Performance Analysis of Distributed Kalman Filtering with Partial Diffusion Over Noisy Network

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

Abstract—**The performance of partial diffusion Kalman filtering (PDKF) algorithm for the networks with noisy links is studied here. A closed-form expression for the steady-state mean square deviation is then derived and theoretically shown that when the links are noisy, the communication-performance trade-off, reported for the PDKF algorithm, does not hold. Additionally, optimal selection of combination weights is investigated and a combination rule along with an adaptive implementation is motivated. The results confirm the theoretical outcome.**

***Keywords*—Adaptive networks, combination weights, Kalman filtering, noisy links, partial diffusion.**

I. Introduction

Distributed collaborating learning with streaming noisy data is a challenging problem that appears in a wide range of existing and emerging applications. Some examples include navigation, target localization, collaborative spectral sensing, demand side management, autoregressive modeling and biomedical signal processing [1]–[4]. So far, different solutions (strategies) have been developed to solve distributed online learning problems. However, among the available methods, the diffusion-based strategies exhibit superior performance compared to others, e.g. incremental strategies (in terms of robustness to the link and node failure) and consensus strategies that operate in a single time-scale (in terms of stability, convergence rate and steady-state performance) [5]. However, for Kalman filters, it can be stated that diffusion is an effective and resource efficient technique, but there are trade-offs when comparing diffusion-based Kalman filters with their consensus counterparts.

In this paper, the problem of distributed state estimation for a multi-agent network is tackled. In [4], [6]–[13] different distributed Kalman filtering algorithms have been reported that are amenable to distributed implementations. In such algorithms, nodes cooperate closely through a topology to estimate the target state precisely. As opposed to the centralized estimation, the distributed estimation is more flexible for topology changes and robust to node/link failures. Diffusion Kalman filter (DKF) algorithm [9], as a distributed fusion and estimation method, shows better performance for the state estimation by diffusing information through a sequence of Kalman iterations and data-aggregation. In the DKF algorithm each agent performs an incremental step, to share the data, followed by a diffusion step, to share the estimate.

In diffusion strategy, each node (agent) in the network collaborates and diffuses information in real-time with its neighbors. Cooperation structure in diffusion-based adaptive networks makes them scalable and more robust to link/node failures [14]–[16]. In diffusion strategies, each node communicates with all other nodes, including itself, being located in the neighborhood to share intermediate estimate with them [17]. However, power and bandwidth resources are the major constraints for realization of a cooperative task in an adaptive network. They are limited for data transmission through radio links. Therefore, although the advantages of diffusion strategies are attained by expanding inter-node communications, they are subject to the communication cost constraints [18].

Reducing the amount of inter-node communications, while maintaining the benefits of cooperation is of practical importance. There exist some efforts related to reducing the amount of communication expenditure, such as reducing the dimension of the estimates [19]–[21], selecting a subset of entries of the intermediate estimate vectors [22]–[27], set-membership filtering [28]–[30], and partial updating [31] have been reported in [32]–[36]. The problem of distributed Kalman filtering was addressed in [37], where the authors proposed an efficient algorithm for large-scale systems. Distributed Kalman filtering with low-cost communication has been proposed previously in [38], where the decentralized state estimators of dynamical stochastic processes were derived and analyzed. Kalman filters with reduced order models have been studied, in e.g., [39], [40], to address the computation burden posed by implementing nth order models. Partitioned update Kalman filter (PUKF) that updates the state using multidimensional measurements in parts is discussed in [41].

Among these methods, we focus on partial diffusion Kalman filter (PDKF) algorithm proposed at [26]. The PDKF algorithm consists of an adaptation phase and a combination phase. In the adaptation phase, each node applies the Kalman filtering using recent input-output data and the existing estimate to obtain an intermediate estimate. In the combination phase, the nodes share a subset of their intermediate estimate entries (produced at the adaptation phase) and combine them to update the local estimates.

The results proposed in [26] are based on the assumption that the communications among the nodes are ideal, i.e., the information transmitted correctly among them. In practice, however, the performance of the adaptive networks is strongly affected by the presence of such a link state, where the communication links are noisy. This issue motivates us to investigate the performance of PDKF algorithm in such scenarios. Some useful results dealing with the effects of noisy links on the performance of diffusion-based strategies

V. Vahidpour, A. Rastegarnia, M. Latifi and A. Khalili and are with the Department of Electrical Engineering, Malayer University, Malayer 65719-95863, Iran (email: v.vahidpour@malayeru.ac.ir, rastegarnia@malayeru.ac.ir; milad.latifi@stu.malayeru.ac.ir; khalili@malayeru.ac.ir).

S. Sanei is with the School of Science and Technology, Nottingham Trent University, Nottingham, NG11 8NS, UK (email: saeid.sanei@ntu.ac.uk).

behavior are presented in [14]–[16], [24], [25]. In this paper, we figure out how the noisy links affect deterioration of the network performance during the exchange of weight estimates. Among other results, our analysis provides some insights in the communication cost and estimation performance for the PDKF algorithms under non-ideal links.

In summary, our contributions in this paper are as follows:

(i) The effect of noisy links on the mean and mean-square stability of the PDKF algorithm under certain statistical conditions is studied.

(ii) We derive the steady-state behavior of the PDKF algorithm in terms of the MSD metric.

(iii) It is shown theoretically that unlike the ideal case, there is no clear trade-off between the MSD performance and the communication cost. We also present the necessary simulations to support our theoretical findings.

(iv) It is demonstrated how the combination coefficients can be adjusted efficiently throughout the learning process to simultaneously lessen the influence of exchanging information over noisy communication links.

The remainder of this paper is organized as follows. We briefly introduce the PDKF algorithm in Section II. In Section III the PDKF algorithm in the presence of noisy links is formulated. We provide our analysis in Section IV. In Section V we discuss more about the communication-performance trade-off for the PDKF algorithm with noisy links. In section VI, an adaptive combination rule is established by an approximate solution that relies on optimizing an upper bound on the network MSD. The simulation results are presented in Section VII. Finally, we conclude in Section VIII.

We adopt small boldface letters for vectors and bold capital letters for matrices. Normal font letters denote scalars. We use subscripts to refer to the time-dependence of vector (matrix) variables, as in $\mathbf{y}_{k,i}$. $(\cdot)^T$ denotes transpose for matrices. The operator $\operatorname{tr}\{\cdot\}$ refers to the trace of its matrix argument and $\operatorname{vec}\{\cdot\}$ stacks the columns of its matrix argument on top of each other. We write $\operatorname{diag}\{\cdot\}$ to denote a (block) diagonal matrix formed from its arguments and and $\operatorname{col}\{\cdot\}$ to denote a column vector formed by stacking its arguments on top of each other. Finally, we use $\otimes$ and $\mathbb{1}$ to denote Kronecker product and a column vector with unity entries, respectively.

II. Background

A. System Description

We consider a connected network with N nodes, each labeled as $k = 1, 2, \ldots, N$. Two nodes k and l are neighbors if they can exchange data with each other, i.e., they are connected by an edge. Here, the definition of neighborhood of node k refers to the case in which k itself is included, i.e. $k \in \mathcal{N}_k$. Each node k takes observation $\mathbf{y}_{k,i} \in \mathbb{R}^P$ of a common environment state $\mathbf{x}_i \in \mathbb{R}^M$ at time instant i with local observation matrix $\mathbf{H}_{k,i} \in \mathbb{R}^{P\times M}$. The state-space model for every node k is of the form:

$$\begin{aligned}\mathbf{x}_{i+1} &= \mathbf{F}_i\mathbf{x}_i + \mathbf{G}_i\mathbf{n}_i\\ \mathbf{y}_{k,i} &= \mathbf{H}_{k,i}\mathbf{x}_i + \mathbf{v}_{k,i}\end{aligned} \tag{1}$$

Algorithm 1 Diffusion Kalman Filter [9]

```
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
    $\mathbf{R}_{e,i} \leftarrow \mathbf{R}_{l,i} + \mathbf{H}_{l,i}\mathbf{P}_{k,i}\mathbf{H}_{l,i}^T$
    $\boldsymbol{\psi}_{k,i} \leftarrow \boldsymbol{\psi}_{k,i} + \mathbf{P}_{k,i}\mathbf{H}_{l,i}^T\mathbf{R}_{e,i}^{-1}[\mathbf{y}_{l,i} - \mathbf{H}_{l,i}\boldsymbol{\psi}_{k,i}]$      (2)
    $\mathbf{P}_{k,i} \leftarrow \mathbf{P}_{k,i} - \mathbf{P}_{k,i}\mathbf{H}_{l,i}^T\mathbf{R}_{e,i}^{-1}\mathbf{H}_{l,i}\mathbf{P}_{k,i}$
end for
Step2: Diffusion Update
    $\hat{\mathbf{x}}_{k,i|i} \leftarrow \sum_{l\in\mathcal{N}_k} c_{lk}\boldsymbol{\psi}_{l,i}$      (3)
    $\mathbf{P}_{k,i|i} \leftarrow \mathbf{P}_{k,i}$
    $\hat{\mathbf{x}}_{k,i+1|i} = \mathbf{F}_i\hat{\mathbf{x}}_{k,i|i}$
    $\mathbf{P}_{k,i+1|i} = \mathbf{F}_i\mathbf{P}_{k,i|i}\mathbf{F}_i^T + \mathbf{G}_i\mathbf{Q}_i\mathbf{G}_i^T$
```

where $\mathbf{F}_i \in \mathbb{R}^{M\times M}$, $\mathbf{G}_i \in \mathbb{R}^{M\times M}$, $\mathbf{n}_i \in \mathbb{R}^M$, and $\mathbf{v}_{k,i} \in \mathbb{R}^P$ denote the model matrix, the state noise matrix, the state noise vector, and the observation noise vector of agent k at time i, respectively. Now, we can make the following assumptions for the state-space model represented by (1):

Assumption 1.

(i) (Gaussian processes). The measurement noise, $\mathbf{v}_{k,i}$, the state noise, $\mathbf{n}_i$, and the initial state of the system, $\mathbf{x}_0$, are Gaussian distributed sequences, with

$$\mathbf{v}_{k,i} \sim \mathcal{N}(0, \mathbf{R}_{k,i}), \mathbf{n}_i \sim \mathcal{N}(0, \mathbf{Q}_i), \mathbf{x}_0 \sim \mathcal{N}(0, \mathbf{\Pi}_0),$$

where $\mathbf{R}_{k,i} \in \mathbb{R}^{P\times P}$, $\mathbf{Q}_i \in \mathbb{R}^{M\times M}$, and $\mathbf{\Pi}_0 \in \mathbb{R}^{M\times M}$ are the corresponding covariance matrices. The measurement noise covariance matrix $\mathbf{R}_i$ of the global noise vector $\mathbf{v}_i = \operatorname{col}\{\mathbf{v}_{1,i}, \ldots, \mathbf{v}_{N,i}\}$ is block-diagonal matrix, i.e., $\mathbf{R}_i = \operatorname{diag}\{\mathbf{R}_{1,i}, \ldots, \mathbf{R}_{N,i}\}$ and positive definite, i.e., $\mathbf{R}_i > 0$

(ii) (Uncorrelated sequences). The measurement noise, the state noise, and the initial state: $\{\mathbf{v}_i, \mathbf{n}_i, \mathbf{x}_0\}$ are uncorrelated random vector sequences.

