# Diffusion Strategies for Distributed Kalman Filtering and Smoothing

Federico S. Cattivelli, *Student Member, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

***Abstract*—We study the problem of distributed Kalman filtering and smoothing, where a set of nodes is required to estimate the state of a linear dynamic system from in a collaborative manner. Our focus is on diffusion strategies, where nodes communicate with their direct neighbors only, and the information is diffused across the network through a sequence of Kalman iterations and data-aggregation. We study the problems of Kalman filtering, fixed-lag smoothing and fixed-point smoothing, and propose diffusion algorithms to solve each one of these problems. We analyze the mean and mean-square performance of the proposed algorithms, provide expressions for their steady-state mean-square performance, and analyze the convergence of the diffusion Kalman filter recursions. Finally, we apply the proposed algorithms to the problem of estimating and tracking the position of a projectile. We compare our simulation results with the theoretical expressions, and note that the proposed approach outperforms existing techniques.**

***Index Terms*—Adaptive networks, diffusion networks, distributed estimation, fixed-lag smoothing, fixed-point smoothing, Kalman filtering.**

## I. Introduction

WE consider the problem of distributed Kalman filtering and smoothing over a network of nodes. It is assumed that some system of interest is evolving according to linear state-space dynamics, and that every node in the network takes measurements that are linearly related to the unobserved state. The objective in the Kalman filtering case is for every node to estimate the state of the system at a particular time $i$, based on local observations and neighboring interactions up to that time. In the fixed-point smoothing case, all nodes are interested in estimating the state at some fixed instant $i_0$, given observations up to time $i \geq i_0$. In the fixed-lag smoothing case, the nodes estimate the state at a time $i-L$, given observations up to time $i$, where $L$ is a fixed positive integer. Applications of distributed Kalman filtering include wireless localization, target tracking and precision agriculture.

The performance of the state estimation procedure will depend heavily on the collaboration strategy employed. In a centralized solution, all nodes send their measurements to a fusion center, which uses a conventional Kalman filtering or smoothing algorithm to obtain the optimal state estimate, and then sends the resulting estimate back to every node. We refer to the estimate obtained in this way as the *global* estimate, since it uses all measurements from all nodes in the network. This strategy may require a large amount of energy for communications [3] and has the potential for a critical failure point at the central node. Distributed strategies are an attractive alternative, since they are in general more robust, may require fewer communications, and allow parallel processing.

Decentralized Kalman filtering has been proposed previously in [4] for a decentralized control problem, where it is assumed that the network is fully connected. The same assumption is used in [5] and [6], where the latter work considers the interesting case of severely quantized communications. In [7], the Kalman filtering iterations are parallelized over a set of sensors, though a fusion center is required to combine the estimates. Another algorithm for distributed Kalman filtering was proposed in [8], where global information about the state covariances is required in order to compute the estimates. Average consensus has also been used for distributed Kalman filtering [9]–[11]. The distributed Kalman filtering work of [12] is also based on average consensus, but uses node hierarchy, and the nodes perform different types of processing and communications. We use the results of [11] and [12] for comparison purposes here. Recent work [13] proposes an efficient algorithm for large-scale distributed Kalman filtering, where the transfer matrices are assumed to be sparse and localized, while [14] considers optimization of the consensus weights for a scalar state-space model.

Our focus is on *diffusion* Kalman filtering and smoothing, where nodes communicate only with their neighbors, and no fusion center is present. Furthermore, the strategies proposed in this work give no hierarchy to the nodes in the network: every node does the same type of processing and communication, therefore enabling a fully distributed solution. Non-hierarchical networks are robust to node and link failure, are flexible for ad-hoc deployment and topology changes, and at the same time do not require complex routing protocols. Moreover, we do not impose strict restrictions on the topology of the network or the sparsity of the model, neither do we require running consensus iterations between measurements.

In this work we extend our previous results on diffusion Kalman filtering [1] and smoothing [2]. The results are also inspired by our previous work on diffusion RLS [15], [16], and by connections between Kalman and RLS filtering as established in [17]–[20]. Our algorithms compute, for every measurement and for every node, a local state estimate using the data from the neighborhood. Subsequently, every node computes a local convex combination of the estimates of the neighborhood.

Our arguments are based on diffusion strategies [15], [16], [21]–[23] as opposed to consensus strategies [9]–[11]. In

Manuscript received February 16, 2009; revised June 26, 2009, November 23, 2009, and December 15, 2009; accepted January 06, 2010. First published February 17, 2010; current version published September 09, 2010. This work was supported in part by the National Science Foundation under awards ECS-0601266, ECS-0725441 and CCF-094936. Recommended by Associate Editor Z. Wang.

The authors are with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mails: fcattiv@ee.ucla.edu; sayed@ee.ucla.edu).

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

Digital Object Identifier 10.1109/TAC.2010.2042987

consensus-based solutions, several averaging iterations are generally required for each new measurement in order to obtain the estimate [24]–[26]. Although algorithms where only one consensus iteration occurs between measurements have been considered in [9]–[11], these algorithms differ from our proposed approach in several ways. First and foremost, the algorithms derived in [9]–[11] are motivated by consenus arguments, where the fundamental objective is that all nodes should obtain the same estimate in steady-state. In contrast, our distributed schemes are motivated by diffusion arguments that do not impose this restriction, and can therefore lead to improved performance. Our work is motivated by our earlier realization that while consensus solutions are useful for computing averages, diffusion solutions are particularly suited for problems involving the recursive minimization of cost functions [16], [21]–[23], [27]–[29]. As such, diffusion solutions are well suited for estimation problems where new measurements are being taken in real time (see also [1], [2], [16]). The idea is that in estimation problems, the objective is to minimize a cost function, and consensus between nodes may not be required. A second difference between our work and [10], [11] is that the latter are motivated by consensus arguments in continuous time and the filters are subsequently discretized. Our derivation motivates the filters by working directly in discrete time and by attempting to minimize mean-square-error cost functions. Moreover, we also provide performance and convergence analysis for the proposed algorithms.

### A. Summary of Contributions and Organization of the Work

We start by providing background material on Kalman filtering and smoothing in Section II. In Section III, we define the problem of distributed Kalman filtering, and motivate and derive algorithms for distributed filtering and smoothing. Our proposed diffusion Kalman filtering algorithm allows every node in the network to obtain a state estimate based on local communications with its neighbors only. We also propose algorithms for distributed fixed-point and fixed-lag smoothing. In Section IV we provide mean and mean-square performance analyses of the algorithms, and also study the convergence of the diffusion Kalman filter. We show that the algorithms will converge as long as all neighborhoods are locally convergent. Finally, in Section V we provide simulations for the case where the network is tracking the position of a projectile. We compare with the theoretical expressions derived in Section IV as well as with other existing algorithms, showing improvement in performance.

## II. Background

### A. The Kalman Filter

Consider a state-space model of the form

$$\begin{aligned} x_{i+1} &= F_i x_i + G_i n_i \\ y_i &= H_i x_i + v_i \end{aligned} \tag{1}$$

where $x_i \in \mathbb{C}^M$ and $y_i \in \mathbb{C}^{pN}$ denote the state and measurement vectors of the system, respectively, at time $i$, and $M$, $N$ and $p$ are positive integers. The signals $n_i$ and $v_i$ denote state and measurement noises, respectively, and are assumed to be zero-mean, uncorrelated and white, with covariance matrices denoted by

$$\mathrm{E}\begin{bmatrix} n_i \\ v_i \end{bmatrix}\begin{bmatrix} n_j \\ v_j \end{bmatrix}^* = \begin{bmatrix} Q_i & 0 \\ 0 & R_i \end{bmatrix}\delta_{ij} \tag{2}$$

where the operator $^*$ denotes complex conjugate transposition and $\delta_{ij}$ is the Kronecker delta. The initial state $x_0$ is assumed to be zero-mean, with covariance matrix $\Pi_0 > 0$, and is uncorrelated with $n_i$ and $v_i$, for all $i$. We further assume that $R_i > 0$. The case where $\mathrm{E}\, n_i v_i^* \neq 0$ can always be transformed into an equivalent problem of the form (2) as explained in [18]. The cases where $\mathrm{E}\, x_0 \neq 0$ or when we have a deterministic input in the state (1) can be handled similarly as shown in Section III-F.

Let $\hat{x}_{i|j}$ denote the linear minimum mean-square error estimate of $x_i$ given observations up to and including time $j$. The Kalman filter in its time- and measurement-update forms can be computed by starting from $\hat{x}_{0|-1} = 0$ and $P_{0|-1} = \Pi_0$ and iterating the following equations [18]–[20]:

$$\begin{aligned} &\underline{\mathrm{Measurement-Update:}} \\ &R_{e,i} = R_i + H_i P_{i|i-1} H_i^* \\ &\hat{x}_{i|i} = \hat{x}_{i|i-1} + P_{i|i-1} H_i^* R_{e,i}^{-1}[y_i - H_i \hat{x}_{i|i-1}] \\ &P_{i|i} = P_{i|i-1} - P_{i|i-1} H_i^* R_{e,i}^{-1} H_i P_{i|i-1} \\ &\underline{\mathrm{Time-Update:}} \\ &\hat{x}_{i+1|i} = F_i \hat{x}_{i|i} \\ &P_{i+1|i} = F_i P_{i|i} F_i^* + G_i Q_i G_i^* \end{aligned} \tag{3}$$

where $P_{i|j}$ denotes the covariance matrix of the estimation error, $\tilde{x}_{i|j} \triangleq x_i - \hat{x}_{i|j}$. An alternative form for the measurement update is

$$\hat{x}_{i|i} = \hat{x}_{i|i-1} + P_{i|i} H_i^* R_i^{-1}[y_i - H_i \hat{x}_{i|i-1}]. \tag{4}$$

When $P_{i|i-1}$ is nonsingular, we also have

$$P_{i|i}^{-1} = P_{i|i-1}^{-1} + H_i^* R_i^{-1} H_i.$$

### B. The Fixed-Point Smoother

We now consider a Kalman smoother, where we wish to estimate the state at some fixed time $i_0$, given all observations up to time $i$. In Appendix A we show that this estimate can be found from the filtered and predicted estimates through

$$\hat{x}_{i_0|i} = \hat{x}_{i_0|i-1} + M_{i_0,i}[\hat{x}_{i|i} - \hat{x}_{i|i-1}] \tag{5}$$

and its covariance matrix is given by

$$P_{i_0|i} = P_{i_0|i-1} + M_{i_0,i}(P_{i|i} - P_{i|i-1})M_{i_0,i}^*. \tag{6}$$

The matrix $M$ satisfies the recursion

$$M_{i_0,i+1} = M_{i_0,i} P_{i|i} F_i^* P_{i+1|i}^{-1}. \tag{7}$$

Equations (5), (6) and (7), together with the Kalman filter recursions (3), give a set of recursions that allow us to compute the estimate of $x_{i_0}$ given observations up to time $i > i_0$, the initial estimate $\hat{x}_{i_0|i_0-1}$, and the error covariance matrix $P_{i_0|i_0-1}$. The initial condition for $M_{i_0,i}$ is $M_{i_0,i_0} = I$.

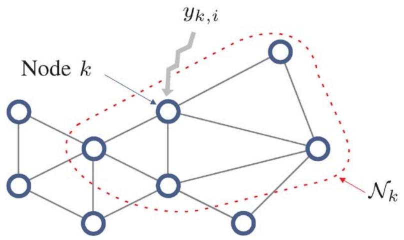

Fig. 1. At every time $i$, node $k$ collects a measurement $y_{k,i}$.

### C. The Fixed-Lag Smoother

The fixed-lag smoother estimates $x_{i-L}$ given observations up to time $i$, where $L$ is some fixed positive integer constant. At time $i$, the Kalman filter algorithm (3) provides the estimates $\hat{x}_{j|j}$ and $\hat{x}_{j|j-1}$ and the error covariances $P_{j|j}$ and $P_{j|j-1}$, for the time instants $j = i-L, \ldots, i$. Thus, starting from $\hat{x}_{i-L|i-L}$, we can use the fixed-point smoother equations to obtain, recursively, the estimate $\hat{x}_{i-L|i}$ as follows:

$$\left\{\begin{array}{l} M_{i-L,i-L} = I \\ \text{for } j = i-L+1 : i \text{ repeat} \\ \quad M_{i-L,j} = M_{i-L,j-1}P_{j-1|j-1}F_{j-1}^*P_{j|j-1}^{-1} \\ \quad \hat{x}_{i-L|j} = \hat{x}_{i-L|j-1} + M_{i-L,j}[\hat{x}_{j|j} - \hat{x}_{j|j-1}] \\ \quad P_{i-L|j} = P_{i-L|j-1} + M_{i-L,j}(P_{j|j} - P_{j|j-1})M_{i-L,j}^* \\ \text{end.} \end{array}\right. \tag{8}$$

Note that $P_{i-L|i}$ is not required for the computation of $\hat{x}_{i-L|i}$.

