

Research Article

Diffusion Strategies for Distributed Kalman Filter with Dynamic Topologies in Virtualized Sensor Networks

Shujie Yang, Tao Huang, Jianfeng Guan, Yongping Xiong, and Mu Wang

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Shujie Yang; zongshi@bupt.edu.cn

Received 23 August 2016; Accepted 25 September 2016

Academic Editor: George Ghinea

Copyright © 2016 Shujie Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Network virtualization has become pervasive and is used in many applications. Through the combination of network virtualization and wireless sensor networks, it can greatly improve the multiple applications of traditional wireless sensor networks. However, because of the dynamic reconfiguration of topologies in the physical layer of virtualized sensor networks (VSNs), it requires a mechanism to guarantee the accuracy of estimate values by sensors. In this paper, we focus on the distributed Kalman filter algorithm with dynamic topologies to support this requirement. As one strategy of distributed Kalman filter algorithms, diffusion Kalman filter algorithm has a better performance on the state estimation. However, the existing diffusion Kalman filter algorithms all focus on the fixed topologies. Considering the dynamic topologies in the physical layer of VSNs mentioned above, we present a diffusion Kalman filter algorithm with dynamic topologies (DKFdt). Then, we emphatically derive the theoretical expressions of the mean and mean-square performance. From the expressions, the feasibility of the algorithm is verified. Finally, simulations confirm that the proposed algorithm achieves a greatly improved performance as compared with a noncooperative manner.

1. Introduction

With the rising of Big Data analysis and Smart City Conception, a variety of mobile applications such as Multipath Data Transfer [1, 2], target tracking [3–5], environmental monitoring [6, 7], Video-On-Demand Services [8–10], and distributed data storage have reshaped our daily life. In this context, wireless sensor networks (WSNs) as a key technology or such applications have drawn a significant attention from both academia and industry [11–16]. However, traditional WSNs consist of large number of heterogeneous sensors having capability of sensing, computation, and wireless communication, which make them fail to share of infrastructure and support the large scale sensing coverage. Fortunately, the rapid development of network virtualization opens a new opportunity for wireless sensor networks and creates a new sensing paradigm named virtualized sensor networks (VSNs) [17–19]. The salient feature of VSNs is enabling heterogeneous WSNs to coexist on a shared physical substrate. This allows VSNs to be more scalable and flexible, and, therefore, VSNs have become a promising solution for large scale sensing

task. Despite the unprecedented opportunities brought about by the VSNs, how to guarantee the sensing accuracy is a critical issue since mobile sensors in VSNs are highly dynamic and fully distributed without central control. To resolve this problem, we focus on the distributed Kalman filter algorithms, which is the combination of distributed algorithms and Kalman algorithms.

On the one hand, in distributed algorithms, a set of nodes can estimate the target state accurately through a cooperation manner. These nodes can be PCs, laptops, cell phones, sensors, and actuators [20]. Applications of distributed estimation algorithms are in several contexts, including wireless and sensor networks, where scalability, robustness, and low power consumption are desirable [21–23].

On the other hand, Kalman filter algorithms are one of the most popular methods for estimating the states of dynamic system from an incomplete and noisy measurement. As a recursion algorithm, Kalman filter has little requirements on calculation and memory space, which makes it more favorable in the real-time system application. Since it was proposed in 1960s, Kalman filter has been widely applied in

many fields, such as navigation, signal processing, control system, and information fusion [24].

As one strategy of distributed Kalman filter algorithms, diffusion Kalman filter algorithm has a better performance on the state estimation by diffusing information through a sequence of Kalman iterations and data-aggregation. In [25, 26], diffusion Kalman filtering has been introduced. According to them, a diffusion Kalman filter algorithm is comprised of the incremental update step and the diffusion update step. In the incremental update step, each node receives the observations from its neighbors and combines these observations to update its existing estimate to an intermediate value. In the diffusion update step, every node combines its neighbors intermediate estimates produced by the last step to update its own estimate. It is worth mentioning that all nodes perform these two steps similarly at the same time. The algorithm has an excellent performance in tracing a moving target as well as a good performance in convergence.

All of the literatures above assumed the links between nodes were ideal. However, in VSNs, the performance of the physical layer is strongly affected by the presence of such link state, where nodes and links may be subject to failure. These issues motivated the study of adaptive networks in dynamic scenarios. In [27], the author proposed a method to analyze the least mean-square (LMS) filter algorithm with changing topologies and normalized data. Similarly, a novel distributed affine projection algorithm (APA) with dynamic diffusion networks was presented in [28]. Although these algorithms give a good performance in dynamic topologies, they are all one kind of Wiener filtering. As we know, Wiener filtering required that the signal and noise must be a smooth process, which greatly restricts its application. And because of the limited shortage of these filters, they fail to obtain accurate measurement for moving targets. Considering these problems, we present a diffusion Kalman filter algorithm with dynamic topologies (DKFdt) to solve them. However, in order to analyze the rationality of the algorithm, we further need to consider the complexity and convergence of the algorithm under certain reasonable assumptions. Therefore, after proposing the new algorithm, our objective is to study whether the algorithm will have a good convergence and whether the performance of diffusion strategies in this scenario is still well. So in this article, we detailedly study the steady-state performance of the mean and the mean square of our algorithm based on link instabilities. All of these will be the basis of our future research into VSNs.

In comparison, we derive the models for the transient and steady-state behavior of the diffusion algorithm. From the results, we can obtain an interesting observation that although communication between nodes is limited to a small fraction, which is due to a probabilistic diffusion protocol, the mean-square performance does not present a significant decrease compared to a theory diffusion strategy.

The remainder of this paper is organized as follows. Section 2 lays the related work about our study and Section 3 describes the background of our research and lays the foundation of this article. The analysis of the DKFdt algorithm is presented in Section 4. In Section 5, we analyze the mean and mean-square performance of the estimate errors based

on the proposed algorithms and then simulation is given in Section 6. The conclusion and future work are described in Section 7.

We summarize the key symbols used in this paper in Notation.

2. Related Work

2.1. Framework of VSNs. Wireless sensor networks (WSNs) have been used in many application domains (e.g., temperature monitoring, security, and trajectory tracking) and become more and more popular in the last few years. However, as WSNs consist of large number of heterogeneous sensor nodes having capability of sensing, computation, communication and so forth, multiple applications sharing the very same WSNs infrastructure become a problem to be solved. Virtualization is a technology that can potentially enable this sharing.

Actually, network virtualization may bring nothing new in terms of technical capabilities and theoretical performance, but it provides a way of organizing networks such that it is possible to overcome some of the practical issues in traditional Internet. Now, in the field of virtualized sensor networks, research mainly focuses on the design and analysis of the VSNs framework [17, 18, 29–31]. In this part, we analyze previous work and summarize overall VSNs architecture shown in Figure 1. As can be seen from it, the VSNs consist of three layers: physical layer, virtualization layer, and user layer.

Physical Layer (PL). This layer consists of large number of heterogeneous sensor nodes for different purposes of sensing the environment like temperature, sound, and trajectory tracking. It offers their resources and observations through programmable interfaces to virtualization layer.

Virtualization Layer (VL). According to the needs of the upper user layers, VL integrates information provided by the physical layers to format a new virtual topology and provide variety of services for end users. It can also create child VLs by partitioning its resources and act as a virtual PL by leasing those child networks to other VLs (Figure 1).

User Layer (UL). The user layer is very similar to the application layer of the traditional network; users will request resource and information from the VL according to their own needs and meanwhile measure the QOS of the whole network.

From the analysis above, we can see that information or estimate value provided by PL affects the whole network performance. And because of the dynamic and distribute properties in VSNs, we should focus on how to get the accuracy estimation value in PL under the dynamic and distributed scenario firstly.

2.2. Distributed Algorithms. We review related research on distributed algorithms about distributed estimate problems. In distributed algorithms, nodes with processing and learning abilities are linked together to solve distributed optimization and estimation in real time through cooperating with each other.

