

Partial Diffusion Kalman Filter with Adaptive Combiners

Azam Khalili, Vahid Vahidpour, Amir Rastegarnia, Wael M. Bazzi, and Saeid Sanei, *Senior Member, IEEE*

Abstract—Adaptive estimation of optimal combination weights for partial-diffusion Kalman filtering together with its mean convergence and stability analysis is proposed here. The simulations confirm its superior performance compared with the existing combiners. Sensor networks with limited accessible power highly benefit from this design.

Index Terms—Adaptive weights, communication cost, diffusion strategy, Kalman filtering, state estimation.

I. INTRODUCTION

DISTRIBUTED Kalman filtering algorithms are efficient and powerful for solving state-space filtering and smoothing problems over networks [1]–[3]. In these algorithms, a set of agents collaborate to estimate the state of a linear dynamic system based on their own individual measurements and information exchange with their immediate neighbors. The distributed Kalman filter has been widely applied to solve a variety of problems in the area of aerospace and electronic systems including navigation, wireless localization, and target tracking [4]–[7]. Among these algorithms, diffusion Kalman filtering (DKF) algorithm can solve the state-space filtering and smoothing problems in a fully distributed and adaptive manner [1]. The advantages of diffusion-based algorithms, however, come at the cost of increased inter-node communications. This issue may affect their performance, specially in the networks with limited power and bandwidth resources. Hence, developing useful schemes that decrease the communication load, while keeping the advantages of collaborative processing is of prime importance.

To this end, in [8], the dimensionality reduction is achieved by the Krylov subspace projection techniques in the set-theoretic estimation framework. In [9], in order to avoid the unlimited bandwidth requirement, the parameter estimate is quantized before the diffusion of information. In the partial diffusion strategies [10]–[13], only a subset of the local estimates is allowed to share among the neighbors. Vahidpour et al. in [13] study the effect of channel noise during the exchange of weight estimates for partial diffusion algorithm. In [14], the nodes transmit the sign of innovation (SOI) sequence in the

distributed state estimation framework. In [15], a probabilistic diffusion protocol is used to limit the communication load in an adaptive network with dynamic topology. Takahashi *et al.* in [16] proposed an algorithm to control the link probabilities by minimizing the estimation error in order to improve the estimation performance of [15]. In [17] and [18] every agent is allowed to select a subset of its neighbors to aggregate the data based on some extra information that portrays the quality of the nodes. These approaches require extra communication to diffuse scalar values to select one node for receiving the data. In the proposed method in [19] the nodes are allowed to dynamically update their estimates and diffuse only when they are updated. In [20] and [21], the communication reduction is achieved by projecting the parameter vectors onto lower dimensional spaces before the transmission. The neighbor-selection diffusion algorithms have been developed in [22] and [23], where every node is allowed to receive the intermediate estimates of only a subset of its neighbors.

In this correspondence, our focus is on partial-diffusion Kalman filtering (PDKF) algorithm [12], in which each node diffuses a subset of entries of its intermediate state estimate vector to its neighbors. This algorithm makes a trade-off between communication cost and estimation performance. Theoretical findings and the numerical experiments reported in [12] demonstrate that, performance deterioration acquired by partial-diffusion depends not only on the probability of entry transmission for each agent but also on other parameters, such as combination coefficients. Several combination policies, such as Metropolis [24] and relative-degree [25] rules, have been previously proposed that are based solely on the network topology. Thus, the performance of such policies may degrade if, for example, the *signal-to-noise* ratio (SNR) at some agents significantly is lower than other agents [26]. Therefore, the design of combination coefficients is crucial in the diffusion networks.

The main contributions of the paper can be summarized as follows:

- The problem of optimal combination weight for PDKF is addressed;
- The problem is formulated as a constrained optimization problem and its optimum solution is found;
- An adaptive scheme which is also developed which is useful for real-time implementation;
- The mean convergence analysis is provided;
- Finally, the simulation results verify that the method with the proposed rule outperforms other existing combiners.

Manuscript received 2020. A. Khalili V. Vahidpour, and A. Rastegarnia are with the Department of Electrical Engineering, Malayer University, Malayer 65719-95863, Iran (email: a-khalili@tabrizu.ac.ir; v.vahidpour@stu.malayeru.ac.ir; rastegarnia@malayeru.ac.ir).

W. M. Bazzi is with the Electrical and Computer Engineering Department, American University in Dubai Dubai, United Arab Emirates, (email: wbazzi@aud.edu).

S. Sanei is with the School of Science and Technology, Nottingham Trent University, Clifton Lane, Nottingham, U.K. (e-mail: saeid.sanei@ntu.ac.uk)
Digital Object Identifier 2020/XX

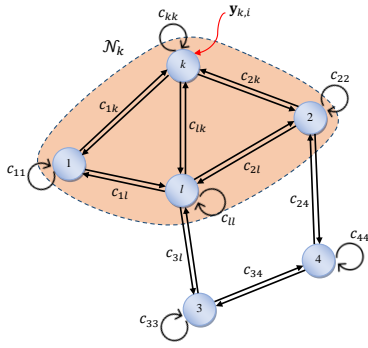


Fig. 1. A network with N nodes. The neighborhood of node k is denoted by $\mathcal{N}_k = \{1, 2, l, k\}$ and the degree of node k is $\eta_k = 4$. At every time instant i , node k collects a measurement $\mathbf{y}_{k,i}$.

The paper is organized as follows. The PDKF is briefly introduced in Section II. In Section III the optimal combination problem is formulated and an adaptive solution is derived. The performance analysis is provided in Section IV. Simulation results are presented in Section V and Section VI closes the paper and points directions for future extensions.

Notation: Small boldface letters are adopted for vectors and bold capital letters for matrices. Normal font letters denote scalars. The transposition operator is denoted by superscript $(\cdot)^T$. We also use \otimes for Kronecker product and $\text{vec}\{\cdot\}$ for a vector formed by stacking the columns of its matrix argument. We further use $\text{col}\{\cdot\}$ to denote a column vector formed by stacking its arguments on top of each other. The $\text{Blkdiag}\{\cdot\}$ denotes block-diagonal matrix and \odot the Hadamard product. The expected value of a random quantity \mathbf{x} is denoted by $\mathbb{E}[\mathbf{x}]$.

II. BACKGROUND

A. Diffusion Kalman Filter

Consider a network with P nodes as shown in Fig. 1. Each node k can only cooperate with its immediate neighbors and spread the information out through the network over a sequence of Kalman iterations and data aggregations [27]. The neighborhood of node k is denoted by \mathcal{N}_k and the degree of node k is denoted by $\eta_k = |\mathcal{N}_k|$. At time instant i , each node k collects a noisy measurement $\mathbf{y}_{k,i} \in \mathbb{R}^m$ from the state vector $\mathbf{s}_i \in \mathbb{R}^n$, where $\mathbf{y}_{k,i}$ and \mathbf{s}_i are related via a linear state-space model [1]:

$$\mathbf{s}_{i+1} = \mathbf{F}_i \mathbf{s}_i + \mathbf{G}_i \mathbf{w}_i, \quad (1a)$$

$$\mathbf{y}_{k,i} = \mathbf{H}_{k,i} \mathbf{s}_i + \mathbf{v}_{k,i} \quad (1b)$$

where \mathbf{F}_i , \mathbf{G}_i and $\mathbf{H}_{k,i}$ are known matrices with appropriate dimensions, and $\mathbf{w}_i \in \mathbb{R}^n$ and $\mathbf{v}_{k,i} \in \mathbb{R}^m$ are respectively the state and measurement noises. For the above network, the following assumptions hold:

Assumption 1.

