Distributed Kalman Filtering in Presence of Unknown Outer Network Actuations

Sayed Pouria Talebi and Stefan Werner

***Abstract*—This paper presents a fully distributed approach for tracking state vector sequences over sensor networks in presence of unknown actuations. The problem arises in large-scale systems where modeling the full dynamics becomes impractical. In this work, the network only considers a subsection of the overall system which it can detect while accounting other inputs as unknown actuations. First a centralized technique that can consolidate all the available observation information is introduced. Then, operations of this optimal centralized solution are decomposed in a manner to allow their implementation in a distributed fashion while allowing each agent to retain an estimate of both the state vector and unknown actuations. The filter is derived in both diffusion and consensus formulations. The diffusion formulation is intended as a cost-effective solution, while the consensus formulation trades implementation complexity for accuracy.**

***Index Terms*—Sensor networks, distributed Kalman filtering, consensus and diffusion filtering, unknown input estimation.**

I. Introduction

Intelligent multi-agent networks form an essential part of most modern surveillance and control systems [1]–[16]. This has made development of distributed filtering and optimization techniques an attractive topic among the signal processing, control, and machine learning communities [8,12]. The fundamental problem in this setting becomes that of tracking the state of a given dynamic system through observations made over a network of sensors [17]–[23]. Although initial works on distributed Kalman filtering date back to the late 1970s with the seminal works in [24]–[26], truly distributed Kalman filtering solutions started to appear with the introduction of consensus [27]–[29] and diffusion [30] frameworks for practical information fusion over networks. In essence, consensus (*cf.* diffusion) based distributed Kalman filtering approaches use local Kalman filters to obtain an intermediate estimate of the state vector based on local observations, which are subsequently fed to consensus (*cf.* diffusion) filters to arrive at a final state vector estimate [20,21,31]–[34].

The proliferation of affordable sensor equipment that can facilitate communication and networking solutions has resulted in sensor networks becoming a viable solution for monitoring systems with an ever increasing degree of complexity. In some cases, the complexity of these systems is such that accurate derivation of the system model based on locally available information would be impractical. Moreover, the local observer might be only interested in tracking a subset of the state vector. One such example is the modern power grid that in some cases spans more than one country and/or regulatory jurisdiction. Modeling such a complex system would be impractical if not impossible. However, the power grid is often divided into more manageable sections, e.g., micro-grids. In this setting, the sensor network is mainly concerned with monitoring the micro-grid. Although the wide-area grid affects the micro-grid in question, partial information regarding the wide-area power grid will only be available to boundary agents.

Although a number of distributed Kalman filtering approaches for high-dimensional system have been presented [7, 35,36], these works either assume a fully connected network or assume a sparse structure for the state transition matrix and rely on elaborate distributed processing techniques, resulting in computationally demanding algorithms. On the other hand, the problem of filtering in presence of unknown inputs has been studied extensively [37]–[41]. However, these techniques are derived from a single agent perspective and are not suitable for decentralized implementation. Therefore, a truly distributed algorithm remains elusive.

This work considers the problem of tracking a state vector sequence in presence of unknown actuations through observations made over a sensor network. To this end, a centralized optimal filter that can incorporate the observations from all sensors in the network is formulated. Then, operations of the formulated centralized filter are decomposed and distributed among agents of the network in a fashion that will facilitate their distributed implementation. The proposed filter is formulated in both diffusion and consensus formats. Although both formulations allow each agent to retain an estimate of the state vector and the unknown actuations, the diffusion formulation is cost-effective as it does not impose computational demands much higher than that of the single agent filter, while the consensus formulation exchanges complexity for accuracy.

Mathematical Notations: Scalars, column vectors, and matrices are denoted by lowercase, bold lowercase, and bold uppercase letters. The state vector at time instant n is denoted by $\mathbf{x}_n$, while $\mathbf{I}$ represents the identity matrix with the same number of rows as the state vector. The Kronecker product is denoted by $\otimes$. The transpose operator is denoted by $(\cdot)^{\mathsf{T}}$ with $\mathbb{E}\left\{\cdot\right\}$ denoting the statistical expectation operator.

II. Problem Formulation

A. The Network & System Models

Akin to previous approaches [6,19]–[21,23,33], the multi-agent network is modeled as an undirected connected graph

Sayed Pouria Talebi is with the Department of Signal Processing and Acoustics, Aalto University, Espoo FI-02150 Finland, E-mail: pouria.talebi@aalto.fi. Stefan Werner is with the Department of Electronic Systems, Faculty of Information Technology and Electrical Engineering, Norwegian University of Science and Technology, Trondheim NO-7491 Norway, E-mail: stefan.werner@ntnu.no. This work was supported in part by Academy of Finland under Grant 296849 and Research Council of Norway.

$\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where the node set $\mathcal{N}$ denotes the agents of the network and the edge set $\mathcal{E}$ represents bidirectional communication links between the agents. The neighborhood of node l is defined as the set of nodes that can communicate with it, which includes self-communication. The neighborhood of node l is represented by the set $\mathcal{N}_l$ whose cardinality is denoted as $|\mathcal{N}_l|$ with $|\mathcal{N}|$ representing the total number of nodes in the network.

The aim is to track a state vector sequence through observations made via a network of agents (sensors). The state vector and observations are related through the state-space model[1]

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{B}\mathbf{z}_n + \boldsymbol{\nu}_n \tag{1a}$$

$$\mathbf{y}_{l,n} = \mathbf{H}_l\mathbf{x}_n + \boldsymbol{\omega}_{l,n} \tag{1b}$$

where, at time instant n, the vectors $\mathbf{z}_n$ and $\boldsymbol{\nu}_n$ denote the unknown inputs to the dynamic system and state evolution noise, while $\mathbf{A}$ and $\mathbf{B}$ denote the state transition matrix and a matrix of appropriate dimensions representing the effect of the unknown inputs on the system, whereas $\mathbf{H}_l$ denotes the observation matrix at node l, with $\mathbf{y}_{l,n}$ and $\boldsymbol{\omega}_{l,n}$ denoting the observation and observation noise at node l at time instant n. The observation and state evolution noises are white Gaussian random sequences with the joint covariance matrix

$$\mathbb{E}\left\{\begin{bmatrix}\boldsymbol{\nu}_n\\ \boldsymbol{\omega}_{l,n}\end{bmatrix}\begin{bmatrix}\boldsymbol{\nu}_m\\ \boldsymbol{\omega}_{k,m}\end{bmatrix}^\mathsf{T}\right\} = \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{\nu}} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}\delta_{l,k}\end{bmatrix}\delta_{n,m} \tag{2}$$

where $\delta_{l,k}$ is the Kronecker delta function.