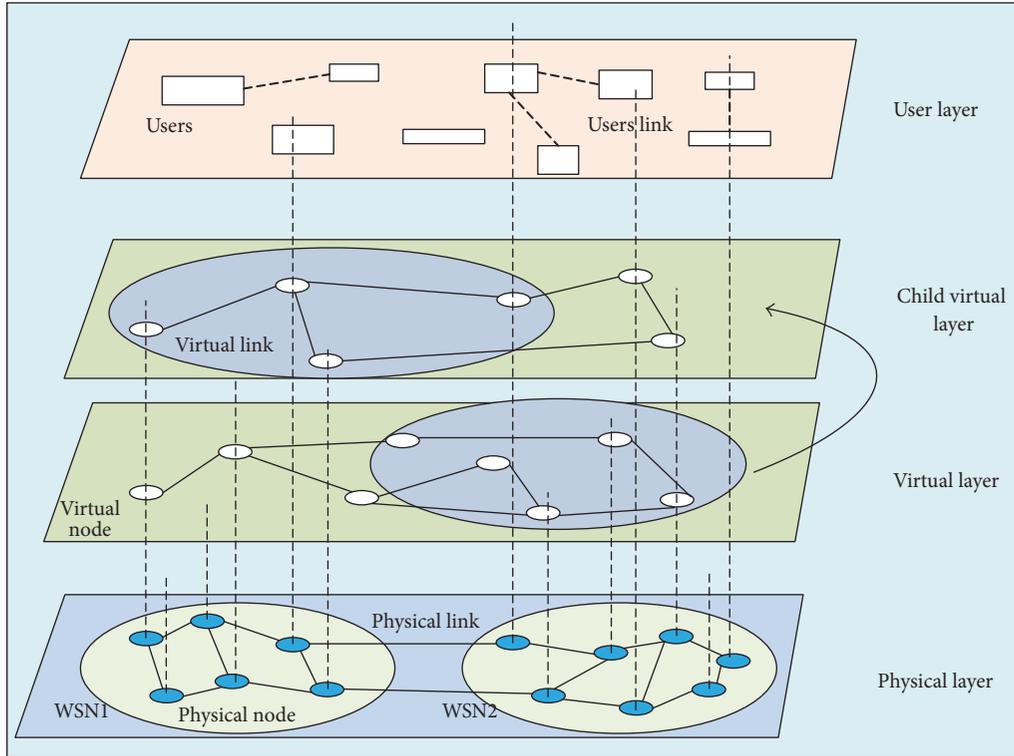


FIGURE 1: Framework of virtualized sensor network.

The existing distributed algorithms are mainly composed of consensus algorithms, gossip algorithms, and diffusion algorithms.

Average consensus and gossip algorithms have been studied extensively in recent years, especially in distributed optimization [32, 33] and distributed estimation problems [34–36]. The consensus strategy was originally proposed in the statistics literature [37] and has since then been developed into an elegant procedure to enforce agreement among cooperating nodes. Consensus Kalman filter algorithm has been studied in [38–40] where it consists of two steps. At first step, nodes collect the measurements across the network and, at the second step, it will iterate enough over the collected data to attain agreement. Unfortunately, consensus strategy hinders the ability to perform real-time recursive estimation and adaptation when measurement data keep streaming in, which is needed in VSNs.

Besides, [41, 42] propose distributed Kalman filters where the agents communicate among themselves using the gossip protocol [34]. At every sensing time step in a gossip Kalman filter, there is a communication round between a single pair of connected sensors, chosen according to the gossip protocol, when these two sensors exchange their current state estimate and their current error covariance matrix provided by the local Kalman filter Riccati equation. Although gossip filters require very low communication bandwidth, their mean square deviation (MSD) is higher and their convergence rate is lower.

Considering problems mentioned above, in this paper, we focus on diffusion strategies, where nodes communicate

with their neighbors only. There are plenty of researches on the diffusion strategies in the last more than ten years [25–28, 43–45]. The initial research is mainly focused on the application and convergence performance of some Wiener filtering algorithms in the fixed topology scenarios [43, 44, 46, 47]. In these articles, each node in the network has individual compute and sensor abilities. When nodes cooperate to estimate the performance of a target, each node will combine information from its neighbors firstly to update its own estimate to an intermediate value. After that, the node combines the intermediate value of its neighbors to obtain its update estimate again. Furthermore, in diffusion algorithms, the step-size parameters are not required to depend on the time index i and do not vanish as $i \rightarrow \infty$ compared with consensus algorithms, which implies networks with continuous adaptation and learning abilities. As the limitation of Wiener filter mentioned in Section 1, research for Kalman filter becomes increasing. In [25, 26, 45], diffusion mechanisms are used in Kalman filter, in which the problem of distributed Kalman filtering is studied. Nodes in these literatures are working directly in a linear dynamic system in a collaborative manner. The same with other diffusion algorithms, nodes in diffusion Kalman filtering algorithm communicate only with their neighbors, and the information is diffused across the network through a sequence of Kalman iterations and data-aggregations. Meanwhile, the mean and mean-square performance are analyzed to study the convergence of the diffusion Kalman filtering in the fix network topology.

In [27, 28], diffusion strategies are used in a changing topology which greatly expand the practicality in wireless

environment. Because of the instability of links, a dynamic topology model is given in these literatures, which provides a new way to study these problems. In this paper, considering dynamic topologies and distributed environment in physical layer of VSNs, we provide the diffusion Kalman filter algorithm in dynamic topologies and study the performance of its convergence through analyzing the value of mean and mean square. We will discuss this algorithm in detail in later sections.

3. System Model

In this section, we provide several fundamental models. First of all, the problem setup is described in Section 3.1. Then, in Section 3.2, the algorithm DKF (diffusion Kalman filter) given in [25] is briefly reviewed. Finally, to facilitate later analysis, the dynamic model is available in Section 3.3.

3.1. Problem Setup. In this part, we describe the dynamical system, local observations, and modeling assumptions, respectively, firstly and then give the well-known Kalman filter algorithm as the basis of the whole article.

Dynamical System. Consider tracking a moving object (assume its trajectory is an ellipse) in WSNs. We use x_i^b to define the state of this object with property b at time i , $b \in M$, of M properties (like position coordinates, velocity, and direction). To describe this problem well, we will introduce a discrete control process of the system. The system can use a linear stochastic difference equation by stack M variables in a position state vector $x_i = [x_i^1, \dots, x_i^M]^T$, defined as

$$x_{i+1} = F_i x_i + G_i u_i + w_i, \quad (1)$$

where $F_i \in \mathbb{C}^{M \times M}$ is a sparse localized model matrix, $u_i = [u_i^1, \dots, u_i^M] \in \mathbb{C}^M$ is the state noise vector, and $G_i \in \mathbb{C}^{M \times M}$ is the state noise matrix. The signal w_i is a deterministic input and is usually equal to zero.

Local Observations. We assume that the random field (1) is observed by N sensors, where each sensor observes only a few properties of the target. We use B_k to denote the number of properties observed by sensor k , $B_k \ll M$. Then, observations at sensor k can be represented by a linear model:

$$y_{k,i} = H_{k,i} x_i + v_{k,i}, \quad k = 1, \dots, N, \quad (2)$$

where $y_{k,i}$ denotes measurements by sensor k at time i , $H_{k,i} \in \mathbb{C}^{B_k \times M}$ is the local observation matrix, and $v_{k,i} \in \mathbb{C}^{B_k}$ is the local observation noise to reflect the inaccuracy in measurements due to sensor precision and other unavoidable constraints.

We can get the global observation model by stacking the observations as follows:

$$y_i = \begin{bmatrix} y_{1,i} \\ \vdots \\ y_{N,i} \end{bmatrix},$$

$$H_i = \begin{bmatrix} H_{1,i} \\ \vdots \\ H_{N,i} \end{bmatrix}, \quad v_i = \begin{bmatrix} v_{1,i} \\ \vdots \\ v_{N,i} \end{bmatrix}. \quad (3)$$

Then, the global observation matrix $y_i \in \mathbb{C}^{\sum_{k=1}^N B_k}$ is given by

$$y_i = H_i x_i + v_i. \quad (4)$$

Modeling assumptions are as follows.

Assumption 1. It is customary to assume that the noise signals u_i and v_i are temporally white and spatially independent with covariance matrices denoted as [28]

$$E \begin{bmatrix} u_i \\ v_i \end{bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix}^* = \begin{bmatrix} Q_i & 0 \\ 0 & R_i \end{bmatrix} \delta_{ij}, \quad (5)$$

where $*$ denotes conjugate transposition.

Assumption 2. We assume the initial state vector x_0 , with zero mean and covariance matrix $\Pi_0 > 0$, is also temporally and spatially independent with the state and measurement noises.

Kalman Filter Algorithm. Based on the analysis above, let $\hat{x}_{i|j}$ denote the linear minimum mean-square error estimate of x_i given observations up to and including time j . And let $P_{i|j}$ denote the covariance matrix of the estimation error $\tilde{x}_{i|j} \triangleq x_i - \hat{x}_{i|j}$. Therefore, the Kalman filter algorithm can start from $\hat{x}_{0|-1} = 0$ and $P_{0|-1} = \Pi_0$ as the following equations [48]:

(1) Measurement update:

$$\begin{aligned} K_i &= P_{i|i-1} H_i^* (R_i + H_i P_{i|i-1} H_i^*)^{-1}, \\ \hat{x}_{i|i} &= \hat{x}_{i|i-1} + K_i [y_i - H_i \hat{x}_{i|i-1}], \\ P_{i|i} &= P_{i|i-1} - K_i H_i P_{i|i-1}. \end{aligned} \quad (6)$$

(2) Time update:

$$\begin{aligned} \hat{x}_{i+1|i} &= F_i x_{i|i} + u_i, \\ P_{i+1|i} &= F_i P_{i|i} F_i^* + G_i Q_i G_i^* \end{aligned} \quad (7)$$

in which K_i is the kalman filtering gain.

