l,i}$
$\boldsymbol{\beta}_{k,i}$
$\boldsymbol{\beta}_{l,i}$
$\boldsymbol{\beta}_{k,i}$
$\boldsymbol{\beta}_{k,i}$
$\boldsymbol{\beta}_{k,i}$
$\boldsymbol{\beta}_{n,i}$
$\boldsymbol{\beta}_{j,i}$
$\boldsymbol{\beta}_{m,i}$

Fig. 2. PDKF update at node k.

Remark 1: At any node k and iteration i, the entry selection matrices, $\mathbf{T}_{k,i}$, are statistically independent of any input–output data. Furthermore, the probability of scattered entries of each node at each iteration is equal and expressed as

$$\rho = \frac{L}{M}. \tag{7}$$

The most fundamental problem we face hinges on the ambiguities in nondiffused elements of the nodes in combination phase. When the intermediate estimates are partially transmitted, the noncommunicated entries are not available to take part in this phase. However, each node requires all entries of the intermediate estimate vectors of its neighbors for combination. To avoid this ambiguity, the nodes can replace the entries of their own intermediate estimates instead of the ones from the neighbors that are not available. The following equation (*combination phase*) is used to aggregate the partially received intermediate estimates:

$$\hat{\mathbf{x}}_{k,i|i} = \boldsymbol{\psi}_{k,i} + \sum_{l \in \mathcal{N}_k \backslash \{k\}} c_{lk} \mathbf{T}_{l,i} (\boldsymbol{\psi}_{l,i} - \boldsymbol{\psi}_{k,i}). \tag{8}$$

Therefore, we substitute the unavailable elements with their equivalent ones in each node's own intermediate estimate vector. Accordingly, our proposed PDKF algorithm employs (6) in the adaptation phase and (8) in the combination phase. The proposed PDKF algorithm is described in Algorithm 2. This process is also demonstrated schematically in Fig. 2.

Remark 2: The computational complexity of the proposed PDKF algorithm is similar to the original DKF algorithm given by Algorithm 1. To show this, note that (8) can be written as

$$\hat{\mathbf{x}}_{k,i|i} = c_{kk}\boldsymbol{\psi}_{k,i} + \sum_{l \in \mathcal{N}_k \backslash \{k\}} c_{lk} [\mathbf{T}_{l,i}\boldsymbol{\psi}_{l,i} + (\mathbf{I}_M - \mathbf{T}_{k,i})\boldsymbol{\psi}_{k,i}]. \tag{9}$$

Comparing (4) and (9) reveals that both expressions require $|\mathcal{N}_k|M$ multiplications and $(|\mathcal{N}_k| - 1)M$ additions per iteration per node. To select L-subset of a set on M elements containing exactly L elements, we employ a similar approach proposed in [15] (see the Appendix for more details).

IV. Performance Analysis

To begin with, let $\tilde{\boldsymbol{\psi}}_{k,i} = \mathbf{x}_i - \boldsymbol{\psi}_{k,i}$ denote the estimation error at the end of the adaptation phase. Then, it can be easily shown that the following expression holds:

$$\begin{aligned} \tilde{\boldsymbol{\psi}}_{k,i} &= \tilde{\mathbf{x}}_{k,i|i-1} - \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}(\mathbf{H}_{k,i}\tilde{\mathbf{x}}_{k,i|i-1} + \mathbf{v}_{k,i}) \\ &= \left(\mathbf{I}_M - \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}\mathbf{H}_{k,i}\right)\tilde{\mathbf{x}}_{k,i|i-1} - \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}\mathbf{v}_{k,i}. \end{aligned} \tag{10}$$

Using the matrix inversion lemma, it is straightforward to show that $\mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{k,i}^{-1} = \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}$. Thus, we can rewrite (10) as

$$\tilde{\boldsymbol{\psi}}_{k,i} = (\mathbf{I}_M - \mathbf{P}_{k,i}\mathbf{S}_{k,i})\tilde{\mathbf{x}}_{k,i|i-1} - \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{k,i}^{-1}\mathbf{v}_{k,i} \tag{11}$$

where $\mathbf{S}_{k,i} \triangleq \mathbf{H}_{k,i}^T\mathbf{R}_{k,i}^{-1}\mathbf{H}_{k,i}$. We also have

$$\tilde{\mathbf{x}}_{k,i|i-1} = \mathbf{F}_{i-1}\tilde{\mathbf{x}}_{k,i-1|i-1} + \mathbf{G}_{i-1}\mathbf{n}_{i-1}. \tag{12}$$

Algorithm 2 PDKF

```
**Initialization:** $\hat{\mathbf{x}}_{k,0|-1} = 0$ and $\mathbf{P}_{k,0|-1} = \boldsymbol{\Pi}_0$
For every time instant $i$, every node $k$ computes
**Step1:** Adaptation phase
$\boldsymbol{\psi}_{k,i} \leftarrow \hat{\mathbf{x}}_{k,i|i-1}$
$\mathbf{P}_{k,i} \leftarrow \mathbf{P}_{k,i|i-1}$
    $\mathbf{R}_{e,i} = \mathbf{R}_{k,i} + \mathbf{H}_{k,i}\mathbf{P}_{k,i}\mathbf{H}_{k,i}^T$
    $\boldsymbol{\psi}_{k,i} = \boldsymbol{\psi}_{k,i} + \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}[\mathbf{y}_{k,i} - \mathbf{H}_{k,i}\boldsymbol{\psi}_{k,i}]$
    $\mathbf{P}_{k,i} = \mathbf{P}_{k,i} - \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}\mathbf{H}_{k,i}\mathbf{P}_{k,i}$
**Step2:** Combination Phase
    $\hat{\mathbf{x}}_{k,i|i} \leftarrow \boldsymbol{\psi}_{k,i} + \sum_{l \in \mathcal{N}_k \backslash \{\mathrm{k}\}} c_{lk}\mathbf{T}_{l,i}(\boldsymbol{\psi}_{l,i} - \boldsymbol{\psi}_{k,i})$
    $\mathbf{P}_{k,i|i} \leftarrow \mathbf{P}_{k,i}$
    $\hat{\mathbf{x}}_{k,i+1|i} = \mathbf{F}_i\hat{\mathbf{x}}_{k,i|i}$
    $\mathbf{P}_{k,i+1|i} = \mathbf{F}_i\mathbf{P}_{k,i|i}\mathbf{F}_i^T + \mathbf{G}_i\mathbf{Q}_i\mathbf{G}_i^T$
```

Substituting (12) into (11) gives

$$\begin{aligned} \tilde{\boldsymbol{\psi}}_{k,i} = (\mathbf{I}_M - \mathbf{P}_{k,i}\mathbf{S}_{k,i})\mathbf{F}_{i-1}\tilde{\mathbf{x}}_{k,i-1|i-1} \\ +(\mathbf{I}_M - \mathbf{P}_{k,i}\mathbf{S}_{k,i})\mathbf{G}_{i-1}\mathbf{n}_{i-1} - \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{k,i}^{-1}\mathbf{v}_{k,i} \end{aligned} \tag{13}$$

