Adaptive Consensus-Based Distributed Target Tracking With Dynamic Cluster in Sensor Networks

Hao Zhang, Xue Zhou, Zhuping Wang, Huaicheng Yan, and Jian Sun

***Abstract*—This paper is concerned with the target tracking problem over a filtering network with dynamic cluster and data fusion. A novel distributed consensus-based adaptive Kalman estimation is developed to track a linear moving target. Both optimal filtering gain and average disagreement of the estimates are considered in the filter design. In order to estimate the states of the target more precisely, an optimal Kalman gain is obtained by minimizing the mean-squared estimation error. An adaptive consensus factor is employed to adjust the optimal gain as well as to acquire a better filtering performance. In the filter's information exchange, dynamic cluster selection and two-stage hierarchical fusion structure are employed to get more accurate estimation. At the first stage, every sensor collects information from its neighbors and runs the Kalman estimation algorithm to obtain a local estimate of system states. At the second stage, each local sensor sends its estimate to the cluster head to get a fused estimation. Finally, an illustrative example is presented to validate the effectiveness of the proposed scheme.**

***Index Terms*—Adaptive filtering, data fusion, distributed Kalman filtering, dynamic cluster selection, sensor networks.**

I. Introduction

Wireless sensor network (WSN) is significant for many applications, such as supervisory systems, rescue activities, target tracking, intelligent transportation [1]–[4], etc. In such scenarios, a large number of sensors are deployed in the sensing field to acquire the target's states. Since the distributed WSN has the advantage of robustness for filter failure, lower communication and computation costs, it has gained particular attention in recent years [5]–[7]. The purpose of a distributed WSN is to provide users with access to required information from data measured by spatially distributed sensors, which addresses the problem on how to efficiently incorporate distributed estimations [8], [9]. A typical WSN consists of a number of battery-powered sensors, which use wireless transmission to communicate. Although centralized filtering network is relatively easy to design, however, the cost of transmitting data is very large and once there exists sensor failure, the whole system will not work. Besides, in centralized filtering networks, each sensor needs to send local data to a fusion center, which needs more time to communicate, so it may not meet the real-time requirements in dynamic target tracking. Therefore, some decentralized works have attracted more and more interests [10]–[12]. A fully decentralized Kalman filter has been proposed in [13], in which each sensor needs to link to every other sensor to compute its local estimate, thus a complex communication topology is required. Based on this, a revised distributed Kalman filtering algorithm is proposed for sensor networks in [14] and [15], in which the amount of the links in the communication topology has been greatly decreased. Different from the centralized estimation [16], [17], the distributed sensors only need their neighbor's data and the coordination among filters, which can enhance the global estimation of the filtering network. The difficulty in designing distributed filters lies in the coupling between the filters is usually complex and optimal filtering performance is not easy to acquire.

Recently, some intelligent algorithms such as neural networks, fuzzy frameworks, adaptive methods on distributed filtering have been used to acquire an optimal filter [18]–[22]. An optimal adaptive filter has been studied in [21], which introduced an adaptive factor for the model-predicted state vector to control its outlying disturbance influences and get the optimal adaptive factor from estimated covariance matrix of predicted residuals. In [22], a heterogeneous sensor network with different processing abilities sensors has been considered and the optimal form of the sensor gain matrix has been obtained. Inspired by consensus theory in networks of multiagents [23]–[25], a consensus-based strategy is proposed in distributed filtering. The consensus-based strategy of the distributed Kalman filter can be broadly classified into two categories: one is to add a consensus term to the Kalman filter at update step [26] and the other is to deal with the consensus of the *a priori* estimate at the Kalman filter prediction step [27], [28]. Some initial efforts have been made on the distributed consensus-based filtering over sensor networks such as [2], [4], [8], and [29]–[31]. In [29], a consensus-based distributed mixture Kalman filter has been developed, combining

Manuscript received August 22, 2017; revised December 1, 2017 and January 30, 2018; accepted February 4, 2018. Date of publication April 24, 2018; date of current version March 5, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant u1764261 and Grant 61773289, in part by the Projects of Shanghai International Cooperation under Grant 18510711100, and in part by the Fundamental Research Funds for the Central Universities. This paper was recommended by Associate Editor Y. Shi. *(Corresponding author: Zhuping Wang.)*

H. Zhang, X. Zhou, and Z. Wang are with the Department of Control Science and Engineering, Tongji University, Shanghai 200092, China (e-mail: zhang_hao@tongji.edu.cn; 1631560@tongji.edu.cn; elewzp@tongji.edu.cn).

H. Yan is with the College of Mechatronics and Control Engineering, Hubei Normal University, Huangshi 435002, China, and also with the Key Laboratory of Advanced Control and Optimization for Chemical Process of Ministry of Education, East China University of Science and Technology, Shanghai 200237, China (e-mail: hcyan@ecust.edu.cn).

J. Sun is with the Department of Transportation Engineering, Tongji University, Shanghai 200092, China (e-mail: sunjian@126.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2805717

2168-2267 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

the particle filter with the traditional Kalman filter, it can estimate the state of conditional dynamic linear systems. In [30], a distributed weighted average consensus algorithm has been proposed to examine several dynamical aspects of average consensus in mobile networks. Demetriou [31] established an abstract framework that considers the distributed filtering of spatially varying processes and incorporated the adaptation of the consensus gains in the disagreement terms of all local filters. In the aforementioned distributed filtering investigations, optimization approach and adaptive method are employed. It is worth pointing out that in the filter design, most existing results use constant consensus gain to design a distributed filter. This may lead to large average disagreement of estimates, which adversely affect the filtering performance. To overcome this defect, an adaptive consensus-based filter has been developed in this paper, which can improve the consensus among all sensors and enhance filtering performance.

It is noted that local estimates obtained by measurements from its neighbors are not optimal in the sense that not all the measurements are used in the distributed WSN. The hierarchical tracking structure [32] is presented to improve each local estimate and reach a common state estimate. On the other hand, the key issue in the WSN-based tracking is how to balance the tracking accuracy with the limited resources, such as energy, bandwidth, computational capabilities, etc. Reasonable clustering methods are important safeguards to reduce network energy consumption and improve the service life of sensor networks [33]. The clustering structure is highly efficient to manage terminal nodes, balance communication loads and reduce collisions in networks. In [34], the problems on collecting local data and generating the data report are studied by proposing the dynamic convoy tree-based collaboration method, which are formulated as a multiple objective optimization problem. Besides, establishing dynamic collaborative clustering of the Kalman filtering, controlling radius of cluster to active nodes, selecting sensors based on Kalman filtering information matrix and other methods have been proposed to frame clusters [35]–[37]. Although many advanced results on cluster method have been reported in [38]–[40], there are some challenging issues to be addressed. For example, how to establish clusters based on the distance between the moving target and coordinate-fixed sensors as well as how to select the cluster head based on energy consumption and how to design a fusion method based on the cluster structure is also a challenge. All these observations motivate us to design a dynamic cluster mechanism and execute a hierarchical two-stage fusion algorithm based on cluster structure for a better robustness and accuracy in distributed filtering.

In view of all the above considerations, a novel distributed adaptive filtering method based on dynamic cluster is presented in this paper for target tracking problem. Specially, a hierarchical two-stage fusion approach is designed by considering both energy-efficiency and robustness. At the first stage, all member nodes in a cluster collect the estimates from its neighbors and itself to generate its own estimate. At the second stage, the cluster head collects the estimates from all the member nodes to form a fused estimate of the moving target. In addition, the cluster head also calculates the distance between the sensors and the target and wakes up those who will enter the cluster at the next time step. Once the cluster head is about to leave the cluster, a new cluster head will be selected according to the residual energy and the current cluster head will transmit all its data to the next cluster head. To analyze the energy status, a model of energy consumption in data transmission will be established. Based on this model, the residual energy of every sensor in the cluster can be computed and a cluster head can also be selected.

The main contributions of this paper can be summarized as follows.

1) A novel adaptive consensus-based distributed filter is provided. The adaptive factor is designed to flexibly adjust the consensus gain according to the disagreement of the estimates among sensors. With this adaptive law, the average disagreement of estimates in filtering network can be decreased and the filtering precision also be improved. All in all, it deals with the relationship between adjustable consensus and better filtering performance.
2) A hierarchical two-stage estimation scheme is proposed and a cluster establishment mechanism is designed. At the first stage, the local sensors use their neighbors' information to form their own estimates. At the second stage, local sensors send their estimates to the fusion node to acquire a more accurate result. This process is executed in a dynamic cluster, which helps to improve estimation performance in an energy-efficient way.

Notations: $\text{diag}\{.\}_m$ stands for a block-diagonal matrix with m elements in the diagonal, $\otimes$ stands for the Kronecker product, $\mathbb{E}\{.\}$ means the expectation, I_m is an m-dimensional unit matrix, $\lambda_{\max}(P)$ represents the largest eigenvalue of matrix P, and $\lambda_i(P)$ represents the ith eigenvalue of matrix P.