3.2. Algorithm DKF

Neighborhood Communication Model. In diffusion strategies, sensors exchange their current estimates of the object state

with their neighbors only to reduce communications costs. Formally, we model the communication structure with a simple, undirected, connected graph $(\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of N sensors and \mathcal{E} is the set of allowable communication links among the sensors. The neighborhood of sensor k is defined by $\mathcal{N}_k \subseteq \mathcal{E}$; then, we can obtain the adjacency matrix Ω with entries $a_{k,l}$ as follows:

$$\Omega = \{a_{k,l}\} = \begin{cases} 1, & l \in \mathcal{N}_k \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

We assume that the diagonal elements of Ω are identically 1, indicating that a sensor k can always communicate with itself. At a particular instant, each sensor aggregates the observation information of its neighbors to have an accurate estimation to the target traction. However, due to the influence of the observation noise $v_{k,i}$, we provide a transfer coefficient for each neighbor to adjust the impact of noise. For definiteness, we assume the following transfer coefficient matrix, which is a left-random matrix and has a detailed description in literature [20]. Define the coefficient matrix $C = [c_{l,k}]$ with the properties:

$$c_{l,k} \geq 0, \quad \sum_{l=1}^N c_{l,k} = 1, \quad (9)$$

$$c_{l,k} = 0 \quad \text{if } l \notin \mathcal{N}_k \text{ for } k = 1, 2, \dots, N.$$

We call C the diffusion matrix, since it governs the diffusion process and plays an important role in the steady-state performance of the network. The entries in C represent the weights that are used by the diffusion algorithm to combine nearby estimates and have a variety of rules for its value. In this article, we will adopt Metropolis rule like [26] to express the set of nonnegative coefficients $c_{l,k}$. We use n_k to denote the degree of node k and we can get $c_{l,k}$ as follows:

$$c_{l,k} = \frac{1}{\max(n_l, n_k)}, \quad \text{if } l \neq k \text{ are linked} \\ c_{l,k} = 0, \quad \text{if } l, k \text{ not linked} \quad (10)$$

$$c_{k,k} = 1 - \sum_{l \in \mathcal{N}_k/k} c_{l,k} \quad \text{if } l = k.$$

The values of the weights $c_{l,k}$ may be chosen as shown in Figure 2.

The Diffusion Kalman Filter. In diffusion implementation, nodes cooperate with their direct neighbors only and diffuse the information across the entire network through a sequence of Kalman iterations and data-aggregation. In [25], diffusion Kalman filter algorithms have been proposed, which contain two steps in information exchange (local information exchange and global information exchange, resp.) and are summarized in Algorithm 1.

The algorithm has better adaptation and tracking abilities. However, in the wireless environment, the probability of link failure will increase significantly. We will discuss this situation detailedly in next section.

Consider the state-space model as in (1).
 Start with $\hat{x}_{k,0|-1} = Ex_0$
 and $P_{k,0|-1} = \Pi_0$ for all k . For every time i , repeat:
Step 1. Incremental Update:

$$P_{k,i|i}^{-1} = P_{k,i|i-1}^{-1} + \sum_{l \in \mathcal{N}_k} H_{l,i}^T R_{l,i}^{-1} H_{l,i}$$

$$\phi_{k,i|i} = \hat{x}_{k,i|i-1} + P_{k,i|i} \sum_{l \in \mathcal{N}_k} H_{l,i}^T R_{l,i}^{-1} (y_{l,i} - H_{l,i} \hat{x}_{k,i|i-1})$$
Step 2. Diffusion Update:

$$\hat{x}_{k,i|i} = \sum_{l \in \mathcal{N}_k} c_{l,k} \phi_{l,i}$$

$$\hat{x}_{k,i+1|i} = F_i \hat{x}_{k,i|i}$$

$$P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*$$

ALGORITHM 1: The DKF algorithm in [25].

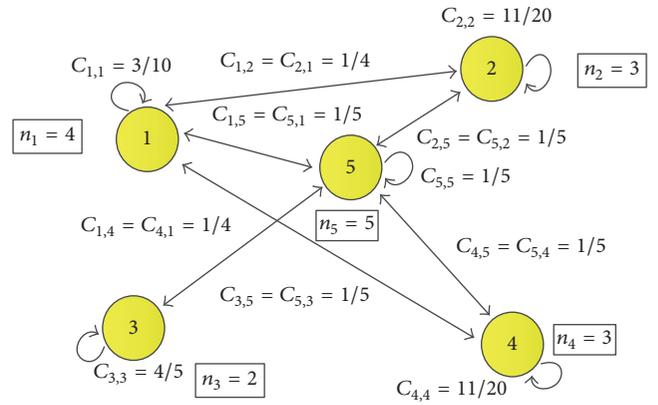


FIGURE 2: Values of the weights $c_{l,k}$ with Metropolis rule.

3.3. Dynamic Topology Scheme. We extend the diffusion Kalman filter to the dynamic topologies by assuming the links between any two points are dynamically linked. We assume that, at any time i , the link weight $c_{l,k}(i)$ (now randomly related to time i) will be either a nominal value $c_{l,k} = c_{k,l}$ with probability $p_{l,k} = p_{k,l}$ or zero with probability $q_{l,k}$:

$$c_{l,k}(i) = \begin{cases} c_{l,k} & \text{with } p_{lk} \\ 0 & \text{with } q_{lk} = 1 - p_{lk}. \end{cases} \quad (11)$$

For simplification, we assume an N -node network with a fixed topology. We adopt m_l to denote the maximum number of its links. Then, with different probabilities p_l , the network can be divided up to 2^{m_l} different subnetworks C_l . A simple example is shown in Figure 3, which describes a 3-node network with $m_l = 2$. For instance, the subnetwork C_2 happens with probability $p_2 = p_{12}q_{23}$.