B. Centralized Solution

In order to formulate a solution, observations across the network are organized into a column vector as

$$\mathbf{y}_{col,n} = \left[\mathbf{y}_{1,n}^\mathsf{T}, \ldots, \mathbf{y}_{|\mathcal{N}|,n}^\mathsf{T}\right]^\mathsf{T}. \tag{3}$$

Now, considering the expression in (1b) the network-wide observations can be modeled as

$$\mathbf{y}_{col,n} = \mathbf{H}_{col}\mathbf{x}_n + \boldsymbol{\omega}_{col}$$

with $\boldsymbol{\omega}_{col,n} = \left[\boldsymbol{\omega}_{1,n}^\mathsf{T}, \ldots, \boldsymbol{\omega}_{|\mathcal{N}|,n}^\mathsf{T}\right]^\mathsf{T}$ and

$$\mathbf{H}_{col} = \left[\mathbf{H}_1^\mathsf{T}, \ldots, \mathbf{H}_{|\mathcal{N}|}^\mathsf{T}\right]^\mathsf{T}. \tag{4}$$

The problem at hand can now be solved in a classical setting using a robust two-stage Kalman filter [40]. The operations of such a two-stage Kalman filter are summarized in Algorithm 1, where $\hat{\mathbf{z}}_n$ and $\hat{\mathbf{x}}_n$ denote the estimates of $\mathbf{z}_n$ and $\mathbf{x}_n$, with $\boldsymbol{\Sigma}_{\boldsymbol{\omega}_{col}} = \mathbb{E}\left\{\boldsymbol{\omega}_{col,n}\boldsymbol{\omega}_{col,n}^\mathsf{T}\right\}$. This assumes that the network-wide system, that is, the state transition equation in (1a) and observation equation in (3), meets the required convergence and stability criteria of conventional Kalman filtering in presence of unknown inputs, e.g., [40,41].[2]

Algorithm 1. Centralized Solution

Initialize with:

$$\hat{\mathbf{x}}_0 = \mathbb{E}\{\mathbf{x}_0\}$$
$$\mathbf{P}_0 = \mathbb{E}\left\{\left(\mathbf{x}_0 - \hat{\mathbf{x}}_{0|0}\right)\left(\mathbf{x}_0 - \hat{\mathbf{x}}_{0|0}\right)^\mathsf{T}\right\} \tag{5}$$

Model update:

$$\boldsymbol{\phi}_{n|n-1} = \mathbf{A}\hat{\mathbf{x}}_{n-1} \tag{6a}$$

$$\boldsymbol{\Phi}_{n|n-1} = \mathbf{A}\mathbf{P}_{n-1}\mathbf{A}^\mathsf{T} + \boldsymbol{\Sigma}_{\boldsymbol{\nu}} \tag{6b}$$

$$\mathbf{T}_n = \mathbf{H}_{col}\boldsymbol{\Phi}_{n|n-1}\mathbf{H}_{col}^\mathsf{T} + \boldsymbol{\Sigma}_{\boldsymbol{\omega}_{col}} \tag{6c}$$

Measurement update:

$$\boldsymbol{\phi}_{n|n} = \boldsymbol{\phi}_{n|n-1} + \boldsymbol{\Phi}_{n|n-1}\mathbf{H}_{col}^\mathsf{T}\mathbf{T}_n^{-1}\left(\mathbf{y}_{col,n} - \mathbf{H}_{col}\boldsymbol{\phi}_{n|n-1}\right) \tag{7a}$$

$$\boldsymbol{\Phi}_{n|n} = \left(\mathbf{I} - \boldsymbol{\Phi}_{n|n-1}\mathbf{H}_{col}^\mathsf{T}\mathbf{T}_n^{-1}\mathbf{H}_{col}\right)\boldsymbol{\Phi}_{n|n-1} \tag{7b}$$

$$\mathbf{S}_n = \mathbf{B}^\mathsf{T}\mathbf{H}_{col}^\mathsf{T}\mathbf{T}_n^{-1}\mathbf{H}_{col}\mathbf{B} \tag{7c}$$

$$\hat{\mathbf{z}}_n = \mathbf{S}_n^{-1}\mathbf{B}^\mathsf{T}\mathbf{H}_{col}^\mathsf{T}\mathbf{T}_n^{-1}\left(\mathbf{y}_{col,n} - \mathbf{H}_{col}\boldsymbol{\phi}_{n|n-1}\right) \tag{7d}$$

Combined update:

$$\mathbf{G}_n = \left(\mathbf{I} - \boldsymbol{\Phi}_{n|n-1}\mathbf{H}_{col}^\mathsf{T}\mathbf{T}_n^{-1}\mathbf{H}_{col}\right)\mathbf{B} \tag{8a}$$

$$\hat{\mathbf{x}}_n = \boldsymbol{\phi}_{n|n} + \mathbf{G}_n\hat{\mathbf{z}}_n \tag{8b}$$

$$\mathbf{P}_n = \boldsymbol{\Phi}_{n|n} + \mathbf{G}_n\mathbf{S}_n\mathbf{G}_n^\mathsf{T} \tag{8c}$$

C. Motivation for Distributed Solution

The centralized solution in Algorithm 1 requires all observations, observation matrices, and observation noise statistics, to be communicated to its central processing unit. This generates a great deal of communication traffic and leaves this approach vulnerable to link failures. Moreover, the centralized approach is also vulnerable to the failure of its processing unit and requires the inversion of large matrices, a computationally heavy operation which is best avoided. Hence, a distributed solution is desired.

III. The Proposed Distribute Solution

From the expression in (6c) and using the Woodbury matrix inversion lemma, we have

$$\mathbf{T}_n^{-1} = \boldsymbol{\Sigma}_{\boldsymbol{\omega}_{col}}^{-1} - \boldsymbol{\Sigma}_{\boldsymbol{\omega}_{col}}^{-1}\mathbf{H}_{col}\boldsymbol{\Psi}_n^{-1}\mathbf{H}_{col}^\mathsf{T}\boldsymbol{\Sigma}_{\boldsymbol{\omega}_{col}}^{-1} \tag{9}$$

where

$$\boldsymbol{\Psi}_n = \boldsymbol{\Phi}_{n|n-1}^{-1} + \mathbf{Q} \quad \text{and} \quad \mathbf{Q} = \mathbf{H}_{col}^\mathsf{T}\boldsymbol{\Sigma}_{\boldsymbol{\omega}_{col}}^{-1}\mathbf{H}_{col}. \tag{10}$$

In turn, using expression in (9), we have

$$\boldsymbol{\Pi}_n = \mathbf{H}_{col}^\mathsf{T}\mathbf{T}_n^{-1}\mathbf{H}_{col} = \mathbf{Q} - \mathbf{Q}\left(\boldsymbol{\Phi}_{n|n-1}^{-1} + \mathbf{Q}\right)^{-1}\mathbf{Q}. \tag{11}$$

Now, replacing (9) and (11) into (7c) gives

$$\mathbf{S}_n = \mathbf{B}^\mathsf{T}\boldsymbol{\Pi}_n\mathbf{B} = \mathbf{B}^\mathsf{T}\mathbf{Q}\mathbf{B} - \mathbf{B}^\mathsf{T}\mathbf{Q}\boldsymbol{\Psi}_n^{-1}\mathbf{Q}\mathbf{B}. \tag{12}$$

Upon substituting (9) into (7d), we have

$$\hat{\mathbf{z}}_n = \mathbf{S}_n^{-1}\mathbf{B}^\mathsf{T}\boldsymbol{\xi}_n - \mathbf{S}_n^{-1}\mathbf{B}^\mathsf{T}\mathbf{Q}\boldsymbol{\Psi}_n^{-1}\boldsymbol{\xi}_n \tag{13}$$

[1] The state-space model is considered to be linear and time invariant with stationary noise sequences for simplicity in presentation. However, the obtained results can be readily generalized.

