Consensus-Based Distributed Batch Estimation in Asynchronous Wireless Sensor Networks

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 2050–2055

Consensus-Based Distributed Batch Estimation in Asynchronous Wireless Sensor Networks ⋆

Ming Li and Zhiyong Sun

Department of Electrical Engineering, Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands.
(E-mails: m.li3@ tue.nl, z.sun@tue.nl, sun.zhiyong.cn@gmail.com).

Abstract: In this paper, to address the distributed estimation problem over an asynchronous wireless sensor network (aWSN), an average consensus-based distributed batch estimation (DBE) method is proposed. The DBE seeks to update the global posterior with a predefined global update period (GUP) and is implemented with a local filter (LF) and a fusion filter (FF). For LF, we develop two different asynchronous batch estimation approaches to align and compute the asynchronous local posteriors of multiple nodes in an aWSN. At FF, an average consensus filter is adopted to compute the global posterior via a proposed DBE fusion rule. Numerical results show that the proposed DBE method has high target-tracking accuracy and is robust to strong asynchronism. Besides, the optimality of DBE fusion can be approximately achieved with a sufficiently large number of particles and consensus iterations.

Keywords: Distributed Estimation; Average Consensus; Asynchronous Wireless Sensor Network.

1. INTRODUCTION

Distributed estimation, which refers to a group of networked sensors collectively estimating the states, has widespread applications for wireless sensor networks (WSNs), such as environment monitoring (Corke et al. (2010)), autonomous navigation of unmanned aerial vehicles (Shima and Rasmussen (2009)), and localization in robotics (Li et al. (2020)). Among the vast investigations (Aysal et al. (2009); Dimakis et al. (2010); Olfati-Saber et al. (2007); Olshevsky and Tsitsiklis (2009)), the *consensus-based* distributed estimation algorithms, which aim to provide a global posterior by exchanging local posteriors among its neighbors throughout the WSN, have attracted considerable attentions recently.

However, the consensus-based distributed estimation algorithms usually suppose that WSN is synchronized (Li and Nehorai (2017)), wherein each node in a WSN has an identical sampling rate and initial timing offset, and the processing-and-communication delays are negligible (Kim et al. (2009); Vemula et al. (2007)). In practice, the synchronous assumption may not be satisfied and will always lead to a degradation of the estimation accuracy and result in divergence.

Based on our investigation, there are only a few papers that address the consensus-based distributed estimation problem in an asynchronous wireless sensor network (aWSN). In one such paper Giannini et al. (2013), the authors propose an asynchronous consensus-based distributed target tracking algorithm using an asynchronous iteration concept to overcome the asynchronism of the WSN. They introduce a maximum-consensus protocol to obtain uniform estimation results. Another efficient solution presented by Katragadda et al. (2017) is an average consensus-based asynchronous tracking filter, where forward prediction is used to ensure node synchronization. This method is named consensus-based sequential estimation (CSE) in the paper.

In contrast to the above approaches, we propose a *distributed batch estimation (DBE)* solution to address the distributed estimation problem in an aWSN. Since aWSNs may only need periodic updates to the surveillance system, our goal is to update global posteriors with a predefined global update period (GUP). In this paper, we present an average consensus-based DBE method that consists of two filters at each node: a *local filter (LF)* and a *fusion filter (FF)*. The LF implements DBE using two different batch estimation approaches: asynchronous Batch-1 (aBatch-1) or aBatch-2. For aBatch-1, we predict the prior information (global posterior available up to the last GUP) to the terminal instant of the current GUP and define a concept called local asynchronous likelihood function (LALF) to acquire time-aligned local likelihood functions. Using the predicted prior information and LALF, we obtain time-aligned local posteriors based on Bayes' rule. For aBatch-2, we use LF to compute the asynchronous local posteriors and predict them to the terminal instant of the current GUP to obtain time-aligned local posteriors. The proposed DBE fusion rule is implemented using an average consensus filter at FF. This results in the fusion of the time-aligned local posteriors, leading to the global posterior.

⋆ The work was partially supported by a starting grant from Eindhoven Artificial Intelligence Systems Institute (EAISI), Eindhoven, the Netherlands.

