

Article

Multi-Sensor Consensus Estimation of State, Sensor Biases and Unknown Input

Jie Zhou, Yan Liang *, Feng Yang, Linfeng Xu and Quan Pan

Key Laboratory of Information Fusion Technology, Ministry of Education, School of Automation, Northwestern Polytechnical University, Xi'an 710129, China; zhoujie@mail.nwpu.edu.cn (J.Z.); yangfeng@nwpu.edu.cn (F.Y.); xulinf@gmail.com (L.X.); quanpan@nwpu.edu.cn (Q.P.)

* Correspondence: liangyan@nwpu.edu.cn; Tel.: +86-29-8843-1305

Academic Editors: Xue-Bo Jin, Feng-Bao Yang, Shuli Sun and Hong Wei

Received: 12 March 2016; Accepted: 23 August 2016; Published: 1 September 2016

Abstract: This paper addresses the problem of the joint estimation of system state and generalized sensor bias (GSB) under a common unknown input (UI) in the case of bias evolution in a heterogeneous sensor network. First, the equivalent UI-free GSB dynamic model is derived and the local optimal estimates of system state and sensor bias are obtained in each sensor node; Second, based on the state and bias estimates obtained by each node from its neighbors, the UI is estimated via the least-squares method, and then the state estimates are fused via consensus processing; Finally, the multi-sensor bias estimates are further refined based on the consensus estimate of the UI. A numerical example of distributed multi-sensor target tracking is presented to illustrate the proposed filter.

Keywords: bias estimation; state estimation; sensor registration; network consensus

1. Introduction

The well-known Kalman filter is optimal given a linear Gaussian model, but its performance will deteriorate in the presence of unknown biases, such as the unknown and time-varying delays in chemical processes [1–4], faults or failures in fault-tolerant diagnosis and control systems [5,6], registration errors in multi-sensor fusion [7–10], or inertial drift in navigation [11–14]. In general, such a bias is represented as an unknown input (UI) to the nominal model. To the best of our knowledge, approaches to UI modeling and the design of corresponding filters typically fall into one of the following categories.

The first type of UI is zero-mean random noise with unknown covariance. In the case of the stationary noise processes in linear dynamic systems, the covariance can be identified via Bayesian or maximum likelihood estimation. A corresponding filter has been applied for orbit determination for near-earth satellites [15,16]. Recently, this type of filter design has been extended to time-varying covariance [17] and jump Markov stochastic systems [18]. Furthermore, an M-robust estimator [19] has been derived for the simultaneous adaptive estimation of unknown states and observation noise statistics. An auto-covariance least-squares method [20] has been presented to achieve lower-variance covariance estimates along with the necessary and sufficient conditions for the uniqueness of the estimated covariances. By treating unknown covariances as missing data, the adaptive estimation problem has been transformed into a problem of the joint optimization of state estimation and parameter identification, allowing it to be solved in an expectation-maximization (EM) iterative processing framework [21].

The second type of UI is unknown but deterministic. Using least-squares estimation and moving-window hypothesis testing, the corresponding filters can cope with cases in which the UI is piecewise constant [22] or a sum of basis functions with piecewise-constant weights [23]. The problem

of state estimation with UIs in both dynamic and measurement models has been addressed using a joint EM optimization scheme for state estimation, parameter identification and iteration termination decision-making [24], and this approach has been extended to a distributed EM algorithm for sensor networks [25], and further extended to an adaptive divided difference filter for nonlinear system with multiplicative parameters [26].

The third type of UI is completely arbitrary, without the availability of any prior knowledge about its evolution. An asymptotically stable and UI-decoupled observer has been derived [27] based on the condition that the rank of its distribution matrix must be less than that of the UI. For the case in which the UI appears only in the process equation, a combination of least-squares estimation and the Kalman filter has been proposed for joint state and UI estimation, and the necessary and sufficient condition for the existence of such a joint estimator has also been presented [28]. In addition, this result has been extended to the case in which the UI appears in both the process and measurement models [29].

The fourth type of UI is norm-bounded, and robust offline filters have been designed to minimize the gain of the transfer function from the UI to the estimation error [30]. As an alternative approach, a method of linear minimax regret estimation has been developed to minimize the worst-case regret over all bounded data [31].

The fifth type of UI is characterized by a randomly switching parameter obeying a known Markov chain. The corresponding solutions fall within the scope of multiple model estimators [32–37], such as the interacting multiple model (IMM), which is well known in maneuvering-target tracking. Motivated by the concept of establishing a general framework for joint state estimation and data association in clutters for the tracking of a maneuvering target, the linear minimum-mean-square-error (MMSE) estimator has been derived for discrete-time Markovian jump linear systems with stochastic coefficient matrices [38].

In general, the filters discussed above are UI-specific because of the significant differences between both the different types of UIs and the solutions designed to address them. However, in many practical applications, much more complicated UIs may be encountered. Recently, the minimum upper bound filter (MUBF) was proposed for a linear stochastic system corrupted by a generalized UI representing an arbitrary linear combination of dynamic UIs, random UIs, and deterministic UIs [39]. The result was further extended to a discrete-time non-linear stochastic system with a UI in its measurements, and an iterative optimization method for joint estimation and parameter identification was derived [40]. In multi-sensor bias estimation [41], the concept of a generalized sensor bias (GSB) has been proposed, which is represented by a dynamic model driven by a structured UI. By deriving an equivalent state-free measurement model and the corresponding UI-free dynamic model for the GSB, the LMMSE can be obtained via the orthogonal principle.

However, the application of the resultant filter is limited to the case in which all sensors have an identical measurement matrix, and only the GSB can be estimated. This problem motivates us to present and solve the problem of the joint estimation of the system state and the GSB under a UI in the case of GSB evolution.

This paper presents the joint estimation of the system state and GSB under a common UI in the case of bias evolution in a heterogeneous sensor network. To the best of the author's knowledge, this is the first attempt to achieve this kind of the joint estimation. In detail, a two-stages cooperative strategy for pursuing consistent estimates in the distributed network is proposed. First, a joint UI-free evolution model is derived, along with its necessary and sufficient conditions for existence. In the derived model, the process noise and measurement noise are correlated. And the optimal local MMSE estimator with correlated noise is obtained. Second, based on the local estimates, the global estimation of the target state and UI are further achieved by network consensus. Through utilizing the global estimation of UI, the GSB is refined. The optimal estimates of the target state, GSB and UI are finally obtained.

Throughout the paper, the superscripts “ -1 ”, “ T ” and “ $+$ ” represent the inverse, transpose, and Moore-Penrose operations, respectively; “ I ” and “ 0 ” represent the identity matrix and the zero matrix, respectively, of the proper dimensions; “ $0_{m \times n}$ ” represents the zero matrix with dimensions of $m \times n$;

$\text{diag}\{\cdot\}$ indicates a diagonal matrix; $\text{col}\{\cdot\}$ denotes column augmentation; $E\{\cdot\}$ denotes the operator of mathematical expectation; the superscript i indicates a matrix or variable related to the i th sensor; $\text{rank}\{\cdot\}$ denotes the rank of the specified matrix; the subscripts “ i ” and “ j ” indicate the i th and j th subblocks of a matrix, respectively; the superscripts “ \wedge ” and “ \sim ” indicate the estimate and residual, respectively, of a random vector; the subscripts “ $k|k+1$ ” represents the estimate of a vector at the time k using measurements up to time $k+1$ and \otimes denotes the Kronecker product.

2. Problem Formulation

Sensor bias (SB) is widespread in multi-sensor systems and originates from two types of sources: SBs of the first type are local SBs (LSBs), which are independent among sensors, such as calibration error, navigation bias and sensor faults/drift, whereas sensor biases of the second type are global SB (GLSB, also called UI in this work), which are common to all sensors, such as electronic countermeasures (ECMs) in target tracking. A GSB model has been proposed based on dynamic evolution driven by both independent LSBs and a common UI [41]. However, the corresponding filter has two shortcomings: first, all sensors must have an identical measurement matrix, and second, the system state and the UI cannot be estimated simultaneously because the SB estimation is obtained by deriving the equivalent GSB model neglecting system state and UI. Here, we present the two assumptions adopted in our work.

Assumption 1. All measurements made by sensors in the network are continuously interrupted by the GSB, which includes LSBs and GLSB.

Assumption 2. The network topology is fixed, and the target can be monitored by all sensors throughout the entire process. There is no fusion centre or lead sensor in this sensor network. In other words, all sensors are equal in status, and they reach a network consensus by exchanging information with their neighbours.

Consider a scenario of collaborative target tracking within a sensor network, as shown in Figure 1. A target is continuously moving within a surveillance zone. Each sensor node obtains raw measurements of the target corrupted by the GSB. At each step, each sensor obtains local estimates of the target state and GSB, exchanges its local system-state estimate with its neighbors, and obtains fused estimates of the target state and UI via network consensus. In other words, this sensor fusion process can be regarded as the joint distributed estimation of the target state, GSB and UI without the limitation that all sensor measurement matrices must be identical.

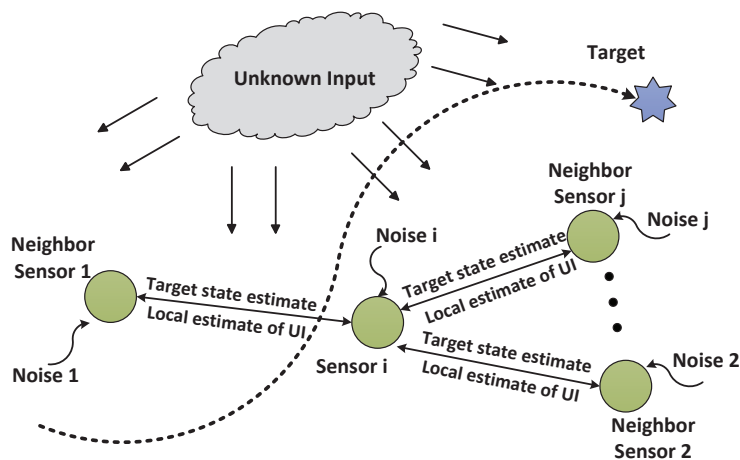


Figure 1. Collaborative target tracking in sensor network.

Consider a network of s sensor nodes. The network topology is represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{1, 2, \dots, s\}$ denotes the set of nodes; $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of permissible communication links; and $\mathcal{A} = [a_{ij}]$ is the graph topology matrix, the elements of which are defined as

$$a_{ij} = \begin{cases} 1, & (i, j) \in \mathcal{E} \text{ or } i = j \\ 0, & (i, j) \notin \mathcal{E} \end{cases}$$

Let $\mathcal{N}_i = \{i \in \mathcal{V} : a_{ij} > 0, i \neq j\}$ denote the set of neighbors of the i th sensor, and let $\mathcal{J}_i = \mathcal{N}_i \cup \{i\}$ denote the inclusive neighbor set of sensor i . Here, the presented network is assumed to be a connected graph. Otherwise, at least two separate sub-networks would exist, and hence, information could only be fused within each sub-network instead of throughout the entire network because information could not be exchanged between the two.

