KALMAN FILTERING AND CLUSTERING IN SENSOR NETWORKS

Sayed Pouria Talebi[†]*, Stefan Werner*[*]*, and Visa Koivunen*[†]

[†]Department of Signal Processing and Acoustics, Aalto University, Espoo FI-02150 Finland
[*]Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim NO-7491 Norway
E-mails: {pouria.talebi, visa.koivunen}@aalto.fi & stefan.werner@ntnu.no

ABSTRACT

In this work, a distributed Kalman filtering and clustering framework for sensor networks tasked with tracking multiple state vector sequences is developed. This is achieved through recursively updating the likelihood of a state vector estimation from one agent offering valid information about the state vector of its neighbors, given the available observation data. These likelihoods then form the diffusion coefficients, used for information fusion over the sensor network. For rigour, the mean and mean square behavior of the developed Kalman filtering and clustering framework is analyzed, convergence criteria are established, and the performance of the developed framework is demonstrated in a simulation example.

***Index Terms*—** Adaptive learning over networks, distributed Kalman filtering, adaptive clustering, multi-task sensor networks.

1. INTRODUCTION

In recent years, distributed learning and optimization techniques have become the prevailing method for implementing signal processing and control operations over large-scale sensor networks [1–16]. In this context, due to the optimality of the Kalman filter for tracking linear Gaussian systems and the flexibility of the state space representation for modeling a wide range of real-world dynamic systems, a great deal of interest has been shown in developing distributed Kalman filtering frameworks [4,8–15], where in-network cooperation between agents of the sensor network is used to enhance various performance criteria, such as accuracy. However, cooperation is not beneficial in all incidences. In so-called multi-task sensor networks, that are used for tracking multiple state vector sequences or targets [17–20], information fusion between agents tasked with tracking different targets can degrade their performance and lead to undesired results that propagate throughout the network [21]. Thus, adaptive real-time techniques that can cluster together agents of a network that are tracking a common target are highly desired.

A number of effective clustering techniques have been proposed in the literature relying upon the assumption that partial prior knowledge is available about the network cluster structure, such as a subset of agents that belong to a given cluster or signal characteristics of interest to agents of different clusters [22–24]. Although partial prior information might be available about the cluster structure of the network in some incidences, in general information of this kind is either not available or it is time dependent. In order to introduce a more inclusive framework for clustering multi-task sensor networks, adaptive approaches have been investigated in [21,25]. Moreover, a class of algorithms originally developed for adaptive estimation of optimal diffusion coefficients in diffusion-based distributed least mean square algorithms have been modified for clustering purposes, conjuring that diffusion cross-coefficients of agents in different clusters tend to zero essentially eliminating cooperation between them and thus resulting in a clustered network [21].

Most adaptive clustering techniques are developed for sensor networks using the distributed least mean square algorithm as their main learning scheme [21–25]. Although these techniques are applicable to sensor networks that employ distributed Kalman filtering frameworks, they fail to exploit the extra level of information available in Kalman filtering such as the estimate of the state vector estimation error covariance matrices and the state evolution model. Based on the available information in Kalman filtering and borrowing ideas from covariance intersection, a number distributed Kalman filtering frameworks with optimal diffusion coefficients have been proposed in the literature [26,27]. However, the results are not applicable for clustering of multi-task sensor networks, as these results were obtained under the firm assumption that all agents of the sensor network are tracking the same state vector sequence. In addition, a fault detection scheme for detecting and isolating agents producing faulty measurements in Kalman filtering networks has been proposed in [28]; however, the results are not efficiently adaptable to clustering applications dealing with large-scale sensor networks.

In this work, we derive a distributed Kalman filtering algorithm drawing upon ideas from one of the authors previous work in [29]. The derivation provides insight into the operations of the distributed Kalman filter allowing for a cost-effective adaptive clustering technique to be developed. The developed clustering technique exploits the covariance information available in the Kalman filtering procedure to recursively update likelihoods of state estimates from one agent in the sensor network offering valid information about the state vector sequence of its neighbors. Then, these likelihoods are used for information fusion and clustering purposes. Finally, the performance of the developed distributed Kalman filtering and clustering framework is analyzed, indicating that it operates in an unbiased fashion and establishing mean square error convergence criteria.

Mathematical Notations: Scalars, column vectors, and matrices are denoted by lowercase, bold lowercase, and bold uppercase letters respectively. The identity matrix is represented by $\mathbf{I}$. The value of the probability density function of a random vector at $\boldsymbol{a}$ conditional on $\boldsymbol{b}$ is denoted as $\mathsf{p}(\boldsymbol{a}|\boldsymbol{b})$ and hereafter is referred to as the likelihood of $\boldsymbol{a}$. The value of the probability density function of a zero-mean Gaussian random vector with covariance matrix $\mathbf{B}$ at $\boldsymbol{a}$ is denoted as $\mathsf{g}(\boldsymbol{a}, \mathbf{B})$. The transpose and statistical expectation operators are denoted by $(\cdot)^{\mathsf{T}}$ and $\mathsf{E}\{\cdot\}$ respectively, whereas $\delta(\cdot)$ denotes the Kronecker delta function. The set of natural numbers is denoted by $\mathbb{N}$.