II. Problem Statement

The sensor network can be modeled as a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ consisting of a set of sensors $\mathcal{V} = \{1, 2, \ldots, N\}$ and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The existence of edge (i, j) means that the ith sensor can communicate with the jth sensor. The set of neighbors of sensor i is denoted by $\mathcal{N}_i = \{j; (i, j) \in \mathcal{V}\}$. Let $d_i = |\mathcal{N}_i|$ be the number of neighboring sensors of the ith sensor. The adjacent matrix $\mathcal{A}(k) = [a_{ij}(k)]$ with nonnegative adjacency elements is defined by $a_{ij}(k) = 1$ if $(i, j) \in \mathcal{E}$, otherwise $a_{ij}(k) = 0$. The degree matrix is described by $\mathcal{D}(k) = \text{diag}\{\mathcal{D}_1(k), \mathcal{D}_2(k), \ldots, \mathcal{D}_N(k)\}$, where the diagonal element is represented as $\mathcal{D}(k) = \sum_{j\in\mathcal{N}_i} a_{ij}(k)$. The Laplacian matrix of the directed graph $\mathcal{G}$ is defined as $\mathcal{L}(k) = \mathcal{D}(k) - \mathcal{A}(k)$ and $\mathcal{L}(k) = [l_{ij}(k)]$.

The target is described as

$$x(k + 1) = Ax(k) + \omega(k) \tag{1}$$

where $x \in \mathbb{R}^m$ is the state vector, the initial state $x(0)$ is zero-mean Gaussian with covariance $\Pi_0 \geq 0$. $\omega(k) \in \mathbb{R}^m$ is the process noise, which is also assumed to be zero-mean white Gaussian with $\mathbb{E}\{\omega_k \omega_j^T\} = \delta_{kj} Q \geq 0$, where $\delta_{kk} = 1$ and $\delta_{kj} = 0$ $(k \neq j)$. $x(0)$ is independent of $\omega(k)$ for all $k \geq 0$.

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

Suppose that the state is observed by N sensors distributed in a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$, then the measurement of the ith sensor is given by

$$y_i(k) = H_i x(k) + v_i(k) \tag{2}$$

where $v_i(k) \in \mathbb{R}^m$ is the zero-mean Gaussian measurement noise with covariance $R_i > 0$. $v_i(k)$ is independent of $x(0)$ and $\omega(k), \forall k, i$, and is independent of $v_j(s)$ when $i \neq j$ or $k \neq s$. Besides, matrices A and H_i have compatible dimensions.

To track the moving target, a consensus-based distributed filter of the ith node is designed as follows:

$$\begin{aligned}\hat{x}_i(k+1) = {}& A\hat{x}_i(k) + K_i(k)\big[y_i(k) - H_i\hat{x}_i(k)\big]\\ & - \alpha_i(k)\varepsilon A \sum_{j\in\mathcal{N}_i} a_{ij}(k)\big[\hat{x}_i(k) - \hat{x}_j(k)\big]\end{aligned} \tag{3}$$

where $\hat{x}_i(k) \in \mathbb{R}^m$ is the estimate of the state $x(k)$ based on the ith sensing node, $K_i(k)$ is the gain matrix, and ε is the consensus gain with the range of $(0, 1/\Delta)$, $\Delta = \max(d_i)$. $\alpha_i(k)$ is the adaptive factor and its form will be introduced in the next section.

Remark 1: Notice that the consensus term contains the adaptive factor, with this adaptive factor, the average disagreement of the estimates among all sensors can be adjusted automatically, so as to the optimal estimation can be obtained under the preset requirements of consensus. Therefore, once the average disagreement of the estimates achieves the requirement, the adaptive law need not work. In other words, if $\alpha_i(k) \equiv 1$, then [41] becomes a special case of this adaptive estimator.

Next, the problem of determination of filter gain matrix $K_i(k)$ is addressed, such that the designed filter (3) is a minimum mean-squared error estimator.

Define the estimated error of the ith sensor as $e_i(k) = \hat{x}_i(k) - x(k)$, where $\hat{x}_i(k)$ is an unbiased estimate of $x(k)$, thus $\mathbb{E}\{e_i(k)\} = 0, \forall i \in \mathcal{V}$. Therefore, the cross error covariance $P_{ij}(k) = \mathbb{E}\{e_i(k)e_j^T(k)\}$. Let $F_i(k) = A - K_i(k)H_i$, then the iterative equation of P_{ij} can be obtained as

$$\begin{aligned}P_{ij}(k+1) = {}& F_i(k)P_{ij}(k)F_j^T(k) + Q + K_i(k)R_{ij}K_j^T(k)\\ & + \alpha_i(k)\alpha_j(k)\varepsilon^2 A \sum_{s\in\mathcal{N}_i}\sum_{r\in\mathcal{N}_j} a_{is}(k)a_{jr}(k)\\ & \times \big[P_{ij}(k) - P_{ir}(k) - P_{sj}(k) + P_{sr}(k)\big]A^T\\ & + \alpha_j(k) \times F_i(k)\sum_{r\in\mathcal{N}_j} a_{jr}\big[P_{ij}(k) - P_{ir}(k)\big]A^T\\ & + \alpha_i(k)\varepsilon \times A\sum_{s\in\mathcal{N}_i} a_{is}(k)\big[P_{ij}(k) - P_{sj}(k)\big]F_j^T(k)\end{aligned} \tag{4}$$

where $R_{ij} = \mathbb{E}\{v_i(k)v_j^T(k)\}$ is the cross-covariance of $v_i(k)$ and $v_j(k)$. Similarly, the error covariance of P_i can be obtained as

$$\begin{aligned}P_i(k+1) = {}& F_i(k)P_i(k)F_i^T(k) + Q + K_i(k)R_iK_i^T(k)\\ & + (\alpha_i(k)\varepsilon)^2 A \sum_{\substack{r,s\in\mathcal{N}_i\\ r\neq s}} a_{is}(k)a_{ir}(k)\\ & \times [P_i(k) - P_{ir}(k) - P_{si}(k) + P_{rs}(k)]A^T\\ & + (\alpha_i(k)\varepsilon)^2 \times A\sum_{r\in\mathcal{N}_i} a_{ir}(k)\\ & \times [P_i(k) + P_r(k) - P_{ir}(k) - P_{ri}(k)]A^T\\ & - \alpha_i(k)\varepsilon F_i(k)\sum_{r\in\mathcal{N}_i} a_{ir}(k)[P_i(k) - P_{ir}(k)]A^T\\ & - \alpha_i(k)\varepsilon A\sum_{r\in\mathcal{N}_i} a_{ir}(k)[P_i(k) - P_{ri}(k)]F_i^T(k).\end{aligned} \tag{5}$$

It follows from (5) that:

$$\begin{aligned}P_i(k+1) = {}& A\Bigg\{P_i(k) - 2\alpha_i(k)\varepsilon\sum_{r\in\mathcal{N}_i} a_{ir}(k)P_i(k) + \alpha_i(k)^2\\ & \times \varepsilon^2\sum_{r\in\mathcal{N}_i} a_{ir}^2(k)P_i(k) + (\alpha_i(k)\varepsilon)^2 \sum_{\substack{r,s\in\mathcal{N}_i\\ r\neq s}} a_{ir}(k)\\ & \times a_{is}(k)P_i(k)\Bigg\}A^T\\ & + A\Bigg\{\sum_{r\in\mathcal{N}_i} a_{ir}(k)\Bigg(\alpha_i(k)\varepsilon - (\alpha_i(k)\varepsilon)^2 a_{ir}(k)\\ & - (\alpha_i(k)\varepsilon)^2\sum_{\substack{s\in\mathcal{N}_i\\ r\neq s}} a_{is}(k)\Bigg)\\ & \times [P_{ri}(k) + P_{ir}(k)]\Bigg\}A^T + (\alpha_i(k)\varepsilon)^2\\ & \times A\sum_{\substack{r,s\in\mathcal{N}_i\\ r\neq s}} a_{ir}(k)a_{is}(k)P_{rs}(k)A^T + (\alpha_i(k)\varepsilon)^2\\ & \times A\sum_{r\in\mathcal{N}_i} a_{ir}^2(k)P_r(k)A^T\\ & - A\Bigg\{P_i(k) + \alpha_i(k) \times \varepsilon\sum_{r\in\mathcal{N}_i} a_{ir}(k)\\ & \times [P_{ri}(k) - P_i(k)]\Bigg\}H_iM_i^{-1}(k)\\ & \times H_i^T\Bigg\{P_i(k) + \alpha_i(k)\varepsilon\sum_{r\in\mathcal{N}_i} a_{ir}(k)\\ & \times [P_{ir}(k) - P_i(k)]\Bigg\}A^T\\ & + \big[K_i(k) - K_i^*(k)\big]M_i(k) \times \big[K_i(k) - K_i^*(k)\big]^T\end{aligned} \tag{6}$$

where $M_i(k) = H_iP_i(k)H_i^T + R_i$. From (6), one can obtain the optimal gain

$$\begin{aligned}K_i^*(k) = {}& A\Bigg\{P_i(k) + \alpha_i(k)\varepsilon\sum_{r\in\mathcal{N}_i} a_{ir}(k)[P_{ri}(k) - P_i(k)]\Bigg\}\\ & \times H_iM_i^{-1}(k).\end{aligned} \tag{7}$$

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

Since the mean-squared error is defined as MSE $= (1/N)\sum_{i=1}^{N} e_i(k)e_i^T(k)$, so minimum $P_i(k)$ will lead to minimum mean-squared error estimator. Moreover, in the optimal gain, $P_{ri}(k)$ is used to eliminate the disagreement of error to make each sensor estimate the states of the target more accurate in the case of coordination estimation.