10.1016/j.ifacol.2023.10.1103

2. SYSTEM MODELS AND PROBLEM FORMULATION

2.1 Network Model

Consider a WSN consisting of N heterogeneous and geographically dispersed nodes. Each node has its own sensing, communication and local information processing capabilities. The network is described in terms of a bidirectional graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the links. In particular, (i, j) belongs to $\mathcal{A}$ if node i can receive data from node $j, j \in \{1, 2, ..., N\}$. For each node $i \in \mathcal{N}, \mathcal{N}_i \triangleq \{j \in \mathcal{N} : (i, j) \in \mathcal{A}\}$ denotes the neighbor set of node i. The total number of nodes in the network will be denoted by $|\mathcal{N}|$, the cardinality of $\mathcal{N}$. We restrict that each node is only able to communicate with its neighboring nodes, which are defined by the nodes within a circle of radius d_a. Besides, we assume that the graph $\mathcal{G}$ is connected, which means that there exists a multi-hop communication route connecting any two nodes in the network.

2.2 Dynamical Model and Measurement Model

Let $\mathbf{x}(t) = [x_1(t), \cdots, x_{n_x}(t)]^\top \in \mathbb{R}^{n_x}$ denote the system state at time t, where n_x is the dimension of the system state. For any time t_k and t_{k-1} with $t_k > t_{k-1}$, the evolution of the system state is given by

$$\mathbf{x}_k = \boldsymbol{f}(\mathbf{x}_{k-1}) + \mathbf{w}_k, \tag{1}$$

where $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ denote the system state at the absolute time instant t_k and t_{k-1}, $\boldsymbol{f}(\cdot)$ is a possibly nonlinear state transition function and $\mathbf{w}_k$ is white driving noise that is independent of the past and current states and whose probability density function is known.

Due to the influence of different sampling rates, initial timing offsets, and the processing-and-communication delays, the measurements of the WSN are assumed asynchronous. In other words, we consider an aWSN in this paper. An illustration of the measurement time series in an aWSN is shown in Fig. 1. Let t_i^l denote the exact time of the lth, $l = 1, 2, ..., N$, measurement sampling time of the ith node. At time t_i^l, the acquired measurement and its corresponding system state are denoted as $\mathbf{z}_i^l \in \mathbb{R}^{n_{z_i}}$ and $\mathbf{x}_i^l \in \mathbb{R}^{n_{x_i}}$, respectively. Then the relation between target state and the acquired measurements is given by

$$\mathbf{z}_i^l = \mathbf{h}_i(\mathbf{x}_i^l) + \mathbf{v}_i^l, \tag{2}$$

where $\mathbf{h}_i(\cdot)$ is a known and possibly nonlinear measurement function of the ith node and $\mathbf{v}_l^i$ is measurement noise that is independent of the past and current states and whose probability density function is known. In addition, the current measurement $\mathbf{z}_i^l$ is conditionally independent of past measurements $\mathbf{z}_i^{1:l-1}$ given the current state $\mathbf{x}_i^l$.

2.3 LF-FF Structure

The average consensus-based DBE method is implemented via two filters: *local filter (LF)* and *fusion filter (FF)* (FF is actually a consensus-based filter). Before running the two filters, a global update period (GUP) is defined at first. For LF, every local node seeks to compute a *predicted local posterior*, which essentially is a time-aligned local posterior

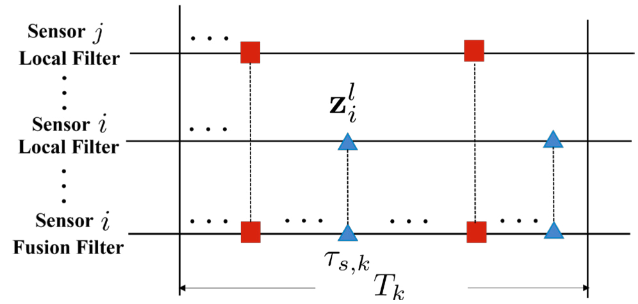


Fig. 1. Measurement time series in an aWSN.

at the terminal instant of the GUP. To achieve this, one of the two asynchronous batch estimation approaches, i.e., aBatch-1 and aBatch-2, is used. In particular, we assume the prior information and its local measurements are available. For aBatch-1, we predict the prior information to the terminal instant of the current GUP, and we use a smoothing strategy to align the local likelihood functions, and thus a LALF can be obtained. Based on the Bayes' rule, using the predicted prior information and LALF, the time-aligned local posterior, can be obtained. For aBatch-2, to obtain the time-aligned local posteriors, we use LF to compute the local posteriors and predict them to the terminal instant of the current GUP. Note that we allow multiple local posteriors to be processed at LF during one single GUP (see Fig. 1). Further, we assume that one per node has a FF, and it consistently fuses predicted local posteriors of multiple nodes in the aWSN. Note that, at FF, the DBE is achieved by an average consensus filter, which will be presented in Section (3.2). Finally, the global posterior gained at the end of each GUP will in turn be used as the prior information of the LF for the next GUP.