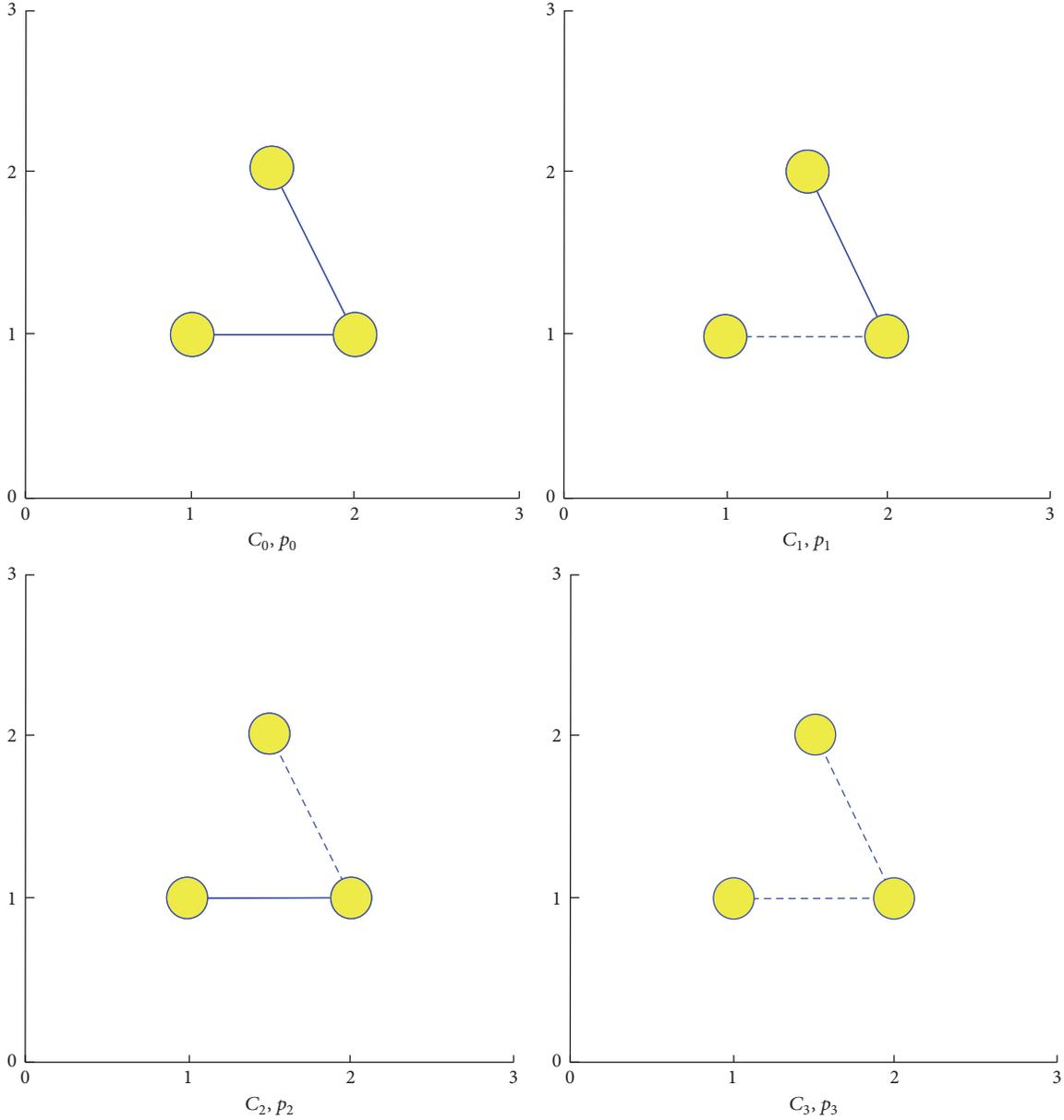


FIGURE 3: The subnetworks C_j with corresponding probability p_l .

In this manner, we can get the mean topology matrices $A = EA_i$ and $\mathcal{A} = E(A_i \odot A_i^{*T})$; namely,

$$\begin{aligned}
 A &= \sum_{l=1}^{2^{m_l}} p_l A_l, \\
 \mathcal{A} &= \sum_{l=1}^{2^{m_l}} p_l (A_l \odot A_l^{*T}),
 \end{aligned} \tag{12}$$

where \odot denotes the block Kronecker product. And $A_i = C_i \otimes I_M$, in which \otimes denotes Kronecker product, I_M denotes the identity matrix of size $M \times M$, and the definition of C_i is now related to time i with the form $C_i = [c_{i,k}(i)]$.

4. Analysis for DKFdt Algorithm

Based on the analysis above, we propose our diffusion Kalman filter with dynamic topologies (DKFdt) algorithm (see Algorithm 2). In our algorithm, we can see the definition of the neighbors of node k ; $\mathcal{N}_{k,i}$ is now varied with time i . At the same time, the diffusion weight matrix $c_{i,k}(t)$ is also related to time compared with Algorithm 1.

Then, we divide the DKFdt algorithm into three steps. To begin with, node k collects the measurements $y_{l,i}$, the measurement matrices $H_{l,i}$, and the noise covariance matrices $R_{l,i}$ from its neighbors l at time i to update its intermediate estimate $\psi_{k,i}$. And other nodes in the whole network will do the same incremental update operations like node k .

Consider the state-space model as in (1).
 Start with $\hat{x}_{k,0|i-1} = Ex_0$
 and $P_{k,0|i-1} = \Pi_0$ for all k . For every time i , repeat:
Step 1. Incremental Update:

$$K_{l,i} = P_{k,i|i-1} H_{l,i}^* (R_{l,i} + H_{l,i} P_{k,i|i-1} H_{l,i}^*)^{-1}$$

$$\tilde{\psi}_{k,i} = \hat{x}_{k,i|i-1} + \sum_{l \in \mathcal{N}_{k,i}} K_{l,i} [y_{l,i} - H_{l,i} \hat{x}_{k,i|i-1}]$$

$$P_{k,i|i} = P_{k,i|i-1} - \sum_{l \in \mathcal{N}_{k,i}} K_{l,i} H_{l,i} P_{k,i|i-1}$$
Step 2. Diffusion update with dynamic topologies:

$$\tilde{x}_{k,i|i} = \sum_{l \in \mathcal{N}_{k,i}} c_{l,k}(i) \tilde{\psi}_{l,i}$$
Step 3. Prediction update:

$$\hat{x}_{k,i+1|i} = F_i \tilde{x}_{k,i|i}$$

$$P_{k,i+1|i} = F_i P_{k,i|i} F_i^* + G_i Q_i G_i^*$$

ALGORITHM 2: Diffusion Kalman filter with dynamic topologies (DKFdt).

Secondly, after all nodes complete the first step, they step into the diffusion update stage. In this stage, nodes combine the intermediate estimates produced in the first step through a coefficient matrix C_i .

The last step is the prediction update stage. The main objective of this step is to predict next state estimate values by using present values; for example, estimate value $\hat{x}_{k,i+1|i}$ at time $i+1$ can be predicted by that at time i using the state model in (1); covariance matrix of the estimation error $P_{k,i+1|i}$ can be predicted by $P_{k,i|i}$ as well.

In the DKFdt algorithm, we combine the DKF algorithm and the dynamic topologies model together. Through this process, the network will be more close to the actual situation, where links between nodes are always instable. To prove the effectiveness of the algorithm, we need to ensure that the algorithm is convergent. So we will analyze its performance in mean and mean square to obtain its convergence behavior. These will be detailedly stated in the following sections.

5. Performance Analysis for Convergence

We now move on to examining the behavior of the DKFdt algorithm. Due to the random nature of the perturbations, it becomes necessary to evaluate the behavior of the algorithms on average. Therefore, we will study the convergence of the weight estimates in both the mean and mean-square performances.

At first, let $\tilde{\psi}_{k,i} = x_i - \hat{\psi}_{k,i}$ denote estimation error at the end of the incremental update. Let $\tilde{x}_{k,i|i-1} = x_i - \hat{x}_{k,i|i-1}$ denote estimation error at node k at the end of the diffusion update, where x_i is the real objective value. Then, we can obtain the following expressions:

$$\begin{aligned} \tilde{\psi}_{k,i} &= x_i - \hat{x}_{k,i|i-1} - \sum_{l \in \mathcal{N}_{k,i}} K_{l,i} [y_{l,i} - H_{l,i} \hat{x}_{k,i|i-1}] \\ &= \tilde{x}_{k,i|i-1} - \sum_{l \in \mathcal{N}_{k,i}} K_{l,i} (H_{l,i} \tilde{x}_{k,i|i-1} + v_{l,i}) \end{aligned}$$

$$= \left(I - \sum_{l \in \mathcal{N}_{k,i}} K_{l,i} H_{l,i} \right) \tilde{x}_{k,i|i-1} - \sum_{l \in \mathcal{N}_{k,i}} K_{l,i} v_{l,i}, \quad (13)$$

where the second equation follows (2):

$$\tilde{x}_{k,i|i-1} = F_{i-1} \tilde{x}_{k,i-1|i-1} + G_{i-1} n_{i-1}. \quad (14)$$

Combining (13) into the diffusion update step of Algorithm 2, we obtain

$$\begin{aligned} \tilde{x}_{k,i|i} &= \sum_{l \in \mathcal{N}_{k,i}} c_{l,k} \tilde{\psi}_{l,i} \\ &= \sum_{l \in \mathcal{N}_{k,i}} c_{l,k} \left[\left(I - \sum_{m \in \mathcal{N}_{l,i}} K_{m,i} H_{m,i} \right) \tilde{x}_{l,i|i-1} \right. \\ &\quad \left. - \sum_{m \in \mathcal{N}_{l,i}} K_{m,i} v_{m,i} \right]. \end{aligned} \quad (15)$$

5.1. Mean Performance. Assuming that the noise signals n_i and v_i are temporally white and spatially independent with each other, we obtain the following formulation when we take expectations of both sides of (14) and (15):

$$E \tilde{x}_{k,i|i-1} = F_{i-1} E \tilde{x}_{k,i-1|i-1} \quad (16)$$

$$E \tilde{x}_{k,i|i} = E \left[\sum_{m \in \mathcal{N}_{l,i}} c_{l,k} (I - S_{l,i}) \tilde{x}_{l,i|i-1} \right]. \quad (17)$$

Here, $S_{l,i} = \sum_{m \in \mathcal{N}_{l,i}} K_{m,i} H_{m,i}$.

Since $E \tilde{x}_{k,0|i-1} = 0$ and $E \tilde{x}_{k,-1|i-1} = 0$ as defined in Section 2, (17) indicates the mean performance of the DKFdt algorithm is unbiased.