Remark 2: If $\varepsilon = 0$, then $K_i(k)^* = AP_i(k)H_iM_i^{-1}(k)$ is a suboptimal Kalman gain and $P_i(k+1) = F_i(k)P_i(k)F_i^T(k) + Q + AP_i(k)H_iM_i^{-1}(k)R_iM_i^{-T}(k)H_i^TP_i^T(k)A^T$ is a suboptimal estimation error covariance, which has been obtained in [42]. This suboptimal filter has relatively less amount of calculation, but its filtering performance is worse than the optimal one.

III. Convergence Analysis

In this section, the stability of the proposed estimator with the designed optimal gain is analyzed. Due to the coupling of the estimation errors among neighboring sensors, it is difficult to prove that the estimation error covariance converges to a unique positive definite matrix as in the centralized Kalman filter, so only an upper and a lower bound of the estimation error covariance is derived. Before moving on, the following assumptions and definition are introduced.

Assumption 1: There exist nonzero real constants $\underline{f}$, $\bar{f}$, $\underline{h}_i$, $\bar{h}_i$, $\underline{q}$, $\bar{q}$, $\underline{r}_i$, and $\bar{r}_i$, for all $k \geq 0$, $i \in N$, such that the following bounds on the matrices are satisfied:

$$\underline{f}^2 I_m \leq AA^T \leq \bar{f}^2 I_m \tag{8}$$

$$\underline{h}_i^2 I_m \leq H_iH_i^T \leq \bar{h}_i^2 I_m \tag{9}$$

$$\underline{q} I_m \leq Q \leq \bar{q} I_m \tag{10}$$

$$\underline{r}_i I_m \leq R_i \leq \bar{r}_i I_m. \tag{11}$$

Assumption 2: The initial error covariance $P_i(0)$ is positive semidefinite.

Definition 1 [43]: If there exist constants $0 < a \leq 1$, $\kappa_1 \geq 0$, and $\kappa_2 > 0$ such that a discrete sequence ξ_k satisfying

$$\xi_k \leq \kappa_1 + \kappa_2(1-a)^k \tag{12}$$

then ξ_k is said to be exponentially bounded with exponent a.

Lemma 1 [44]: Let X, Y, U, and V be given matrices with appropriate dimensions. If X, Y, and $Y^{-1} + VX^{-1}U$ are invertible, then the following equation holds:

$$(X + UYV)^{-1} = X^{-1} - X^{-1}U(Y^{-1} + VX^{-1}U)^{-1}VX^{-1}.$$

Now, we are ready to present the main results of this paper.

Theorem 1: Suppose that Assumptions 1 and 2 hold. If there exist constants φ, ϱ, $\bar{K}$, and κ, making the error covariance of each sensor bounded, that is $P_i(k+1) \leq \varphi I_m$ and $P_i(k+1) \geq \varrho I_m$, then the designed filter is convergent, thus the sensors can track the target. The aforementioned convergent boundary can be represented as $\varrho = \underline{q} + \kappa$, $\varphi = \bar{P}_k(\bar{f}^2 + \bar{h}_i^2\bar{K}^2)(1+\xi)$, where $\bar{P}_k = (\bar{f} + \bar{h}_i\bar{K})^{2k}\bar{P}_0 + ([\underline{q} + \bar{r}_i\bar{K}^2]/[1 - (\bar{f} + \bar{h}_i\bar{K})^2])$, $\bar{P}_0 = \max\{\lambda_{\max}(P_i(0), i \in N)\}$, $\xi = ([\bar{r}_i\bar{K}^2 + \bar{q} + \kappa]/[\bar{f}^2(\underline{q} + \kappa)])$.

Proof: According to (5), $P_i(k+1)$ can be rewritten as

$$P_i(k+1) = F_i(k)P_i(k)F_i^T(k) + Q + K_i(k)R_iK_i^T(k) + \Delta P_i(k) \tag{13}$$

where

$$\begin{aligned}\Delta P_i(k) = {} & (\alpha_i(k)\varepsilon)^2 A \sum_{r,s\in\mathcal{N}_i, r\neq s} a_{is}(k)a_{ir}(k) \\ & \times [P_i(k) - P_{ir}(k) - P_{si}(k) + P_{rs}(k)]A^T \\ & + (\alpha_i(k)\varepsilon)^2 A \sum_{r\in\mathcal{N}_i} a_{ir}(k) \\ & \times [P_i(k) + P_r(k) - P_{ir}(k) - P_{ri}(k)]A^T - \alpha_i(k) \\ & \times \varepsilon F_i(k) \sum_{r\in\mathcal{N}_i} a_{ir}(k)[P_i(k) - P_{ir}(k)]A^T - \alpha_i(k)\varepsilon \\ & \times A \sum_{r\in\mathcal{N}_i} a_{ir}(k)[P_i(k) - P_{ri}(k)]F_i^T(k).\end{aligned}$$

■

It has been proved in [41] that there exists a sufficiently small constant ε such that $\|\Delta P_i(k)\| < \kappa$, thus $P_i(k+1)$ can be written approximatively as

$$P_i(k+1) = F_i(k)P_i(k)F_i^T(k) + K_i(k)R_iK_i^T(k) + Q + \kappa I_m. \tag{14}$$

Suppose that $K_i(k)K_i^T(k) \leq \bar{K}^2 I_m$. For simplicity, denote $K_i(k)R_iK_i^T(k) + Q + \kappa I_m = \Phi(k)$. According to Assumption 1, one has

$$\Phi(k) \leq \left(\bar{r}_i\bar{K}^2 + \bar{q} + \kappa\right)I_m. \tag{15}$$

Then, the following inequality can be derived by Lemma 1:

$$\begin{aligned}(P_i(k+1))^{-1} &= \Phi(k)^{-1} - \Phi(k)^{-1}F_i(k) \\ &\quad \times \left(F_i^T(k)\Phi(k)^{-1} \times F_i(k) + P_i(k)^{-1}\right)^{-1} \\ &\quad \times F_i^T(k)\Phi(k)^{-1} \\ &\leq \Phi(k)^{-1} \\ &= (Q + \kappa I_m)^{-1} - (Q + \kappa I_m)^{-1}K_i(k) \\ &\quad \times \left(K_i^T(k) \times (Q + \kappa I_m)^{-1}K_i(k) + R_i^{-1}\right)^{-1} \\ &\quad \times K_i^T(k) \times (Q + \kappa I_m)^{-1} \\ &\leq (Q + \kappa I_m)^{-1}.\end{aligned} \tag{16}$$

It follows from (16) that:

$$\Phi(k) \geq Q + \kappa I_m \geq \left(\underline{q} + \kappa\right)I_m. \tag{17}$$

Substituting (17) into (16), one obtains

$$P_i(k+1) \geq \left(\underline{q} + \kappa\right)I_m \tag{18}$$

where $\underline{q} + \kappa$ is denoted as ϱ. Thus the lower bound of $P_i(k+1)$ has been obtained.

Now, in order to derive the upper bound of $P_i(k+1)$, the inverse of $P_i(k+1)$ is expressed by another form

$$\begin{aligned}(P_i(k+1))^{-1} &= \left(F_i(k)P_i(k)F_i^T(k) + F_i(k)F_i^{-1}(k)\Phi(k) \right. \\ &\qquad \left. \times F_i^{-T}(k)F_i^T(k)\right)^{-1} \\ &= F_i^{-T}(k)\left(P_i(k) + F_i^{-1}(k)\Phi(k)F_i^{-T}(k)\right)^{-1} \\ &\qquad \times F_i^{-1}(k).\end{aligned} \tag{19}$$

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

By using Lemma 1, one has

$$\begin{aligned}F_i^{-1}(k) &= (A - K_i(k)H_i)^{-1}\\ &= A^{-1} - A^{-1}K_i(k)\\ &\quad\times \left(I_m A^{-1}K_i(k) + (-H_i)^{-1}\right)^{-1} I_m A^{-1}\\ &\leq A^{-1} \end{aligned}\tag{20}$$

$$F_i^{-1}(k)F_i^{-T}(k) \leq (AA^T)^{-1} \leq \frac{1}{\bar{f}^2} I_m. \tag{21}$$

According to Assumption 1, one can obtain that

$$\begin{aligned}F_i(k)F_i^T(k) &\leq AA^T + K_i(k)H_iH_i^TK_i^T(k)\\ &\leq \left(\bar{f}^2 + \bar{h}_i^2\bar{K}^2\right)I_m. \end{aligned}\tag{22}$$

Thus,

$$F_i^{-1}(k)\Phi(k)F_i^{-T}(k) \leq \left(\frac{\bar{r}_i\bar{K}^2 + \bar{q} + \kappa}{\bar{f}^2}\right)I_m. \tag{23}$$

Assume that there exists a positive scalar ξ, such that $F_i^{-1}(k)\Phi(k)F_i^{-T}(k) \leq \xi P_i(k+1)$, from (18), one has

$$F_i^{-1}(k)\Phi(k)F_i^{-T}(k) \leq \xi\left(\underline{q} + \kappa\right)I_m. \tag{24}$$

According to (23), it is obtained that

$$\begin{aligned}\xi\left(\underline{q} + \kappa\right) &= \frac{\bar{r}_i\bar{K}^2 + \bar{q} + \kappa}{\bar{f}^2}\\ \xi &= \frac{\bar{r}_i\bar{K}^2 + \bar{q} + \kappa}{\bar{f}^2\left(\underline{q} + \kappa\right)}. \end{aligned}\tag{25}$$

Then, combining this with (19), we have

$$\begin{aligned}(P_i(k+1))^{-1} &\geq F_i^{-T}(k)(P_i(k) + \xi P_i(k))^{-1}F_i^{-1}(k)\\ &= (1+\xi)^{-1}F_i^{-T}(k)P_i^{-1}(k)F_i^{-1}(k). \end{aligned}\tag{26}$$

Thus,

$$P_i(k+1) \leq F_i(k)P_i(k)F_i^T(k)(1+\xi). \tag{27}$$

Now, the mathematical induction is used to get the upper bound of $P_i(k+1)$. It can be divided into two steps.