## III. Distributed Kalman Filtering

Consider a set of $N$ nodes spatially distributed over some region. We say that two nodes are connected if they can communicate directly with each other. A node is always conected to itself. The set of nodes connected to a certain node $k$ is called the *neighborhood* of node $k$ and is denoted by $\mathcal{N}_k$ (notice that $k \in \mathcal{N}_k$). The number of neighbors of node $k$ is called the *degree* of node $k$ and is denoted by $n_k$. We define an $N$ by $N$ adjacency matrix $A$ as follows:

$$[A]_{l,k} = \begin{cases} 1 & \text{if } l \in \mathcal{N}_k \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

That is, the $(l,k)$ entry of $A$ is one if nodes $l$ and $k$ are connected, and zero otherwise. It is assumed that at time $i$, every node $k$ collects a measurement $y_{k,i} \in \mathbb{C}^p$ according to model (1) as follows:

$$y_{k,i} = H_{k,i}x_i + v_{k,i} \qquad k = 1, \ldots, N. \tag{10}$$

The process is shown schematically in Fig. 1. Model (1) is related to (10) by stacking the $N$ measurements across all nodes at time $i$ as follows:

$$y_i = \begin{bmatrix} y_{1,i} \\ \vdots \\ y_{N,i} \end{bmatrix}, \quad H_i = \begin{bmatrix} H_{1,i} \\ \vdots \\ H_{N,i} \end{bmatrix}, \quad v_i = \begin{bmatrix} v_{1,i} \\ \vdots \\ v_{N,i} \end{bmatrix}. \tag{11}$$

We further assume that the measurement noises $v_{k,i}$ are spatially uncorrelated, i.e.

$$\mathrm{E}\, v_{k,i}v_{l,j}^* = R_{k,i}\delta_{i,j}\delta_{k,l}$$

where $R_{k,i} > 0$ for all $k$, $i$.

The objective in distributed Kalman filtering and smoothing implementations is for every node $k$ in the network to compute an estimate of the unknown state $x_i$, while sharing data only with its neighbors $\{l \in \mathcal{N}_k\}$. The challenge is to obtain a state estimate that is as accurate as the global state estimate that would result if each node had access to all measurements across all nodes in the network. We will denote the estimates of $x_i$ obtained by node $k$ and based on local observations up to time $j$ as $\hat{x}_{k,i|j}$; these estimates are based only on information collected from the neighborhood of node $k$.

### A. Local Kalman Filter

Assume for the sake of argument that a node $k$ has access to the measurements of its neighbors $\mathcal{N}_k$. A local estimate at node $k$ can be computed from (3) by running multiple measurement-updates, one for every neighbor of $k$ [18, p. 329]. The iterations are shown in (12)

$$\begin{array}{l} \text{At node } k \text{ and at time } i, \text{ start with} \\ \psi_{k,i} \leftarrow \hat{x}_{k,i|i-1} \\ P_{k,i} \leftarrow P_{k,i|i-1} \\ \text{Then, for each } l \in \mathcal{N}_k \text{ repeat :} \\ \quad R_e \leftarrow R_{l,i} + H_{l,i}P_{k,i}H_{l,i}^* \\ \quad \psi_{k,i} \leftarrow \psi_{k,i} + P_{k,i}H_{l,i}^*R_e^{-1}[y_{l,i} - H_{l,i}\psi_{k,i}] \\ \quad P_{k,i} \leftarrow P_{k,i} - P_{k,i}H_{l,i}^*R_e^{-1}H_{l,i}P_{k,i} \\ \text{end} \\ \hat{x}_{k,i|i} \leftarrow \psi_{k,i} \\ P_{k,i|i} \leftarrow P_{k,i} \\ \hat{x}_{k,i+1|i} = F_i\hat{x}_{k,i|i} \\ P_{k,i+1|i} = F_iP_{k,i|i}F_i^* + G_iQ_iG_i^* \end{array} \tag{12}$$

where "$\leftarrow$" denotes a sequential, or non-concurrent assignment.

An alternative form of (12) that will be useful in future sections is given below, where we assume $P_{k,i|i-1} > 0$ for all $k$ and $i$ (see Appendix B for a derivation)

$$\begin{array}{l} \text{At node } k \text{ and at time } i: \\ P_{k,i|i}^{-1} = P_{k,i|i-1}^{-1} + \displaystyle\sum_{l\in\mathcal{N}_k} H_{l,i}^*R_{l,i}^{-1}H_{l,i} \\ \hat{x}_{k,i|i} = \hat{x}_{k,i|i-1} + P_{k,i|i}\displaystyle\sum_{l\in\mathcal{N}_k} H_{l,i}^*R_{l,i}^{-1}[y_{l,i} - H_{l,i}\hat{x}_{k,i|i-1}] \\ \hat{x}_{k,i+1|i} = F_i\hat{x}_{k,i|i} \\ P_{k,i+1|i} = F_iP_{k,i|i}F_i^* + G_iQ_iG_i^*. \end{array} \tag{13}$$

We refer to the measurement update of (12) as the *incremental* step [29] since the optimal local estimate is generated by incrementally incorporating data sequentially from the neighborhood. The iterations (12) compute the optimal estimate for node $k$ by incorporating *only* the measurements $\{y_{l,i}\}$ from its neighbors $l \in \mathcal{N}_k$. However, recursions (12) do not exploit the fact that besides measurements $\{y_{l,i}\}$, the neighbors of node $k$ also have their own estimates for the same state vector $x_i$, say $\{\psi_{l,i}\}$. According to our notation and initialization procedure in (12), the symbol $\psi_{l,i}$ is used to denote the intermediate local estimate that is available at a neighboring node $l$ before it incorporates information from its own neighbors. It would therefore be useful to take advantage of both $\{y_{l,i}, \psi_{l,i}\}$ in (12) or (13). This is what the diffusion step will do, as we explain next.

### *B. Motivation for the Diffusion Step*

The diffusion Kalman filtering algorithm that is motivated here is based on extending our previous work on diffusion RLS filtering [15], [16] to the stochastic domain. The diffusion step in [16] was motivated by a least-squares deterministic argument. We now resort to a stochastic argument to motivate the diffusion step in the Kalman filtering context.

Starting from (12), we see that at the end of the incremental update, each node $k$ ends up with an optimal local estimate $\psi_{k,i}$ after incorporating all measurements from its neighborhood $\mathcal{N}_k$. In (12), this estimate becomes the new updated state estimate $\hat{x}_{k,i|i}$ and its error covariance matrix is $P_{k,i|i}$. The question now is how to define the updated estimate $\hat{x}_{k,i|i}$ not only in terms of $\psi_{k,i}$ but also in terms of the local estimates $\{\psi_{l,i}\}$ in the neighborhood $\mathcal{N}_k$. The motivation for doing so is to attempt to have the resulting local estimate $\hat{x}_{k,i|i}$ be closer to the global estimate $\hat{x}_{i|i}$ that is based on the data across the entire network. To see how this can be done, let us examine the relation between the $\{\psi_{l,i}\}$ in the neighborhood $\mathcal{N}_k$ and the global estimate $\hat{x}_{i|i}$.

Thus, assume initially for the sake of argument that the adjacency matrix $A$ of the network satisfies

$$\sum_{k=1}^{N} [A]_{l,k}\gamma_k = 1 \quad \text{for all } l = 1, \ldots, N \tag{14}$$

for some weights $\gamma_k$. For example, if $A$ is invertible, then the vector of weights $\gamma = \text{col}\{\gamma_k\}$ is given by $\gamma = A^{-1}\mathbb{1}$ and condition (14) is always satisfied. In general, however, it may be the case that (14) cannot be satisfied for a given $A$. Fortunately, there is a way around this problem which does not modify the topology of the network. Every node, by definition, is connected to itself, and therefore $[A]_{k,k} = 1$ for all $k$. However, we may decide that a particular node $k$ does not need to communicate with itself, and we can set $[A]_{k,k} = 0$. In this way, if $A$ is not invertible, we can show (see Appendix C) that we can always make $A$ invertible by appropriately setting some of its diagonal elements to zero. Thus, condition (14) will be satisfied without affecting the network topology.

Nonetheless, as the presentation will reveal, assumption (14) is only used to motivate the diffusion update, but is not needed for the diffusion algorithms. Since the measurement noises $\{v_{k,i}\}$ at different nodes are assumed uncorrelated (and zero-mean), it can be shown (see Appendix D) that the global and local estimates are related through

$$P_{i|i}^{-1}\hat{x}_{i|i} = \sum_{k=1}^{N} \gamma_k P_{k,i|i}^{-1}\psi_{k,i} \tag{15}$$

where $P_{i|i}$ denotes the covariance matrix of the global estimate error, $\tilde{x}_{i|i}$, and $P_{k,i|i}$ denotes the covariance matrix of the individual estimate error, $\tilde{x}_{k,i|i}$. These matrices are related through

$$P_{i|i}^{-1} = \sum_{k=1}^{N} \gamma_k P_{k,i|i}^{-1} - \left(\sum_{k=1}^{N} \gamma_k - 1\right)\Pi_i^{-1} \tag{16}$$

where $\Pi_i$ is the covariance matrix of $x_i$. Now, as suggested by (13), since the covariance matrices of the individual errors, $P_{k,i|i}$ are expected to get smaller with time, the first term on the right hand side becomes dominant. The above relation can be approximated by

$$P_{i|i}^{-1} \approx \sum_{k=1}^{N} \gamma_k P_{k,i|i}^{-1}$$

and therefore

$$\hat{x}_{i|i} \approx \left[\sum_{k=1}^{N} \gamma_k P_{k,i|i}^{-1}\right]^{-1} \sum_{k=1}^{N} \gamma_k P_{k,i|i}^{-1}\psi_{k,i}.$$

The above relation has the form of a weighted average of the form

$$\hat{x}_{i|i} \approx \sum_{k=1}^{N} \Gamma_k \psi_{k,i} \tag{17}$$

where the averaging matrices satisfy

$$\sum_{k=1}^{N} \Gamma_k = I. \tag{18}$$

Result (17) suggests one approximate way by which the estimates $\{\psi_{l,i}\}$ at all nodes in the neighborhood of $k$ can be fused locally. For example, for any two nodes $l$ and $k$, we can assign a non-negative weight $c_{l,k}$, and we set $c_{l,k} = 0$ if $l$ and $k$ are not connected. We select the weights such that

$$\sum_{l \in \mathcal{N}_k} c_{l,k} = 1$$

and then, motivated by (17), we replace the assignment $\hat{x}_{k,i|i} \leftarrow \psi_{k,i}$ in (12) by the following computation:

$$\hat{x}_{k,i|i} = \sum_{l \in \mathcal{N}_k} c_{l,k}\psi_{l,i}. \tag{19}$$

We call (19) the *diffusion* update. Convex combinations as in (19) have been considered before in the context of adaptive filtering [16], [22], [30]–[32]. A distributed Kalman filtering algorithm based on this update is discussed next.

### *C. The Diffusion Kalman Filter Algorithm (diffKF)*

The diffusion KF algorithm and its variants therefore require the introduction of a matrix $C \in \mathbb{R}^{N \times N}$ with the following properties:

$$\mathbb{1}^T C = \mathbb{1}^T \qquad c_{l,k} = 0 \text{ if } l \neq \mathcal{N}_k \qquad c_{l,k} \geq 0 \tag{20}$$

where $\mathbb{1}$ is an $N \times 1$ column vector with unit entries, and $c_{l,k}$ is the $(l, k)$ element of matrix $C$. We call $C$ the *diffusion* matrix, since it governs the diffusion process, and plays an important role in the steady-state performance of the network. Note that $C$ is a column-stochastic matrix. The entries in $C$ represent the weights used by the diffusion algorithm to combine neighborhood estimates as in (19).

The diffusion Kalman filtering algorithm is derived by adding the diffusion step (19) after the Kalman filter measurement update of the local Kalman filter (12). The diffusion step is an attempt to approximate the global KF performance via local node interactions. The diffusion KF algorithm is described below.

**Algorithm 1: Diffusion Kalman filter (time- and measurement-update form)**

Consider a state-space model as in (1) and a diffusion matrix as in (20). Start with $\hat{x}_{k,0|-1} = 0$ and $P_{k,0|-1} = \Pi_0$ for all $k$, and at every time instant $i$, compute at every node $k$:

```
Step 1: Incremental Update
        $\psi_{k,i} \leftarrow \hat{x}_{k,i|i-1}$
        $P_{k,i} \leftarrow P_{k,i|i-1}$
    for every neighboring node $l \in \mathcal{N}_k$, repeat
        $R_e \leftarrow R_{l,i} + H_{l,i} P_{k,i} H_{l,i}^*$
        $\psi_{k,i} \leftarrow \psi_{k,i} + P_{k,i} H_{l,i}^* R_e^{-1} [y_{l,i} - H_{l,i} \psi_{k,i}]$
        $P_{k,i} \leftarrow P_{k,i} - P_{k,i} H_{l,i}^* R_e^{-1} H_{l,i} P_{k,i}$
    end
Step 2: Diffusion Update
        $\hat{x}_{k,i|i} \leftarrow \sum_{l \in \mathcal{N}_k} c_{l,k} \psi_{l,i}$
        $P_{k,i|i} \leftarrow P_{k,i}$
        $\hat{x}_{k,i+1|i} = F_i \hat{x}_{k,i|i}$
        $P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*$.
```

Algorithm 1 requires that at every instant $i$, nodes communicate to their neighbors their measurement matrices $H_{k,i}$, the covariance matrices $R_{k,i}$, and the measurements $y_{k,i}$ for the incremental update, and the intermediate estimates $\psi_{k,i}$ for the diffusion update. The total communication requirement for every node and for every measurement is $pM + M + p^2/2 + 3p/2$ complex scalars, and it requires one $p \times p$ matrix inversion for each neighbor in the incremental update.

