Distributed Information Fusion Filter with Intermittent Observations

Du Yong Kim, Ju Hong Yoon, Yong Hoon Kim, Vladimir Shin
Dept. of Mechatronics
Gwangju Institute of Science and Technology
Gwangju, Korea
{duyong, jhyoon, yhkim, vishin}@gist.ac.kr

Abstract - We present a robust distributed fusion algorithm with intermittent observations via an interacting multiple model (IMM) approach and sliding window strategy that can be applied to a large-scale sensor network. The communication channel is modelled as a jump Markov system and a posterior probability distribution for communication channel characteristics is calculated and incorporated into the filter to allow distributed Kalman filtering to automatically handle the intermittent observation situations. To implement distributed Kalman filtering, a Kalman-Consensus filter (KCF) is then used to obtain the average consensus based on the estimates of distributed sensors over a large-scale sensor network. From a target-tracking example for a large-scale sensor network with intermittent observations, the advantages of proposed algorithms are subsequently verified.

Keywords: Kalman filtering, distributed fusion, intermittent observation, estimation.

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 trying to utilize this concept in various disciplines; for example, data fusion in sensor networks [1], distributed camera networks [11], and mobile robotics [7]. In literature, the development of distributed signal processing algorithms is not a new topic. It has been investigated over the past few years, and several types of distributed signal processing algorithms are well-known [3, 8].

Recently, real implementations of distributed signal processing algorithms have 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 [2]. To this end, the topology of a network can be understood via algebraic graph theory, with individual network nodes employing Kalman filter used to consider the limited communication bandwidth between neighbouring nodes.

Another important issue of distributed signal processing implementations are imperfect communication channels. Because a huge number of sensors are randomly distributed and communicate with each other through wireless channels, communication links occasionally break down and become unstable, thereby delaying observations and incurring packet losses. In previous researches, the communication delay problem has been investigated as an out-of-sequence measurement problem [4].

To model an unreliable communication channel, a latent variable for the observation system is considered. The arrival of the observation is controlled by a latent variable; under this formulation a statistical convergence analysis was performed in [9]. The intermittent observation is modelled as a conditional probability distribution

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

where v_t, γ_t, R, σ are zero mean white Gaussian observation noise, latent variable, noise covariance with no loss, and unreliable noise deviation, i.e., $\sigma\to\infty$ means the absence of observation, respectively. According to the research conducted under this formulation, the latent variable γ_t is assumed to be a Bernoulli process and depending on the state space model there exists a critical arrival probability value at which the estimation error covariance is bounded [9]. Because the algebraic Ricatti equation becomes a stochastic differential equation, only bound analysis is 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 hyper-parameter of observation noise, the characteristics of the communication channel are modelled using the multiple model adaptive estimation (MMAE) approach [6]. To continuously adjust the observation mode switching, we propose two algorithms. First, an interacting multiple model (IMM) filter is applied to solve the MMAE problem. Second, by using a sliding window, the posterior probability of link failure is calculated and incorporated into the information fusion filtering. The sliding window collects the most recent set of observations which are then sequentially processed to calculate the posterior probability of the mode of observation (absence or presence). Two MMAE solutions are implemented in distributed Kalman filtering to ensure that intermittent observation situations are efficiently handled in a large-scale sensor network.

The remainder of paper is organized as follows. Section 2 highlights the problem formulation. Preliminaries including information fusion filtering and MMAE are then given in Section 3. We provide details of the proposed algorithm in Section 4, and advantages of the proposed algorithm are evaluated in Section 5. Finally we conclude this paper in Section 6.

2 Problem formulation

Consider the discrete-time dynamical 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 is to obtain an accurate estimate given multiple observations, i.e., $E\left(x_t \left| y_t^1,\ldots,y_t^N\right.\right)$, under an unreliable communication channel. In the proposed algorithm, we estimate another state referred to as the communication characteristic state θ_t to cope with intermittent observations. In this framework, each sensor calculates the posterior probability of mode $p\left(\theta_t \left| y_t^i\right.\right)$, which will then be incorporated into the Kalman filtering equation as a characteristic of the communication channel at the current time.

On the other hand, in multiple sensory environments, a central fusion scheme is intuitive, where all observations are collected at one centre 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 is not suitable [3]. To satisfy these conditions, a decentralized or distributed fusion algorithm has been suggested and is currently the focus of rigorous investigations. In a state estimation under multiple sensors in a flexible network environment, having an efficient distributed signal processing algorithm is essential in practical point of view.