[2] Derivation of the centralized solution given in Algortihm 1 and discussions on its optimality closely follow that of the framework presented in [37,40] and have, therefore, been omitted.

where

$$\boldsymbol{\xi}_n = \mathbf{H}_{col}^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_{col}}^{-1} \left(\mathbf{y}_{col,n} - \mathbf{H}_{col} \boldsymbol{\phi}_{n|n-1} \right). \tag{14}$$

In addition, from (2) it follows that $\boldsymbol{\Sigma}_{\boldsymbol{\omega}_{col}}$ is a block-diagonal matrix so that

$$\boldsymbol{\Sigma}_{\boldsymbol{\omega}_{col}} = \text{block-diag}\{\boldsymbol{\Sigma}_{\boldsymbol{\omega}_l} : \forall l \in \mathcal{N}\}. \tag{15}$$

After some mathematical manipulation and replacing (15) into (14), the expression in (13) can be expressed as the summation

$$\hat{\mathbf{z}}_n = \mathbf{S}_n^{-1} \mathbf{B}^{\mathsf{T}} \left(\mathbf{I} - \mathbf{Q} \boldsymbol{\Psi}_n^{-1} \right) \sum_{\forall l \in \mathcal{N}} \boldsymbol{\xi}_{l,n} \tag{16}$$

with

$$\begin{aligned} \boldsymbol{\xi}_{l,n} =& \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}^{-1} \left(\mathbf{y}_{l,n} - \mathbf{H}_l \boldsymbol{\phi}_{n|n-1} \right) \\ =& \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}^{-1} \mathbf{y}_{l,n} - \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}^{-1} \mathbf{H}_l \boldsymbol{\phi}_{n|n-1}. \end{aligned} \tag{17}$$

Alternatively, from (17) and (16) it follows that

$$\begin{aligned} \hat{\mathbf{z}}_n =& \mathbf{S}_n^{-1} \mathbf{B}^{\mathsf{T}} \left(\mathbf{I} - \mathbf{Q} \boldsymbol{\Psi}^{-1} \right) \sum_{\forall l \in \mathcal{N}} \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}^{-1} \mathbf{y}_{l,n} \\ & - \mathbf{S}_n^{-1} \mathbf{B}^{\mathsf{T}} \left(\mathbf{I} - \mathbf{Q} \boldsymbol{\Psi}_n^{-1} \right) \mathbf{Q} \boldsymbol{\phi}_{n|n-1}. \end{aligned} \tag{18}$$

In a similar fashion, replacing (9) and (11) into (7a) gives

$$\begin{aligned} \boldsymbol{\phi}_{n|n} =& \boldsymbol{\phi}_{n|n-1} + \sum_{\forall l \in \mathcal{N}} \boldsymbol{\Phi}_{n|n-1} \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l} \left(\mathbf{y}_{l,n} - \mathbf{H}_l \boldsymbol{\phi}_{n|n-1} \right) \\ & - \sum_{\forall l \in \mathcal{N}} \boldsymbol{\Phi}_{n|n-1} \mathbf{Q} \boldsymbol{\Psi}_n^{-1} \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}^{-1} \left(\mathbf{y}_{l,n} - \mathbf{H}_l \boldsymbol{\phi}_{n|n-1} \right) \\ =& \boldsymbol{\phi}_{n|n-1} + \boldsymbol{\Phi}_{n|n-1} \left(\mathbf{I} - \mathbf{Q} \boldsymbol{\Psi}_n^{-1} \right) \sum_{\forall l \in \mathcal{N}} \boldsymbol{\xi}_{l,n}. \end{aligned} \tag{19}$$

From (17) and (19), $\boldsymbol{\phi}_{n|n}$ can alternatively be formulated as

$$\begin{aligned} \boldsymbol{\phi}_{n|n} =& \boldsymbol{\phi}_{n|n-1} + \boldsymbol{\Phi}_{n|n-1} \left(\mathbf{I} - \mathbf{Q} \boldsymbol{\Psi}_n^{-1} \right) \sum_{\forall l \in \mathcal{N}} \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}^{-1} \mathbf{y}_{l,n} \\ & - \boldsymbol{\Phi}_{n|n-1} \left(\mathbf{I} - \mathbf{Q} \boldsymbol{\Psi}_n^{-1} \right) \mathbf{Q} \boldsymbol{\phi}_{n|n-1}. \end{aligned} \tag{20}$$

A. Diffusion Formulation

From replacing (15) and (4) into (10) we have

$$\mathbf{Q} = \sum_{\forall l \in \mathcal{N}} \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}^{-1} \mathbf{H}_l. \tag{21}$$

Thus, similar to approaches in [6,19,33], assuming node $l \in \mathcal{N}$ receives $\{\mathbf{H}_k^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_k} \mathbf{H}_k, \mathbf{H}_k^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_k} \mathbf{y}_{k,n} : \forall k \in \mathcal{N}_l\}$ from its neighbors, the expressions in (9)-(20) allow node l to run a local filtering operation.[3] This leaves each agent with a local estimate of the state vector, which can be combined in a diffusion setting to improve their accuracy. Operations of such a filter are summarized in Algorithm 2, where $\hat{\mathbf{x}}_{l,n}$ and $\hat{\mathbf{z}}_{l,n}$ denote the estimate of $\mathbf{x}_n$ and $\mathbf{z}_n$ at node l, while $\alpha_{l,k} \geqslant 0$ are real-valued diffusion coefficients selected so that

$$\forall l \in \mathcal{N} : \sum_{\forall k \in \mathcal{N}_l} \alpha_{l,k} = 1. \tag{22}$$

