Journal of Sensor Science and Technology
Vol. 28, No. 5 (2019) pp. 277-283
http://dx.doi.org/10.5369/JSST.2019.28.5.277
pISSN 1225-5475/eISSN 2093-7563

Distributed Fusion Estimation for Sensor Network

Il Young Song+, Jin Mo Song, Woong Ji Jeong, and Myoung Sool Gong

Abstract

In this paper, we propose a distributed fusion estimation for sensor networks using a receding horizon strategy. Communication channels were modelled as Markov jump systems, and a posterior probability distribution for communication channel characteristics was calculated and incorporated into the filter to allow distributed fusion estimation to handle path loss observation situations automatically. To implement distributed fusion estimation, a Kalman-Consensus filter was then used to obtain the average consensus, based on the estimates of sensors randomly distributed across sensor networks. The advantages of the proposed algorithms were then verified using a large-scale sensor network example.

Keywords: Distributed fusion, Fusion formula, Kalman filter, Multi-sensor, Sliding window

1. INTRODUCTION

Distributed computing has been a crucial philosophy in engineering problems, due to its ease of scalability, efficiency, and reliability. Many researchers today are attempting to apply this concept to various disciplines, including data fusion in sensor networks [1], distributed camera networks [2], and mobile robotics [3]. In the literature, the development of distributed signal processing algorithms is not a new topic, having been investigated over the past few years, and several types of distributed signal processing algorithms are well-known [4,5].

Recently, implementation of distributed signal processing algorithms has faced practical issues, such as varied network topologies and imperfect communication channels. As such, network scalability has been discussed in distributed Kalman filtering, in an attempt to address issues related to ad-hoc network topologies [6]. Accordingly, the topology of a network can be understood via algebraic graph theory, with individual network nodes employing Kalman filters to consider the limited communication bandwidth between neighboring nodes.

Imperfect communication channels have been another important issue for distributed signal processing implementation. As many sensors are randomly distributed and communicate with each other through wireless channels, communication links occasionally break down and become unstable. This can delay observations and incur packet losses. In previous research, the communication delay issue has been investigated as an out-of-sequence measurement problem [7].

To model an unreliable communication channel, a latent variable for the observation system is considered. The arrival of the observation is controlled by this latent variable, and under this formulation, a statistical convergence analysis was performed in [8]. The intermittent observation was modelled as a conditional probability distribution

$$p\left(v_t \middle| \gamma_t\right)=\begin{cases} N\left(0,R\right), & \gamma_t=1 \\ N\left(0,\sigma^2 I\right), & \gamma_t=0 \end{cases} \tag{1}$$

where v_t, γ_t, R, σ are zero mean white Gaussian observation noise, a latent variable, noise covariance with no loss, and unreliable noise deviation (i.e., $\sigma \rightarrow \infty$ means the absence of observation), respectively. In research conducted under this formulation, it was assumed that the latent variable γ_t was a Bernoulli process dependent on the state space model, and that there was a critical arrival probability value at which the estimation error covariance was bounded [8]. Because the algebraic Ricatti equation becomes a stochastic differential equation, only a bound analysis was available.

When observation noise is controlled using a latent variable, it is not easy to determine or model the value of the hyper-parameter σ that describes the characteristics of the communication channel. In this paper, rather than using the observation noise hyper-parameter, the characteristics of the communication channel have been modelled using the multiple model adaptive estimation (MMAE) approach [9].

To adjust for an observation mode that is switching continuously, we have proposed two algorithms. First, an interacting multiple model

Laser Applied System, Hanwha Corporation Defense R&D Center, 305 Pangyo-ro, Bundang-gu, Seongnam-si, 13488, Korea
+Corresponding author: com21dud@hanwha.com
(Received: Mar. 2, 2019, Revised: Jun. 22, Jul. 9, 2019, Accepted: Aug. 2, 2019)

(IMM) filter was applied to solve the MMAE problem. Second, by using a sliding window, the posterior probability of link failure was calculated and incorporated into the information fusion filtering. The sliding window collected the most recent set of observations, which were then sequentially processed, to calculate the posterior probability of the observation mode (absence or presence). Two MMAE solutions were implemented in distributed Kalman filtering, to ensure that intermittent observation situations were handled efficiently in a large-scale sensor network.

The remainder of paper has been organized as follows. Section 2 focusses on problem formulation, while preliminaries, including information fusion filtering and MMAE, are then given in Section 3. We provide details of the proposed algorithm in Section 4, and the advantages of the proposed algorithm have been evaluated in Section 5, before we present our conclusions in Section 6.