Here, we formulate the following problem of distributed sensor fusion in the presence of time-varying sensors subject to a common UI:

$$x_{k+1} = \Phi_k x_k + \zeta_k, \quad (1)$$

$$y_{k+1}^i = H_{k+1}^i x_{k+1} + N_{k+1}^i b_{k+1}^i + v_{k+1}^i \quad (2)$$

$$b_{k+1}^i = F_k^i b_k^i + G_k^i d_k + w_k^i \quad (3)$$

where $x_k \in \mathbb{R}^n$ is the state vector; $y_k^i \in \mathbb{R}^{m_i}$ is the measurement vector of the i th sensor; $b_k^i \in \mathbb{R}^{p_i}$ is the i th GSB, where $i = \{1, 2, \dots, s\}$; $d_k \in \mathbb{R}^q$ is the common UI; and ζ_k , w_k^i and v_k^i are all independent, zero-mean, Gaussian, white-noise components with covariances of Q_k , S_k^i and R_k^i , respectively. The initial target state x_0 and i th GSB state b_0^i are Gaussian distributed with respective known means of \bar{x}_0 and \bar{b}_0^i and associated covariances of Σ_0^x and $\Sigma_0^{i,b}$. Φ_k , H_k^i , N_k^i , F_k^i , and G_k^i are known and have appropriate dimensions. Under the assumption that $m^i > p^i \geq q^i$, the G_k^i are of full column rank.

Our aim is to design an unbiased minimum-variance filter to jointly estimate the target state x_k , the GSBs b_k^i and the UI d_k given the distributed measurements recorded up to time k .

Remark 1. The GSB b_k^i in Equation (3) represents a generalized bias which includes multiple bias types as the special case:

- (1). If $F_k^i = I$, $G_k^i = 0$, $S_k^i = 0$, then $b_{k+1}^i = b_k^i$ represents a constant-value bias.
- (2). If $F_k^i \neq 0$, $G_k^i = 0$, then $b_{k+1}^i = F_k^i b_k^i + w_k^i$ represents a time-varying bias with known dynamic model.
- (3). If both $F_k^i = 0$, $G_k^i = 0$, then $b_{k+1}^i = w_k^i$ represents a zero-mean random error.
- (4). If $G_k^i \neq 0$, then b_k^i represents the dynamically evolving bias driven by arbitrary UI.

The second type of the UI mentioned in the introduction is equivalent to the case (1).

The third type of the UI mentioned in the introduction is equivalent to the case (4).

The first and fourth types of the UI mentioned in the introduction can be contributed to the case (4) when these two types of UI can be decoupled from Equation (3).

The above four types of UI can be accommodated within the single-model system proposed in this work. However, for the fifth type of UI mentioned in the introduction, it is usually regarded as a randomly switching parameter obeying a known Markov chain which is classified to the multiple-models system, so this type of UI cannot be accommodated within the presented model in this work. Nevertheless, for this type of UI, if GSB model can be established for the each mode of Markov chain. According to Equation (3), GSB model can be written as

$$b_{k+1}^{i,j} = F_k^{i,j} b_k^{i,j} + G_k^{i,j} d_k + w_k^{i,j}, j = 1, 2, \dots, r \quad (4)$$

where r represents the number of states of the Markov chain. Meanwhile, according to Equation (10), the decoupling condition of the corresponding GSB model can be also confirmed as

$$\text{rank} \left(\bar{H}_{k+1}^i \bar{G}_k^{i,j} \right) = \text{rank} \left(\bar{G}_k^{i,j} \right), j = 1, 2, \dots, r \quad (5)$$

These decoupling conditions for a UI obeying a Markov chain are more stringent than those considered in this work, as Equation (5) must hold for every r . Then the fifth type of UI can be accommodated within the proposed framework.

Remark 2. There are two distinct strategies of networked sensor fusion. The first is the centralized strategy [41,42], i.e., all sensors transmit their measurements to a fusion centre, which is responsible for integrated data processing for state estimation and/or parameter identification. The second is the distributed strategy [25], i.e., each sensor processes its own measurements and then shares its processing results with its neighbors in an iterative manner to achieve consistent fusion. Obviously, the distributed fusion strategy allows the computational burden to be shared among sensors and consequently is more desirable for large-scale networks. However, distributed fusion is much more complex, involving both local processing and global fusion. Currently, distributed fusion in the presence of time-varying sensor biases driven by a common UI remains an open problem.

Remark 3. Concerning our proposed problem, there exist two possible solution approaches. One is iterative optimization between state estimation and UI identification, as in the well-known expectation-maximization (EM) scheme [25]. As shown later via simulation, the EM scheme is not desirable here because it must treat the UI as a constant or slowly varying parameter in each iterative window. The other approach is UI decoupling plus state estimation, which serves as the basis for our proposed method, as derived later. The main technical difficulty of this approach is to determine how to adaptively decouple the UI and then fuse the local state and UI estimates.

3. Main Results

Because of the presence of the UI, the target state and GSB cannot be estimated directly. In this section, an equivalent UI-free dynamic model of the bias is derived and the necessary and sufficient conditions for the UI-free model are explored. Afterwards, optimal estimators, in the sense of minimum variance, are proposed for the target state x_k and the local GSBs b_k^i .

3.1. Local LMMSE Filter for System State and Local SB

Lemma 1. For matrices Γ , Ψ and I with proper dimensions, if the matrix Ψ is of full column rank, the equality

$$\Psi \left[I - (\Gamma\Psi)^+ (\Gamma\Psi) \right] = 0 \quad (6)$$

will hold if and only if

$$\text{rank}\{\Gamma\Psi\} = \text{rank}\{\Psi\} \quad (7)$$

Proof. See Appendix A. \square

For convenience of derivation, the model defined in Equations (1) and (2) can be rewritten as follows:

$$z_{k+1}^i = A_k^i z_k^i + \bar{G}_k^i d_k + \bar{\zeta}_k^i \quad (8)$$

$$y_{k+1}^i = \bar{H}_{k+1}^i z_{k+1}^i + v_{k+1}^i \quad (9)$$

where $z_k^i = \text{col}\{x_k, b_k^i\}$, $A_k^i = \text{diag}\{\Phi_k, F_k^i\}$, $\bar{G}_k^i = \text{col}\{0_{n \times q}, G_k^i\}$, $\bar{\zeta}_k^i = \text{col}\{\zeta_k, w_k^i\}$ and $\bar{H}_k^i = \begin{bmatrix} H_k^i & N_k^i \end{bmatrix}$. It is easy to find that $\bar{S}_k^i \triangleq E \left[\bar{\zeta}_k^i (\bar{\zeta}_k^i)^T \right] = \text{diag}\{Q_k, S_k^i\}$.

Theorem 1. (UI decoupling). Consider the system model defined in Equations (8) and (9). If and only if

$$\text{rank} \left(\bar{H}_{k+1}^i \bar{G}_k^i \right) = \text{rank} \left(\bar{G}_k^i \right) \quad (10)$$

then the UI will be decoupled from z_{k+1}^i , i.e., there will exist the following UI-free dynamic model of the sensor bias:

$$z_{k+1}^i = \bar{F}_k^i A_k^i z_k^i + \bar{C}_k^i y_{k+1}^i + \bar{B}_k^i \bar{w}_k^i \quad (11)$$

where

$$\Pi_k^i = \left(\bar{H}_{k+1}^i \bar{G}_k^i \right)^+ \quad (12)$$

$$\bar{C}_k^i = \bar{G}_k^i \Pi_k^i \quad (13)$$

$$\bar{F}_k^i = I_{n+p^i} - \bar{C}_k^i \bar{H}_{k+1}^i \quad (14)$$

$$\bar{w}_k^i = \text{col} \{ \bar{z}_k^i, v_{k+1}^i \} \quad (15)$$

$$\bar{B}_k^i = \begin{bmatrix} \bar{F}_k^i & -\bar{C}_k^i \end{bmatrix} \quad (16)$$

Proof. See Appendix B. \square

Remark 4. If the matrices Γ and Ψ satisfy Equation (7), then the row rank of the matrix Γ should be no less than the column rank of the matrix Ψ in Lemma 1. As shown in Equation (10), this means that the maximum number of UIs that can be decoupled from the system must be fewer than the number of independent measurement components of all sensors.

As shown in Equation (11), the measurements y_k^i are introduced into the equivalent bias dynamic model to decouple d_k from b_k^i . Defining $\bar{Q}_k^i \doteq E \left(\bar{w}_k^i (\bar{w}_k^i)^T \right)$ and $M_{k+1}^i \doteq E \left(\bar{B}_k^i \bar{w}_k^i (v_{k+1}^i)^T \right)$, we have

$$\bar{Q}_k^i = \text{diag} \{ \bar{S}_k^i, R_{k+1}^i \} \quad (17)$$

$$M_{k+1}^i = -\bar{C}_k^i R_{k+1}^i \quad (18)$$

Here, $M_{k+1}^i \neq 0$ means that the process noise \bar{w}_k^i is correlated with the measurement noise v_k^i . Thus, designing an unbiased minimum-variance linear filter for the system defined in Equations (1) and (2) is equivalent to finding the MMSE estimation for the system defined in Equations (9) and (11) with correlated process noise and measurement noise.

Theorem 2. The MMSE estimators for the UI-free system defined in Equations (9) and (11) have the following recursion relations:

$$\hat{z}_{k+1|k}^i = \bar{F}_k^i A_k^i \hat{z}_{k|k}^i + \bar{C}_k^i y_{k+1}^i \quad (19)$$

$$P_{k+1|k}^i = \bar{F}_k^i A_k^i P_{k|k}^i \left(\bar{F}_k^i A_k^i \right)^T + \bar{B}_k^i \bar{Q}_k^i \left(\bar{B}_k^i \right)^T + \bar{C}_k^i R_{k+1}^i \left(\bar{C}_k^i \right)^T \quad (20)$$

$$\hat{z}_{k+1|k+1}^i = \hat{z}_{k+1|k}^i + K_{k+1}^i \left[y_{k+1}^i - \bar{H}_{k+1}^i \hat{z}_{k+1|k}^i \right] \quad (21)$$

$$P_{k+1|k+1}^i = P_{k+1|k}^i - K_{k+1}^i \left[\bar{H}_{k+1}^i P_{k+1|k}^i + \left(M_{k+1}^i \right)^T \right] \quad (22)$$

where the estimator gain is

$$K_{k+1}^i = \left[P_{k+1|k}^i \left(\bar{H}_{k+1}^i \right)^T + M_{k+1}^i \right] \left[\bar{H}_{k+1}^i P_{k+1|k}^i \left(\bar{H}_{k+1}^i \right)^T + \bar{H}_{k+1}^i M_{k+1}^i + \left(\bar{H}_{k+1}^i M_{k+1}^i \right)^T + R_{k+1}^T \right]^{-1} \quad (23)$$

Proof. The term $\bar{C}_k^i y_{k+1}^i$ is an additive known input. Equations (9) and (11) are linear, with additional Gaussian noise. Hence, the optimal filter with correlated process noise and measurement noise [43] can be directly utilized. \square

The framework and pseudo-code for the proposed local MMSE estimation are presented in Figure 2 and Table 1, respectively.

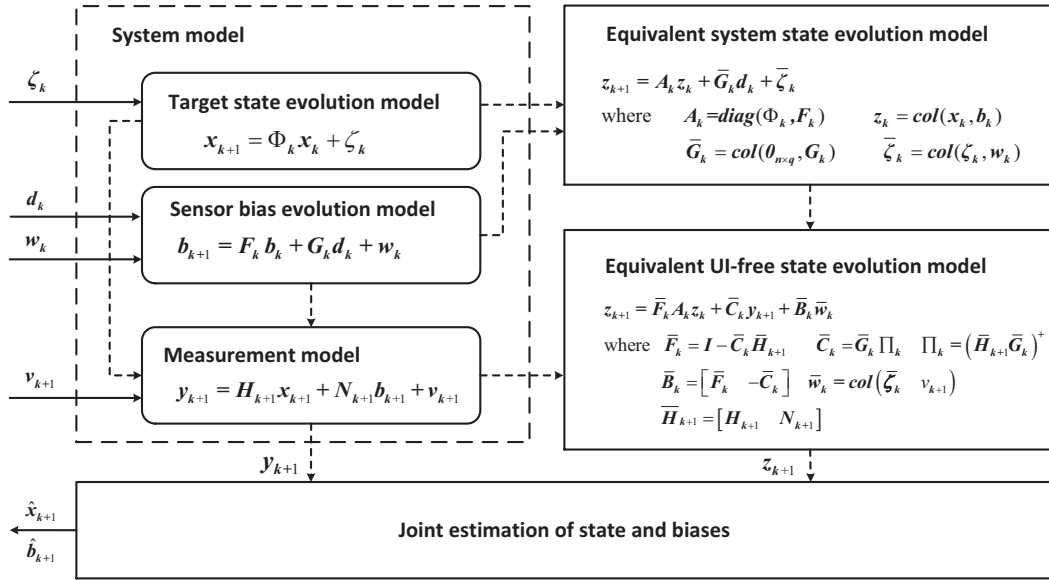


Figure 2. The scheme of joint MMSE estimation of the state and the SB.