Algorithm 2. Diffusion-Based Distributed Solution

For node $l \in \mathcal{N}$:
Initialize with:

$$\begin{aligned} \hat{\mathbf{x}}_{l,0} &= \mathbb{E}\{\mathbf{x}_0\} \\ \mathbf{P}_{l,0} &= \mathbb{E}\left\{ \left(\mathbf{x}_0 - \hat{\mathbf{x}}_{l,0|0} \right) \left(\mathbf{x}_0 - \hat{\mathbf{x}}_{l,0|0} \right)^{\mathsf{T}} \right\} \end{aligned} \tag{23}$$

Model update:

$$\boldsymbol{\phi}_{l,n|n-1} = \mathbf{A} \hat{\mathbf{x}}_{l,n-1} \tag{24a}$$

$$\boldsymbol{\Phi}_{l,n|n-1} = \mathbf{A} \mathbf{P}_{l,n-1} \mathbf{A}^{\mathsf{T}} + \boldsymbol{\Sigma}_{\boldsymbol{\nu}} \tag{24b}$$

$$\mathbf{Q}_l = \sum_{\forall k \in \mathcal{N}_l} \mathbf{H}_k^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_k}^{-1} \mathbf{H}_k \tag{24c}$$

$$\boldsymbol{\Psi}_{l,n} = \boldsymbol{\Phi}_{l,n|n-1}^{-1} + \mathbf{Q}_l \tag{24d}$$

Measurement update:

$$\begin{aligned} \boldsymbol{\phi}_{l,n|n} =& \boldsymbol{\phi}_{l,n|n-1} \\ & + \boldsymbol{\Phi}_{l,n|n-1} \left(\mathbf{I} - \mathbf{Q}_l \boldsymbol{\Psi}_{l,n}^{-1} \right) \sum_{\forall k \in \mathcal{N}_l} \mathbf{H}_k^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_k}^{-1} \mathbf{y}_{k,n} \\ & - \boldsymbol{\Phi}_{l,n|n-1} \left(\mathbf{I} - \mathbf{Q}_l \boldsymbol{\Psi}_{l,n}^{-1} \right) \mathbf{Q}_l \boldsymbol{\phi}_{l,n|n-1} \end{aligned} \tag{25a}$$

$$\boldsymbol{\Pi}_{l,n} = \mathbf{Q}_l - \mathbf{Q}_l \left(\boldsymbol{\Phi}_{l,n|n-1}^{-1} + \mathbf{Q}_l \right)^{-1} \mathbf{Q}_l \tag{25b}$$

$$\boldsymbol{\Phi}_{l,n|n} = \left(\mathbf{I} - \boldsymbol{\Phi}_{n|n-1} \boldsymbol{\Pi}_{l,n} \right) \boldsymbol{\Phi}_{l,n|n-1} \tag{25c}$$

$$\mathbf{S}_{l,n} = \mathbf{B}^{\mathsf{T}} \boldsymbol{\Pi}_{l,n} \mathbf{B} \tag{25d}$$

$$\begin{aligned} \hat{\mathbf{z}}_{l,n} =& \mathbf{S}_{l,n}^{-1} \mathbf{B}^{\mathsf{T}} \left(\mathbf{I} - \mathbf{Q}_l \boldsymbol{\Psi}_{l,n}^{-1} \right) \sum_{\forall k \in \mathcal{N}_l} \mathbf{H}_k^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_k}^{-1} \mathbf{y}_{k,n} \\ & - \mathbf{S}_{l,n}^{-1} \mathbf{B}^{\mathsf{T}} \left(\mathbf{I} - \mathbf{Q}_l \boldsymbol{\Psi}_{l,n}^{-1} \right) \mathbf{Q}_l \boldsymbol{\phi}_{l,n|n-1} \end{aligned} \tag{25e}$$

Combined update:

$$\mathbf{G}_{l,n} = \left(\mathbf{I} - \boldsymbol{\Phi}_{l,n|n-1} \boldsymbol{\Pi}_{l,n} \right) \mathbf{B} \tag{26a}$$

$$\hat{\mathbf{x}}_{l,n} = \sum_{\forall k \in \mathcal{N}_l} \alpha_{l,k} \left(\boldsymbol{\phi}_{k,n|n} + \mathbf{G}_{k,n} \hat{\mathbf{z}}_{k,n} \right) \tag{26b}$$

$$\mathbf{P}_{l,n} = \boldsymbol{\Phi}_{l,n|n} + \mathbf{G}_{l,n} \mathbf{S}_{l,n} \mathbf{G}_{l,n}^{\mathsf{T}} \tag{26c}$$

B. Consensus Formulation

Considering decomposition of the operations of Algorithm 1 in (9)-(20) and recognizing that $\mathbf{Q}$ and $\sum_{\forall l \in \mathcal{N}} \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_l^{-1} \mathbf{y}_{l,n}$ are network summations of local quantities, the centralized solution (Algorithm 1) can be replicated at each agent using approximations of these summation. To this end, (21) yields

$$\mathbf{Q} = \sum_{\forall l \in \mathcal{N}} \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}^{-1} \mathbf{H}_l = \frac{1}{|\mathcal{N}|} \sum_{\forall l \in \mathcal{N}} |\mathcal{N}| \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_{\boldsymbol{\omega}_l}^{-1} \mathbf{H}_l. \tag{27}$$

In a similar manner we have

$$\sum_{\forall l \in \mathcal{N}} \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_l^{-1} \mathbf{y}_{l,n} = \frac{1}{|\mathcal{N}|} \sum_{\forall l \in \mathcal{N}} |\mathcal{N}| \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_l^{-1} \mathbf{y}_{l,n}. \tag{28}$$

Thus, $\mathbf{Q}$ and $\sum_{\forall l \in \mathcal{N}} \mathbf{H}_l^{\mathsf{T}} \boldsymbol{\Sigma}_l^{-1} \mathbf{y}_{l,n}$ can be estimated in a distributed fashion via the average consensus filter[4] (ACF)

$$\mathbf{F}_{i,(\eta)} = \mathbf{F}_{i,(\eta-1)} + \sum_{\forall j \in \mathcal{N}_i} w_{i,j} \left(\mathbf{F}_{j,(\eta-1)} - \mathbf{F}_{i,(\eta-1)} \right) \tag{29}$$

[3] That is, node l implements the operations in Algorithm 1 for the network consisted of itself and its neighbors. In this case, node l is at the center of a star network with its neighboring nodes forming the spikes. Also these local filtering operations are only reliant on single-hop communication links.