This process is shown schematically in Fig. 2, where the transmission of $R_{k,i}$ has been omitted to simplify the figure, and due to the following argument. Note that communication of $R_{k,i}$ may not be necessary if its Cholesky factor is computed, $R_{k,i} = L_{k,i} L_{k,i}^*$, and $\bar{H}_{k,i} = L_{k,i}^{-1} H_{k,i}$ and $\bar{y}_{k,i} = L_{k,i}^{-1} y_{k,i}$ are transmitted instead of $H_{k,i}$ and $y_{k,i}$. In this case, the error covariance is updated using $R_e \leftarrow I + \bar{H}_{l,i} P_{k,i} \bar{H}_{l,i}^*$, and the remaining recursions replace $H_{k,i}$ and $y_{k,i}$ by $\bar{H}_{k,i}$ and $\bar{y}_{k,i}$. In this scenario, Algorithm 1 requires transmission of $pM + M + p$ complex scalars per node per measurement.

It is important to note that even though the notation $P_{k,i|i}$ and $P_{k,i|i-1}$ has been retained for simplicity in Alg. 1, these matrices do *not* represent the covariances of the state estimation errors $\tilde{x}_{k,i|i}$ and $\tilde{x}_{k,i|i-1}$ any longer, since the diffusion update is not taken into account in the recursions for these matrices. Exact expressions for the covariances of the state estimates will be derived in Section IV.

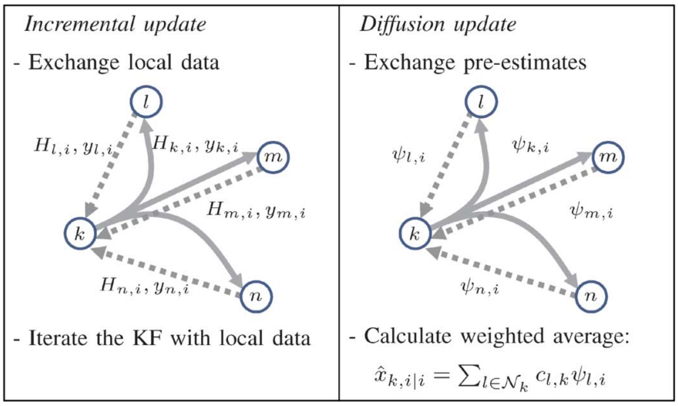

Fig. 2. Diffusion Kalman filter update at node $k$ (transmission of $R_{k,i}$ has been omitted to simplify the figure).

An alternative formulation of Algorithm 1 may be obtained by using the information form of the Kalman filter (13) instead of (12) to compute the incremental update. We also assume that $P_{k,i|i-1} > 0$ for all $k$ and $i$ in this alternative form.[1] We refer to this form as Algorithm 2, and present it as follows.

**Algorithm 2: Diffusion Kalman filter (information form)**

Consider a state-space model as in (1) and a diffusion matrix as in (20). Start with $\hat{x}_{k,0|-1} = 0$ and $P_{k,0|-1} = \Pi_0$ for all $k$, and at every time instant $i$, compute at every node $k$:

```
Step 1: Incremental Update:
        $S_{k,i} = \sum_{l \in \mathcal{N}_k} H_{l,i}^* R_{l,i}^{-1} H_{l,i}$
        $q_{k,i} = \sum_{l \in \mathcal{N}_k} H_{l,i}^* R_{l,i}^{-1} y_{l,i}$
        $P_{k,i|i}^{-1} = P_{k,i|i-1}^{-1} + S_{k,i}$
        $\psi_{k,i} = \hat{x}_{k,i|i-1} + P_{k,i|i} [q_{k,i} - S_{k,i} \hat{x}_{k,i|i-1}]$.
Step 2: Diffusion Update:
        $\hat{x}_{k,i|i} = \sum_{l \in \mathcal{N}_k} c_{l,k} \psi_{l,i}$
        $\hat{x}_{k,i+1|i} = F_i \hat{x}_{k,i|i}$
        $P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*$.
```

After every instant $i$, node $k$ communicates the quantities $H_{k,i}^* R_{k,i}^{-1} H_{k,i}$, $H_{k,i}^* R_{k,i}^{-1} y_{k,i}$ and $\psi_{k,i}$ to its neighbors. The total communication required per measurement per node, is $M^2/2 + 5M/2$ scalars, and it requires two matrix inversions per incremental update of sizes $M \times M$ and $p \times p$. Algorithms 1 and 2 are mathematically equivalent under the assumption $P_{k,i|i-1} > 0$.

The incremental update of Algorithm 2 is similar to the update used in [11]. An important difference in the algorithms is in the diffusion step. In [11], the author uses a continuous-time

[1] A sufficient condition for this to hold is that $F_i$ be invertible [18].

consensus-based approach for averaging and arrives at the following relation (Algorithm 2 in [11]):

$$\begin{aligned}\hat{x}_{k,i|i} &= \psi_{k,i} + \epsilon \sum_{l\in\mathcal{N}_k} (\psi_{l,i} - \psi_{k,i}) \\ &= (1-(n_k-1)\epsilon)\,\psi_{k,i} + \epsilon \sum_{l\in\mathcal{N}_k-\{k\}} \psi_{l,i}. \qquad (21)\end{aligned}$$

Observe that the weights are $\epsilon$ and $1-(n_k-1)\epsilon$. In contrast, we use a convex combination of the estimates of the neighbors as in (19) with more general weights $\{c_{l,k}\}$, motivated by the optimal fusion analysis of Appendix D. This is a key difference with [11], which produces improvement in performance as shown by the simulation results of Section V. Moreover, in [33] we allow $\{c_{l,k}\}$ to change with time and show how to adapt the time-varying weights $\{c_{l,k,i}\}$ as well.

### *D. Diffusion Fixed-Point Smoother*

From (5) and (7) we know that the Kalman smoother update can be computed by using knowledge of the Kalman filtering variables $\hat{x}_{i|i}$ and $\hat{x}_{i|i-1}$. Thus, a diffusion Kalman smoothing algorithm can be motivated by adding recursions for $\hat{x}_{k,i_0|i}$ and $M_{k,i_0,i}$ as shown below.

Algorithm 3 uses the diffusion Kalman filter (Alg. 1 or 2) to compute the estimates $\hat{x}_{k,i|i}$ and $\hat{x}_{k,i|i-1}$, and uses these to update the fixed-point estimate $\hat{x}_{k,i_0|i}$. Note that by using either Alg. 1 or 2 in the first two steps of Alg. 3, we can formulate two different versions of the algorithm: time-and measurement-update form, and information form.

**Algorithm 3: Diffusion fixed-point smoother**

Consider a state-space model as in (1) and a diffusion matrix as in (20). Start with $\hat{x}_{k,0|-1} = 0$, $P_{k,0|-1} = \Pi_0$ and $M_{k,i_0,i_0} = I$ for all $k$, and at every time instant $i$, compute at every node $k$:

```
    Steps 1 and 2: Run steps 1 and 2 of either Algorithm 1 or 2.
    Step 3: Smoother update:
      if $i \geq i_0$ :
        $M_{k,i_0,i} = M_{k,i_0,i-1} P_{k,i-1|i-1} F_{i-1}^* P_{k,i|i-1}^{-1}$
        $\hat{x}_{k,i_0|i} = \hat{x}_{k,i_0|i-1} + M_{k,i_0,i}(\hat{x}_{k,i|i} - \hat{x}_{k,i|i-1})$
      else
        $\hat{x}_{k,i_0|i} = F_{i_0-1} F_{i_0-2} \ldots F_i \hat{x}_{k,i|i}$
      end.
```

### *E. Diffusion Fixed-Lag Smoother*

Likewise, from (8) we know that the fixed-lag estimate can be computed by using knowledge of the Kalman filtering variables $\hat{x}_{i|i}$ and $\hat{x}_{i|i-1}$. Thus, a diffusion fixed-lag smoother can be motivated by adding the recursions for $\hat{x}_{k,i-L|j}$ and $M_{k,j,j}$ for $j = i-L+1, \ldots, i$ as shown below.

Alg. 4 uses the diffusion Kalman filter (Alg. 1 or 2) to compute the estimates $\hat{x}_{k,j|j}$ and $\hat{x}_{k,j|j-1}$ for instants $j = i-L, \ldots, i$, and uses these to calculate the fixed-lag estimate $\hat{x}_{k,i|i+L-1}$. Note again that by using either Alg. 1 or 2 in the first two steps of Alg. 4, we can formulate two different versions of the algorithm: time-and measurement-update form, and information form.

**Algorithm 4: Diffusion fixed-lag smoother**

Consider a state-space model as in (1) and a diffusion matrix as in (20). Start with $\hat{x}_{k,0|-1} = 0$ and $P_{k,0|-1} = \Pi_0$ for all $k$, and at every time instant $i$, compute at every node $k$:

```
    Steps 1 and 2: Run steps 1 and 2 of either Algorithm 1 or 2.
    Step 3: Smoother update:
If $i \geq L-1$,
  $M_{k,i-L,i-L} = I$
  for $j = i-L+1, \ldots, i$ repeat
    $M_{k,i-L,j} = M_{k,i-L,j-1} P_{k,j-1|j-1} F_{j-1}^* P_{k,j|j-1}^{-1}$
    $\hat{x}_{k,i-L|j} = \hat{x}_{k,i-L|j-1} + M_{k,i-L,j}[\hat{x}_{k,j|j} - \hat{x}_{k,j|j-1}]$
    $P_{k,i-L|j} = P_{k,i-L|j-1}$
        $+ M_{k,i-L,j}(P_{k,j|j} - P_{k,j|j-1}) M_{k,i-L,j}^*$
  end
end.
```

### *F. Non-Zero Mean and Non-Zero Input Case*

In some cases we may encounter state-space systems of the form

$$\begin{cases} w_{i+1} = F_i w_i + G_i n_i + u_i \\ z_{k,i} = H_{k,i} w_i + v_{k,i}, \quad k = 1, \ldots N \end{cases} \qquad (22)$$

where $u_i$ is a deterministic input, and $\mathrm{E}\, w_0 \neq 0$. Note that we use the notation $w_i$ instead of $x_i$ for the state variable at time $i$, and we use $z_{k,i}$ instead of $y_{k,i}$ for the measurement obtained by node $k$ at time $i$. Now introduce the zero-mean variables $x_i \triangleq w_i - \mathrm{E}\, w_i$ and $y_{k,i} \triangleq z_i - \mathrm{E}\, z_i$. Then we have

$$\begin{cases} \mathrm{E}\, w_{i+1} = F_i \mathrm{E}\, w_i + u_i \\ \mathrm{E}\, z_{k,i} = H_{k,i} \mathrm{E}\, w_i, \quad k = 1, \ldots N. \end{cases} \qquad (23)$$

Therefore, from $\mathrm{E}\, w_0$ and the model parameters, we can calculate $\mathrm{E}\, w_i$ and $\mathrm{E}\, z_{k,i}$ for all $k$ and $i$. Subtracting (23) from (22), we obtain

$$\begin{cases} x_{i+1} = F_i x_i + G_i n_i \\ y_{k,i} = H_{k,i} x_i + v_{k,i}, \quad k = 1, \ldots N \end{cases} \qquad (24)$$

which now has the same form as (1). Therefore, we can apply the diffusion Kalman filtering algorithms proposed in this section to model (24) to estimate $x_i$ given the observations $y_{k,i}$ up to a certain time $j$. In order to estimate $w_i$ from the observations of $z_{k,i}$ up to time $j$ in model (22), we use

$$\hat{w}_{i|j} = \mathrm{E}\, w_i + \hat{x}_{i|j}.$$

Note that the errors satisfy $\tilde{w}_{i|j} = \tilde{x}_{i|j}$ and therefore have the same covariance matrices.

## IV. Performance Analysis

### A. Summary of Main Results

In this section we analyze the mean, mean-square and convergence performance of the diffusion Kalman filtering algorithms. The main results of this section are summarized as follows. We start by expressing the estimation error at every node and every time instant through the convenient expression (27). Then, in Section IV-C we use this result to show that the diffusion Kalman filter estimates are unbiased for all instants $i \geq 0$. In Section IV-D we study the mean-square performance of the filter, in terms of the mean-square deviation (MSD) for every node, which is defined for node $k$ as

$$\mathrm{MSD}_{k,i} = \mathrm{E}\|x_i - \hat{x}_{k,i|i}\|^2.$$

The MSD is indexed by time $i$ and node $k$, since for diffusion algorithms, different nodes produce different estimates in general. We show that the state estimation covariance across the entire network satisfies the Lyapunov-like recursion (32). Then we introduce two assumptions: first, that the state-space model is time invariant, and second, that if every node were to use a conventional Kalman filter on the measurements from its neighborhood, its estimate would converge (more precisely, the pair $\{F, H_k^{\mathrm{loc}}\}$ needs to be detectable for all $k$). Under these assumptions, we prove in Section IV-F that (32) converges, and provide a closed form expression for the steady-state MSD performance in (41). We also extend our results to the proposed smoothing algorithms in Section IV-E. We formalize our results in Theorem 2 at the end of the section.