B. Diffusion Kalman Filtering Algorithm

For every node in the network the objective is to estimate the state vector $\mathbf{x}_i \in \mathbb{R}^M$ by exploiting its cooperation with the other nodes. As we mentioned earlier, the DKF algorithm is an effective tool for performing network-wide distributed Kalman filtering problem. The DKF algorithm in its time-and-measurement update form is given in Algorithm 1. 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. Then, the DKF 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 algorithm consists of two steps, namely the *incremental update* and *diffusion update*. In the incremental update step, first the nodes exchange local data $\{\mathbf{y}_{k,i}, \mathbf{H}_{k,i}, \mathbf{R}_{k,i}\}$ with their neighbors and at every time instant i compute $\boldsymbol{\psi}_{k,i} \leftarrow \hat{\mathbf{x}}_{k,i|i-1}$ and $\mathbf{P}_{k,i} \leftarrow \mathbf{P}_{k,i|i-1}$. Then, each node performs KF with the available data to obtain the intermediate estimates $\boldsymbol{\psi}_{k,i}$ as given by (2).

The symbol $\leftarrow$ denotes a sequential assignment. In the diffusion step, the nodes share intermediate estimates $\boldsymbol{\psi}_{k,i}$ and use (3) to compute a convex combination of intermediate estimates and obtain the local estimate $\hat{\mathbf{x}}_{k,i|i}$. The scalars $\{c_{lk}\}$ denote nonnegative convex combination coefficients that satisfy the following conditions:

$$\sum_{l=1}^{N} c_{lk} = 1, \quad c_{lk} = 0 \text{ if } l \notin \mathcal{N}_k, \ \forall l, k \tag{4}$$

The coefficients $\{c_{lk}\}$ are free weighting parameters and their selection influences the algorithm performance. If all the coefficients are combined in 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., $\mathbb{1}^T \mathbf{C} = \mathbb{1}^T$, where $\mathbb{1}$ denotes an $N \times 1$ vector with unit entries. This means that $\mathbf{C}$ is left stochastic matrix, i.e., the magnitudes of its eigenvalues are bounded by one.

C. Partial Diffusion Kalman Filtering

The PDKF algorithm proposed in [26] is shown by Algorithm 2. For every agent k, the objective of PDKF implementation is to recursively estimate the unknown state $\mathbf{x}_i \in \mathbb{R}^M$, while sharing a subset of its intermediate estimate vector with its neighbors $l \in \mathcal{N}_k$. To further reduce the communication costs in PDKF algorithm, the nodes do not share their local information $\{\mathbf{y}_{k,i}, \mathbf{H}_{k,i}, \mathbf{R}_{k,i}\}$. Therefore, this algorithm particularly depends on communication of chosen entries of $\boldsymbol{\psi}_{k,i}$.

To reduce the communication burden (by means of reducing the internode communications) every node k at any time instant i is allowed to select and propagate a subset of its intermediate state estimate vector (say L out of M, $0 \leq L \leq M$, entries). In the partial diffusion algorithm, at time instant i the nodes need to know which elements of their neighbors intermediate estimates have diffused. Consequently, the address, i.e., position in the vector of communicated elements should be transmitted as well. In the next section, we present two partial diffusion schemes that bias the need for addressing. The selecting and scattering task can be realized by a diagonal selection matrix, $\mathbf{T}_{k,i} \in \mathbb{R}^{M \times M}$ which has L ones and $M - L$ zeros on its diagonal. The positions of the ones determine which entries of the intermediate state estimate of node k are selected to be diffused at time i. It is obvious that multiplication of $\boldsymbol{\psi}_{k,i}$ by $\mathbf{T}_{k,i}$ gives a vector whose non-selected entries (determined by $\mathbf{T}_{k,i}$) are zero.

Remark 1. *Note that in practice L is fixed and pre-specified by the designer [22].*

At every time instant i, the agents run Kalman filtering algorithm for diffusion step and communicate to their neighbors the intermediate estimate for aggregation step. Equation (6) is utilized to aggregate the partially received intermediate estimates. The combination coefficients $\{c_{lk}\}$ in (6) are the same as those in (4).

Algorithm 2 Partial Diffusion Kalman Filter [26]

Initialization: $\hat{\mathbf{x}}_{k,0|-1} = 0$ and $\mathbf{P}_{k,0|-1} = \mathbf{\Pi}_0$
For $i \geq 0$, every node k computes

Step1: Adaptation phase
$\mathbf{P}_{k,i} \leftarrow \mathbf{P}_{k,i|i-1}$

$$\begin{aligned} &\mathbf{R}_{e,i} \leftarrow \mathbf{R}_{k,i} + \mathbf{H}_{k,i}\mathbf{P}_{k,i}\mathbf{H}_{k,i}^T \\ &\boldsymbol{\psi}_{k,i} \leftarrow \hat{\mathbf{x}}_{k,i|i-1} + \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}[\mathbf{y}_{k,i} - \mathbf{H}_{k,i}\hat{\mathbf{x}}_{k,i|i-1}] \\ &\mathbf{P}_{k,i} \leftarrow \mathbf{P}_{k,i} - \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}\mathbf{H}_{k,i}\mathbf{P}_{k,i} \end{aligned} \tag{5}$$

Step2: Combination Phase

$$\hat{\mathbf{x}}_{k,i|i} = \boldsymbol{\psi}_{k,i} + \sum_{l \in \mathcal{N}_k / \{k\}} c_{lk}\mathbf{T}_{l,i}(\boldsymbol{\psi}_{l,i} - \boldsymbol{\psi}_{k,i}) \tag{6}$$

$$\begin{aligned} \mathbf{P}_{k,i|i} &\leftarrow \mathbf{P}_{k,i} \\ \hat{\mathbf{x}}_{k,i+1|i} &= \mathbf{F}_i\hat{\mathbf{x}}_{k,i|i} \\ \mathbf{P}_{k,i+1|i} &= \mathbf{F}_i\mathbf{P}_{k,i|i}\mathbf{F}_i^T + \mathbf{G}_i\mathbf{Q}_i\mathbf{G}_i^T \end{aligned}$$

From equation (6), it is obvious that the unavailable elements are replaced by their equivalent ones in each node's own intermediate estimate vector.

III. The PDKF with Additive Noise

In this paper, we investigate the performance of PDKF algorithm in a more realistic condition, when the data exchanges are corrupted by additive noise. This will be developed further in a later section which describes the performance analysis both in mean and mean-square senses. Doing so, we model the noisy data received by node k from its neighbor l as follows:

$$\boldsymbol{\psi}_{lk,i} = \boldsymbol{\psi}_{l,i} + \mathbf{w}_{lk,i} \tag{7}$$

where $\mathbf{w}_{lk,i}$ $(M \times 1)$ is the noise vector that perturbs the exchanged data over the link lk. The subscript lk is used to designate node l as the source and node k as the sink.

To proceed further, we start by imposing the assumptions on the link noise as follows.

Assumption 2.

(i) (Independent Noise Sequence). The additive noises $\{\mathbf{w}_{lk,i}\}_{1 \leq l,k \leq N, i \geq 0} \in \mathbb{R}^M$ are temporally white and spatially independent random processes with zero mean and variances $\sigma_{w,lk}^2$ so that

$$\begin{aligned} \mathbb{E}\left[\mathbf{w}_{lk,i}\right] = 0, \quad \forall 1 \leq l, k \leq N, \quad i \geq 0, \\ \mathbb{E}\left[\mathbf{w}_{lk,i}\mathbf{w}_{lk,i}^T\right] = \mathbf{Z}_{lk} = \sigma_{w,lk}^2\mathbf{I}_M \end{aligned}$$

(ii) (Independent Sequences). The link noise processes are independent of noise and state sequences $\{\mathbf{v}_i, \mathbf{n}_i, \mathbf{x}_i\}$.

Thus, using the perturbed data (7), the PDKF algorithm with imperfect exchange can be described by (5) as its adaptation phase and the following equation as its combination phase:

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAES.2019.2933961, IEEE Transactions on Aerospace and Electronic Systems

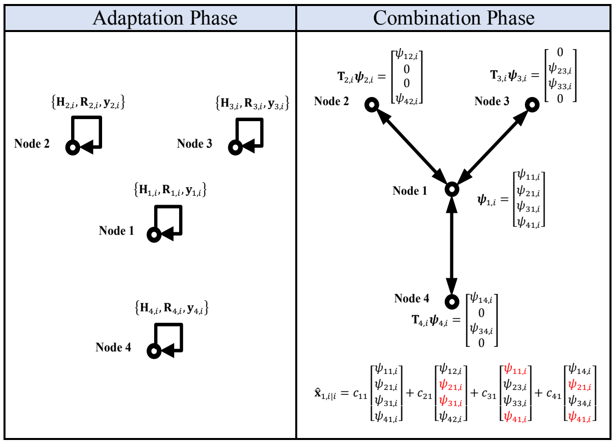

Adaptation Phase
Combination Phase
$\{\mathbf{H}_{2,i}, \mathbf{R}_{2,i}, \mathbf{y}_{2,i}\}$
Node 2
$\{\mathbf{H}_{3,i}, \mathbf{R}_{3,i}, \mathbf{y}_{3,i}\}$
Node 3
$\{\mathbf{H}_{1,i}, \mathbf{R}_{1,i}, \mathbf{y}_{1,i}\}$
Node 1
$\{\mathbf{H}_{4,i}, \mathbf{R}_{4,i}, \mathbf{y}_{4,i}\}$
Node 4
$\mathbf{T}_{2,i}\boldsymbol{\psi}_{2,i} = \begin{bmatrix}\psi_{12,i}\\0\\0\\\psi_{42,i}\end{bmatrix}$
Node 2
$\mathbf{T}_{3,i}\boldsymbol{\psi}_{3,i} = \begin{bmatrix}0\\\psi_{23,i}\\\psi_{33,i}\\0\end{bmatrix}$
Node 3
Node 1
$\boldsymbol{\psi}_{1,i} = \begin{bmatrix}\psi_{11,i}\\\psi_{21,i}\\\psi_{31,i}\\\psi_{41,i}\end{bmatrix}$
Node 4
$\mathbf{T}_{4,i}\boldsymbol{\psi}_{4,i} = \begin{bmatrix}\psi_{14,i}\\0\\\psi_{34,i}\\0\end{bmatrix}$
$\hat{\mathbf{x}}_{1,i|i} = c_{11}\begin{bmatrix}\psi_{11,i}\\\psi_{21,i}\\\psi_{31,i}\\\psi_{41,i}\end{bmatrix} + c_{21}\begin{bmatrix}\psi_{12,i}\\\psi_{21,i}\\\psi_{31,i}\\\psi_{42,i}\end{bmatrix} + c_{31}\begin{bmatrix}\psi_{11,i}\\\psi_{23,i}\\\psi_{33,i}\\\psi_{41,i}\end{bmatrix} + c_{41}\begin{bmatrix}\psi_{14,i}\\\psi_{21,i}\\\psi_{34,i}\\\psi_{41,i}\end{bmatrix}$

Fig. 1. Partial diffusion Kalman filter update at node k and time i.

This process is also demonstrated schematically in Fig. 1. Equation (8) can be written as:

$$\hat{\mathbf{x}}_{k,i|i} = \boldsymbol{\psi}_{k,i} + \sum_{l \in \mathcal{N}_k/\{k\}} c_{lk}\mathbf{T}_{l,i}(\boldsymbol{\psi}_{l,i} - \boldsymbol{\psi}_{k,i}) + \mathbf{w}_{k,i} \tag{9}$$

where in (9), $\mathbf{w}_{k,i} \in \mathbb{R}^M$ denotes the aggregate link noise signal given by:

$$\mathbf{w}_{k,i} = \sum_{l \in \mathcal{N}_k/\{k\}} c_{lk}\mathbf{T}_{l,i}\mathbf{w}_{lk,i} \tag{10}$$

This noise term represents the total impact on agent k of all the selected exchange noises from its neighbors while exchanging the estimates $\{\boldsymbol{\psi}_{l,i}, l \in \mathcal{N}_k/\{k\}\}$ during the aggregation phase. $\mathbf{w}_{k,i}$ is zero mean with covariance matrix of $\mathbf{Z}_k$.