[4] For more information on consensus filters see [27]–[29].

where $\mathbf{F}_{i,(\eta)}$ denotes the output of the iterative consensus filter at node i after η iterations, while $w_{i,j}$ denotes a positive real-valued weight. For analysis purposes, the ACF in (29) is formulated from a network-wide perspective as

$$\boldsymbol{\mathcal{F}}_{(\eta)} = (\boldsymbol{\mathcal{W}} \otimes \boldsymbol{\mathcal{I}})\boldsymbol{\mathcal{F}}_{(\eta-1)} = (\boldsymbol{\mathcal{W}}^{\eta} \otimes \boldsymbol{\mathcal{I}})\boldsymbol{\mathcal{F}}_{(0)} \tag{30}$$

where $\boldsymbol{\mathcal{I}}$ is an identity matrix of appropriate size, whereas $\boldsymbol{\mathcal{F}}_{(\eta)} = [\mathbf{F}_{1,(\eta)}^{\mathsf{T}}, \mathbf{F}_{2,(\eta)}^{\mathsf{T}}, \ldots, \mathbf{F}_{|\mathcal{N}|,(\eta)}^{\mathsf{T}}]^{\mathsf{T}}$ and the element on the i^{th} row and j^{th} column of $\boldsymbol{\mathcal{W}}$ are

$$\boldsymbol{\mathcal{W}}_{i,j} = \begin{cases} 1 + w_{i,i} - \sum\limits_{\forall l \in \mathcal{N}_i} w_{i,l} & \text{if } i = j, \\ w_{i,j} & \text{if } i \in \mathcal{N}_j \backslash j \\ 0 & \text{otherwise.} \end{cases}$$

Remark 1. If the weights $w_{i,j}$ are selected to meet conditions in [27] and make $\boldsymbol{\mathcal{W}}$ doubly stochastic; then, from [27], it follows that $\mathbf{F}_{i,(\eta)} \to \frac{1}{|\mathcal{N}|}\sum_{\forall j \in \mathcal{N}} \mathbf{F}_{j,(0)}$ as $\eta \to \infty$.

From the expression in (27) and *Remark* 1, it follows that each agent can reach an estimate of $\mathbf{Q}$ using $\{|\mathcal{N}|\mathbf{H}_l^{\mathsf{T}}\boldsymbol{\Sigma}_l^{-1}\mathbf{H}_l : \forall l \in \mathcal{N}\}$ as inputs of the ACF in (29). In addition, from (28), the summation $\sum_{\forall l \in \mathcal{N}} \mathbf{H}_l^{\mathsf{T}}\boldsymbol{\Sigma}_l^{-1}\mathbf{y}_{l,n}$ can be approximated using $\{|\mathcal{N}|\mathbf{H}_l^{\mathsf{T}}\boldsymbol{\Sigma}_l^{-1}\mathbf{y}_{l,n} : \forall l \in \mathcal{N}\}$ as inputs of the ACF in (29). Therefore, each agent can replicate the operations of Algorithm 1 in a distributed manner. In practice, however, the ACF can only undergo a finite number of iterations at each time instant. This leaves each agent with an estimate of the state vector (*cf.* covariance information) that differs from that of its neighbors. Motivated by the desire to force an agreement among the agents, additional ACFs are employed to force a consensus among the agents regarding the state vector estimates (*cf.* covariance information). The operations of such a consensus-based distributed solution are summarized in Algorithm 3, where for the sake of simplicity in presentation, the operation of the ACF at node i after η iterations is represented via the schematic

$$\mathbf{F}_{i,(\eta)} \leftarrow \boxed{\text{ACF}} \leftarrow \{\mathbf{F}_{j,(0)} : \forall j \in \mathcal{N}\}$$

where $\{\mathbf{F}_{j,(0)} : \forall j \in \mathcal{N}\}$ is the network-wide inputs to the ACF and $\mathbf{F}_{i,(\eta)}$ is the output at node i after η iterations.

C. Convergence and Stability

Without loss of generality, we focus on the introduced diffusion-based filter. After some tedious mathematical manipulations, from Algorithm 2, we have

$$\begin{aligned} \boldsymbol{\phi}_{l,n|n} + \mathbf{G}_{l,n}\hat{\mathbf{z}}_{l,n} = & \left(\mathbf{I} - \mathbf{L}_n\mathbf{Q}_l\right)\mathbf{A}\hat{\mathbf{x}}_{l,n-1} \\ & + \mathbf{L}_{l,n} \sum_{\forall k \in \mathcal{N}_l} \mathbf{H}_k^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{\omega}_k}^{-1}\mathbf{y}_{k,n} \end{aligned} \tag{35}$$

where $\mathbf{L}_{l,n} = \left(\boldsymbol{\Phi}_{l,n|n-1} + \mathbf{G}_{l,n}\mathbf{S}_{l,n}^{-1}\mathbf{B}^{\mathsf{T}}\right)\left(\mathbf{I} - \mathbf{Q}_l\boldsymbol{\Psi}_{l,n}^{-1}\right)$. Given (24c), upon substituting (1) into the left hand side of (35), we have

$$\begin{aligned} \text{LHS(35)} = & \left(\mathbf{I} - \mathbf{L}_{l,n}\mathbf{Q}_l\right)\mathbf{A}\hat{\mathbf{x}}_{l,n-1} + \mathbf{L}_{l,n}\mathbf{Q}_l\mathbf{A}\mathbf{x}_{n-1} \\ & + \mathbf{L}_{l,n}\mathbf{Q}_l\mathbf{B}\mathbf{z}_{n-1} + \mathbf{L}_{l,n}\mathbf{Q}_l\boldsymbol{\nu}_{n-1} + \mathbf{L}_{l,n}\mathbf{r}_{l,n} \end{aligned} \tag{36}$$

where LHS(35) denotes the left hand side of the expression in (35) and $\mathbf{r}_{l,n} = \sum_{\forall k \in \mathcal{N}_l} \mathbf{H}_k^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{\omega}_k}^{-1}\boldsymbol{\omega}_{k,n}$.

Algorithm 3. Consensus-Based Distributed Solution

For node $l \in \mathcal{N}$:

Initialize with:

$$\begin{aligned} \hat{\mathbf{x}}_{l,0} &= \mathbb{E}\left\{\mathbf{x}_0\right\} \\ \mathbf{P}_{l,0} &= \mathbb{E}\left\{\left(\mathbf{x}_0 - \hat{\mathbf{x}}_{l,0|0}\right)\left(\mathbf{x}_0 - \hat{\mathbf{x}}_{l,0|0}\right)^{\mathsf{T}}\right\} \end{aligned} \tag{31}$$

Model update:

$$\boldsymbol{\phi}_{l,n|n-1} = \mathbf{A}\hat{\mathbf{x}}_{l,n-1} \tag{32a}$$

$$\boldsymbol{\Phi}_{l,n|n-1} = \mathbf{A}\mathbf{P}_{l,n-1}\mathbf{A}^{\mathsf{T}} + \boldsymbol{\Sigma}_{\boldsymbol{\nu}} \tag{32b}$$

$$\boldsymbol{\Upsilon}_{l,n} \leftarrow \boxed{\text{ACF}} \leftarrow \{|\mathcal{N}|\mathbf{H}_k^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{\omega}_k}^{-1}\mathbf{H}_k : \forall k \in \mathcal{N}\} \tag{32c}$$

$$\boldsymbol{\Psi}_{l,n} = \boldsymbol{\Phi}_{l,n|n-1}^{-1} + \boldsymbol{\Upsilon}_{l,n} \tag{32d}$$