This work was supported by The Academy of Finland under Grant 296849.

2. PROBLEM FORMULATION

Consider a connected sensor network modeled as an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where the node set $\mathcal{N}$ represents agents of the sensor network, while the edge set $\mathcal{E}$ denotes bidirectional communication links between agents in the network. The neighborhood of node $i \in \mathcal{N}$, is denoted as $\mathcal{N}_i$ and is defined as the set of all nodes connected to node i including itself, with the cardinality of the set $\mathcal{N}_i$ denoted by $|\mathcal{N}_i|$. The nodes of the network are tasked with estimating the state vector sequences $\{\boldsymbol{x}_{i,n} : i \in \mathcal{N}, n \in \mathbb{N}\}$ through local observation sequences $\{\boldsymbol{y}_{i,n} : i \in \mathcal{N}, n \in \mathbb{N}\}$. The observation and state vector sequences are coupled via a state space model given by

$$\begin{aligned} \boldsymbol{x}_{i,n} &= \mathbf{A}_n \boldsymbol{x}_{i,n-1} + \boldsymbol{\nu}_{i,n} \\ \boldsymbol{y}_{i,n} &= \mathbf{H}_{i,n} \boldsymbol{x}_{i,n} + \boldsymbol{\omega}_{i,n} \end{aligned} \tag{1}$$

where $\mathbf{A}_n$ is the state transition matrix at time instant n which is assumed identical throughout the network[1], while $\mathbf{H}_{i,n}$ is the observation matrix at node i at time instant n, whereas the process noise sequences $\{\boldsymbol{\nu}_{i,n} : i \in \mathcal{N}, n \in \mathbb{N}\}$ are assumed to be white Gaussian processes that are identical for all nodes in a cluster and finally the observation noise sequences $\{\boldsymbol{\omega}_{i,n} : i \in \mathcal{N}, n \in \mathbb{N}\}$ are temporally uncorrelated and spatially independent zero-mean white Gaussian noise processes. The joint covariance matrix of the observation and process noise sequences can be expressed as

$$\mathsf{E}\left\{ \begin{bmatrix} \boldsymbol{\nu}_{i,n} \\ \boldsymbol{\omega}_{i,n} \end{bmatrix} \begin{bmatrix} \boldsymbol{\nu}_{l,m}^{\mathsf{T}} & \boldsymbol{\omega}_{l,m}^{\mathsf{T}} \end{bmatrix} \right\} = \begin{bmatrix} \mathbf{C}_{\boldsymbol{\nu}_{i,n}} \alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{\boldsymbol{\omega}_{i,n}} \beta \end{bmatrix} \delta(m-n) \tag{2}$$

where $\alpha = 1$ if i and l are in the same cluster and $\alpha = 0$ otherwise, while $\beta = \delta(i-l)$.

The optimal solution to this problem, in the mean square error sense, comes in the form of a centralized Kalman filter implemented at each node [15,29], the operations of which are summarized as:
For node $i \in \mathcal{N}$:
Initialize with:

$$\begin{aligned} \hat{\boldsymbol{x}}_{i,0|0} &= \mathsf{E}\{\boldsymbol{x}_{i,0}\} \\ \mathbf{M}_{i,0|0} &= \mathsf{E}\left\{(\boldsymbol{x}_{i,0} - \mathsf{E}\{\boldsymbol{x}_{i,0}\})(\boldsymbol{x}_{i,0} - \mathsf{E}\{\boldsymbol{x}_{i,0}\})^{\mathsf{T}}\right\} \end{aligned} \tag{3}$$

Model update:

$$\begin{aligned} \hat{\boldsymbol{x}}_{i,n|n-1} &= \mathbf{A}_n \hat{\boldsymbol{x}}_{i,n-1|n-1} \\ \mathbf{M}_{i,n|n-1} &= \mathbf{A}_n \mathbf{M}_{i,n-1|n-1} \mathbf{A}_n^{\mathsf{T}} + \mathbf{C}_{\boldsymbol{\nu}_{i,n}} \end{aligned} \tag{4}$$

Measurement update:

$$\mathbf{S}_{i,n} = \sum_{l \in \mathcal{C}_i} \mathbf{H}_{l,n}^{\mathsf{T}} \mathbf{C}_{\boldsymbol{\omega}_{l,n}}^{-1} \mathbf{H}_{l,n} \tag{5a}$$

$$\boldsymbol{q}_{i,n} = \sum_{l \in \mathcal{C}_i} \mathbf{H}_{l,n}^{\mathsf{T}} \mathbf{C}_{\boldsymbol{\omega}_{l,n}}^{-1} \boldsymbol{y}_{l,n} \tag{5b}$$

$$\mathbf{M}_{i,n|n}^{-1} = \mathbf{M}_{i,n|n-1}^{-1} + \mathbf{S}_{i,n} \tag{5c}$$