To proceed, we define the augmented state-error vectors $\tilde{\boldsymbol{\mathcal{X}}}_{i|i}$ and $\tilde{\boldsymbol{\Psi}}_i$, measurement noise vectors $\mathbf{v}_i$, and block-diagonal matrices $\boldsymbol{\mathcal{H}}_i$, $\boldsymbol{\mathcal{P}}_{i|i}$, $\boldsymbol{\mathcal{S}}_i$, and $\boldsymbol{\mathcal{B}}_i$ as follows:

$$\begin{aligned} \tilde{\boldsymbol{\mathcal{X}}}_{i|i} &= \mathrm{col}\{\tilde{\mathbf{x}}_{1,i|i}, \tilde{\mathbf{x}}_{2,i|i}, \ldots, \tilde{\mathbf{x}}_{N,i|i}\} \\ \tilde{\boldsymbol{\Psi}}_i &= \mathrm{col}\{\tilde{\boldsymbol{\psi}}_{1,i}, \tilde{\boldsymbol{\psi}}_{2,i}, \ldots, \tilde{\boldsymbol{\psi}}_{N,i}\} \\ \mathbf{v}_i &= \mathrm{col}\{\mathbf{v}_{1,i}, \ldots, \mathbf{v}_{N,i}\} \\ \boldsymbol{\mathcal{H}}_i &= \mathrm{diag}\{\mathbf{H}_{1,i}, \mathbf{H}_{2,i}, \ldots, \mathbf{H}_{N,i}\} \\ \boldsymbol{\mathcal{P}}_{i|i} &= \mathrm{diag}\{\mathbf{P}_{1,i|i}, \mathbf{P}_{2,i|i}, \ldots, \mathbf{P}_{N,i|i}\} \\ \boldsymbol{\mathcal{S}}_i &= \mathrm{diag}\{\mathbf{S}_{1,i}, \mathbf{S}_{2,i}, \ldots, \mathbf{S}_{N,i}\} \\ \boldsymbol{\mathcal{B}}_i &= [\mathbf{B}_{p,q,i}] \end{aligned}$$

where

$$\mathbf{B}_{p,q,i} = \begin{cases} \mathbf{I}_M - \sum_{l \in \mathcal{N}_p \backslash \{p\}} c_{lp}\mathbf{T}_{l,i} & \text{if } p = q \\ c_{qp}\mathbf{T}_{q,i} & \text{if } q \in \mathcal{N}_p \backslash \{p\} \\ \mathbf{O}_M & \text{otherwise.} \end{cases}$$

Using the above-mentioned definitions, we now can express (9) and (13) in a global form as

$$\tilde{\boldsymbol{\mathcal{X}}}_{i|i} = \boldsymbol{\mathcal{B}}_i\tilde{\boldsymbol{\Psi}}_i \tag{14}$$

$$\begin{aligned} \tilde{\boldsymbol{\Psi}}_i &= (\mathbf{I}_{MN} - \boldsymbol{\mathcal{P}}_{i|i}\boldsymbol{\mathcal{S}}_i)[(\mathbf{I}_N \otimes \mathbf{F}_{i-1})\tilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1} \\ &\quad +(\mathbf{I}_N \otimes \mathbf{G}_{i-1})(\mathbb{1} \otimes \mathbf{n}_{i-1})] - \boldsymbol{\mathcal{P}}_{i|i}\boldsymbol{\mathcal{H}}_i^T\mathbf{R}_i^{-1}\mathbf{v}_i \end{aligned} \tag{15}$$

where $\mathbf{R}_i = \mathbb{E}[\mathbf{v}_i\mathbf{v}_i^T]$ is a block-diagonal matrix. Note that (14) describes the evolution of entire network. Furthermore, note that (15) can be rewritten in a more compact form as

$$\tilde{\boldsymbol{\Psi}}_i = \boldsymbol{\mathcal{F}}_i\tilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1} + \boldsymbol{\mathcal{G}}_i(\mathbb{1} \otimes \mathbf{n}_{i-1}) - \boldsymbol{\mathcal{D}}_i\mathbf{v}_i \tag{16}$$

where

$$\begin{aligned} \boldsymbol{\mathcal{F}}_i &= (\mathbf{I}_{MN} - \boldsymbol{\mathcal{P}}_{i|i}\boldsymbol{\mathcal{S}}_i)(\mathbf{I}_N \otimes \mathbf{F}_{i-1}) \\ \boldsymbol{\mathcal{G}}_i &= (\mathbf{I}_{MN} - \boldsymbol{\mathcal{P}}_{i|i}\boldsymbol{\mathcal{S}}_i)(\mathbf{I}_N \otimes \mathbf{G}_{i-1}) \\ \boldsymbol{\mathcal{D}}_i &= \boldsymbol{\mathcal{P}}_{i|i}\boldsymbol{\mathcal{H}}_i^T\mathbf{R}_i^{-1}. \end{aligned}$$

Now, substituting (16) into (14), the update equation for the network state error vector becomes

$$\widetilde{\boldsymbol{\mathcal{X}}}_{i|i} = \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{F}}_i \widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1} + \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{G}}_i (\mathbb{1} \otimes \mathbf{n}_{i-1}) - \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{D}}_i \mathbf{v}_i. \quad (17)$$

Proposition 1 (Mean Performance): Under Assumption 1, the PDKF is convergent in the mean and asymptotically unbiased.

Proof: Based on Assumption 1, $\boldsymbol{\mathcal{B}}_i$, $\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}$, and $\widetilde{\boldsymbol{\Psi}}_i$ are jointly independent. Therefore, by taking expectation of both sides of (17), we obtain

$$\mathbb{E}[\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}] = \boldsymbol{\mathcal{M}} \boldsymbol{\mathcal{F}}_i \mathbb{E}[\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}]. \quad (18)$$

As stated in [15, Appendix], all entries of $\boldsymbol{\mathcal{M}} = \mathbb{E}[\boldsymbol{\mathcal{B}}_i]$ are real nonnegative with each row summing to unity, i.e., $\boldsymbol{\mathcal{M}}\mathbb{1} = \mathbb{1}$. Since $\mathbb{E}[\hat{\mathbf{x}}_{k,0|-1}] = 0$ and $\mathbb{E}[\mathbf{x}_0] = 0$, we have $\mathbb{E}[\hat{\mathbf{x}}_{k,-1|-1}] = 0$ for all k. Therefore, the following equation holds:

$$\lim_{i \to \infty} \mathbb{E}[\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}] = \mathbf{0}_{MN}$$

where $\mathbf{0}_{MN}$ denotes an $MN \times 1$ zero vector. This means that the PDKF is convergent in the mean and asymptotically unbiased. □

To investigate the mean-square performance of the PDKF, we first introduce the following assumption.

Assumption 2:

1) The matrices in (1) and (2) are time invariant, i.e., the matrices $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$, $\mathbf{R}$, and $\mathbf{Q}$ do not depend on time i.
2) The pair of matrices $\{\mathbf{F}, \mathbf{H}_k\}$ is detectable for every node k and $\{\mathbf{F}, \mathbf{G}\mathbf{Q}^{\frac{1}{2}}\}$ is controllable on the unit circle. Moreover, $\{\mathbf{F}, \mathbf{G}\mathbf{Q}^{\frac{1}{2}}\}$ is stabilizable [10].