A. Entry Selection Methods

To reduce the internode communications, it is permitted to transmit L-subset of a set on M elements containing exactly L elements of the intermediate estimate vectors from each node k at each iteration i. Doing so, we employ two different schemes called coordinated and uncoordinated partial diffusion, presented in [23]. These schemes need less memory and are significantly easier to implement in comparison with the schemes proposed in [22], called sequential and stochastic partial diffusion. Let us define $M \times M$ the entry-selection matrix of node k at time instant i as

$$\mathbf{T}_{k,i} = \text{diag}\left\{t_{1k,i}, t_{2k,i}, \ldots, t_{Mk,i}\right\}$$

and update it via a circular shift as

$$\begin{aligned}\mathbf{T}_{k,i} &= \mathbf{U}_M\mathbf{T}_{k,i-1}\mathbf{U}_M^T\\ &= \text{diag}\left\{t_{Mk,i-1}, t_{1k,i-1}, \ldots, t_{(M-1)k,i-1}\right\}\end{aligned}$$

where $\mathbf{U}_M$ is an $M \times M$ lower shift matrix with ones below the main diagonal and zeros elsewhere. The initial entry-selection matrix of node k, $\mathbf{T}_{k,1}$ has L ones and $M - L$ zeros on its main diagonal. As mentioned earlier, the ones on the diagonal of $\mathbf{T}_{k,i}$ indicate the selected entries of local estimate at node k for the transmission at iteration i. Again, $\boldsymbol{\psi}_{k,i}$ by $\mathbf{T}_{k,i}$ that has L ones and $M - L$ zeros on its diagonal replaces its non-selected entries with zero.

Remark 2. *The transmission probability for every entry at any node k is*

$$\rho = \frac{L}{M}$$

This implies that each entry is communicated L times within each M consecutive iteration.

If all nodes utilize the same initial entry-selection matrices, i.e.,

$$\mathbf{T}_{1,1} = \mathbf{T}_{2,1} = \ldots = \mathbf{T}_{N,1}$$

the partial diffusion scheme is named *coordinated* scheme. On the other hand, different initial entry-selection matrices for every node leads to *uncoordinated* partial diffusion scheme. 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.

IV. Performance Analysis

In this section, we examine the performance of the PDKF algorithm in terms of mean and mean-square senses considering both coordinated and uncoordinated partial diffusion schemes. We also provide an expression for its steady-state MSD. For every node k, the MSD metric is defined as follows:

$$\text{MSD}_{k,i} = \mathbb{E}\left[\|\mathbf{x}_i - \hat{\mathbf{x}}_{k,i|i}\|^2\right] \tag{11}$$

In the analysis, we need to consider the fact that the input data is stochastic in nature and utilize the energy conservation argument to analyze the stability of algorithms in mean and mean square senses.

A. Network Update Equation

To proceed with the analysis, we derive the network update equation in the following section. Let $\tilde{\boldsymbol{\psi}}_{k,i} = \mathbf{x}_i - \boldsymbol{\psi}_{k,i}$ denote the estimation error at the end of adaptation phase and $\tilde{\mathbf{x}}_{k,i|i-1} = \mathbf{x}_i - \hat{\mathbf{x}}_{k,i|i-1}$ denote the estimation error at node k at the end of aggregation phase. Subtracting (5) from $\mathbf{x}_i$ and using model (1) gives

$$\begin{aligned}\tilde{\boldsymbol{\psi}}_{k,i} &= \tilde{\mathbf{x}}_{k,i|i-1} - \mathbf{P}_{k,i|i-1}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}(\mathbf{H}_{k,i}\tilde{\mathbf{x}}_{k,i|i-1} + \mathbf{v}_{k,i})\\ &= (\mathbf{I}_M - \mathbf{P}_{k,i|i-1}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}\mathbf{H}_{k,i})\tilde{\mathbf{x}}_{k,i|i-1}\\ &\qquad - \mathbf{P}_{k,i|i-1}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}\mathbf{v}_{k,i}\end{aligned} \tag{12}$$

Using the matrix inversion lemma (see appendix A for details), we obtain $\mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{k,i}^{-1} = \mathbf{P}_{k,i|i-1}\mathbf{H}_{k,i}^T\mathbf{R}_{e,i}^{-1}$ and conclude

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

where $\mathbf{S}_{k,i} \triangleq \mathbf{H}_{k,i}^T\mathbf{R}_{k,i}^{-1}\mathbf{H}_{k,i}$. Using the state-space model (1) we have

$$\tilde{\mathbf{x}}_{k,i|i-1} = \mathbf{F}_{i-1}\tilde{\mathbf{x}}_{k,i-1|i-1} + \mathbf{G}_{i-1}\mathbf{n}_{i-1} \tag{14}$$

Substituting (14) into (13) gives

$$\begin{aligned}\widetilde{\boldsymbol{\psi}}_{k,i} &= (\mathbf{I}_M - \mathbf{P}_{k,i}\mathbf{S}_{k,i})\mathbf{F}_{i-1}\tilde{\mathbf{x}}_{k,i-1|i-1} \\ &\quad + (\mathbf{I}_M - \mathbf{P}_{k,i}\mathbf{S}_{k,i})\mathbf{G}_{i-1}\mathbf{n}_{i-1} - \mathbf{P}_{k,i}\mathbf{H}_{k,i}^T\mathbf{R}_{k,i}^{-1}\mathbf{v}_{k,i}\end{aligned} \tag{15}$$

To derive the network update equation, we define the augmented state-error vectors $\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}$ and $\widetilde{\boldsymbol{\Psi}}_i$, measurement noise vector $\mathbf{v}_i$, the aggregate link noise signals $\mathbf{w}_i$ and block-diagonal matrices $\boldsymbol{\mathcal{H}}_i$, $\boldsymbol{\mathcal{P}}_{i|i}$, $\boldsymbol{\mathcal{S}}_i$ and $\boldsymbol{\mathcal{B}}_i$ as follows

$$\begin{aligned}\widetilde{\boldsymbol{\mathcal{X}}}_{i|i} &= \operatorname{col}\left\{\widetilde{\mathbf{x}}_{1,i|i}, \widetilde{\mathbf{x}}_{2,i|i}, \ldots, \widetilde{\mathbf{x}}_{N,i|i}\right\} \\ \widetilde{\boldsymbol{\Psi}}_i &= \operatorname{col}\left\{\widetilde{\boldsymbol{\psi}}_{1,i}, \widetilde{\boldsymbol{\psi}}_{2,i}, \ldots, \widetilde{\boldsymbol{\psi}}_{N,i}\right\} \\ \mathbf{v}_i &= \operatorname{col}\left\{\mathbf{v}_{1,i}, \ldots, \mathbf{v}_{N,i}\right\} \\ \mathbf{w}_i &= \operatorname{col}\left\{\mathbf{w}_{1,i}, \ldots, \mathbf{w}_{N,i}\right\} \\ \boldsymbol{\mathcal{H}}_i &= \operatorname{diag}\left\{\mathbf{H}_{1,i}, \mathbf{H}_{2,i}, \ldots, \mathbf{H}_{N,i}\right\} \\ \boldsymbol{\mathcal{P}}_i &= \operatorname{diag}\left\{\mathbf{P}_{1,i}, \mathbf{P}_{2,i}, \ldots, \mathbf{P}_{N,i}\right\} \\ \boldsymbol{\mathcal{S}}_i &= \operatorname{diag}\left\{\mathbf{S}_{1,i}, \mathbf{S}_{2,i}, \ldots, \mathbf{S}_{N,i}\right\} \\ \boldsymbol{\mathcal{B}}_i &= \begin{bmatrix}\mathbf{B}_{1,1,i} & \cdots & \mathbf{B}_{1,N,i} \\ \vdots & \ddots & \vdots \\ \mathbf{B}_{N,1,i} & \cdots & \mathbf{B}_{N,N,i}\end{bmatrix}.\end{aligned}$$

where

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

Using the above definitions, we can express (9) and (15) in a global form that captures the evolution of entire network:

$$\widetilde{\boldsymbol{\mathcal{X}}}_{i|i} = \boldsymbol{\mathcal{B}}_i\widetilde{\boldsymbol{\Psi}}_i - \mathbf{w}_i \tag{16}$$

$$\begin{aligned}\widetilde{\boldsymbol{\Psi}}_i &= (\mathbf{I}_{MN} - \boldsymbol{\mathcal{P}}_i\boldsymbol{\mathcal{S}}_i)\left[(\mathbf{I}_N \otimes \mathbf{F}_{i-1})\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}\right. \\ &\quad \left. + (\mathbf{I}_N \otimes \mathbf{G}_{i-1})(\mathbb{1} \otimes \mathbf{n}_{i-1})\right] - \boldsymbol{\mathcal{P}}_i\boldsymbol{\mathcal{H}}_i^T\mathbf{R}_i^{-1}\mathbf{v}_i\end{aligned} \tag{17}$$

Equation (17) can be rewritten in a more compact form as

$$\widetilde{\boldsymbol{\Psi}}_i = \boldsymbol{\mathcal{F}}_i\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1} + \boldsymbol{\mathcal{G}}_i(\mathbb{1} \otimes \mathbf{n}_{i-1}) - \boldsymbol{\mathcal{D}}_i\mathbf{v}_i \tag{18}$$

where

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

Substituting (18) into (16) gives the following recursion for the update of the network state-error vectors:

$$\begin{aligned}\widetilde{\boldsymbol{\mathcal{X}}}_{i|i} &= \boldsymbol{\mathcal{B}}_i\boldsymbol{\mathcal{F}}_i\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1} + \boldsymbol{\mathcal{B}}_i\boldsymbol{\mathcal{G}}_i(\mathbb{1} \otimes \mathbf{n}_{i-1}) \\ &\quad - \boldsymbol{\mathcal{B}}_i\boldsymbol{\mathcal{D}}_i\mathbf{v}_i - \mathbf{w}_i\end{aligned} \tag{19}$$

Equation (19) shows how the network weight error vector $\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}$ evolves in time. In the sequel, we will use this recursive equation to investigate the convergence and steady-state behavior of PDKF algorithm with noisy links.

In order to analyze the mean-square steady-state performance, we introduce the following assumptions:

Assumption 3.

(i) (Time Dependency). The matrices $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$, $\mathbf{R}$ *and* $\mathbf{Q}$ *described in model* (1) *are time invariant.*

(ii) (Stability) we assume that matrix $\boldsymbol{\mathcal{F}} = \lim_{i\to\infty}\boldsymbol{\mathcal{F}}_i$ *is stable, i.e., its eigenvalues locate inside the unit disc.*

(iii) (Ergodicity). The matrices $\{\mathbf{F}, \mathbf{H}_k\}$ *are detectable for every* k *and* $\{\mathbf{F}, \mathbf{G}\mathbf{Q}^{\frac{1}{2}}\}$ *is stabilizable [26].*

Under *Assumption* 3-(iii), $\mathbf{P}_{k,i|i-1}$ converges to $\mathbf{P}_k^-$ and $\mathbf{P}_{k,i|i}$ converges to $\mathbf{P}_k$, for all k. Under these assumptions, the 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

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

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.

B. Mean Performance

Proposition 1 summarizes the mean performance of the PDKF algorithm over a network with noisy links.