5.2. Mean-Square Stability. To analyze mean-square stability, we will introduce filter mean square deviation (MSD) as the steady-state measure for node k [20]:

$$\text{MSD}_{k,i} = E \|x_i - \hat{x}_{k,i|i}\|^2. \quad (18)$$

Then, we give the definition of the global vectors as follows:

$$\begin{aligned} \tilde{\mathcal{X}}_{i|i} &\triangleq \text{col} \{ \tilde{x}_{1,i|i}, \dots, \tilde{x}_{N,i|i} \}, \\ \mathcal{K}_i &\triangleq \text{diag} \{ K_{1,i}, \dots, K_{N,i} \}, \\ \mathcal{H}_i &\triangleq \text{diag} \{ H_{1,i}, \dots, H_{N,i} \}, \\ \mathcal{S}_i &\triangleq \text{diag} \{ S_{1,i}, \dots, S_{N,i} \}. \end{aligned} \quad (19)$$

Consider the matrices $\mathcal{A}_i = C_i \otimes I_M$. We may now express (14) and (15) in a global form:

$$\tilde{\mathcal{X}}_{i|i} = A_i^T \left[\mathcal{F}_i^C \tilde{\mathcal{X}}_{i-1|i-1} + \mathcal{G}_i^C (\mathbf{1} \otimes n_{i-1}) - \mathcal{D}_i^C v_i \right]. \quad (20)$$

Here,

$$\begin{aligned} \mathcal{F}_i^C &\triangleq (I - \mathcal{S}_i) (I \otimes \mathcal{F}_{i-1}), \\ \mathcal{G}_i^C &\triangleq (I - \mathcal{S}_i) (I \otimes G_{i-1}), \\ \mathcal{D}_i^C &\triangleq \mathcal{K}_i R_i^{-1}, \end{aligned} \quad (21)$$

where T denotes the transpose of matrix A_i , $R_i = E v_i v_i^*$ is a block-diagonal matrix with v_i defined in (4), and $\mathbf{1}$ is the vector of size $N \times 1$ and whose entries are all equal to one.

Let $\mathcal{P}_{\bar{x},i} = E \tilde{\mathcal{X}}_{i|i} \tilde{\mathcal{X}}_{i|i}^*$ denote the covariance matrix of $\tilde{\mathcal{X}}_{i|i}$. Then, the recursion can be obtained by (21) as follows:

$$\begin{aligned} \mathcal{P}_{\bar{x},i} &= EA_i^T \left[\mathcal{F}_i^C \mathcal{P}_{\bar{x},i-1} \mathcal{F}_i^{C*} \right. \\ &\quad \left. + \mathcal{G}_i^C (\mathbf{1}\mathbf{1}^T \otimes Q_{i-1}) \mathcal{G}_i^{C*} + E (\mathcal{D}_i^C R_i \mathcal{D}_i^{C*}) \right] EA_i. \end{aligned} \quad (22)$$

In order to analyze the mean-square stability, we also adopt the same assumption in [26] as follows.

Assumption 3. The matrices in model (1) are time-invariant; that is, the matrices F, G, H, R , and Q do not depend on i .

Assumption 4. A Kalman filter that uses data from a neighborhood converges for every neighborhood (see [49] for conditions on Kalman filter convergence).

Based on the assumptions, the matrix (21) can be denoted as the steady-state value as follows:

$$\begin{aligned} \mathcal{F}^C &\triangleq \lim_{i \rightarrow \infty} \mathcal{F}_i^C = (I - \mathcal{S})(I \otimes F), \\ \mathcal{G}^C &\triangleq \lim_{i \rightarrow \infty} \mathcal{G}_i^C = (I - \mathcal{S})(I \otimes G), \\ \mathcal{D}^C &\triangleq \lim_{i \rightarrow \infty} \mathcal{D}_i^C = \mathcal{K}R^{-1}. \end{aligned} \quad (23)$$

Then, (22) can be denoted as follows:

$$\begin{aligned} \mathcal{P}_{\bar{x}} &= EA_i^T \left[\mathcal{F}^C \mathcal{P}_{\bar{x}} \mathcal{F}^{C*} + \mathcal{G}^C (\mathbf{1}\mathbf{1}^T \otimes Q) \mathcal{G}^{C*} \right. \\ &\quad \left. + E (\mathcal{D}^C R \mathcal{D}^{C*}) \right] EA_i. \end{aligned} \quad (24)$$

From (12) and the temporally and spatially independent assumptions on the state and measurement noises, (24) converges to the unique solution of the Lyapunov equation [50]:

$$\mathcal{P}_{\bar{x},i} = \mathcal{F} \mathcal{P}_{\bar{x},i-1} \mathcal{F}^* + \mathcal{G} (\mathbf{1}\mathbf{1}^T \otimes Q) \mathcal{G}^* + \mathcal{D} R \mathcal{D}^*. \quad (25)$$

Here,

$$\begin{aligned} \mathcal{F} &\triangleq A^T F_i^C = A^T (I - \mathcal{S}_i) (I \otimes F_{i-1}), \\ \mathcal{G} &\triangleq A^T G_i^C = A^T (I - \mathcal{S}_i) (I \otimes G_{i-1}), \\ \mathcal{D} &\triangleq A^T E D_i^C = A^T \mathcal{K}_i R_i^{-1}. \end{aligned} \quad (26)$$

Now, based on the DKFdt algorithm, we can get the same solution from the analysis of the mean-square stability as the existing DKF algorithms. Then, we also express the solution by using the operator, which is obtained by stacking the columns of a matrix. Then, (22) can be written as

$$\begin{aligned} \text{vec}(\mathcal{P}_{\bar{x}}) &= (I - \mathcal{F}^{*T} \otimes \mathcal{F})^{-1} \\ &\quad \cdot \text{vec} \left[\mathcal{G} (\mathbf{1}\mathbf{1}^T \otimes Q) \mathcal{G}^* + \mathcal{D} R \mathcal{D}^* \right]. \end{aligned} \quad (27)$$

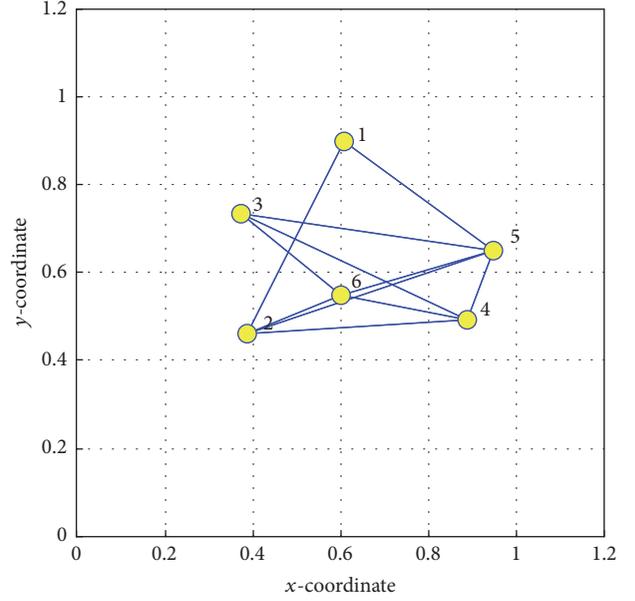


FIGURE 4: Network topology with $N = 6$.

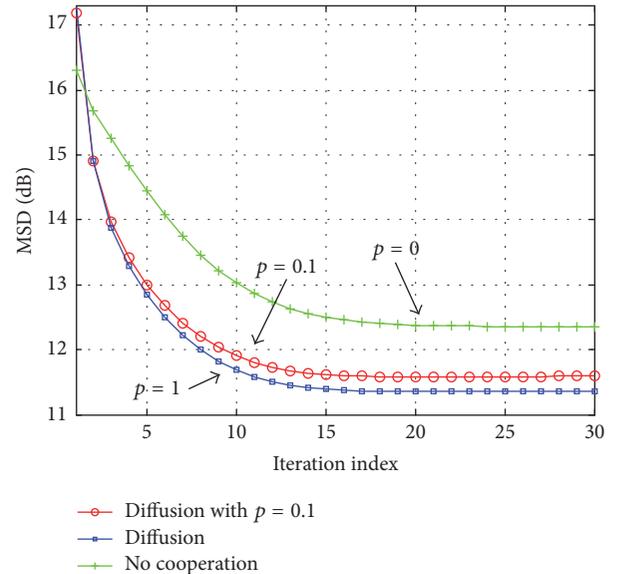


FIGURE 5: Global MSD performance for $p = 0$ (no cooperation), $p = 0.1$ (DKFdt), and $p = 1$ (standard diffusion) with $N = 6$.