### B. Preliminaries

For our analysis, we use Algorithm 1 to derive the expressions. The analysis holds also for Algorithm 2 when $P_{k,i|i-1} > 0$, since in this case both algorithms are mathematically equivalent. Let $\tilde{\psi}_{k,i} = x_i - \psi_{k,i}$ denote the estimation error at the *end* of the incremental update. Let $n_k$ denote the degree of node $k$, and let the set $\{k_j\}, j = 1, \ldots, n_k$ denote indices of the neighbors of node $k$. Consider the incremental step of Algorithm 1, and let $P_{k,i}^{(j)}$ and $R_e^{(j)}$ denote the values of $P_{k,i}$ and $R_e$ at iteration $j$, where $j = 1, \ldots, n_k$, $P_{k,i}^{(0)} = P_{k,i|i-1}$ and $P_{k,i}^{n_k} = P_{k,i|i}$. Iterating the incremental step over the neighbors of node $k$ we obtain

$$\begin{aligned}
\tilde{\psi}_{k,i} = {} & \left[I - P_{k,i}^{(n_k-1)} H_{k_{n_k},i}^* \left(R_e^{(n_k)}\right)^{-1} H_{k_{n_k},i}\right] \cdots \\
& \left[I - P_{k,i}^{(0)} H_{k_1,i}^* \left(R_e^{(1)}\right)^{-1} H_{k_1,i}\right] \tilde{x}_{k,i|i-1} \\
& + \left[I - P_{k,i}^{(n_k-1)} H_{k_{n_k},i}^* \left(R_e^{(n_k)}\right)^{-1} H_{k_{n_k},i}\right] \cdots \\
& \left[I - P_{k,i}^{(1)} H_{k_2,i}^* \left(R_e^{(2)}\right)^{-1} H_{k_2,i}\right] \\
& \times P_{k,i}^{(0)} H_{k_1,i}^* \left(R_e^{(1)}\right)^{-1} v_{k_1,i} \\
& \vdots \\
& + P_{k,i}^{(n_k-1)} H_{k_{n_k},i}^* \left(R_e^{(n_k)}\right)^{-1} v_{k_{n_k},i}.
\end{aligned}$$

Noting that $P_{k,i}^{(j)} H_{k_j,i}^* R_{k_j,i}^{-1} = P_{k,i}^{(j-1)} H_{k_j,i}^* (R_e^{(j)})^{-1}$, the above expression can be rewritten

$$\begin{aligned}
\tilde{\psi}_{k,i} &= \tilde{x}_{k,i|i-1} - P_{k,i|i} \sum_{l \in \mathcal{N}_k} H_{l,i}^* R_{l,i}^{-1} \left(H_{l,i} \tilde{x}_{k,i|i-1} + v_{l,i}\right) \\
&= (I - P_{k,i|i} S_{k,i}) \tilde{x}_{k,i|i-1} - P_{k,i|i} \sum_{l \in \mathcal{N}_k} H_{l,i}^* R_{l,i}^{-1} v_{l,i} \qquad (25)
\end{aligned}$$

where $\tilde{x}_{k,i|i-1} = x_i - \hat{x}_{k,i|i-1}$ denotes the estimation error at node $k$ at the end of the diffusion update, and $S_{k,i}$ was defined in Algorithm 2. We also have

$$\tilde{x}_{k,i|i-1} = F_{i-1} \tilde{x}_{k,i-1|i-1} + G_{i-1} n_{i-1}. \qquad (26)$$

Combining (25) into the diffusion step of Algorithm 1, we obtain

$$\begin{aligned}
\tilde{x}_{k,i|i} &= \sum_{l \in \mathcal{N}_k} c_{l,k} \tilde{\psi}_{l,i} \\
&= \sum_{l \in \mathcal{N}_k} c_{l,k} \Big[(I - P_{l,i|i} S_{l,i}) \tilde{x}_{l,i|i-1} \\
&\qquad - P_{l,i|i} \sum_{m \in \mathcal{N}_l} H_{m,i}^* R_{m,i}^{-1} v_{m,i}\Big]. \qquad (27)
\end{aligned}$$

### C. Mean Performance

Taking expectations of both sides of (26) and (27), we obtain the following recursions for the expectations of the estimates by the diffusion KF algorithm:

$$\mathrm{E}\tilde{x}_{k,i|i-1} = F_{i-1} \mathrm{E}\tilde{x}_{k,i-1|i-1} \qquad (28)$$

$$\mathrm{E}\tilde{x}_{k,i|i} = \sum_{l \in \mathcal{N}_k} c_{l,k} (I - P_{l,i|i} S_{l,i}) \mathrm{E}\tilde{x}_{l,i|i-1}. \qquad (29)$$

Since $\hat{x}_{k,0|-1} = 0$ and $\mathrm{E}\, x_0 = 0$, we have $\mathrm{E}\tilde{x}_{k,0|-1} = 0$ for all $k$ and therefore

$$\mathrm{E}\tilde{x}_{k,0|0} = \sum_{l \in \mathcal{N}_k} c_{l,k} (I - P_{l,0|0} S_{l,0}) \mathrm{E}\tilde{x}_{l,0|-1} = 0.$$

Thus, we conclude by iterating (28) and (29) that the diffusion KF estimates are unbiased for all $i \geq 0$.

### D. Mean-Square Performance of the Diffusion Kalman Filter

Consider the augmented state-error vector $\tilde{\chi}_{i|i}$ and the block-diagonal matrices $\mathcal{H}_i$, $\mathcal{P}_{i|i}$ and $\mathcal{S}_i$ defined as follows:

$$\begin{aligned}
\tilde{\chi}_{i|i} &\triangleq \mathrm{col}\{\tilde{x}_{1,i|i}, \ldots, \tilde{x}_{N,i|i}\} \\
\mathcal{H}_i &\triangleq \mathrm{diag}\{H_{1,i}, \ldots, H_{N,i}\} \\
\mathcal{P}_{i|i} &\triangleq \mathrm{diag}\{P_{1,i|i}, \ldots, P_{N,i|i}\} \\
\mathcal{S}_i &\triangleq \mathrm{diag}\{S_{1,i}, \ldots, S_{N,i}\}.
\end{aligned}$$

The vector $\tilde{\chi}_{i|i}$ collects the state errors across all nodes in the network. Consider also the extended matrices [from (20) and (9)]:

$$\mathcal{C} \triangleq C \otimes I_M \qquad \mathcal{A} \triangleq A \otimes I_M$$

where $\otimes$ denotes Kronecker product. We may now express (26) and (27) in a global form that captures the evolution of the entire network

$$\begin{aligned}\tilde{\chi}_{i|i} = \mathcal{C}^T(I-\mathcal{P}_{i|i}\mathcal{S}_i)\big[&(I\otimes F_{i-1})\tilde{\chi}_{i-1|i-1}\\ &+(I\otimes G_{i-1})(\mathbb{1}\otimes n_{i-1})\big] - \mathcal{C}^T\mathcal{P}_{i|i}\mathcal{A}^T\mathcal{H}_i^* R_i^{-1} v_i\end{aligned} \tag{30}$$

where $v_i$ was defined in (11) and $R_i = \mathrm{E}v_i v_i^*$ is a block-diagonal matrix. Equation (30) can be rewritten more compactly as

$$\tilde{\chi}_{i|i} = \mathcal{F}_i\tilde{\chi}_{i-1|i-1} + \mathcal{G}_i(\mathbb{1}\otimes n_{i-1}) - \mathcal{D}_i v_i \tag{31}$$

where

$$\begin{aligned}\mathcal{F}_i &\triangleq \mathcal{C}^T(I-\mathcal{P}_{i|i}\mathcal{S}_i)(I\otimes F_{i-1})\\ \mathcal{G}_i &\triangleq \mathcal{C}^T(I-\mathcal{P}_{i|i}\mathcal{S}_i)(I\otimes G_{i-1})\\ \mathcal{D}_i &\triangleq \mathcal{C}^T\mathcal{P}_{i|i}\mathcal{A}^T\mathcal{H}_i^* R_i^{-1}.\end{aligned}$$

Let $\mathcal{P}_{\tilde{\chi},i} = \mathrm{E}\tilde{\chi}_{i|i}\tilde{\chi}_{i|i}^*$ denote the covariance matrix of $\tilde{\chi}_{i|i}$. From (31) and the whiteness assumptions on the state and measurement noises, we obtain the recursion:

$$\boxed{\mathcal{P}_{\tilde{\chi},i} = \mathcal{F}_i\mathcal{P}_{\tilde{\chi},i-1}\mathcal{F}_i^* + \mathcal{G}_i(\mathbb{1}\mathbb{1}^T\otimes Q_{i-1})\mathcal{G}_i^* + \mathcal{D}_i R_i\mathcal{D}_i^*} \tag{32}$$

where we have used the property of Kronecker products that $(A\otimes B)(C\otimes D) = (AC)\otimes(BD)$. In order to analyze the mean-square steady-state performance, we assume that model (1) is time-invariant.

*Assumption 1:* The matrices in model (1) are time-invariant, i.e., the matrices $F$, $G$, $H$, $R$ and $Q$ do not depend on $i$.

We also assume that the local Kalman filtering recursions (12) converge for every neighborhood. By this we mean the following. Again, let $n_k$ denote the degree of node $k$ and let the set $\{k_m\}, m = 1,\ldots,n_k$ denote the indices of the neighbors of node $k$. We collect the observation matrices of the neighbors of node $k$ as follows:

$$\begin{aligned}H_k^{\mathrm{loc}} &\triangleq \mathrm{col}\left\{H_{k_1}, H_{k_2},\ldots,H_{k_{n_k}}\right\}\\ R_k^{\mathrm{loc}} &\triangleq \mathrm{diag}\left\{R_{k_1}, R_{k_2},\ldots,R_{k_{n_k}}\right\}\end{aligned}$$

where $H_{k_m}$ and $R_{k_m}$ are used instead of $H_{k_m,i}$ and $R_{k_m,i}$ since these matrices are now time-invariant under Assumption 1. The measurement update in Algorithms 1 to 4 can be rewritten

$$P_{k,i|i} = P_{k,i|i-1} - P_{k,i|i-1}\left(H_k^{\mathrm{loc}}\right)^* R_{e,k,i}^{-1} H_k^{\mathrm{loc}} P_{k,i|i-1} \tag{33}$$

where $R_{e,k,i} = R_k^{\mathrm{loc}} + H_k^{\mathrm{loc}}P_{k,i|i-1}(H_k^{\mathrm{loc}})^*$. Using (33) and the time update of Algorithms 1 to 4, we obtain that the matrix $P_{k,i+1|i}$ in these algorithms satisfies the following Riccati recursion:

$$P_{k,i+1|i} = FP_{k,i|i-1}F^* + GQG^* - K_{p,k,i}R_{e,k,i}K_{p,k,i}^* \tag{34}$$

where $K_{p,k,i} = FP_{k,i|i-1}(H_k^{\mathrm{loc}})^* R_{e,k,i}^{-1}$. Now, let $P_k^-$ denote the unique stabilizing solution of the discrete-time algebraic Riccati equation (DARE)

$$P_k^- = FP_k^-F^* + GQG^* - K_{p,k}R_{e,k}K_{p,k}^* \tag{35}$$

where $K_{p,k} = FP_k^-(H_k^{\mathrm{loc}})^*R_{e,k}^{-1}$ and $R_{e,k} = R_k^{\mathrm{loc}} + H_k^{\mathrm{loc}}P_k^-(H_k^{\mathrm{loc}})^*$. Furthermore, we define

$$P_k = P_k^- - P_k^-\left(H_k^{\mathrm{loc}}\right)^*\left[R_k^{\mathrm{loc}} + H_k^{\mathrm{loc}}P_k^-\left(H_k^{\mathrm{loc}}\right)^*\right]^{-1}H_k^{\mathrm{loc}}P_k^-. \tag{36}$$

*Assumption 2:* The pair $\{F, H_k^{\mathrm{loc}}\}$ is detectable for every $k$, i.e., there exists a matrix $L$ such that $F - LH_k^{\mathrm{loc}}$ is stable (all of its eigenvalues lie inside the unit circle). Moreover, $\{F, GQ^{1/2}\}$ is stabilizable, i.e., there exists a matrix $K$ such that $F - GQ^{1/2}K$ is stable as well.

Assumption 2 guarantees the existence of $P_k^-$. Moreover, the matrices $P_{k,i+1|i}$ will converge to $P_k^-$ for any initial condition $\Pi_0 \geq 0$ as $i\to\infty$ [18].[2] Assumption 2 also guarantees that $R_{e,k,i} > 0$ for all $i$ and $k$. Moreover, $P_{k,i|i}$ will also converge, for all $k$, to the matrix $P_k$ in (36).

Under Assumptions 1 and 2, the matrices $\mathcal{P}_{i|i}$, $\mathcal{F}_i$, $\mathcal{G}_i$ and $\mathcal{D}_i$ also converge in steady-state to

$$\begin{aligned}\mathcal{P} &\triangleq \lim_{i\to\infty}\mathcal{P}_{i|i} = \mathrm{diag}\{P_1,\ldots,P_N\}\\ \mathcal{F} &\triangleq \lim_{i\to\infty}\mathcal{F}_i = \mathcal{C}^T(I-\mathcal{P}\mathcal{S})(I\otimes F)\\ \mathcal{G} &\triangleq \lim_{i\to\infty}\mathcal{G}_i = \mathcal{C}^T(I-\mathcal{P}\mathcal{S})(I\otimes G)\\ \mathcal{D} &\triangleq \lim_{i\to\infty}\mathcal{D}_i = \mathcal{C}^T\mathcal{P}\mathcal{A}^T\mathcal{H}^*R^{-1}\end{aligned} \tag{37}$$

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

Assumptions 1 and 2 are sufficient to guarantee the convergence of the diffusion KF algorithm. Specifically, we show later in Section IV-F that the matrix $\mathcal{F}$ in (37) is stable, and that (32) converges to the unique solution of the Lyapunov equation

$$\mathcal{P}_{\tilde{\chi}} = \mathcal{F}\mathcal{P}_{\tilde{\chi}}\mathcal{F}^* + \mathcal{G}(\mathbb{1}\mathbb{1}^T\otimes Q)\mathcal{G}^* + \mathcal{D}R\mathcal{D}^*. \tag{38}$$

Now we can solve for the steady-state covariance matrix of the estimation errors of the diffusion KF algorithm, $\mathcal{P}_{\tilde{\chi}}$. The solution may be expressed using the *vec* operator, which vectorizes a matrix by stacking its columns, and by using the property that $\mathrm{vec}(P\Sigma Q) = (Q^T\otimes P)\mathrm{vec}(\Sigma)$. In this case, we obtain from (38) that

$$\boxed{\mathrm{vec}(\mathcal{P}_{\tilde{\chi}}) = (I-\mathcal{F}^{*T}\otimes\mathcal{F})^{-1}\mathrm{vec}\left[\mathcal{G}(\mathbb{1}\mathbb{1}^T\otimes Q)\mathcal{G}^* + \mathcal{D}R\mathcal{D}^*\right]} \tag{39}$$

and we can recover $\mathcal{P}_{\tilde{\chi}}$ from $\mathrm{vec}(\mathcal{P}_{\tilde{\chi}})$. Note that since $\mathcal{F}$ is stable, the matrix $I - \mathcal{F}^{*T}\otimes\mathcal{F}$ is non-singular.