Remark 3: The time-invariant assumption is considered to make the analysis tractable. Such an assumption is not a necessary condition for using our proposed algorithm for linear state estimation in connected network. On the other hand, Assumption 2-2) is a common assumption in the analysis of Kalman filtering algorithms (see more details in [37, pp. 310–332]). Under the second item of Assumption 2, the matrices $\mathbf{P}_{k,i+1|i}$ will converge to $\mathbf{P}_k^-$ for any initial condition $\mathbf{P}_{k,0|-1} > 0$ as $i \to \infty$ where

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_k^-\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T - \mathbf{K}_{p,k}\mathbf{R}_{e,k}\mathbf{K}_{p,k}^T \quad (19)$$

with

$$\begin{aligned} \mathbf{K}_{p,k} &= \mathbf{F}\mathbf{P}_k^-\mathbf{H}_k^T\mathbf{R}_{e,k}^{-1} \\ \mathbf{R}_{e,k} &= \mathbf{R}_k + \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{R}_{e,k}^{-1}\mathbf{H}_k\mathbf{P}_k^-. \end{aligned} \quad (20)$$

Note that $\mathbf{P}_k^-$ is the unique stabilizing solution of discrete-time algebraic Riccati equation (19). Moreover, the matrix $\mathbf{P}_{k,i|i}$ converges to matrix $\mathbf{P}_k$ in (20).

Under Assumption 2, matrices $\boldsymbol{\mathcal{F}}_i$, $\boldsymbol{\mathcal{G}}_i$, and $\boldsymbol{\mathcal{D}}_i$ also converge in steady state, and their steady-state values are given by

$$\begin{aligned} \boldsymbol{\mathcal{P}} &\triangleq \lim_{i\to\infty} \boldsymbol{\mathcal{P}}_{i|i} = \mathrm{diag}\{\mathbf{P}_1, \ldots, \mathbf{P}_N\} \\ \boldsymbol{\mathcal{P}}^- &\triangleq \lim_{i\to\infty} \boldsymbol{\mathcal{P}}_{i|i-1} = \mathrm{diag}\{\mathbf{P}_1^-, \ldots, \mathbf{P}_N^-\} \\ \boldsymbol{\mathcal{F}} &\triangleq \lim_{i\to\infty} \boldsymbol{\mathcal{F}}_i = (\mathbf{I}_{MN} - \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{S}})(\mathbf{I}_N \otimes \mathbf{F}) \\ \boldsymbol{\mathcal{G}} &\triangleq \lim_{i\to\infty} \boldsymbol{\mathcal{G}}_i = (\mathbf{I}_{MN} - \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{S}})(\mathbf{I}_N \otimes \mathbf{G}) \\ \boldsymbol{\mathcal{D}} &\triangleq \lim_{i\to\infty} \boldsymbol{\mathcal{D}}_i = \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{H}}^T\mathbf{R}^{-1} \end{aligned}$$

where $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{H}}$ are used instead of $\boldsymbol{\mathcal{S}}_i$ and $\boldsymbol{\mathcal{H}}_i$ since these matrices are now time-invariant.

Proposition 2 (Mean-Square Stability): The partial DKF (PDKF) algorithm given by Algorithm 2 is convergent and stable in mean-square sense.

Proof: Taking the squared weighted Euclidean norm of both sides of (17) and applying the expectation operator with *Assumption* 2 yield the following weighted variance relation:

$$\begin{aligned} \mathbb{E}\big[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}\|_{\boldsymbol{\Sigma}}^2\big] &= \mathbb{E}\big[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}\|_{\boldsymbol{\Gamma}}^2\big] \\ &\quad + \mathbb{E}\big[(\mathbb{1} \otimes \mathbf{n}_{i-1})^T \boldsymbol{\mathcal{G}}_i^T \boldsymbol{\mathcal{B}}_i^T \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{G}}_i (\mathbb{1} \otimes \mathbf{n}_{i-1})\big] \\ &\quad + \mathbb{E}\big[\mathbf{v}_i^T \boldsymbol{\mathcal{D}}_i^T \boldsymbol{\mathcal{B}}_i^T \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{D}}_i \mathbf{v}_i\big] \quad (21) \\ \boldsymbol{\Gamma} &= \boldsymbol{\mathcal{B}}_i^T \boldsymbol{\mathcal{F}}_i^T \boldsymbol{\Sigma} \boldsymbol{\mathcal{F}}_i \boldsymbol{\mathcal{B}}_i \quad (22) \end{aligned}$$

where $\boldsymbol{\Sigma}$ is an arbitrary symmetric nonnegative definite matrix. Since $\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}$ is independent of $\boldsymbol{\Gamma}$, we have

$$\mathbb{E}\big[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}\|_{\boldsymbol{\Gamma}}^2\big] = \mathbb{E}\big[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}\|_{\mathbb{E}[\boldsymbol{\Gamma}]}^2\big]. \quad (23)$$

Define

$$\boldsymbol{\gamma} \triangleq \mathrm{vec}\{\mathbb{E}[\boldsymbol{\Gamma}]\}, \quad \text{and } \boldsymbol{\sigma} \triangleq \mathrm{vec}\{\boldsymbol{\Sigma}\} \quad (24)$$

where vec$\{\cdot\}$ denotes a linear transformation which converts the matrix into a column vector by stacking all columns of its matrix argument. The transpose of a vectorized matrix is also denoted by vec$^T\{\cdot\}$. Using (23) and (24), we can alter (21) to

$$\begin{aligned} \mathbb{E}\big[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}\|_{\boldsymbol{\sigma}}^2\big] &= \mathbb{E}\big[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}\|_{\boldsymbol{\gamma}}^2\big] \\ &\quad + \mathbb{E}\big[(\mathbb{1} \otimes \mathbf{n}_{i-1})^T \boldsymbol{\mathcal{G}}_i^T \boldsymbol{\mathcal{B}}_i^T \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{G}}_i (\mathbb{1} \otimes \mathbf{n}_{i-1})\big] \\ &\quad + \mathbb{E}\big[\mathbf{v}_i^T \boldsymbol{\mathcal{D}}_i^T \boldsymbol{\mathcal{B}}_i^T \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{D}}_i \mathbf{v}_i\big] \quad (25) \end{aligned}$$

where $\mathbb{E}[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}\|_{\boldsymbol{\sigma}}^2]$ and $\mathbb{E}[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}\|_{\boldsymbol{\gamma}}^2]$ are the same quantities as $\mathbb{E}[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}\|_{\boldsymbol{\Sigma}}^2]$ and $\mathbb{E}[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}\|_{\mathbb{E}[\boldsymbol{\Gamma}]}^2]$, respectively. The relationship between the Kronecker product and vec$\{\cdot\}$ operator [38], i.e., vec$\{\mathbf{XYZ}\} = (\mathbf{Z}^T \otimes \mathbf{X})vec\{\mathbf{Y}\}$ and also the commutative property of the expectation and vectorization operations enable us to write (22) in terms of $\boldsymbol{\gamma}$ and $\boldsymbol{\sigma}$ as