- (i) The state and measurement noises $\{\mathbf{w}_i, \mathbf{v}_{k,i}\}$ are zero-mean white noise processes with positive-definite covariance matrices $\mathbb{E}[\mathbf{w}_i \mathbf{w}_i^T] = \mathbf{W}_i$, $\mathbb{E}[\mathbf{v}_{k,i} \mathbf{v}_{k,i}^T] = \mathbf{V}_{k,i}$.
- (ii) The state noise and measurement noise $\{\mathbf{w}_i, \mathbf{v}_{k,i}\}$ are statistically uncorrelated, i.e., $\mathbb{E}[\mathbf{v}_{k,i} \mathbf{w}_i^T] = \mathbf{0}$.

- (iii) The initial state \mathbf{s}_0 is zero-mean with positive-definite covariance matrix $\mathbb{E}[\mathbf{s}_0 \mathbf{s}_0^T] = \mathbf{S}_0$.
- (iv) The initial state vector \mathbf{s}_0 is statistically uncorrelated with both $\{\mathbf{w}_i, \mathbf{v}_{k,i}\}$ for all i and k .
- (v) The parameter matrices, $\{\mathbf{F}_i, \mathbf{G}_i, \mathbf{W}_i, \mathbf{V}_{k,i}, \mathbf{H}_{k,i}, \mathbf{S}_0\}$, are assumed to be known by each node k .

Let $\hat{\mathbf{s}}_{k,i|j}$ denote the linear minimum mean-square error (MMSE) estimate of \mathbf{s}_i that node k computes at time i using the available information and measurements up to and including time j . Through the existing DKF algorithm [1] (see Algorithm 1), the objective for each node k is to obtain a real-time linear estimate $\hat{\mathbf{s}}_{k,i|i}$ of the state vector \mathbf{s}_i . According to the state-space model, the DKF algorithm is performed in three steps: measurement-update, diffusion-update, and time-update.

First, in measurement-update, each node k receives time observation, i.e. $\{\mathbf{y}_{l,i}, \mathbf{H}_{l,i}, \mathbf{V}_{l,i}\}$, from its direct neighbors at time instant i to update its intermediate estimate $\phi_{k,i}$. Secondly, in the diffusion-update phase, the nodes communicate with each other to obtain the estimate value $\hat{\mathbf{s}}_{k,i|i}$ at time i by collecting its neighborhood estimate value at sampling time $i-1$ and combine them through a predetermined coefficient matrix $\mathbf{C} = [c_{lk}(i)]_{1 \leq l, k \leq P}$. \mathbf{C} is a column-stochastic matrix and is associated with the graph topology, where the scalars $\{c_{lk}(i)\}$ are non-negative coefficients and satisfy

$$\mathbf{C}^T \mathbf{1}_P = \mathbf{1}_P \quad (2)$$

$$\text{and } c_{lk}(i) = 0 \text{ if } l \notin \mathcal{N}_k \text{ for } k = 1, \dots, P$$

so that for each node k , the weights $\{c_{lk}(i)\}$ are add-up to unity (each column summing to one). The symbol $\mathbf{1}_P$ denotes all ones vector of size $P \times 1$. Finally, in the time-update step, the nodes update their current estimate through using the state model in (1) to predict the next state estimate value, e.g. $\hat{\mathbf{s}}_{k,i+1|i}$ at time $i+1$ can be predicted by value at time i . The covariance matrix of the estimation error $\tilde{\mathbf{s}}_{i|j} \triangleq \mathbf{s}_i - \hat{\mathbf{s}}_{k,i|j}$ is denoted by $\mathbf{\Pi}_{k,i|j}$. The DKF algorithm in its time- and measurement-update form begins with $\hat{\mathbf{s}}_{k,0|-1} = \mathbb{E}[\mathbf{s}_0]$ and $\mathbf{\Pi}_{k,0|-1} = \mathbf{S}_0$, where $\mathbf{\Pi}_{k,0|-1} \in \mathbb{R}^{n \times n}$.

B. Partial-Diffusion Kalman Filtering

The proposed PDKF algorithm in [12] is summarized in Algorithm 2. There are mainly two clear differences between DKF and PDKF algorithms. First, in the DKF algorithm, the exchange of local information can meet the data processing requirements. Thus, to reduce the communication burdens, in the PDKF algorithm, each node does not share its local information $\{\mathbf{y}_{l,i}, \mathbf{H}_{l,i}, \mathbf{R}_{l,i}\}$. Consequently, this algorithm particularly depends on the communication of chosen entries of $\phi_{k,i}$. Secondly, in the PDKF algorithm, the objective for each node k is to estimate the unknown state $\mathbf{s}_i \in \mathbb{R}^n$, recursively, while sharing a fraction of its intermediate state estimate vector with its neighbors $l \in \mathcal{N}_k$. Thus, every node k at any time instant i is allowed to select and diffuse a subset, i.e. r out of n , $0 \leq r \leq n$, entries of its intermediate state estimate vector. Ignoring local information exchange and considering such a subset can significantly reduce the communication and computation cost among the nodes. Thus, in the PDKF algorithm, the agents are more willing to have less

Algorithm 1 Diffusion Kalman filter [1]

Begin with $\hat{\mathbf{s}}_{k,0|-1} = \mathbb{E}[\mathbf{s}_0]$ and $\mathbf{\Pi}_{k,0|-1} = \mathbf{S}_0$.
Repeat at every agent k for $i \geq 0$
Step 1: Measurement-update
set auxiliary variables:
 $\phi_{k,i} \leftarrow \hat{\mathbf{s}}_{k,i|i-1}$
 $\mathbf{\Pi}_{k,i} \leftarrow \mathbf{\Pi}_{k,i|i-1}$
for $l \in \mathcal{N}_k$ do
 $\mathbf{R}_{e,i} \leftarrow \mathbf{V}_{l,i} + \mathbf{H}_{l,i} \mathbf{\Pi}_{k,i} \mathbf{H}_{l,i}^T$
 $\phi_{k,i} \leftarrow \phi_{k,i} + \mathbf{\Pi}_{k,i} \mathbf{H}_{l,i}^T \mathbf{R}_{e,i}^{-1} [\mathbf{y}_{l,i} - \mathbf{H}_{l,i} \phi_{k,i}]$ (3)
 $\mathbf{\Pi}_{k,i|i} \leftarrow \mathbf{\Pi}_{k,i} - \mathbf{\Pi}_{k,i} \mathbf{H}_{l,i}^T \mathbf{R}_{e,i}^{-1} \mathbf{H}_{l,i} \mathbf{\Pi}_{k,i}$
end for
Step 2: Diffusion-update
 $\hat{\mathbf{s}}_{k,i|i} \leftarrow \sum_{l \in \mathcal{N}_k} c_{lk}(i) \phi_{l,i}$ (4)
Step 3: Time-update
 $\hat{\mathbf{s}}_{k,i+1|i} = \mathbf{F}_i \hat{\mathbf{s}}_{k,i|i}$
 $\mathbf{\Pi}_{k,i+1|i} = \mathbf{F}_i \mathbf{\Pi}_{k,i|i} \mathbf{F}_i^T + \mathbf{G}_i \mathbf{W}_i \mathbf{G}_i^T$
End

communication with their neighbors to save more resources and bandwidth.