The steady-state MSD at node $k$ may now be expressed as

$$\boxed{\mathrm{MSD}_k = \lim_{i\to\infty}\mathrm{E}\|x_i - \hat{x}_{k,i|i}\|^2 = \mathrm{Tr}(\mathcal{P}_{\tilde{\chi}}\mathcal{I}_k)} \tag{40}$$

where $\mathcal{I}_k$ is an $NM\times NM$ block matrix with blocks of size $M\times M$, with an identity matrix at block $(k,k)$ and zeros elsewhere. Finally, the average steady-state MSD across the network is

$$\boxed{\mathrm{MSD}^{\mathrm{ave}} = \tfrac{1}{N}\mathrm{Tr}(\mathcal{P}_{\tilde{\chi}}).} \tag{41}$$

[2] A weaker condition that does not require stabilizability can be found in [18] under some restrictions on $\Pi_0$.

### E. Mean-Square Performance of the Diffusion Smoother

The diffusion smoother satisfies the following recursion for $i > i_0$:

$$\begin{aligned}\tilde{\chi}_{i_0|i} &= \tilde{\chi}_{i_0|i-1} + M_{i_0,i}[\tilde{\chi}_{i|i} - \tilde{\chi}_{i|i-1}] \\ &= \tilde{\chi}_{i_0|i_0} + \sum_{j=i_0+1}^{i} M_{i_0,j}[\tilde{\chi}_{j|j} - \tilde{\chi}_{j|j-1}]. \end{aligned} \quad (42)$$

Moreover, from (26) and (31), we have

$$\tilde{\chi}_{i|i} - \tilde{\chi}_{i|i-1} = \mathcal{F}'_i \tilde{\chi}_{i-1|i-1} + \mathcal{G}'_i(\mathbb{1} \otimes n_{i-1}) - \mathcal{D}_i v_i \quad (43)$$

where

$$\mathcal{F}'_i \triangleq \mathcal{F}_i - (I \otimes F_{i-1}) \quad \text{and} \quad \mathcal{G}'_i \triangleq \mathcal{G}_i - (I \otimes G_{i-1}).$$

Replacing (43) into (42), we obtain

$$\begin{aligned}\tilde{\chi}_{i_0|i} = \tilde{\chi}_{i_0|i_0} + \sum_{j=i_0+1}^{i} M_{i_0,j} \\ \times \left[\mathcal{F}'_j \tilde{\chi}_{j-1|j-1} + \mathcal{G}'_j(\mathbb{1} \otimes n_{j-1}) - \mathcal{D}_j v_j\right]. \end{aligned} \quad (44)$$

Iterating (31) and defining $\mathcal{F}_{i:j} \triangleq \mathcal{F}_i \mathcal{F}_{i-1} \ldots \mathcal{F}_j$ with $\mathcal{F}_{i-1:i} \triangleq I$, we obtain

$$\tilde{\chi}_{i|i} = \mathcal{F}_{i:i_0+1}\tilde{\chi}_{i_0|i_0} + \sum_{k=i_0+1}^{i} \mathcal{F}_{i:k+1}\left[\mathcal{G}_k(\mathbb{1} \otimes n_{k-1}) - \mathcal{D}_k v_k\right]. \quad (45)$$

Replacing (45) into (44), we obtain

$$\tilde{\chi}_{i_0|i} = \mathcal{F}''_i \tilde{\chi}_{i_0|i_0} + \sum_{j=i_0+1}^{i}\left[\mathcal{G}''_{j,i}(\mathbb{1} \otimes n_{j-1}) - \mathcal{D}''_{j,i} v_j\right] \quad (46)$$

where

$$\begin{aligned}\mathcal{F}''_i &= I + \sum_{j=i_0+1}^{i} M_{i_0,j}\mathcal{F}'_j \mathcal{F}_{j-1:i_0+1} \\ \mathcal{G}''_{j,i} &= M_{i_0,j}\mathcal{G}'_j + \sum_{k=j+1}^{i} M_{i_0,k}\mathcal{F}'_k \mathcal{F}_{k-1:j+1}\mathcal{G}_j \quad (47) \\ \mathcal{D}''_{j,i} &= M_{i_0,j}\mathcal{D}_j + \sum_{k=j+1}^{i} M_{i_0,k}\mathcal{F}'_k \mathcal{F}_{k-1:j+1}\mathcal{D}_j. \quad (48)\end{aligned}$$

In order to obtain (47) and (48), we used the fact that for any $\{X_{j,k}\}$

$$\sum_{j=i_0+1}^{i}\sum_{k=i_0+1}^{j-1} X_{j,k} = \sum_{j=i_0+1}^{i-1}\sum_{k=j+1}^{i} X_{k,j}.$$

Finally, from (46) we have

$$\begin{aligned}\mathrm{E}\left[\tilde{\chi}_{i_0|i}\tilde{\chi}^*_{i_0|i}\right] &= \mathcal{F}''_i \mathcal{P}_{\tilde{\chi},i_0}\left(\mathcal{F}''_i\right)^* + \sum_{j=i_0+1}^{i}\mathcal{D}''_{j,i}(R_j)\left(\mathcal{D}''_{j,i}\right)^* \\ &\quad + \sum_{j=i_0+1}^{i}\mathcal{G}''_{j,i}(\mathbb{1}\mathbb{1}^T \otimes Q_{j-1})\left(\mathcal{G}''_{j,i}\right)^*. \end{aligned} \quad (49)$$

Equation (49) allows us to calculate the covariance of the estimation error at time $i_0$ given observations up to time $i$, where $i \geq i_0$. For the fixed-point smoother, we need to compute $\mathcal{P}_{\tilde{\chi},i_0}$, and calculate (49) for a sufficiently large value of $i$. The initial covariance can be found by iterating (32), starting from

$$\begin{aligned}\mathcal{P}_{\tilde{\chi},0} = \mathcal{C}^T\mathcal{P}_{0|0}\Big[\mathcal{P}^{-1}_{0|-1}\mathrm{E}\left[\tilde{\chi}_{0|-1}\tilde{\chi}^*_{0|-1}\right]\mathcal{P}^{-1}_{0|-1} \\ +\mathcal{A}^T\mathcal{H}^*_0 R_0^{-1}\mathcal{H}_0\mathcal{A}\Big]\mathcal{P}_{0|0}\mathcal{C}\end{aligned}$$

and

$$\mathrm{E}\left[\tilde{\chi}_{0|-1}\tilde{\chi}^*_{0|-1}\right] = (\mathbb{1}\mathbb{1}^T) \otimes \Pi_0.$$

For the case of the fixed-lag smoother, we can also compute the steady-state estimation error from (49), by setting $i_0 = i - L$. In steady-state, $\mathcal{P}_{\tilde{\chi},i-L}$ is the steady-state error of the diffusion Kalman filter, which is found from (39). Then we can compute $\mathrm{E}[\tilde{\chi}_{i-L|i}\tilde{\chi}^*_{i-L|i}]$ from (49).

### F. Convergence Analysis

We start by showing that recursion (32) converges to the solution of the Lyapunov equation (38).

*Theorem 1:* Consider a recursion of the form

$$X_{i+1} = A_i X_i A_i^* + B_i \quad (50)$$

where $A_i$ and $B_i$ converge uniformly to $A$ and $B$, respectively, as $i \to \infty$, and where $A$ is a stable matrix. Then, $X_i$ converges to $X$, the solution of the Lyapunov equation

$$X = AXA^* + B. \quad (51)$$

*Proof:* See Appendix E.□

We now proceed to show that the matrix $\mathcal{F}$ in (37) is stable. We begin by rewriting $\mathcal{F} = \mathcal{C}^T\mathcal{M}$ where

$$\mathcal{M} = \mathrm{diag}\left\{(I - P_1 S_1)F, \ldots, (I - P_N S_N)F\right\}. \quad (52)$$

*Lemma 1:* The matrix $\mathcal{M}$ in (52) is stable under Assumptions 1 and 2.

*Proof:* We show that all the $M \times M$ blocks of $\mathcal{M}$ are stable. Consider block $k$, given by $\mathcal{M}_k = (I - P_k S_k)F$. We know from Assumption 2 that for every neighborhood, $P_{k,i|i-1}$ converges to the unique stabilizing solution $P_k^-$ of the DARE (35). That is, $P_k^-$ is such that the matrix $F_{p,k} = F - K_{p,k}H_k^{\mathrm{loc}}$ is stable for all $k$. We have

$$F_{p,k} = F\left[I - P_k^-\left(H_k^{\mathrm{loc}}\right)^* R_{e,k}^{-1} H_k^{\mathrm{loc}}\right].$$

Moreover, since $S_k = (H_k^{\mathrm{loc}})^*(R_k^{\mathrm{loc}})^{-1}H_k^{\mathrm{loc}}$ and using (36), we have

$$\begin{aligned}I - P_k S_k &= I - P_k^- S_k + P_k^-\left(H_k^{\mathrm{loc}}\right)^* R_{e,k}^{-1} H_k^{\mathrm{loc}} P_k^- S_k \\ &= I - P_k^-\left(H_k^{\mathrm{loc}}\right)^* R_{e,k}^{-1} H_k^{\mathrm{loc}}.\end{aligned}$$

Thus, we have $F_{p,k} = F(I - P_k S_k)$ and $\mathcal{M}_k = (I - P_k S_k)F$. Since $F_{p,k}$ is stable, we know that $\lim_{i\to\infty}(F_{p,k})^i = 0$. Moreover, since $(\mathcal{M}_k)^i = (I - P_k S_k)(F_{p,k})^{i-1}F$, we conclude that $\lim_{i\to\infty}(\mathcal{M}_k)^i = 0$ and therefore $\mathcal{M}_k$ is also stable. □

*Lemma 2:* The matrix $\mathcal{F}$ in (37) is stable under Assumptions 1 and 2.

*Proof:* A known result in matrix theory [18, p.554] states that for every square matrix $\mathcal{M}$, there exists a sub-multiplicative matrix norm[3] $\|.\|_\rho$ such that $\|\mathcal{M}\|_\rho \le \rho(\mathcal{M}) + \epsilon$, where $\rho(\mathcal{M})$ denotes the spectral radius of $\mathcal{M}$, i.e., the maximum absolute eigenvalue

$$\rho(\mathcal{M}) = \max_i |\lambda_i(\mathcal{M})|\,.$$

Since $\mathcal{M}$ is a stable matrix (from Lemma 1), let $\rho(\mathcal{M}) = \gamma < 1$. Then we can choose $\epsilon > 0$ such that $\gamma + \epsilon = \lambda < 1$ and $\|\mathcal{M}\|_\rho \le \lambda < 1$. Taking norms of $\mathcal{F}^i$, we have

$$\|\mathcal{F}^i\|_\rho \le \left\|\left(\mathcal{C}^T\right)^i\right\|_\rho \cdot \|\mathcal{M}^i\|_\rho \le \lambda^i \left\|\left(\mathcal{C}^T\right)^i\right\|_\rho.$$

Now, since $(\mathcal{C}^T)^i$ has non-negative entries with rows that add up to one (i.e., $(\mathcal{C}^T)^i \mathbb{1} = \mathbb{1}$), $(\mathcal{C}^T)^i$ is element-wise bounded by unity. This implies that the Frobenius norm of $(\mathcal{C}^T)^i$ is bounded, and by the equivalence of norms, so is any norm, and in particular $\|(\mathcal{C}^T)^i\|_\rho$. Thus, we have

$$\lim_{i\to\infty} \|\mathcal{F}^i\|_\rho = 0$$

so $\mathcal{F}^i$ converges to the zero matrix for large $i$. Consider the Jordan canonical decomposition of $\mathcal{F}$, namely $\mathcal{F} = TJT^{-1}$. Then, $J^i$ also converges to zero for large $i$, which can only be true if all the eigenvalues of $\mathcal{F}$ lie inside the unit circle. Then $\mathcal{F}$ is stable. □

We summarize our results with the following Theorem.

*Theorem 2:* Under Assumptions 1 and 2, the diffusion KF algorithm (Algorithms 1 or 2) is unbiased and converges, and the steady-state mean-square deviation for every node is given by (40).

*Proof:* The unbiasedness follows from Section IV-C. The convergence follows from Lemma 2 and the implication from Theorem 1 that $\mathcal{P}_{\tilde{\chi},i}$ in recursion (32) converges to the solution of (38). The steady-state mean-square deviation follows from the derivation of (40). □

## V. Simulations

In order to illustrate the performance of the diffusion Kalman filtering and smoothing algorithms, we present a simulation example in Figs. 3–9. We compare our simulation results with the theoretical values obtained in Section IV, and with the distributed solutions proposed in [11] and [12].