1) *Initial Step:* It is noted that $P_i(0)$ is positive semidefinite, thus for time step $l = 0$, $P_i(0) \leq \lambda_{\max}(P_i(0))I_m$. One can find $\bar{P}_0 = \max\{\lambda_{\max}(P_i(0)), i \in N\}$, making $P_i(0) \leq \bar{P}_0 I_m$ satisfied.

   For $l = 1$, based on (14), it is obtained that

$$P_i(1) \leq \left\{\left(\bar{f}^2 + \bar{h}_i^2\bar{K}^2\right)\bar{P}_0 + \bar{r}_i\bar{K}^2 + \bar{q}\right\}I_m. \tag{28}$$

2) *Inductive Step:* Suppose that at the step $l = m$, the following inequality holds:

$$P_i(m) \leq \left\{\left(\bar{f}^2 + \bar{h}_i^2\bar{K}^2\right)\bar{P}_{m-1} + \bar{r}\bar{K}^2 + \bar{q}\right\}I_m \tag{29}$$

   where $\bar{P}_{m-1} = (\bar{f}^2 + \bar{h}_i^2\bar{K}^2)^{m-1}\bar{P}_0 + \sum_{k_s=0}^{m-2}(\bar{f}^2 + \bar{h}_i^2\bar{K}^2)^{k_s}(\bar{r}_i\bar{K}^2 + \bar{q})$.

It can be concluded that at time step $l = m+1$

$$P_i(m+1) \leq \left\{\left(\bar{f}^2 + \bar{h}_i^2\bar{K}^2\right)\bar{P}_m + \bar{r}_i\bar{K}^2 + \bar{q}\right\}I_m \tag{30}$$

where $\bar{P}_m = (\bar{f}^2 + \bar{h}_i^2\bar{K}^2)^m\bar{P}_0 + \sum_{k_s=0}^{m-1}(\bar{f}^2 + \bar{h}_i^2\bar{K}^2)^{k_s}(\bar{r}_i\bar{K}^2 + \bar{q})$, which yields that $P_i(k) \leq \bar{P}_k I_m$ with

$$\begin{aligned}\bar{P}_k &= \left(\bar{f}^2 + \bar{h}_i^2\bar{K}^2\right)^k\bar{P}_0 + \left(\underline{q} + \bar{r}_i\bar{K}^2\right)\sum_{k_s=0}^{k-1}\left(\bar{f}^2 + \bar{h}_i^2\bar{K}^2\right)^{k_s}\\ &\leq \left(\bar{f} + \bar{h}_i\bar{K}\right)^{2k}\bar{P}_0 + \left(\underline{q} + \bar{r}_i\bar{K}^2\right)\sum_{k_s=0}^{\infty}\left(\bar{f} + \bar{h}_i\bar{K}\right)^{2k_s}. \end{aligned}\tag{31}$$

When $0 < (\bar{f} + \bar{h}_i\bar{K})^2 \leq 1$

$$\bar{P}_k \leq \left(\bar{f} + \bar{h}_i\bar{K}\right)^{2k}\bar{P}_0 + \frac{\underline{q} + \bar{r}_i\bar{K}^2}{1 - \left(\bar{f} + \bar{h}_i\bar{K}\right)^2}. \tag{32}$$

It is noted that the upper bound $\bar{P}_k$ is related to $(\bar{f} + \bar{h}_i\bar{K})^2$, when $\bar{K} < -(\bar{f}/\bar{h}_i)$, bigger $\bar{K}$ leads to faster convergence speed; otherwise, bigger $\bar{K}$ will lead to slower convergence speed.

By using Definition 1, one can see that $\bar{P}_k$ is exponentially bounded with exponent$(1 - (\bar{f} + \bar{h}_i\bar{K})^2)$. Substituting $P_i(k) \leq \bar{P}_k I_m$ into (26) yields

$$P_i(k+1) \leq \bar{P}_k\left(\bar{f}^2 + \bar{h}_i^2\bar{K}^2\right)(1+\xi)I_m \tag{33}$$

where $\bar{P}_k(\bar{f}^2 + \bar{h}_i^2\bar{K}^2)(1+\xi)$ is denoted as φ. Consequently, when k approaches to infinite, $P_i(k)$ is bounded.

In order to satisfy $0 < (\bar{f} + \bar{h}_i\bar{K})^2 \leq 1$, the adaptive law is introduced. When ε is sufficiently small, $K_i(k)$ can be written as

$$\begin{aligned}K_i(k) &= A(P_i(k) + \alpha_i(k)\theta I_m)H_i\\ &\quad\times\left(R_i^{-1} - R_i^{-1}H_i \times \left(P_i^{-1}(k) + H_i^TR_i^{-1}H_i\right)H_i^TR_i^{-1}\right)\end{aligned}$$

where θ is a constant. Consequently,

$$\begin{aligned}K_i(k)H_i^TA^T &= A(P_i(k) + \alpha_i(k)\theta I_m)H_iR_i^{-1}H_i^TA^T\\ &\quad - A(P_i(k) + \alpha_i(k)\theta I_m)H_iR_i^{-1}H_i\\ &\quad\times\left(P_i^{-1}(k) + H_i^TR_i^{-1}H_i\right)\\ &\quad\times H_i^TR_i^{-1}H_i^TA^T\\ &\leq A(P_i(k) + \alpha_i(k)\theta I_m)H_iR_i^{-1}H_i^TA^T.\end{aligned}$$

Afterwards,

$$\begin{aligned}K_i(k)H_i^TA^TAH_iK_i^T(k) &\leq A(P_i(k) + |\alpha_i(k)\theta|I_m)H_iR_i^{-1}H_i^TA^TA\\ &\quad\times H_iR_i^{-T}H_i^T(P_i(k) + |\alpha_i(k)\theta|I_m)^TA^T.\end{aligned}$$

Considering Assumption 1 and the lower bound of $P_i(k)$, one can obtain the upper bound of $K_i(k)$ as

$$\|K_i(k)\| \leq |\bar{K}| \tag{34}$$

where $|\bar{K}| = |([\underline{f}^2\underline{h}_i^2(\underline{q} + \kappa + |\alpha_i(k)\theta|)]/[\bar{r}_i\bar{f}\bar{h}_i])|$. Since $0 < (\bar{f} + \bar{h}_i\bar{K})^2 \leq 1$ means $0 < \bar{K}^2 \leq [(1-\bar{f})^2/\bar{h}_i^2]$, so $|\underline{q} + \kappa| + |\alpha_i(k)\theta| \leq |([(1-\bar{f})\bar{r}_i\bar{f}]/[\underline{f}^2\underline{h}_i^2])|$, then one has

$$|\alpha_i(k)\theta| \leq \left|\frac{(1-\bar{f})\bar{r}_i\bar{f}}{\underline{f}^2\underline{h}_i^2}\right| - \left|\underline{q} + \kappa\right|.$$

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

Therefore, the initial value of $\alpha_i(0)$ must satisfy the following restrictions:

$$|\alpha_i(0)| \le \left|\frac{(1-\bar{f})\bar{r}_i\bar{f}}{\underline{f}^2\underline{h}_i^2\theta}\right| - \left|\frac{\underline{q}+\kappa}{\theta}\right|. \tag{35}$$

The adaptive law can be designed as

$$\alpha_i(k+1) = \begin{cases} \alpha_i(k), & \text{if} \quad \text{ADoE} \le \beta \\ \alpha_i(k) - a\alpha_i(k)\Big(l - \frac{1}{l-\text{ADoE}}\Big)^b, & \text{if} \quad \text{ADoE} > \beta \end{cases} \tag{36}$$

where ADoE is the average disagreement of the estimates among all sensors. It can be characterized as, $\text{ADoE} = (1/N)\sum_{i=1}^{N}\|\hat{x}_i - (1/N)\sum_{i=1}^{N}\hat{x}_i\|^2$. a is used to decide the rangeability, b coordinates the decrease proportion, l represents the past l steps. β is the needed threshold value, it is decided by demand. In the past l steps, computing the ADoE, if it is beyond the requirement, then $\alpha_i(k)$ should be changed to adjust the consensus gain ε, otherwise, it will not change, that is to say, once the consensus meets the requirement, adaptive strategies stop working, only when the consensus destroyed by external disturbances or other factors, the adaptive strategy will comeback to work. Moreover, from (7) and (33), it is noted that $\bar{K}$ and $P_i(k+1)$ vary with $\alpha_i(k)$, when $\alpha_i(k)$ decreases, $\bar{K}$ is decreased, thus, the upper bound of $P_i(k+1)$ is becoming smaller. Since a smaller upper bound of $P_i(k+1)$ indicates a better estimation performance, therefore the estimation performance has been improved.