Proposition 1. *Under Assumptions* (1)-(2) *the PDKF algorithm over a network with noisy links is convergent in the mean sense and asymptotically unbiased.*

Proof: In view of *Assumptions* (1) and (2), we can find that after tacking the expectation on both sides of (19), the mean error vector evolves according to the following recursion

$$\mathbb{E}\left[\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}\right] = \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{F}}_i\mathbb{E}\left[\widetilde{\boldsymbol{\mathcal{X}}}_{i-1|i-1}\right] \tag{20}$$

where $\boldsymbol{\mathcal{M}} = \mathbb{E}\left[\boldsymbol{\mathcal{B}}_i\right]$. To guarantee the convergence of (20), the coefficient matrix $\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{F}}_i$ must be stable. As stated in Appendix A of [23], all entries of $\boldsymbol{\mathcal{M}} = \mathbb{E}\left[\boldsymbol{\mathcal{B}}_i\right]$ are real non-negative and each of the rows sums to unity, i.e., $\boldsymbol{\mathcal{M}}\mathbb{1} = \mathbb{1}$. This implies that $\boldsymbol{\mathcal{M}}$ is a *right-stochastic* matrix. The Perron-Frobenius theorem ensures that the spectral radius (the eigenvalue with the largest absolute value) of a stochastic matrix is equal to one [42]. Therefore, as $i \to \infty$, the mean stability and asymptotic unbiasedness of the algorithm is achieved if matrix $\boldsymbol{\mathcal{F}}$ is stable. It can be shown that the matrix $\boldsymbol{\mathcal{F}}$ is stable under *Assumptions* (3) (see [9], for details). Therefore, we have

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

where $\boldsymbol{\mathcal{O}}_{MN} \in \mathbb{R}^{MN}$ denotes zero vector. This means at the presence of disturbances during information exchange, the PDKF algorithm, is convergent in the mean sense and asymptotically unbiased. ■

C. Mean-square Performance

Let $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}},i} = \mathbb{E}\left[\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}^T\right]$ denote the covariance matrix of estimation errors of the PDKF algorithm, $\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}$. From (16) and Assumptions (1) and (2) the following relation can be obtained:

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

Employing vectorization operator and using the properties of Kronecker product[1]

$$\begin{aligned}\text{vec}\left\{\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}},i}\right\} =& \mathfrak{B}\left(\boldsymbol{\mathcal{F}}_i\otimes\boldsymbol{\mathcal{F}}_i\right)\text{vec}\left\{\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}},i-1}\right\}\\ &+\mathfrak{B}\text{vec}\left\{\boldsymbol{\mathcal{G}}_i\left(\mathbb{1}\mathbb{1}^T\otimes\mathbf{Q}_{i-1}\right)\boldsymbol{\mathcal{G}}_i^T+\boldsymbol{\mathcal{D}}_i\mathbf{R}_i\boldsymbol{\mathcal{D}}_i^T\right\}\\ &+\text{vec}\left\{\mathbf{Z}\right\} \qquad (22)\end{aligned}$$

where $\mathbf{Z} = \mathbb{E}\left[\mathbf{w}_i\mathbf{w}_i^T\right] = \text{diag}\left\{\mathbf{Z}_1,\ldots,\mathbf{Z}_N\right\}$, and $\mathfrak{B} = \mathbb{E}\left[\boldsymbol{\mathcal{B}}_i\otimes\boldsymbol{\mathcal{B}}_i\right]$. We also use the fact that the expectation and vectorization operations commute.

Let $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}$ denote the steady-state values of matrix $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}},i}$. We can solve (22) to obtain the steady-state covariance matrix of $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}},i}$ (i.e. $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}$) as:

$$\begin{aligned}\text{vec}\left\{\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}\right\} =& \left[\mathbf{I}_{M^2N^2}-\mathfrak{B}\left(\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}}\right)\right]^{-1}\mathfrak{B}\text{vec}\left\{\boldsymbol{\mathcal{L}}+\boldsymbol{\mathcal{K}}\right\}\\ &+\left[\mathbf{I}_{M^2N^2}-\mathfrak{B}\left(\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}}\right)\right]^{-1}\text{vec}\left\{\mathbf{Z}\right\}\\ &\qquad (23)\end{aligned}$$

where $\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{G}}\left(\mathbb{1}\mathbb{1}^T\otimes\mathbf{Q}\right)\boldsymbol{\mathcal{G}}^T$ and $\boldsymbol{\mathcal{K}} = \boldsymbol{\mathcal{D}}\mathbf{R}\boldsymbol{\mathcal{D}}^T$. The matrix covariance $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}$ can be recovered from $\text{vec}\left\{\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}\right\}$. It is noteworthy to state that as $\boldsymbol{\mathcal{F}}$ is stable $\left[\mathbf{I}_{M^2N^2}-\mathfrak{B}\left(\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}}\right)\right]^{-1}$ is invertible.

The MSD at node k can be obtained by weighting $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}$ with a single-entry block matrix, $\boldsymbol{\mathcal{I}}_{NM}^k \in \mathbb{R}^{MN\times MN}$, with block of size $M\times M$, which its elements are all zeros, except block (k,k) which is an identity matrix. So, the MSD at node k can be expressed as:

$$\begin{aligned}\text{MSD}_k &= \lim_{i\to\infty}\mathbb{E}\left[\left\|\mathbf{x}_i-\hat{\mathbf{x}}_{k,i|i}\right\|^2\right]\\ &= \lim_{i\to\infty}\mathbb{E}\left[\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}\widetilde{\boldsymbol{\mathcal{X}}}_{i|i}^T\boldsymbol{\mathcal{I}}_{NM}^k\right] = \text{tr}\left\{\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}\boldsymbol{\mathcal{I}}_{NM}^k\right\} \quad (24)\end{aligned}$$

Remark 3. *In order to better clarify* (24)*, and discuss in more detail the extraction of MSD expression, we note that the kth block on the main diagonal of* $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}$ *is* $\mathbb{E}\left[\tilde{\mathbf{x}}_{k,i|i}\tilde{\mathbf{x}}_{k,i|i}^T\right]$*. Multiplying* $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}$ *by* $\boldsymbol{\mathcal{I}}_{NM}^k$ *gives the kth block of* $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}$*. Therefore, the trace of* $\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}\boldsymbol{\mathcal{I}}_{NM}^k$ *is the steady-state variance value* $\lim_{i\to\infty} E\|\tilde{\mathbf{x}}_{k,i|i}\|^2$*. It is easy to generalize this for the network MSD in the same way. Thus, the network steady-state MSD, expressed as the average steady-state MSD of all the agents, is*

$$\text{MSD}^{\text{network}} = \frac{1}{N}\text{tr}\left\{\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}\right\} \qquad (25)$$

Applying transpose to both sides of (23) and multiplying it by $\text{vec}\left\{\mathbf{I}_{MN}\right\}$ yields the following relation:

$$\begin{aligned}\text{tr}\left\{\boldsymbol{\mathcal{P}}_{\bar{\boldsymbol{\mathcal{X}}}}\right\} =& \left(\text{vec}^T\left\{\boldsymbol{\mathcal{L}}+\boldsymbol{\mathcal{K}}\right\}\mathfrak{B}^T+\text{vec}^T\left\{\mathbf{Z}\right\}\right)\times\\ &\left(\mathbf{I}_{M^2N^2}-(\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T)\mathfrak{B}^T\right)^{-1}\mathbf{h} \quad (26)\end{aligned}$$

where $\mathbf{h} = \text{vec}\left\{\mathbf{I}_{MN}\right\}$. We summarize the steady-state performance of the PDKF algorithm with noisy links in the following Proposition.

Proposition 2. *Let Assumptions* (1)-(3) *hold. Then, for the PDKF algorithm over a network with noisy links it holds that*

$$\begin{aligned}&\lim_{i\to\infty}\frac{1}{N}\sum_{k=1}^N\mathbb{E}\left[\left\|\mathbf{x}_i-\hat{\mathbf{x}}_{k,i|i}\right\|^2\right] =\\ &\frac{1}{N}\left(\text{vec}^T\left\{\boldsymbol{\mathcal{L}}+\boldsymbol{\mathcal{K}}\right\}\mathfrak{B}^T+\text{vec}^T\left\{\mathbf{Z}\right\}\right)\times\\ &\left(\mathbf{I}_{M^2N^2}-(\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T)\mathfrak{B}^T\right)^{-1}\mathbf{h}\end{aligned}$$

Now, we use recursion (22) to derive the required conditions for mean-square convergence as in the following proposition.

Proposition 3. *Under the Assumptions* (1)-(3)*, the recursion* (22) *is stable and convergent if and only if, the matrix* $\mathfrak{B}\left(\boldsymbol{\mathcal{F}}_i\otimes\boldsymbol{\mathcal{F}}_i\right)$ *is stable.*

Proof: All the entries of $\mathfrak{B}$ are real and non-negative. Moreover, we have

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

This means that the elements of each row of $\mathfrak{B}$ adds up to one. Since the block matrix $\mathfrak{B}$ is *right-stochastic* matrix, as a result of Perron-Frobenius theorem [42], it has unit spectral radius (see *Appendix B* of [23] for more details on how $\mathfrak{B}$ right-stochastic). In steady-state, as $i\to\infty$, the PDKF algorithm is stable in the meas-square sense if, and only if, $(\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}})$ is stable. Moreover, the eigenvalues of $A\otimes A$ are square of the eigenvalues of A [43]. Therefore, the stability of $(\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}})$ has the same conditions as the stability of $\boldsymbol{\mathcal{F}}$. this means that, as $\boldsymbol{\mathcal{F}}$ is stable (see [9], for details) then, the partial diffusion KF algorithm under imperfect information exchange is convergent and stable in mean-square sense, and the steady-state network MSD is given by (25).

V. Detailed Discussion on the Communication Trade-off

A. Comparison with Ideal Links

As we mentioned earlier, when connecting links among the nodes are ideal, there is a trade-off between the estimation

[1] For any arbitrary matrices $\{\mathbf{X}, Y, Z, W\}$ of compatible dimensions: we have the following recursion

$$\begin{aligned}\left(\mathbf{X}\otimes\mathbf{Y}\right)\left(\mathbf{Z}\otimes\mathbf{W}\right) &= \left(\mathbf{XZ}\right)\otimes\left(\mathbf{YW}\right)\\ \text{vec}\left\{\mathbf{XYZ}\right\} &= \left(\mathbf{Z}^T\otimes\mathbf{X}\right)\text{vec}\left\{\mathbf{Y}\right\}\\ \text{tr}\left\{\mathbf{X}^T\mathbf{Y}\right\} &= \text{vec}^T\left\{\mathbf{X}\right\}\text{vec}\left\{\mathbf{Y}\right\}\end{aligned}$$

performance and the communication cost. In other words, increasing the number of selected entries improves the estimation performance, but the communication load increases as well. In the sequel, we explicitly show that in the presence of noisy links the mentioned communication-performance trade-off does not hold. To this end, we consider the coordinated scheme in our analysis, but the analysis can be easily extended to include the uncoordinated partial diffusion. Moreover, to make the analysis more tractable we consider the following assumptions.

Assumption 4.

(i) During any M consecutive iterations, each node's intermediate estimate vector does not change considerably.

(ii) The matrix $\mathbf{C}$ is doubly stochastic.

Under first item on Assumption 4, at every M successive time iterations, all the entries of node's intermediate estimate vectors are scattered at any L iterations. This translates to approximate of coordinate partial diffusion scheme by periodic diffusion. In light of *Appendix C* at [22] , for periodic diffusion scheme, we have

$$\mathfrak{B} = (1-\rho)\,\mathbf{I}_{M^2N^2} + \rho\mathfrak{C}\otimes\mathfrak{C} \tag{27}$$

where $\mathfrak{C} = \mathbf{C}\otimes\mathbf{I}_M$.

To proceed, let's denote the average steady-state network MSD of the PDKF algorithm (with L selected entries) under ideal and noisy links conditions by $\acute{\eta}_L$ and η_L, respectively. The following proposition indicates that when the communication links are noisy, the stated communication-performance trade-off for the PDLF algorithm reported in [26] does not hold.