$$\boldsymbol{\gamma} = \mathfrak{B}(\boldsymbol{\mathcal{F}}_i^T \otimes \boldsymbol{\mathcal{F}}_i^T)\boldsymbol{\sigma} \quad (26)$$

where $\mathfrak{B} = \mathbb{E}[\boldsymbol{\mathcal{B}}_i^T \otimes \boldsymbol{\mathcal{B}}_i^T]$. In addition, by applying the following property from linear algebra [16]:

$$\mathrm{tr}\{\mathbf{X}^T\mathbf{Y}\} = \mathrm{vec}^T\{\mathbf{Y}\}\mathrm{vec}\{\mathbf{X}\}$$

and the symmetry of $\boldsymbol{\Sigma}$, we have

$$\begin{aligned} &\mathbb{E}\big[(\mathbb{1} \otimes \mathbf{n}_{i-1})^T \boldsymbol{\mathcal{G}}_i^T \boldsymbol{\mathcal{B}}_i^T \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{G}}_i (\mathbb{1} \otimes \mathbf{n}_{i-1})\big] \\ &\quad = \mathbb{E}\big[\mathrm{tr}\big\{\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}_i\boldsymbol{\mathcal{G}}_i\big(\mathbb{1} \otimes \mathbf{n}_{i-1}\big)\big(\mathbb{1} \otimes \mathbf{n}_{i-1}\big)^T \boldsymbol{\mathcal{G}}_i^T \boldsymbol{\mathcal{B}}_i^T\big\}\big] \\ &\quad = \mathrm{vec}^T\big\{\mathbb{E}\big[\big\{\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}_i\boldsymbol{\mathcal{G}}_i\big(\mathbb{1} \otimes \mathbf{n}_{i-1}\big)\big(\mathbb{1} \otimes \mathbf{n}_{i-1}\big)^T \boldsymbol{\mathcal{G}}_i^T \boldsymbol{\mathcal{B}}_i^T\big\}\big]\big\} \\ &\quad = \big(\mathfrak{B}^T \mathrm{vec}\big\{\mathbb{E}\big[\boldsymbol{\mathcal{G}}_i\big(\mathbb{1} \otimes \mathbf{n}_{i-1}\big)\big(\mathbb{1} \otimes \mathbf{n}_{i-1}\big)^T \boldsymbol{\mathcal{G}}_i^T\big]\big\}\big)^T \boldsymbol{\sigma} \\ &\quad = \mathrm{vec}^T\big\{\boldsymbol{\mathcal{L}}\big\}\mathfrak{B}\boldsymbol{\sigma} \quad (27) \end{aligned}$$

where $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{G}}_i(\mathbb{1}\mathbb{1}^T \otimes \boldsymbol{\mathcal{Q}}_{i-1})\boldsymbol{\mathcal{G}}_i^T$. The last term of (25) can be written as

$$\begin{aligned} \mathbb{E}\big[\mathbf{v}_i^T \boldsymbol{\mathcal{D}}_i^T \boldsymbol{\mathcal{B}}_i^T \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}_i \boldsymbol{\mathcal{D}}_i \mathbf{v}_i\big] &= \mathbb{E}\big[\mathrm{tr}\big\{\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}_i\boldsymbol{\mathcal{D}}_i\mathbf{v}_i\mathbf{v}_i^T\boldsymbol{\mathcal{D}}_i^T\boldsymbol{\mathcal{B}}_i^T\big]\big\} \\ &= \mathbb{E}\big[\mathrm{vec}^T\big\{\boldsymbol{\mathcal{B}}_i\boldsymbol{\mathcal{D}}_i\mathbf{v}_i\mathbf{v}_i^T\boldsymbol{\mathcal{D}}_i^T\boldsymbol{\mathcal{B}}_i^T\big\}\boldsymbol{\sigma}\big] \\ &= \big(\mathfrak{B}^T \mathrm{vec}\big\{\mathbb{E}\big[\boldsymbol{\mathcal{D}}_i\mathbf{v}_i\mathbf{v}_i^T\boldsymbol{\mathcal{D}}_i^T\big]\big\}\big)^T \boldsymbol{\sigma} \\ &= \mathrm{vec}^T\big\{\boldsymbol{\mathcal{K}}\big\}\mathfrak{B}\boldsymbol{\sigma} \quad (28) \end{aligned}$$

where $\boldsymbol{\mathcal{K}} = \boldsymbol{\mathcal{D}}_i\mathbf{R}_i\boldsymbol{\mathcal{D}}_i^T$. Substituting (26)–(28) into (25) gives

$$\mathbb{E}\big[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}\|_{\boldsymbol{\sigma}}^2\big] = \mathbb{E}\big[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}\|_{\mathfrak{B}\left(\boldsymbol{\mathcal{F}}_i^T \otimes \boldsymbol{\mathcal{F}}_i^T\right)\boldsymbol{\sigma}}^2\big] + \mathrm{vec}^T\big\{\boldsymbol{\Omega}\big\}\mathfrak{B}\boldsymbol{\sigma} \quad (29)$$

where $\boldsymbol{\Omega} = \boldsymbol{\mathcal{K}} + \boldsymbol{\mathcal{L}}$. The recursion (29) is convergent and stable if, and only if, the matrix $\mathfrak{B}(\boldsymbol{\mathcal{F}}^T \otimes \boldsymbol{\mathcal{F}}^T)$ is stable, or equivalently, all its eigenvalues lie inside the unite circle. The entries of $\mathfrak{B}$ are real and nonnegative. Moreover, we have

$$\begin{aligned}\mathfrak{B}^T\mathbb{1}_{M^2N^2} &= \mathbb{E}[\mathcal{B}_i \otimes \mathcal{B}_i]\mathbb{1}_{M^2N^2}\\ &= \mathbb{E}[\mathcal{B}_i\mathbb{1}_{MN} \otimes \mathcal{B}_i\mathbb{1}_{MN}] = \mathbb{1}_{M^2N^2}\end{aligned}$$

which means that each column of $\mathfrak{B}$ sums up to unity. We say that $\mathfrak{B}$ is a left stochastic matrix and has unit spectral radius. Therefore, the PDKF algorithm is stable in mean-square sense if, and only if, $(\boldsymbol{\mathcal{F}}^T \otimes \boldsymbol{\mathcal{F}}^T)$ is stable. The matrix $(\boldsymbol{\mathcal{F}}^T \otimes \boldsymbol{\mathcal{F}}^T)$ is stable if, and only if, $\boldsymbol{\mathcal{F}}$ is stable. The items in *Assumption* 2 are sufficient to guarantee the stability of matrix $\boldsymbol{\mathcal{F}}$. □

In the sequel, we turn our attention into obtaining the steady-state network MSD for the PDKF algorithm. To this end, we first define the network MSD as a steady-state performance metric as

$$\eta_L = \lim_{i\to\infty}\frac{1}{N}\sum_{k=1}^{N}\mathbb{E}[\|\mathbf{x}_i - \hat{\mathbf{x}}_{k,i|i}\|^2]. \tag{30}$$

The following proposition summarizes the steady-state performance of the PDKF algorithm.