Here, we use the property that

$$\text{vec}(P\Sigma Q) = (Q^T \otimes P) \text{vec}(\Sigma). \quad (28)$$

According to (27) and using the following equalities for arbitrary matrices $\{U, W, \sigma\}$ of compatible dimensions,

$$\text{Tr}(\Sigma W) = [\text{vec}(W^T)]^T \sigma, \quad (29)$$

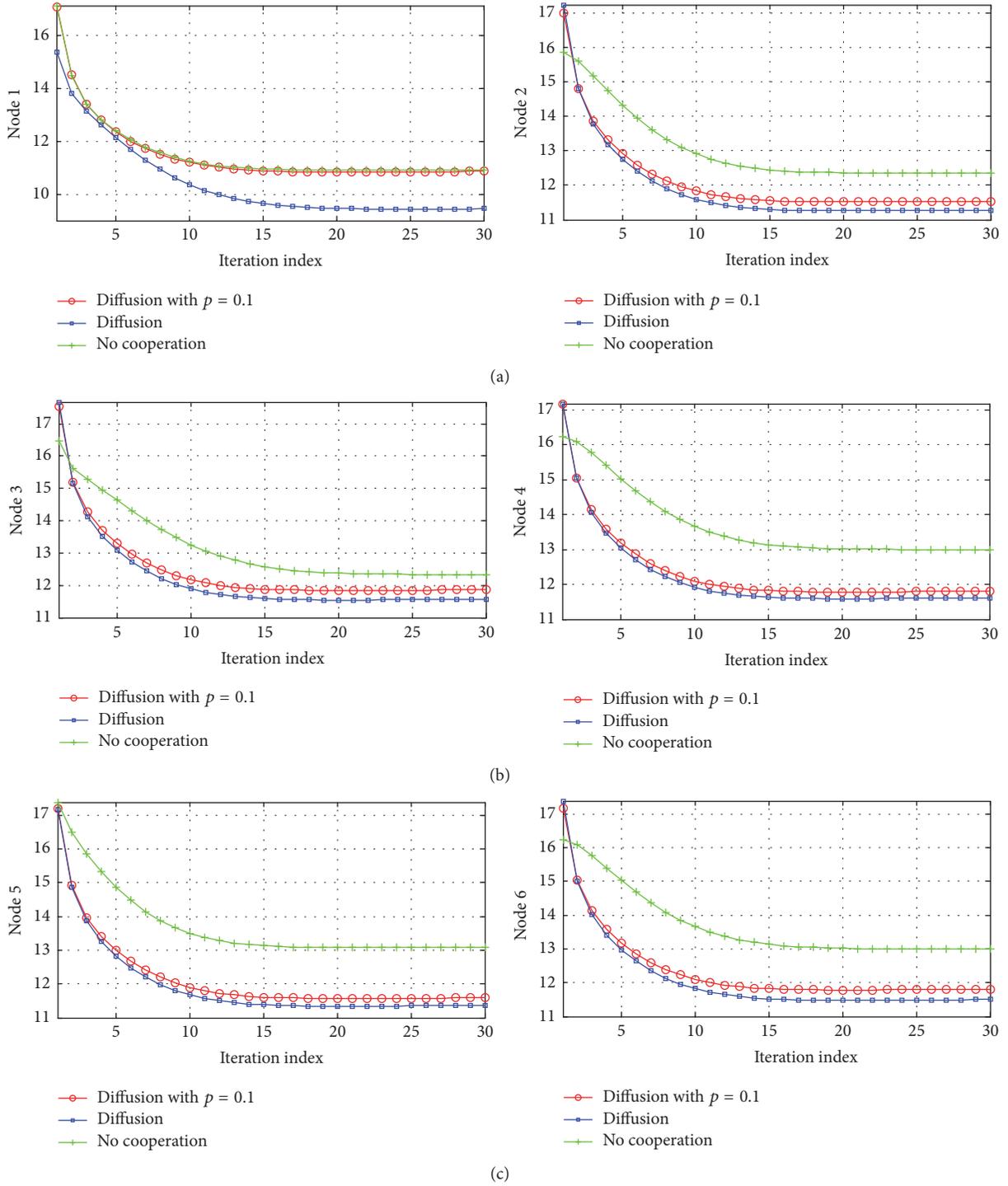


FIGURE 6: MSD per node (local).

the steady-state MSD at node k and the average steady-state MSD across the network may now be expressed as

$$\begin{aligned} \text{MSD}_k &= \lim_{i \rightarrow \infty} E \|x_i - \hat{x}_{k,i|i}\|^2 = \text{Tr}(\mathcal{P}_{\hat{x}} \mathcal{J}_k), \\ \text{MSD}^{\text{ave}} &= \frac{1}{N} \text{Tr}(\mathcal{P}_{\hat{x}}), \end{aligned} \quad (30)$$

where $\mathcal{J}_k = B_{N \times N} \otimes I_M$ and $B_{N \times N}$ is an $N \times N$ dimension matrix with all unit entries and I_M is the M dimension unit matrix.

Then, we can draw a conclusion that the DKFdt algorithm is unbiased and stability based on analysis of the mean and mean-square performance.

6. Number Evaluation

In this section, we give the simulation to illustrate the performance of the proposed algorithm. Consider the case that a set of sensors in WSNs attempts to track the trajectory of an ellipse object. We numerically evaluate the MSD performance of the DKFdt and compare it against the noncooperate Kalman filter (KF) and the diffusion Kalman filter (DKF) in [25]. To achieve this objective, we build a time-varying random system.

For simplicity, at first we use a small-scale network model to verify our algorithm. This network consists of six sensors with the topology shown in Figure 4, in which the connection lines between nodes only represent the information communication with two agents rather than the real link. At the same time, each sensor node can independently obtain the target status and communicate with its neighbors.

In our simulation, the matrices in (1) are

$$F = \begin{bmatrix} 0.992 & -0.1247 \\ 0.1247 & 0.992 \end{bmatrix},$$

$$Q = I_2,$$

$$G = \begin{bmatrix} 0.625 & 0 \\ 0 & 0.625 \end{bmatrix},$$

$$R_{k,i} = 10\sqrt{k}, \quad k = 1, \dots, N$$
(31)

and the initial conditions are

$$x_0 = (15, -10)^T,$$

$$P_0 = 10I_2.$$
(32)

Because the nodes take measures of the unknown position of the object, the measurement matrix $H_{k,i}$ is chosen to be either $H_{k,i} = H_x = (1, 0)$ or $H_{k,i} = H_y = (0, 1)$.

In Figure 5, we present the global MSD evolution for $p = 1$ (Algorithm 1), $p = 0.1$ (DKFdt, Algorithm 2), and $p = 0$ (without cooperation), where variable p represent the probability of the connection between any two points defined as $p_{k,l}$ or $p_{l,k}$ in Section 3.3. In the figure, the x -coordinate represents the iteration index and the y -coordinate represents the MSD value. From it, we can see that, with the passage of time, all the algorithm can converge to a limited range with different convergence speed and MSD values. Through comparison and analyzing the simulation among three different algorithms, we can conclude that when $p = 0.1$, which indicates the low utilization of the links, there is still a significant improvement over the situation without cooperation. These results confirm advantages of diffusion strategies for adaptive networks. Moreover, in order to embody the characteristics of our algorithm, we illustrate the performance of MSD of each node. As shown in Figure 6, although performance of convergence in each node is not different because of the effect of noise, they all can reach convergence through communication with each other.

In order to further illustrate the characteristics of our algorithm, we simulate another example in a larger network

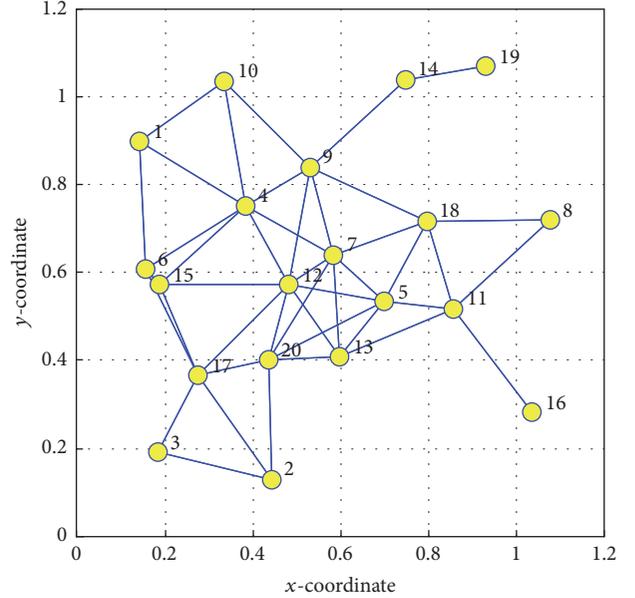


FIGURE 7: Network topology with $N = 20$.