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

$$\acute{\eta}_M < \cdots < \acute{\eta}_L < \cdots < \acute{\eta}_1 < \eta_1 < \cdots < \eta_L < \cdots < \eta_M \tag{28}$$

Proof: The network MSD can be expressed in an alternative series expansion form. Since $\left(\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T\right)\mathfrak{B}^T$ is stable when the algorithm is mean-square stable, the term $\left(\mathbf{I}_{M^2N^2} - (\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T)\mathfrak{B}^T\right)^{-1}$ can be expanded as

$$\begin{aligned}\left(\mathbf{I}_{M^2N^2} - (\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T)\mathfrak{B}^T\right)^{-1} &= \\ \mathbf{I}_{M^2N^2} + (\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T)\mathfrak{B}^T &+ \left((\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T)\mathfrak{B}^T\right)^2 + \ldots \\ &= \sum_{j=0}^{\infty}\left[\left(\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T\right)\mathfrak{B}^T\right]^j\end{aligned}$$

which means that we can express the network MSD as follows:

$$\begin{aligned}\eta_L &= \frac{1}{N}\mathrm{vec}^T\{\Omega\}\,\mathfrak{B}^T\sum_{j=0}^{\infty}\left[\left(\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T\right)\mathfrak{B}^T\right]^j\mathbf{h} \\ &+ \frac{1}{N}\mathrm{vec}^T\{\mathbf{Z}\}\sum_{j=0}^{\infty}\left[\left(\boldsymbol{\mathcal{F}}^T\otimes\boldsymbol{\mathcal{F}}^T\right)\mathfrak{B}^T\right]^j\mathbf{h}\end{aligned} \tag{29}$$

where $\Omega = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{K}}$. Substituting (27) to (29) gives

$$\begin{aligned}\eta_L &= \frac{1}{N}\sum_{j=0}^{\infty}(1-\rho)^{j+1}\mathrm{vec}^T\{\boldsymbol{\Omega}\}\left[(\boldsymbol{\mathcal{F}}^T)^j\otimes(\boldsymbol{\mathcal{F}}^T)^j\right]\mathbf{h} \\ &+ \frac{1}{N}\sum_{j=0}^{\infty}\rho^{j+1}\mathrm{vec}^T\{\boldsymbol{\Omega}\}\left[\mathfrak{C}^T(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\otimes\mathfrak{C}^T(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right]\mathbf{h} \\ &+ \frac{1}{N}\sum_{j=0}^{\infty}(1-\rho)^j\mathrm{vec}^T\{\mathbf{Z}\}\left[(\boldsymbol{\mathcal{F}}^T)^j\otimes(\boldsymbol{\mathcal{F}}^T)^j\right]\mathbf{h} \\ &+ \frac{1}{N}\sum_{j=0}^{\infty}\rho^j\mathrm{vec}^T\{\mathbf{Z}\}\left[(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\otimes(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right]\mathbf{h} \\ &= \frac{1}{N}\sum_{j=0}^{\infty}(1-\rho)^{j+1}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\ &+ \frac{1}{N}\sum_{j=0}^{\infty}\rho^{j+1}\mathrm{tr}\left\{(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\} \\ &+ \frac{1}{N}\sum_{j=0}^{\infty}(1-\rho)^j\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\ &+ \frac{1}{N}\sum_{j=0}^{\infty}\rho^j\mathrm{tr}\left\{(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\}\end{aligned} \tag{30}$$

The same argument that was used to obtain the network MSD, η_L, then leads to

$$\begin{aligned}\acute{\eta}_L &= \frac{1}{N}\sum_{j=0}^{\infty}(1-\rho)^{j+1}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\ &+ \frac{1}{N}\sum_{j=0}^{\infty}\rho^{j+1}\mathrm{tr}\left\{(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\}\end{aligned} \tag{31}$$

Obviously, it holds that

$$\begin{aligned}\mathrm{tr}\left\{(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\} &< \mathrm{tr}\left\{(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\} \\ &+ \mathrm{tr}\left\{(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\}\end{aligned} \tag{32}$$

Afterward, we obtain

$$\begin{aligned}\mathrm{tr}\left\{(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\} &= \mathrm{tr}\left\{(\mathbf{C}^T)^{j+1}\mathbf{C}^{j+1}\right\} \\ &\times\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\}\end{aligned} \tag{33}$$

where $\mathrm{tr}\left\{(\mathbf{C}^T)^{j+1}\mathbf{C}^{j+1}\right\} = \sum_{k=1}^{N}\|c_{k,j+1}\|^2 \quad \forall j\geq 0$ and $c_{k,j+1}$ is the kth row of $\mathbf{C}^{j+1}$. Since $\mathbf{C}$ is doubly-stochastic, $\mathbf{C}^{j+1}$, $j\geq 0$, is also doubly-stochastic. In addition, in a connected network we have

$$\|c_{k,j+1}\|^2 \leq 1, \quad j\geq 0, \quad \forall k$$

This implies that the following inequalities hold for $M =$

$\{1, \ldots, L-1\}, (0 < \rho < 1)$:

$$\begin{aligned}
\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\boldsymbol{\Omega}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\} &< (1-\rho)^{j+1}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\
&\quad +\rho^{j+1}\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\boldsymbol{\Omega}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}})^j\right\} \\
&< (1-\rho)^{j+1}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\
&\quad +\rho^{j+1}\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\boldsymbol{\Omega}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\} \\
&\quad +(1-\rho)^{j}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\
&\quad +\rho^{j}\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\mathbf{Z}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\} \\
&< \mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\boldsymbol{\Omega}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\} \\
&\quad +\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\mathbf{Z}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\}
\end{aligned} \tag{34}$$

Moreover, for the cases of full-diffusion under ideal and noisy links $(L = M, \rho = 1)$ algorithms, we have

$$\eta'_M = \frac{1}{N}\sum_{j=0}^{\infty}\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\boldsymbol{\Omega}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\} \tag{35}$$

$$\begin{aligned}
\eta_M = \frac{1}{N}\sum_{j=0}^{\infty}&\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\boldsymbol{\Omega}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\} \\
&+\frac{1}{N}\sum_{j=0}^{\infty}\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\}
\end{aligned} \tag{36}$$

respectively. Summing (34) for all $j \geq 0$ alongside considering (30), (31), (35), and (36), we arrive at (28), which completes the proof. ■

It is noteworthy to mention that the performance degradation incurred by noisy links depends not only on the link noise variances, $\sigma^2_{w,lk}$, but also on the other parameters of the network, i.e., the measurement and state noise variances, the network topology, and combination weights.

B. Comparison with Non-cooperative Algorithm

Here, we compare the performance of PDKF algorithm (30) with that of non-cooperative processing (37), $(L = 0, \rho = 0)$, where the nodes act independently.

$$\eta_0 = \frac{1}{N}\sum_{j=0}^{\infty}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} \tag{37}$$

We establish a procedure to determine the minimum permissible link noise variances in which the communication/performance trade-off collapses. This comparison is meaningful only when the communications among agents are restricted to noisy exchange. In other words, because of imperfect information exchange, the PDKF algorithm does not always fall behind the non-cooperative operation modes. The following theorem states the desired claim.

Proposition 5. *Let Assumptions 1-4 hold. In order to guarantee*

$$\eta_0 < \eta_L, \quad \forall \quad L \geq 1 \tag{38}$$

The noise variances across communication links, $\sigma^2_{w,lk}$, should be lower bounded;

$$\begin{aligned}
\sigma^2_{w,lk} > &\frac{\left[1-(1-\rho)^{j+1}-\rho^{j+1}\sum_{k=1}^{N}\left\|c_{k,j+1}\right\|^2\right]}{\left[(1-\rho)^{j}+\rho^{j}\sum_{k=1}^{N}\left\|c_{k,j}\right\|^2\right]} \\
&\times\frac{\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\}}{\left\|\boldsymbol{\mathcal{F}}^j\right\|_f^2}
\end{aligned} \tag{39}$$

where $\left\|\boldsymbol{\mathcal{F}}^j\right\|_f$ is the Frobenius norm of matrix $\boldsymbol{\mathcal{F}}^j$.

Proof: The inequality (38) implies that

$$\begin{aligned}
\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} &< (1-\rho)^{j+1}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\
&\quad +\rho^{j+1}\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\boldsymbol{\Omega}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\} \\
&\quad +(1-\rho)^{j}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\
&\quad +\rho^{j}\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\}
\end{aligned} \tag{40}$$

Subsequently, we obtain

$$\begin{aligned}
\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\boldsymbol{\mathfrak{C}}\boldsymbol{\Omega}\boldsymbol{\mathfrak{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\} &= \mathrm{tr}\left\{(\mathbf{C}^T)^{j+1}\mathbf{C}^{j+1}\right\} \\
&\quad \times\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\}
\end{aligned} \tag{41}$$

and

$$\begin{aligned}
\mathrm{tr}\left\{(\boldsymbol{\mathfrak{C}}\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathfrak{C}}^T)^j\right\} &= \mathrm{tr}\left\{(\mathbf{C}^T)^{j}\mathbf{C}^{j}\right\} \\
&\quad \times\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T)^j\right\}
\end{aligned} \tag{42}$$

where $\mathrm{tr}\left\{(\mathbf{C}^T)^{j+1}\mathbf{C}^{j+1}\right\} = \sum_{k=1}^{N}\left\|c_{k,j+1}\right\|^2 \quad \forall j \geq 0$. Since $\mathbf{C}$ is doubly stochastic, $\mathbf{C}^{j+1}$, $j \geq 0$, is also doubly stochastic. Moreover, in a connected network it holds that

$$\left\|c_{k,j+1}\right\|^2 \leq 1, \quad j \geq 0, \quad \forall k$$

Using (41) and (42), we have

$$\begin{aligned}
\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} &< (1-\rho)^{j+1}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\
&\quad +\rho^{j+1}\sum_{k=1}^{N}\left\|c_{k,j+1}\right\|^2\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\
&\quad +(1-\rho)^{j}\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T)^j\right\} \\
&\quad +\rho^{j}\sum_{k=1}^{N}\left\|c_{k,j}\right\|^2\mathrm{tr}\left\{(\boldsymbol{\mathcal{F}})^j\mathbf{Z}(\boldsymbol{\mathcal{F}}^T)^j\right\}
\end{aligned} \tag{43}$$

Under Assumption 2-i, we have

$$\left[1-(1-\rho)^{j+1}-\rho^{j+1}\sum_{k=1}^{N}\|c_{k,j+1}\|^2\right]\operatorname{tr}\left\{(\boldsymbol{\mathcal{F}})^j\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^j\right\} < \sigma^2_{w,lk}\left[(1-\rho)^j+\rho^j\sum_{k=1}^{N}\|c_{k,j}\|^2\right]\operatorname{tr}\left\{(\boldsymbol{\mathcal{F}})^j(\boldsymbol{\mathcal{F}}^T)^j\right\} \tag{44}$$

It is very easy to prove the remaining parts of this proposition. Accordingly, it follows from (44) that (39) holds. ■

Remark 4. *Proposition 5 demonstrates that the PDKF algorithm with noisy links cannot always outperform the non-cooperative DKF algorithm unless the variances of link noise are bounded as in* (39)

VI. Adaptive Combination Weights

In this section, we firstly formulate the selection of optimal combination weights and obtained a closed-form solution for it. Then, an adaptive scheme is developed to obtain the optimal weights which runs in real-time along with the learning process. To begin with, we formulate the selection of optimal combination weights $\{c_{lk}\}$ as follows

$$\min_{\mathbf{C}} \quad \text{MSD}$$
$$\text{subject to} \quad \mathbb{1}^T\mathbf{C} = \mathbb{1}^T, \quad c_{lk}=0 \text{ if } l\notin\mathcal{N}_k,\ \forall l,k \tag{45}$$

Minimizing the MSD expression (25) for the PDKF algorithm over left-stochastic $\mathbf{C}$ is generally non-trivial. Here, however, a different approximate technique is proposed which leads to one adaptive solution that acts equally well in practice. It inspired by [16] and depends on optimizing an upper bound to the MSD.