Remark 3: It is noted that $\bar{K}$ belongs to $[-|([1-\bar{f}]/\bar{h}_i)|, -(\bar{f}/\bar{h}_i)) \bigcup (-(\bar{f}/\bar{h}_i), |([1-\bar{f}]/\bar{h}_i)|]$. In order to clarify the convergence speed with the parameter selection in the adaptive law, we take one side for example, that is $\bar{K}$ belongs to $(-(\bar{f}/\bar{h}_i), |([1-\bar{f}]/\bar{h}_i)|]$. From (36), one can see that the rangeability of $\alpha_i(k)$ is related to a, so bigger a is selected, smaller $\alpha_i(k)$ is derived, when smaller $\alpha_i(k)$ is, smaller $\bar{K}$ can be derived, which leads to faster convergence speed. The other side of $\bar{K}$ can be analyzed by the same way.

IV. Data Fusion and Cluster Establishment

A. Data Fusion

A two-stage hierarchical fusion structure is employed to estimate the target's states. In the first stage, an optimal distributed filter is designed to generate local estimates of the mobile target. In the second stage, a Kalman fusion approach will be presented to fuse local estimates from the cluster members to produce the fused estimates of the mobile target.

Lemma 2 [45]: Let $\hat{x}_i$, $i = 1, 2, \ldots, l$ be the unbiased estimators of n-dimensional stochastic vector x. Let estimation errors be $\tilde{x}_i = x - \hat{x}_i$. Assume that $\tilde{x}_i$ and $\tilde{x}_j$, $i \ne j$, are correlated. Cross-covariance are P_{ij} and P_{ii}. Then the optimal information fusion is

$$\hat{x}_0 = \bar{A}_1\hat{x}_1 + \bar{A}_2\hat{x}_2 + \cdots + \bar{A}_l\hat{x}_l \tag{37}$$

where the optimal matrix weights $\bar{A}_i, i = 1, 2, \ldots, l$, are computed by

$$\bar{A} = \Sigma^{-1}e\Big(e^T\Sigma^{-1}e\Big)^{-1} \tag{38}$$

where $\bar{A} = [\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_l]^T$ and $e = [I_n, \ldots, I_n]^T$ are both $nl \times n$ matrices, and $\Sigma = (P_{ij})_{nl\times nl}$, $i, j = 1, 2, \ldots, l$ are both $nl \times nl$ positive definite matrix. The error covariance of the optimal information fusion estimator with matrix weights is computed by $P_0 = (e^T\Sigma^{-1}e)^{-1}$.

Data fusion is based on cluster, the cluster head of each cluster fuses the data. When local estimates calculated by the estimators in (3) are available at fusion node h, h collects them to generate a fused estimate according to the fusion rule in Lemma 2.

It is supposed that a cluster has l_s member nodes. Define $\hat{x}(k) = [\hat{x}_1^T(k), \hat{x}_2^T(k), \ldots, \hat{x}_{l_s}^T(k)]^T$, $e(k) = [e_1^T(k), e_2^T(k), \ldots, e_{l_s}^T(k)]^T$, and $v(k) = [v_1^T(k), v_2^T(k), \ldots, v_{l_s}^T(k)]^T$. By stacking all the estimation errors in vector form, one has

$$\begin{aligned} e(k+1) &= \Big((I_{l_s}\otimes A) - \varepsilon\text{diag}\{I_m\alpha_i(k)\}(\mathcal{L}(k)\otimes A) \\ &\quad - \text{diag}\{K_i(k)H_i\}\Big)e(k) + \text{diag}\{K_i(k)\}v(k) \\ &\quad - 1_{l_s}\otimes\omega(k) \\ &= \Upsilon(k)e(k) + \Omega(k) \end{aligned} \tag{39}$$

where $\Upsilon(k) = (I_{l_s}\otimes A) - \varepsilon\text{diag}\{I_m\alpha_i(k)\}(\mathcal{L}(k)\otimes A) - \text{diag}\{K_i(k)H_i\}$ and $\Omega(k) = \text{diag}\{K_i(k)\}v(k) - 1_{l_s}\otimes\omega(k)$.

Let $P(k) = \mathbb{E}\{e(k)e^T(k)\}$, then

$$P(k+1) = \Upsilon(k)P(k)\Upsilon^T(k) + E\big\{\Omega(k)\Omega^T(k)\big\}. \tag{40}$$

According to the Kalman fusion rule proposed in Lemma 2, one can obtain that

$$\begin{aligned} \hat{x}_0(k) &= P^{-T}(k)e^T\Big(e^TP^{-1}(k)e\Big)^{-T}\hat{x}(k) \\ P_0(k) &= \Big(e^TP^{-1}(k)e\Big)^{-1} \end{aligned} \tag{41}$$

where $e = [I_m, \ldots, I_m]^T$.

Remark 4: Most existing results on distributed consensus-based filters (see [8], [41], [46]) are derived only based on the first stage, but since each local estimator can only estimate the state based on the information obtained from itself and its neighbors, the resulting estimate is not very accurate. Different from the literatures mentioned above, the two-stage hierarchical fusion estimation proposed in this paper can be used in some situations where higher estimation accuracy is required. Moreover, traditional fusion estimation only estimate at the fusion center, each local filter just sends their data to the fusion center, and without local estimation. If the fusion center paralyzed, it may lead to a breakdown of the whole network. However, the two-stage estimation in this paper is robust, even if the cluster head paralyzed, the state information can still be obtained from each member sensor in the current cluster before a new cluster head generated. Afterwards, the distance information of the paralyzed head node can be calculated from its neighbor. When the cluster head does not satisfy the distance limit, it will be kicked out of the cluster, then a new cluster head will produce to continue the fusion strategy.

B. Cluster Establishment

Now, the working scheme on the distributed state estimation will be introduced. The framework of WSN is shown in Fig. 1.

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

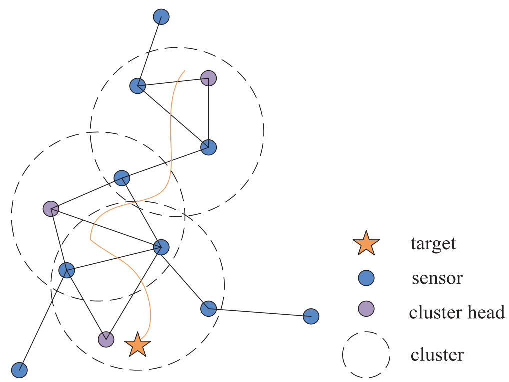


Fig. 1. Structure of distributed estimation systems.

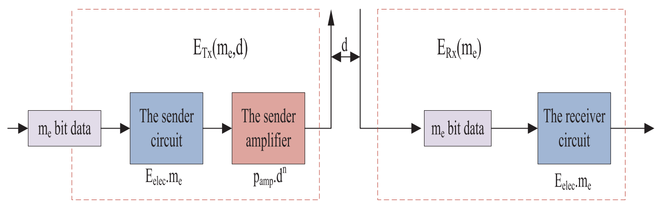


Fig. 2. Energy consumption model of wireless sensor node.

Denote $\vartheta(k)$ as the position component of $x(k)$. Suppose that the initial position of the target $\vartheta(0)$ is known. The mobile target is detected by the odometry method. The ultrasonic sensor nodes in the monitoring area with Cartesian coordinates has fixed-coordinate, and denoted by $\vartheta_{si}(k)$. The target is considered as the center of a circle, and the sensor within the radius of the circle will be activated, that is to say, when the distance between the target and the sensor i is within a constraint $\bar{d}$, sensor i starts to work, otherwise sensor i stays asleep. At the initial state, the distance between sensors and the target can be computed out, then the sensors which meet the conditions are activated, thus the first cluster has been formed. As the target moves, some nodes in the cluster become far away from the target and are kicked out. In each cluster, a node will be selected as the cluster head. The cluster head only fuses the data to obtain more complete and accurate information about the target, and compute the distance between every sensor and the target in the next step. It will broadcast the entire network which sensors are awakened. As the target moves, a new cluster head will be selected when the cluster head is kicked out.

The wireless sensor node energy consumption model is given in Fig. 2. When a sensor node transmits a m_e-bit data packet to the receiver node with a distance d, the sender node will consume energy $E_{Tx}(m_e, d)$

$$E_{Tx}(m_e, d) = \begin{cases} E_{\text{elec}}.m_e + p_{\text{fs}}.m_e.d^2, & d < d_0 \\ E_{\text{elec}}.m_e + p_{\text{mp}}.m_e.d^4, & d \geq d_0. \end{cases} \tag{42}$$

The consumption energy of the receiver is

$$E_{\text{Rx}}(m_e) = E_{\text{elec}}.m_e \tag{43}$$

where d_0 is the critical distance, d is the distance between the sender and the receiver, E_{elec} represents the circuit energy consumption, and p_{fs} and p_{mp} represent for the amplifier energy consumption factor under different channel models. When the transmission distance is short, the free-space model with p_{fs} is used to calculate energy consumption, otherwise the multipath attenuation model with p_{mp} is used. Besides, the energy of data fusion is E_{DA}. Next, one analyzes the energy consumption of cluster head and member nodes. Each member node has three states.

1) *Sleep State:* Sensors do not estimate the target state and do not communicate with other nodes. For simplicity, it is supposed that this state does not consume energy.
2) *Receiving State:* Sensors collect the information of the target and estimate the target's states. Energy consumption is related to the data size.
3) *Transmission State:* Member sensors transmit the estimated value to other sensor nodes. Energy consumption is also related to the data size.