We consider the problem of estimating and tracking the position of a projectile. We assume that the projectile is in proximity of an adaptive network, where the sensors obtain noisy measurements of the position of the projectile. In the Kalman filtering problem, we are interested in estimating the exact position of the projectile at every time instant. In the fixed-lag smoothing problem, we use all the measurements up to the current time to estimate the position a fixed number of time instants earlier, this way obtaining a better estimate than the one produced by the Kalman filter. In the fixed-point smoothing problem, we wish to estimate the initial position of the motion of the projectile.

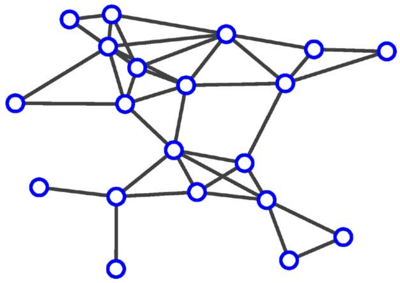

Fig. 3. Network topology with $N = 20$ nodes.

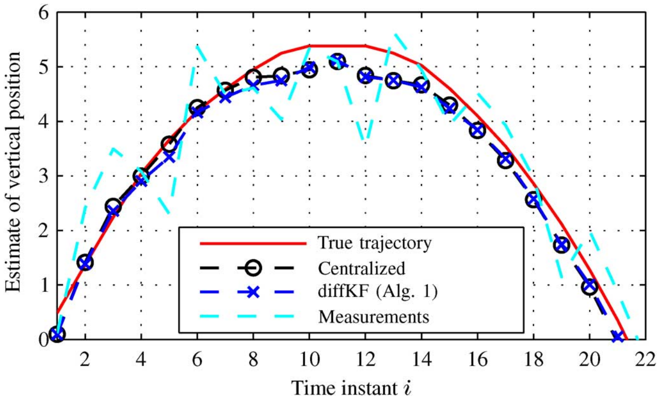

Fig. 4. Estimate of vertical position at node 1, for different algorithms.

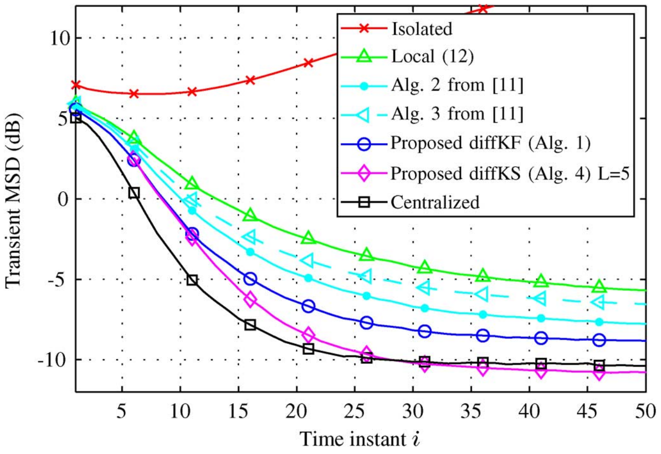

Fig. 5. Transient MSD performance of different algorithms.

In our example, the acceleration, velocity and position of the projectile, respectively, are

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \; v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \; d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

and where the subscripts 1, 2 and 3 correspond to the three spatial dimensions, 3 being the vertical one. For projectile motion, we have

$$a = \dot{v} \quad v = \dot{d} \quad a_1 = a_2 = 0 \quad a_3 = -g$$

[3] A sub-multiplicative matrix norm satisfies $\|AB\| \le \|A\| \cdot \|B\|$.

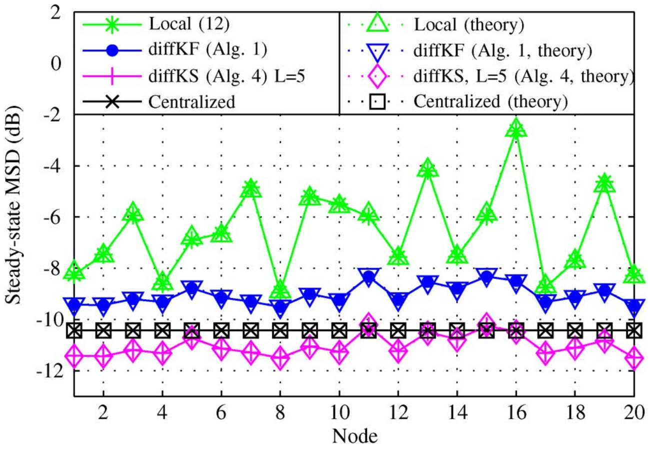

Fig. 6. Steady-state MSD performance of different algorithms.

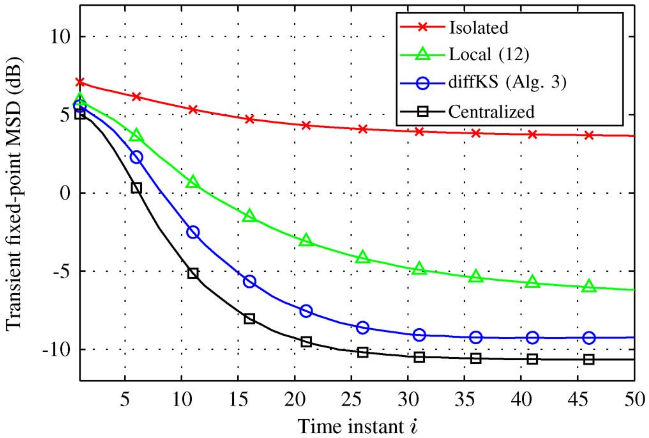

Fig. 7. Transient fixed-point MSD performance for different algorithms.

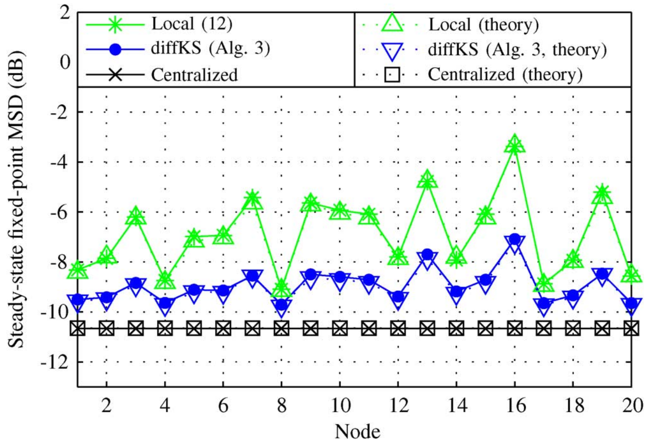

Fig. 8. Steady-state fixed-point MSD performance of different algorithms.

where $g$ is the gravity constant (we use $g = 10$). The state $x$ of the system is a vector of dimension 6, formed by stacking the velocity and position of the object, which evolves as follows:

$$\underbrace{\begin{bmatrix}\dot{v}\\ \dot{d}\end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix}0 & 0\\ I_3 & 0\end{bmatrix}}_{\Phi}\underbrace{\begin{bmatrix}v\\ d\end{bmatrix}}_{x} + \underbrace{\begin{bmatrix}\begin{bmatrix}0\\ 0\\ -g\end{bmatrix}\\ 0\end{bmatrix}}_{c}.$$

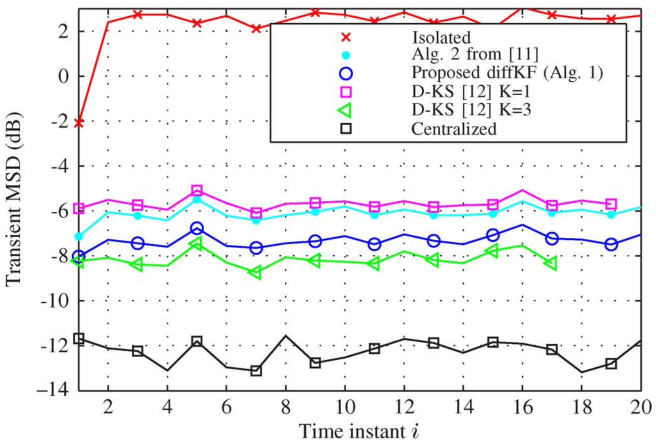

Fig. 9. Transient MSD performance of different algorithms.

For a system of the form $\dot{x} = \Phi x + c$, where $\Phi$ and $c$ are constants, we have

$$x(t+\delta) = e^{\Phi\delta}x(t) + \int_{t}^{t+\delta} e^{\Phi(t+\delta-\tau)}c d\tau.$$

Noting that for the $\Phi$ matrix above

$$e^{\Phi\delta} = I + \delta\Phi \quad \text{and} \quad \int_{t}^{t+\delta} e^{\Phi(t+\delta-\tau)}d\tau = \delta I + \delta^2\Phi/2$$

we conclude that the state satisfies the following recursion:

$$x(t+\delta) = [I + \delta\Phi]x(t) + [\delta I + \delta^2\Phi/2]c.$$

Given a time-step $\delta$, we now define

$$F \triangleq I + \delta\Phi \quad \text{and} \quad u \triangleq [\delta I + \delta^2\Phi/2]c.$$

We assume that every node measures the position of the unknown object in either the two horizontal dimensions, or a combination of one horizontal dimension and the vertical dimension. Thus, individual nodes do not have direct measurements of the position in the three dimensions. The assignment of which pair is observable by every node is done at random, but taking care that for every neighborhood, there is at least one node of each type (to guarantee detectability of the pair $\{F, H_k^{\text{loc}}\}$). Therefore, we have, $H_{k,i} = [0 \ \text{diag}([1\ 1\ 0])]$ for the case where only the horizontal dimensions are observed, or $H_{k,i} = [0 \ \text{diag}([1\ 0\ 1])]$ for the case where one horizontal dimension and the vertical dimension are observed.

Denoting $w_i = x(i\delta)$, and taking into account the effect of noisy states and measurements, we arrive at the discrete state-space model

$$\begin{aligned} w_{i+1} &= Fw_i + G_i n_i + u \\ z_{k,i} &= H_{k,i}w_i + v_{k,i} \end{aligned}$$

where $z_{k,i}$ are the individual measurements obtained by node $k$ at time $i$, $n_i$ accounts for modeling errors, and $v_{k,i}$ is the measurement noise at node $k$. Note that this model has the same form as the biased model presented in (22), and can therefore be reduced to a model of the form (24) as shown in Section III-F.

In our experiment we use a network with $N = 20$ nodes, with topology shown in Fig. 3. The values of the parameters are $\delta = 0.1$, $G_i = I$, $Q_i = (0.001)I$, $S_i = 0$ and $R_{k,i} = \sqrt{k}PR_0P^T$ with $R_0 = 0.5 \times \text{diag}[1\ 4\ 7]$ and $P$ being a permutation matrix, chosen at random for every node. The factor of $\sqrt{k}$ in $R_{k,i}$ allows us to consider nodes with diverse noise conditions. The expected value of the initial state is $\text{E}\, x_0 = [10\, 2\, 8\, 0.1\, 0.1\, 0.1]^T$ and its covariance $P_0 = I$. The diffusion matrix $C$ was chosen such that every neighbor is weighted according to the number of neighbors it has, as follows:

$$c_{lk} = \begin{cases} \alpha_k n_k & \text{if } l \in \mathcal{N}_k \\ 0 & \text{otherwise} \end{cases}$$

where $n_k$ is the degree of node $k$, and $\alpha_k$ is a normalization parameter chosen such that $\mathbb{1}^T C = \mathbb{1}^T$. The results were averaged over 1000 independent experiments over the same network topology.

Fig. 4 depicts the state estimation performance when estimating the vertical position, for different algorithms, at node 1. The solid curve corresponds to the true vertical trajectory. The dashed curve represents the noisy measurements of the vertical position obtained by node 1. The remaining two curves represent the estimates obtained by the centralized Kalman filter (black, circles) and those obtained by the diffusion Kalman filter (blue, crosses) of either Algorithm 1 or 2. We observe that the estimates produced by the diffusion Kalman filter algorithm are close to the true trajectory.

We now consider a more quantitative evaluation of the performance of the algorithms. We define the mean-square deviation (MSD) at node $k$ and at time $i$, given observations up to time $j$, as follows:

$$\text{MSD}_{k,i|j} = \text{E}\|x_i - \hat{x}_{k,i|j}\|^2.$$

Thus, when we refer to the MSD of a diffusion Kalman filter at time $i$, we are referring to $\text{MSD}_{k,i|i}$. When we refer to the MSD of a fixed-lag smoother with lag $L$ at time $i$, we are referring to $\text{MSD}_{k,i|i+L}$. Finally, when we refer to the MSD of a fixed-point smoother at time $i$, we are referring to $\text{MSD}_{k,i_0|i}$. The *network* MSD is defined as the average MSD over all nodes, that is

$$\text{MSD}^{\text{network}}_{i|j} = \frac{1}{N}\sum_{k=1}^{N} \text{MSD}_{k,i|j}.$$

Fig. 5 shows the transient network MSD for different algorithms as a function of time. The algorithm denoted “Isolated” corresponds to the case where the nodes do not cooperate with each other. In this case the estimation error is high, since nodes do not have access to measurements of the three coordinates of the moving projectile and the pair $\{F, H_k\}$ is not detectable. The algorithm denoted “Local” is computed assuming there is no diffusion process, but every node has access to the data of its neighbors as in (12). Therefore, every node would run a conventional Kalman filter using the data from its neighborhood. This algorithm is included for comparison, to evaluate the performance improvement introduced by the diffusion exchange. Also shown are Algorithms 2 and 3 from [11], which are consensus-based. The algorithms use $\epsilon = 0.1$. The algorithm denoted “diffKF” corresponds to our proposed diffusion Kalman filtering algorithm (Alg. 1 or Alg. 2), and the algorithm denoted “diffKS” corresponds to our proposed diffusion fixed-lag smoothing algorithm (Alg. 4), using a lag of $L = 5$. Finally, the algorithm denoted “Centralized” corresponds to a conventional Kalman filter that has access to all the data in the network. It can be observed from the plots that the diffusion KF algorithm improves over the “Local” and consensus-based algorithms by about 2–3 dB in this example. Moreover, the fixed-lag smoother using a lag of 5 time instants outperforms all other solutions in steady state (note that for the sake of clarity, we have omitted the centralized version of the fixed-lag smoother, which would outperform the diffusion version).