In [12], the selecting and scattering tasks have been realized by a diagonal selection matrix, $\mathbf{\Lambda}_{k,i} \in \mathbb{R}^{n \times n}$ which has r ones and $n - r$ zeros on its diagonal. The positions of ones determine which entries of the intermediate state estimate of node k are selected to be scattered at time instant i . It is obvious that multiplication of $\phi_{k,i}$ by $\mathbf{\Lambda}_{k,i}$ gives a vector whose non-selected entries are zero. In the partial diffusion algorithm, however; at time instant i the nodes need to know which elements of their neighbors intermediate estimates are shared. Consequently, the address, i.e., position of ones in the vector of communicated elements should be transmitted as well. To bypass the need for addressing, Arablouei et al. in [11] have proposed two different schemes called coordinated and uncoordinated schemes, requiring less memory and are generally easier to implement.

Remark 1. It should be noted that, unlike the original diffusion Kalman filter [1], in the diffusion update step of PDKF, only an updated mean is calculated but the covariance matrix remains unchanged. Such modification deteriorates the steady-state performance, but the communication cost is reduced. As shown in [12], such modification does not make the algorithm unstable. Therefore, PDKF is a suitable solution in practical applications (such as wireless sensor networks) where the power consumption is an important issue.

III. ADAPTIVE COMBINERS

In diffusion implementations, the combination weights $\{c_{lk}(i)\}$ play a crucial role. Some examples of static combination rules are demonstrated in Table I. These rules keep the combination weights $\{c_{lk}(i)\}$ constant and calculate the weights based solely on the network topology. Such schemes, however, are sensitive to the spatial variation of signal and noise statistics through the network. Generally speaking, high SNR conditions could deteriorate the accuracy of estimates by

TABLE I
DIFFERENT COMBINATION RULES FOR DIFFUSION NETWORKS

| Combination Rule | Entries of Combination Matrix \mathbf{C} |
|----------------------|---|
| Uniform [28] | $c_{lk} = 1/\eta_k$ if $l \in \mathcal{N}_k$ |
| Metropolis [24] | $c_{lk} = \begin{cases} 1/\max\{\eta_l, \eta_k\} & \text{if } l \neq k \\ 1 - \sum_{m \in \mathcal{N}_k/\{k\}} c_{mk} & \text{if } l = k \end{cases}$ |
| Maximum-degree [29] | $c_{lk} = \begin{cases} 1/P & \text{if } l \neq k \\ 1 - (\eta_k - 1)/P & \text{if } l = k \end{cases}$ |
| Relative-degree [25] | $c_{lk} = \eta_l / \sum_{m \in \mathcal{N}_k} \eta_m$ if $l \in \mathcal{N}_k$ |
| Non-cooperation | $c_{lk} = \begin{cases} 0 & \text{if } l \neq k \\ 1 & \text{if } l = k \end{cases}$ |

allocating less weights to the estimates from neighbors with lower SNR conditions. Consequently, static combination rules are likely to result in performance degradation.

To improve the robustness to such cases, we shall design the combination weights $\{c_{lk}(i)\}$ in the diffusion phase of Algorithm 1. Doing so, an optimization problem is formulated that its solution leads to calculation of a set of weights. In what follows, the optimal combiners are approximated by a stochastic gradient type algorithm. The algorithm is fully distributed and runs in real-time where there is no necessity to access the global information.

A. Problem Formulation

Let accumulate all the intermediate estimates (at the end of measurement-update) in a row-wise manner as

$$\Phi_i \triangleq [\phi_{1,i}, \phi_{2,i}, \dots, \phi_{P,i}], \quad (n \times P)$$

Here, the selected coefficient weights are defined as

$$a_{lk}(i) \triangleq c_{lk}(i) \mathbf{\Lambda}_{l,i}$$

Algorithm 2 Partial-Diffusion Kalman filter [12]

Begin with $\hat{\mathbf{s}}_{k,0|-1} = \mathbb{E}[\mathbf{s}_0]$ and $\mathbf{\Pi}_{k,0|-1} = \mathbf{S}_0$.
Repeat at every agent k for $i \geq 0$
Step 1: Measurement-update
set auxiliary variables:
 $\phi_{k,i} \leftarrow \hat{\mathbf{s}}_{k,i|i-1}$
 $\mathbf{\Pi}_{k,i} \leftarrow \mathbf{\Pi}_{k,i|i-1}$
compute
 $\mathbf{R}_{e,i} \leftarrow \mathbf{V}_{k,i} + \mathbf{H}_{k,i} \mathbf{\Pi}_{k,i} \mathbf{H}_{k,i}^T$
 $\phi_{k,i} \leftarrow \phi_{k,i} + \mathbf{\Pi}_{k,i} \mathbf{H}_{k,i}^T \mathbf{R}_{e,i}^{-1} [\mathbf{y}_{k,i} - \mathbf{H}_{k,i} \phi_{k,i}]$ (5)
 $\mathbf{\Pi}_{k,i|i} \leftarrow \mathbf{\Pi}_{k,i} - \mathbf{\Pi}_{k,i} \mathbf{H}_{k,i}^T \mathbf{R}_{e,i}^{-1} \mathbf{H}_{k,i} \mathbf{\Pi}_{k,i}$
Step 2: Diffusion-update
 $\hat{\mathbf{s}}_{k,i|i} \leftarrow \phi_{k,i} + \sum_{l \in \mathcal{N}_k/\{k\}} c_{lk}(i) \mathbf{\Lambda}_{l,i} (\phi_{l,i} - \phi_{k,i})$ (6)
Step 3: Time-update
 $\hat{\mathbf{s}}_{k,i+1|i} = \mathbf{F}_i \hat{\mathbf{s}}_{k,i|i}$
 $\mathbf{\Pi}_{k,i+1|i} = \mathbf{F}_i \mathbf{\Pi}_{k,i|i} \mathbf{F}_i^T + \mathbf{G}_i \mathbf{W}_i \mathbf{G}_i^T$
End

Then, the k -th column of diffusion matrix \mathbf{C} , denoted by $\mathbf{c}_{k,i} \in \mathbb{R}^P$, and the k -th column of to be selected coefficient weights, denoted by $\mathbf{a}_{k,i}$, as

$$\begin{aligned}\mathbf{c}_{k,i} &\triangleq \text{col} \{c_{1k}(i), c_{2k}(i), \dots, c_{Pk}(i)\} \\ \mathbf{a}_{k,i} &\triangleq \text{col} \{a_{1k}(i), a_{2k}(i), \dots, a_{Pk}(i)\}\end{aligned}$$

Note that $\mathbf{a}_{k,i} \in \mathbb{R}^P$ represents the combination coefficient vector at agent k and at time instant i to be optimized. For each node k , our approach is to obtain a set of weights $\{a_{lk}(i)\}_{l=1,\dots,P}$ that solve the following optimization problem

$$\arg \min_{\mathbf{a}_{k,i}} \mathbb{E} \left[\|\mathbf{s}_i - ((\Phi_i - \phi_{k,i} \mathbf{1}_P^T) \mathbf{a}_{k,i})\|^2 \right] \quad (7)$$

Subjected to $a_{lk}(i) = 0$ if $l \notin \mathcal{N}_k$ and $\mathbf{1}^T \mathbf{a}_{k,i} = 1$