A. An Upper Bound on MSD

Let $\|\Theta\|_{\dagger}$ denote the trace norm of matrix Θ [44], defined as the sum of singular values of Θ. When Θ is Hermitian and non-negative definite ($\Theta \geq 0$), we have

$$\|\Theta\|_{\dagger} = \operatorname{tr}\{\Theta\}$$

From (28), we conclude that η_L, (30), is bounded from above by η_M, (36). Therefore, we need to upper bound η_M instead of η_L. If $\|\Theta\|_{b,\infty}$ denotes the block maximum norm of matrix Θ, then we have

$$\begin{aligned}
&\operatorname{tr}\left\{(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\left(\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right)(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\}\\
&\quad=\left\|(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\left(\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right)(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\|_{\dagger}\\
&\quad\leq\left\|(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\right\|_{\dagger}.\left\|\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\|_{\dagger}.\left\|(\boldsymbol{\mathcal{F}}^T\mathfrak{C}^T)^j\right\|_{\dagger}\\
&\quad\leq\kappa^2.\left\|(\mathfrak{C}\boldsymbol{\mathcal{F}})^j\right\|^2_{b,\infty}.\operatorname{tr}\left\{\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\}\\
&\quad\leq\kappa^2.\left\|(\mathfrak{C}\boldsymbol{\mathcal{F}})\right\|^{2j}_{b,\infty}.\operatorname{tr}\left\{\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\}\\
&\quad\overset{(a)}{\leq}\kappa^2.\left(\|(\mathfrak{C})\|_{b,\infty}\|(\boldsymbol{\mathcal{F}})\|_{b,\infty}\right)^{2j}\times\operatorname{tr}\left\{\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\}\\
&\quad=\kappa^2.\rho\left(\boldsymbol{\mathcal{F}}\right)^{2j}.\operatorname{tr}\left\{\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\}
\end{aligned}\tag{46}$$

where κ is some finite positive constant such that $\|\Theta\|_{\dagger}\leq\kappa\|\Theta\|_{b,\infty}$ [45]. To justify step (a), we use results of Appendix A at [16]. Thus, the network MSD can be upper bounded as follows

$$\begin{aligned}
\eta_M&\leq\frac{1}{N}\sum_{j=0}^{\infty}\kappa^2.\rho\left(\boldsymbol{\mathcal{F}}\right)^{2j}.\operatorname{tr}\left\{\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\}\\
&=\frac{\kappa^2}{N.\left(1-\left[\rho\left(\boldsymbol{\mathcal{F}}\right)\right]^2\right)}.\operatorname{tr}\left\{\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\}
\end{aligned}\tag{47}$$

Expression (47) illustrates that the norm of series (36) is bounded by a scaled multiple of $\operatorname{tr}\left\{\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\}$ and the scaling constant is independent of the combination matrix $\mathfrak{C}$.

B. Minimizing the Upper Bound

We therefore replace (45) with the simpler optimization problem:

$$\min_{\mathbf{C}} \ \operatorname{tr}\left\{\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\}$$
$$\text{subject to } \mathbb{1}^T\mathbf{C}=\mathbb{1}^T, \quad c_{lk}=0 \text{ if } l\notin\mathcal{N}_k,\ \forall l,k \tag{48}$$

Let $\mathbf{Q}=\sigma^2_Q\mathbf{I}_M$ and $\mathbf{R}_k=\sigma^2_{v,k}\mathbf{I}_P$. Because matrix $\mathbf{C}$ is assumed left-stochastic, the cost function in (48) can be written in terms of the combination coefficients as follows

$$\begin{aligned}
&\operatorname{tr}\left\{\mathfrak{C}\boldsymbol{\Omega}\mathfrak{C}^T+\mathbf{Z}\right\}\\
&\quad=\sum_{k=1}^{N}\sum_{l\in\mathcal{N}_k}c^2_{lk}[\sigma^2_Q\operatorname{tr}\left\{\mathbf{G}_l\mathbf{G}_l^T\right\}+\sigma^2_{v,l}\operatorname{tr}\left\{\mathbf{P}_l\mathbf{S}_l\mathbf{S}_l^T\mathbf{P}_l^T\right\}\\
&\qquad+\operatorname{tr}\left\{\mathbf{Z}_{lk}\right\}]
\end{aligned}\tag{49}$$

where $\mathbf{G}_l=(\mathbf{I}_M-\mathbf{P}_l\mathbf{S}_l)\,\mathbf{G}$. We can therefore decouple the problem (48) into N separate optimization problems as follows

$$\begin{aligned}
&\min_{\{c_{lk}\}_{l=1}^N}\sum_{l\in\mathcal{N}_k}c^2_{lk}[\sigma^2_Q\operatorname{tr}\left\{\mathbf{G}_l\mathbf{G}_l^T\right\}+\sigma^2_{v,l}\operatorname{tr}\left\{\mathbf{P}_l\mathbf{S}_l\mathbf{S}_l^T\mathbf{P}_l^T\right\}\\
&\qquad+\operatorname{tr}\left\{\mathbf{Z}_{lk}\right\}],\ k=1,\ldots,N\\
&\text{subject to}\\
&\sum_{l\in\mathcal{N}_k}c_{lk}=1,\ c_{lk}\geq0,\ c_{lk}=0 \text{ if } l\notin\mathcal{N}_k,\ \forall l,k
\end{aligned}\tag{50}$$

Introducing the following as non-negative variance product measure

$$\begin{aligned}
\mu^2_{lk}\triangleq\sigma^2_Q\operatorname{tr}\left\{\mathbf{G}_l\mathbf{G}_l^T\right\}+\sigma^2_{v,l}\operatorname{tr}\left\{\mathbf{P}_l\mathbf{S}_l\mathbf{S}_l^T\mathbf{P}_l^T\right\}\\
+\operatorname{tr}\left\{\mathbf{Z}_{lk}\right\},\ k\in\mathcal{N}_l
\end{aligned}\tag{51}$$

the solution of problem (50) is given by

$$c_{lk}=\begin{cases}\frac{\mu^{-2}_{lk}}{\sum_{m\in\mathcal{N}_k}\mu^{-2}_{mk}} & \text{if}\, l\in\mathcal{N}_k\\ 0 & \text{otherwise}\end{cases}\tag{52}$$

Referring to this combination rule as the *relative-variance combination rule* [46], it results in a left-stochastic matrix $\mathbf{C}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAES.2019.2933961, IEEE Transactions on Aerospace and Electronic Systems

C. Adaptive Solution

To evaluate the relative variance combination rule (52), the nodes need to know the variance products, $\left\{\mu_{lk}^2\right\}$, of their neighbors, which are often not available beforehand. Therefore, an adaptive combination rule is desirable, where individual nodes learn their combination coefficients (52) using the available data. We Refer to the Kalman recursion (5) and (7) and use the output equation (1) to write for node l

$$\boldsymbol{\psi}_{lk,i} = \hat{\mathbf{x}}_{l,i|i-1} + \mathbf{P}_{l,i}\mathbf{H}_{l,i}^T\mathbf{R}_{e,i}^{-1}[\mathbf{H}_{l,i}\tilde{\mathbf{x}}_{l,i|i-1} + \mathbf{v}_{l,i}] + \mathbf{w}_{lk,i} \tag{53}$$

so that we obtain in limit as $i \to \infty$

$$\mathbb{E}\left[\left\|\boldsymbol{\psi}_{lk,i} - \hat{\mathbf{x}}_{l,i|i-1}\right\|^2\right] \approx \mu_{lk}^2, \quad \forall l \in \mathcal{N}_k \tag{54}$$

As the algorithm approaches steady-state, all the estimates $\left\{\hat{\mathbf{x}}_{k,i|i}\right\}$ tend close to $\mathbf{x}_i$. Using this fact, it is straightforward to estimate μ_{lk}^2 for node k by using instantaneous realizations of $\left\|\boldsymbol{\psi}_{lk,i} - \hat{\mathbf{x}}_{k,i|i-1}\right\|^2$, where $\hat{\mathbf{x}}_{l,i|i-1}$ is replaced by $\hat{\mathbf{x}}_{k,i|i-1}$. Thus, let $\hat{\boldsymbol{\mu}}_{lk}^2$ denote an estimator for μ_{lk}^2 computed by node k at time i. Then, one way to evaluate $\hat{\boldsymbol{\mu}}_{lk}^2$ is through the recursion:

$$\hat{\boldsymbol{\mu}}_{lk}^2(i) = (1-\tau_k)\,\hat{\boldsymbol{\mu}}_{lk}^2(i-1) + \tau_k\left\|\boldsymbol{\psi}_{lk,i} - \hat{\mathbf{x}}_{k,i|i-1}\right\|^2 \tag{55}$$

where $\tau_k \in (0,1)$ is a positive coefficient smaller than one. In this way, we can replace the weights (52) by those adaptively computed as

$$\mathbf{c}_{lk}(i) = \begin{cases} \dfrac{\hat{\boldsymbol{\mu}}_{lk}^{-2}(i)}{\sum_{m\in\mathcal{N}_k}\hat{\boldsymbol{\mu}}_{mk}^{-2}(i)} & \text{if } l \in \mathcal{N}_k \\ 0 & \text{otherwise} \end{cases} \tag{56}$$

VII. Simulations

Consider an adaptive network, randomly generated and has a total of $N = 10$ with topology shown in Fig. 2. Each node is, on average, connected to two other nodes. The size of the unknown parameter of the system is $M = 4$. We apply the PDKF algorithm to the problem of estimating and tracking the position of a projectile. This could be useful in places where a certain projectile is in the proximity of an adaptive network and a set of nodes are sensing the position of the projectile. They are subject to noisy links. The state of the system is an unknown 2-dimensional location vector of an object, i.e. (x, y), where x and y are first and second entries, respectively. In our simulation example, the position, the velocity and the acceleration of projectile, respectively, are

$$\boldsymbol{d} = \begin{bmatrix} d_x \\ d_y \end{bmatrix}, \boldsymbol{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \boldsymbol{a} = \begin{bmatrix} a_x \\ a_y \end{bmatrix},$$

For a projectile motion we have

$$\mathbf{v} = \dot{\boldsymbol{d}},\ \boldsymbol{a} = \dot{\boldsymbol{v}},\ a_x = 0,\ a_y = -g$$

where $g = 10$ denotes the gravity constant. The state $\mathbf{x}$ of the system is formed by stacking the position and velocity of projectile. Therefore, the state equation is described as follows:

$$\begin{bmatrix} \dot{\boldsymbol{d}} \\ \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I}_2 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{d} \\ \boldsymbol{v} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \begin{bmatrix} 0 \\ -g \end{bmatrix} \end{bmatrix}$$

This can be rewritten in a compact form of $\dot{\mathbf{x}} = \boldsymbol{\theta}\mathbf{x} + \boldsymbol{\nu}$. Note that for the $\boldsymbol{\theta}$ matrix

$$e^{\boldsymbol{\theta}\delta} = \mathbf{I} + \delta\boldsymbol{\theta} \quad \text{and} \quad \int_{t_0}^{t_0+\delta} e^{\boldsymbol{\theta}(t_0+\delta-\tau)}d\tau$$

Therefore, the state satisfies the following equation

$$\mathbf{x}(t+\delta) = [\mathbf{I} + \delta\boldsymbol{\theta}]\,\mathbf{x}(t) + \left[\delta\mathbf{I} - \delta^2\boldsymbol{\theta}/2\right]\boldsymbol{\nu}$$

For a given time-step δ, and $\mathbf{u}$ can be denoted as follows

$$\mathbf{F} \triangleq \mathbf{I} + \delta\boldsymbol{\theta} \quad \text{and} \quad \mathbf{u} \triangleq \left[\delta\mathbf{I} - \delta^2\boldsymbol{\theta}/2\right]\boldsymbol{\nu}$$

We presume that each node measures the position of the unknown object in the two, i.e. x and y, dimensions. So, we have $\mathbf{H}_{k,i}$ as the following matrix

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

Let us denote $\mathbf{x}_i = \mathbf{x}(i\delta)$ and take into account the effect of state and measurement noises, which results in the following discrete state-space model:

$$\begin{aligned} \mathbf{x}_{i+1} &= \mathbf{F}\mathbf{x}_i + \mathbf{G}_i\mathbf{n}_i + \mathbf{u} \\ \mathbf{y}_{k,i} &= \mathbf{H}_{k,i}\mathbf{x}_i + \mathbf{v}_{k,i} \end{aligned}$$

where $\mathbf{y}_{k,i}$ are the individual measurements obtained by node k at time i, $\mathbf{n}_i$ accounts for modeling errors, and $\mathbf{v}_{k,i}$ is the measurement noise at node k. It is noteworthy to mention that this model has the same form as the biased model presented in [9] and can therefore be reduced to a model similar to (1). The state-space model matrices in (1) are:

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

The measurement noise covariance matrix at agent k is, $\mathbf{R}_{k,i} = \sigma_{k,i}^2\mathbf{I}_2$, where the noise variance $\sigma_{k,i}^2$ across the agents is selected randomly in the range [0 0.5]. The experimental results are obtained by taking the ensemble-average over 200 independent trials and the steady-state values are calculated by averaging over 1000 steady-state iterations.