The steady-state expressions from Section IV are compared to the simulation results in Fig. 6, where we show the individual steady-state MSD for every node. The theoretical expression for the diffusion KF algorithm was obtained using (40), and the theoretical expression for the steady-state MSD of the fixed-lag smoother was obtained by employing recursion (49). Since both the local and centralized solutions are conventional Kalman filters (they differ in what data they can access), their MSDs can be obtained as the trace of the error covariance matrix of every node. In all cases, the expressions derived show good agreement with the simulation results.

Fig. 7 shows the transient performance of different algorithms for the fixed-point problem. We assume that $i_0 = 0$, and show results for $\text{MSD}^{\text{network}}_{i_0|i}$ as a function of time $i$. We observe that the diffusion Kalman smoother (Alg. 3) outperforms the local and isolated solutions, and is close to the centralized solution. Fig. 8 compares the steady-state performance with the theoretical expressions of Section IV.

Finally, we compare the performance of our algorithm to the consensus-based algorithm of [12]. In order to produce a fair comparison, and since our algorithm uses one diffusion step between measurements, we set $K = 1$ in [12], corresponding to one consensus iteration between measurements. Then, the state $x_i$ is estimated using (in the notation of [12]) $\hat{s}_j(i|i; i : i+1)$. For simplicity, we also consider a scalar state-space model as in [12], where $F = G = P_0 = 1$, $\text{E}\, x_0 = 0$, $R = 1.5$, $Q = 4$ and $H_k$ is drawn from a normal distribution. The network topology is the same as in Fig. 3, and bridge nodes were selected as outlined in [12]. Fig. 9 shows the transient network MSD for different algorithms as a function of time, averaged over 100 independent experiments. We observe that the diffusion KF (Alg. 1) outperforms [12] when $K = 1$. We also show for comparison the effect of increasing $K$ in [12]. We observe that when $K = 3$, [12] outperforms diffusion KF (Alg. 1). We should note that in this case, 3 times more information needs to be communicated compared to the case $K = 1$.

## VI. Conclusion

We presented diffusion Kalman filtering strategies for distributed state estimation in linear systems. We proposed algorithms for diffusion Kalman filtering, fixed-point smoothing and fixed-lag smoothing. The algorithms require every node to communicate only with its neighbors: first to share the data, and second to share the estimates, and the diffusion procedure ensures that information is propagated throughout the network. We analyzed the convergence of the algorithms and provided steady-state mean and mean-square analysis, showing good agreement with simulation results.

## APPENDIX

### A. Derivation of (5), (6) and (7)

Since $\hat{x}_{i+1|i} = F_i \hat{x}_{i|i}$, it holds that for $i_0 > i$

$$\hat{x}_{i_0|i} = F_{i_0-1} F_{i_0-2} \ldots F_i \hat{x}_{i|i}$$

where $\hat{x}_{i|i}$ is found from the Kalman filtering recursions described in the previous section. We now proceed to study the case $i > i_0$. Consider the so-called innovations [18] at time $j$, namely

$$e_j = y_j - \hat{y}_{j|j-1} = y_j - H_j \hat{x}_{j|j-1} = H_j \tilde{x}_{j|j-1} + v_j$$

and its covariance matrix

$$R_{e,j} = \mathrm{E} e_j e_j^* = H_j P_{j|j-1} H_j^* + R_j.$$

Then, from the orthogonality of the innovations we have [18, p.371]

$$\hat{x}_{i_0|i} = \sum_{j=0}^{i} \mathrm{E}\left[x_{i_0} e_j^*\right] R_{e,j}^{-1} e_j = \hat{x}_{i_0|i_0-1} + \sum_{j=i_0}^{i} \mathrm{E}\left[x_{i_0} e_j^*\right] R_{e,j}^{-1} e_j. \tag{53}$$

For $j \geq i_0$, it holds that $\mathrm{E}[x_{i_0} e_j^*] = P_{i_0,j} H_j^*$ where $P_{i,j} = \mathrm{E}\tilde{x}_{i|i-1}\tilde{x}_{j|j-1}^*$. Moreover

$$P_{i_0|i} = P_{i_0|i_0-1} - \sum_{j=i_0}^{i} P_{i_0,j} H_j^* R_{e,j}^{-1} H_j P_{i_0,j}^*. \tag{54}$$

In [18, p.373] it is shown that for the standard state-space model (1), and for $j \geq i$ we have

$$P_{i,j} = P_{i|i-1} \Phi_p^*(j,i)$$

where

$$\Phi_p(j,i) = \begin{cases} F_{p,j-1} F_{p,j-2} \ldots F_{p,i} & j > i \\ I & j = i \end{cases}$$
$$F_{p,i} = F_i - K_{p,i} H_i \qquad K_{p,i} = F_i P_{i|i-1} H_i^* R_{e,i}^{-1}. \tag{55}$$

Applying the matrix inversion lemma, and assuming $P_{i|i-1} > 0$ for all $i$, it is straightforward to show that

$$F_{p,i} = F_i P_{i|i} P_{i|i-1}^{-1}. \tag{56}$$

We now derive new useful recursive updates for the quantities $\hat{x}_{i_0|i}$ and $P_{i_0|i}$ appearing in (53) and (54). We start by defining the matrix

$$M_{i_0,i} \triangleq P_{i_0|i_0-1} \Phi_p^*(i,i_0) P_{i|i-1}^{-1}. \tag{57}$$

From (53) and (3) we have for $i \geq i_0$

$$\begin{aligned} \hat{x}_{i_0|i} &= \hat{x}_{i_0|i-1} + P_{i_0,i} H_i^* R_{e,i}^{-1} e_i \\ &= \hat{x}_{i_0|i-1} + P_{i_0|i_0-1} \Phi_p^*(i,i_0) H_i^* R_{e,i}^{-1} e_i \\ &= \hat{x}_{i_0|i-1} + M_{i_0,i} P_{i|i-1} H_i^* R_{e,i}^{-1} e_i \\ \hat{x}_{i|i} &= \hat{x}_{i|i-1} + P_{i|i-1} H_i^* R_{e,i}^{-1} e_i. \end{aligned}$$

The above two equations can be combined to obtain (5). Moreover, from (54) and (3) we have for $i \geq i_0$:

$$\begin{aligned} P_{i_0|i} &= P_{i_0|i-1} - P_{i_0,i} H_i^* R_{e,i}^{-1} H_i P_{i_0,i} \\ &= P_{i_0|i-1} - M_{i_0,i} P_{i|i-1} H_i^* R_{e,i}^{-1} H_i P_{i|i-1} M_{i_0,i}^* \\ P_{i|i} &= P_{i|i-1} - P_{i|i-1} H_i^* R_{e,i}^{-1} H_i P_{i|i-1}. \end{aligned}$$

The above two equations can be combined to obtain (6). We now need to compute a recursion for the matrix $M_{i_0,i}$. From (55), (56), and (57) we have

$$\begin{aligned} M_{i_0,i+1} &= P_{i_0|i_0-1} \Phi_p^*(i+1,i_0) P_{i+1|i}^{-1} \\ &= P_{i_0|i_0-1} \Phi_p^*(i,i_0) F_{p,i}^* P_{i+1|i}^{-1} \end{aligned}$$

so that (7) follows.

### B. Derivation of (13) from (12)

We show that the measurement update (12) can be computed using (13), provided $P_{k,i|i-1}$ is invertible. Let $\{k_1, \ldots, k_{n_k}\}$ denote the set of neighbors of node $k$, and let $\psi_{k,i}^{(m)}$ and $P_{k,i}^{(m)}$ denote the values of $\psi_{k,i}$ and $P_{k,i}$ after the $m$th iteration of (12). Then, the measurement update of (12) can be written

```
$\psi_{k,i}^{(0)} = \hat{x}_{k,i|i-1}$
$P_{k,i}^{(0)} = P_{k,i|i-1}$
for $m = 1$ to $n_k$, repeat :
    $R_e^{(m)} = R_{k_m,i} + H_{k_m,i} P_{k,i}^{(m-1)} H_{k_m,i}^*$
    $\psi_{k,i}^{(m)} = \psi_{k,i}^{(m-1)} + P_{k,i}^{(m-1)} H_{k_m,i}^*$
        $\times \left(R_e^{(m)}\right)^{-1} \left[y_{k_m,i} - H_{k_m,i} \psi_{k,i}^{(m-1)}\right]$
    $P_{k,i}^{(m)} = P_{k,i}^{(m-1)} - P_{k,i}^{(m-1)} H_{k_m,i}^*$
        $\times \left(R_e^{(m)}\right)^{-1} H_{k_m,i} P_{k,i}^{(m-1)}$
end
$\hat{x}_{k,i|i} = \psi_{k,i}^{(n_k)}$
$P_{k,i|i} = P_{k,i}^{(n_k)}.$
```

Now, using the matrix inversion lemma, we obtain

$$\left(P_{k,i}^{(m)}\right)^{-1} = \left(P_{k,i}^{(m-1)}\right)^{-1} + H_{k_m,i}^* R_{k_m,i}^{-1} H_{k_m,i} \tag{58}$$

which is well defined for all $m$ since $P_{k,i}^{(0)}$ is invertible by assumption. Iterating (58) we obtain the update for $P_{k,i|i}$ in (13), namely

$$P_{k,i|i}^{-1} = P_{k,i|i-1}^{-1} + \sum_{l \in \mathcal{N}_k} H_{l,i}^* R_{l,i}^{-1} H_{l,i}. \tag{59}$$

We also have that when $P_{k,i}^{(m-1)}$ is invertible

$$\begin{aligned}\psi_{k,i}^{(m)} &= \left[P_{k,i}^{(m-1)} - P_{k,i}^{(m-1)} H^*_{k_m,i}\left(R_e^{(m)}\right)^{-1} H_{k_m,i} P_{k,i}^{(m-1)}\right]\\ &\quad \times \left(P_{k,i}^{(m-1)}\right)^{-1}\psi_{k,i}^{(m-1)}\\ &\quad + P_{k,i}^{(m-1)} H^*_{k_m,i}\left(R_e^{(m)}\right)^{-1} y_{k_m,i}\\ &= P_{k,i}^{(m)}\left(P_{k,i}^{(m-1)}\right)^{-1}\psi_{k,i}^{(m-1)}\\ &\quad + P_{k,i}^{(m-1)} H^*_{k_m,i}\left(R_e^{(m)}\right)^{-1} y_{k_m,i}.\end{aligned}$$

Using the matrix inversion lemma, we obtain $P_{k,i}^{(m-1)} H^*_{k_m,i}(R_e^{(m)})^{-1} = P_{k,i}^{(m)} H^*_{k_m,i} R^{-1}_{k_m,i}$ and we conclude

$$\left(P_{k,i}^{(m)}\right)^{-1}\psi_{k,i}^{(m)} = \left(P_{k,i}^{(m-1)}\right)^{-1}\psi_{k,i}^{(m-1)} + H^*_{k_m,i} R^{-1}_{k_m,i} y_{k_m,i}. \tag{60}$$

Iterating (60), we arrive at:

$$(P_{k,i|i})^{-1}\hat{x}_{k,i|i} = (P_{k,i|i-1})^{-1}\hat{x}_{k,i|i-1} + \sum_{l\in\mathcal{N}_k} H^*_{l,i} R^{-1}_{l,i} y_{l,i}.$$

Using (59) and multiplying by $P_{k,i|i}$ we finally obtain the measurement update for $\hat{x}_{k,i|i}$ in (13), namely

$$\hat{x}_{k,i|i} = \hat{x}_{k,i|i-1} + P_{k,i|i}\sum_{l\in\mathcal{N}_k} H^*_{l,i} R^{-1}_{l,i}[y_{l,i} - H_{l,i}\hat{x}_{k,i|i-1}].$$

### *C. Invertibility of Adjacency Matrix*

In this section we show that given any $N \times N$ adjacency matrix $A$ as defined in (9), we can always make $A$ invertible by appropriately flipping some of its diagonal elements, where by flipping we mean that a zero becomes one and vice-versa. When $A$ is invertible, no flipping is needed. When $A$ is singular, we have $\det(A) = 0$. This determinant can be written as follows:

$$\begin{aligned}\det(A) &= \sum_{i=1}^{N-1}(-1)^{i+N}[A]_{i,N}\det(A_{i,N})\\ &\quad + [A]_{N,N}\det(A_{N,N})\\ &= 0\end{aligned}$$

where $A_{i,j}$ is the matrix obtained by removing row $i$ and column $j$ of $A$. Now, if $\det(A_{N,N}) \neq 0$, it is clear that by flipping the value of $[A]_{N,N}$, we have $\det(A) \neq 0$ and therefore $A$ would be invertible. For the case $\det(A_{N,N}) = 0$, we need to flip a diagonal element of $A_{N,N}$ that will make it non-singular. Thus, we can repeat the above procedure until one of the leading principal minors of $A$ is non-zero. Note that the first-order principal minor is $[A]_{1,1}$ which can always be made non-zero through the flipping operation.