2.4 Notations for DBE

In this paper, we are interested in estimating the global posterior with a GUP T_k, and this is practical due to that sometimes an aWSN only needs to update the surveillance system periodically. To this end, we define the duration of the kth GUP as

$$T_k = t_k - t_{k-1}, \tag{3}$$

where t_{k-1} and t_k denote the initial and terminal time instant for the kth GUP, respectively. Note that T_k can be adaptively adjusted based on the specific estimation task. As shown in Fig. 1, during the kth GUP, we assume that there are $n_{i,k}$ measurements collected by the LF of node i. It means that, for an average consensus-based DBE, ideally, there are altogether N_k local posteriors can be acquired by the FF of node i. Then we have

$$N_k = \sum_{i=1}^{N} n_{i,k}. \tag{4}$$

Note that N_k is possible to be zero when T_k is small. For a batch processing, we stack the N_k asynchronous measurements in time order and define the asynchronous batch measurements during the kth GUP as $\tilde{\mathbf{z}}_k$.

$$\tilde{\mathbf{z}}_k \triangleq [(\mathbf{z}(\tau_{1,k}))^\top, (\mathbf{z}(\tau_{2,k}))^\top, ..., (\mathbf{z}(\tau_{N_k,k}))^\top]^\top. \tag{5}$$

Herein, $\mathbf{z}(\tau_{s,k})$ is the sth, $\{s, s \in 1, 2, ..., N_k\}$, measurement of all measurements in T_k, and the time sequence satisfies

$$\tau_{1,k} \leq ... \leq \tau_{s,k} ... \leq \tau_{N_k,k}. \tag{6}$$

As shown in Fig. 1, the measurement at time $\tau_{s,k}$ is actually the lth measurement collected by node i, hence we have $\mathbf{z}_i^l = \mathbf{z}(\tau_{s,k})$.

2.5 Problem Statement

In Fig. 1, we define all available measurements up to t_k as

$$\tilde{\mathbf{z}}_{1:k} \triangleq [\tilde{\mathbf{z}}_1^\top, \tilde{\mathbf{z}}_2^\top, ..., \tilde{\mathbf{z}}_k^\top]^\top. \quad (7)$$

From a Bayesian perspective, our goal is to calculate the posterior pdf $p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k})$ using the measurements $\tilde{\mathbf{z}}_{1:k}$. For the sake of providing a distributed implementation, we propose to obtain the global posterior using a LF-FF structure. Note that, since the term $\tilde{\mathbf{z}}_{1:k}$ is a collection of all asynchronous measurements of an aWSN, this estimation task is totally different from the synchronous cases (Li and Nehorai (2017)).

3. AVERAGE CONSENSUS-BASED DISTRIBUTED BATCH ESTIMATION

In this section, starting from the centralized batch estimation, the DBE fusion rule is proposed. We implement the DBE fusion rule via a LF-FF structure. At LF, each node uses one of the two asynchronous batch estimation approaches, i.e., aBatch-1 or aBatch-2 proposed in Section 3.3, to align and compute the asynchronous local posteriors. As a result, the predicted local posteriors are obtained. At FF, the average consensus filter is used to implement the DBE fusion rule, where the global posterior is obtained by an aggregation of the predicted local posteriors.

According to the Bayes' rule, the desired global posterior $p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k})$ can be obtained by sequentially applying prediction and update steps. Specifically, suppose that the global posterior $p(\mathbf{x}_{k-1}|\tilde{\mathbf{z}}_{1:k-1})$ at time $k-1$ is available, the prediction step from $k-1$ to k is accomplished using the Chapman−Kolmogorov equation:

$$p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\tilde{\mathbf{z}}_{1:k-1})d\mathbf{x}_{k-1}, \quad (8)$$

where $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is defined by the dynamical model (1), and it is assumed that the initial posterior $p(\mathbf{x}_0|\tilde{\mathbf{z}}_0) \equiv p(\mathbf{x}_0)$. Next, when the asynchronous measurement $\tilde{\mathbf{z}}_k$ is available at time k, then $p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k})$ can be computed via

$$\begin{aligned} p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k}) &= \frac{p(\tilde{\mathbf{z}}_k|\mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1})p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1})}{p(\tilde{\mathbf{z}}_k|\tilde{\mathbf{z}}_{1:k-1})} \\ &= \frac{1}{c}p(\tilde{\mathbf{z}}_k|\mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1})p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1}), \end{aligned} \quad (9)$$

where

$$c = \int p(\tilde{\mathbf{z}}_k|\mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1})p(\mathbf{x}_k|\mathbf{z}_{1:k-1})d\mathbf{x}_k$$

is a normalizing constant, which does not depend on the variable $\mathbf{x}_k$.