The constraint $\mathbf{1}_P^T \mathbf{a}_{k,i} = 1$ states that the coefficients add up to one. The dimension of problem (7) can be lowered from P to η_k by defining an auxiliary variable, Γ_k , as

$$\Gamma_k = \begin{bmatrix} \mathbf{1}_{k_1}, \dots, \mathbf{1}_{k_{\eta_k}} \end{bmatrix} \quad (P \times \eta_k)$$

where $\mathbf{1}_l$ refers to the l -th column of $P \times P$ identity matrix and $\{k_1, \dots, k_{\eta_k}\}$ denote the indexes of the node k neighbors. Therefore, any vector $\mathbf{a}_{k,i} \in \mathbb{R}^P$ that satisfies the constraints in (7) can be rewritten as

$$\mathbf{a}_{k,i} \triangleq \Gamma_k \mathbf{b}_{k,i} \quad (8)$$

using some $\mathbf{b}_{k,i} \in \mathbb{R}^{\eta_k}$ that satisfies $\mathbf{1}_{\eta_k}^T \mathbf{b}_{k,i} = 1$ and contains non-zero entries of $\mathbf{a}_{k,i}$. Here, $\mathbf{1}_{\eta_k} \triangleq \Gamma_k \mathbf{1}_P$ is all one vectors of length η_k . Hence, substituting (8) into (7), the optimization problem (7) is modified as follows:

$$\arg \min_{\mathbf{b}_{k,i} \in \mathbb{R}^{\eta_k}} \mathbf{f}(\mathbf{b}_{k,i}) \triangleq \mathbb{E} \left[\|\mathbf{s}_i - \Delta_{k,i} \mathbf{b}_{k,i}\|^2 \right] \quad (9)$$

Subjected to $\mathbf{b}_{k,i} \in V_k \triangleq \{\nu \in \mathbb{R}^{\eta_k} | \mathbf{1}_{\eta_k}^T \nu = 1\}$

where $\Delta_{k,i} \triangleq (\Phi_i - \phi_{k,i} \mathbf{1}_P^T) \Gamma_k$ and the constraint V_k of (9) refers to a bounded hyperplane [26].

Remark 2. The computational complexity of the PDKF algorithms without and with adaptive combiner is similar to that of the original DKF algorithm given by Algorithm 1. Comparing (4) and (6) reveals that both expressions require $|\mathcal{N}_k| n$ multiplications and $(|\mathcal{N}_k| - 1) n$ additions per iteration per node.

B. Steepest-Descent Solution

We can solve (9) by employing a Lagrange multiplier argument and consider the extended cost function below

$$\begin{aligned}\mathbf{J}(\mathbf{b}_k, \lambda_k) &= \mathbb{E}[\mathbf{s}_i^T \mathbf{s}_i] - 2\theta_{k,i}^T \mathbf{b}_{k,i} \\ &\quad + \mathbf{b}_{k,i}^T \Theta_{k,i} \mathbf{b}_{k,i} + 2\lambda_k (\mathbf{1}_{\eta_k}^T \mathbf{b}_{k,i} - 1)\end{aligned} \quad (10)$$

in terms of $\lambda_k (\mathbf{1}_{\eta_k}^T \mathbf{b}_{k,i} - 1)$, where $\theta_{k,i} = \mathbb{E}[\Delta_{k,i}^T \mathbf{s}_i]$ and $\Theta_{k,i} = \mathbb{E}[\Delta_{k,i}^T \Delta_{k,i}]$. Setting the individual gradients of $\mathbf{J}(\mathbf{b}_k, \lambda_k)$ with respect to $\mathbf{b}_{k,i}$ assuming that $\theta_{k,i}$ is positive-definite, we obtain

$$\mathbf{b}_{k,i}^o = \Theta_{k,i}^{-1} [\theta_{k,i} - \lambda_k \mathbf{1}_{\eta_k}] \quad (11)$$

Differentiating (10) with respect to λ_k , setting the result to zero gives and employing the constraint $\mathbf{1}_{\eta_k}^T \mathbf{b}_{k,i} = 1$, we have

$$\lambda_k^o = \frac{\mathbf{1}_{\eta_k}^T \Theta_{k,i}^{-1} \theta_{k,i} - 1}{\mathbf{1}_{\eta_k}^T \Theta_{k,i}^{-1} \mathbf{1}_{\eta_k}} \quad (12)$$

Therefore, the solution of (7) can be obtained from (8) as $\mathbf{a}_{k,i}^o \triangleq \Gamma_k \mathbf{b}_{k,i}^o$. In order to apply the standard steepest-descent method to (9) it is required to eliminate the constraint V_k . Doing so, we apply a similar technique proposed in [30]. Let \mathbb{P}_{V_k} denote the metric projection from \mathbb{R}^{η_k} onto V_k . In light of appendix in [26], \mathbb{P}_{V_k} is defined and given by

$$\mathbb{P}_{V_k}(\nu) = \left(\mathbf{I}_{\eta_k} - \frac{\mathbf{1}_{\eta_k} \mathbf{1}_{\eta_k}^T}{\eta_k} \right) \nu + \frac{\mathbf{1}_{\eta_k}}{\eta_k} \quad \forall \nu \in \mathbb{R}^{\eta_k} \quad (13)$$

The transformation $\mathbb{P}_{V_k} : \mathbb{R}^{\eta_k} \rightarrow V_k$ maps any vector $\nu \in \mathbb{R}^{\eta_k}$ into a vector $\mathbf{b}_{k,i} \in V_k$ satisfying $\mathbf{1}_{\eta_k}^T \mathbf{b}_{k,i} = 1$, i.e.

$$\mathbf{a}_{k,i} = \mathbb{P}_{V_k}(\mathbf{b}_{k,i}) \quad (14)$$

Thus, substituting (14) into (7), we arrive at the following unconstrained problem

$$\min_{\mathbf{b}_{k,i}} \mathbb{E} \left[\|\mathbf{s}_i - \Delta_{k,i} \mathbb{P}_{V_k}(\mathbf{b}_{k,i})\|^2 \right] \quad (15)$$

Consequently, we suggest minimizing (15) employing a gradient-descent algorithm as

$$\mathbf{b}_{k,i+1} = \mathbf{b}_{k,i} + \mu_{k,i} \mathbf{V}_k [\mathbf{b}_{k,i} - \Theta_{k,i} \mathbb{P}_{V_k}(\mathbf{b}_{k,i})] \quad (16)$$

where $\mu_{k,i} \geq 0$ is a step-size parameter and \mathbf{V}_k is defined as

$$\mathbf{V}_k \triangleq \mathbf{I}_{\eta_k} - \frac{\mathbf{1}_{\eta_k} \mathbf{1}_{\eta_k}^T}{\eta_k}$$

Since $\mathbf{a}_{k,i} \in V_k$ is equivalent to $\mathbb{P}_{V_k}(\mathbf{a}_{k,i}) = \mathbf{a}_{k,i}$, our recursion is simplified as follows

$$\mathbf{b}_{k,i+1} = \mathbf{b}_{k,i} + \mu_{k,i} \mathbf{V}_k [\theta_{k,i} - \Theta_{k,i} \mathbf{b}_{k,i}] \quad (17)$$

where $\mathbf{a}_{k,0}$ must satisfy $\mathbf{1}_{\eta_k}^T \mathbf{a}_{k,0} = 1$. Remember that the desired coefficients $\mathbf{a}_{k,i}$ can be obtained through $\mathbf{a}_{k,i+1} = \Gamma_k \mathbf{b}_{k,i+1}$.

C. Adaptive Solution

In order to derive an adaptive version of recursion (17), we replace the quantities $\Theta_{k,i}$ and $\theta_{k,i}$ by their instantaneous approximations

$$\begin{aligned}\theta_{k,i} &= \mathbb{E}[\Delta_{k,i}^T \mathbf{s}_i] \approx \Delta_{k,i-1}^T \hat{\mathbf{s}}_{k,i-1|i} \\ \Theta_{k,i} &= \mathbb{E}[\Delta_{k,i}^T \Delta_{k,i}] \approx \Delta_{k,i-1}^T \Delta_{k,i-1}\end{aligned}$$

Substituting these approximations in (17), the adaptive weights algorithm, which is summarized in Algorithm 3, is obtained.