### *D. Fusion of Local Kalman Estimates*

Depending on how much data is available at each node, we can consider three different Kalman filtering solutions, as follows. The individual estimate at node $k$, denoted by $\hat{x}^{\text{ind}}_{k,i|i}$, corresponds to the optimal linear estimate of $x_i$ given only the observations $y_{k,j}$ at node $k$ for $j = 0,\ldots,i$. The local estimate at node $k$, denoted by $\hat{x}^{\text{loc}}_{k,i|i}$, corresponds to the optimal linear estimate of $x_i$ given observations $y_{l,j}$ for $j = 0,\ldots,i$ across the neighborhood of $k$, i.e., $l \in \mathcal{N}_k$. Finally, we denote the global estimate by $\hat{x}_{i|i}$, which corresponds to the optimal linear estimate of $x_i$ given observations $y_{k,j}$ for $j = 0,\ldots,i$ *and* across all nodes $k = 1,\ldots,N$. In this section we study how the local estimates $\hat{x}^{\text{loc}}_{k,i|i}$ are related to the global solution. The covariance matrices for the individual, local and global estimation errors are denoted by $P^{\text{ind}}_{k,i|i}$, $P^{\text{loc}}_{k,i|i}$ and $P_{i|i}$, respectively.

Since the measurement noises at different nodes are assumed uncorrelated (and zero-mean), it can be shown (see [19, p. 89]) that the global and individual estimates are related via

$$\begin{aligned}P^{-1}_{i|i}\hat{x}_{i|i} &= \sum_{k=1}^{N}\left(P^{\text{ind}}_{k,i|i}\right)^{-1}\hat{x}^{\text{ind}}_{k,i|i}\\ P^{-1}_{i|i} &= \sum_{k=1}^{N}\left(P^{\text{ind}}_{k,i|i}\right)^{-1} - (N-1)\Pi_i^{-1}\end{aligned}$$

where $\Pi_i$ is the covariance matrix of $x_i$. The above expressions relate the local estimates to the individual estimates as follows:

$$\begin{aligned}\left(P^{\text{loc}}_{k,i|i}\right)^{-1}\hat{x}^{\text{loc}}_{k,i|i} &= \sum_{l=1}^{N}[A]_{l,k}\left(P^{\text{ind}}_{l,i|i}\right)^{-1}\hat{x}^{\text{ind}}_{l,i|i}\\ \left(P^{\text{loc}}_{k,i|i}\right)^{-1} &= \sum_{l=1}^{N}[A]_{l,k}\left(P^{\text{ind}}_{l,i|i}\right)^{-1} - \left(\sum_{l=1}^{N}[A]_{l,k} - 1\right)\Pi_i^{-1}\end{aligned}$$

where $[A]_{l,k}$ is given by (9), and it is unity if nodes $l$ and $k$ are neighbors and zero otherwise. Now consider a set of real weights $\gamma_k$, $k = 1,\ldots,N$, and the following combinations:

$$\begin{aligned}&\sum_{k=1}^{N}\gamma_k\left(P^{\text{loc}}_{k,i|i}\right)^{-1}\hat{x}^{\text{loc}}_{k,i|i}\\ &\quad = \sum_{l=1}^{N}\sum_{k=1}^{N}\gamma_k[A]_{l,k}\left(P^{\text{ind}}_{l,i|i}\right)^{-1}\hat{x}^{\text{ind}}_{l,i|i}\end{aligned} \tag{61}$$

$$\begin{aligned}&\sum_{k=1}^{N}\gamma_k\left(P^{\text{loc}}_{k,i|i}\right)^{-1}\\ &\quad = \sum_{l=1}^{N}\sum_{k=1}^{N}\gamma_k[A]_{l,k}\left(P^{\text{ind}}_{l,i|i}\right)^{-1}\\ &\qquad - \left(\sum_{l=1}^{N}\sum_{k=1}^{N}\gamma_k[A]_{l,k} - \sum_{k=1}^{N}\gamma_k\right)\Pi_i^{-1}.\end{aligned} \tag{62}$$

If we can find a set of weights such that $\sum_{k=1}^{N}\gamma_k[A]_{l,k} = 1$ for all $l$, and using the notation $\hat{x}^{\text{loc}}_{k,i|i} = \psi_{k,i}$ we have that (61) reduces to (15), and that (62) implies (16).

### *E. Proof of Theorem 1*

A result in linear algebra states that for every matrix $A$, there exists a sub-multiplicative matrix norm $\|.\|_\rho$ such that $\|A\|_\rho \leq \rho(A) + \epsilon$, where $\rho(A)$ is the maximum absolute eigenvalue of $A$ [18, p.554]. Since $A$ is stable, let $\rho(A) = \gamma < 1$, and we can always choose $\epsilon$ such that $\gamma + \epsilon = \lambda < 1$, and then $\|A\|_\rho \leq \lambda <$

1. Also, from the equivalence of norms, $\lim_{i\to\infty}\|A - A_i\|_\rho = \lim_{i\to\infty}\|B - B_i\|_\rho = 0$. This implies that $\forall\epsilon > 0$, $\exists i_0$ such that $\forall i \geq i_0$, $A_i = A + \epsilon Q_{A,i}$ and $B_i = B + \epsilon Q_{B,i}$, where $\|Q_{A,i}\|_\rho \leq 1$ and $\|Q_{B,i}\|_\rho \leq 1$.

Now, subtracting (51) from (50), we have for $i \geq i_0$

$$\begin{aligned}\|X_{i+1} - X\|_\rho = \big\| & A(X_i - X)A^* + \epsilon Q_{A,i}X_iA^* + \epsilon AX_iQ^*_{A,i} \\ & + \epsilon^2 Q_{A,i}X_iQ^*_{A,i} + \epsilon Q_{B,i}\big\|_\rho. \qquad (63)\end{aligned}$$

From the triangle inequality, we obtain

$$\|X_{i+1} - X\|_\rho \leq \lambda^2\|X_i - X\|_\rho + 2\epsilon\lambda\|X_i\|_\rho + \epsilon^2\|X_i\|_\rho + \epsilon.$$

We also have $\|X_i\| \leq \|X_i - X\| + \|X\|$, and therefore

$$\|X_{i+1} - X\|_\rho \leq (\lambda+\epsilon)^2\|X_i - X\|_\rho + \underbrace{\epsilon + 2\epsilon\lambda\|X\|_\rho + \epsilon^2\|X\|_\rho}_{\epsilon_c}.$$

Iterating the previous recursion for $i \geq i_0$, we obtain

$$\|X_{i+1} - X\|_\rho \leq (\lambda+\epsilon)^{2(i-i_1)}\,\|X_{i_1} - X\|_\rho + \epsilon_c\sum_{j=0}^{i-i_1}(\lambda+\epsilon)^{2j}.$$

Since $\lambda = \gamma + \epsilon < 1$, we can always choose $\epsilon$ such that $\lambda + \epsilon < 1$. In this case, we obtain

$$\lim_{i\to\infty}\|X_{i+1} - X\|_\rho \leq \frac{\epsilon_c}{1 - (\lambda+\epsilon)^2} \triangleq \epsilon_d.$$

Since we can make $\epsilon_d$ arbitrarily small by reducing $\epsilon$, we conclude that $X_{i+1}$ converges to $X$ as $i \to \infty$.

## Acknowledgment

The authors would like to thank C. G. Lopes for his useful advice during the earlier stages of this work.

## References

[1] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering: Formulation and performance analysis," in *Proc. Cognitive Inform. Processing*, Santorini, Greece, Jun. 2008, [CD ROM].

[2] F. S. Cattivelli and A. H. Sayed, "Diffusion mechanisms for fixed-point distributed Kalman smoothing," in *Proc. EUSIPCO*, Lausanne, Switzerland, Aug. 2008, [CD ROM].

[3] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. ICASSP*, Salt Lake City, UT, May 2001, vol. 4, pp. 2033–2036.

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

[5] B. S. Rao and H. F. Durrant-Whyte, "Fully decentralised algorithm for multisensor Kalman filtering," *Proc. Inst. Elect. Eng. D*, vol. 138, no. 5, pp. 413–420, Sep. 1991.

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

[7] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Trans. Autom. Control*, vol. AC-33, no. 1, pp. 88–94, Jan. 1988.

[8] P. Alriksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging," in *Proc. 17th Int. Symp. Math. Theory Networks Syst.*, Kyoto, Japan, Jul. 2006, [CD ROM].

[9] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Approximate distributed Kalman filtering in sensor networks with quantifiable performance," in *Proc. IPSN*, Los Angeles, CA, Apr. 2005, pp. 133–139.

[10] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. 44th IEEE Conf. Decision Control*, Sevilla, Spain, Dec. 2005, pp. 8179–8184.

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

[12] I. D. Schizas, G. B. Giannakis, S. I. Roumeliotis, and A. Ribeiro, "Consensus in ad hoc WSNs with noisy links—Part ii: Distributed estimation and smoothing of random signals," *IEEE Trans. Signal Processing*, vol. 56, no. 4, pp. 1650–1666, Apr. 2008.

[13] U. A. Khan and J. M. F. Moura, "Distributing the Kalman filter for large-scale systems," *IEEE Trans. Signal Processing*, vol. 56, no. 10, pt. 1, pp. 4919–4935, Oct. 2008.

[14] R. Carli, A. Chiuse, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE J. Selected Areas Commun.*, vol. 26, no. 4, pp. 622–633, May 2008.

[15] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "A diffusion RLS scheme for distributed estimation over adaptive networks," in *Proc. IEEE SPAWC*, Helsinki, Finland, Jun. 2007, pp. 1–5.

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

[17] A. H. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Mag.*, vol. 11, no. 3, pp. 18–60, Jul. 1994.

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

[19] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[20] A. H. Sayed, *Adaptive Filters*. New York: Wiley, 2008.

[21] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Trans. Fund. Electron., Commun. Comp. Sci.*, vol. E90-A, no. 8, pp. 1504–1510, Aug. 2007.

[22] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

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

[24] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. Joint 44th IEEE Conf. Decision Control Eur. Control Conf. (CDC-ECC)*, Seville, Spain, Dec. 2005, pp. 2996–3000.

[25] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. IPSN*, Los Angeles, CA, Apr. 2005, pp. 63–70.

[26] L. Xiao, S. Boyd, and S. J. Kim, "Distributed average consensus with least-mean-square deviation," in *Proc. 17th Int. Symp. Math. Theory Networks Syst. (MTNS)*, Kyoto, Japan, Jul. 2006, pp. 2768–2776.

[27] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," in *Proc. Adaptive Sensor Array Processing Workshop*, Lexington, MA, Jun. 2006, [CD ROM].

[28] F. Cattivelli and A. H. Sayed, "Self organization in bird flight formations using diffusion adaptation," in *Proc. CAMSAP*, Aruba, Dutch Antilles, Dec. 2009, pp. 49–52.

[29] C. G. Lopes and A. H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. Signal Processing*, vol. 55, no. 8, pp. 4064–4077, Aug. 2007.

[30] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Processing*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[31] Y. Zhang and J. A. Chambers, "Convex combination of adaptive filters for a variable tap-length LMS algorithm," *IEEE Signal Processing Lett.*, vol. 13, no. 10, pp. 628–631, Oct. 2006.

[32] M. Silva and V. H. Nascimento, "Convex combination of adaptive filters with different tracking capabilities," in *Proc. IEEE ICASSP*, Honolulu, HI, Apr. 2007, vol. 3, pp. 925–928.

[33] F. Cattivelli and A. H. Sayed, "Diffusion distributed Kalman filtering with adaptive weights," in *Proc. Asilomar Conf. Signals, Syst. Comp.*, Pacific Grove, CA, Nov. 2009, pp. 908–912.

**Federico S. Cattivelli** (S'03) received the B.S. degree in electronics engineering from Universidad ORT, Uruguay, in 2004 and the M.S. degree from the University of California, Los Angeles (UCLA), in 2006, where he is currently pursuing the Ph.D. degree in electrical engineering.

He is a member of the Adaptive Systems Laboratory at UCLA. His research interests are in signal processing, adaptive filtering, estimation theory and optimization. His current research activities are focused in the areas of distributed estimation and detection over adaptive networks. Previous research activities include Doppler shift frequency estimation for space applications (in collaboration with the NASA Jet Propulsion Laboratory), and state estimation and modeling in biomedical systems (in collaboration with the UCLA School of Medicine).

Mr. Cattivelli received the UCLA Graduate Division and UCLA Henry Samueli SEAS Fellowships, in 2005 and 2006, respectively.

**Ali H. Sayed** (F'01) is Professor and Chairman of Electrical Engineering at the University of California, Los Angeles (UCLA), where he directs the Adaptive Systems Laboratory. He has published widely in the areas of adaptive systems, statistical signal processing, estimation theory, adaptive and cognitive networks, and signal processing for communications. He has authored or co-authored several books including *Adaptive Filters* (Wiley, 2008), *Fundamentals of Adaptive Filtering* (Wiley, 2003), and *Linear Estimation* (Prentice Hall, 2000).

Dr. Sayed has served on the editorial boards of several journals, including Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He received the 1996 IEEE Donald G. Fink Award, the 2003 Kuwait Prize, the 2005 Terman Award, and the 2002 Best Paper Award and 2005 Young Author Best Paper Award from the IEEE Signal Processing Society. He has served as a 2005 Distinguished Lecturer of the same society and as General Chairman of ICASSP 2008. He is serving as the Vice-President at Publications of the IEEE Signal Processing Society.