$$\hat{\boldsymbol{x}}_{i,n|n} = \hat{\boldsymbol{x}}_{i,n|n-1} + \mathbf{M}_{i,n|n}\big(\boldsymbol{q}_{i,n} - \mathbf{S}_{i,n} \hat{\boldsymbol{x}}_{i,n|n-1}\big) \tag{5d}$$

where $\hat{\boldsymbol{x}}_{i,n|n-1}$ and $\hat{\boldsymbol{x}}_{i,n|n}$ denote the *a priori* and *a posteriori* estimates of $\boldsymbol{x}_{i,n}$, whereas $\mathcal{C}_i$ denotes the set of nodes in the same cluster as node i. From [29], if i and l are in the same cluster, we have $\mathbf{M}_{i,n|n} = \mathbf{M}_{l,n|n}$. Finally, note that it is assumed a central regulating unit that has prior knowledge of the network clustering structure organizes the distribution of information over the network.

[1]The results obtained in this work are readily expandable to the case where state transition matrices are only assumed identical within a cluster; however, discussions on this case have been omitted due to space restrictions of the venue.

3. DISTRIBUTED TRACKING AND CLUSTERING

Although the centralized Kalman filter presents the optimal solution, the assumption of prior knowledge of the cluster structure of the network and reliance on a central organizing unit for distribution of data make the algorithm impractical for implementation in large-scale sensor networks with a changing clustering formation. To this end, a fully decentralized Kalman filtering and clustering framework is developed in this section.

Considering the expression in (5a)-(5d), replacing (5a) and (5b) into (5d) yields

$$\begin{aligned} \hat{\boldsymbol{x}}_{i,n|n} = \hat{\boldsymbol{x}}_{i,n|n-1} &+ \mathbf{M}_{i,n|n} \sum_{\forall l \in \mathcal{C}_i} \mathbf{H}_{l,n}^{\mathsf{T}} \mathbf{C}_{\boldsymbol{\omega}_{l,n}}^{-1} \boldsymbol{y}_{l,n} \\ &- \mathbf{M}_{i,n|n} \sum_{\forall l \in \mathcal{C}_i} \mathbf{H}_{l,n}^{\mathsf{T}} \mathbf{C}_{\boldsymbol{\omega}_{l,n}}^{-1} \mathbf{H}_{l,n} \hat{\boldsymbol{x}}_{i,n|n-1}. \end{aligned}$$

which in a similar fashion ot [29] can be rearranged to give

$$\begin{aligned} \hat{\boldsymbol{x}}_{i,n|n} =& \hat{\boldsymbol{x}}_{i,n|n-1} \\ &+ \sum_{\forall l \in \mathcal{C}_i} \mathbf{M}_{l,n|n} \mathbf{H}_{l,n}^{\mathsf{T}} \mathbf{C}_{\boldsymbol{\omega}_{l,n}}^{-1} \left(\boldsymbol{y}_{l,n} - \mathbf{H}_{l,n} \hat{\boldsymbol{x}}_{i,n|n-1}\right). \end{aligned} \tag{6}$$

Now, the principle of diffusion can be used to simultaneously force a consensus among nodes of the same cluster and approximate the *a posteriori* estimate of the state vector as given in (6) through the summation

$$\hat{\boldsymbol{x}}_{i,n|n} = \frac{1}{|\mathcal{C}_i|} \sum_{\forall l \in \mathcal{C}_i} \boldsymbol{\psi}_{l,n} \tag{7}$$

where $|\mathcal{C}_i|$ is the cardinality of $\mathcal{C}_i$, while[2]

$$\begin{aligned} \boldsymbol{\psi}_{l,n} =& \hat{\boldsymbol{x}}_{l,n|n-1} \\ &+ |\mathcal{C}_l| \mathbf{M}_{l,n|n} \mathbf{H}_{l,n}^{\mathsf{T}} \mathbf{C}_{\boldsymbol{\omega}_{l,n}}^{-1} \left(\boldsymbol{y}_{l,n} - \mathbf{H}_{l,n} \hat{\boldsymbol{x}}_{l,n|n-1}\right). \end{aligned} \tag{8}$$

In a similar fashion to what was described for the state vector estimation, replacing (5a) into (5c) yields

$$\begin{aligned} \mathbf{M}_{i,n|n}^{-1} =& \mathbf{M}_{i,n|n-1}^{-1} + \sum_{\forall l \in \mathcal{C}_i} \mathbf{H}_{l,n}^{\mathsf{T}} \mathbf{C}_{\boldsymbol{\omega}_{l,n}}^{-1} \mathbf{H}_{l,n} \\ =& \frac{1}{|\mathcal{C}_i|} \sum_{\forall l \in \mathcal{C}_i} \boldsymbol{\Gamma}_{l,n} \end{aligned} \tag{9}$$

where

$$\boldsymbol{\Gamma}_{l,n} = \mathbf{M}_{l,n|n-1}^{-1} + |\mathcal{C}_l| \mathbf{H}_{l,n}^{\mathsf{T}} \mathbf{C}_{\boldsymbol{\omega}_{l,n}}^{-1} \mathbf{H}_{l,n}. \tag{10}$$