As we will see in the next step, the PDKF algorithm is stable in mean sense if the combination weights for $\{a_{lk}(i)\}_{l \in \mathcal{N}_k}$ is convex. Since the constraint $\mathbf{1}_P^T \mathbf{a}_{k,i} = 1$ is enforced by Algorithm 3, if $a_{lk}(i) \geq 0$ for all $l \in \{1, \dots, P\}$ the weight

Algorithm 3 Adaptive weights for PDKF

For every agent k , start with $\mathbf{b}_{k,0} \in \mathbb{R}^{\eta_k}$, such that $\mathbf{1}_{\eta_k}^T \mathbf{b}_{k,0} = 1$. Then, for every time instant $i \geq 0$, repeat

$$\begin{cases} \mathbf{b}_{k,i+1} = \mathbf{b}_{k,i} + \mu_{k,i} \mathbf{V}_k \Theta_{k,i-1}^T (\hat{\mathbf{s}}_{k,i-1|i} - \Theta_{k,i-1} \mathbf{a}_{k,i}) \\ \mathbf{a}_{k,i+1} = \Gamma_k \mathbf{b}_{k,i+1} \end{cases} \quad (18)$$

vectors $\mathbf{a}_{k,i}$ become convex combination. One possible choice for $\mu_{k,i}$ that guarantees $a_{lk}(i) \geq 0$ is the following

$$\begin{aligned} \mu_{k,i+1} &= \max \{0, \{\mathbf{b}_{k,i+1}(j) \mid 1 \leq j \leq \eta_k\}\} \\ \mathbf{a}_{k,i+1} &= \frac{1}{\mathbf{1}_{\eta_k}^T \mu_{k,i+1}} \Gamma_k \mu_{k,i+1} \end{aligned}$$

for all k and i , where $\mathbf{b}_{k,i+1}(j)$ is the j -th component of $\mathbf{b}_{k,i+1}$.

IV. MEAN PERFORMANCE

As the proposed algorithm is an iterative estimator to solve the estimation problem, its stability and statistical bias should be investigated. Thus, in this section the mean analysis of PDKF algorithm with any adaptive combiners, including the proposed combiner is examined. We consider both coordinated and uncoordinated partial-diffusion schemes. To begin with, let us derive the network update equation in the following section.

A. Network Update Equation

Returning to the recursion equation (5), as the predicted estimate of \mathbf{s}_i obtained by node k , $\hat{\mathbf{s}}_{k,i|i-1}$ is a better estimate for \mathbf{s}_i than $\phi_{k,i}$, we obtain

$$\phi_{k,i} = \hat{\mathbf{s}}_{k,i|i-1} + \Pi_{k,i} \mathbf{H}_{k,i}^T \mathbf{R}_{e,i}^{-1} [\mathbf{y}_{k,i} - \mathbf{H}_{k,i} \hat{\mathbf{s}}_{k,i|i-1}] \quad (19)$$

Let define the following weight errors:

$$\tilde{\phi}_{k,i} \triangleq \mathbf{s}_i - \phi_{k,i}, \quad \tilde{\mathbf{s}}_{k,i|i-1} \triangleq \mathbf{s}_i - \hat{\mathbf{s}}_{k,i|i-1}$$

where $\tilde{\phi}_{k,i}$ and $\tilde{\mathbf{s}}_{k,i|i-1}$ represent the estimation errors at the end of measurement and diffusion-update, respectively. Then, subtracting (19) from \mathbf{s}_i and using state-space model (1) gives

$$\begin{aligned} \tilde{\phi}_{k,i} &= \tilde{\mathbf{s}}_{k,i|i-1} - \Pi_{k,i} \mathbf{H}_{k,i}^T \mathbf{R}_{e,i}^{-1} (\mathbf{H}_{k,i} \tilde{\mathbf{s}}_{k,i|i-1} + \mathbf{v}_{k,i}) \\ &= (\mathbf{I}_n - \Pi_{k,i} \mathbf{H}_{k,i}^T \mathbf{R}_{e,i}^{-1} \mathbf{H}_{k,i}) \tilde{\mathbf{s}}_{k,i|i-1} - \Pi_{k,i} \mathbf{H}_{k,i}^T \mathbf{R}_{e,i}^{-1} \mathbf{v}_{k,i} \end{aligned} \quad (20)$$

Employing the matrix inversion lemma, we obtain $\Pi_{k,i} \mathbf{H}_{k,i}^T \mathbf{R}_{e,i}^{-1} \mathbf{V}_{k,i} = \Pi_{k,i} \mathbf{H}_{k,i}^T \mathbf{R}_{e,i}^{-1}$ (see appendix A in [13]) and conclude

$$\tilde{\phi}_{k,i} = (\mathbf{I}_n - \Pi_{k,i} \Omega_{k,i}) \tilde{\mathbf{s}}_{k,i|i-1} - \Pi_{k,i} \mathbf{H}_{k,i}^T \mathbf{V}_{k,i}^{-1} \mathbf{v}_{k,i} \quad (21)$$

where $\Omega_{k,i} \triangleq \mathbf{H}_{k,i}^T \mathbf{V}_{k,i}^{-1} \mathbf{H}_{k,i}$. Using (1) we have

$$\tilde{\mathbf{s}}_{k,i|i-1} = \mathbf{F}_{i-1} \tilde{\mathbf{s}}_{k,i-1|i-1} + \mathbf{G}_{i-1} \mathbf{w}_{i-1} \quad (22)$$

Substituting (22) into (21) gives

$$\begin{aligned} \tilde{\phi}_{k,i} &= (\mathbf{I}_n - \Pi_{k,i} \Omega_{k,i}) \mathbf{F}_{i-1} \tilde{\mathbf{s}}_{k,i-1|i-1} \\ &\quad + (\mathbf{I}_n - \Pi_{k,i} \Omega_{k,i}) \mathbf{G}_{i-1} \mathbf{w}_{i-1} - \Pi_{k,i} \mathbf{H}_{k,i}^T \mathbf{V}_{k,i}^{-1} \mathbf{v}_{k,i} \end{aligned} \quad (23)$$

To derive the network update equation in terms of the global quantities, we introduce the following global quantities:

$$\begin{aligned} \tilde{\mathbf{s}}_{i|i} &\triangleq \text{col} \{ \tilde{\mathbf{s}}_{1,i|i}, \tilde{\mathbf{s}}_{2,i|i}, \dots, \tilde{\mathbf{s}}_{P,i|i} \} \\ \tilde{\phi}_i &\triangleq \text{col} \{ \tilde{\phi}_{1,i}, \tilde{\phi}_{2,i}, \dots, \tilde{\phi}_{P,i} \} \\ \mathbf{v}_i &\triangleq \text{col} \{ \mathbf{v}_{1,i}, \dots, \mathbf{v}_{P,i} \} \\ \mathbf{V}_i &\triangleq \text{Blkdiag} \{ \mathbf{V}_{1,i}, \mathbf{V}_{2,i}, \dots, \mathbf{V}_{P,i} \} \\ \mathbf{H}_i &\triangleq \text{Blkdiag} \{ \mathbf{H}_{1,i}, \mathbf{H}_{2,i}, \dots, \mathbf{H}_{P,i} \} \\ \Pi_i &\triangleq \text{Blkdiag} \{ \Pi_{1,i}, \Pi_{2,i}, \dots, \Pi_{P,i} \} \\ \Omega_i &\triangleq \text{Blkdiag} \{ \Omega_{1,i}, \Omega_{2,i}, \dots, \Omega_{P,i} \} \\ \mathcal{K}_i &= \begin{bmatrix} \mathbf{K}_{1,1,i} & \cdots & \mathbf{K}_{1,P,i} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{P,1,i} & \cdots & \mathbf{K}_{P,P,i} \end{bmatrix}. \end{aligned}$$

where

$$\mathbf{K}_{p,q,i} = \begin{cases} \mathbf{I}_n - \sum_{l \in \mathcal{N}_p \setminus \{p\}} c_{lp}(i) \Lambda_{l,i} & \text{if } p = q \\ c_{qp}(i) \Lambda_{q,i} & \text{if } q \in \mathcal{N}_p \setminus \{p\} \\ \mathbf{O}_n & \text{otherwise} \end{cases}$$