2. PROBLEM FORMULATION

Consider the discrete-time dynamic linear system:

$$\begin{aligned} x_{t+1} &= A_t x_t + w_t \\ y_t^i &= C_t^i x_t + v_t^i, \quad t = 0,1,\ldots, \quad i = 1,\ldots,N, \end{aligned} \tag{2}$$

where $A_t \in \Re^{n\times n}$ is the system matrix, $C_t^i \in \Re^{m\times n}$ is the observation matrix for the 'i'th sensor among N sensors, $x_t \in \Re^n$ is the state vector, $y_t^i \in \Re^m$ is the output vector (observation) of the 'i'th sensor in the network, and $w_t \in \Re^n$ and $v_t^i \in \Re^m$ are Gaussian random vectors with zero mean and covariance $Q \geq 0$ and $R^i > 0$, respectively. In addition, w_t is independent of w_s for $s < t$, and the initial state vector x_0 is also assumed to be Gaussian, with zero mean, and covariance P_0.

Here, the main goal was to obtain an accurate estimate given multiple observations, i.e., $E(x_t | y_t^1, \ldots, y_t^N)$, under an unreliable communication channel. In the proposed algorithm, we estimated another state, referred to as the communication characteristic state, θ_t, to cope with intermittent observations. In this framework, each sensor calculated the posterior probability of mode $p(\theta_t | y_t^i)$, which was then incorporated into the Kalman filtering equation, as a characteristic of the communication channel, at the current time.

In multiple sensory environments, a central fusion scheme is intuitive, where all observations are collected at one center and processed at once. As the number of sensors increases, the network size also grows and the topology is time-varying. In this case, a central fusion scheme would not be suitable [4]. A decentralized (or distributed fusion) algorithm has been suggested to satisfy these conditions, and is currently the focus of rigorous investigation. In a state estimation under multiple sensors in a flexible network environment, having an efficient distributed signal processing algorithm is essential from a practical perspective.

Considering the target system in (2), decentralized Kalman filtering is known to be globally optimal under perfect communication conditions [1]. However, following the recent initial work that has reported on state estimation with intermittent observations [8], the problem has been extended to multiple sensory systems, and discussion has been based on a graphical understanding of the network [10].

3. PRELIMINARIES

In this preliminary section, an information fusion filter [1, 4] has been introduced as a basic tool for fusing distributed sensors over the network. Further, a multiple model adaptive estimation has been discussed, for use in the mode probability calculation to manage intermittent observations.

3.1 Information Fusion Filter

In the centralized fusion set up, the observation system in (2) can be reformulated into a composite form, as shown in Eq. (3):

$$\begin{aligned} &C_t = \left[\left(C_t^1\right)^T, \ldots, \left(C_t^N\right)^T\right]^T, \quad v_t = \left[\left(v_t^1\right)^T, \ldots, \left(v_t^N\right)^T\right]^T, \\ &R = diag\left\{R^1, \ldots, R^N\right\}, \quad Y_t = \left[\left(y_t^1\right)^T, \ldots, \left(y_t^N\right)^T\right]^T. \end{aligned} \tag{3}$$

Then, the information forms of Kalman filtering equations (information filter) are given as follows:

Observation Update;

$$\begin{aligned} S_t &= C_t^T R^{-1} C_t, \\ z_t &= C_t^T R^{-1} Y_t, \\ M_t &= \left(P_t^{-1} + S_t\right)^{-1}, \\ \hat{x}_t &= \bar{x}_t + M_t\left[z_t - S_t \bar{x}_t\right], \end{aligned} \tag{4}$$

Time Update;

$$\begin{aligned} \bar{x}_{t+1} &= A_t \hat{x}_t, \\ P_{t+1} &= A_t M_t A_t^T + Q, \end{aligned} \tag{5}$$

where S_t and z_t represent the contribution terms of the state and information, respectively. To derive the decentralized fusion filter, a mathematically equivalent, decentralized form of the information filter can then be obtained, from the parallelization of the contribution terms, as shown in equations (6) and (7):

$$\begin{aligned} S_t &= C_t^T R^{-1} C_t \\ &= \sum\nolimits_{i=1}^{N} \left(C_t^i\right)^T \left(R^i\right)^{-1} C_t^i, \end{aligned} \tag{6}$$

$$\begin{aligned} z_t &= C_t^T R^{-1} Y_t \\ &= \sum\nolimits_{i=1}^{N} \left(C_t^i\right)^T \left(R^i\right)^{-1} y_t^i. \end{aligned} \tag{7}$$

Therefore, equations (4)–(7) define the information fusion filter that will be used in the proposed algorithm for distributed fusion, and the optimality of the distributed fusion algorithm is guaranteed due to mathematical equivalence.