Now, $\mathbf{M}_{i,n|n}$ in the formulation given in (9) and $\hat{\boldsymbol{x}}_{i,n|n}$ in the formulation given in (7) can be calculated through the use of diffusion and/or consensus filters over local state vector estimates given in (8) and their associate covariance information given in (10), that is, if each node can detect other nodes in its neighborhood that are estimating the same state vector. To this end, we wish to determine the likelihood of the local estimate at one node, $\boldsymbol{\psi}_{l,n}$, also being a valid estimate for the state vector at its neighboring node, given the available information, that is $\forall l \in \mathcal{N}_i : \mathsf{p}(\boldsymbol{\psi}_{l,n} | \boldsymbol{y}_{i,1:n})$, where $y_{i,1:n} = \{\boldsymbol{y}_{i,k} : k = 1, \ldots, n\}$ is the collection of observation information available to node i up to time instant n.

The value of the *posterior* density function, $\mathsf{p}(\boldsymbol{\psi}_{l,n} | \boldsymbol{y}_{i,1:n})$, according to Bayes theorem, can be calculated up to a scaling factor as

$$\mathsf{p}(\boldsymbol{\psi}_{l,n} | \boldsymbol{y}_{i,1:n}) \propto \frac{\mathsf{p}(\boldsymbol{y}_{i,n} | \boldsymbol{\psi}_{l,n}) \mathsf{p}(\boldsymbol{\psi}_{l,n} | \boldsymbol{y}_{i,1:n-1})}{\sum_{\forall j \in \mathcal{N}_i} \mathsf{p}(\boldsymbol{y}_{i,n} | \boldsymbol{\psi}_{j,n}) \mathsf{p}(\boldsymbol{\psi}_{j,n} | \boldsymbol{y}_{i,1:n-1})}. \tag{11}$$

[2]Note that if nodes i and l belong to the same cluster, we have $\mathcal{C}_i = \mathcal{C}_l$.

Considering the mapping of the local state estimate at node l onto the observation space at node i, that is $\mathbf{H}_{i,n}\boldsymbol{\psi}_{l,n}$, we have

$$\boldsymbol{\theta}_{(i,l),n} = \boldsymbol{y}_{i,n} - \mathbf{H}_{i,n}\boldsymbol{\psi}_{l,n} = \mathbf{H}_{i,n}\left(\boldsymbol{x}_{i,n} - \boldsymbol{\psi}_{l,n}\right) + \boldsymbol{\omega}_{i,n}. \quad (12)$$

Then, if $\boldsymbol{\psi}_{l,n}$ is indeed a valid estimate of $\boldsymbol{x}_{i,n}$, given the linear Gaussian framework of the Kalman filter, $\boldsymbol{\psi}_{l,n}$ and by extension $\boldsymbol{\theta}_{(i,l),n}$, also have Gaussian distributions, where based on (12), it can be conjectured that

$$\mathbf{C}_{\boldsymbol{\theta}_{(i,l),n}} = \mathsf{E}\left\{\boldsymbol{\theta}_{(i,l),n}\boldsymbol{\theta}_{(i,l),n}^{\mathsf{T}}\right\} = \mathbf{H}_{i,n}\boldsymbol{\Gamma}_{l,n}^{-1}\mathbf{H}_{i,n}^{\mathsf{T}} + \mathbf{C}_{\boldsymbol{\omega}_{i,n}}. \quad (13)$$

Thus, from the expressions in (12) and (13), the likelihood of observing $\boldsymbol{y}_{i,n}$ under the assumption that $\boldsymbol{\psi}_{l,n}$ is a valid estimate of the state vector at node i is given by

$$\mathsf{p}(\boldsymbol{y}_{i,n}|\boldsymbol{\psi}_{l,n}) = \mathsf{g}\left(\boldsymbol{\theta}_{(i,l),n}, \mathbf{C}_{\boldsymbol{\theta}_{(i,l),n}}\right) \quad (14)$$

which can be evaluated at node i as node l is required to share $\{\boldsymbol{\psi}_{l,n}, \boldsymbol{\Gamma}_{l,n}\}$ with all of its neighbors.

Considering the Kalman filter produces optimal estimates of the state vector given the available observation sequence, it is held true that $\hat{\boldsymbol{x}}_{i,n-1|n-1}$ is indicative of the information held in $\boldsymbol{y}_{i,1:n}$; hence, we use the approximation (see Remark 2)

$$\mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n-1}) \approx \mathsf{p}(\boldsymbol{\psi}_{l,n}|\hat{\boldsymbol{x}}_{i,n-1|n-1}). \quad (15)$$

