Convergence of Distributed Flooding and Its Application for Distributed Bayesian Filtering

Tiancheng Li, Juan M. Corchado, *Member, IEEE*, and Javier Prieto, *Member, IEEE*

***Abstract*—Distributed flooding is a fundamental information sharing method to get network consensus via peer-to-peer communication. However, a unified consensus-oriented formulation of the algorithm and its convergence performance are not yet explicitly available in the literature. To fill this void in this paper, set-theoretic flooding rules are defined by encapsulating the information of interest in finite sets (one set per node), namely distributed set-theoretic information flooding (DSIF). This leads to a new type of consensus referred to as "collecting consensus," which aims to ensure that all nodes get the same information. Convergence and optimality analyses are provided based on a consistent measure of the degree of consensus of the network. Compared with the prevailing averaging consensus, the proposed DSIF protocol benefits from avoiding repeated use of any information and offering the highest converging efficiency for network consensus while being exposed to increasing node-storage requirements against communication iterations and higher communication load. The protocol has been advocated for distributed nonlinear Bayesian filtering, where each node operates a separate particle filter, and the collecting consensus is pursued on the sensor data alone or jointly with intermediate local estimates. Simulations are provided in detail to demonstrate the theoretical findings.**

***Index Terms*—Consensus, diffusion, distributed tracking, particle filter, sensor network.**

I. Introduction

DSTRIBUTED computation has gained immense attention in the past decade, accompanying the rapid development and popularity of wireless sensor networks. In the successful networking operation, it is often of high interest that each node iteratively shares information with its intermediate neighbors (namely peer-to-peer communication) and consequently the entire network tends to reach a global alignment [1]/consensus [2]–[4] (to a certain degree). Compared to the centralized networking solutions based on a fusion center, distributed networking offers several advantages regarding scalability to adding or removing nodes, immunity to node failure, and dynamic adaptability to network topology changes.

However, there is a significant conflict between the degree of consensus (DoC) and communication requirement, as a higher DoC requires more communication, either more communicating iterations or higher communicating bandwidth which are limited by real time implementation and the communicating affordability of the nodes, respectively. Therefore, it is of paramount significance to seek a good balance for the trade-off so that the network achieves a satisfactory consensus in real-time and with affordable communication costs, which forms the majority of the research in the literature.

One of the most fundamental solutions for network information sharing is the flooding carried out in a distributed manner, by which all nodes synchronously broadcast their information to neighbors from the near to the distant. This protocol is well known in a few areas such as the communications [6]. Given that the network is strongly connected, it is able to achieve complete consensus (CC, i.e., all nodes have exactly the same set of information) after a certain number of iterations of peer-to-peer communication. This is quite appealing in theory but poses crucial challenges to the storage and communication requirement for large networks in practice. In fact, flooding has rarely been investigated in literature for distributed filtering (except for a few works, e.g., [7], [8]) even for small networks, regardless of its fast convergence (to be explicitly demonstrated in this paper) and ease of implementation. To note, there is one work that also refers to distributed flooding [9] which transmits the information of one single node over the network for routing. It is very different from the distributed protocol we consider here.

To date, a unified consensus-oriented formulation of the flooding algorithm and its convergence analysis are still missing in the literature. On one hand, there are cases in which DoC is required in the first priority while the sensor nodes have sufficient node-storage and communicating affordability to do so, for which flooding or even CC is simply preferable. On the other hand, instead of CC, it is more desired to perform flooding in a fewer, affordable, number of iterations for real time realization. What then can be expected and how can the number of iterations be properly determined?

To address the void and to answer the questions above, this paper aims to contribute from three aspects:

1) Formulate the flooding algorithm by encapsulating the information of interest in a finite set at each node and define set-theoretic flooding rules, namely *distributed set-theoretic information flooding* (DSIF), raising a new type of consensus termed *collecting consensus*.
2) Analysis the convergence and optimality of the DSIF protocol, based on a novel, consistent, metric of the DoC.

Manuscript received March 11, 2016; revised July 4, 2016 and October 4, 2016; accepted November 5, 2016. Date of publication; date of current version. This work was supported in part by the Marie Skodowska-Curie Individual Fellowship (H2020-MSCA-IF-2015) and in part by the Spanish Ministry of Economy and Competitiveness under Torres Quevedo Grant PTQ-14-06928. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Vincenzo Matta.