3.1 Centralized Batch Estimation Fusion Rule

For most existing distributed estimation methods, the measurement $\mathbf{v}_i^l$ is assumed to be independent of the local measurement of the other nodes, $\mathbf{v}_{i'}^l$ for $i \neq i'$ (Hlinka et al. (2012)). We use the same assumption in this paper, and a batch estimation fusion rule is given as follows.

Theorem 1. *For an aWSN, at time $\tau_{s,k}$, the measurements $\tilde{\mathbf{z}}_{1:k-1}$ and the local measurement $\mathbf{z}(\tau_{s,k})$ are available for each node. The centralized batch estimation fusion rule is given by*

$$\begin{aligned} &\log(p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k})) + (N_k-1)\log(p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1})) \\ &= \sum_{s=1}^{N_k} \log(p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})) + \bar{c}, \end{aligned} \quad (10)$$

where $\bar{c} = -\log(c) + \sum_{j=1}^{N_k} \log(p(\mathbf{z}(\tau_{s,k})|\tilde{\mathbf{z}}_{1:k-1}))$.

Proof. See Appendix A. □

As we see, (10) presents a centralized batch estimation fusion rule. In particular, $p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})$ on the right-hand side of (10) is a predicted local posterior pdf of $\mathbf{x}_k$ obtained by node i. The predicted means that it is obtained via a prediction of the local posterior $p(\mathbf{x}(\tau_{s,k})|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})$ using the equation (8). Further, $p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k})$ on the left-hand side of (10) is the global posterior of interest. There are two other terms in (10), namely the constant term and the prediction term. The constant term will disappear when we normalize $p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k})$ so that it integrates to 1. The prediction term $p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1})$ can be calculated according to (8).

3.2 FF: Average Consensus Filter and DBE Fusion Rule

As we see in (10), the global posterior is essentially obtained by a summation of the logarithmic result of the local posteriors, hence the centralized batch estimation fusion rule (10) can be implemented in a distributed manner through an average consensus algorithm.

$$\zeta_i^{\xi+1}(\mathbf{x}_k) = W_{ii}\zeta_i^{\xi}(\mathbf{x}_k) + \sum_{j\in\mathcal{N}_i} W_{ij}\zeta_j^{\xi}(\mathbf{x}_k) \quad (11)$$

where $\zeta_i^{\xi}(\mathbf{x}_k)$ is the consensus state variable at node i, i.e., $\log(p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1}))$ in (10). Herein, ξ denotes the ξth iteration of the average consensus algorithm during the kth GUP, and W_{ij} is the Metropolis weight (Xiao et al. (2005)) defined as

$$W_{ij} = \begin{cases} 1/\max\{|\mathcal{N}_i|, |\mathcal{N}_j|\} & \text{if} \quad (i,j) \in \mathcal{A} \\ 1 - \sum_{o\in\mathcal{N}_i} W_{io} & \text{if} \quad i = j \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

In the distributed fusion step, every node sends its current local posteriors to its neighbors and updates it with the local posteriors received from its neighbors. Note that (11) can also be reached from another perspective by minimizing the weighted average Kullback-Leibler distance between the fused posterior and the posteriors to be fused and then taking the logarithm. Besides, as pointed in Li and Nehorai (2017), under certain conditions, for $\forall k$,

$$\lim_{\xi\to\infty} \log(\zeta_i^{\xi}(\mathbf{x}_k)) = \frac{1}{N_k}\sum_{i=1}^{N_k} \log(\zeta_i^0(\mathbf{x}_k)) + \underline{c}, \quad (13)$$

where the constant term $\underline{c}$ is added simply for the purpose of normalization.

Combining the results in (10) and (13), we have the following theorem for the DBE fusion rule.

Theorem 2. *For an aWSN, at time $\tau_{s,k}$, the measurements $\tilde{\mathbf{z}}_{1:k-1}$ and the local measurement $\mathbf{z}(\tau_{s,k})$ are available for each node. The DBE fusion rule is given by*

$$p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k}) = \frac{\left(\lim_{\xi\to\infty}(\zeta_i^{\xi}(\mathbf{x}_k))\right)^{N_k}}{(p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1}))^{N_k-1}} \cdot \breve{c}, \tag{14}$$

where $\breve{c} = e^{\bar{c}-N_k\underline{c}}$ is a constant, and K is the number of iterations.