Once again, taking into account the Gaussian linear framework in Kalman filtering approaches, mapping $\hat{\boldsymbol{x}}_{i,n-1|n-1}$ onto the next time instant yields

$$\boldsymbol{\varepsilon}_{(i,l),n} = \boldsymbol{\psi}_{l,n} - \mathbf{A}_n\hat{\boldsymbol{x}}_{i,n-1|n-1} = \boldsymbol{\psi}_{l,n} - \hat{\boldsymbol{x}}_{i,n|n-1} \quad (16)$$

which given the expression in (4) and conditional on $\boldsymbol{\psi}_{l,n}$ being convergent to $\boldsymbol{x}_{i,n}$ we have

$$\mathbf{C}_{\boldsymbol{\varepsilon}_{(i,l),n}} = \mathsf{E}\left\{\boldsymbol{\varepsilon}_{(i,l),n}\boldsymbol{\varepsilon}_{(i,l),n}^{\mathsf{T}}\right\} = \mathbf{M}_{i,n|n-1}. \quad (17)$$

Thus, from (15)-(17), the value of the *prior* density function, $\mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n-1})$, can be approximated as

$$\begin{aligned}\mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n-1}) \approx& \mathsf{p}(\boldsymbol{\psi}_{l,n}|\hat{\boldsymbol{x}}_{i,n-1|n-1}) \\ \approx& \mathsf{g}\left(\boldsymbol{\varepsilon}_{(i,l),n}, \mathbf{C}_{\boldsymbol{\varepsilon}_{(i,l),n}}\right).\end{aligned} \quad (18)$$

Given the expressions in (11)-(18), each agent can now calculate required values of the *posterior* distribution as expressed in (11) for all of its neighbors, allowing these values to be used as coefficients of diffusion filters that approximate the summations in (7) and (9) which in turn allow the operations of the centralized Kalman filter to be implemented in a distributed fashion. The operations of such a distributed Kalman filter are summarized in Algorithm 1.

Algorithm 1. Distributed Kalman Filtering and Clustering

For node $i \in \mathcal{N}$:
Initialize as in (3).
Implement model update equations as given in (4).
Implement measurement update as:

$$\begin{aligned}\boldsymbol{\Gamma}_{i,n} &= \mathbf{M}_{i,n|n-1}^{-1} + \mathbf{H}_{i,n}^{\mathsf{T}}\mathbf{C}_{\boldsymbol{\omega}_{i,n}}^{-1}\mathbf{H}_{i,n} \\ \mathbf{M}_{i,n|n}^{-1} &= \sum_{\forall l\in\mathcal{N}_i} \mathsf{p}(\boldsymbol{\psi}_{l,n-1}|\boldsymbol{y}_{i,1:n-1})\boldsymbol{\Gamma}_{l,n} \\ \mathbf{G}_{i,n} &= \mathbf{M}_{i,n|n}\mathbf{H}_{i,n}^{\mathsf{T}}\mathbf{C}_{\boldsymbol{\omega}_{i,n}}^{-1} \\ \boldsymbol{\psi}_{i,n} &= \hat{\boldsymbol{x}}_{i,n|n-1} + \mathbf{G}_{i,n}\left(\boldsymbol{y}_{i,n} - \mathbf{H}_{i,n}\hat{\boldsymbol{x}}_{i,n|n-1}\right)\end{aligned}$$

Share $\{\boldsymbol{\psi}_{i,n}, \boldsymbol{\Gamma}_{i,n}\}$ and evaluate $\{\mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n}) : \forall l \in \mathcal{N}_i\}$ using the received information; then, calculate:

$$\hat{\boldsymbol{x}}_{i,n|n} = \sum_{\forall l\in\mathcal{N}_i} \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n})\boldsymbol{\psi}_{l,n}$$

Remark 1: In Algorithm 1, diffusion coefficients are updated using the *a priori* state vector estimates that are in turn calculated using the diffusion coefficients at the previous time instant, resulting in a recursive updating pattern.
Remark 2: From the expression in (11), it can be observed that $\forall i \in \mathcal{N} : \sum_{\forall l\in\mathcal{N}_i} \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n}) = 1$, providing for unbiased information fusion over the network and indicating that: i) only the nominators in (11) have to be calculated and normalized to find the suitable diffusion coefficients, ii) the approximation in (15) only needs to hold true up to a scaling factor for the nodes in $\mathcal{N}_i$.

4. CONVERGENCE ANALYSIS

The error of the local state vector estimate at node i at time instant n can be formulated as

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \boldsymbol{x}_{i,n} - \boldsymbol{\psi}_{i,n} \\ =& \boldsymbol{x}_{i,n} - \hat{\boldsymbol{x}}_{i,n|n-1} - \mathbf{G}_{i,n}\left(\boldsymbol{y}_{i,n} - \mathbf{H}_{i,n}\hat{\boldsymbol{x}}_{i,n|n-1}\right)\end{aligned} \quad (19)$$

where substituting

$$\boldsymbol{y}_{i,n} = \mathbf{H}_{i,n}\boldsymbol{x}_{i,n} + \boldsymbol{\omega}_{i,n} \quad \text{and} \quad \boldsymbol{\epsilon}_{i,n|n-1} = \boldsymbol{x}_{i,n} - \hat{\boldsymbol{x}}_{i,n|n-1}$$

into the expression in (19) yields

$$\boldsymbol{\epsilon}_{i,n} = \left(\mathbf{I} - \mathbf{G}_{i,n}\mathbf{H}_{i,n}\right)\boldsymbol{\epsilon}_{i,n|n-1} - \mathbf{G}_{i,n}\boldsymbol{\omega}_{i,n}. \quad (20)$$