Table 1. Pseudo-code for the local MMSE estimation.

<p>Check the Applicability of the Method Step 1: Check the existence condition given by Equation (10). If the condition is satisfied, then perform the following computation. Otherwise, the proposed method is not applicable.</p>
<p>Offline Equivalence Transformation Step 2: Construct the equivalent system-state evolution model by defining \bar{F}_k^i, A_k^i, \bar{C}_k^i, \bar{B}_k^i, and \bar{w}_k^i using Equations (12)–(16). Step 3: Determine the parameters Π_k^i and M_k^i using Equations (12) and (18).</p>
<p>Online Estimator Implementation Step 4: Initialize the estimators: $P_{0 0}^i = P^i(0)$ and $z_0^i = z^i(0)$. Step 5: Compute the predictions $\hat{z}_{k+1 k}^i$ and the corresponding covariances $P_{k+1 k}^i$ using Equations (19) and (20). Step 6: Calculate the filter gains K_{k+1}^i using Equation (23). Step 7: Obtain the $\hat{z}_{k+1 k+1}^i$ and the corresponding covariances $P_{k+1 k+1}^i$ using Equations (21) and (22).</p>
<p>Step 8: Set $k = k + 1$ and go to Step 1.</p>

3.2. Consensus Estimation of the Target State and the UI

Given the local estimate $\hat{z}_{k+1|k+1}^i$ of the i th sensor derived in the previous section, each sensor exchanges its own estimates with its neighbours to reach consistent consensus estimates of x_k and d_k . The corresponding consensus estimates consist of two parts: an average consensus filter for x_k and a distributed least-squares (DLS) estimate for d_k .

3.2.1. Local LS Estimator of UI

Let us rewrite the sensor bias evolution model defined in Equation (3) as

$$\begin{aligned} G_k^i d_k &= b_{k+1}^i - F_k^i b_k^i - w_k^i \\ &= \hat{b}_{k+1|k+1}^i - F_k^i \hat{b}_{k|k}^i + \left(\tilde{b}_{k+1|k+1}^i - F_k^i \tilde{b}_{k|k}^i - w_k^i \right) \\ &= \bar{b}_k^i + \zeta_k^i \end{aligned} \tag{24}$$

where $\bar{b}_k^i = \hat{b}_{k+1|k+1}^i - F_k^i \hat{b}_{k|k}^i$ and $\zeta_k^i = \tilde{b}_{k+1|k+1}^i - F_k^i \tilde{b}_{k|k}^i - w_k^i$. $\tilde{b}_{k+1|k+1}^i$ is the unbiased estimate of z_{k+1}^i in Section 2. It is easily confirmed that $E(\zeta_k^i) = 0$. We can rewrite Equation (24) as the following compact expression:

$$\mathcal{G}_k d_k = \mathcal{B}_k + \zeta_k$$

where $\mathcal{G}_k = \text{col} \{G_k^j : j \in \mathcal{J}_i\}$, $\mathcal{B}_k = \text{col} \{b_k^j : j \in \mathcal{J}_i\}$, and $\zeta_k = \text{col} \{\zeta_k^j : j \in \mathcal{J}_i\}$. Furthermore, based on the local GSB estimates of the i th node and its neighbours, the local DLS estimate of d_k and the corresponding covariance of the i th sensor are obtained as follows:

$$\hat{d}_k^i = \mathcal{G}_k^T (\mathcal{G}_k)^{-1} \mathcal{G}_k^T \mathcal{B}_k \tag{25}$$

$$P_k^{dd,i} = (\mathcal{G}_k^T \mathcal{G}_k)^{-1} \tag{26}$$

3.2.2. Average Consensus Filter for the Target State and the UI

Now, estimates of the target state and the UI have been obtained, but they are local. In other words, the estimates from different nodes are not identical to each other. Therefore, an average consensus fusion (ACF) is presented here to obtain a uniform result at each node via iterative interaction.

Let $\mathcal{W}(l) = [w_{ij}(l)]$ denote the linear weight matrix in the l th step of iteration. Here, $w_{ij}(l)$ is the measurement weight of node j at node i , which satisfies

$$\begin{cases} \sum_{j=1}^s w_{ij}(l) = 1 \\ \sum_{i=1}^s w_{ij}(l) = 1 \\ w_{ij}(l) \geq 0 \end{cases}$$

Let $\mathcal{W}_k^x(l) = [w_{ij}^x(l)]$ and $\mathcal{W}_k^d(l) = [w_{ij}^d(l)]$ represent the weight matrices for the target state and the UI, respectively.

Given the consensus covariances of the state estimate and bias estimate of node i after the l th iteration, denoted by $P_{k|k}^{xx,i}(l)$ and $P_{k|k}^{dd,i}(l)$, respectively, we design the weight matrices as follows:

$$\begin{cases} w_{ij}^x(l+1) = \frac{\text{tr} \left[\left(P_{k|k}^{xx,j}(l) \right)^{-1} \right]}{\sum_{t \in \mathcal{J}_i} \text{tr} \left[\left(P_{k|k}^{xx,t}(l) \right)^{-1} \right]} & j \in \mathcal{J}_i \\ w_{ij}^x(l+1) = 0 & \text{otherwise} \end{cases}$$

and

$$\begin{cases} w_{ij}^d(l+1) = \frac{\text{tr} \left[\left(P_{k|k}^{dd,j}(l) \right)^{-1} \right]}{\sum_{t \in \mathcal{J}_i} \text{tr} \left[\left(P_{k|k}^{dd,t}(l) \right)^{-1} \right]} & j \in \mathcal{J}_i \\ w_{ij}^d(l+1) = 0 & \text{otherwise} \end{cases}$$

such that local estimates of higher accuracy will be assigned higher weights in the weighted fusion process.

Then, the initial values can be set to $\hat{x}_{k|k}^i(0) \doteq \hat{x}_{k|k}^i$, $\hat{d}_k^i(0) \doteq \hat{d}_k^i$, $P_{k|k}^{xx,i}(0) \doteq P_{k|k}^{xx,i}$, and $P_k^{dd,i}(0) \doteq P_k^{dd,i}$, where $\hat{x}_{k|k}^i$ and $P_{k|k}^{xx,i}$ are subblocks of $\hat{z}_{k|k}^i$ and $P_{k|k}^i$, respectively, i.e., $\hat{z}_{k|k}^i = \begin{bmatrix} \hat{x}_{k|k}^i \\ \hat{b}_{k|k}^i \end{bmatrix}$ and $P_{k|k}^i = \begin{bmatrix} P_{k|k}^{xx,i} & P_{k|k}^{xb,i} \\ P_{k|k}^{bx,i} & P_{k|k}^{bb,i} \end{bmatrix}$.

Theorem 3. Consider the multi-sensor system defined in Equations (1)–(3), with topology $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ and weight matrices $[w_{ij}^x(l)]$ and $[w_{ij}^d(l)]$ for the target state and UI, respectively, at the l th step of iteration. The ACFs for x_k and d_k are

$$\hat{x}_{k|k}^i(l+1) = \sum_{j \in \mathcal{J}_i} w_{ij}^x(l) \hat{x}_{k|k}^j(l) \quad (27)$$

$$\hat{d}_k^i(l+1) = \sum_{j \in \mathcal{J}_i} w_{ij}^d(l) \hat{d}_k^j(l) \quad (28)$$

where

$$P_{k|k}^{xx,i}(l+1) = \sum_{j \in \mathcal{J}_i} w_{ij}^x(l) \left[P_{k|k}^{xx,j}(l) + \hat{x}_{k|k}^j(l) \left(\hat{x}_{k|k}^j(l) \right)^T \right] - \hat{x}_{k|k}^i(l+1) \left(\hat{x}_{k|k}^i(l+1) \right)^T \quad (29)$$

$$P_k^{dd,i}(l+1) = \sum_{j \in \mathcal{J}_i} w_{ij}^d(l) \left[P_k^{dd,j}(l) + \hat{d}_k^j(l) \left(\hat{d}_k^j(l) \right)^T \right] - \hat{d}_k^i(l+1) \left(\hat{d}_k^i(l+1) \right)^T \quad (30)$$

Proof. See Appendix C. \square

In the i th node, Equations (27)–(30) will remain iterative until

$$\frac{\left| \text{tr} \left[P_{k|k}^{xx,i}(l+1) \right] - \text{tr} \left[P_{k|k}^{xx,i}(l) \right] \right|}{\text{tr} \left[P_{k|k}^{xx,i}(l) \right]} \leq \eta^x \quad (31)$$

and

$$\frac{\left| \text{tr} \left[P_k^{dd,i}(l+1) \right] - \text{tr} \left[P_k^{dd,i}(l) \right] \right|}{\text{tr} \left[P_k^{dd,i}(l) \right]} \leq \eta^d \quad (32)$$

where η^x and η^d are positive thresholds for system state and UI, respectively.

3.2.3. Refinement of the Local Bias Estimates

After the consensus iteration of $\hat{x}_{k|k}^i$ and \hat{d}_{k-1}^i , the sensor biases can be refined according to Equation (3), that is,

$$\hat{b}_{k|k}^i = F_{k-1}^i \hat{b}_{k-1|k-1}^i + G_{k-1}^i \hat{d}_{k-1}^i \quad (33)$$

where \hat{d}_{k-1}^i is the consensus estimate at the k th sampling instant. The covariance of $\hat{b}_{k|k}^i$ is as follows:

$$P_{k|k}^{bb,i} = F_{k-1}^i P_{k-1|k-1}^{bb,i} \left(F_{k-1}^i \right)^T + G_{k-1}^i P_{k-1}^{dd,i} \left(G_{k-1}^i \right)^T + S_{k-1}^i \quad (34)$$

Thus, the estimates $\hat{b}_{k|k}^i$, $P_{k|k}^{bb,i}$, $\hat{x}_{k|k}^i$, $P_{k|k}^{xx,i}$, \hat{d}_{k-1}^i and $P_{k-1}^{dd,i}$ will be output as the final results at the k th sampling instant.

The framework and pseudo-code for the iterative consensus process are presented in Figure 3 and Table 2, respectively.