Proposition 3 (Steady-State Performance): Let Assumptions 1 and 2 hold. Then, the network steady-state MSD for the PDKF algorithm is given by

$$\eta_L = \frac{1}{N}\text{vec}^T\{\boldsymbol{\Omega}\}\mathfrak{B} \times (\mathbf{I}_{M^2N^2} - \mathfrak{B}(\boldsymbol{\mathcal{F}}^T \otimes \boldsymbol{\mathcal{F}}^T))^{-1}\mathbf{h} \tag{31}$$

where $\mathbf{h} \triangleq \text{vec}\{\mathbf{I}_{MN}\}$.

Proof: By taking the limit of (29) as $i \to \infty$, we have

$$\lim_{i\to\infty}\mathbb{E}\big[\|\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}\|^2_{(\mathbf{I}_{M^2N^2}-\mathfrak{B}(\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T))\boldsymbol{\sigma}}\big] = \text{vec}^T\{\boldsymbol{\Omega}\}\mathfrak{B}\boldsymbol{\sigma}. \tag{32}$$

Note that since $\boldsymbol{\mathcal{F}}$ is stable[1] then matrix $\mathbf{I}_M - \mathfrak{B}(\boldsymbol{\mathcal{F}}_i^T \otimes \boldsymbol{\mathcal{F}}_i^T)$ is nonsingular. We can recover η_L from (32) by setting $\boldsymbol{\sigma} = \text{vec}\{\mathbf{I}_{MN}\}$, which gives (31). □

V. Discussion on the Communication Trade-off

To show the performance-communication trade-off provided by the PDKF algorithm performance precisely, we introduce the following assumptions to examine its steady-state network MSD, both for the sequential schemes.

Assumption 3:

1) Matrix $\mathbf{C}$ is doubly stochastic.
2) During any M consecutive iterations, the node's intermediate estimate vector does not vary considerably.

Note that Assumption 3-1) is considered only to conduct the analysis and obtain a better understanding of the performance of the PDKF algorithm. Such an assumption is not required to implement the proposed algorithm. As discussed in [15], when Assumption 3-2) holds, in every M consecutive time iterations, all the agents concurrently diffuse all the entries of their intermediate estimate vectors to their neighbors at only L iterations. This assumption translates the approximation of the sequential partial-diffusion scheme into a periodic diffusion scheme. In lights of [15, Appendix], with this scheme, we have

$$\mathfrak{B} = (1-\rho)\mathbf{I}_{M^2N^2} + \rho\mathfrak{C}\otimes\mathfrak{C} \tag{33}$$

where $\mathfrak{C} = \mathbf{C} \otimes \mathbf{I}_M$. The following proposition indicates the communication–performance trade-off for the PDKF algorithm.

Proposition 4: Under Assumptions 1 and 3, the following inequalities hold for the steady-state network MSD of the PDKF algorithm:

$$\eta_M < \eta_{M-1} < \cdots < \eta_L < \cdots < \eta_1 < \eta_0. \tag{34}$$

Proof: First, it should be noted that η_L in (31) can be rewritten as

$$\eta_L = \frac{1}{N}\text{vec}^T\{\boldsymbol{\Omega}\}\mathfrak{B}\sum_{i=0}^{\infty}[\mathfrak{B}(\boldsymbol{\mathcal{F}}^T \otimes \boldsymbol{\mathcal{F}}^T)]^i\mathbf{h} \tag{35}$$

Substituting (33) into (35) gives

$$\begin{aligned}\eta_L &= \frac{1}{N}\sum_{i=0}^{\infty}(1-\rho)^{i+1}\text{vec}^T\{\boldsymbol{\Omega}\}[(\boldsymbol{\mathcal{F}}^T)^i \otimes (\boldsymbol{\mathcal{F}}^T)^i]\mathbf{h}\\ &\quad+\frac{1}{N}\sum_{i=0}^{\infty}\rho^{i+1}\text{vec}^T\{\boldsymbol{\Omega}\}[\mathfrak{C}(\boldsymbol{\mathcal{F}}^T\mathfrak{C})^i \otimes \mathfrak{C}(\boldsymbol{\mathcal{F}}^T\mathfrak{C})^i]\mathbf{h}\\ &= \frac{1}{N}\sum_{i=0}^{\infty}(1-\rho)^{i+1}\text{vec}^T\{\boldsymbol{\mathcal{F}}^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\}\mathbf{h}\\ &\quad+\frac{1}{N}\sum_{i=0}^{\infty}\rho^{i+1}\text{vec}^T\{(\mathfrak{C}^T\boldsymbol{\mathcal{F}})^i\mathfrak{C}^T\boldsymbol{\Omega}\mathfrak{C}(\boldsymbol{\mathcal{F}}^T\mathfrak{C})^i\}\mathbf{h}\\ &= \frac{1}{N}\sum_{i=0}^{\infty}(1-\rho)^{i+1}\text{tr}\{\boldsymbol{\mathcal{F}}^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\}\\ &\quad+\frac{1}{N}\sum_{i=0}^{\infty}\rho^{i+1}\text{tr}\{(\mathfrak{C}^T\boldsymbol{\mathcal{F}})^i\mathfrak{C}^T\boldsymbol{\Omega}\mathfrak{C}(\boldsymbol{\mathcal{F}}^T\mathfrak{C})^i\}.\end{aligned} \tag{36}$$

With the first item in Assumption 1, we have

$$\text{tr}\{(\mathfrak{C}^T\boldsymbol{\mathcal{F}})^i\mathfrak{C}^T\boldsymbol{\Omega}\mathfrak{C}(\boldsymbol{\mathcal{F}}^T\mathfrak{C})^i\} = \text{tr}\{(\mathbf{C}^T)^{i+1}\mathbf{C}^{i+1}\} \times \text{tr}\{\boldsymbol{\mathcal{F}}^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\}$$

where

$$\text{tr}\{(\mathbf{C}^T)^{i+1}\mathbf{C}^{i+1}\} = \sum_{k=1}^{N}\|c_{k,i+1}\|, \quad i \geq 0$$

where $c_{k,i+1}$ denotes the kth row of $\mathbf{C}^{i+1}$. Since $\mathbf{C}$ is doubly stochastic, $\mathbf{C}^{i+1}$, $i \geq 0$, is also doubly stochastic. Moreover, in a connected network, each node is coupled with its neighbor (at least one other node). This fact, for any k, leads to $\|c_{k,i+1}\|^2 \leq 1, i \ \geq 0$, and therefore,

$$\text{tr}\{(\mathfrak{C}^T\boldsymbol{\mathcal{F}})^i\mathfrak{C}^T\boldsymbol{\Omega}\mathfrak{C}(\boldsymbol{\mathcal{F}}^T\mathfrak{C})^i\} < \text{tr}\{\boldsymbol{\mathcal{F}}^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\}. \tag{37}$$