3.2 Estimate Fusion within Link Failure between Cluster Heads and Nodes

When a system has parametric uncertainties, it can be modelled with a set of multiple models. A well-known example of multiple models in state estimation is the problem of tracking maneuvering targets. Target maneuvers have a set of distinctive models, including constant velocity, constant acceleration, and turning motion. Hence, by pre-setting a possible set of models, the system is expected to operate as one of the models. Existing solutions for the MMAE problem are to use a joint Lainiotis Kalman filter (LKF) [9] in static mode, and an interacting multiple model filter (IMM filter) [11] in dynamic mode switching, respectively. In the multiple model setting, the state space model (2) can be represented by considering the mode state θ_t, such that:

$$\begin{aligned} &x_{t+1} = A_t x_t + w_t, \\ &y_t^i = C_t^i\left(\theta_t^j\right)x_t + v_t^i\left(\theta_t^j\right), \\ &t = 0,1,\ldots,\quad i = 1,\ldots,N,\quad j = 1,2, \end{aligned} \tag{8}$$

where θ_t represents the time-varying model of an observation system, as shown in Eq. (9):

$$\theta_t^j = \begin{cases} 1, & j = 1 \text{ (signal present)} \\ 0, & j = 2 \text{ (signal absent).} \end{cases} \tag{9}$$

Under this assumption, the multiple model observation system defined by Equations (8)–(9) basically includes the intermittent observation model in (1). Solutions given in LKF and IMM can then utilize the likelihood probability $p\left(y_t^i \middle| \theta_t^j\right)$ to determine the current mode, which is subsequently used to obtain an accurate state estimate. Given the initial prior probability of each model $p\left(\theta_0^j\right)$, the recursion for posterior probability, using Bayes rule, is given as shown in Eq. (10):

$$p\left(\theta_{t+1}^j \middle| y_{t+1}^i\right) = \frac{p\left(y_{t+1}^i \middle| \theta_{t+1}^j\right)}{\sum_{j=1}^{2} p\left(y_{t+1}^i \middle| \theta_{t+1}^j\right)} p\left(\theta_{t+1}^j \middle| y_t^i\right),\quad j = 1,2. \tag{10}$$

The likelihood probability $p\left(y_{t+1}^i \middle| \theta_{t+1}^j\right)$ can then be calculated from the normalized residual, as shown in (11):

$$\begin{aligned} &p\left(y_{t+1}^i \middle| \theta_{t+1}^j\right) = \left|\bar{P}_{t+1}^i\left(\theta_{t+1}^j\right)\right|^{-1/2} \exp\left(-\left(\tilde{y}_{t+1}^i\left(\theta_{t+1}^j\right)\right)^T \left(\bar{P}_{t+1}^i\left(\theta_{t+1}^j\right)\right)^{-1} \tilde{y}_{t+1}^i\left(\theta_{t+1}^j\right)\right), \\ &\bar{P}_{t+1}^i\left(\theta_{t+1}^j\right) = C_{t+1}^i\left(\theta_{t+1}^j\right) M_t \left(C_{t+1}^i\left(\theta_{t+1}^j\right)\right)^T + R^i, \\ &\tilde{y}_{t+1}^i\left(\theta_{t+1}^j\right) = y_{t+1}^i - C_{t+1}^i\left(\theta_{t+1}^j\right) x_t^i. \end{aligned} \tag{11}$$

In this paper, we have proposed using MMAE solutions in distributed sensor networks to handle intermittent observations efficiently. To estimate the state vector in intermittent observation circumstances adaptively, an IMM-based KCF, and a sliding window-based KCF have been proposed. The IMM-based approach for intermittent observations can be more effective, as the observation mode is arbitrarily switched without knowing the switching times. However, this approach requires both prior knowledge of the probability of switching between modes and additional computations. In the second approach, the transition matrix and additional computations have been avoided by using a sliding window-type algorithm instead of sacrificing a little accuracy. Details of the proposed algorithms have been presented in the following section.

4. PROPOSED ALGORITHMS

4.1 Basic Framework

By incorporating IMM and the sliding window approach into the filtering algorithm, we have proposed two algorithms to adjust intermittent observations in the distributed sensor network adaptively. In the IMM-based approach, intermittent observations have been handled by calculating the mode probability of the observation system (8)–(9) using a special mixing process involving mode probability, estimates, and covariance.

To address the computational complexity issue inherent in the IMM approach, the second method (sliding window approach) has been proposed, which considers a recent observation set from the i^{th} sensor $y_{t-\Delta:t}^i = \left\{y_{t-\Delta}^i, y_{t-\Delta+1}^i, \ldots, y_t^i\right\}$. The purpose is to calculate mode probability $p\left(\theta_t^j \middle| y_{t-\Delta:t}^i\right)$, where Δ is the window length. This approach can be seen as a modified version of the LKF algorithm.