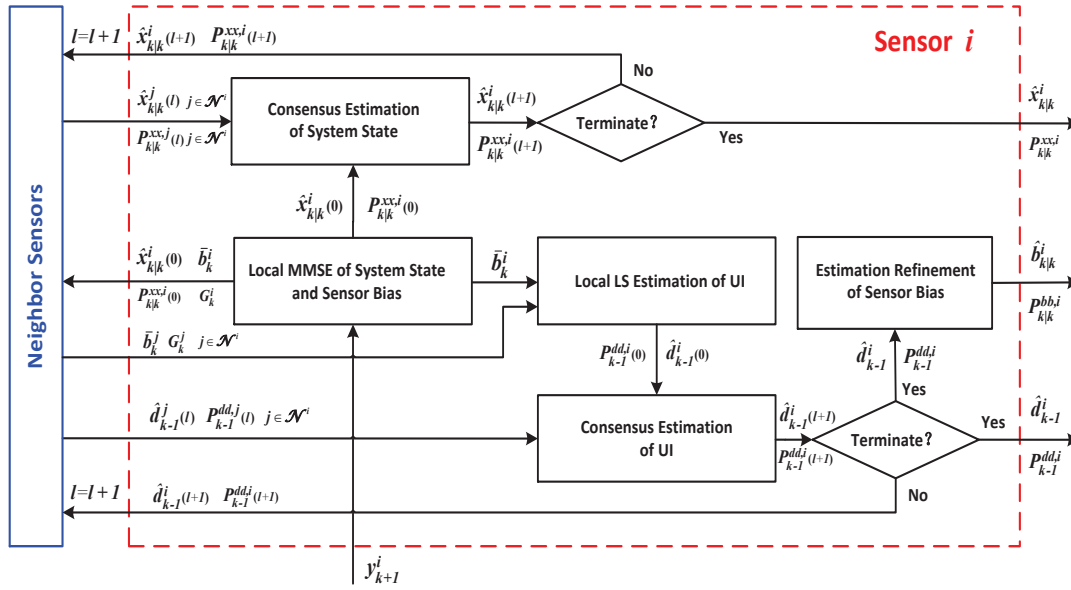


Figure 3. The scheme of the consensus estimation process for the target state and the UI.

Table 2. Pseudo-code for the consensus estimation.

<p>Formulate the Local Information Before Transmission Step 1: Calculate the estimators \bar{b}_k^i using Equation (24).</p>
<p>Exchange Information with All Neighbours of the ith Sensor Step 2: Exchange G_k^i, $\hat{x}_{k k}^i$, $P_{k k}^{xx,i}$, and \bar{b}_k^i with all neighbouring sensors.</p>
<p>Formulate the Local LS Estimators of the UI Step 3: Define the parameters \mathcal{G}_k and \mathcal{B}_k. Step 4: Use the exchanged information to calculate \hat{d}_{k-1}^i and $P_{k-1}^{dd,i}$ via Equations (25) and (26).</p>
<p>Consensus Estimation after Transmission Step 5: Set $l = 1$, $\hat{x}_{k k}^i(0) = \hat{x}_{k k}^i$, $P_{k k}^{xx,i}(0) = P_{k k}^{xx,i}$, $\hat{d}_{k-1}^i(0) = \hat{d}_{k-1}^i$ and $P_{k-1}^{dd,i}(0) = P_{k-1}^{dd,i}$. Step 6: Calculate the ACF estimators $\hat{x}_{k k}^i(l)$ and $P_{k k}^{xx,i}(l)$ using Equations (27) and (29). Step 7: Calculate the ACF estimators $\hat{d}_{k-1}^i(l)$ and $P_{k-1}^{dd,i}(l)$ using Equations (28) and (30). Step 8: If Equations (31) and (32) hold, go to Step 9; otherwise, set $l = l + 1$ and go to Step 6.</p>
<p>Refine the Local Sensor Bias Estimators after Consensus Step 9: Refine the estimates $b_{k k}^i$ and $P_{k k}^{bb,i}$ using Equations (33) and (34). Step 10: Set $k = k + 1$ and go to Step 1.</p>

Remark 5. In this paper, a fusion algorithm for multiple cooperative sensors subject to interference from GSBs is proposed. However, when this algorithm is applied to continuous target tracking in a wireless sensor networks (WSN), a new problem should be considered. Because of the sensors' limited detection capabilities and low energy storage, a WSN is divided into many clusters. Each cluster forms a sub-WSN and is responsible for tracking the target for a certain period of time. When the target moves out of the surveillance area of the current cluster, one or more new clusters must be generated dynamically; thus, a strategy for generating such clusters must be proposed. This strategy must specify how to choose the leader and followers in each cluster and how to transmit estimate information between clusters. This problem will be an interesting topic for future research regarding the joint optimization of signal processing and distributed cooperative detection. Currently, this topic has been tried in our research group [44].

Remark 6. For the case of multiple-target tracking with dense clutter, the association relationships between the targets and measurements are unknown. Therefore, the problem of data association must be considered

when our proposed algorithm is applied for multiple-target tracking. To this end, the optimal estimation for a single target must be extended to the joint optimization of state estimation and target identification for multiple targets. However, because of the limited detection capabilities of nodes in a WSN, as also mentioned in Remark 5, a WSN is often divided into many clusters, each of which monitors only a relatively small region. Consequently, the probability of multiple targets being present in any single cluster is statistically low. Therefore, our algorithm is focused only on single-target tracking.

4. Numerical Example

Consider examples of distributed networks tracking one constant-velocity target. The target state is $[x_k, \dot{x}_k, y_k, \dot{y}_k]^T$, where $[x_k, \dot{x}_k]^T$ and $[y_k, \dot{y}_k]^T$ represent the target's Cartesian position and velocity on the x and y axes, respectively. The sampling interval is $T = 1$ s, and the number of the samples is 60. Each sensor node synchronously measures the position of the target at each sampling instant. The measurements of each sensor node are corrupted by a local dynamic system bias, and all local system biases originate from a common UI. This situation can be modelled using Equations (1) and (3) with the following parameters:

$$\Phi_k = I_2 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, F_b^i = \begin{bmatrix} -0.05 & -0.84 \\ 0.5170 & 0.8069 \end{bmatrix}, G_k^i = \begin{bmatrix} 2.58 \\ -2.5008 \end{bmatrix}, H_k^i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The initial position of the target is $[0, 0]^T$, in units of m , and the initial velocity is $[50, 60]^T$, in m/s . The initial value of the bias is $\bar{b}_0^i = [1, 1]^T$, in m . The covariance matrices are as follows (all in m^2/s^4):

$$R_k^i = \begin{bmatrix} 16 & 0 \\ 0 & 16 \end{bmatrix}, Q_k = 0.01 \times I_2 \otimes \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \times \begin{bmatrix} T^2/2 & T \end{bmatrix}, S_k^i = \begin{bmatrix} 0.36 & 0.342 \\ 0.342 & 0.3249 \end{bmatrix}$$

4.1. A Distributed Network of Twelve Sensors

Consider an example of a distributed network of twelve sensors, and the matrices N_k^i for the 12 sensor nodes are

$$\begin{aligned} N_k^1 &= \text{diag}(3, -5), N_k^2 = \text{diag}(7, 1), N_k^3 = \text{diag}(-3, 7), N_k^4 = \text{diag}(4, -6), \\ N_k^5 &= \text{diag}(7, 3), N_k^6 = \text{diag}(2, -8), N_k^7 = \text{diag}(-7, 5), N_k^8 = \text{diag}(3, 8), \\ N_k^9 &= \text{diag}(2, -5), N_k^{10} = \text{diag}(7, 4), N_k^{11} = \text{diag}(-9, 8), N_k^{12} = \text{diag}(8, 4) \end{aligned}$$

The dimensions of the surveillance area are $16 \text{ km} \times 14 \text{ km}$. The topological map is depicted in Figure 4a, and the sensors' true locations and the target trajectory are depicted in Figure 4b.

The UI, plotted in Figure 5, is

$$d_k = \begin{cases} 0m & 0 < t \leq 5 \\ 5 \times \sin \frac{\pi(t-5)}{10} m & 5 < t \leq 25 \\ 0m & 25 < t \leq 35 \\ 0.5 \times (t - 45) m & 35 < t \leq 55 \\ 0m & 55 < t \leq 60 \end{cases}$$

The histograms shown in Figure 6 depict the numbers of steps of iteration for different thresholds. Clearly, the number of iterations required for consensus increases with a decreasing threshold. Moreover, when the threshold is greater than 0.1, the number of iteration steps varies only slowly. However, when the threshold is less than 0.1, the number of iteration steps increases sharply, and the computational cost is dramatically increased. Considering both calculation and accuracy, the consensus threshold values for both target state and UI estimation are chosen to be $\eta^x = \eta^d = 0.1$.

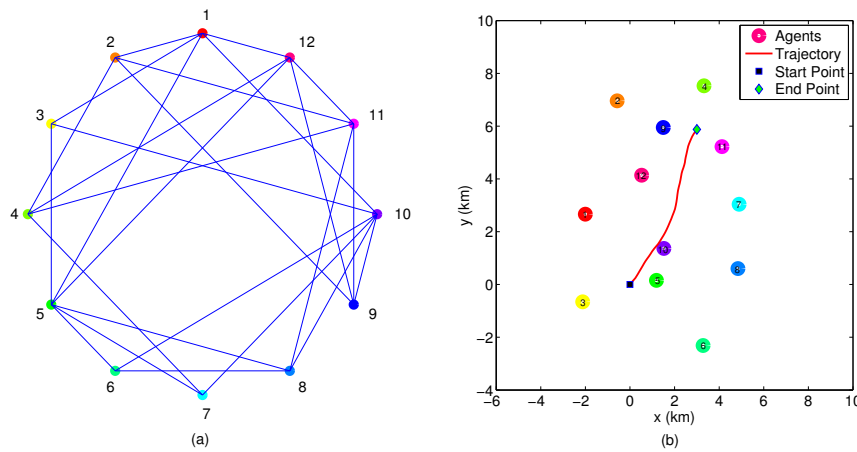


Figure 4. Examples of networks with 12 sensor nodes: (a) Graph-topological map of the network (b) Sensors' real locations and target trajectory.

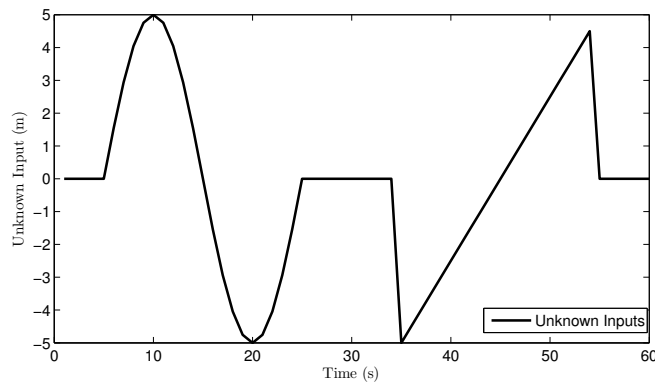


Figure 5. Real unknown input in the multi-sensor system.

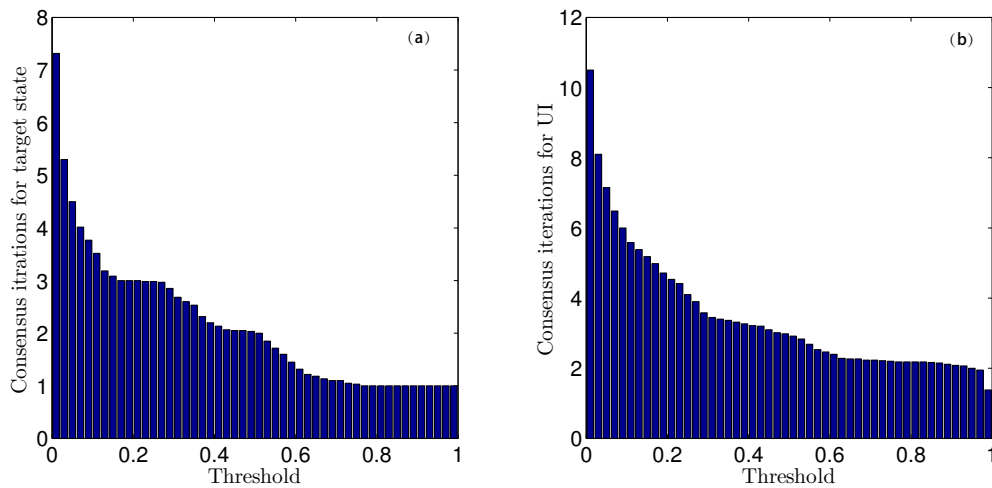


Figure 6. The mean numbers of iteration steps for (a) state and (b) UI estimation with different thresholds.