Remark 1. *Assume that an aWSN is properly synchronized, i.e., the measurements are all aligned at the terminal instant of a GUP. The DBE rule (14) will degrade to the distributed fusion rule given in Li and Nehorai (2017), which designs for a synchronous WSN.*

Remark 2. *At FF, if the GUP T_k is chosen to be the interval of two consecutive local posteriors, we will obtain $N_k = 1$. Then the global posterior in (14) is equivalent to the local posterior for each node. Thereby the DBE fusion rule will reduce to the CSE proposed in Katragadda et al. (2017).*

Remark 3. *For the DBE fusion rule given in (14), it is noted that the exact average consensus can only be achieved when $\xi \to \infty$. For the simulations in Section 5, only a finite number of iterations are used, while we find that almost perfect consensus can be achieved with a relatively large ξ. As a reference, analysis about the influence of the consensus iteration on an average consensus filter can be found in Hlinka et al. (2014).*

3.3 LF: Compute Predicted Local Posterior

In the last two subsections, we have discussed how to provide a distributed implementation of (10) using an average consensus filter. In other words, we have discussed how to obtain the DBE fusion rule given in (14). However, the question how to compute the consensus state variable $\zeta_i^{\xi}(\mathbf{x}_k)$ given in (14), i.e., the predicted local posterior $\log(p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1}))$, has not been answered.

Generally, the predicted local posterior can be computed using two approaches: aBatch-1 and aBatch-2.

- *Approach 1: aBatch-1*

For aBatch-1, we write the predicted local posterior as

$$p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1}) = \frac{p(\mathbf{x}_k|, \tilde{\mathbf{z}}_{1:k-1})p(\mathbf{z}(\tau_{s,k})|\mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1})}{p(\mathbf{z}(\tau_{s,k})|\tilde{\mathbf{z}}_{1:k-1})}. \tag{15}$$

Using aBatch-1 to compute (15) consists of two steps.

Step 1: Prediction. At time $k-1$, each node computes the predicted pdf $p(\mathbf{x}_k|, \tilde{\mathbf{z}}_{1:k-1})$ using (8).

Step 2: Smoothing. Compute the LALF according to

$$\begin{aligned} &p(\mathbf{z}(\tau_{s,k})|\mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1}) \\ &= \int p(\mathbf{z}(\tau_{s,k})|\mathbf{x}(\tau_{s,k}))p(\mathbf{x}(\tau_{s,k})|\mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1})d\mathbf{x}(\tau_{s,k}), \end{aligned} \tag{16}$$

where $p(\mathbf{z}(\tau_{s,k})|\mathbf{x}(\tau_{s,k}))$ is a local likelihood function which can be obtained according to the measurement model (2). Meantime, using the Bayes' rule, the term $p(\mathbf{x}(\tau_{s,k})|\mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1})$ can be expressed as

$$p(\mathbf{x}(\tau_{s,k})|\mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1}) = \frac{p(\mathbf{x}_k|\mathbf{x}(\tau_{s,k}))p(\mathbf{x}(\tau_{s,k})|\tilde{\mathbf{z}}_{1:k-1})}{p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1})}, \tag{17}$$

where $p(\mathbf{x}_k|\mathbf{x}(\tau_{s,k}))$ can be obtained based on the dynamical model (1), and $p(\mathbf{x}(\tau_{s,k})|\tilde{\mathbf{z}}_{1:k-1})$ can be computed using (8).

- *Approach 2: aBatch-2*

For aBatch-2, we denote the predicted local posterior as:

$$\begin{aligned} &p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1}) \\ &= \int p(\mathbf{x}(\tau_{s,k})|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})p(\mathbf{x}_k|\mathbf{x}(\tau_{s,k}))d\mathbf{x}(\tau_{s,k}). \end{aligned} \tag{18}$$

To obtain $p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})$, we have two steps as well.

Step 1: Filtering. Compute the current local posterior $p(\mathbf{x}(\tau_{s,k})|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})$ using (8) and (9).

Step 2: Prediction. Use time transition probability density function $p(\mathbf{x}_k|\mathbf{x}(\tau_{s,k}))$ to complete the prediction of local posterior $p(\mathbf{x}(\tau_{s,k})|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})$. As a result, $p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})$ at time k can be obtained via (18).

Remark 4. *The ideas of the two asynchronous batch estimation approaches, i.e., aBatch-1 and aBatch-2, can be explained as follows. For the first approach, we predict the system state to the terminal instant of a GUP, i.e., time t_k. Then we use a smoothing probability density function, i.e. $p(\mathbf{x}(\tau_{s,k})|\mathbf{x}_k, \tilde{\mathbf{z}}_{1:k-1})$, to align the local likelihood function $p(\mathbf{z}(\tau_{s,k})|\mathbf{x}(\tau_{s,k}))$ to time t_k. For the second approach, we compute the local posterior $p(\mathbf{x}(\tau_{s,k})|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})$ at first, and make predictions to time t_k. The predicted local posteriors obtained by these two approaches are essentially equivalent.*

4. SIMULATION RESULTS

In this section, a target tracking example in an aWSN is considered. We demonstrate that the proposed average consensus-based DBE method has high target-tracking accuracy and is robust to strong asynchronism. Besides, we implement the DBE fusion via a particle filter (Arulampalam et al. (2002)), and the optimality can be approximately achieved with a sufficiently large number of particles and consensus iterations.