When the mode probability $\mu_t^i \quad p\left(\theta_t \middle| y_t^i\right)$ is available, it can be incorporated into the information fusion filtering equation to handle the intermittent communication channels, as shown in Equations (12) and (13):

Observation Update:

$$\begin{aligned} &S_t = \sum_{i=1}^{N} \underbrace{\mu_t^i \left(C_t^i\right)^T \left(R^i\right)^{-1} C_t^i}_{U_t^i}, \\ &z_t = \sum_{i=1}^{N} \underbrace{\mu_t^i \left(C_t^i\right)^T \left(R^i\right)^{-1} y_t^i}_{u_t^i}, \\ &M_t = \left(P_t^{-1} + S_t\right)^{-1},\quad \hat{x}_t = \bar{x}_t + M_t\left[z_t - S_t \bar{x}_t\right], \end{aligned} \tag{12}$$

Time Update:

$$\bar{x}_{t+1} = A_t \hat{x}_t, \tag{13}$$

Under the modified information filtering framework of equations (12)–(13), we have proposed two algorithms for distributed Kalman filtering that have intermittent observations, by calculating mode probability using IMM and sliding window-based LKFs, respectively.

4.2 Distributed Information Fusion Filtering with Intermittent Observation via the IMM Approach

Information fusion filters have often been used for decentralized fusion algorithms in sensor networks, but can only be used in local sensor nodes in a large-scale network. The scalability and topology of these networks are not typically considered, even though they are crucial factors in real situations. To satisfy these requirements, a distributed Kalman filtering algorithm that uses a consensus algorithm—which has been referred to as a Kalman-Consensus filter (KCF)—was proposed recently [6].

Unlike other data fusion algorithms, there is no fusion center in the KCF; individual sensor nodes calculate their own state estimates instead, and communicate messages (contribution terms and local estimates for each node) to make a global agreement that converges to a certain value. As mentioned in the introduction, the scalability and topology of this network can be understood using algebraic graph theory [6].

In brief, suppose there is a large scale network with an ad-hoc topology, described by the undirected graph $G=(V,E)$, and N nodes. Vertices $V=\{1,2,...,N\}$ denote the sensor nodes, and edges $E\subset V\times V$, refer to the communication links between them. An example of a large-scale sensor network of randomly distributed sensors is displayed in Figure 1. Here, the KCF serves as a micro-filter for a network that only shares messages with its neighbors $L_i : J_i = L_i \cup \{i\}$.

The IMM filter is a well-known method for MMAE problems requiring dynamic mode switching. Therefore, a natural distributed information fusion filtering solution for having intermittent observations is to embed the IMM method into a KCF micro-filter. The first proposed algorithm (IMM-based adaptive KCF (AKCF)) is described below, as Algorithm 1.

Algorithm 1: IMM-based AKCF of node i

Given $P_t^{i,j}$ $\bar{x}_t^{i,j}$, parameter ε, mode probability $p\left(\theta_{t-1}^j \middle| y_{t-1}^i\right)$, and the transition matrix π_{kj}, between $p\left(\theta_{t-1}^k \middle| y_{t-1}^i\right)$ and $p\left(\theta_{t-1}^j \middle| y_{t-1}^i\right)$ where $j=1,2, \quad k=1,2.$

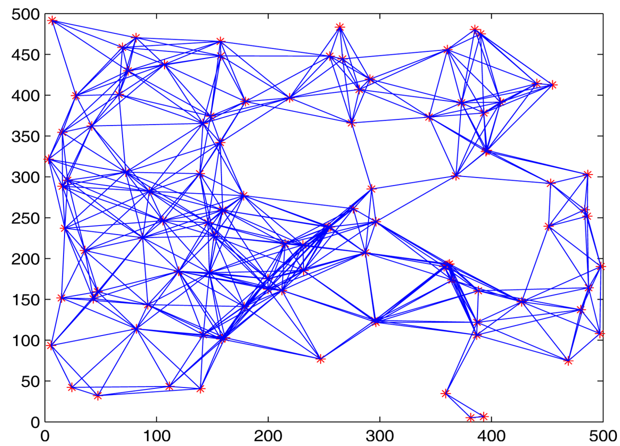

Fig. 1. A large-scale sensor network with an ad-hoc topology (100 nodes)

1. Predicted mode probability

$$p\left(\theta_t^j \middle| y_{t-1}^i\right)=\sum\nolimits_{k=1}^{2}\pi_{kj}\,p\left(\theta_{t-1}^j \middle| y_{t-1}^i\right)$$

2. Mixing weight

$$p\left(\theta_{t-1}^k \middle| \theta_t^j, y_{t-1}^i\right)=\pi_{kj}\,p\left(\theta_{t-1}^k \middle| y_{t-1}^i\right) / \, p\left(\theta_t^j \middle| y_{t-1}^i\right)$$

3. Mixing estimate

$$\bar{x}_{t-1}^{i,j}=E\left[x_{t-1}^i \middle| \theta_t^j, y_{t-1}^i\right]=\sum\nolimits_{k=1}^{2}\bar{x}_{t-1}^{i,k}\,p\left(\theta_{t-1}^k \middle| \theta_t^j, y_{t-1}^i\right)$$