Equation (37) implies that the following inequality holds for $M = \{1, \ldots, L-1\}$, $(0 < \rho < 1)$:

$$\begin{aligned}&\text{tr}\{(\mathfrak{C}^T\boldsymbol{\mathcal{F}})^i\mathfrak{C}^T\boldsymbol{\Omega}\mathfrak{C}(\boldsymbol{\mathcal{F}}^T\mathfrak{C})^i\}\\ &\quad< (1-\rho)^{i+1}\text{tr}\{\boldsymbol{\mathcal{F}}^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\}\\ &\qquad+\rho^{i+1}\text{tr}\{(\mathfrak{C}^T\boldsymbol{\mathcal{F}})^i\mathfrak{C}^T\boldsymbol{\Omega}\mathfrak{C}(\boldsymbol{\mathcal{F}}^T\mathfrak{C})^i\} < \text{tr}\{\boldsymbol{\mathcal{F}}^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\}.\end{aligned} \tag{38}$$

In addition, for the cases of noncooperative algorithm ($L = 0, \rho = 0$) and full-diffusion ($L = M, \rho = 1$) algorithm, we, respectively, have

$$\eta_0 = \frac{1}{N}\sum_{i=0}^{\infty}\text{tr}\{\boldsymbol{\mathcal{F}}^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\} \tag{39}$$

$$\eta_M = \frac{1}{N}\sum_{i=0}^{\infty}\text{tr}\{(\mathfrak{C}^T\boldsymbol{\mathcal{F}})^i\mathfrak{C}^T\boldsymbol{\Omega}\mathfrak{C}(\boldsymbol{\mathcal{F}}^T\mathfrak{C})^i\}. \tag{40}$$

From (35)–(38), we arrive at (34), which completes the proof. □

This implies that the better steady-state network MSD performance is achieved, the more entries are scattered at each iteration. Indeed, a communication–performance trade-off is realized by partial-diffusion algorithm. However, it should be noted that the performance offered by the PDKF algorithm depends not only on ρ but also on the other parameters such as the combination weights as well as the quality of measurements.

[1] See [10] for a detailed proof.

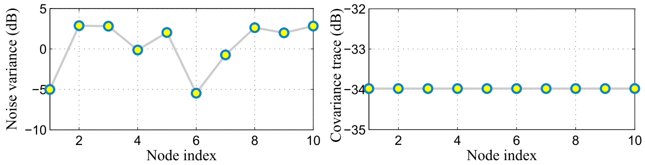

Fig. 3. Variance of the observation noise and covariance matrix trace of the state noise at each node.

VI. Simulation Results

We now present the simulation results to evaluate the performance of PDKF algorithm and draw a comparison between its performance and the theoretical results of Section IV. To this end, we consider a randomly generated network with $N = 10$ nodes, where each node is, on average, linked with two other nodes. The size of the unknown parameter of the system is $M = 4$. The state of the system is unknown 2-D vector location of an object, i.e., (x, y), where x and y are the first and second entries, respectively. The state-space model matrices in (1) and (2) are

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{G} = 0.625\mathbf{I}_4, \;\; \mathbf{Q} = 0.001\mathbf{I}_4.$$

We assume that every node measures the position of the unknown object in either the two horizontal dimensions or a combination of one horizontal and one vertical dimensions. Thus, the individual nodes do not have direct (three dimensional) measurements of the position. Therefore, we have

$$\mathbf{H}_{k,i} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{H}_{k,i} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

at random, but with the requirement that every neighborhood should have nodes with both types of matrices to guarantee that the convergence of the local Kalman filter follows Assumption 1. Finally, the measurement noise covariance matrix for agent k is, $\mathbf{R}_{k,i} = \sigma_{k,i}^2\mathbf{I}_3$, where the noise variance $\sigma_{k,i}^2$ across agents is selected randomly in the range [0 0.5]. The experimental results are achieved by ensemble averaging over 200 independent runs. The steady-state values are also assessed by taking an average over 1000 iterations. We also use the uniform combination rule [10] in the combination phase and initialize the estimates to zero vectors. The state-noise covariance matrix traces and observation noise variances are illustrated in Fig. 3.

In Fig. 4, we plot the average network MSD curves of the PDKF algorithm, using both sequential and stochastic partial-diffusion schemes for different numbers of entries transmitted at each time update, L. We can observe that there is a good match between the derived theoretical expression and simulation results. The validity of our theoretical analysis is also shown in Fig. 5 where the experimental and theoretical steady-state MSDs of all the nodes for different numbers of entries transmitted at each iteration, L, are illustrated. Fig. 5 also shows a trade-off between the communication cost and the estimation performance in Section V. Fig 6 shows the comparison of the steady-state network MSD performance of different algorithms including noncooperative KF algorithm, the proposed PDKF algorithm with different values of L, and the original DKF algorithm. From this figure, we can again see that the PDKF algorithm provides a trade-off between the estimation performance and the communication cost. Note that the difference between full-diffusion PDKF (i.e., $L = M, \rho = 1$) and the DKF algorithm stems from the fact that in the PDKF algorithm the nodes do not exchange local data $\{\mathbf{y}_{l,i}, \mathbf{H}_{l,i}, \mathbf{R}_{l,i}\}$, $l \in \mathcal{N}_k$ with their neighbors.

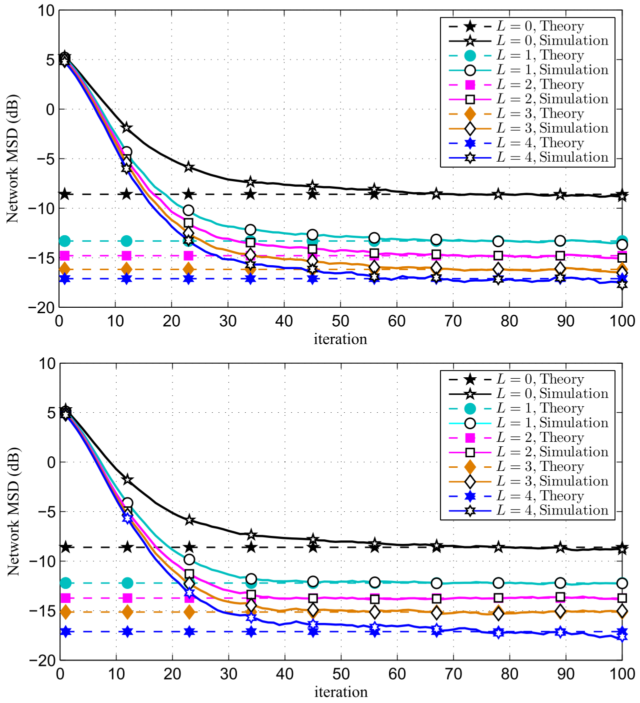

Fig. 4. Network MSD curves of PDKF algorithm for different numbers of entries communicated at each iteration: stochastic scheme (top) and sequential scheme (bottom).