Using the above definitions, the following state-space model for PDKF algorithm with adaptive combiner is obtained:

$$\tilde{\mathbf{s}}_{i|i} = \mathcal{K}_i \tilde{\phi}_i \quad (24)$$

$$\tilde{\phi}_i = \mathcal{F}_i \tilde{\mathbf{s}}_{i-1|i-1} + \mathcal{G}_i (\mathbf{1} \otimes \mathbf{n}_{i-1}) - \mathcal{D}_i \mathbf{v}_i \quad (25)$$

where

$$\begin{aligned} \mathcal{F}_i &= (\mathbf{I}_{nP} - \Pi_i \Omega_i) (\mathbf{I}_P \otimes \mathbf{F}_{i-1}) \\ \mathcal{G}_i &= (\mathbf{I}_{nP} - \Pi_i \Omega_i) (\mathbf{I}_P \otimes \mathbf{G}_{i-1}) \\ \mathcal{D}_i &= \Pi_i \mathbf{H}_i^T \mathbf{V}_i^{-1} \end{aligned}$$

Substituting (25) into (24) reveals that the global error $\tilde{\mathbf{s}}_{i|i}$ evolves according to the following recursion:

$$\tilde{\mathbf{s}}_{i|i} = \mathcal{K}_i \mathcal{F}_i \tilde{\mathbf{s}}_{i-1|i-1} + \mathcal{K}_i \mathcal{G}_i (\mathbf{1} \otimes \mathbf{w}_{i-1}) - \mathcal{K}_i \mathcal{D}_i \mathbf{v}_i \quad (26)$$

Note that (26) shows how the network weight error vector $\tilde{\mathbf{s}}_{i|i}$ evolves in time. In the sequel, we shall use this recursive equation to investigate the mean behavior of PDKF algorithm with adaptive combiner.

Here, in order to analyze the steady-state performance of PDKF, we assume the following throughout our analysis:

Assumption 2.

- (i) The matrices $\{\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{V}, \mathbf{W}\}$ described in steady-state model (1) are time invariant.
- (ii) The matrix \mathcal{F} is stable.
- (iii) The pair $\{\mathbf{F}, \mathbf{H}_k\}$ is detectable for all k , and $\{\mathbf{F}, \mathbf{G}\mathbf{W}^{\frac{1}{2}}\}$ is reachable [12].

Under Assumption 2, $\Pi_{k,i|i}$ converges to the matrix Π_k , for all k . Moreover, \mathcal{F}_i , \mathcal{G}_i and \mathcal{D}_i also converge in steady-state,

and their steady-state values are given by

$$\begin{aligned}\mathbf{\Pi} &\triangleq \lim_{i \rightarrow \infty} \mathbf{\Pi}_i = \text{Blkdiag} \{ \mathbf{\Pi}_1, \dots, \mathbf{\Pi}_N \} \\ \mathcal{F} &\triangleq \lim_{i \rightarrow \infty} \mathcal{F}_i = (\mathbf{I}_{nP} - \mathbf{\Pi}\mathbf{\Omega}) (\mathbf{I}_P \otimes \mathbf{F}) \\ \mathcal{G} &\triangleq \lim_{i \rightarrow \infty} \mathcal{G}_i = (\mathbf{I}_{nP} - \mathbf{\Pi}\mathbf{\Omega}) (\mathbf{I}_P \otimes \mathbf{G}) \\ \mathcal{D} &\triangleq \lim_{i \rightarrow \infty} \mathcal{D}_i = \mathbf{\Pi}\mathbf{H}^T \mathbf{V}^{-1}\end{aligned}$$

Since $\mathbf{\Omega}_i$ and \mathbf{H}_i are now time-invariant, $\mathbf{\Omega}$ and \mathbf{H} are used instead of them. Proposition 1 summarizes the mean performance of the PDKF algorithm over a network with adaptive combiner.

Proposition 1. Under Assumptions (1) and (2) the PDKF algorithm with adaptive combiner is convergent in the mean sense and asymptotically unbiased.

Proof: See Appendix A. ■

V. SIMULATION RESULTS

We apply the PDKF with adaptive combiner to the problem of estimating and tracking the position of a moving target by an adaptive network. Doing so, a network topology, randomly generated, with $N = 20$ nodes, in which each node is, on average, connected to two other nodes is considered. The size of system parameter is $n = 4$. The target state vector is denoted by $\mathbf{s}_i = [d_i, \dot{d}_i, v_i, \dot{v}_i]^T$, where (d_i, v_i) and (\dot{d}_i, \dot{v}_i) are the position and velocity components at time instant i , respectively. Therefore, the state equation is modeled as follows:

$$\mathbf{s}_i = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{s}_{i-1} + \mathbf{w}_{i-1}$$

where T is the sampling time which is taken as $T = 1$. The process noise \mathbf{w}_{i-1} is zero-mean white Gaussian with covariance matrix as

$$\mathbf{W} = 0.04 \begin{bmatrix} T^4/4 & T^3/2 & 0 & 0 \\ T^3/2 & T^2 & 0 & 0 \\ 0 & 0 & T^4/4 & T^3/2 \\ 0 & 0 & T^3/2 & T^2 \end{bmatrix}$$

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

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

So, the measurement at node k at time i is

$$\mathbf{y}_{k,i} = \mathbf{H}_{k,i} \mathbf{s}_i + \mathbf{v}_{k,i}$$

The measurement noise covariance matrix at agent k is, $\mathbf{V}_{k,i} = \sigma_{k,i}^2 \mathbf{I}_3$, where the noise variance $\sigma_{k,i}^2$ across the agents is selected randomly in the range $[0, 0.5]$. In the simulations, the initial target state is given by $\mathbf{s}_0 = [10, 1.5, 10, 1.2]$. The initial estimate is taken as a combination of the true state and a bias drawn from a Gaussian distribution with

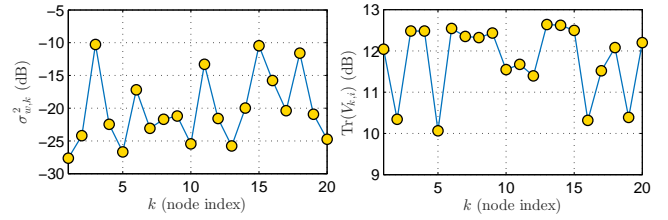


Fig. 2. State noise variances (left), and trace of observation noise covariances at all nodes (right).

mean $[0.5, 0.5, 0.5, 0.5]^T$. The initial covariance is set to be $\mathbf{\Pi}_{k,0|0} = \text{diag} \{100, 10, 100, 10\}$. The initial conditions are the same for all nodes. Moreover, 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.