4. Mixing covariance

$$P_{t-1}^{i,j}=\sum\nolimits_{k=1}^{2}(P_{t-1}^{i,k}+\left(\bar{x}_{t-1}^{i,j}-\hat{x}_{t-1}^{i,k}\right)$$
$$\times\left(\bar{x}_{t-1}^{i,j}-\hat{x}_{t-1}^{i,k}\right)^T)\,p\left(\theta_{t-1}^{i,k} \middle| \theta_t^{i,j}, y_{t-1}^i\right)$$

5. KCF procedure

A. Obtain measurement $y_t^i=C_t^i\left(\theta_t^j\right)x_t+v_t^i, \quad i=1,...,N.$

B. Calculate mode likelihood $p\left(y_t^i \middle| \theta_t^j\right)$ using (11)

C. Update the mode probability as

$$p\left(\theta_t^j \middle| y_t^i\right)=\frac{p\left(y_t^i \middle| \theta_t^j\right)}{\sum_{j=1}^{2} p\left(y_t^i \middle| \theta_t^j\right) p\left(\theta_t^j \middle| y_{t-1}^i\right)}\,p\left(\theta_t^j \middle| y_{t-1}^i\right)$$

D. Compute the contribution term from the information state and matrix, such that

$$u_t^{i,j}=p\left(\theta_t^j \middle| y_t^i\right)\left(C_t^i\right)^T\left(R^i\right)^{-1}y_t^i$$

$$U_t^{i,j}=p\left(\theta_t^j \middle| y_t^i\right)\left(C_t^i\right)^T\left(R^i\right)^{-1}C_t^i.$$

E. Broadcast message $m_t^{i,j}=\left(u_t^{i,j},\, U_t^{i,j},\, \bar{x}_t^{i,j}\right)$ to neighbors in L_i

F. Collect messages $m_t^{r,j}=\left(u_t^{r,j}, U_t^{r,j}, \bar{x}_t^{r,j}\right)$ from neighbors

G. Aggregate neighbor information states and matrices, including node i: $J_i=L_i\cup\{i\}$.

$$z_t^{i,j}=\sum\nolimits_{r\in J_i}u_t^{r,j}, \quad S_t^{i,j}=\sum\nolimits_{r\in J_i}U_t^{r,j}$$

H. Compute the Kalman-Consensus estimate

$$M_t^{i,j}=\begin{cases}\left(\left(P_t^{i,j}\right)^{-1}+S_t^{i,j}\right)^{-1}, & j=1,\\ \left(\left(P_t^{i,j}\right)^{-1}\right)^{-1}, & j=2,\end{cases},$$

$$\hat{x}_{t+1}^{i,j}=\bar{x}_t^{i,j}+M_t^{i,j}\left(z_t^i-S_t^{i,j}\bar{x}_t^{i,j}\right)+\varepsilon\frac{M_t^{i,j}}{1+\left\|M_t^{i,j}\right\|}\sum\nolimits_{r\in J_i}\left(\bar{x}_t^{r,j}-\bar{x}_t^{i,j}\right)$$

I. Update stage

$$P_{t+1}^{i,j}\leftarrow A_t M_t^{i,j}A_t^T+Q,$$
$$\bar{x}_{t+1}^{i,j}\leftarrow A_t\hat{x}_{t+1}^{i,j}$$

J. Final estimate

$$\hat{x}_{t+1}^i=\sum\nolimits_{j=1}^{2}p\left(\theta_t^j \middle| y_t^i\right)\hat{x}_{t+1}^{i,j}$$

As shown in Algorithm 1, every node in the IMM-based AKCF requires all the above calculations for each mode. In addition, mixing for weight and estimate is required to adjust for arbitrary mode switching. However, mode switching does not occur frequently in real situations, because the loss of packets can be thought of as a rare event. Under this assumption, we proposed our second algorithm to reduce computational cost while still showing reasonable performance, based on using LKF with a sliding window approach.

4.3 Distributed Information Fusion Filtering with Intermittent Observation via a Sliding Window-based LKF

From a static mode case, initialization of the mode probability needs to be considered to implement LKF in the dynamic mode switching case. In this proposed method, we set the window to a fixed size, which was receding and being processed to calculate mode probability.

In brief, details of the algorithm are as follows: given the initial prior probability, using (10) the mode probability was calculated until it converged. The initial length of window Δ was then set as the current time of convergence, and the sliding window containing the most recent set of observations was created, and started to calculate mode probability $p\left(\theta_t^j \middle| y_{t-\Delta:t}^i\right)$.

For every sliding window, we set the initial probability as shown in equation (14):

$$p\left(\theta_{t-\Delta}^j \middle| y_{t-\Delta}^i\right) = \begin{cases} \alpha, & j=1, \\ 1-\alpha, & j=2, \end{cases} \tag{14}$$

where α is the prior probability of the arrival of an observation. Figure 2 illustrates the sliding window scheme for calculating mode probability. In the proposed approach, the mode probability asymptotically converges (either at the signal presence or signal absence) for the threshold for the fixed window size, under the mild assumption mentioned in the previous section.