Assume that the sensor consumes energy E_e when it estimates one-*bit* data. Then according to Fig. 2, it is concluded that at every time step, each member sensor consumes energy $e^i_{\text{member}} = \varsigma_i(E_{\text{elec}}.m_e + p_{\text{amp}}.m_e.d^n + E_e.m_e)$, where $\varsigma_i = \{1, 0\}$, $n = \{2, 4\}$, and $p_{\text{amp}} = \{p_{\text{fs}}, p_{\text{mp}}\}$.

Correspondingly, the cluster head has four states.

1) *Sleep State:* The cluster head does not calculate the distance between the target and the member nodes to wake up the member nodes, nor does it perform data fusion. This state does not consume energy.
2) *Receiving State:* The cluster head receives the target's estimated value from member sensors.
3) *Transmission State:* The cluster head broadcasts message to sensors that are about to enter the target's range.
4) *Processing State:* The cluster head computes the distance between target and member nodes, and fuses data to get a more accurate estimation. Suppose that the cluster head consumes energy E_c when it computes 1-bit distance information.

Therefore, at every time step, the cluster head consumes energy $e_{\text{head}} = \varsigma(2.E_{\text{elec}}.m_e + p_{\text{amp}}.m_e.d^n + E_c.m_e + E_{\text{DA}})$.

Now, we are ready to present how to select the cluster. It is supposed that the initial energy of every sensor is E^i_0, in the initial cluster, the cluster head is chosen randomly. Then, after every time step, the sensors calculate the residual energy $E^i_k = E^i_{k-1} - e^i_{\text{member}}$ or $E^i_k = E^i_{k-1} - e_{\text{head}}$. If the last cluster head is kicked out of the cluster, then the node which has the highest energy will be chosen as the new cluster head in the current cluster (if there are sensors having same energy, the cluster head will be decided randomly in these energy-same nodes).

In order to more clearly explain the cluster member, cluster head selection mechanism and the two-stage estimation process, Algorithm 1 is introduced as follows.

It is noted that each cluster node transmits the local estimate to the cluster head through the multihop route, and the cluster head also uses a multihop broadcast route to wake up the nodes that will work in the next cluster.

Remark 5: The proposed filtering approach in this paper is suitable for situations where high accuracy and robustness localization is required. Take the localization of disaster relief as an example. Sensors are deployed densely

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Sensor and Cluster Head Selection Algorithm Based on Energy Residual and Distance Range

1: Initialize $\hat{x}_i(0)$, compute $d_i(0) = \|\vartheta(0) - \vartheta_{si}\|$, if $d_i(0) \leq \bar{d}$, sensor i is chosen to form the first cluster and senor h with the highest energy is selected as the cluster head.
2: **for** k = 0: M.
3: Sensor i estimates the target's state $\hat{x}_i(k+1)$ according to filter (3).
4: Every sensor in the cluster sends its $\hat{x}_i(k+1)$ to the cluster head.
5: The cluster head uses Kalman fusion (41) to get a more accurate state $\hat{x}(k+1)$ of the target.
6: The cluster head extracts the target's position $\vartheta(k+1)$ and compute $d_i(k+1) = \|\vartheta(k+1) - \vartheta_{si}\|$. If $d_i(k+1) > \bar{d}$, the cluster head broadcasts to sensor i that it will be kicked out, else if $d_i(k+1) \leq \bar{d}$ and sensor i is asleep, the cluster head wakes it up.
7: Judge whether $d_h(k+1) \leq \bar{d}$, if not, go to step 8, else back to step 3.
8: Calculate the residual energy E_k^i of each sensor, select the highest energy node as the cluster head at the next time step, go to step 3.
9: **end**

in disaster areas. Each sensor is equipped with a small embedded microprocessor, which is responsible for collecting information, estimating states and calculating energy residual based on the energy model given in Fig. 2. Information exchange among sensors is through WiFi. A WiFi network could adopt IEEE 802.11n, which is handled on 2.4-GHz bands and a maximum network data from 54 and 600 MB. When the rescue robots move in the disaster area searching for survivors, with robots as the center, the sensors within a radius of 50 m is activated. Then the active sensors use Algorithm 1 to estimate the position of robots and fusion nodes use 4G network to send data to a computer terminal. The position of robots can be derived from the terminal, thus once the survivor is found, rescue operations can be started quickly.

V. Simulation Results

In this section, simulations of a maneuvering target tracking system [41] are presented to demonstrate the effectiveness of the proposed estimator design method, where the parameters of state space model in (1) and (2) are

$$A = \begin{bmatrix} 1.01 & 0 \\ 0 & 1.01 \end{bmatrix}, \quad Q = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$
$$H_i = \begin{bmatrix} 2\delta_i & 0 \\ 0 & 2\delta_i \end{bmatrix}, \quad R_i = \begin{bmatrix} 2\nu_i & 0 \\ 0 & 2\nu_i \end{bmatrix}$$

where $\delta_i, \nu_i \in (0, 1]$ for all i. The initial state is given by $x(0) = [1.253; 2.877]$, where state $x^{(1)}$ stands for the velocity component and $x^{(2)}$ stands for the position component. Besides, $\bar{f} = 1.2, \underline{f} = 0.8, \bar{h}_i = 2, \underline{h}_i = 0.1, \bar{q} = 2, \underline{q} = 1, \bar{r}_i = 2, \underline{r}_i = 0.1, \theta = 15$, and $\kappa = 0.5$. In filter (3), ε is chosen as 0.01, and $\alpha_i(0) = 1$ to guarantee (35). It is worth pointing out that our main results are valid even if the nominal target plant is unstable. Considering a network consists of 1000 sensors, the topology of sensor networks is characterized in Fig. 3. The sensors are stochastically deployed in a square area of 3.5×3.5 km. The longest distance between two adjacent sensors is 0.2 km and the links among all sensors are stochastic. Since each cluster has the same working mechanism, so only 100 sensors in the first cluster are taken as an example to illustrate the estimation performance.

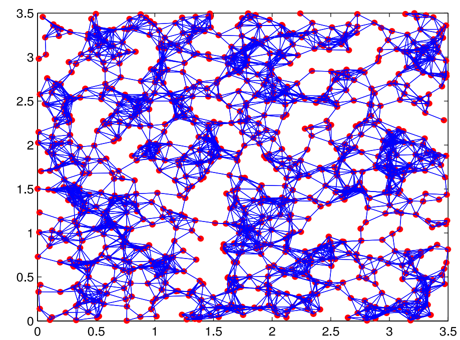
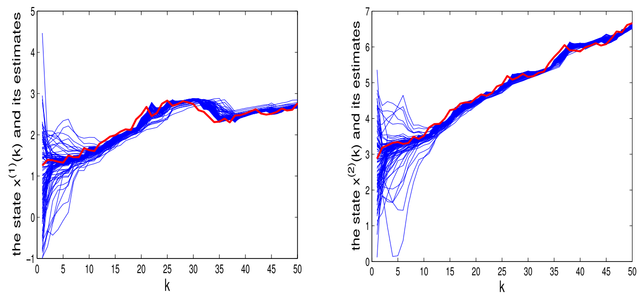


Fig. 3. Network topology with 1000 sensors.



(a) (b)

Fig. 4. States (a) $x^{(1)}(k)$ and (b) $x^{(2)}(k)$, and their estimates.

Fig. 4(a) displays the true value of target's position (the red curve) and the estimated trajectory of sensors i (the blue curve), from which, one can observe that all 100 sensors can well track the target position and maintain a satisfactory disagreement between them. Fig. 4(b) demonstrates the velocity of the moving target, where the red curve is the target's true value, and the blue curve is the estimated value of the 100 sensors. It shows that sensors can track the speed of the target.

If the adaptive factor in (3) is fixed as 1, then a normal distributed filter in [41] is obtained. Fig. 5 depicts the trace of error covariance of the filter proposed in [41] (the red one) and the filter designed in (3) (the blue one). The adaptive law has parameters $a = 0.6, b = 1, \beta = 0.1$, and $l = 2$. Each sensor checks its past two steps at time step k_l with $\mathrm{mod}(k_l, 2) = 0$, and decide whether $\alpha_i(k)$ will change according to rule (36). For clarity, only sensor 6's error covariance is plotted. To quantify the performance difference, the error covariance is characterized by the following equation:

$$\Delta_p = \mathrm{trace}(P_e) - \mathrm{tarce}(P_k) \tag{44}$$

where P_e represents the error covariance of the filter proposed in [41] and P_k represents the error covariance of the filter designed in this paper. In order to more intuitively describe this difference, the frequency distribution histograms of Δ_p is plotted in Fig. 6. From Figs. 5 and 6, it is observed

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

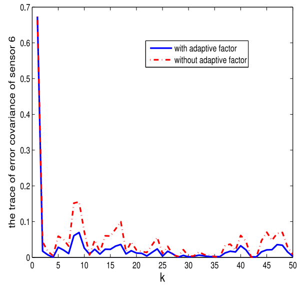


Fig. 5. Adaptive factor's influence to estimation performance.

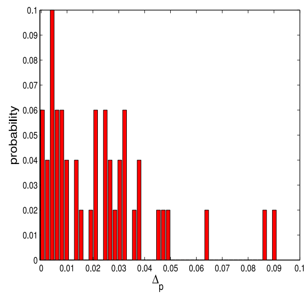


Fig. 6. Frequency distribution histograms of Δ_p.