Considering the target system in (2), decentralized Kalman filtering is known to be globally optimal under perfect communication conditions [1]. Recently, however, following the initial work reported the state estimation with intermittent observations [9], the problem has been extended to multiple sensory systems and discussed based on a graphical understanding of the network [10].

3 Preliminaries

In this preliminary section, an information fusion filter [1, 3] is introduced as a basic tool for fusing distributed sensors over the network. And the multiple model adaptive estimation is 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

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

Then, the information form 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. 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

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

$$z_t = C_t^T R^{-1} Y_t = \sum_{i=1}^{N} \left(C_t^i\right)^T \left(R^i\right)^{-1} y_t^i. \quad (7)$$

Therefore, (4)-(7) define the information fusion filter that will be used in the proposed algorithm for distributed fusion. Because of mathematical equivalence, optimality of the distributed fusion algorithm is guaranteed.

3.2 Multiple model adaptive estimation

When a system has parametric uncertainties it can be modelled with a set of multiple models. A well-known example of multiple models in the state estimation is the tracking problem for manoeuvring targets. Target manoeuvres have a set of distinctive models, for example, constant velocity, constant acceleration, turning motion, etc. 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 Lainiotis Kalman filter (LKF) [6] in static mode and interacting multiple model filter (IMM filter) [5] 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

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

where θ_t represents the time-varying model of observation system as

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

Under this assumption, the multiple model observation system defined in (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 (10)

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

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

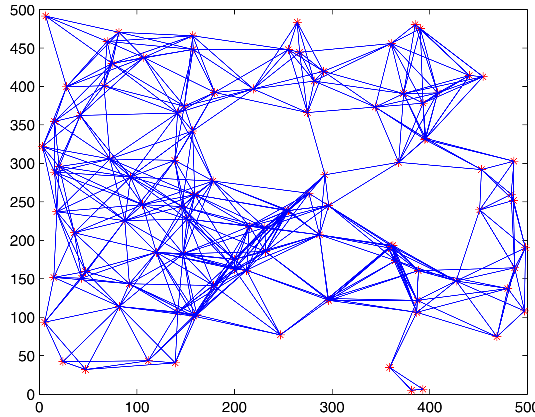

Figure 1. A large-scale sensor network with an ad-hoc topology (100 nodes)

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

In this paper we propose the use of MMAE solutions in distributed sensor networks to efficiently handle intermittent observations. To adaptively estimate the state vector with intermittent observation circumstances, an IMM-based KCF and a sliding window-based KCF are proposed. The IMM-based approach for intermittent observations can be more effective because the observation mode is arbitrarily switched without knowing the switching times. But 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 are avoided by using a sliding window-type algorithm instead of sacrificing a little accuracy. Details of the proposed algorithms are given in the following section.

4 Proposed algorithm

4.1 Basic framework

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

To reduce the computational complexity of the IMM approach, the second method, the sliding window approach—which considers a recent observation set from the i^{th} sensor $y_{t-\Delta:t}^i = \left\{y_{t-\Delta}^i, y_{t-\Delta+1}^i, \ldots, y_t^i\right\}$ —is proposed to calculate the 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 $\mu_t^i \triangleq p\left(\theta_t \middle| y_t^i\right)$ is available it can be incorporated into the information fusion filtering equation to handle the intermittent communication channels as follows,

Observation Update;

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

Time Update;

$$
\begin{aligned}
\overline{x}_{t+1} &= A_t \hat{x}_t,\\
P_{t+1} &= A_t M_t A_t^T + Q,
\end{aligned}
\tag{13}
$$

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

4.2 Distributed information fusion filtering with intermittent observation via IMM approach

The information fusion filter has 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 was recently proposed that uses a consensus algorithm, referred to as a Kalman consensus filter (KCF) [2].

Unlike other data fusion algorithms, there is no fusion centre; insteads individual sensor nodes calculate their own state estimates and communicate messages (contribution terms and local estimate of each node) to make a global agreement to converge to a certain value. As mentioned in the introduction, the scalability and topology of this network can be understood using algebraic graph theory [2].

In brief, suppose there is a large scale network with an ad-hoc topology described by the undirected graph $G=\left(V,E\right)$ and N nodes. Vertices $V=\{1,2,...,N\}$ denote the sensor nodes and the edges $E \subset V \times V$, refer to the communication links between the sensor nodes. 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 of a network that only shares messages with its neighbours $L_i : J_i = L_i \cup \{i\}$.