For Algorithm 2, to draw a comparison with we our adaptive combiner $a_{lk}(i)$, we used the Metropolis rule to select matrix \mathbf{C} (see Table I). The state noise variances and trace of observation noise covariances at every node are generated randomly and shown in Fig. 2. The performance is measured in terms of the network Mean-Square Deviation (MSD), defined for node k at time i as:

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

The network MSD is defined as the average over all nodes.

$$\text{MSD}_{\text{net}} = \frac{1}{P} \sum_{k=1}^P \text{MSD}_{k,i}$$

Fig. 3 demonstrates the transient MSD curves of PDKF algorithm with some static and dynamic combiners and draw a comparison between adaptive combiner and some other combiners. The steady-state values for each individual node is also presented in Fig. 3. We observe that the proposed PDKF with adaptive weights considerably outperforms the non-adaptive PDKF. Fig. 4 shows the result of the nodes cooperating together to track the position of an projectile object. We can see that partial-update has a certain effect on accuracy. It also has a good performance on trajectory tracking using adaptive combiner.

It is worth mentioning that the proposed combination rule can be applied to *any* connected network. To show this, we apply the PDKF algorithm with the proposed combination rule to networks with different topologies. To generate different topologies, we assume $N = 20$ and for each network, set the number of neighbors for each individual node to 2, 4, 6, \dots 20. Fig. 5 shows the network MSD values in terms of different number of neighbors. Clearly, the PDKF algorithm with the proposed combination rule provides accurate estimates of the state vector for different network topologies.

VI. CONCLUSION

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. Here, the optimal choices for the

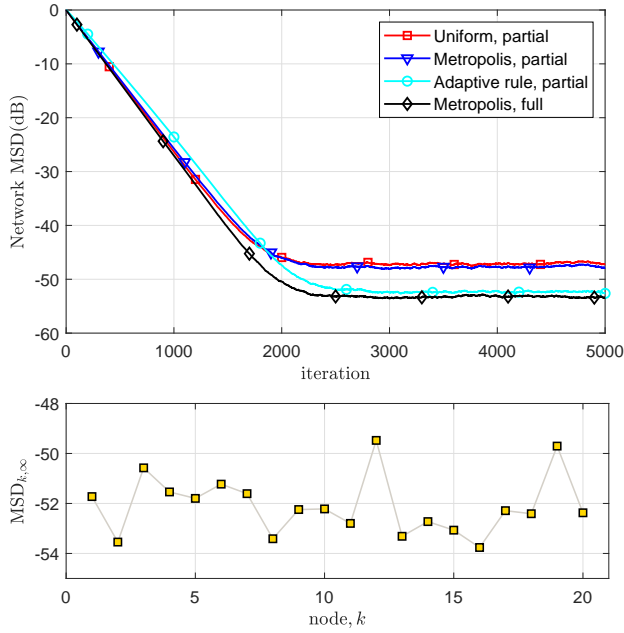


Fig. 3. Learning curves of the network MSD for different combination rules (top), steady-state values for each individual node (bottom).

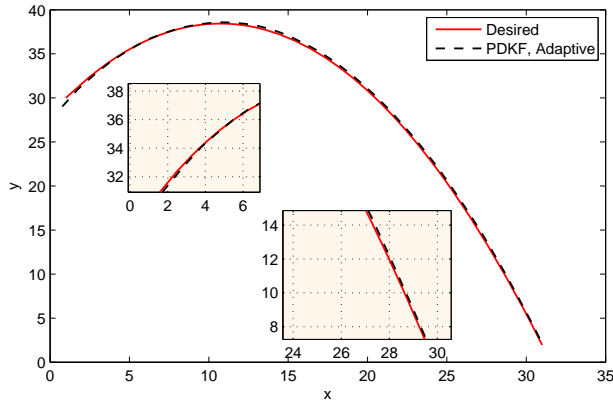


Fig. 4. Tracking a projectile object for adaptive and metropolis rule.

combination weights in PDKF have been discussed and an adaptive variant that can be computed in real-time has been proposed. The proposed algorithm outperforms the existing techniques. More importantly, simulation findings reveal that the proposed combination coefficients can greatly improve the performance of diffusion adaptation. The presented algorithm may be used in e.g. wireless sensor networks to perform detection/estimation of localized events [31], [32].

APPENDIX A PROOF OF PROPOSITION 1

Tacking the expectation on both sides of (26) and employing Assumptions 1 and 2, we find that the mean error vector

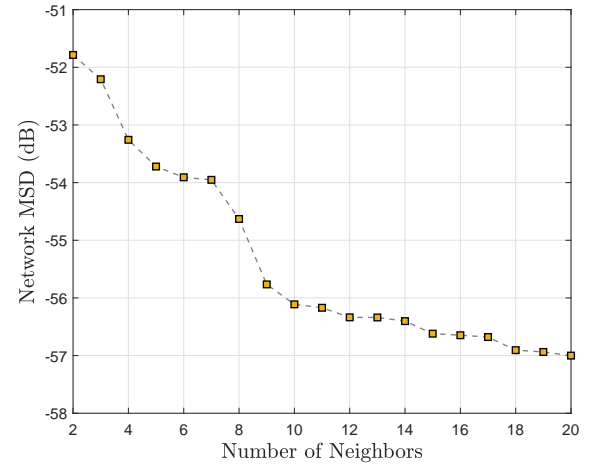


Fig. 5. Network MSD for different values of the number of neighbors.

evolves according to the following recursion:

$$\mathbb{E}[\tilde{\mathbf{s}}_{i|i}] = \mathcal{K}\mathcal{F}_i\mathbb{E}[\tilde{\mathbf{s}}_{i-1|i-1}] \quad (\text{A.1})$$

where $\mathcal{K} = \mathbb{E}[\mathcal{K}_i]$. The convergence of equation (A.1) is guaranteed, if matrix $\mathcal{K}\mathcal{F}_i$ is stable. In the light of Lemma 1, \mathcal{K} is *right-stochastic*. The Perron-Frobenius theorem [33] ensures that all eigenvalues of \mathcal{K} lie inside the unit disc. Consequently, as $i \rightarrow \infty$, the mean stability and asymptotic unbiasedness of the algorithm is achieved if matrix \mathcal{F} is stable. Therefore, we have

$$\lim_{i \rightarrow \infty} \mathbb{E}[\tilde{\mathbf{s}}_{i|i}] = \mathbf{O}_{nP}$$

where $\mathbf{O}_{nP} \in \mathbb{R}^{MN}$ denotes zero vector. This means that the PDKF algorithm with adaptive combiner is convergent in the mean sense and is asymptotically unbiased.

Lemma 1. \mathcal{K}_i is a *right-stochastic* matrix.