From (4) it can be concluded that $\boldsymbol{\epsilon}_{i,n|n-1} = \mathbf{A}_n\boldsymbol{\epsilon}_{i,n-1|n-1} + \boldsymbol{\nu}_{i,n}$, which upon replacing into (20) yields

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n} =& \left(\mathbf{I} - \mathbf{G}_{i,n}\mathbf{H}_{i,n}\right)\mathbf{A}_n\boldsymbol{\epsilon}_{i,n-1|n-1} \\ &+ \left(\mathbf{I} - \mathbf{G}_{i,n}\mathbf{H}_{i,n}\right)\boldsymbol{\nu}_{i,n} - \mathbf{G}_{i,n}\boldsymbol{\omega}_{i,n}.\end{aligned} \quad (21)$$

Given the expression in (1), error of the *a posteriori* state vector estimate at node i can be formulated as

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n|n} =& \boldsymbol{x}_{i,n} - \hat{\boldsymbol{x}}_{i,n|n} = \boldsymbol{x}_{i,n} - \sum_{\forall l\in\mathcal{N}_i} \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n})\boldsymbol{\psi}_{l,n} \\ =& \sum_{\forall l\in\mathcal{N}_i} \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n})\boldsymbol{\epsilon}_{l,n}.\end{aligned} \quad (22)$$

Now, replacing (21) into (22) gives a recursive expression for the state vector estimation error as

$$\begin{aligned}\boldsymbol{\epsilon}_{i,n|n} =& \sum_{\forall l\in\mathcal{N}_i} \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n})\left(\mathbf{I} - \mathbf{G}_{l,n}\mathbf{H}_{l,n}\right)\mathbf{A}_n\boldsymbol{\epsilon}_{l,n-1|n-1} \\ &+ \sum_{\forall l\in\mathcal{N}_i} \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n})\left(\mathbf{I} - \mathbf{G}_{l,n}\mathbf{H}_{l,n}\right)\boldsymbol{\nu}_{l,n} \\ &- \sum_{\forall l\in\mathcal{N}_i} \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n})\mathbf{G}_{l,n}\boldsymbol{\omega}_{l,n}.\end{aligned} \quad (23)$$

Assuming the network cluster structure becomes time invariant and diffusion coefficients $\{\mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n}) : \forall l \in \mathcal{N}_i\}$ have stabilized[3], taking the statistical expectation of $\boldsymbol{\epsilon}_{i,n|n}$ as expressed in (23) and considering that

$$\forall l \in \mathcal{N} : \mathsf{E}\left\{\boldsymbol{\nu}_{l,n}\right\} = 0 \quad \text{and} \quad \mathsf{E}\left\{\boldsymbol{\omega}_{l,n}\right\} = 0$$

we have

$$\mathsf{E}\left\{\boldsymbol{\epsilon}_{i,n|n}\right\} = \sum_{\forall l\in\mathcal{N}_i} \boldsymbol{\Upsilon}_{l,n}\mathsf{E}\left\{\boldsymbol{\epsilon}_{l,n-1|n-1}\right\} \quad (24)$$

where $\boldsymbol{\Upsilon}_{l,n} = \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n})\left(\mathbf{I} - \mathbf{G}_{l,n}\mathbf{H}_{l,n}\right)\mathbf{A}_n$. Hence, given the initialization condition in Algorithm 1, the expression in (24) indicates that the developed algorithm operates in an unbiased fashion.

[3] That is the diffusion coefficients in question have converged with steady-state variations reasonably low to be assumed constant or alternatively they are held constant to their long-term averaged values.

The recursive expression for the state vector estimation error in (23) can now be used to formulate the state vector estimation error covariance matrix at node i and time instant n as

$$\mathbf{C}_{\boldsymbol{\epsilon}_{i,n|n}} = \sum_{l\in\mathcal{N}_i}\sum_{k\in\mathcal{N}_i} \mathsf{E}\left\{\boldsymbol{\zeta}_{l,n}\boldsymbol{\zeta}_{k,n}^{\mathsf{T}}\right\} + \sum_{l\in\mathcal{N}_i}\sum_{k\in\mathcal{N}_i} \mathsf{E}\left\{\boldsymbol{\xi}_{l,n}\boldsymbol{\xi}_{k,n}^{\mathsf{T}}\right\} + \sum_{l\in\mathcal{N}_i}\sum_{k\in\mathcal{N}_i} \mathsf{E}\left\{\boldsymbol{\chi}_{l,n}\boldsymbol{\chi}_{k,n}^{\mathsf{T}}\right\} \tag{25}$$

where $\mathbf{C}_{\boldsymbol{\epsilon}_{i,n|n}} = \mathsf{E}\left\{\boldsymbol{\epsilon}_{i,n|n}\boldsymbol{\epsilon}_{i,n|n}^{\mathsf{T}}\right\}$ and $\boldsymbol{\zeta}_{l,n} = \boldsymbol{\Upsilon}_{l,n}\boldsymbol{\epsilon}_{l,n-1|n-1}$, whereas

$$\begin{aligned}\boldsymbol{\xi}_{l,n} &= \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n})\left(\mathbf{I}-\mathbf{G}_{l,n}\mathbf{H}_{l,n}\right)\boldsymbol{\nu}_{l,n}\\ \boldsymbol{\chi}_{l,n} &= \mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n})\mathbf{G}_{l,n}\boldsymbol{\omega}_{l,n}.\end{aligned}$$