Measurement update:

$$\boldsymbol{\varphi}_{l,n} \leftarrow \boxed{\text{ACF}} \leftarrow \{|\mathcal{N}|\mathbf{H}_k^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{\omega}_k}^{-1}\mathbf{y}_{k,n} : \forall k \in \mathcal{N}\} \tag{33a}$$

$$\begin{aligned} \boldsymbol{\phi}_{l,n|n} = & \boldsymbol{\phi}_{l,n|n-1} + \boldsymbol{\Phi}_{l,n|n-1}\left(\mathbf{I} - \boldsymbol{\Upsilon}_{l,n}\boldsymbol{\Psi}_{l,n}^{-1}\right)\boldsymbol{\varphi}_{l,n} \\ & - \boldsymbol{\Phi}_{l,n|n-1}\left(\mathbf{I} - \boldsymbol{\Upsilon}_{l,n}\boldsymbol{\Psi}_{l,n}^{-1}\right)\boldsymbol{\Upsilon}_{l,n}\boldsymbol{\phi}_{l,n|n-1} \end{aligned} \tag{33b}$$

$$\boldsymbol{\Pi}_{l,n} = \boldsymbol{\Upsilon}_{l,n} - \boldsymbol{\Upsilon}_{l,n}\left(\boldsymbol{\Phi}_{l,n|n-1}^{-1} + \boldsymbol{\Upsilon}_{l,n}\right)^{-1}\boldsymbol{\Upsilon}_{l,n} \tag{33c}$$

$$\boldsymbol{\Phi}_{l,n|n} = \left(\mathbf{I} - \boldsymbol{\Phi}_{n|n-1}\boldsymbol{\Pi}_{l,n}\right)\boldsymbol{\Phi}_{l,n|n-1} \tag{33d}$$

$$\mathbf{S}_{l,n} = \mathbf{B}^{\mathsf{T}}\boldsymbol{\Pi}_{l,n}\mathbf{B} \tag{33e}$$

$$\begin{aligned} \hat{\mathbf{z}}_{l,n} = & \mathbf{S}_{l,n}^{-1}\mathbf{B}^{\mathsf{T}}\left(\mathbf{I} - \boldsymbol{\Upsilon}_{l,n}\boldsymbol{\Psi}_{l,n}^{-1}\right)\boldsymbol{\varphi}_{l,n} \\ & - \mathbf{S}_{l,n}^{-1}\mathbf{B}^{\mathsf{T}}\left(\mathbf{I} - \boldsymbol{\Upsilon}_{l,n}\boldsymbol{\Psi}_{l,n}^{-1}\right)\boldsymbol{\Upsilon}_{l,n}\boldsymbol{\phi}_{l,n|n-1} \end{aligned} \tag{33f}$$

Combined update:

$$\mathbf{G}_{l,n} = \left(\mathbf{I} - \boldsymbol{\Phi}_{l,n|n-1}\boldsymbol{\Pi}_{l,n}\right)\mathbf{B} \tag{34a}$$

$$\hat{\mathbf{x}}_{l,n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\boldsymbol{\phi}_{k,n|n} + \mathbf{G}_{k,n}\hat{\mathbf{z}}_{k,n} : \forall k \in \mathcal{N}\} \tag{34b}$$

$$\mathbf{P}_{l,n} \leftarrow \boxed{\text{ACF}} \leftarrow \{\boldsymbol{\Phi}_{l,n|n} + \mathbf{G}_{l,n}\mathbf{S}_{l,n}\mathbf{G}_{l,n}^{\mathsf{T}} : \forall k \in \mathcal{N}\} \tag{34c}$$

Now, consider the state vector estimation error given by

$$\boldsymbol{\epsilon}_{l,n} = \mathbf{x}_n - \hat{\mathbf{x}}_{l,n} \ \text{ and } \ \boldsymbol{\mathcal{E}}_n = \left[\boldsymbol{\epsilon}_{1,n}^{\mathsf{T}}, \ldots, \boldsymbol{\epsilon}_{|\mathcal{N}|,n}^{\mathsf{T}}\right]^{\mathsf{T}}. \tag{37}$$

From (26b), (36) and (37) the network-wide evolution of state vector estimation error terms can be expressed as

$$\begin{aligned} \boldsymbol{\mathcal{E}}_n = & \left(\boldsymbol{\mathcal{M}} \otimes \boldsymbol{\mathcal{I}}\right)\left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{L}}_n\boldsymbol{\mathcal{Q}}\right)\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{E}}_{n-1} - \left(\boldsymbol{\mathcal{M}} \otimes \boldsymbol{\mathcal{I}}\right)\boldsymbol{\mathcal{L}}_n\mathbf{r}_{col,n} \\ & + \left(\boldsymbol{\mathcal{M}} \otimes \boldsymbol{\mathcal{I}}\right)\left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{L}}_n\boldsymbol{\mathcal{Q}}\right)\boldsymbol{\mathcal{B}}\mathbf{z}_{n-1} \\ & + \left(\boldsymbol{\mathcal{M}} \otimes \boldsymbol{\mathcal{I}}\right)\left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{L}}_n\boldsymbol{\mathcal{Q}}\right)\boldsymbol{\nu}_{col,n} \end{aligned} \tag{38}$$

where $\mathbf{r}_{col,n} = \left[\mathbf{r}_{1,n}^{\mathsf{T}}, \ldots, \mathbf{r}_{|\mathcal{N}|,n}^{\mathsf{T}}\right]^{\mathsf{T}}$, $\boldsymbol{\nu}_{col,n} = \left[\boldsymbol{\nu}_n^{\mathsf{T}}, \ldots, \boldsymbol{\nu}_n^{\mathsf{T}}\right]^{\mathsf{T}}$, $\boldsymbol{\mathcal{B}} = \left[\mathbf{B}^{\mathsf{T}}, \ldots, \mathbf{B}^{\mathsf{T}}\right]^{\mathsf{T}}$, $\boldsymbol{\mathcal{A}} = \text{block-diag}\{\mathbf{A}, \ldots, \mathbf{A}\}$, and

$$\begin{aligned} \boldsymbol{\mathcal{L}}_n = & \text{block-diag}\{\mathbf{L}_{l,n} : \forall l \in \mathcal{N}\} \\ \boldsymbol{\mathcal{Q}} = & \text{block-diag}\{\mathbf{Q}_l : \forall l \in \mathcal{N}\} \end{aligned}$$