In accordance with Theorem 2, the rank of the filter matrix is $rank(\bar{H}_{k+1}^i \bar{G}_k^i) = rank(\bar{G}_k^i) = 1$, which satisfies Equation (10).

The proposed algorithm is compared with the EM algorithm [25] in this simulation. The iteration threshold for the EM algorithm is $\delta = 10^{-4}$, and the processing window length is $l = 5$. We also apply

the standard Kalman filter, neglecting the presence of a UI. The estimation of the Kalman filter shows that the fusion performance will deteriorate considerably when bias exists in the sensor measurements.

Because of the page limitation and the similarity in the data variation trends among the simulated sensors, only the estimation results for the fourth sensor are presented here. Figures 7 and 8 show the target-state estimation errors for the Kalman filter (called 'KF'), the EM method, and the proposed method without consensus (called 'DP no consensus') and with consensus (called 'DP consensus'). It can be seen that in the DP methods, the UI is decoupled from the system. Furthermore, the estimation error of the DP consensus method is less than that of the DP no consensus method, and the estimation accuracy is also improved for the DP consensus method. It is also observed that the estimation results based on the KF and EM methods are not as good as those achieved using the DP methods. Two reasons for these findings can be identified. One is that the UI is ignored in the KF method, causing the KF algorithm to become invalid. The other is that the UI is assumed to take a constant value in the iterative process of the EM method. Similar conclusions can also be drawn from the results of the sensor bias estimation, as shown in Figure 9.

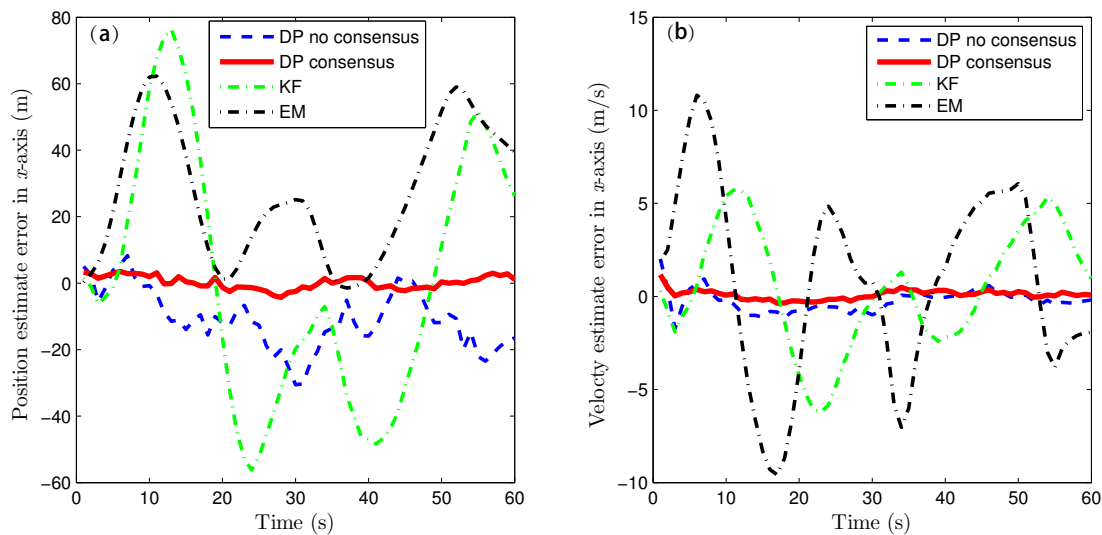


Figure 7. State estimation errors in the x direction using the KF, DP consensus, DP no consensus and EM methods for sensor 4. (a) position estimation; (b) velocity estimation.

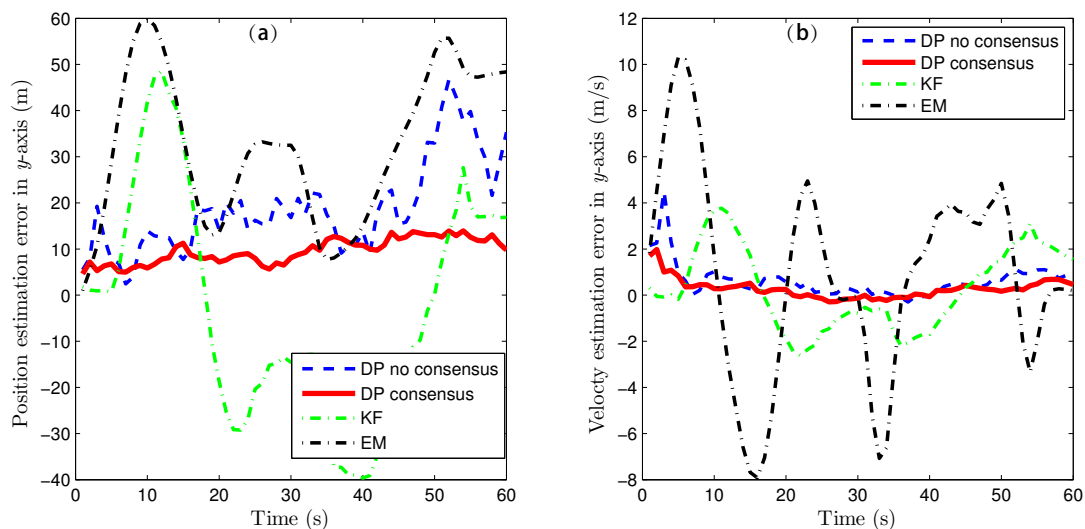


Figure 8. State estimation errors in the y direction using the KF, DP consensus, DP no consensus and EM methods for sensor 4. (a) position estimation; (b) velocity estimation.

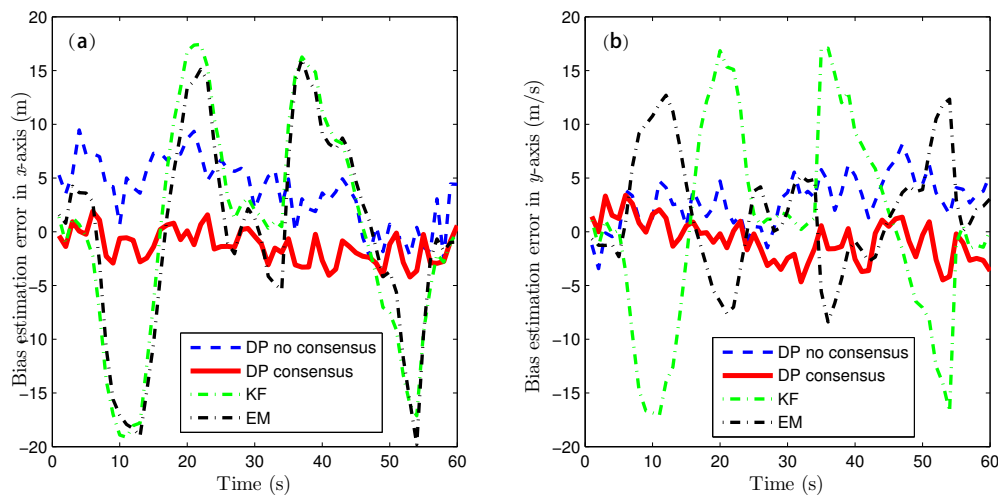


Figure 9. Bias estimation errors using the KF, DP consensus, DP no consensus and EM methods for sensor 4. (a) x-axis; (b) y-axis.

The estimation errors for the UI based on the DP methods with and without consensus are plotted in Figure 10. It can be seen that the UI estimation error is greatly reduced by the consensus processing.

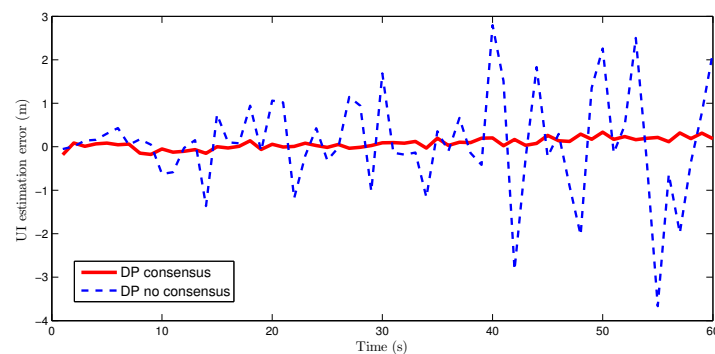


Figure 10. UI estimation errors of the DP consensus and DP no consensus methods for sensor 4.

The mean and peak values of the estimation errors for the target state $[x, \dot{x}, y, \dot{y}]^T$ and the bias $[b_x, b_y]^T$ based on the KF, EM and DP methods are listed in Tables 3 and 4, respectively. It can be seen from Table 3 that the mean estimation error values of the DP consensus method are all smaller than those of the KF, EM and DP no consensus methods. As shown in Table 4, the peak estimation error values are also smallest for the DP consensus method. Thus, the proposed method is proven to be effective.

The run times are also compared as listed in Table 5. It can be seen that the run time for the DP no consensus method is merely 0.33 s because of its non-iteration operation. After iteration, the run time of the DP consensus method is 4.89 s, which is nevertheless shorter than the 6.57 s required for the EM method.

Table 3. Comparison of mean estimation error values.

	x	\dot{x}	y	\dot{y}	b_x	b_y
KF (m)	1.18	0.48	−2.66	0.29	1.16	−0.28
EM (m)	26.83	0.79	32.33	0.91	−6.89	5.76
DP no consensus (m)	−10.87	−0.26	18.81	0.63	2.92	−1.51
DP consensus (m)	0.19	0.09	−9.55	0.27	−1.37	0.39

Table 4. Comparison of peak estimation error values.

	x	\dot{x}	y	\dot{y}	b_x	b_y
KF (m)	76.27	6.21	48.52	3.76	19.11	17.07
EM (m)	62.26	10.81	59.95	10.30	19.84	12.33
DP no consensus (m)	30.55	1.71	46.73	4.56	9.47	8.12
DP consensus (m)	4.28	1.17	13.90	1.98	4.08	4.67

Table 5. Total run times (60 samples).

	EM	DP No Consensus	DP Consensus
time (s)	6.57	0.33	4.89

4.2. A Distributed Network of Sixty Sensors

Now we consider an example of distributed network of sixty sensors, and the matrices N_k^i of the sensor nodes are

$$\begin{aligned}
 N_k^1 = N_k^{13} = N_k^{25} = N_k^{37} = N_k^{49} &= \text{diag}(3, -5), N_k^2 = N_k^{14} = N_k^{26} = N_k^{38} = N_k^{50} = \text{diag}(7, 1), \\
 N_k^3 = N_k^{15} = N_k^{27} = N_k^{39} = N_k^{51} &= \text{diag}(-3, 7), N_k^4 = N_k^{16} = N_k^{28} = N_k^{40} = N_k^{52} = \text{diag}(4, -6), \\
 N_k^5 = N_k^{17} = N_k^{29} = N_k^{41} = N_k^{53} &= \text{diag}(7, 3), N_k^6 = N_k^{18} = N_k^{30} = N_k^{42} = N_k^{54} = \text{diag}(2, -8), \\
 N_k^7 = N_k^{19} = N_k^{31} = N_k^{43} = N_k^{55} &= \text{diag}(-7, 5), N_k^8 = N_k^{20} = N_k^{32} = N_k^{44} = N_k^{56} = \text{diag}(3, 8), \\
 N_k^9 = N_k^{21} = N_k^{33} = N_k^{45} = N_k^{57} &= \text{diag}(2, -5), N_k^{10} = N_k^{22} = N_k^{34} = N_k^{46} = N_k^{58} = \text{diag}(7, 4), \\
 N_k^{11} = N_k^{23} = N_k^{35} = N_k^{47} = N_k^{59} &= \text{diag}(-9, 8), N_k^{12} = N_k^{24} = N_k^{36} = N_k^{48} = N_k^{60} = \text{diag}(8, 4)
 \end{aligned}$$

The dimensions of the surveillance area are 18 km \times 16 km. The topological map is shown in Figure 11a. Sensors' true locations and the target trajectory are depicted in Figure 11b.