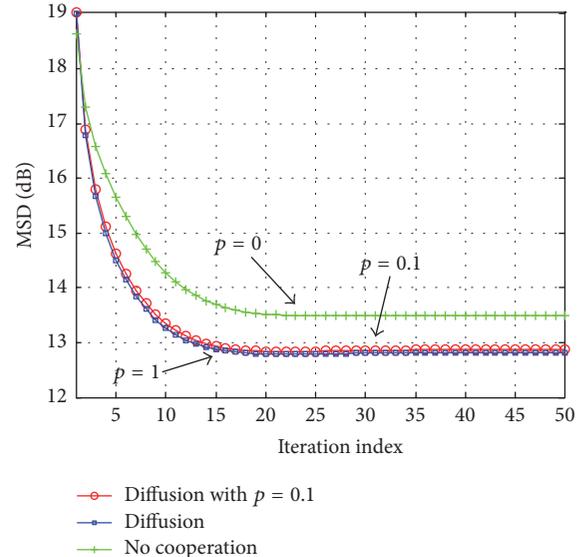


FIGURE 8: Global MSD performance for $p = 0$ (no cooperation), $p = 0.1$ (DKFdt), and $p = 1$ (standard diffusion) with $N = 20$.

topology with $N = 20$ in Figure 7 and other parameters remain the same with that of six-point topology network. In Figure 8, we take the same simulation to illustrate the performance of MSD with different probability of link connections. We have observed that values of diffusion algorithm with $p = 0.1$ are closer with diffusion algorithm in reliable link (with $p = 1$) than that in six-point network, which reflect that collaboration among more sensors can reduce the effect on instability of links.

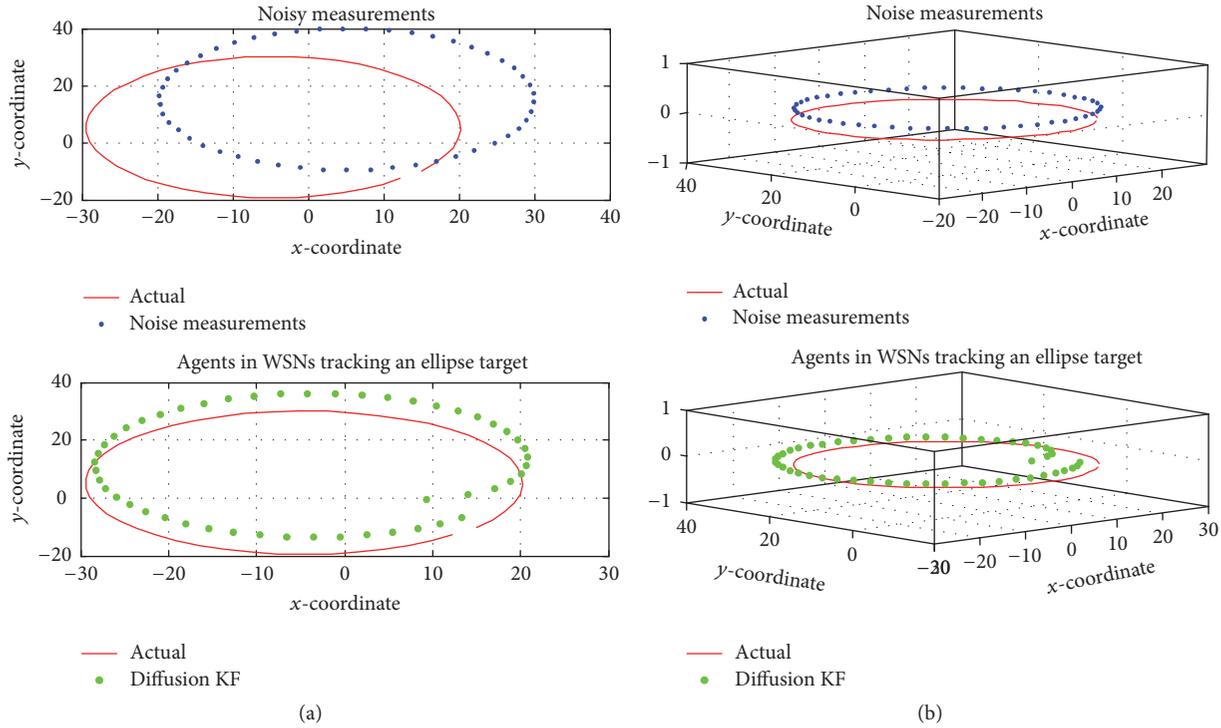


FIGURE 9: (a) Noisy measurement and tracking an ellipse target. (b) Noisy measurement and tracking the position of a moving object in 3D space in WSNs.

At last, we demonstrate the result of the nodes cooperating together to track the position of an ellipse object in Figure 9. And we show the result through 2D and 3D model, respectively. As we can see, the networks still have a good performance in estimating the position of the target through nodes diffusing information with each other continuously, although nodes are affected greatly by noises.

7. Conclusion and Further Work

In this paper, we provide a diffusion Kalman filter algorithm with dynamic topologies (DKFdt) to achieve this requirement in physical layer of VSNs. We also analyze the mean and the mean-square performance in dynamic scenarios. The result shows that the global estimation error can converge to a unique solution well. Through the simulation, our proposed diffusion strategy with cooperative method performs well in dynamic networks. It achieves an improved performance over the noncooperative strategy ($p = 0$) even at a high dynamic environment with low link probability ($p = 0.1$).

Although the algorithm is provided under the background of VSNs, its applications are not confined to this field. Based on what is mentioned above, we will further study other problems on the distributed Kalman filter algorithms and, at the same time, their applications in other fields are also the emphasis of our studies.

Notation

- \mathbb{R} : The real domain
- \mathbb{C} : The complex domain.

Key Symbols

- x_i : State vector of object at time i
- $y_{k,i}$: Observation value at sensor k at time i
- F_i : Dynamical system matrix
- $H_{k,i}$: Observation matrix at sensor k
- Q_i : System noisy covariance
- R_i : Observation noise covariance
- K_i : Kalman filtering gain
- $P_{i|j}$: The covariance matrix of the estimation error
- ω : The adjacency matrix
- C : The diffusion left-random matrix
- C_i : The diffusion left-random matrix varied with time i
- A : The mean topology matrix.

Competing Interests

The authors declare that they have no competing interests.

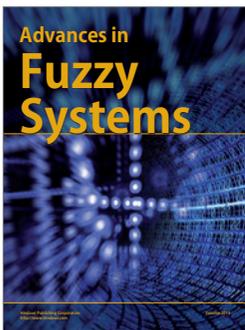
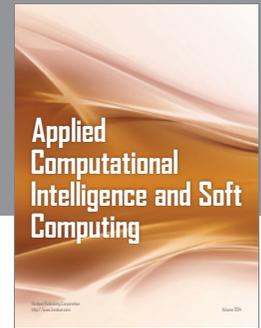
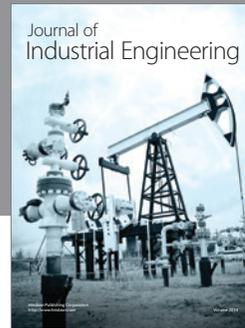
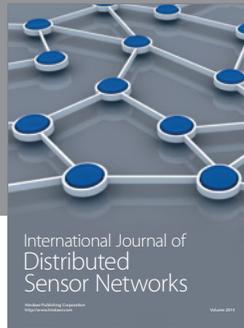
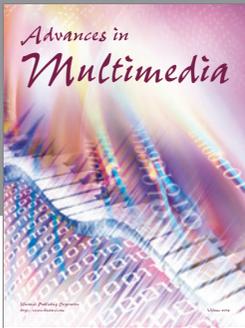
Acknowledgments

This work is supported by the National Science and Technology Major Project (Grant no. 2015ZX03003002-002), in part by the National Natural Science Foundation of China (NSFC) under Grant nos. 61522103, 61372112, and 61232017, in part by the National Key Basic Research Program of China (973 Program) under Grant 2013CB329102, and in part by the Beijing Natural Science Foundation under Grant 4142037.