We also use the uniform combination rule [17] at the aggregation step and the estimate vectors are initialized to zero. The state noise covariance matrix traces, observation noise variances at all the nodes and the variance profile for the link noises are generated randomly and shown in Fig. 3. We use white Gaussian link noise signal such that $\mathbf{Z}_{lk} = \sigma_{w,lk}^2\mathbf{I}_M$. To assign the link number we employ the same approach used in [16].

The learning curves of PDKF algorithm with noisy links (in terms of network MSD) for different values of L are shown in Fig. 4. The MSD curves for both coordinated and uncoordinated partial diffusion schemes are considered in Fig. 4. Experimental and theoretical steady-state MSDs of all nodes for different numbers of entries propagated at each time instant i, are illustrated in Fig. 5. The steady-state network MSD as a function of communicated entries at each iteration for different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAES.2019.2933961, IEEE Transactions on Aerospace and Electronic Systems

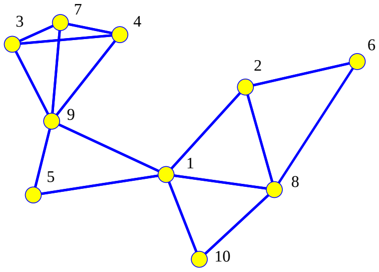


Fig. 2. Network Graph with $N = 10$ nodes.

values of $\sigma^2_{w,lk}$ using both coordinated and uncoordinated schemes are demonstrated in Fig. 6. We examine the PDKF algorithm over noisy information exchange under two different combination rules, the uniform weighting and the adaptation rule in (56) with $\{\tau_k = 0.05\}$. From Fig. 7, we see that the adaptive rule attains lower MSD level at steady-state, compared to the uniform rules. The global MSD, in terms of $\sigma^2_{w,lk}$ for different values of L are shown in Fig. 8. We can see that as $\sigma^2_{w,lk}$ increases, the non-cooperative solution performs better than the cooperative solution.

From the results above, we summarize the main conclusions as follows:

- As expected, the presence of noisy links deteriorates the performance of PDKF algorithm.
- Unlike ideal communication case, there is not a clear trade-off between the communication load and estimation performance. Increasing the number of communicated entries at each iteration does not result in any better steady-state network MSD performance.
- The convergence of PDKF algorithm in a network with noisy links is possible provided that the required conditions for mean and mean-square are satisfied.
- One can see from Figs. 4-6 that the theoretical results agree well with the simulation ones.
- Simulation results, Fig. 7 illustrate that the PDKF algorithm with proposed adaptive combiners outperforms those with existing static combiners.

VIII. Conclusions

In the PDKF algorithm, every node is permitted to share only a subset of its intermediate estimate vectors at each iteration among its neighbors, which reduces the amount of inter-node communications. In this paper, the performance of PDKF algorithm in a more practical scenario is investigated, where the exchange of weight estimates among the neighboring nodes is subject to noise. To show the deterioration of network performance, we considered the MSD as steady-state performance metric and derived a theoretical expression for it. We further analyzed the convergence behavior of PDKF algorithm in both mean and mean-square senses. We illustrated that unlike the established statements on PDKF scheme under ideal links, the trade-off between MSD performance and the number of selected entries of the intermediate estimate vectors as a sign of communication cost is mitigated. To reduce the effect of channel noise in the aggregation phase, we consider the problem of optimal selection of combination weights. We presented some numerical examples to verify our theoretical findings. More importantly, the simulation findings reveal that the proposed combination coefficients can greatly improve the performance of diffusion adaptation with noisy communication links.

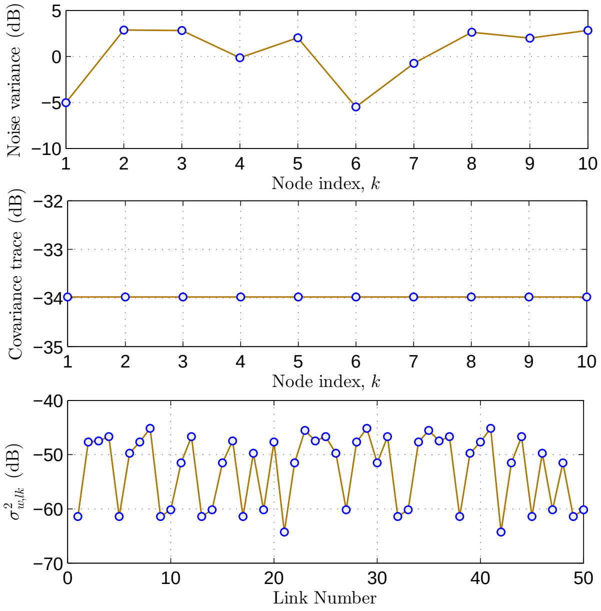


Fig. 3. Observation noise variances, the state noise covariance matrix traces at all nodes and the variance profile for link noises.

Appendix A

By applying the matrix inversion lemma, we can express $\mathbf{P}_{k,i}\mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{k,i} = \mathbf{P}_{k,i|i-1}\mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{e,i}$. This equality can be approached as follows

$$\begin{aligned}\mathbf{P}_{k,i|i-1}\mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{e,i} &= \mathbf{P}_{k,i|i-1}\mathbf{H}^T_{k,i}\left(\mathbf{R}_{k,i} + \mathbf{H}_{k,i}\mathbf{P}_{k,i}\mathbf{H}^T_{k,i}\right)^{-1}\\ &= \mathbf{P}_{k,i|i-1}\mathbf{H}^T_{k,i}\Big[\mathbf{R}^{-1}_{k,i}\\ &\qquad - \mathbf{R}^{-1}_{k,i}\mathbf{H}_{k,i}\big(\mathbf{P}^{-1}_{k,i|i-1} + \mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{k,i}\mathbf{H}_{k,i}\big)^{-1}\mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{k,i}\Big]\end{aligned}$$

Applying the matrix inversion lemma to the therm $\left(\mathbf{P}^{-1}_{k,i|i-1} + \mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{k,i}\mathbf{H}_{k,i}\right)^{-1}$ leads to

$$\begin{aligned}\mathbf{P}_{k,i|i-1}\mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{e,i} &= \mathbf{P}_{k,i|i-1}\left[\mathbf{I}_M - \mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{e,i}\mathbf{H}_{k,i}\mathbf{P}_{k,i|i-1}\right]\mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{k,i}\\ &= \left[\mathbf{P}_{k,i|i-1} - \mathbf{P}_{k,i|i-1}\mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{e,i}\mathbf{H}_{k,i}\mathbf{P}_{k,i|i-1}\right]\mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{k,i}\\ &= \mathbf{P}_{k,i}\mathbf{H}^T_{k,i}\mathbf{R}^{-1}_{k,i}\end{aligned}$$

References

[1] A. H. Sayed, “Adaptation, learning, and optimization over networks,” *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

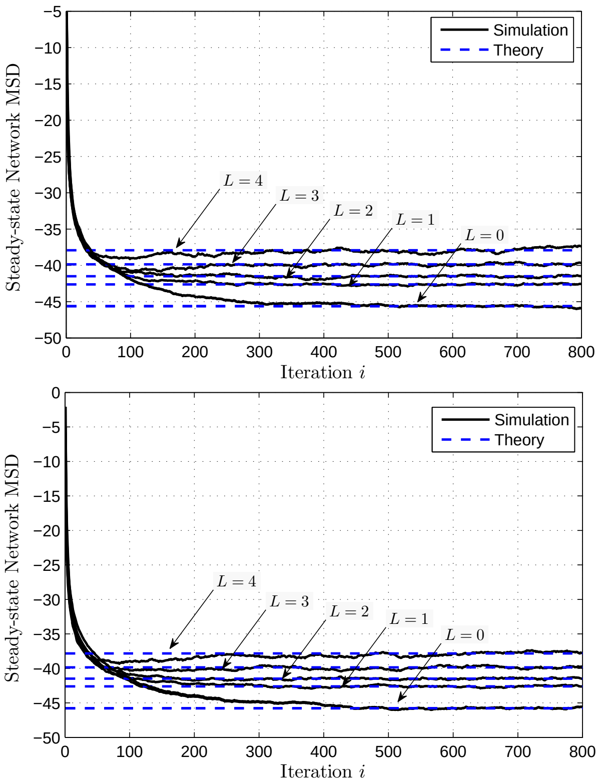


Fig. 4. Network MSD curves of the PDKF algorithm using coordinated (top) and uncoordinated (bottom) partial diffusion schemes for different numbers of entries communicated at each iteration when links assumed to be noisy.

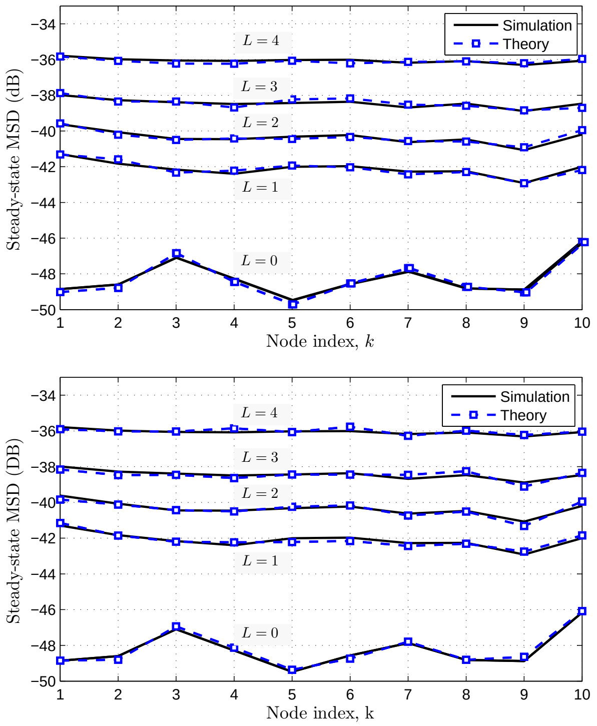


Fig. 5. Theoretical and experimental steady-state MSDs at each node for different values of L using coordinated (top) and uncoordinated (bottom) partial diffusion scheme in the PDKF algorithm, when links assumed to be noisy. Note that in all the simulations, the differences between the theoretical and simulation MSDs are less than 0.2 dB.