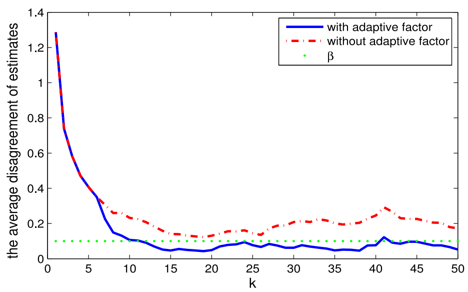


Fig. 7. Average disagreement of estimates among all sensors.

that the designed adaptive filter obtained almost guarantee improved performance. Additionally, Fig. 7 displays the average disagreement of estimates among all filters. It reveals that compared with the filter without adaptive rule, the ADoE of the adaptive filter is decreased, and it is under the preset line β.

It is worth noticing that when the inequality (35) does not hold, the filter will divergent. According to the parameters given in this paper, the upper bound of initial value $\alpha_i(0)$ should be 4.9. So, to verify α_i has initial limit, set $\alpha_i(0) = 10$ and plot the estimation error of sensor 6 in Fig. 8. From it, one can see that the estimation error is beyond acceptable range and the filter can no longer track the target.

In order to verify that the convergence speed of filter is related to the adaptive parameter a, when l and b are fixed, different values of a and the corresponding convergence time step k are recorded in Table I. From Table I, one can see that in the range of $\bar{K} \in (-(\bar{f}/\bar{h}_i), |([1-\bar{f}]/\bar{h}_i)|]$, the convergence time becomes longer with the decrease of a, which is in line with the theory results.

The advantage of the two-stage fusion estimation strategy is further shown in Fig. 9(a) and (b), which plots the true value

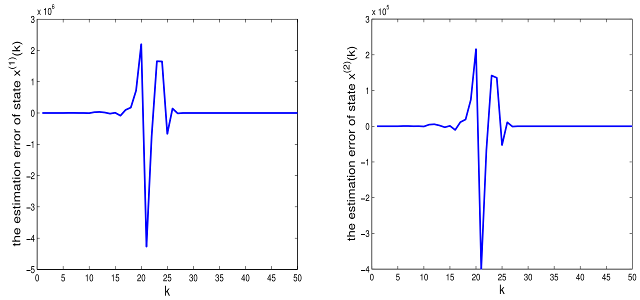
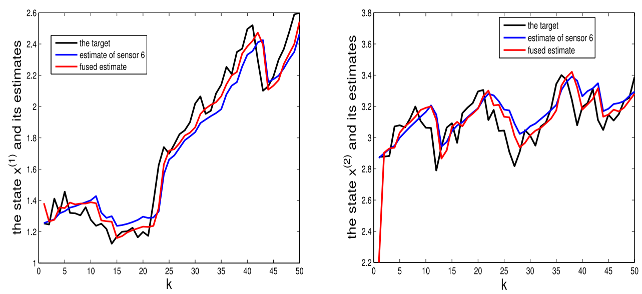


Fig. 8. Estimation error of sensor 6 when the initial value of $\alpha_i(0)$ does not meet the requirement (35).

TABLE I
CONVERGENCE SPEED WITH DIFFERENT a

a	1	0.5	0.1	0.05	0.01	0.005	0.001
k	12	13	17	20	24	31	42



(a) (b)

Fig. 9. Fusion estimate and estimates for (a) $x^{(1)}(k)$ and (b) $x^{(2)}(k)$.

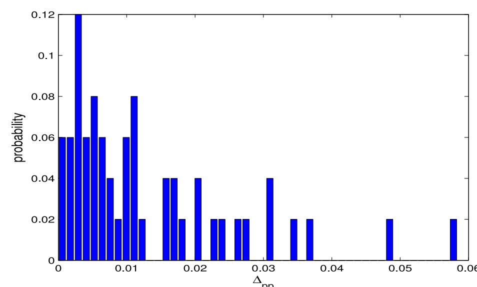


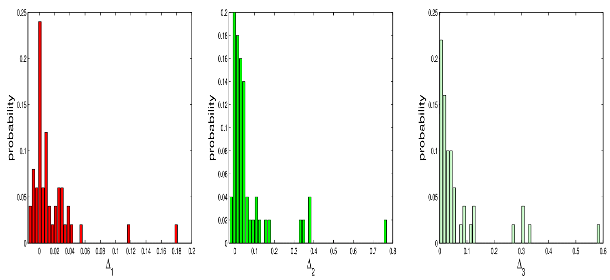
Fig. 10. Frequency distribution histograms of Δ_{pp}.

of the target (the black curve), the fusion estimates (the red curve), and the estimated value of sensor 6 (the blue curve). Here, the fusion estimates is obtained by (41). To clearly characterize the advantage of fusion estimation, the difference of error covariance is plotted in Fig. 10. Analogously, use (44) to specify the decrease of error after using fusion estimation, which is represented as Δ_{pp}. From it, one knows that the fused estimation has almost lower error covariance.

Next, some comparisons between the proposed estimator (ACDF) with the distributed estimator (DF) and the consensus-based distributed estimator (CDF) are made. Take the estimate of the state $x^{(1)}$ as an example to plot the estimation of the estimators. It is clearly shown in Fig. 11 that the trajectory of adaptive consensus-based estimation (ADCF) is the closest to

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

Fig. 11. Comparison between CDF, ACDF, and DF.



(a) (b) (c)

Fig. 12. Frequency distribution histograms of (a) Δ_1, (b) Δ_2, and (c) Δ_3.

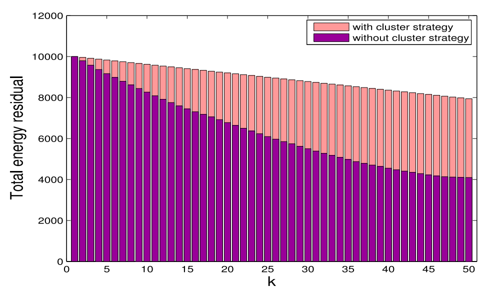


Fig. 13. Total energy residual in sensor network.

the target trajectory, and the difference between the DF trajectory and the target is the greatest. Similarly, the difference of error covariance for ACDF and CDF, ACDF and DF, and CDF and DF, denoted as Δ_1, Δ_2, and Δ_3, respectively, are illustrated in Fig. 12(a)–(c), which reveals that the proposed ACDF algorithm has the smallest mean-squared estimation error. This is in line with our expectations.

To elaborate the energy saving performance of the dynamic cluster strategy, the energy residual in the whole sensor network with and without cluster strategy is plotted in Fig. 13. Each sensor is randomly assigned the initial energy value of 5–15 J. The parameters in energy consumption model are $E_{\text{elec}} = 0.002$ J, $p_{\text{fs}} = 0.0013$, $p_{\text{mp}} = 0.0008$, and $d_0 = 87$ m. Suppose that sensors transmit information to others need $m_e = 1$ bit data, each sensor estimates 1-bit data consumes $E_e = 0.02$ J and the cluster head consumes $E_{\text{DA}} = 0.05$ J when fusing the received estimates, consumes $E_c = 0.001$ J when calculating 1-bit distance data. Besides, sensors within the target 50 m is activated. From Fig. 13, one can intuitively see that the energy consumption of the entire sensor network is greatly reduced after the dynamic cluster establishment mechanism is adopted.

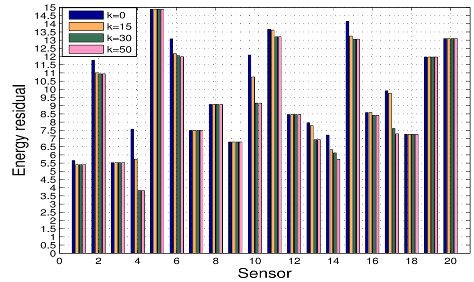


Fig. 14. Energy residual in the first 20 sensors.

In order to further specify the energy consumption in individual sensor, the energy residual of the first 20 sensors at different moments are depicted in Fig. 14. It indicates that some sensors, like 3, 7, 8, and so on, are never activated, so their energy is not lost. Some senors like 4 and 11, due to the leaving of cluster, at $k = 30$ and $k = 50$, the energy does not change. Besides, sensors 6, 14, and 17 are consuming energy all the time since they maintain active in the time step from $k = 0$ to $k = 50$.

VI. Conclusion

In this paper, a distributed estimator with an adaptive factor is designed. By calculating the average disagreement of the estimates, an adaptive factor is proposed and it can improve the tracking performance. A dynamic cluster establishment scheme is proposed to reduce energy consumption in estimation, and a hierarchical two-stage estimation method has been proposed to improve estimation. In future work, analysis on the selection of parameters, how to transmit the data in a more efficient way, and how to satisfy filtering performance under the packet loss, network attack and other situations are the issues to be considered.

References

[1] H. Zhang *et al.*, "Codesign of event-triggered and distributed h_∞ filtering for active semi-vehicle suspension systems," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 2, pp. 1047–1058, Apr. 2017.

[2] H. Li, Y. Pan, and Q. Zhou, "Filter design for interval type-2 fuzzy systems with D stability constraints under a unified frame," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 3, pp. 719–725, Jun. 2015.

[3] J. Sun, T. Li, M. Yu, and H. M. Zhang, "Exploring the congestion pattern at long-queued tunnel sag and increasing the efficiency by control," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2017.2780103.

[4] L. Wu, P. Shi, and H. Gao, "State estimation and sliding-mode control of Markovian jump singular systems," *IEEE Trans. Autom. Control*, vol. 55, no. 5, pp. 1213–1219, May 2010.