In accordance with Theorem 2, the rank of filter matrix $\text{rank}(\bar{H}_{k+1}^i \bar{G}_k^i) = \text{rank}(\bar{G}_k^i) = 1$ for each sensor in the network, which satisfies Equation (10).

The UI, as plotted in Figure 12, is a stochastic noise obeying Gaussian distribution with mean 0 and covariance 15^2 .

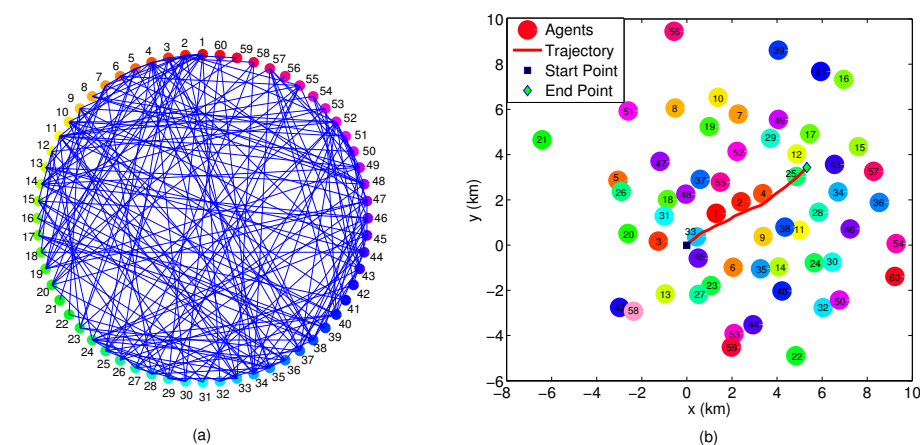


Figure 11. Examples of networks with 60 sensors: (a) Graph-topological map of the network; (b) Sensors' true locations and target trajectory.

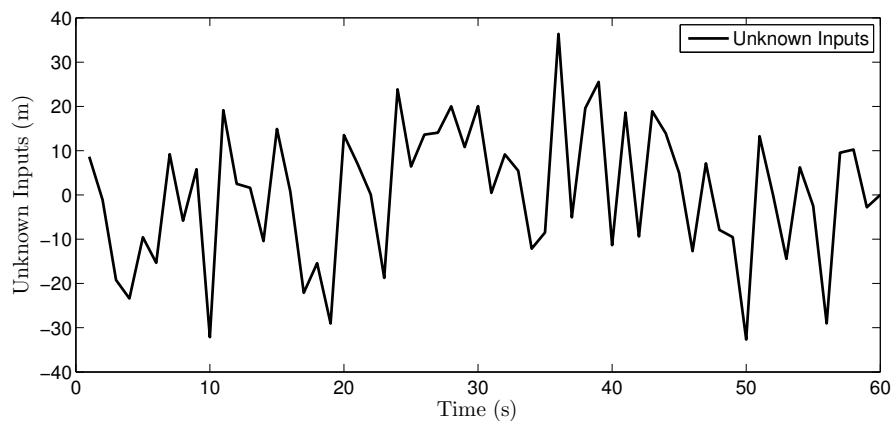


Figure 12. Real unknown input in the multi-sensor system of 60 sensors.

The histograms in Figure 13 depict the mean numbers of consensus iteration with different thresholds. Similarly, the consensus iteration number increases with thresholds decreasing, and when the thresholds are less than 0.1, the iteration step number increase sharply. Therefore, we choose the threshold as $\eta^x = \eta^d = 0.1$. It is interesting to see from the histograms that the consensus iteration needed for UI is roughly one more time than that needed for sensor states.

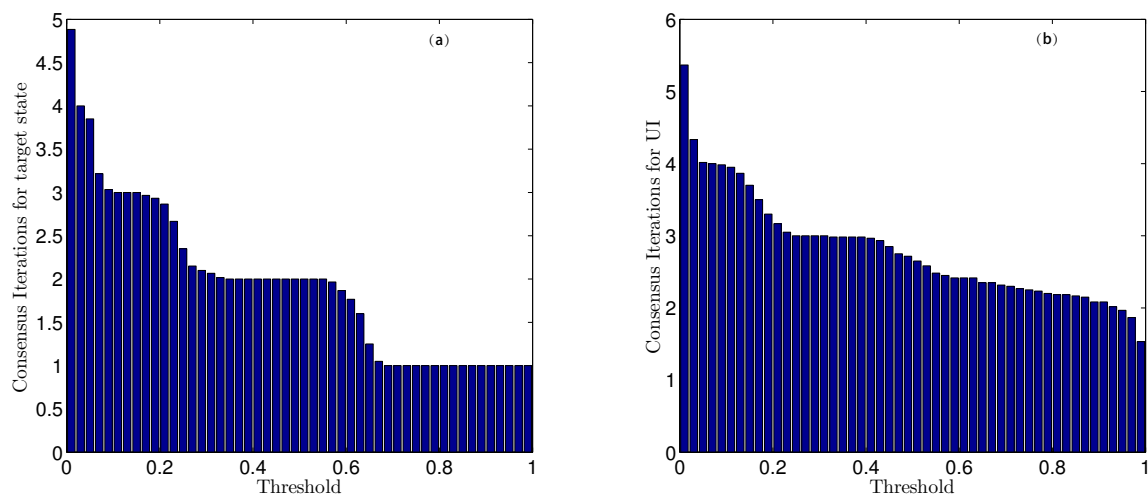


Figure 13. The mean numbers of iteration steps for (a) state and (b) UI estimation with different thresholds in the multi-sensor system of 60 sensors.

Due to the page limitation and the similar data variation trend in simulation figures, only the estimation results for the fourth sensor is presented here. The iteration threshold for EM algorithm is $\delta = 10^{-4}$ and the processing window length is $l = 5$ too. Figures 14–17 show the proposed algorithm estimation results are obviously better than those of KF and EM methods. Indeed, The estimation errors are greatly reduced by the DP consensus operation.

The mean and peak values of the estimation errors of the target state $[x, \dot{x}, y, \dot{y}]^T$ and the bias $[b_x, b_y]^T$ based on the KF, EM and DP methods are correspondingly computed and listed in Tables 6 and 7, respectively. It can be seen from the Tables that our proposed method is proven to be effective.

The run times listed in Table 8 show that the run time of DP consensus is slightly shorter than that of EM method.

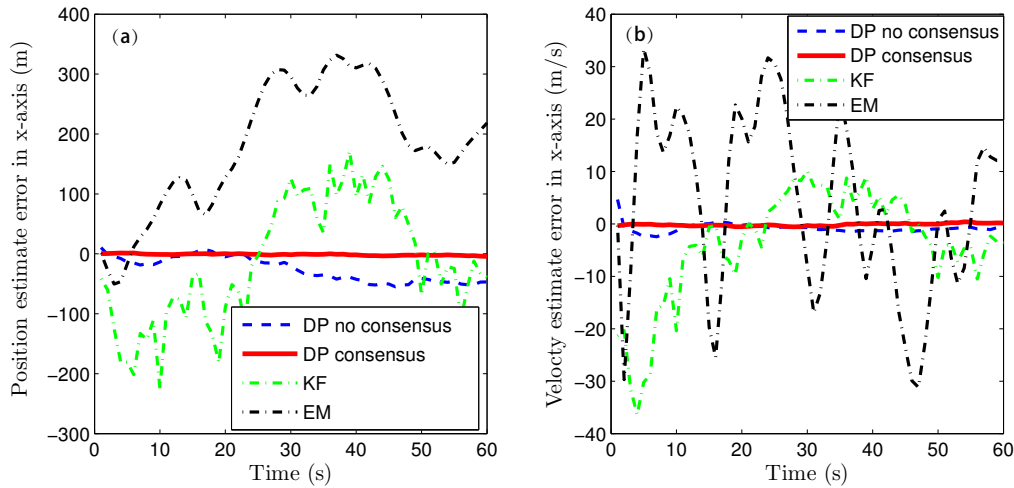


Figure 14. State estimation errors in the x direction using the KF, DP consensus, DP no consensus and EM methods for sensor 4 in the multi-sensor system of 60 sensors. (a) position estimation; (b) velocity estimation.

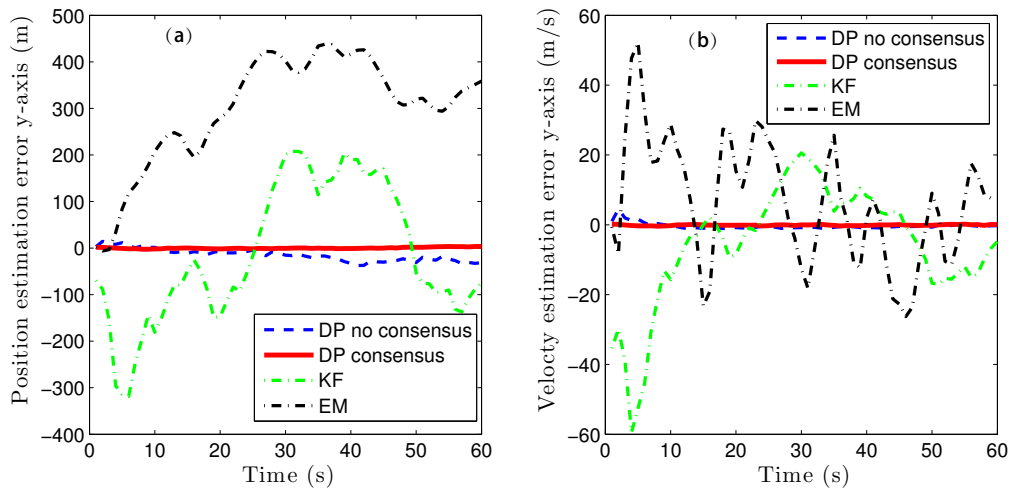


Figure 15. State estimation errors in the y direction using the KF, DP consensus, DP no consensus and EM methods for sensor 4 in the multi-sensor system of 60 sensors. (a) position estimation; (b) velocity estimation.