Now, the following typical conditions in Kalman filtering analysis are held to be true [14,30,31]:

- The state evolution and observation functions for all nodes in the network become time invariant.
- The state evolution and observation noises become stationary.
- The matrix pairs $\forall l \in \mathcal{N} : \{\mathbf{A}_n, \mathbf{H}_{l,n}\}$ are observable and the matrix pairs $\forall l \in \mathcal{N} : \{\mathbf{A}_n, \mathbf{C}_{\boldsymbol{\nu}_{l,n}}^{\frac{1}{2}}\}$ are controllable.

Then, if the diffusion coefficients $\{\mathsf{p}(\boldsymbol{\psi}_{l,n}|\boldsymbol{y}_{i,1:n}) : \forall l, i \in \mathcal{N}\}$ are held constant, that is the cluster structure of the network becomes time invariant, from the framework introduced in [14,30], it follows that the matrices $\{\mathbf{M}_{l,n|n} : l \in \mathcal{N}\}$ become time invariant. Moreover, from Algorithm 1, time invariant matrices $\{\mathbf{M}_{l,n|n} : l \in \mathcal{N}\}$ result in the matrices $\{\mathbf{G}_{l,n} : l \in \mathcal{N}\}$ also becoming time invariant and therefore $\mathbf{C}_{\boldsymbol{\epsilon}_{i,n|n}}$ as expressed in (25) converges.

5. SIMULATION EXAMPLE

In order to demonstrate the performance of the developed Kalman filtering and clustering framework, we considered a target tracking application over the network shown in Fig. 1. The state vector, $\boldsymbol{x}_{i,n} = [X_{i,n}, Y_{i,n}, \dot{X}_{i,n}, \dot{Y}_{i,n}]^{\mathsf{T}}$, consists of the positions, $\{X_{i,n}, Y_{i,n}\}$, and velocities, $\{\dot{X}_{i,n}, \dot{Y}_{i,n}\}$, in the horizontal and vertical directions of the i^{th} target.

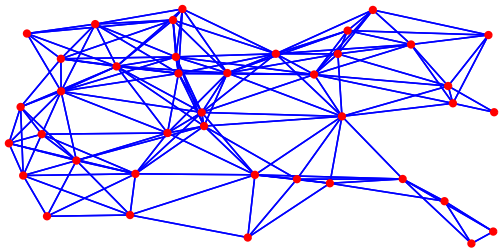

Fig. 1: The network of 40 nodes and 159 edges used in simulations.

The state is assumed to experience an unknown acceleration which is modeled as the process noise $\boldsymbol{\nu}_{i,n} = [\ddot{X}_{i,n}, \ddot{Y}_{i,n}]^{\mathsf{T}}$. The state space equations for this problem are given as

$$\boldsymbol{x}_{i,n} = \begin{bmatrix}1 & 0 & \Delta T & 0\\ 0 & 1 & 0 & \Delta T\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\boldsymbol{x}_{i,n-1} + \begin{bmatrix}\frac{1}{2}(\Delta T)^2 & 0\\ 0 & \frac{1}{2}(\Delta T)^2\\ \Delta T & 0\\ 0 & \Delta T\end{bmatrix}\boldsymbol{\nu}_{i,n}$$

$$\boldsymbol{y}_{i,n} = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}\boldsymbol{x}_{i,n} + \boldsymbol{\omega}_{i,n}$$

where $\Delta T = 1/25\,\text{s}$ is the sampling interval.

The network was tasked with tracking two maneuvering objects. For the first five seconds both objects where maneuvering in a close formation allowing all nodes in the network to cooperate with each other, tracking the formation as one target. Then, the objects split their formation, at which point each agent in the sensor network committed to tracking one of the objects, requiring the developed algorithm to correctly cluster the agents that were tracking the same object. The estimates of the position and speed of both objects are shown in Fig. 2. Note that the developed distributed Kalman filtering and clustering technique in Algorithm 1 was able to track both objects when they were moving collectively in a formation. In addition, when the objects had split their formation and were moving individuality, the proposed technique successfully clustered the network allowing for continuous accurate tracking of both objects.