[5] H. Zhang, R. Yang, H. Yan, and F. Yang, "h_∞ consensus of event-based multi-agent systems with switching topology," *Inf. Sci.*, vols. 370–371, pp. 623–635, Nov. 2016.

[6] H. Yan, Q. Yang, H. Zhang, F. Yang, and X. Zhan, "Distributed h_∞ state estimation for a class of filtering networks with time-varying switching topologies and packet losses," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2708507.

[7] Q. Liu, Z. Wang, X. He, and D. H. Zhou, "Event-based recursive distributed filtering over wireless sensor networks," *IEEE Trans. Autom. Control*, vol. 60, no. 9, pp. 2470–2475, Sep. 2015.

[8] W.-A. Zhang, G. Feng, and L. Yu, "Multi-rate distributed fusion estimation for sensor networks with packet losses," *Automatica*, vol. 48, no. 9, pp. 2016–2028, 2012.

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

[9] H. Yan, H. Zhang, F. Yang, C. Huang, and S. Chen, "Distributed h_∞ filtering for switched repeated scalar nonlinear systems with randomly occurred sensor nonlinearities and asynchronous switching," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2708507.

[10] W. Zhao and Y. Liang, "Energy-efficient and robust in-network inference in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2105–2118, Oct. 2015.

[11] S. Mahfouz, F. Mourad-Chehade, P. Honeine, J. Farah, and H. Snoussi, "Target tracking using machine learning and Kalman filter in wireless sensor networks," *IEEE Sensors J.*, vol. 14, no. 10, pp. 3715–3725, Oct. 2014.

[12] Y. Liu, Z. Wang, J. Liang, and X. Liu, "Synchronization and state estimation for discrete-time complex networks with distributed delays," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1314–1325, Oct. 2008.

[13] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *Int. J. Robot. Res.*, vol. 12, no. 1, pp. 20–44, 1993.

[14] R. Olfati-Saber, "Distributed Kalman filtering for sensor networks," in *Proc. 46th IEEE Conf. Decis. Control*, New Orleans, LA, USA, 2007, pp. 5492–5498.

[15] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, "SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4782–4795, Dec. 2006.

[16] B. Sinopoli *et al.*, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.

[17] F. Tatari, M.-R. Akbarzadeh-T, M. Mazouchi, and G. Javid, "Agent-based centralized fuzzy Kalman filtering for uncertain stochastic estimation," in *Proc. 5th Int. Conf. Soft Comput. Words Perceptions Syst. Anal. Decis. Control*, 2009, pp. 1–4.

[18] H. Yan, H. Zhang, F. Yang, X. Zhan, and C. Peng, "Event-triggered asynchronous guaranteed cost control for Markov jump discrete-time neural networks with distributed delay and channel fading," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2017.2732240.

[19] J. Gao, A. A. Proctor, Y. Shi, and C. Bradley, "Hierarchical model predictive image-based visual servoing of underwater vehicles with adaptive neural network dynamic control," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2323–2334, Oct. 2016.

[20] C. Peng, J. Zhang, and H. Yan, "Adaptive event-triggering h_∞ load frequency control for network-based power systems," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1685–1694, Feb. 2018.

[21] Y. Xu, C. Liu, J.-Y. Li, C.-Y. Su, and T. Huang, "Finite-horizon h_∞ state estimation for time-varying neural networks with periodic inner coupling and measurements scheduling," *Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2018.2791575.

[22] S. Zhu, C. Chen, W. Li, B. Yang, and X. Guan, "Distributed optimal consensus filter for target tracking in heterogeneous sensor networks," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1963–1976, Dec. 2013.

[23] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[24] J. Qin, Q. Ma, Y. Shi, and L. Wang, "Recent advances in consensus of multi-agent systems: A brief survey," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 4972–4983, Jun. 2017.

[25] Y. Hong, J. Hu, and L. Gao, "Tracking control for multi-agent consensus with an active leader and variable topology," *Automatica*, vol. 42, no. 7, pp. 1177–1182, 2006.

[26] R. Olfati-Saber and J. S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proc. 44th IEEE Conf. Decis. Control*, Seville, Spain, 2005, pp. 6698–6703.

[27] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[28] M. S. Stanković, S. S. Stanković, and D. M. Stipanović, "Consensus based overlapping decentralized estimation with missing observations and communication faults," *Automatica*, vol. 45, no. 6, pp. 1397–1406, 2009.

[29] Y. Yu, "Consensus-based distributed mixture Kalman filter for maneuvering target tracking in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8669–8681, Oct. 2016.

[30] D. P. Spanos, R. O. Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *Proc. 16th IFAC World Congr.*, 2005, pp. 7068–7072.

[31] M. A. Demetriou, "Design of consensus and adaptive consensus filters for distributed parameter systems," *Automatica*, vol. 46, no. 2, pp. 300–311, 2010.

[32] Y. Bar-Shalom, "On hierarchical tracking for the real world," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 3, pp. 846–850, Jul. 2006.

[33] O. N. Gharehshiran and V. Krishnamurthy, "Coalition formation for bearings-only localization in sensor networks—A cooperative game approach," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4322–4338, Aug. 2010.

[34] W. Zhang and G. Cao, "DCTC: Dynamic convoy tree-based collaboration for target tracking in sensor networks," *IEEE Trans. Wireless Commun.*, vol. 3, no. 5, pp. 1689–1701, Sep. 2004.

[35] B. Wu, Y.-P. Feng, H.-Y. Zheng, and X. Chen, "Dynamic cluster members scheduling for target tracking in sensor networks," *IEEE Sensors J.*, vol. 16, no. 19, pp. 7242–7249, Oct. 2016.

[36] X. Wang, H. Zhang, L. Han, and P. Tang, "Sensor selection based on the fisher information of the Kalman filter for target tracking in WSNs," in *Proc. 33rd Chin. Control Conf.*, Nanjing, China, 2014, pp. 383–388.

[37] L. M. Kaplan, "Global node selection for localization in a distributed sensor network," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 113–135, Jan. 2006.

[38] G. Chen, C. Li, M. Ye, and J. Wu, "An unequal cluster-based routing protocol in wireless sensor networks," *Wireless Netw.*, vol. 15, no. 2, pp. 193–207, 2009.

[39] D. Wei, Y. Jin, S. Vural, K. Moessner, and R. Tafazolli, "An energy-efficient clustering solution for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 11, pp. 3973–3983, Nov. 2011.

[40] J. Wu, L. Zhang, Y. Bai, and Y. Sun, "Cluster-based consensus time synchronization for wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1404–1413, Mar. 2015.

[41] W. Yang, G. Chen, X. Wang, and L. Shi, "Stochastic sensor activation for distributed state estimation over a sensor network," *Automatica*, vol. 50, no. 8, pp. 2070–2076, 2014.

[42] H. Zhang, X. Song, and L. Shi, "Convergence and mean square stability of suboptimal estimator for systems with measurement packet dropping," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1248–1253, May 2012.

[43] A. P. Dani, S.-J. Chung, and S. Hutchinson, "Observer design for stochastic nonlinear systems via contraction-based incremental stability," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, Mar. 2015.

[44] D. Ding, Z. Wang, G. Wei, and F. E. Alsaadi, "Event-based security control for discrete-time stochastic systems," *IET Control Theory Appl.*, vol. 10, no. 15, pp. 1808–1815, Oct. 2016.

[45] S.-L. Sun and Z.-L. Deng, "Multi-sensor optimal information fusion Kalman filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, 2004.

[46] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 622–633, May 2008.

Hao Zhang received the B.Sc. degree in automatic control from the Wuhan University of Technology, Wuhan, China, in 2001, and the Ph.D. degree in control theory and control engineering from the Huazhong University of Science and Technology, Wuhan, in 2007.

She is currently a Professor with the School of Electronics and Information Engineering, Tongji University, Shanghai, China. Her current research interests include network-based control systems, multiagent systems, and complex networks.

Xue Zhou received the B.Eng. degree in automation from the Hefei University of Technology, Hefei, China, in 2016. She is currently pursuing the M.Sc. degree in control theory and control engineering with Tongji University, Shanghai, China.

Her current research interests include wireless sensor networks and network security.

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.

Zhuping Wang received the B.Eng. and M.Eng. degrees in automatic control from Northwestern Polytechnic University, Xi'an, China, in 1994 and 1997, respectively, and the Ph.D. degree from the National University of Singapore, Singapore, in 2003.

She is currently a Professor with the College of Electronics and Information Engineering, Tongji University, Shanghai, China. Her current research interests include intelligent control of robotic systems, self-driving vehicles, and nonholonomic control systems.

Huaicheng Yan received the B.Sc. degree in automatic control from the Wuhan University of Technology, Wuhan, China, in 2001, and the Ph.D. degree in control theory and control engineering from the Huazhong University of Science and Technology, Wuhan, in 2007.

He is currently a Professor with the School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China. His current research interests include networked systems and multiagent systems.

Jian Sun received the B.Eng. degree in application engineering of carrier from the Shandong University of Technology, Zibo, China, in 2000, the M.Eng. degree in transportation planning and management from Jilin University, Changchun, China, in 2003, and the Ph.D. degree in traffic information engineering and control from Tongji University, Shanghai, China, in 2006.

He is currently a Professor with the School of Transportation Engineering, Tongji University. His current research interest includes integrated traffic information.

Authorized licensed use limited to: Imperial College London. Downloaded on August 16,2024 at 22:22:49 UTC from IEEE Xplore. Restrictions apply.