In the simulation, we employ $M = 25$ nodes to monitor a 40m × 40m area, and each node with a $d_a \times d_a$ sensing and communicating ability, where $d_a = 18$. Fig. 2(a) gives an intuitive exhibition of the WSN and a target tracking problem is considered. We assume that the asynchronism is coming from two major factors: different sampling rates and communication delays. To model the first factor, we assume that the local update period for each node is randomly generated from the set $T_{\text{loc}} = [1, 2, 3]$. For the second factor, we consider that the processing-and-communication delay is known and modeled by an exponential distribution with mean $\mu = 0.2$. The GUP T_k is chosen to be $T_k = 2$.

We follow the target motion model and measurement model used in Hlinka et al. (2012). Let $\mathbf{x}_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^\top$ denote the target state time t_k, where (x_k, y_k) and $(\dot{x}_k, \dot{y}_k)$ represent the 2D position and velocity, respectively. We assume that the target follows a nearly constant velocity motion described by

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k, \tag{19}$$

where $\mathbf{F}_k$ is the state transition matrix, defined as

$$\mathbf{F}_k = \mathbf{I}_2 \otimes \begin{bmatrix} 1 & T_k \\ 0 & 1 \end{bmatrix}, \tag{20}$$

and $\mathbf{w}_k$ is a zero-mean Gaussian process noise during the time interval $[t_k, t_{k-1}]$ with covariance matrix

$$\mathbf{Q}_{k,k-1} = \tilde{q}\mathbf{I}_2 \otimes \begin{bmatrix} T_k^3/3 & T_k^2/2 \\ T_k^2/2 & T_k \end{bmatrix}, \tag{21}$$

where $\mathbf{I}_2$ denotes the 2×2 identity matrix, $\otimes$ is the Kronecker product, and $\tilde{q}$ is the power spectral density of process noise. The initial target state is chosen as $[4\text{m}, 0.25\text{m/s}, 4\text{m}, 0.25\text{m/s}]^\top$, $\tilde{q} = 0.001^2\ \text{km}^2/\text{s}^3$, and the total observation period is 60s.

The local node measures the range and bearing of the target. At time t_l^i, the measurement model of the ith node is given by

$$\mathbf{z}_l^i = \begin{pmatrix} r_l^i \\ \theta_l^i \end{pmatrix} = \begin{pmatrix} \sqrt{\left(x_l^i - x_s^i\right)^2 + \left(y_l^i - y_s^i\right)^2} \\ \tan^{-1}\left(y_l^i - y_s^i, x_l^i - x_s^i\right) \end{pmatrix} + \mathbf{v}_l^i \tag{22}$$

where (x_s^i, y_s^i) indicates the 2D position of the ith node and (x_l^i, y_l^i) is the 2D position of target state at time t_l^i. The measurement noise $\mathbf{v}_l^i$ is zero-mean Gaussian with covariance matrix $\mathbf{R}_l^i = [\text{diag}(\sigma_r^i; \sigma_\theta^i)]^2$. The particle number size (PNS) for the posterior pdf is set as $N_p = 1000$. Besides, for the secondary sampling strategy used for computing LALF, the PNS is $M = 50$.

Let $(x_{k,m}, y_{k,m})$ and $(\hat{x}_{k,m}, \hat{y}_{k,m})$ denote the true and estimated 2D positions at time t_k respectively for the mth Monte Carlo run. To evaluate the tracking performance of the proposed algorithm, the root-mean-square error (RMSE) of position at time t_k is defined as

$$\text{RMSE}_k = \sqrt{\frac{1}{N_M}\sum_{m=1}^{N_M}(\hat{x}_{k,m} - x_{k,m})^2 + (\hat{y}_{k,m} - y_{k,m})^2}, \tag{23}$$

where N_M is the number of Monte Carlo runs and we set $N_M = 500$.