Proof: Note that can \mathcal{K}_i be expressed in a compact-form as

$$\mathcal{K}_i = \mathbf{C}_i \odot \mathbf{\Lambda}_i + (\mathbf{I}_{nP} - \mathbf{\Lambda}_i \mathbf{C}_i) \odot \mathbf{I}_{nP} \quad (\text{B.1})$$

where

$$\mathbf{C}_i = \mathbf{C}_i \otimes \mathbf{I}_n, \quad \mathbf{\Lambda}_i = \mathbf{1}_P \otimes [\mathbf{\Lambda}_{1,i}, \dots, \mathbf{\Lambda}_{P,i}]$$

Since the probability of to be transmitted entries is equal for all the nodes, we can write

$$\begin{aligned} \mathbb{E}[\mathbf{\Lambda}_i] &= \mathbf{1}_P \otimes \mathbb{E}[\mathbf{\Lambda}_{1,i}, \dots, \mathbf{\Lambda}_{P,i}] \\ &= \mathbf{1}_P \otimes \mathbf{1}_P^T \otimes \rho \mathbf{I}_n = \rho \mathbf{J}_P \otimes \mathbf{I}_n, \end{aligned} \quad (\text{A.2})$$

$$\begin{aligned} \mathbb{E}[\mathbf{\Lambda}_i] \mathbf{C} &= \rho (\mathbf{J}_P \otimes \mathbf{I}_n) (\mathbf{C} \otimes \mathbf{I}_n) \\ &= \rho \mathbf{J}_P \mathbf{C} \otimes \mathbf{I}_n = \rho \mathbf{J}_P \otimes \mathbf{I}_n \end{aligned} \quad (\text{A.3})$$

and

$$\mathbf{C}^T \odot \mathbb{E}[\mathbf{\Lambda}_i] = \rho \mathbf{C}^T \odot (\mathbf{J}_P \otimes \mathbf{I}_n) = \rho \mathbf{C}^T \quad (\text{A.4})$$

where ρ denotes the probability of transmission for each entry at any node k and \mathbf{J}_P is an all one matrix of size $P \times P$. Note that $\mathcal{C} = \mathbb{E}[\mathcal{C}_i]$. Therefore, we have

$$\begin{aligned} \mathcal{K} &= \mathbb{E}[\mathcal{K}_i] = \mathcal{C} \odot \mathbb{E}[\mathbf{\Lambda}_i] + (\mathbf{I}_{nP} - \mathbb{E}[\mathbf{\Lambda}_i] \mathcal{C}) \odot \mathbf{I}_{nP} \\ &= \rho \mathcal{C}^T + (\mathbf{I}_{nP} - \rho \mathbf{J}_P \otimes \mathbf{I}_n) \odot \mathbf{I}_{nP} \\ &= \rho \mathcal{C}^T + (1 - \rho) \mathbf{I}_{nP} \end{aligned} \quad (\text{A.5})$$

Using the definition of \mathcal{K}_i and $\mathbf{K}_{p,q,i}$, we have

$$\begin{aligned} \sum_{q=1}^P \mathbf{K}_{p,q,i} &= \mathbf{I}_n - \sum_{l \in \mathcal{N}_p / \{p\}} c_{lp} \mathbf{\Lambda}_{l,i} + \sum_{q \in \mathcal{N}_q / \{q\}} c_{qp} \mathbf{\Lambda}_{q,i} \\ &= \mathbf{I}_n, \quad p = \{1, \dots, P\} \\ \mathcal{K}_i \mathbf{1}_{nP} &= \begin{bmatrix} \sum_{q=1, \dots, P} \mathbf{K}_{1,q,i} \mathbf{1}_n \\ \vdots \\ \sum_{q=1, \dots, P} \mathbf{K}_{P,q,i} \mathbf{1}_n \end{bmatrix} = \mathbf{1}_{nP} \end{aligned}$$

REFERENCES

- [1] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *Automatic Control, IEEE Transactions on*, vol. 55, no. 9, pp. 2069–2084, 2010.
- [2] 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.
- [3] F. Meyer, O. Hlinka, and F. Hlawatsch, "Sigma point belief propagation," *IEEE Signal Processing Letters*, vol. 21, no. 2, pp. 145–149, 2014.
- [4] J. Yi, X. Wan, and D. Li, "Exactly decoupled Kalman filtering for multitarget state estimation with sensor bias," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.
- [5] 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, 2017.
- [6] S. Li, Y. Cheng, D. Brown, R. Tharmarasa, G. Zhou, and T. Kirubarajan, "Comprehensive time-offset estimation for multisensor target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 2351–2373, 2019.
- [7] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Transactions on aerospace and electronic systems*, vol. 39, no. 4, pp. 1333–1364, 2003.
- [8] 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.
- [9] S. Xie and H. Li, "Distributed lms estimation over networks with quantised communications," *International Journal of Control*, vol. 86, no. 3, pp. 478–492, 2013.
- [10] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay, "Distributed least mean-square estimation with partial diffusion," *Signal Processing, IEEE Transactions on*, vol. 62, no. 2, pp. 472–484, 2014.
- [11] R. Arablouei, K. Dogancay, S. Werner, and Y.-F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *Signal Processing, IEEE Transactions on*, vol. 62, no. 14, pp. 3510–3522, 2014.
- [12] V. Vahidpour, A. Rastegarnia, A. Khalili, W. Bazzi, and S. Sanei, "Partial diffusion kalman filtering for distributed state estimation in multiagent networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3839–3846, Dec 2019.
- [13] V. Vahidpour, A. Rastegarnia, M. Latifi, A. Khalili, and S. Sanei, "Performance analysis of distributed kalman filtering with partial diffusion over noisy network," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.
- [14] 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.
- [15] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.
- [16] N. Takahashi and I. Yamada, "Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3518–3521.
- [17] X. Zhao and A. H. Sayed, "Single-link diffusion strategies over adaptive networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3749–3752.
- [18] O. L. Rortveit, J. H. Husoy, and A. H. Sayed, "Diffusion lms with communication constraints," in *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*. IEEE, 2010, pp. 1645–1649.
- [19] J.-W. Lee, S.-E. Kim, and W.-J. Song, "Data-selective diffusion LMS for reducing communication overhead," *Signal Processing*, vol. 113, pp. 211–217, 2015.
- [20] M. O. Sayin and S. S. Kozat, "Single bit and reduced dimension diffusion strategies over distributed networks," *Signal Processing Letters, IEEE*, vol. 20, no. 10, pp. 976–979, 2013.
- [21] —, "Compressive diffusion strategies over distributed networks for reduced communication load," *Signal Processing, IEEE Transactions on*, vol. 62, no. 20, pp. 5308–5323, 2014.
- [22] R. Arablouei, S. Werner, K. Dogancay, and Y.-F. Huang, "Analysis of a reduced-communication diffusion LMS algorithm," *Signal Processing*, vol. 117, pp. 355–361, 2015.
- [23] A. Rastegarnia, "Reduced-communication diffusion rls for distributed estimation over multi-agent networks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2019.
- [24] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.
- [25] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, 2008.
- [26] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4795–4810, 2010.
- [27] S. Yang, C. Xu, X. Qiu, and D. O. Wu, "Diffusion Kalman filter with quantized information exchange in distributed mobile crowdsensing," *IEEE Internet of Things Journal*, 2018.
- [28] W. Kocay and D. L. Kreher, *Graphs, algorithms, and optimization*. Chapman and Hall/CRC, 2016.
- [29] D. S. Scherber and H. C. Papadopoulos, "Locally constructed algorithms for distributed computations in ad-hoc networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 11–19.
- [30] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," 2005.
- [31] D. Ciuonzo and P. S. Rossi, "Distributed detection of a non-cooperative target via generalized locally-optimum approaches," *Information Fusion*, vol. 36, pp. 261 – 274, 2017.
- [32] D. Ciuonzo and P. S. Rossi, "Quantizer design for generalized locally optimum detectors in wireless sensor networks," *IEEE Wireless Communications Letters*, vol. 7, no. 2, pp. 162–165, 2018.
- [33] C. D. Meyer, *Matrix analysis and applied linear algebra*. Siam, 2000, vol. 2.



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.



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.



Wael M. Bazzi graduated from the American University of Beirut (AUB), Lebanon, in 1996. He received the M.E. degree from AUB in 1999 and the Ph.D. degree from the University of Waterloo, Canada, in 2001. He is currently an associate professor at the American University in Dubai. His research interests include wireless communication and networks, especially the optimization and modeling aspects of communication networks and systems.



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 monographs, 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.