Algorithm 2: Sliding window-based AKCF of node i

Given P_t^i, $\bar{x}_t^i$, parameter ε

1. Obtain measurement $y_t^i = C_t^i(\theta_t^j)x_t + v_t^i,\ i = 1,\ldots,N$

2. Calculate mode probability Given $p\left(\theta_{t-\Delta}^j \middle| y_{t-\Delta}^i\right)$

For $s = t-\Delta+1 : t$ $p\left(\theta_t^j \middle| y_{t-\Delta:t}^i\right)$

Evaluate (11) for y_s^i

Evaluate the Bayes recursion (10), where $p\left(\theta_s^j \middle| y_{s-1}^i\right)$ $p\left(\theta_{s-1}^j \middle| y_{s-1}^i\right)$

End

3. Compute the contribution term for the information state and matrix, such that

$$u_t^i = \sum_{j=1}^{2} p\left(\theta_t^j \middle| y_{t-\Delta:t}^i\right)\left(C_t^i\right)^T \left(R^i\right)^{-1} y_t^i$$

$$U_t^i = \sum_{j=1}^{2} p\left(\theta_t^j \middle| y_{t-\Delta:t}^i\right)\left(C_t^i\right)^T \left(R^i\right)^{-1} C_t^i.$$

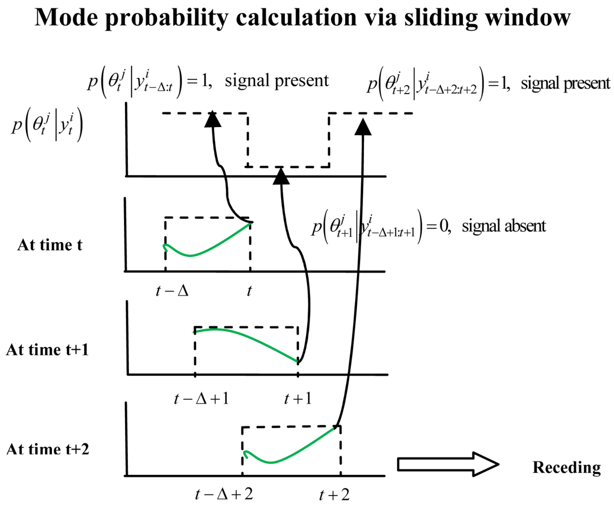

Fig. 2. Sliding window approach to determining mode probability calculations at individual sensors

4. Broadcast message $m_t^i = \left(u_t^i,\ U_t^i,\ \bar{x}_t^i\right)$ to neighbors in L_i

5. Collect messages $m_t^r = \left(u_t^r, U_t^r, \bar{x}_t^r\right)$ from neighbors

6. Aggregate the information states and matrices from neighbors, including node: $J_i = L_i \cup \{i\}$.

$$z_t^i = \sum_{r\in J_i} u_t^r,\quad S_t^i = \sum_{r\in J_i} U_t^r$$

7. Compute the Kalman-Consensus estimate

$$M_t^i = \left(\left(P_t^i\right)^{-1} + S_t^i\right)^{-1},$$

$$\hat{x}_{t+1}^i = \bar{x}_t^i + M_t^i\left(z_t^i - S_t^i\bar{x}_t^i\right) + \varepsilon\frac{M_t^i}{1+\left\|M_t^i\right\|}\sum_{r\in J_i}\left(\bar{x}_t^r - \bar{x}_t^i\right)$$

8. Update stage

$$P_{t+1}^i \leftarrow A_t M_t^i A_t^T + Q,$$

$$\bar{x}_{t+1}^i \leftarrow A_t \hat{x}_{t+1}^i$$

It should be noted that the sliding window-based AKCF does not inherently take into account the mixing procedure as in the IMM method; hence, it is less adaptive when the mode is quickly switching. However, it does have significant advantages in terms of computational time, because the filter bank is not deployed. Therefore, the sliding window approach enables moderate observation mode switching, as it monitors the temporal history of mode probability.

5. EXPERIMENTAL RESULTS

To validate the advantages of the proposed algorithms and then compare them, a target tracking example has been considered.