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

Algorithm 1 IMM-based AKCF of node i

Given $P_t^{i,j}$ $\overline{x}_t^{i,j}$, 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 $j=1,2, \quad k=1,2.$

1. Predicted mode probability
$$p\left(\theta_t^j \middle| y_{t-1}^i\right) = \sum_{k=1}^{2} \pi_{kj}\, p\left(\theta_{t-1}^j \middle| y_{t-1}^i\right)$$
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
$$\overline{x}_{t-1}^{i,j} = E\left[x_{t-1}^i \middle| \theta_t^j, y_{t-1}^i\right] = \sum_{k=1}^{2} \overline{x}_{t-1}^{i,k}\, p\left(\theta_{t-1}^k \middle| \theta_t^j, y_{t-1}^i\right)$$
4. Mixing covariance
$$P_{t-1}^{i,j} = \sum_{k=1}^{2} (P_{t-1}^{i,k} + \left(\overline{x}_{t-1}^{i,j} - \hat{x}_{t-1}^{i,k}\right) \times \left(\overline{x}_{t-1}^{i,j} - \hat{x}_{t-1}^{i,k}\right)^T)\, p\left(\theta_{t-1}^{i,k} \middle| \theta_t^{i,j}, y_{t-1}^i\right)$$
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 of 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 the message $m_t^{i,j} = \left(u_t^{i,j},\, U_t^{i,j},\, \overline{x}_t^{i,j}\right)$ to neighbours in L_i.
   F. Collect the messages $m_t^{r,j} = \left(u_t^{r,j}, U_t^{r,j}, \overline{x}_t^{r,j}\right)$ from neighbors.
   G. Aggregate the information states and matrices of neighbours including node i: $J_i = L_i \cup \{i\}$.
   $$z_t^{i,j} = \sum_{r \in J_i} u_t^{r,j}, \quad S_t^{i,j} = \sum_{r \in J_i} U_t^{r,j}$$

H. Compute the Kalman-Consensus estimate

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

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

I. Update stage

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

$$\overline{x}_{t+1}^{i,j} \leftarrow A_t \hat{x}_{t+1}^{i,j}$$

J. Final estimate

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

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 needed in order to adjust the arbitrary switching of modes. However, the mode switching does not frequently occur in real situation, because the loss of packets can be thought of as a rare event. Under this assumption, we subsequently propose the second algorithm to reduce the computational cost and still show reasonable performance based on the use of LKF with sliding a window approach.

4.3 Distributed information fusion filtering with intermittent observation via a sliding window-based LKF

LKF is basically the solution of the MMAE problem for the static mode case. Therefore, to implement LKF in the dynamic mode switching case, initialization of the mode probability should be considered. In the proposed method, we set the window to a fixed size, which is receding and being processed to calculate the mode probability. In brief, details of the algorithm are as follows.

Given the initial prior probability, using (10) the mode probability is calculated until it converges. The initial length of window Δ is then set as the current time of convergence, and the sliding window that contains the most recent set of observations is created and starts 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,

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

where α is the prior probability of arrival of observation. Figure 2 illustrates the sliding window scheme for calculating the 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.

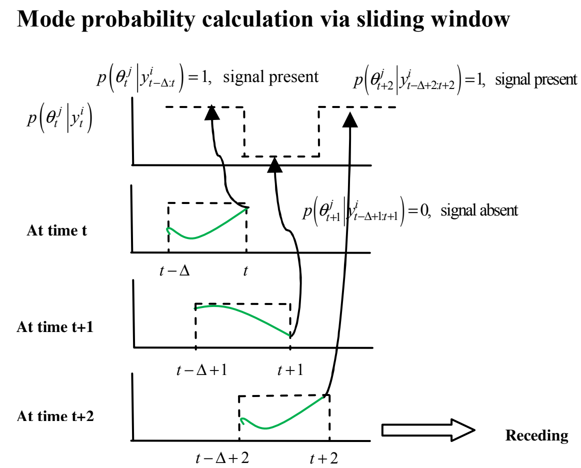

Figure 2. Sliding window approach to determine mode probability calculation at individual sensors

Algorithm 2 Sliding window-based AKCF of node i

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

1. Obtain measurement $y_t^i = C_t^i\left(\theta_t^j\right) x_t + v_t^i, \quad i = 1, \ldots, N.$
2. Calculate mode probability $p\left(\theta_t^j \middle| y_{t-\Delta:t}^i \right)$

   Given $p\left(\theta_{t-\Delta}^j \middle| y_{t-\Delta}^i \right)$,

   For $s = t - \Delta + 1 : t$

   Evaluate (11) for y_s^i.

   Evaluate the Bayes recursion (10),

   where $p\left(\theta_s^j \middle| y_{s-1}^i \right) \triangleq p\left(\theta_{s-1}^j \middle| y_{s-1}^i \right)$.