As shown in Fig. 2(b), in order to demonstrate the effectiveness of the proposed methods, the tracking performance of different algorithms are compared. They are the proposed average consensus-based DBE (ACB-DBE) algorithm (To distinguish the algorithms that use aBatch-1 and aBatch-2 to compute predicted local posteriors, they are named as ACB-DBE-1 and ACB-DBE-2, respectively) with the aCDTT [1], and CSE (Katragadda et al. (2017)). Clearly, the proposed ACB-DBE-1 and ACB-DBE-2 are quite effective to deal with asynchronous data, where they have better tracking performance than the aCDTT. Notably, the aCDTT algorithm does not fuse local information of the nodes in the WSN, where there is no uncertainty reduction on the state. CSE is recognized as a benchmark of the two proposed algorithms. Due to that the RMSE curves of the two ACB-DBE algorithms are close to the curve of CSE, which further demonstrate the effectiveness of the two proposed algorithms.

In addition, to exhibit that ACB-DBE-1 and ACB-DBE-2 are robust to the influence of different extent of asynchronous degree, a new definition of asynchronism is given as follows. According to Fig. 1, we define the asynchronism of the ith FF during the kth update time as:

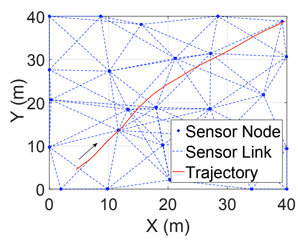
$$\gamma = \frac{1}{N_k}\sum_{s=1}^{N_k}|t_k - \tau_{i,s}|, \tag{24}$$

where $|\cdot|$ refers to the operation for taking the absolute value. Note that the definition of asynchronism (24) is different from the definition in Katragadda et al. (2017), where the asynchronism is the upper bound of the time instant captured by the LF, i.e., $\gamma = \max_{s=1,\ldots,N_k}\{\tau_{i,s}\}$. The new definition is provided because, for both of our methods, local posteriors in the kth GUP have to complete the prediction or smoothing tasks at LF, and every local posterior makes influence to the estimation performance. To this end, we take all asynchronous local posteriors into consideration to evaluate the robustness of our algorithms to different asynchronism. In particular, according (24), three terms, N_k, t_k and $\tau_{i,k}$, can change asynchronism. Here, by carefully designing t_k and the number of nodes N (which affects N_k) besides the local update period, 6 simulations with different asynchronisms, $\gamma = 2, 5, 10, 15, 25, 40$, can be given. For instance, $\gamma = 2$ is ensured by $T_u = 2$, $N = 2$, and $T_{\text{loc}} = 1$ for all nodes with no time delays. The influence of the asynchronsim to the tracking performance is shown in Fig. 2(d). Note that we use the (time-)averaged RMSE (ARMSE) to evaluate the tracking performance for one particular scenario with a specific asynchronism value, which is defined as the square root of the average of RMSE over all simulated time instants. Moreover, the tracking performance of CSE is given as a benchmark to compare the two proposed methods. It can be seen that the tracking accuracy of all methods decreases with the increase of asynchronism, while there is no divergence phenomenon appeared even with a high level of asynchronism ($\gamma = 40$). Besides, the ARMSE curves of the two proposed methods are always close to the curve of the CSE, which further proves that the ACB-DBE-1 and ACB-DBE-1 are robust to asynchronism. Finally, we use the same simulation condition as that in proving the effectiveness of our methods in Fig. 2(b). As shown in Fig. 2(c) anf Fig. 2(d), the optimality of the two DBE algorithms can be achieved with a sufficiently large number of particle samples (> 5000) and over 8 consensus iterations.

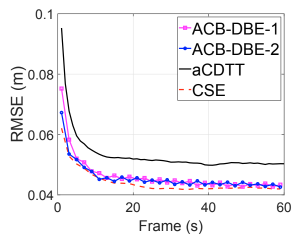
5. CONCLUSION

This paper addresses the distributed estimation problem in an aWSN and proposes a DBE fusion rule to update the global posterior in a GUP. The DBE fusion rule, implemented with an LF-FF structure, allows for an approximately optimal global posterior with a sufficiently large number of consensus iterations. At LF, local posteriors are aligned and computed using either aBatch-1 or aBatch-2, which involve prediction-smoothing and filtering-prediction steps, respectively. At FF, an average consensus filter is employed to fuse the aligned local posteriors and obtain the expected global posterior. Numerical

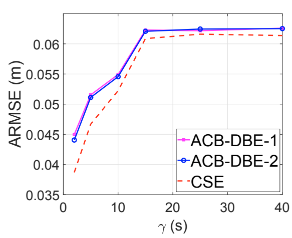
[1] We made some modifications to the aCDTT, where the event-based approach is not simulated. But its major feature, i.e., keeping the maximum certain local states, is retained.



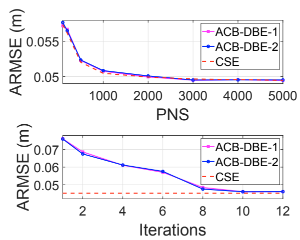
(a) A typical topology example of the target tracking in an aWSN.