while $\boldsymbol{\mathcal{M}}$ is the combination matrix consisted of diffusion weights. In a steady-state situation, where $\mathbf{z}_n$ becomes constant, $\mathbf{z}_n \to \mathbf{z}$, from (38), it follows that $\boldsymbol{\mathcal{E}}_n$ will have stable first and second-order moments if $\left(\boldsymbol{\mathcal{M}} \otimes \boldsymbol{\mathcal{I}}\right)\left(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{L}}_n\boldsymbol{\mathcal{Q}}\right)\boldsymbol{\mathcal{A}}$ has a spectral radius of strictly less than one. However, due to (22), $\boldsymbol{\mathcal{M}}$ is left stochastic and hence all its eigenvalues lie on or within the unit circle. Thus, it suffices that all block-diagonal elements of $(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{L}}_n \boldsymbol{\mathcal{Q}})\, \boldsymbol{\mathcal{A}}$, i.e., $\{(\mathbf{I} - \mathbf{L}_{l,n}\mathbf{Q}_l)\, \mathbf{A} : \forall l \in \mathcal{N}\}$ remain stable. This condition is determined by the recursions governing $\mathbf{P}_{l,n}$ in Algorithm 2. Such recursion have been considered in [41] and sufficient conditions for their stability, under the assumption that rank$(\mathbf{H}_l) \geqslant$ rank$(\mathbf{H}_l\mathbf{B}) =$ rank$(\mathbf{B})$, has been provided.

Remark 2. For the case of the consensus-based filter in Algorithm 3, all discussions in this subsection follow similarly, where $\boldsymbol{\mathcal{M}}$ should be replaced with $\boldsymbol{\mathcal{W}}^{\eta}$.

Remark 3. In essence, if the local filtering operations meet requirements for convergence; then, each agent will obtain local state vector estimates and the diffusion (*cf.* consensus) step will have the effect of reducing the uncertainty on these local state vector estimates. Therefore, the network-wide system will remain convergent.

IV. Numerical Example

The performance of Algorithm 2 and Algorithm 3 is demonstrated and compared to that of the optimal centralized solution (Algorithm 1) in a simulation example. To this end, consider the general discrete-time system with

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

while $\forall l \in \mathcal{N} : \mathbf{H}_l = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$.

The observation noise at node $l \in \mathcal{N}$ was considered to be a white Gaussian random process with the covariance matrix

$$\boldsymbol{\Sigma}_{\boldsymbol{\omega}_l} = a_l \begin{bmatrix} 0.052 & 0.01 \\ 0.01 & 0.052 \end{bmatrix} \quad \text{with} \quad a_l = \begin{cases} 1 & \text{if } l \text{ odd} \\ 4 & \text{if } l \text{ even} \end{cases}$$

while the state evolution noise covariance was

$$\boldsymbol{\Sigma}_{\boldsymbol{\nu}} = \left(\begin{bmatrix} 0.025 & 0.5 \\ 0.5 & 10 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \times 10^{-3}$$

and the network of 200 nodes with the topology shown in Fig. 1 was used in the simulation example.

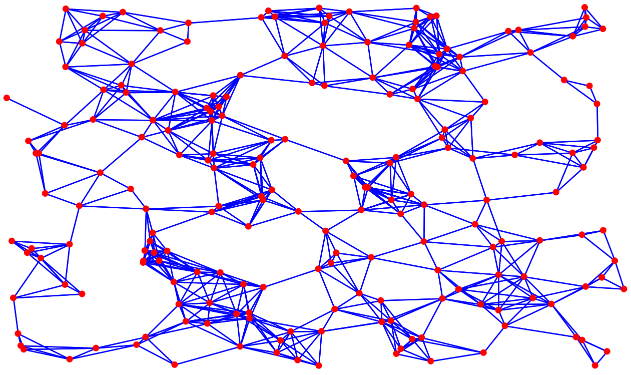

Fig. 1. Network of 200 nodes and 732 edges used for simulations.

The state vector consisted of four elements $\{x_{\text{1stD}}, x_{\text{2ndD}}, x_{\text{3rdD}}, x_{\text{4thD}}\}$. In Fig. 2, the state vector estimates obtained through the diffusion and consensus frameworks proposed in Algorithm 2 and Algorithm 3 are compared to that of the centralized solution in Algorithm 1. Observe that the introduced distributed framework both in its diffusion and consensus formulation tracked the state vector sequence and obtained a performance level close to that of the centralized solution in Algorithm 1.

The unknown actuations forced upon the dynamic system was a vector of two elements $\{z_{\text{1stD}}, z_{\text{2ndD}}\}$. In Fig. 3, the estimates of the unknown actuations obtained through the proposed framework of Algorithm 2 and Algorithm 3 are shown. Although both the diffusion and consensus formulations estimated the unknown actuations correctly, estimates obtained via the consensus formulation appeared to be consistently more accurate. It should be noted that the accuracy of the consensus formulation comes at a higher communication traffic cost as the ACFs where iterated 10 times at each time instant in order to achieve consensus.

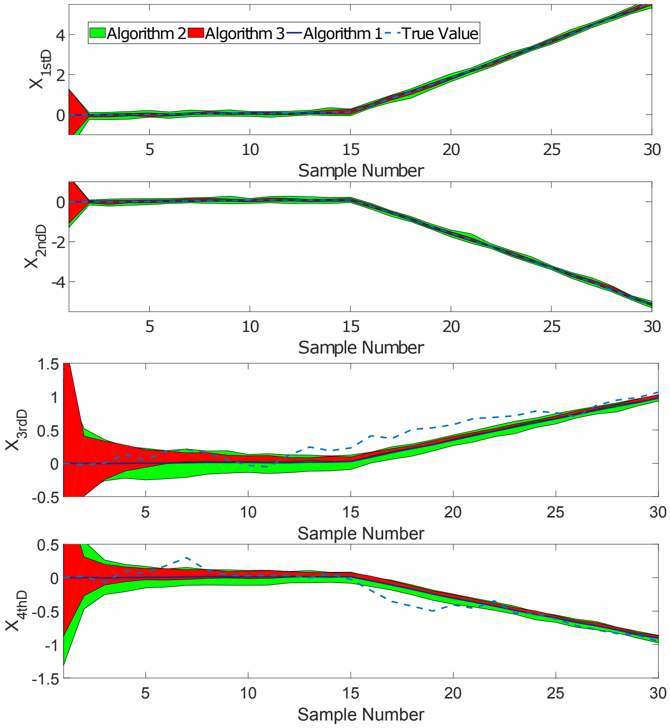

Fig. 2. The state vector estimate obtained through the proposed distributed solutions in Algorithm 2 and Algorithm 3 are shown along side estimates obtained via the centralized framework of Algorithm 1. Estimates obtained through the diffusion (*cf.* consensus) framework across all nodes lie in the region in green (*cf.* red).