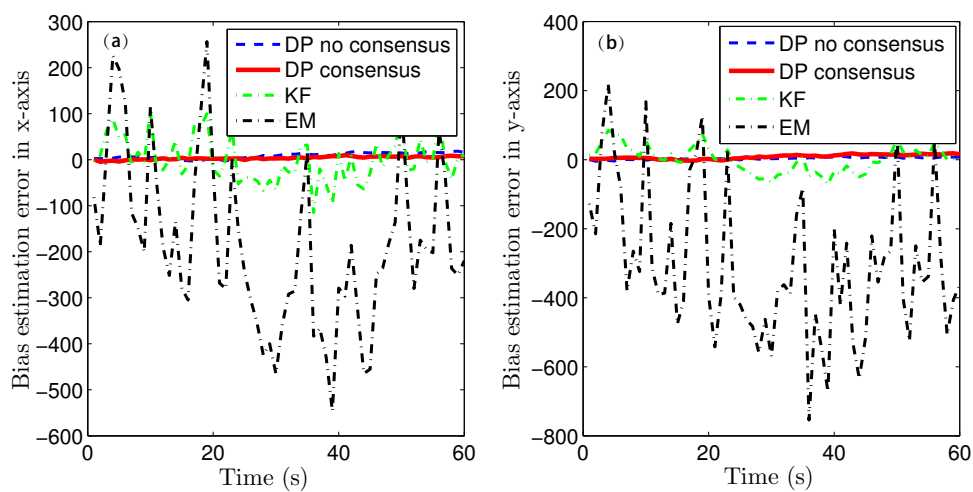


Figure 16. Bias estimation errors using the KF, DP consensus, DP no consensus and EM methods for sensor 4 in the multi-sensor system of 60 sensors. (a) x-axis; (b) y-axis.

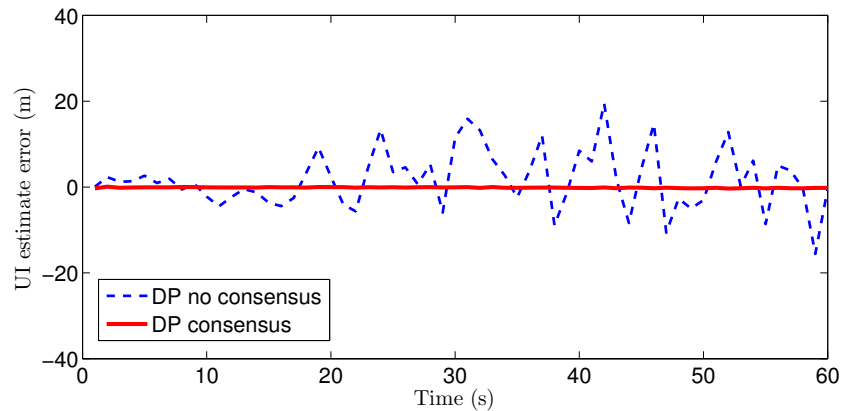


Figure 17. UI estimation errors of the DP consensus and DP no consensus methods for sensor 4 in the multi-sensor system of 60 sensors.

Table 6. Comparison of mean estimation error values of the network of 60 sensors.

	x	\dot{x}	y	\dot{y}	b_x	b_y
KF (m)	-12.721	-3.86	-13.54	-5.72	-0.40	1.19
EM (m)	180.14	4.16	299.97	6.58	-182.47	-306.07
DP no consensus (m)	-20.73	-0.84	-15.03	-0.34	8.56	3.63
DP consensus (m)	-1.21	-0.15	0.063	-0.09	3.63	8.56

Table 7. Comparison of peak estimation error values of the network of 60 sensors.

	x	\dot{x}	y	\dot{y}	b_x	b_y
KF (m)	396.46	80.08	526.54	80.03	220.43	156.67
EM (m)	381.98	64.60	443.66	78.49	803.40	973.76
DP no consensus (m)	65.62	7.12	52.05	5.54	21.61	14.26
DP consensus (m)	5.73	0.94	5.76	0.59	12.91	21.61

Table 8. Total run times (60 samples) of the network of 60 sensors.

	EM	DP No Consensus	DP Consensus
time (s)	27.39	1.65	24.56

4.3. Computational Burden Analysis

Computation burden is also a critical issue to determine the practical applicability of the proposed method and worth to be discussed. Generally speaking, the computational burden is strongly related to the total sensors' number and the connectivity of the sensor network. Thus, the mean values of run times of DP consensus method are tested with different sensor numbers and connectivity probabilities. In the simulation, the network are generated randomly with different connective probabilities. And the simulation program runs ten times with every connective probability and sensor number. The mean values of run time for one sensor of processing 60 samples are recorded to measure the computational burden.

According to Figures 6 and 13, the thresholds are chosen as $\eta^x = \eta^d = 0.1$ in this simulation. Further, Figure 18 depicted the relationship between the mean of the run time and the number of sensor at different connectivity probabilities ('CP' in Figure 18). It can be seen from Figure 18 that the run time increases with sensor number increase. It means more iterations are needed to reach a consensus when sensor number increase. Particularly, when the number of the sensor are 12 and 60 with connectivity probability of 1, the run times reach 0.08 s and 0.28 s, respectively. Furthermore, when the sensor

number fixed, the run time increases with the connectivity probability decreases. This is because that a looser sensor network topology means fewer neighboring sensors, therefore, more jumpers are needed between each pair of sensors. So more iterations are needed to reach consensus, and the computational cost is corresponding high. It is also worth to mention that when the connectivity probability is larger than 0.2, run time almost have no difference. It implies that our proposed algorithm is somewhat robust when the connectivity probability is larger than 0.2.

As Figure 18 shows, with the sensor number increases, the mean values of run time for one sensor increases linearly. Thus, our proposed method can handle a relatively large network in a reasonable period of time.

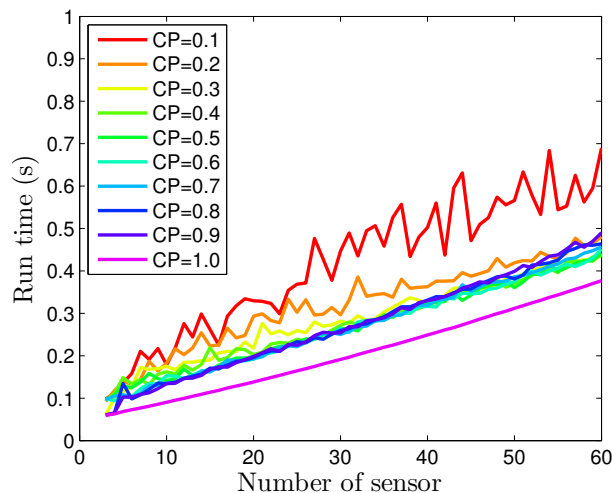


Figure 18. The mean values of the run times with different sensor numbers.

5. Conclusions

The joint estimation of system state and sensor bias based on a generalized system model is proposed. The conditions for UI decoupling derived in this paper are helpful for guiding sensor choice and deployment. Local UI estimates are obtained via the LS approach using all estimates from neighboring sensors. Network consensus processing is then applied to obtain more accurate estimates of the system (target) state and the UI. Future work may address distributive and collaborative processing in sensor networks in the case in which bias exists in only a subset of the sensors.

Acknowledgments: This research is supported by the National Science Foundation Council of China under Grant 61135001, 61374023, 61374159, 61403309 and Aerospace Support Technology Foundation of China under Grant 2014-HT-XGD.

Author Contributions: Jie Zhou and Yan Liang conceived of, designed and performed the simulations and wrote the manuscript. Feng Yang, Linfeng Xu and Quan Pan reviewed the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

UI	Unknown input
GSB	generalized sensor biases
EM	Expectation-maximization
IMM	Interacting multiple model
LMMSE	Linear minimum-mean-square-error
MUBF	Minimum upper bound filter
SB	Sensor bias
ECM	Electronic counter-measure
LSB	Local sensor bias

GLSB	Global sensor bias
ACF	Average consensus fusion
DLS	Distributed least squares
WSN	Wireless sensor networks
KF	Kalman Filter
DP no consensus	The proposed method without consensus processing
DP consensus	The proposed method with consensus processing
CP	Connectivity probability

Appendix A

Proof of Lemma 1. Sufficient: If $\text{rank}(\Gamma\Psi) = \text{rank}(\Psi)$ and Ψ is of full column rank, then the matrix $\Gamma\Psi$ is also of full column rank, and hence, its left pseudo-inverse matrix $(\Gamma\Psi)^+ = [(\Gamma\Psi)^T(\Gamma\Psi)]^{-1}(\Gamma\Psi)^T$ exists, satisfying

$$(\Gamma\Psi)^+ \Gamma\Psi = I \quad (\text{A1})$$

Clearly, Equation (6) holds.

Necessary: If Equation (6) holds, then $\Psi = \Psi(\Gamma\Psi)^+(\Gamma\Psi)$ or $\Psi^T = (\Gamma\Psi)^T[\Psi(\Gamma\Psi)^+]^T$, implying that Ψ^T belongs to the space of the matrix $(\Gamma\Psi)^T$; hence, we have the following inequality:

$$\text{rank}\{(\Psi)^T\} \leq \text{rank}\{(\Gamma\Psi)^T\}$$

or

$$\text{rank}\{\Psi\} \leq \text{rank}\{\Gamma\Psi\} \quad (\text{A2})$$

Meanwhile, we know that

$$\text{rank}\{\Gamma\Psi\} \leq \min(\text{rank}\{\Gamma\}, \text{rank}\{\Psi\}) \leq \text{rank}\{\Psi\} \quad (\text{A3})$$

Combining Equations (A2) and (A3), we obtain Equation (7). \square

Appendix B

Proof of Theorem 2. By left-multiplying both sides of Equation (8) by \bar{H}_{k+1}^i , we obtain

$$\bar{H}_{k+1}^i z_{k+1}^i = \bar{H}_{k+1}^i A_k^i z_k^i + \bar{H}_{k+1}^i \bar{G}_k^i d_k + \bar{H}_{k+1}^i \bar{\zeta}_k^i \quad (\text{B1})$$

Meanwhile, Equation (9) can be rewritten as

$$\bar{H}_{k+1}^i z_{k+1}^i = y_{k+1}^i - v_{k+1}^i \quad (\text{B2})$$

Then, substituting Equation (B1) into Equation (B2) yields

$$\bar{H}_{k+1}^i \bar{G}_k^i d_k = y_{k+1}^i - v_{k+1}^i - \bar{H}_{k+1}^i A_k^i z_k^i - \bar{H}_{k+1}^i \bar{\zeta}_k^i$$

with the following solution:

$$d_k = \Pi_k^i \left(y_{k+1}^i - v_{k+1}^i - \bar{H}_{k+1}^i A_k^i z_k^i - \bar{H}_{k+1}^i \bar{\zeta}_k^i \right) + \left[I_q - \Pi_k^i \left(\bar{H}_{k+1}^i \bar{G}_k^i \right) \right] \kappa \quad (\text{B3})$$

where $\Pi_k^i \doteq \left(\bar{H}_{k+1}^i \bar{G}_k^i \right)^+ = \left[\left(\bar{H}_{k+1}^i \bar{G}_k^i \right)^T \bar{H}_{k+1}^i \bar{G}_k^i \right]^{-1} \left(\bar{H}_{k+1}^i \bar{G}_k^i \right)^T$ in Equation (12) and Equation (B3) and κ is an arbitrary matrix with the proper dimensions.