   End
3. Compute contribution term of information state and matrix such that

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

   $$U_t^i = \sum_{j=1}^{2} p\left(\theta_t^j \middle| y_{t-\Delta:t}^i \right) \left(C_t^i \right)^T \left(R^i \right)^{-1} C_t^i.$$
4. Broadcast message $m_t^i = \left(u_t^i,\ U_t^i,\ \overline{x}_t^i \right)$ to neighbours in L_i.
5. Collect messages $m_t^r = \left(u_t^r, U_t^r, \overline{x}_t^r \right)$ from neighbours.
6. Aggregate the information states and matrices of neighbours including node i: $J_i = L_i \cup \{i\}$.

   $$z_t^i = \sum_{r \in J_i} u_t^r, \quad S_t^i = \sum_{r \in J_i} U_t^r$$
7. Compute the Kalman-Consensus estimate

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

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

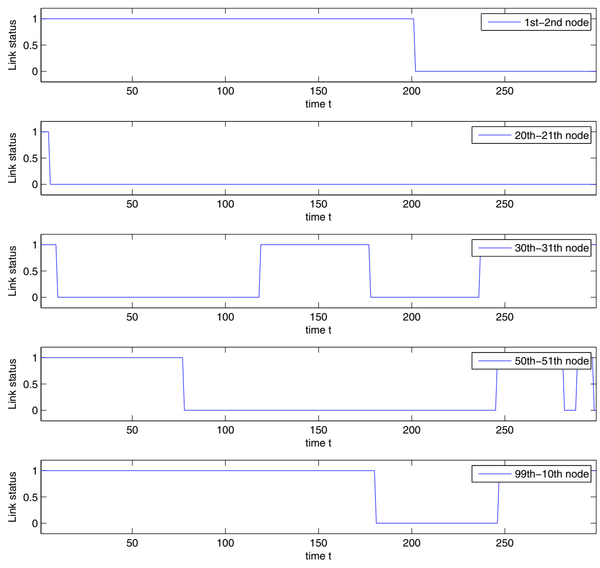

Figure 3. Link status of selected sensor nodes in the example

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

It should be noted that the sliding window-based AKCF does not inherently take into account the mixing procedure as in the IMM method so that it is less adaptive when the mode is quickly switching. However, it has 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 the mode probability.

5 Experimental results

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

Given the target dynamics of a circular movement

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

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

$A = I_2 + \varepsilon A_0 + \frac{\varepsilon^2}{2} A_0^2 + \frac{\varepsilon^2}{6} A_0^3$, 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 $\varepsilon = 0.015$, 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.,

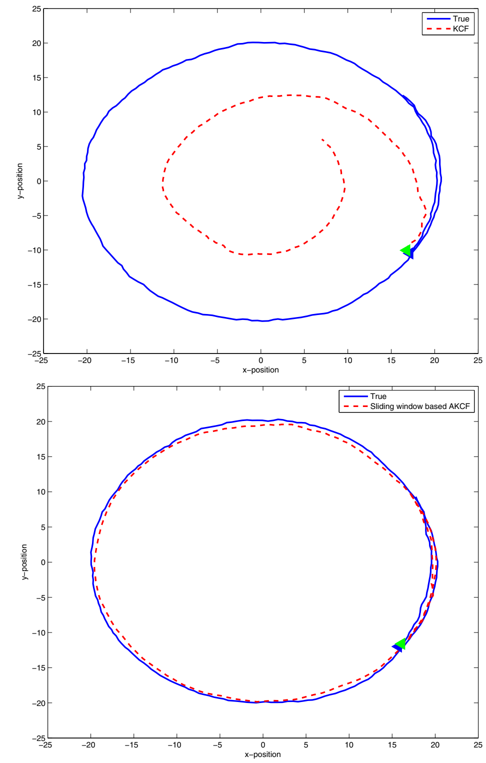

Figure 4. Estimated trajectory comparison (KCF and Sliding window-based AKCF)

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

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 based on link failure probability, i.e., $P\left(\theta_t^{j=2}\right) \equiv P\left(u(0,1) < 0.01\right)$, where $u(0,1)$ is a uniform random distribution. Of course, link failure probability can be modeled as a Markov chain. In such case, proposed algorithms are 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 is set at 3.