(b) The tracking performance of ACB-DBE-1, ACB-DBE-2, aCDTT and CSE.



(c) The tracking performance of ACB-DBE-1, ACB-DBE-2 and CSE influenced by different γ.



(d) The tracking performance of the ACB-DBE-1 and ACB-DBE-2 algorithms versus the PNS and Iterations.

Fig. 2. Simulation Results: WSN settings and tracking performance.

results demonstrate the high estimation accuracy, robustness to strong asynchronism, and convergence properties of the proposed DBE method.

Appendix A. PROOF OF THEOREM 1

According to (9) and write the global likelihood function as a product of local likelihood functions, we know that

$$p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k}) = \frac{1}{c}p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1}) \times \prod_{s=1}^{N_k} \frac{p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})p(\mathbf{z}(\tau_{s,k})|\tilde{\mathbf{z}}_{1:k-1})}{p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1})}, \tag{A.1}$$

Rewrite (A.1), easy to have

$$c \cdot p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k}) \prod_{s=1}^{N_k-1} p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1}) = \prod_{s=1}^{N_k} p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})p(\mathbf{z}(\tau_{s,k})|\tilde{\mathbf{z}}_{1:k-1}). \tag{A.2}$$

Finally, taking the logorithmic form of (A.2), the equation simplifies to

$$\log(p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k})) + (N_k - 1)\log(p(\mathbf{x}_k|\tilde{\mathbf{z}}_{1:k-1})) = \sum_{j=1}^{N_k} \log(p(\mathbf{x}_k|\mathbf{z}(\tau_{s,k}), \tilde{\mathbf{z}}_{1:k-1})) + \bar{c}.$$

where $\bar{c} = -\log(c) + \prod_{s=1}^{N_k} \log(p(\mathbf{z}(\tau_{s,k})|\tilde{\mathbf{z}}_{1:k-1}))$. We prove that Theorem 1 holds.

REFERENCES

Arulampalam, M.S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 174–188.

Aysal, T.C., Yildiz, M.E., Sarwate, A.D., and Scaglione, A. (2009). Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal processing*, 57(7), 2748–2761.

Corke, P., Wark, T., Jurdak, R., Hu, W., Valencia, P., and Moore, D. (2010). Environmental wireless sensor networks. *Proceedings of the IEEE*, 98(11), 1903–1917.

Dimakis, A.G., Kar, S., Moura, J., Rabbat, M.G., and Scaglione, A. (2010). Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11), 1847–1864.

Giannini, S., Petitti, A., Di Paola, D., and Rizzo, A. (2013). Asynchronous consensus-based distributed target tracking. In *52nd IEEE Conference on Decision and Control*, 2006–2011. IEEE.

Hlinka, O., Hlawatsch, F., and Djuric, P.M. (2012). Distributed particle filtering in agent networks: A survey, classification, and comparison. *IEEE Signal Processing Magazine*, 30(1), 61–81.

Hlinka, O., Slučiak, O., Hlawatsch, F., and Rupp, M. (2014). Distributed data fusion using iterative covariance intersection. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1861–1865. IEEE.

Katragadda, S., Regazzoni, C.S., and Cavallaro, A. (2017). Average consensus-based asynchronous tracking. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4401–4405. IEEE.

Kim, J.S., Lee, J., Serpedin, E., and Qaraqe, K. (2009). A robust estimation scheme for clock phase offsets in wireless sensor networks in the presence of non-Gaussian random delays. *Signal Processing*, 89(6), 1155–1161.

Li, J. and Nehorai, A. (2017). Distributed particle filtering via optimal fusion of Gaussian mixtures. *IEEE Transactions on Signal and Information Processing over Networks*, 4(2), 280–292.

Li, M., Liang, G.Q., Luo, H.B., Qian, H.H., and Lam, T.L. (2020). Robot-to-robot relative pose estimation based on semidefinite relaxation optimization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Olfati-Saber, R., Fax, J.A., and Murray, R.M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233.

Olshevsky, A. and Tsitsiklis, J.N. (2009). Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1), 33–55.

Shima, T. and Rasmussen, S. (2009). *UAV cooperative decision and control: challenges and practical approaches.* SIAM.

Vemula, M., Miguez, J., and Artes-Rodriguez, A. (2007). A sequential monte carlo method for target tracking in an asynchronous wireless sensor network. In *2007 4th Workshop on Positioning, Navigation and Communication*, 49–54. IEEE.

Xiao, L., Boyd, S., and Lall, S. (2005). A scheme for robust distributed sensor fusion based on average consensus. In *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, 63–70. IEEE.