Further substituting Equation (B3) into Equation (8) yields

$$\begin{aligned} z_{k+1}^i &= A_k^i z_k^i + \bar{\zeta}_k^i + \bar{C}_k^i \left(y_{k+1}^i - v_{k+1}^i - \bar{H}_{k+1}^i A_k^i z_k^i - \bar{H}_{k+1}^i \bar{\zeta}_k^i \right) + \bar{G}_k^i \left(I_q - \Pi_k^i \bar{H}_{k+1}^i \bar{G}_k^i \right) \kappa \\ &= \bar{F}_k^i A_k^i z_k^i + \bar{C}_k^i y_{k+1}^i + \bar{B}_k^i \bar{w}_k^i + \bar{G}_k^i \left[I_q - \Pi_k^i \bar{H}_{k+1}^i \bar{G}_k^i \right] \kappa \end{aligned} \quad (\text{B4})$$

where \bar{C}_k^i and \bar{B}_k^i are given in Equations (13) and (16), respectively. For any κ , the necessary and sufficient condition for guaranteeing the uniqueness of z_{k+1}^i in Equation (B4) is

$$\bar{G}_k^i \left(I_q - \Pi_k^i \bar{H}_{k+1}^i \bar{G}_k^i \right) = \bar{G}_k^i \left(I_q - \left(\bar{H}_{k+1}^i \bar{G}_k^i \right)^+ \bar{H}_{k+1}^i \bar{G}_k^i \right) = 0 \quad (\text{B5})$$

According to Lemma 1, Equation (B5) will hold if and only if Equation (10) holds. Finally, substituting Equation (B5) into Equation (B4) results in Equations (11)–(13). \square

Appendix C

Proof of Theorem 4. In Equation (27), $w_{ij}^x(l) \geq 0$ denotes the probability of being in state i at iteration l . The covariance of the mixture is

$$\begin{aligned} & E \left\{ \left[x_k - \hat{x}_{k|k}^i(l+1) \right] \left[x_k - \hat{x}_{k|k}^i(l+1) \right]^T \right\} \\ &= E \sum_{j \in \mathcal{J}_i} w_{ij}^x(l) \left\{ \left[x_k - \hat{x}_{k|k}^j(l) + \hat{x}_{k|k}^j(l) - \hat{x}_{k|k}^i(l+1) \right] \left[x_k - \hat{x}_{k|k}^j(l) + \hat{x}_{k|k}^j(l) - \hat{x}_{k|k}^i(l+1) \right]^T \right\} \\ &= E \sum_{j \in \mathcal{J}_i} w_{ij}^x(l) \left\{ \left[\left(x_k - \hat{x}_{k|k}^j(l) \right) \left(x_k - \hat{x}_{k|k}^j(l) \right)^T \right] + \left[\hat{x}_{k|k}^j(l) - \hat{x}_{k|k}^i(l+1) \right] \left[\hat{x}_{k|k}^j(l) - \hat{x}_{k|k}^i(l+1) \right]^T \right\} \\ &= \sum_{j \in \mathcal{J}_i} w_{ij}^x(l) P_{k|k}^{xx,i}(l) + \sum_{j \in \mathcal{J}_i} w_{ij}^x(l) \left[\hat{x}_{k|k}^j(l) - \hat{x}_{k|k}^i(l+1) \right] \left[\hat{x}_{k|k}^j(l) - \hat{x}_{k|k}^i(l+1) \right]^T \end{aligned} \quad (\text{C1})$$

The second term in Equation (C1) has the following alternative expression:

$$\begin{aligned} & \sum_{j \in \mathcal{J}_i} w_{ij}^x(l) \left[\hat{x}_{k|k}^j(l) - \hat{x}_{k|k}^i(l+1) \right] \left[\hat{x}_{k|k}^j(l) - \hat{x}_{k|k}^i(l+1) \right]^T \\ &= \sum_{j \in \mathcal{J}_i} w_{ij}^x(l) \hat{x}_{k|k}^j(l) \left(\hat{x}_{k|k}^j(l) \right)^T - \hat{x}_{k|k}^i(l+1) \left(\hat{x}_{k|k}^i(l+1) \right)^T \end{aligned} \quad (\text{C2})$$

Substituting Equation (C2) into Equation (C1) yields Equation (29), and Equation (30) can be obtained similarly. \square

References

1. Kurz, H.; Goedecke, W. Digital parameter-adaptive control of processes with unknown dead time. *Automatica* **1981**, *17*, 245–252.
2. Henson, M.A.; Seborg, D.E. Time delay compensation for nonlinear processes. *Ind. Eng. Chem. Res.* **1994**, *33*, 1493–1500.
3. Wang, X.X.; Liang, Y.; Pan, Q.; Wang, Y.G. Measurement random latency probability identification. *IEEE Trans. Autom. Control* **2016**, *PP*, DOI:10.1109/TAC.2015.2514259.
4. Wang, X.X.; Liang, Y.; Pan, Q.; Zhao, C.H.; Yang, F. Design and implementation of Gaussian filter for nonlinear system with randomly delayed measurements and correlated noises. *Appl. Math. Comput.* **2014**, *232*, 1011–1024.
5. Shi, Y.; Fang, H. Kalman filter-based identification for systems with randomly missing measurements in a network environment. *Int. J. Control* **2010**, *83*, 538–551.
6. Chilin, D.; Liu, J.; delaPeña, D.M.; Christofides, P.D.; Davis, J.F. Detection, isolation and handling of actuator faults in distributed model predictive control systems. *J. Process Control* **2010**, *20*, 1059–1075.
7. Okello, N.N.; Challa, S. Joint sensor registration and track-to-track fusion for distributed trackers. *Aerosp. Electron. Syst.* **2004**, *40*, 808–823.

8. Lin, X.; Bar-Shalom, Y.; Kirubarajan, T. Multisensor multitarget bias estimation for general asynchronous sensors. *Aerosp. Electron. Syst.* **2005**, *41*, 899–921.
9. Spingarn, K.; Robinson, B.H. Attitude determination with UD implementation of decoupled bias estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1294–1304.
10. Wang, G.B.; Chen, L.; Jia, S.Y. Optimized bias estimation model for 3-D radar considering platform attitude errors. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *27*, 19–24.
11. Sarma, A. Mathematical modeling of INS error dynamics for integration/debiasing. *Position Locat. Navig. Symp. PLANS* **2014**, 136–146, doi:10.1109/PLANS.2014.6851367.
12. Zhong, M.; Guo, J.; Cao, Q. On designing PMI Kalman filter for INS/GPS integrated systems with unknown sensor errors. *IEEE Sens. J.* **2015**, *15*, 535–544.
13. Pin, G.; Chen, B.; Parisini, T.; Bodson, M. Robust sinusoid identification with structured and unstructured measurement uncertainties. *IEEE Trans. Autom. Control* **2014**, *59*, 1588–1593.
14. Karniely, H.; Siegelmann, H.T. Sensor registration using neural networks. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 85–101.
15. Mehra, R.K. Approaches to adaptive filtering. *IEEE Trans. Autom. Control* **1972**, *17*, 693–698.
16. Myers, K.; Tapley, B.D. Adaptive sequential estimation with unknown noise statistics. *IEEE Trans. Autom. Control* **1976**, *21*, 520–523.
17. Liang, Y.; Zhou, D.H.; Pan, Q. A finite-horizon adaptive Kalman filter for linear systems with unknown disturbances. *Signal Process.* **2004**, *84*, 2175–2194.
18. Liang, Y.; Zhou, D.H.; Pan, Q. Estimation of varying time delay and parameters of a class of jump Markov nonlinear stochastic systems. *Comput. Chem. Eng.* **2003**, *27*, 1761–1778.
19. Durovic, Z.M.; Kovacevic, B.D. Robust estimation with unknown noise statistics. *IEEE Trans. Autom. Control* **1999**, *44*, 1292–1296.
20. Odelson, B.J.; Rajamani, M.R.; Rawlings, J.B. A new autocovariance least-squares method for estimating noise covariances. *Automatica* **2006**, *42*, 303–308.
21. Bavdekar, V.A.; Deshpande, A.P.; Patwardhan, S.C. Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter. *J. Process Control* **2011**, *21*, 585–601.
22. Bogler, P.L. Tracking a maneuvering target using input estimation. *IEEE Trans. Aerosp. Electron. Syst.* **1987**, *3*, 298–310.
23. Lee, H.; Tahk, M.J. Generalized input-estimation technique for tracking maneuvering targets. *IEEE Trans. Aerosp. Electron. Syst.* **1999**, *35*, 1388–1402.
24. Lan, H.; Liang, Y.; Yang, F.; Wang, Z.; Pan, Q. Joint estimation and identification for stochastic systems with unknown inputs. *IET Control Theory Appl.* **2013**, *7*, 1377–1386.
25. Lan, H.; Bishop, A.N.; Pan, Q. Distributed joint estimation and identification for sensor networks with unknown inputs. In Proceedings of the IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, Singapore, 21–24 April 2014.
26. Wang, X.X.; Song, B.; Pan, Q.; Geng, H. EM-based adaptive divided difference filter for nonlinear system with multiplicative parameter. *Int. J. Robust Nonlin.* (accepted).
27. Darouach, M.; Zasadzinski, M.; Boutayeb, M. Extension of minimum variance estimation for systems with unknown inputs. *Automatica* **2003**, *39*, 867–876.
28. Gillijns, S.; de Moor, B. Unbiased minimum-variance input and state estimation for linear discrete-time systems. *Automatica* **2007**, *43*, 111–116.
29. Gillijns, S.; de Moor, B. Unbiased minimum-variance input and state estimation for linear discrete-time systems with direct feedthrough. *Automatica* **2007**, *43*, 934–937.
30. El Ghaoui, L.; Calafiore, G. Robust filtering for discrete-time systems with bounded noise and parametric uncertainty. *IEEE Trans. Autom. Control* **2001**, *46*, 1084–1089.
31. Eldar, Y.C.; Ben-Tal, A.; Nemirovski, A. Linear minimax regret estimation of deterministic parameters with bounded data uncertainties. *IEEE Trans. Signal Process.* **2004**, *52*, 2177–2188.
32. Mazor, E.; Averbuch, A.; Bar-Shalom, Y.; Dayan, J. Interacting multiple model methods in target tracking: A survey. *IEEE Trans. Aerosp. Electron. Syst.* **1998**, *34*, 103–123.
33. Chen, B.; Tugnait, J.K. Interacting multiple model fixed-lag smoothing algorithm for Markovian switching systems. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 243–250.

34. Wang, Z.; Lam, J.; Liu, X. Robust filtering for discrete-time Markovian jump delay systems. *IEEE Signal Process. Lett.* **2004**, *11*, 659–662.
35. Johnston, L.; Krishnamurthy, V. An improvement to the interacting multiple model (IMM) algorithm. *IEEE Trans. Signal Process.* **2001**, *49*, 2909–2923.
36. Katsikas, S.K.; Likothanassis, S.D.; Beligiannis, G.N. Genetically determined variable structure multiple model estimation. *IEEE Trans. Signal Process.* **2001**, *49*, 2253–2261.
37. Xu, L.F.; Li, X.R.; Duan, Z. Hybrid grid multiple-model estimation with application to maneuvering target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 122–136.
38. Yang, Y.; Liang, Y.; Pan, Q. Linear minimum-mean-square error estimation of Markovian jump linear systems with Stochastic coefficient matrices. *IET Control Theory Appl.* **2014**, *8*, 1112–1126.
39. Liang, Y.; Zhou, D.; Zhang, L. Adaptive filtering for stochastic systems with generalized disturbance inputs. *IEEE Signal Process. Lett.* **2008**, *15*, 645–648.
40. Qin, Y.; Liang, Y.; Yang, Y. Adaptive filter of non-linear systems with generalised unknown disturbances. *IET Radar Sonar Navig.* **2014**, *8*, 307–317.
41. Zhou, L.; Liang, Y.; Zhou, J. Linear minimum mean squared estimation of measurement bias driven by structured unknown inputs. *IET Radar Sonar Navig.* **2014**, *8*, 977–986.
42. Liang, Y.; Chen, T.; Pan, Q. Multi-rate stochastic H_∞ filtering for networked multi-sensor fusion. *Automatica* **2010**, *46*, 437–444.
43. Simon, D. Kalman filter generalizations. In *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
44. Liang, Y.; Feng, X.; Yang, F. The distributed infectious disease model and its application to collaborative sensor wakeup of wireless sensor networks. *Inform. Sci.* **2013**, *223*, 192–204.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).