References

- [1] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2193–2205, 2013.
- [2] C. Xu, Z. Li, J. Li, H. Zhang, and G.-M. Muntean, "Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 7, pp. 1175–1189, 2015.
- [3] S. Barath Kumar and G. Myilsamy, "Multi-target tracking in mobility sensor networks using Ant Colony Optimization," in *Proceedings of the International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN '13)*, pp. 350–354, Tirunelveli, India, March 2013.
- [4] M. Z. Bhuiyan, G. Wang, and A. V. Vasilakos, "Local area prediction-based mobile target tracking in wireless sensor networks," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1968–1982, 2015.
- [5] E. T. Yazdi, A. Moravejsharieh, and S. K. Ray, "Study of target tracking and handover in mobile wireless sensor network," in *Proceedings of the 28th International Conference on Information Networking (ICOIN '14)*, pp. 120–125, Phuket, Thailand, February 2014.
- [6] C. Türkay, C. G. Aksakalli, K. Özmal, and H. K. Ekenel, "A mobile plant identification application for environmental monitoring," in *Proceedings of the 21st Signal Processing and Communications Applications Conference (SIU '13)*, pp. 1–4, April 2013.
- [7] M. DArienzo, M. Iacono, S. Marrone, and R. Nardone, "Estimation of the energy consumption of mobile sensors in WSN environmental monitoring applications," in *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA '13)*, pp. 1588–1593, IEEE, Barcelona, Spain, March 2013.
- [8] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "QoE-driven user-centric vod services in urban multihomed P2P-based vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 2273–2289, 2013.
- [9] C. Xu, S. Jia, L. Zhong, H. Zhang, and G.-M. Muntean, "Ant-inspired mini-community-based solution for video-on-demand services in wireless mobile networks," *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 322–335, 2014.
- [10] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang, and G.-M. Muntean, "Performance-aware mobile community-based VoD streaming over vehicular Ad Hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 3, pp. 1201–1217, 2015.
- [11] D. P. Mishra, G. M. Asutkar, and S. S. Dorale, "An application of wireless sensor network in intelligent transportation system," in *Proceedings of the 6th International Conference on Emerging Trends in Engineering and Technology (ICETET '13)*, pp. 90–91, Maharashtra, India, December 2013.
- [12] Z. Rasin, H. Hamzah, and M. S. M. Aras, "Application and evaluation of high power Zigbee based wireless sensor network in water irrigation control monitoring system," in *Proceedings of the IEEE Symposium on Industrial Electronics & Applications (ISIEA '09)*, vol. 2, pp. 548–551, Kuala Lumpur, Malaysia, October 2009.
- [13] G. Wu, C. Wu, and S. Xu, "Application of wireless sensor network in the monitoring system of boiler," in *Proceedings of the 26th Chinese Control and Decision Conference (CCDC '14)*, pp. 1751–1754, IEEE, Changsha, China, May 2014.
- [14] W. Jing and L. Tingting, "Application of wireless sensor network in Yangtze River basin water environment monitoring," in *Proceedings of the 27th Chinese Control and Decision Conference (CCDC '15)*, pp. 5981–5985, IEEE, Qingdao, China, May 2015.
- [15] S. Rajba, P. Raif, T. Rajba, and M. Mahmud, "Wireless sensor networks in application to patients health monitoring," in *Proceedings of the IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE '13)*, pp. 94–98, Singapore, April 2013.
- [16] N. M. Hassan, O. M. Olaniyi, A. Ahmed, and E. M. Dogo, "Wireless sensor networks for remote healthcare monitoring in Nigeria: challenges and way forward," in *Proceedings of the 2nd IEEE International Conference on Emerging and Sustainable Technologies for Power and ICT in a Developing Society (IEEE NIGERCON '13)*, pp. 182–187, Owerri, Nigeria, November 2013.
- [17] C. Liang and F. R. Yu, "Wireless network virtualization: a survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 358–380, 2015.
- [18] I. Khan, T. SudParis, F. Belqasmi, and R. Glitho, "Wireless sensor network virtualization: a survey," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 553–576, 2015.
- [19] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: early architecture and research perspectives," *IEEE Network*, vol. 29, no. 3, pp. 104–112, 2015.
- [20] M. S. E. Abadi and Z. Saffari, "Distributed estimation over an adaptive diffusion network based on the family of affine projection algorithms," in *Proceedings of the 2012 6th International Symposium on Telecommunications (IST '12)*, pp. 607–611, Tehran, Iran, November 2012.
- [21] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.
- [22] Y. J. Chun, M. O. Hasna, and A. Ghayeb, "Adaptive network coding for spectrum sharing systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 2, pp. 639–654, 2015.
- [23] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.
- [24] Y. Huo, Z. Cai, W. Gong, and Q. Liu, "A new adaptive Kalman filter by combining evolutionary algorithm and fuzzy inference system," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '14)*, pp. 2893–2900, July 2014.
- [25] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010.
- [26] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering: formulation and performance analysis," in *Proceedings of the Cognitive Information Processing*, Santorini, Greece, June 2008.
- [27] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 3285–3288, IEEE, Las Vegas, Nev, USA, March-April 2008.
- [28] L. Li and J. A. Chambers, "Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology," in *Proceedings of the IEEE/SP 15th Workshop on Statistical Signal Processing (SSP '09)*, pp. 757–760, Cardiff, UK, September 2009.
- [29] I. Khan, F. Belqasmi, R. Glitho, and N. Crespi, "A multi-layer architecture for wireless sensor network virtualization," in

- Proceedings of the 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC '13)*, pp. 1–4, IEEE, Dubai, UAE, April 2013.
- [30] C. Delgado, J. R. Gallego, M. Canales, J. Ortin, S. Bousnina, and M. Cesana, “An optimization framework for resource allocation in virtual sensor networks,” in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '15)*, pp. 1–7, IEEE, San Diego, Calif, USA, December 2015.
- [31] I. Khan, “Design and analysis of virtualization framework for wireless sensor networks,” in *Proceedings of the IEEE 14th International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '13)*, pp. 1–2, IEEE, Madrid, Spain, June 2013.
- [32] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.
- [33] A. Nedic and A. Ozdaglar, “Cooperative distributed multi-agent optimization,” in *Convex Optimization in Signal Processing and Communications*, Y. Eldar and D. Palomar, Eds., pp. 340–386, University of Cambridge, Cambridge, UK, 2009.
- [34] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.
- [35] S. Kar and J. M. F. Moura, “Convergence rate analysis of distributed gossip (linear parameter) estimation: fundamental limits and tradeoffs,” *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 674–690, 2011.
- [36] I. D. Schizas, G. Mateos, and G. B. Giannakis, “Distributed LMS for consensus-based in-network adaptive processing,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.
- [37] M. H. DeGroot, “Reaching a consensus,” *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.
- [38] H. Long, Z. Qu, X. Fan, and S. Liu, “Distributed extended Kalman filter based on consensus filter for wireless sensor network,” in *Proceedings of the 10th World Congress on Intelligent Control and Automation (WCICA '12)*, pp. 4315–4319, Beijing, China, July 2012.
- [39] R. Grasso, P. Braca, S. Fortunati, and F. Gini, “Distributed underwater glider network with consensus kalman filter for environmental field estimation,” in *Proceedings of the OCEANS*, pp. 1–7, Genova, Italy, May 2015.
- [40] R. Olfati-Saber, “Kalman-Consensus filter: optimality, stability, and performance,” in *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 7036–7042, Chinghai, China, 2009.
- [41] S. Kar and J. M. F. Moura, “Gossip and distributed Kalman filtering: weak consensus under weak detectability,” *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1766–1784, 2011.
- [42] D. Li, S. Kar, J. M. F. Moura, H. V. Poor, and S. Cui, “Distributed Kalman filtering over massive data sets: analysis through large deviations of random Riccati equations,” *IEEE Transactions on Information Theory*, vol. 61, no. 3, pp. 1351–1372, 2015.
- [43] R. Abdolee and B. Champagne, “Diffusion LMS strategies in sensor networks with noisy input data,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 3–14, 2016.
- [44] S. Vlaski, L. Vandenberghe, and H. Ali, “Diffusion stochastic optimization with non-smooth regularizers,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '16)*, pp. 4149–4153, March 2016.
- [45] F. S. Cattivelli and A. H. Sayed, “Diffusion mechanisms for fixed-point distributed Kalman smoothing,” in *Proceedings of the 16th European Signal Processing Conference (EUSIPCO '08)*, pp. 1–4, Lausanne, Switzerland, August 2008.
- [46] Y. Wang, W. P. Tay, and W. Hu, “An energy-efficient diffusion strategy over adaptive networks,” in *Proceedings of the 10th International Conference on Information, Communications and Signal Processing (ICICS '15)*, pp. 1–5, Singapore, December 2015.
- [47] J. Chen, C. Richard, and A. H. Sayed, “Diffusion LMS over multitask networks,” *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.
- [48] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 5492–5498, New Orleans, La, USA, December 2007.
- [49] F. S. Cattivelli and A. H. Sayed, “Diffusion mechanisms for fixed point distributed Kalman smoothing,” in *Proceedings of the 2008 16th European Signal Processing Conference (EUSIPCO '08)*, Lausanne, Switzerland, August 2008.
- [50] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, New Jersey, NJ, USA, 2000.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