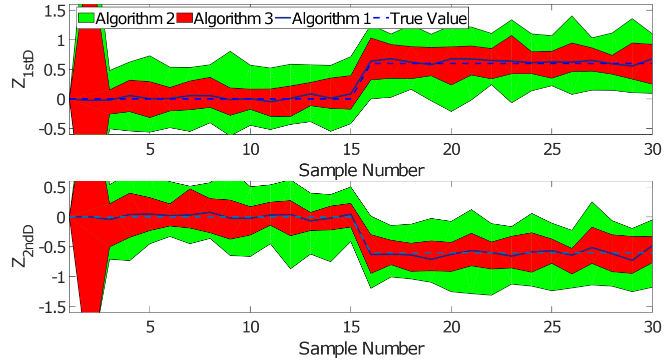

Fig. 3. Estimates of the unknown outer network actuations obtained through the proposed distributed solution in Algorithm 2 and Algorithm 3. Estimates obtained through the diffusion (*cf.* consensus) framework across all nodes lie in the region in green (*cf.* red).

V. Conclusion

The problem of tracking/estimating state vector sequences in presence of unknown actuations through observations made over sensor networks has been considered. A centralized solution to this problem has been proposed based upon two-stage Kalman filtering techniques. Most importantly, it has been shown that the operations of such a centralized solution can be decomposed into a format suitable for distributed implementation via networked agents. This derived format has been transformed into two formulations for a distributed filtering technique, one based on diffusion and one based on consensus. The diffusion formulation is a cost-effective solution to the posed problem, while in the context of the consensus formulation complexity is traded for accuracy. Finally, performance of the derived filtering algorithms has been demonstrated and compared to their centralized counterpart in a simulation example.

References

[1] S. Xu, K. Dogancay, and H. Hmam, "Distributed pseudolinear estimation and UAV path optimization for 3D AOA target tracking," *Signal Processing*, vol. 133, no. Supplement C, pp. 64–78, 2017.

[2] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, March 2006.

[3] S. Fattahi, G. Fazelnia, J. Lavaei, and M. Arcak, "Transformation of optimal centralized controllers into near-globally optimal static distributed controllers," *IEEE Transactions on Automatic Control*, Inprint, 2018.

[4] E. Franco, R. Olfati-Saber, T. Parisini, and M. M. Polycarpou, "Distributed fault diagnosis using sensor networks and consensus-based filters," *In Proceedings of IEEE International Conference on Decision and Control*, pp. 386–391, December 2006.

[5] F. Dorfler, J. W. Simpson-Porco, and F. Bullo, "Breaking the hierarchy: Distributed control and economic optimality in microgrids," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 3, pp. 241–253, September 2016.

[6] S. P. Talebi, S. Kanna, and D. P. Mandic, "A distributed quaternion Kalman filter with applications to smart grid and target tracking," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 4, pp. 477–488, December 2016.

[7] A. G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, 1998.

[8] J. S. Shamma, *Cooperative Control of Distributed Multi-Agent Systems*. John Wiley & Sons, 2007.

[9] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, January 2013.

[10] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Systems*, vol. 34, no. 4, pp. 87–97, August 2014.

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

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

[13] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative control of multi-agent systems: Optimal design and adaptive control*. Springer-Verlag, 2014.

[14] W. Han, H. L. Trentelman, Z. Wang, and Y. Shen, "A simple approach to distributed observer design for linear systems," *IEEE Transactions on Automatic Control*, Inprint, 2018.

[15] A. Mitra and S. Sundaram, "Distributed observers for LTI systems," *IEEE Transactions on Automatic Control*, Inprint, 2018.

[16] L. Wang and A. S. Morse, "A distributed observer for a time-invariant linear system," *In Proceedings of American Control Conference*, pp. 2020–2025, May 2017.

[17] B. Jia, K. D. Pham, E. Blasch, D. Shen, Z. Wang, and G. Chen, "Cooperative space object tracking using space-based optical sensors via consensus-based filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1908–1936, August 2016.

[18] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," *In Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 7036–7042, December 2009.

[19] S. P. Talebi, S. Kanna, Y. Xia, and D. P. Mandic, "Cost-effective diffusion Kalman filtering with implicit measurement exchanges," *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 4411–4415, 2017.

[20] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, September 2010.

[21] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," *In Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 8179–8184, December 2005.

[22] M. Alighanbari and J. P. How, "An unbiased Kalman consensus algorithm," *In Proceedings of American Control Conference*, pp. 3519–3524, June 2006.

[23] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," *In Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 5492–5498, December 2007.

[24] J. Speyer, "Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem," *IEEE Transactions on Automatic Control*, vol. 24, no. 2, pp. 266–269, April 1979.

[25] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Transactions on Automatic Control*, vol. 33, no. 1, pp. 88–94, 1988.

[26] B. S. Rao and H. F. Durrant-Whyte, "Fully decentralised algorithm for multisensor Kalman filtering," *IEE Proceedings D-Control Theory and Applications*, vol. 138, no. 5, pp. 413–420, September 1991.

[27] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *In Proceedings of Fourth International Symposium on Information Processing in Sensor Networks*, pp. 63–70, 2005.

[28] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," *In Proceedings of IEEE Conference on Decision and Control*, December 2005.

[29] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[30] A. H. Sayed and C. G. Lopes, "Adaptive processing over distributed networks," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 8, pp. 1504–1510, August 2007.

[31] W. Li, G. Wei, F. Han, and Y. Liu, "Weighted average consensus-based unscented Kalman filtering," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 558–567, February 2016.

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

[33] S. P. Talebi and S. Werner, "Distributed kalman filtering: Consensus, diffusion, and mixed," *In Proceedings of IEEE Conference on Control Technology and Applications*, Inprint, 2018.

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

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

[36] 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, October 2008.

[37] M. Hou and R. J. Patton, "Optimal filtering for systems with unknown inputs," *IEEE Transactions on Automatic Control*, vol. 43, no. 3, pp. 445–449, March 1998.

[38] C. Hsieh and F. Chen, "General two-stage Kalman filters," *IEEE Transactions on Automatic Control*, vol. 45, no. 4, pp. 819–824, April 2000.

[39] S. Gillijns and B. De Moor, "Unbiased minimum-variance input and state estimation for linear discrete-time systems," *Automatica*, vol. 43, no. 1, pp. 111–116, 2007.

[40] C. Hsieh, "Robust two-stage Kalman filters for systems with unknown inputs," *IEEE Transactions on Automatic Control*, vol. 45, no. 12, pp. 2374–2378, December 2000.

[41] M. Darouach and M. Zasadzinski, "Unbiased minimum variance estimation for systems with unknown exogenous inputs," *Automatica*, vol. 33, no. 4, pp. 717–719, 1997.