Given the target dynamics of a circular movement

$$x_{t+1} = Ax_t + Bw_t$$

where $A_0 = 2\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, $B_0 = 5^2 I_2$,

$$A = I_2 + \varepsilon A_0 + \frac{\varepsilon^2}{2} A_0^2 + \frac{\varepsilon^2}{6} A_0^3, \text{ and } B = \varepsilon B_0$$

In addition, I_2 is a 2 x 2 identity matrix, which is a discretized model with a step-size, ε, = 0.015, and the initial position and uncertainty are $x_0 = (15, -10)^T$, and $P_0 = 10 I_2$, respectively. A moving target having a circular motion can then be observed via the large-scale sensor network in Figure 1, with 100 sensor nodes. Here, the sensor nodes measure the target position with intermittent sensor observations linked to the node, i.e.,

$$y_t^i = C_t^i\left(\theta_t^j\right)x_t + v_t^i, \quad t = 0,1,..., \quad i = 1,...,100, \quad j = 1,2,$$

where either $C_t^i\left(\theta_t^j\right) = \begin{cases} [1 \quad 0], & j = 1 \\ [0 \quad 0], & j = 2 \end{cases}$, or

$C_t^i\left(\theta_t^j\right) = \begin{cases} [0 \quad 1], & j = 1 \\ [0 \quad 0], & j = 2 \end{cases}$. In this case, the data loss in the communication channel link is modelled using link failure probability—that is, $P\left(\theta_t^{j=2}\right) \equiv P\left(u(0,1) < 0.01\right)$—where $u(0,1)$ is a uniform random distribution. Of course, link failure probability can be modeled as a Markov chain, and in such a case, the proposed algorithms would be expected to be more efficient.

The observation noise for each sensor is white Gaussian noise, with $v_t^i \sim N\left(0, 30^2\sqrt{i}\right)$. For the sliding window-based AKCF, the window length, Δ, was set at 3.

In this example, the target was not fully observable by individual sensors, and intermittent observations occurred randomly. Figure 3 illustrates the status of links in the communication channels of selected node pairs during the experimentation period. From a practical viewpoint, the arbitrary switching model is reasonable for describing intermittent observations, as we really do not know exactly when there are packet losses in channels. In intermittent observation situations, estimated trajectories for the sliding window-based AKCF and KCF have been compared in Figure 4.

The experimental results clearly show that KCF performance was seriously degraded by the effect of intermittent observations. In contrast, AKCF adaptively adjusted for intermittent observations, thereby allowing it to estimate the object's position accurately.

The same experiment was then performed using the IMM-based AKCF, and the results obtained by comparing the simulation outcome with the other algorithms have been displayed in Figure 5. Whereas tracking accuracies were easily compared between the KCF and sliding window-based AKCF in Figure 4, the two AKCF algorithms showed almost the same performance, and so their performance evaluations have been displayed based on mean square error (MSE), calculated using 1000 Monte Carlo runs. Note that in every Monte Carlo run, the distributed sensor networks for different topologies varied, and communication channel conditions were randomly selected. The MSE comparison confirmed that the two proposed