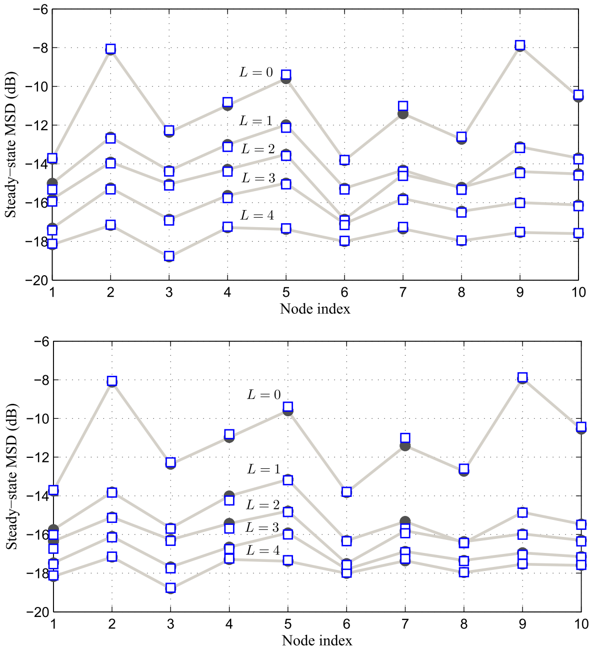

Fig. 5. Theoretical and experimental steady-state MSDs at each node for different values of L using the PDKF algorithm with stochastic scheme (top) and sequential scheme (bottom). Note that the points denoted by symbol “□″ represent the results obtained from theory.

To show the reduction of communication burden rate offered by the PDKF algorithm, we define the communication saving metrics S_{PDKF} and S_{DKF}, which are defined as the rate of communication saved by the PDKF algorithm at each iteration with respect to full-diffusion PDKF $(L = M, \rho = 1)$ and the DKF algorithm,

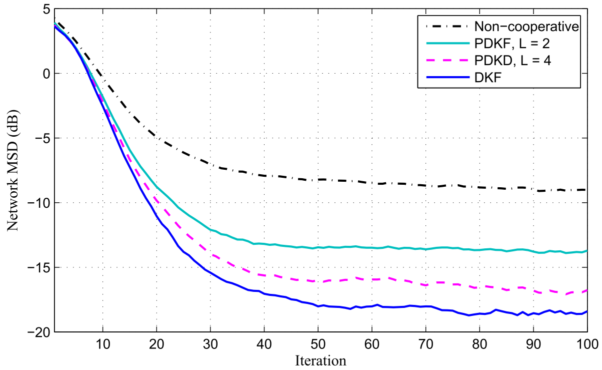

Fig. 6. Steady-state network MSD for different algorithms.

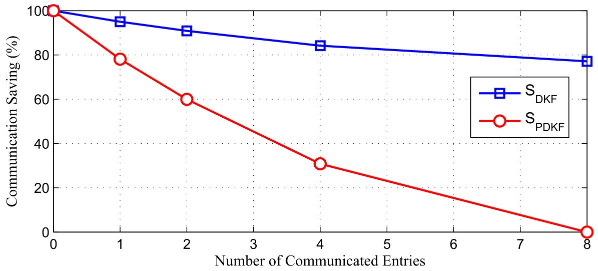

Fig. 7. Communication saving of the PDKF algorithm versus the number of communicated entries at each iteration with respect to full-diffusion PDKF and DKF algorithms.

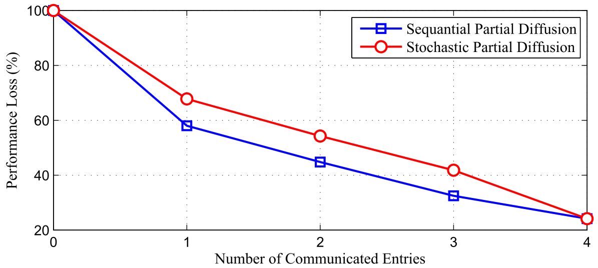

Fig. 8. Performance loss versus the number of communicated entries at each iteration for both stochastic and sequential PDKF schemes.

respectively,

$$S_{\mathrm{PDKF}} = 1 - \frac{Lb + \log_2 \binom{M}{L}}{Mb}$$

$$S_{\mathrm{DKF}} = 1 - \frac{Lb + \log_2 \binom{M}{L}}{b(M + MP + P)}$$

where b denotes the number of bits that describe each intermediate estimate vector elements and $\binom{M}{L}$ returns combinations of L out of M elements. Fig. 7 shows the communication saving metrics S_{PDKF} and S_{DKF} where it is clear that the algorithm needs less communication overhead compared to the conventional DKF method. Finally, the expected degradation on the performance of PDKF algorithm in comparison to DKF is expressed in terms of the performance loss metric, which is defined as

$$\text{Performance loss} = \frac{\eta_L - \eta'_M}{\eta_0 - \eta'_M}$$

where η'_M, η_L, and η_0 denote the steady-state MSD provided by DKF, PDKF, and noncooperative KF algorithms in decibels scale, respectively. Fig. 8 shows the performance losses as the function of number of entries communicated at each iteration for both stochastic and sequential schemes. We can see that the PDKF algorithm with the sequential scheme outperforms the algorithm with stochastic selection pattern.

VII. Conclusion

In this brief, we examined the PDKF algorithm for distributed state estimation. The PDKF enables reduced internode communications by allowing each node to transmit only a subset of intermediate state estimates to its close neighbors at every iteration. This fact directly leads to savings in bandwidth usage as well as power consumption of communication resources. To select the entries, communicating at each iteration, the nodes employ two different protocols, namely, stochastic and sequential partial-diffusion schemes. Consequently, using the PDKF algorithm, the required communications between the nodes are decreased in contrast to the case where all entries of the intermediate estimates are continuously transmitted. We analyzed the convergence of the algorithms and provided steady-state mean and mean-square analysis showing a good agreement with the simulation results. Theoretical analysis and numerical simulations provided valuable insights into the performance of PDKF algorithm and illustrated that it offers a trade-off between the communication cost and the estimation performance.

Appendix
Entry Selection Method

There exist two different schemes to select an L-subset of a set on M elements containing exactly L elements, namely, sequential and stochastic partial diffusion. These methods are analogous to the selection processes in sequential and stochastic partial-update schemes [39]. In the sequential partial diffusion, the entry selection matrices $\mathbf{T}_{k,i}$ are

$$\mathbf{T}_{k,i} = \begin{bmatrix} t_{1,i} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & t_{M,i} \end{bmatrix}, \quad t_{l,i} = \begin{cases} 1 & \text{if } l \in \mathcal{J}_{(i \bmod \bar{\theta})+1} \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

with $\bar{\theta} = \lceil M/L \rceil$. The number of selection entries at each iteration is restricted by L. The coefficient subsets $\mathcal{J}_i$ are not unique as long as they meet the following requirements [38].

1) Cardinality of $\mathcal{J}_i$ is between 1 and L.
2) $\bigcup_{\tau=1}^{\bar{\theta}} \mathcal{J}_\tau = \mathcal{S}$ where $\mathcal{S} = \{1, 2, \ldots, M\}$.
3) $\mathcal{J}_\tau \cap \mathcal{J}_\upsilon = \varnothing, \forall \tau, \upsilon \in \{1, \ldots \bar{\theta}\}$ and $\tau \neq \upsilon$.