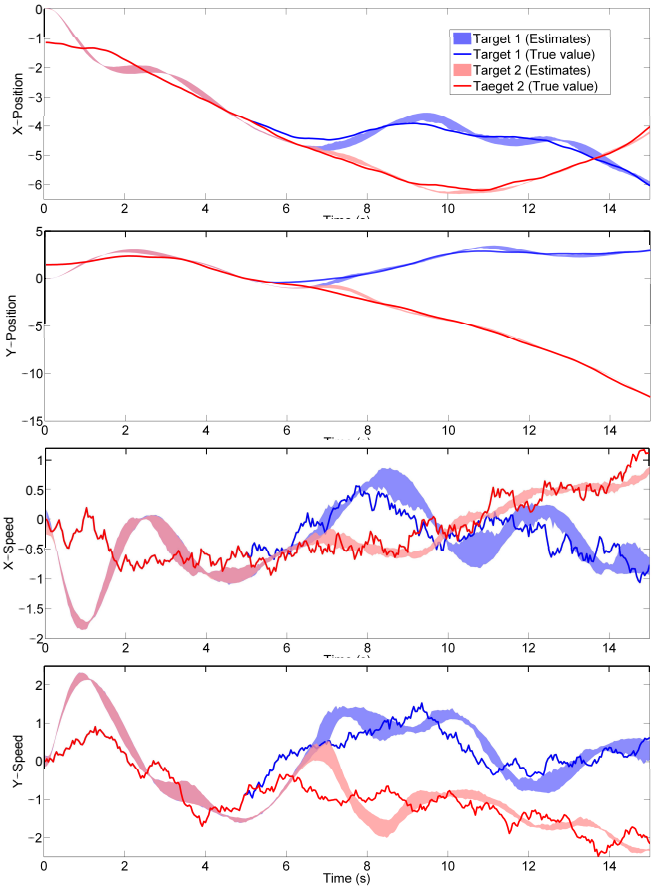

Fig. 2: Tracking performance of the proposed distributed Kalman filtering and clustering framework across all 40 nodes. Estimates of the position and speed of both objects are shown.

6. CONCLUSION

A distributed Kalman filtering and clustering framework has been developed in order to provide real-time estimation/tracking capabilities over large-scale multi-task sensor networks. The developed framework only requires the sharing of local state vector estimates alongside their associated covariance information and imposes limited communicational and computational load on the network. The work also includes a mathematical performance analysis resulting in the establishment of the mean square error convergence criteria.

7. REFERENCES

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

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

[3] 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.

[4] M. S. Mahmoud and H. M. Khalid, "Distributed Kalman filtering: A bibliographic review," *IET Control Theory Applications*, vol. 7, no. 4, pp. 483–501, March 2013.

[5] 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.

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

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

[8] 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.

[9] 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.

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

[11] Z. Hidayat, R. Babuska, B. De Schutter, and A. Nunez, "Decentralized Kalman filter comparison for distributed-parameter systems: A case study for a 1D heat conduction process," *In Proceedings of the* 16*th IEEE Conference on Emerging Technologies & Factory Automation*, pp. 1–8, September 2011.

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

[13] 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.

[14] 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.

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

[16] S. P. Talebi and D. P. Mandic, "Distributed particle filtering of α-stable signals," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1862–1866, December 2017.

[17] J. Liu, M. Chu, and J. E. Reich, "Multitarget tracking in distributed sensor networks," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 36–46, May 2007.

[18] X. Zhang, "Adaptive control and reconfiguration of mobile wireless sensor networks for dynamic multi-target tracking," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2429–2444, October 2011.

[19] I. Hwang, K. Roy, H. Balakrishnan, and C. Tomlin, "A distributed multiple-target identity management algorithm in sensor networks," *In Proceedings of the* 43*rd IEEE Conference on Decision and Control*, vol. 1, pp. 728–734, December 2004.

[20] J. Shin, L. G. Leonidas, and F. Zhao, "A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks," *In Proceedings of the* 2*nd International Conference on Information Processing in Sensor Networks*, pp. 223 – 238, April 2003.

[21] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, May 2013.

[22] A. Bertrandand and M. Moonen, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2196–2210, May 2011.

[23] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion LMS with sparsity-based regularization," *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3516–3520, April 2015.

[24] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion LMS over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, August 2014.

[25] J. Chen, C. Richard, and A. H. Sayed, "Adaptive clustering for multitask diffusion networks," *In Proceedings of the* 23*rd European Signal Processing Conference*, pp. 200–204, August 2015.

[26] 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.

[27] Y. Zhang, C. Wang, N. Li, and J. Chambers, "Diffusion Kalman filter based on local estimate exchanges," *In Proceedings of IEEE International Conference on Digital Signal Processing*, pp. 828–832, July 2015.

[28] S. P. Talebi, S. Kanna, Y. Xia, and D. P. Mandic, "A distributed quaternion Kalman filter with applications to fly-by-wire systems," *In Proceedings of IEEE International Conference on Digital Signal Processing*, pp. 30–34, October 2016.

[29] 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, March 2017.

[30] M. Kamgarpour and C. Tomlin, "Convergence properties of a decentralized kalman filter," *In Proceedings of the* 47*th IEEE Conference on Decision and Control*, pp. 3205–3210, December 2008.

[31] R. F. Stengel, *Optimal Control and Estimation*, Dover Publications, 1994.