In this example, the target is not fully observable by individual sensors and intermittent observations occur randomly. Figure 3 illustrates the status of links in communication channel of selected node pairs during the experimentation time. From a practical point of view, the arbitrary switching model is reasonable for describing intermittent observations because we really do not know exactly when there are packet losses in channels. In intermittent observation situations, estimated trajectories of the sliding window-based AKCF and KCF are compared in Figure 4. The experimental results clearly show that performance of KCF is seriously degraded due to the effect of intermittent observations. In contrast to the KCF, the proposed AKCF adaptively adjusts for

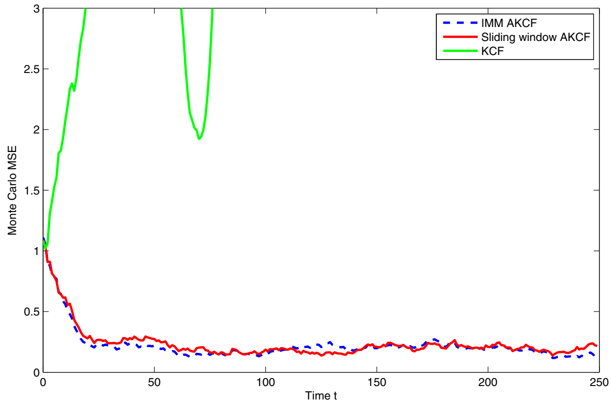

Figure 5. Monte Carlo MSE comparison

intermittent observations, thereby allowing it to accurately estimate the object position.

The same experiment was then performed for the IMM -based AKCF and Figure 5 compares the simulated result with other algorithms. Whereas tracking accuracies were easily compared between KCF and sliding window based AKCF in Figure 4, the two AKCF algorithms show almost the same performances. Hence, the performance evaluations are displayed based on the mean square error (MSE) calculated based on 1000 Monte Carlo runs. Note that in every Monte Carlo run, distributed sensor networks for different topologies are realized and communication channel conditions are randomly selected. The MSE comparison confirms that the two proposed algorithms (IMM-based AKCF and sliding window-based AKCF) are robust and accurate against intermittent observations in a distributed sensor network. In addition, under mild conditions (intermittent observation rarely occur), the sliding window-based AKCF efficiently handles uncertainty in communication channels with fewer computations required compared to the IMM-based approach, and without no loss of advantages. Note that two proposed algorithms can be used in complementary manner that when switching is frequent: IMM-based, and rare: sliding window-based algorithm, respectively.

6 Conclusion

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

7 Acknowledgement

This work is partially supported by ADD project (contract no. 912176201) and by BK 21 program at GIST.

References

[1] A. Makarenko and H. Durrant Whyte, Decentralized data fusion and control in active sensor networks, in *International Conference on Information Fusion*, Stockholm, Sweden, June, 2004.

[2] R. Olfati-Saber and J. S. Shamma, Distributed Kalman filtering for sensor networks, *46th IEEE Conference on Decision and Control,* 2007.

[3] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, A fully decentralized multi-sensor system for tracking and surveillance, *International Journal of Robotics Research, 12(1):20-44,* Feb 1993.

[4] Y. Bar-Shalom, Update with out-of-sequence measurements in tracking: Exact solution, *IEEE Transactions on Aerospace and Electronic Systems, 38(3):769-778,* July, 2002.

[5] H. Blom and Y. Bar-Shalom, The interacting multiple model algorithm for systems with markovian switching coefficients, *IEEE Transactions on Automatic Control, 33(8):780-783,* August, 1988.

[6] D. G. Lainiotis, Partitioning: A unifying framework for adaptive systems, I: Estimation, *Proc. Of IEEE, 64(8):1126-1143,* August, 1976.

[7] G. M. Hoffmann, C. J. Tomlin, Mobile sensor network control using mutual information methods and particle filters, *IEEE Transactions on Automatic Control, 55(1):32-47,* January, 2010.

[8] P. Alriksson and A. Rantzer, Distributed Kalman filtering using weighted averaging, in *Proc. 17th International Symposium Math. Theory Networks Sytems,* Kyoto, Japan, 2006.

[9] B. Sinopoli, L. Schenato, M. Franseschetti, K. Poolla, M. Jordan, and S. Sastry, Kalman filtering with intermittent observations, *IEEE Transactions on Automatic Control, 49(9):1453-1464,* September, 2004.

[10] L. Shi, Kalman filtering over graphs: Theory and applications, *IEEE Transactions on Automatic Control, 54(9):2230-2234,* September, 2009.

[11] H. Medeiros, J. Park, and A. C. Kak, Distributed object tracking using a cluster-based Kalman filter in wireless camera networks, *IEEE Journal of Selected Topics in Signal Processing, 2(4):448-463,* August, 2008.