[2] S. Monajemi, K. Eftaxias, S. Sanei, and S. H. Ong, "An informed multitask diffusion adaptation approach to study tremor in parkinson's disease," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 7, pp. 1306–1314, Oct 2016.

[3] M. Latifi, A. Khalili, A. Rastegarnia, and S. Sanei, "Fully distributed demand response using the adaptive diffusion-stackelberg algorithm," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2291–2301, Oct 2017.

[4] W. Li and Y. Jia, "Distributed estimation for markov jump systems via diffusion strategies," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 448–460, Feb 2017.

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

[6] F. S. Cattivelli and A. H. Sayed, "Diffusion mechanisms for fixed-point distributed kalman smoothing," in *2008 16th European Signal Processing Conference*, Aug 2008, pp. 1–5.

[7] F. Cattivelli and A. H. Sayed, "Diffusion distributed kalman filtering with adaptive weights," in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, Nov 2009, pp. 908–912.

[8] F. S. Cattivelli and A. H. Sayed, "Distributed nonlinear kalman filtering with applications to wireless localization," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 3522–3525.

[9] ——, "Diffusion strategies for distributed Kalman filtering and smoothing," *Automatic Control, IEEE Transactions on*, vol. 55, no. 9, pp. 2069–2084, 2010.

[10] J. Hu, L. Xie, and C. Zhang, "Diffusion kalman filtering based on covariance intersection," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 891–902, Feb 2012.

[11] S. Yang, T. Huang, J. Guan, Y. Xiong, and M. Wang, "Diffusion strategies for distributed kalman filter with dynamic topologies in virtualized sensor networks," *Mobile Information Systems*, vol. 2016, pp. 8 695 102:1–8 695 102:13, 2016.

[12] S. P. Talebi and S. Werner, "Distributed kalman filtering and control through embedded average consensus information fusion," *IEEE Transactions on Automatic Control*, 2019.

[13] ——, "Distributed kalman filtering in presence of unknown outer network actuations," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 186–191, 2019.

[14] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. A. Chambers, "Transient analysis of diffusion leastmean squares adaptive networks with noisy channels," *International Journal of Adaptive Control and Signal Processing*, vol. 26, no. 2, pp. 171–180, 2012.

[15] A. Khalili, M. A. Tinati, A. Rastegarnia, and J. Chambers, "Steady-state analysis of diffusion LMS adaptive networks with noisy links," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 974–979, 2012.

[16] X. Zhao, S.-Y. Tu, and A. H. Sayed, "Diffusion adaptation over networks under imperfect information exchange and non-stationary data," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, 2012.

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

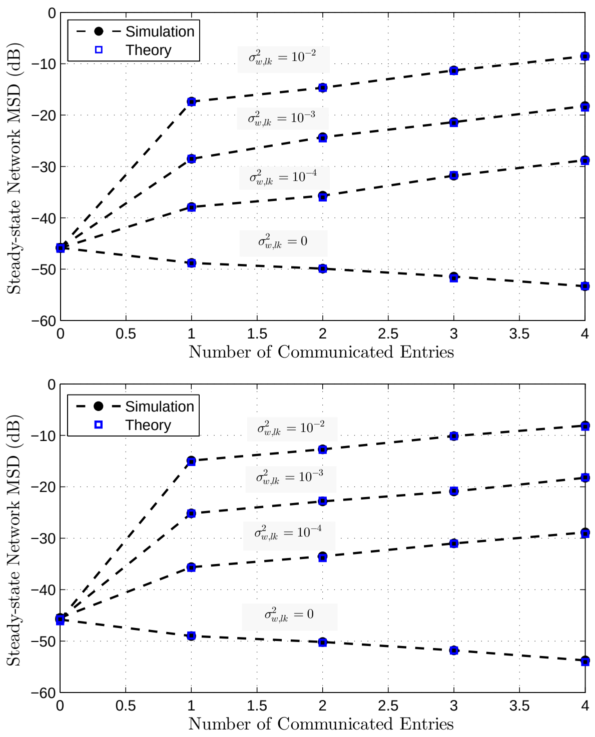


Fig. 6. Theoretical and experimental steady-state MSDs as a function of number of communicated entries for different values of $\sigma^2_{w,lk}$ using coordinate (top) and uncoordinated (bottom) partial diffusion schemes in PDKF algorithm.

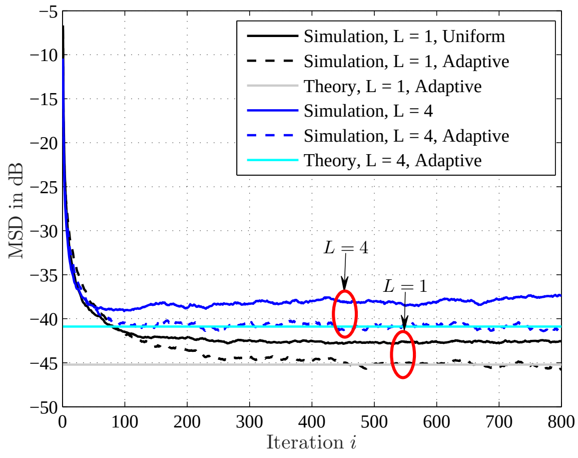


Fig. 7. Simulated network MSD curves with two different combination rules under noisy information exchange.

[18] R. Arablouei, S. Werner, K. Dougançay, and Y.-F. Huang, "Analysis of a reduced-communication diffusion LMS algorithm," *Signal Processing*, vol. 117, pp. 355–361, 2015.

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

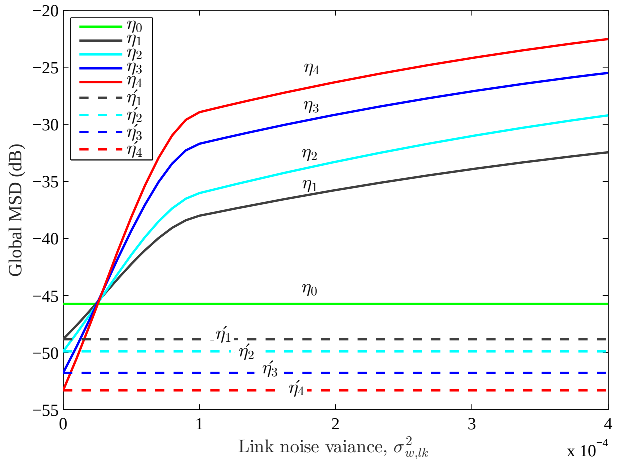


Fig. 8. The global MSD (dB) versus $\sigma^2_{w,lk}$. Note that the value of the lower bound from (39) is 2.5×10^{-5}.

[20] ——, "Compressive diffusion strategies over distributed networks for reduced communication load," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5308–5323, 2014.

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

[22] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–484, 2014.

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

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

[25] 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 Transactions on Network Science and Engineering*, vol. 5, no. 2, pp. 101–112, 2018.

[26] V. Vahidpour, A. Rastegarnia, A. Khalili, and S. Sanei, "Partial diffusion kalman filtering for distributed state estimation in multiagent networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[27] G. Wang, N. Li, and Y. Zhang, "Diffusion distributed kalman filter over sensor networks without exchanging raw measurements," *Signal Process.*, vol. 132, no. C, pp. 1–7, Mar. 2017.

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

[29] 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 Processing Letters*, vol. 5, no. 5, pp. 111–114, 1998.

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

[31] K. Dogancay, *Partial-update adaptive signal processing: Design Analysis and Implementation*. Academic Press, 2008.

[32] S. Werner, T. Riihonen, and Y.-F. Huang, "Energy-efficient distributed parameter estimation with partial updates," in *Green Circuits and*

Systems (ICGCS), 2010 International Conference on. IEEE, 2010, pp. 36–40.

[33] A. Malipatil, Y.-F. Huang, and S. Werner, "An SMF approach to distributed average consensus in clustered sensor networks," in *Signal Processing Advances in Wireless Communications, 2009. SPAWC'09. IEEE 10th Workshop on*. IEEE, 2009, pp. 81–85.

[34] S. Werner, Y.-F. Huang, M. L. R. De Campos, and V. Koivunen, "Distributed parameter estimation with selective cooperation," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 2849–2852.

[35] S. Werner, M. Mohammed, Y.-F. Huang, and V. Koivunen, "Decentralized set-membership adaptive estimation for clustered sensor networks," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 3573–3576.

[36] S. Werner and Y.-F. Huang, "Time-and coefficient-selective diffusion strategies for distributed parameter estimation," in *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*. IEEE, 2010, pp. 696–697.

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

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

[39] T. M. Berg and H. F. Durrant-Whyte, "Model distribution in decentralized multi-sensor data fusion," in *American Control Conference, 1991*. IEEE, 1991, pp. 2292–2293.

[40] A. G. O. Mutambara, *Decentralized estimation and control for multisensor systems*. CRC press, 1998.

[41] M. Raitoharju, R. Piché, J. Ala-Luhtala, and S. Ali-Löytty, "Partitioned update Kalman filter," *arXiv preprint arXiv:1503.02857*, 2015.

[42] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 2.

[43] K. M. Abadir and J. R. Magnus, *Matrix algebra*. Cambridge University Press, 2005, vol. 1.

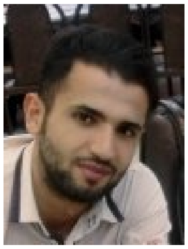
[44] A. J. Laub, *Matrix analysis for scientists and engineers*. Siam, 2005, vol. 91.

[45] R. A. Horn, R. A. Horn, and C. R. Johnson, *Matrix analysis*. Cambridge university press, 1990.

[46] S.-Y. Tu and A. H. Sayed, "Optimal combination rules for adaptation and learning over networks," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2011 4th IEEE International Workshop on*. IEEE, 2011, pp. 317–320.

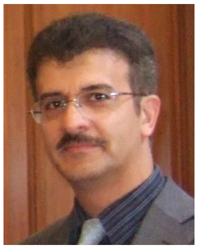
Vahid Vahidpour received the M.Sc degree in communication engineering from Malayer University, Hamedan, Iran, in 2016. His research interests include underwater acoustic, distributed and adaptive signal processing, and biomedical signal processing. Mr. Vahidpour is a student member of the IEEE.

Amir Rastegarnia completed his PhD degree in the electrical engineering at the University of Tabriz, Tabriz, Iran, in 2011. In 2011, he joined the Department of Electrical Engineering, Malayer University, as Assistant Professor. His current research interests are theory and methods for adaptive and statistical signal processing, distributed adaptive estimation, as well as signal processing for communications. He is a Member of IEEE.



Milad latifi received his M.Sc degree in communication engineering from Malayer University, Hamedan, Iran, in 2017. His research interests include advanced signal processing, adaptive filtering, cooperative learning, multi-agent networking, and distributed optimization with the application in smart power grids. Mr. Latifi is a student member of the IEEE.

Azam Khalili received the PhD degree in electrical engineering from the University of Tabriz, Tabriz, Iran, in 2011. In 2011, she joined the Department of Electrical Engineering, Malayer University, as Assistant Professor. Her current research interests are theory and methods for adaptive filtering, distributed adaptive estimation, as well as signal processing for communications. She is a Member of IEEE.



Saeid Sanei (SM05) received his PhD in signal processing from Imperial College London, UK. He has been a member of academic staff in Iran, Singapore, and the UK. He has published three monograms, a number of book chapters, and over 320 papers in peer reviewed journals and conference proceedings. His research interest is in adaptive filtering, cooperative learning, multi-way, multimodal, and multichannel signal processing with applications to biomedical, audio, biometrics, and communication signals and images. He has served as an Associate Editor for the IEEE Signal Processing Letters, IEEE Signal Processing Magazine, and Journal of Computational Intelligence and Neuroscience.