As an example, if $\mathcal{S} = \{1, 2, 3, 4\}$, for $M = 4$ and $L = 2$, the subsets of $\mathcal{S}$ containing two elements are, $\mathcal{I}_1 = \{1, 2\}$, $\mathcal{I}_2 = \{1, 3\}$, $\mathcal{I}_3 = \{1, 4\}$, $\mathcal{I}_4 = \{2, 3\}$, $\mathcal{I}_5 = \{2, 4\}$, and $\mathcal{I}_6 = \{3, 4\}$. one possible way to assign $\mathcal{J}_1$ and $\mathcal{J}_2$ is $\mathcal{J}_1 = \mathcal{I}_1$ and $\mathcal{J}_2 = \mathcal{I}_2$.

The description of the entry selection matrices, $\mathbf{T}_{k,i}$, in stochastic partial diffusion is similar to that of sequential one. The only difference is as follows. At a given iteration i, for the sequential case, one of the sets $\mathcal{J}_\tau$, $\tau = 1, \ldots, \bar{\theta}$, is determined in advance, whereas for the stochastic case, one of the sets $\mathcal{J}_\tau$ is sampled at random from $\{\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_{\bar{\theta}}\}$. We do this because the nodes need to know which entries of their neighbors' intermediate estimates have been transmitted at each iteration. These schemes bypass the need for any addressing procedure.

References

[1] Y. Liu, Z. Wang, Y. Yuan, and F. E. Alsaadi, "Partial-nodes-based state estimation for complex networks with unbounded distributed delays," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3906–3912, Aug. 2018.

[2] J. L. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Trans. Autom. Control*, vol. AC-24, no. 2, pp. 266–269, Apr. 1979.

[3] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Approximate distributed Kalman filtering in sensor networks with quantifiable performance," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2005, pp. 133–139.

[4] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conf. Decis. Control (CDC-ECC)*, Dec. 2005, pp. 8179–8184.

[5] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. IEEE 46th Conf. Decis. Control*, Dec. 2007, pp. 5492–5498.

[6] S. M. S. Alam, B. Natarajan, and A. Pahwa, "Distributed agent-based dynamic state estimation over a lossy network," in *Proc. 5th Int. Workshop Netw. Cooper. Objects Smart Cities (UBICITEC)*, 2014, pp. 1–15.

[7] S. M. S. Alam, B. Natarajan, and A. Pahwa, "Agent based optimally weighted Kalman consensus filter over a lossy network," in *Proc. Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[8] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process. (ICASSP)*, vol. 4, May 2001, pp. 2033–2036.

[9] P. Alriksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging," in *Proc. 17th Int. Symp. Math. Theory Netw. Syst.*, 2006, pp. 2445–2450.

[10] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[11] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[12] M. O. Sayin and S. S. Kozat, "Single bit and reduced dimension diffusion strategies over distributed networks," *IEEE Signal Process. Lett.*, vol. 20, no. 10, pp. 976–979, Oct. 2013.

[13] M. O. Sayin and S. S. Kozat, "Compressive diffusion strategies over distributed networks for reduced communication load," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5308–5323, Oct. 2014.

[14] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 257–273, Apr. 2013.

[15] R. Arablouei, S. Werner, Y.-F. Huang, and K. Doğançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 472–484, Jan. 2014.

[16] R. Arablouei, K. Doğançay, S. Werner, and Y.-F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3510–3522, Jul. 2014.

[17] V. Vahidpour, A. Rastegarnia, A. Khalili, and S. Sanei, "Analysis of partial diffusion recursive least squares adaptation over noisy links," *IET Signal Process.*, vol. 11, no. 6, pp. 749–757, 2017.

[18] V. Vahidpour, A. Rastegarnia, A. Khalili, W. M. Bazzi, and S. Sanei, "Analysis of partial diffusion LMS for adaptive estimation over networks with noisy links," *IEEE Trans. Netw. Sci. Eng.*, vol. 5, no. 2, pp. 101–112, Apr./Jun. 2018.

[19] J. R. Deller, Jr., and Y. F. Huang, "Set-membership identification and filtering for signal processing applications," *Circuits, Syst. Signal Process.*, vol. 21, no. 1, pp. 69–82, 2002.

[20] S. Gollamudi, S. Nagaraj, S. Kapoor, and Y.-F. Huang, "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size," *IEEE Signal Process. Lett.*, vol. 5, no. 5, pp. 111–114, May 1998.

[21] J. R. Deller, Jr., M. Nayeri, and S. F. Odeh, "Least-square identification with error bounds for real-time signal processing and control," *Proc. IEEE*, vol. 81, no. 6, pp. 815–849, Jun. 1993.

[22] S. Werner, T. Riihonen, and Y.-F. Huang, "Energy-efficient distributed parameter estimation with partial updates," in *Proc. Conf. Green Circuits Syst. (ICGCS)*, Jun. 2010, pp. 36–40.

[23] A. Malipatil, Y.-F. Huang, and S. Werner, "An SMF approach to distributed average consensus in clustered sensor networks," in *Proc. IEEE 10th Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jun. 2009, pp. 81–85.

[24] S. Werner, Y.-F. Huang, M. L. R. De Campos, and V. Koivunen, "Distributed parameter estimation with selective cooperation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 2849–2852.

[25] S. Werner, M. Mohammed, Y.-F. Huang, and V. Koivunen, "Decentralized set-membership adaptive estimation for clustered sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Mar./Apr. 2008, pp. 3573–3576.

[26] S. Werner and Y.-F. Huang, "Time- and coefficient-selective diffusion strategies for distributed parameter estimation," in *Proc. 44th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2010, pp. 696–697.

[27] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4919–4935, Oct. 2008.

[28] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, "SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4782–4795, Dec. 2006.

[29] T. M. Berg and H. F. Durrant-Whyte, "Model distribution in decentralized multi-sensor data fusion," in *Proc. Amer. Control Conf.*, Jun. 1991, pp. 2292–2293.

[30] M. Raitoharju, R. Piché, J. Ala-Luhtala, and S. Ali-Löytty. (2015). "Partitioned update Kalman filter." [Online]. Available: https://arxiv.org/abs/1503.02857

[31] E. J. Msechu, S. I. Roumeliotis, A. Ribeiro, and G. B. Giannakis, "Decentralized quantized Kalman filtering with scalable communication cost," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3727–3741, Aug. 2008.

[32] S. S. Pereira and A. Pages-Zamora, "Distributed consensus in wireless sensor networks with quantized information exchange," in *Proc. IEEE 9th Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2008, pp. 241–245.

[33] X. He, W. Xue, and H. Fang, "Consistent distributed state estimation with global observability over sensor network," *Automatica*, vol. 92, pp. 162–172, Jun. 2018.

[34] X. He, C. Hu, W. Xue, and H. Fang, "On event-based distributed Kalman filter with information matrix triggers," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14308–14313, 2017.

[35] X. He, C. Hu, Y. Hong, L. Shi, and H. Fang. (2017). "Distributed Kalman filters with state equality constraints: Time-based and event-triggered communications." [Online]. Available: https://arxiv.org/abs/1711.05010

[36] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[37] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2000.

[38] K. Dogancay, *Partial-Update Adaptive Signal Processing: Design Analysis and Implementation*. New York, NY, USA: Academic, 2008.

[39] M. Godavarti and A. O. Hero, "Partial update LMS algorithms," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2382–2399, Jul. 2005.