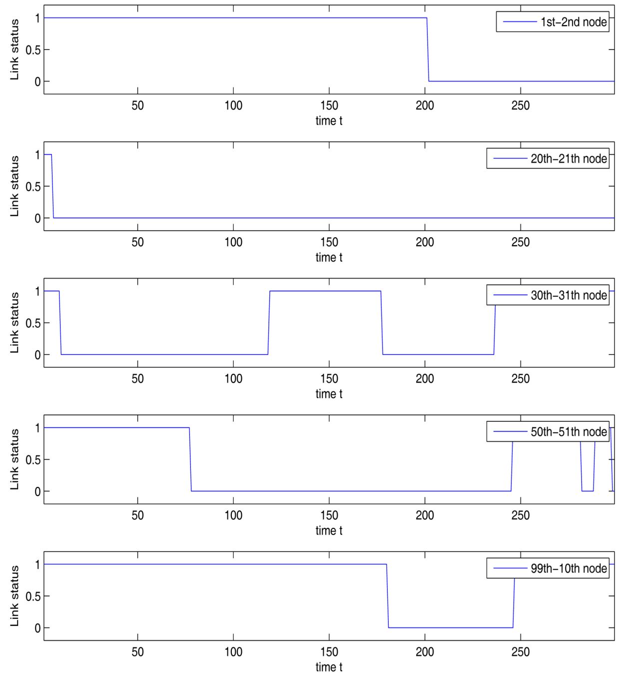

Fig. 3. Link status of selected sensor nodes in the example

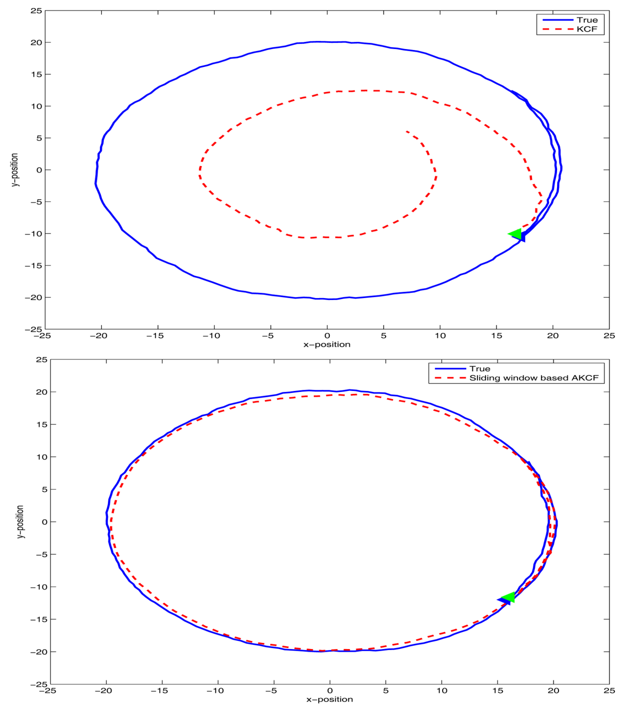

Fig. 4. Estimated trajectory comparison (KCF vs Sliding window-based AKCF)

algorithms (IMM-based AKCF and sliding window-based AKCF) are robust and accurate when faced with intermittent observations (in a distributed sensor network). In addition, under mild conditions (intermittent observations rarely occuring), the sliding window-based

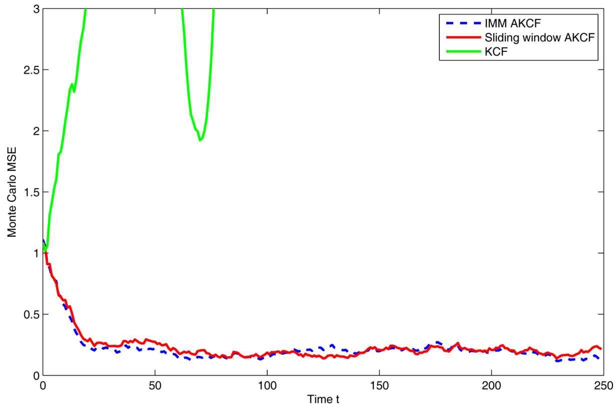

Fig. 5. Monte Carlo MSE comparison

AKCF handled uncertainty in communication channels efficiently, with fewer computations required compared to the IMM-based approach and without loss of advantages.

Note that the two proposed algorithms can be used in a complementary manner: when switching is frequent, use the IMM-based algorithm, and when rare, use the sliding window-based algorithm.

6. CONLUSION

In this paper, the state estimation problem for a large-scale sensor network with intermittent observations has been discussed. Two adaptive algorithms were subsequently suggested to alleviate the inaccuracies caused by imperfect communication links in this type of sensor network. Unlike the approach used in other studies, the proposed approach automatically manages data loss in the channel without requiring additional indicators. Using a target tracking example under a reasonable assumption, significant improvements have been confirmed. Further, by using an alternative method, computational complexity was reduced without degrading the perceived improvements.

REFERENCES

[1] A. Makarenko and H. F. Durrant-Whyte, "Decentralized data fusion and control in active sensor networks", *Int. Conf. on Inf. Fusion,* pp. 479-486, Stockholm, Sweden, 2004.

[2] H. Medeiros, J. Park, and A. C. Kak, "Distributed object tracking using a cluster-based Kalman filter in wireless camera networks", *IEEE J. Sel. Top. Signal Process.*, Vol. 2, No. 4, pp. 448-463, 2008.

[3] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters", *IEEE Trans. Automat. Contr.*, Vol. 55, No. 1, pp. 32-47, 2010.

[4] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance", *Int. J. Rob. Res.*, Vol. 12, No. 1, pp. 20-44, 1993.

[5] P. Alriksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging", *Proc. 17th Int. Symp. Math. Theory Netw. Syst.*, pp. 8179-8184, Kyoto, Japan, 2006.

[6] R. Olfati-Saber and J. S. Shamma, "Distributed Kalman filtering for sensor networks", *46th IEEE Conf. on Decis. Control*, pp. 5492-5498, City, Country, 2007.

[7] Y. Bar-Shalom, "Update with out-of-sequence measurements in tracking: Exact solution", *IEEE Trans. Aerosp. Electron. Syst.*, Vol. 38, No. 3, pp. 769-778, 2002.

[8] B. Sinopoli, L. Schenato, M. Franseschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations", *IEEE Transactions on Automat. Contr.*, Vol. 49, No. 9, pp. 1453-1464, 2004.

[9] D. G. Lainiotis, "Partitioning: A unifying framework for adaptive systems, I: Estimation", *Proc. IEEE*, Vol. 64, No. 8, pp. 1126-1143, 1976.

[10] L. Shi, "Kalman filtering over graphs: Theory and applications", *IEEE Transactions on Automat. Contr.*, Vol. 54, No, 9, pp. 2230-2234, 2009.

[11] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with markovian switching coefficients", *IEEE Transactions on Automat. Contr.*, Vol. 33, No, 8, pp. 780-783, 1988.