T. Li is with the University of Salamanca, BISITE Research Group, Salamanca 37002, Spain (e-mail: t.c.li@usal.es).

J. M. Corchado is with the University of Salamanca, BISITE Research Group, Salamanca 37002, Spain, and also with the Osaka Institute of Technology, Osaka, Japan (e-mail: corchado@usal.es).

J. Prieto is with the University of Salamanca, BISITE Research Group, Salamanca 37002, Spain, and also with the StageMotion, R&D Department, Palencia 34005, Spain (e-mail: javierp@usal.es).

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

Digital Object Identifier 10.1109/TSIPN.2016.2631944

It is shown that DSIF enjoys the highest efficiency for network consensus among all distributed peer-to-peer communication schemes while suffering from heavier storage and communicational costs.

3) Show how the proposed DSIF scheme can be applied for and can benefit distributed Bayesian filtering, in which each node runs a separate particle filter (PF). Local PFs share sensor data alone or jointly with intermediate posteriors via DSIF. The latter usually needs to be parameterized in order to to reduce the communication cost. The gain and loss to do so are analyzed and demonstrated in simulations.

The reminder of this paper is organized as follows. Notations and definitions regarding networking and three prevailing distributed information sharing protocols are given in Section II. As the main theoretical contribution, the collecting consensus-oriented DSIF is formulated in Section III with an analysis of its convergence and optimality. Section IV shows how to apply DSIF for distributed PF (DPF), with a brief literature review for DPF also given. Simulations are given in Section V and we conclude in Section VI.

II. Notation, Definition and Background

A. Notation

The network topology is represented by a directed graph $G = (V, E)$ with the set of nodes $V = \{1, 2, \cdots, N\}$ and the set of edges $E \subseteq V \times V$. In the directed graph, any edge is denoted by an ordered pair of nodes $(i, j) \in E$, which means node j is directly reachable from node i, where i is called the in-neighbor of j while j is the out-neighbor of i. For any $j \in V$, denote $N_j := \{i \in V | (i, j) \in E, i \neq j\}$, which is the set of all the in-neighbors of node j excluding node j itself. Undirected graph is a special type of directed graph where for any $(i, j) \in E$, we must have $(j, i) \in E$. If there exists a sequence of connected edges as follows

$$\{(i, \cdot), \cdots, (\cdot, j)\} \subseteq E \tag{1}$$

This sequence of edges is called a path from node i to node j (denoted as $Path_{i-\cdots-j}$) and node j is said to be "reachable" from node i. A digraph is said to be "strongly connected" (SC) if any node is reachable from all the other nodes, which is the digraph that can reach CC. For undirected graphs, SC is the same as connectivity [3].

The length of a path is given by the number of edges on that path. The length of the shortest path (perhaps, not unique) from node i to node j is called the distance from node i to node j, denoted as $D(i - j)$. Particularly, $D(i - i) = 0$ and if $(i, j) \in E$, $D(i - j) = 1$. We denote the set of all the nodes that are of distance $t \in \mathbb{N} = \{0, 1, 2, ...\}$ to j as $N_j(t)$ and that of distance $t \in \mathbb{N}$ or smaller to j as $N_j(\leq t)$, namely $N_j(t) := \{n \in V | D(n - j) = t\}, N_j(\leq t) := \{n \in V | D(n - j) \leq t\}$. Obviously, we have $N_j(1) = N_j, N_j(0) = j$.

The largest distance between any two nodes, denoted as D_m, is called the diameter of the graph which is given by

$$D_m = \max_{i,j \in V} D(i - j) \tag{2}$$

Clearly, D_m only exist in SC networks. For any SC networks of at least two nodes, we have $D_m \in [1, N - 1]$, where the left bound corresponds to the fully connected network in which all nodes are in-neighbors of the others, while the right bound corresponds to the weakest connected network where all nodes are on a single chain in order, each having no more than two intermediate neighbors.

With particular regard to the distributed filtering problem, we assume that each node has independent abilities for: (i) filtering calculation, (ii) sensing to collect observations and (iii) communicating to neighbors. The communication is carried out in recursive iterations between neighboring nodes, each iteration consisting of sending no more than one data packet and receiving no more than one data packet. In addition, we need to clarify the following two definitions.

Definition 1 (*real time communication*): Communication is fully carried out between two successive observations and causes no sensor data missing or time-delay to the filter.

Definition 2 (*communication bandwidth*): The maximum size of the data packet that one node can send to or receive from its neighbor per communication.

B. Averaging/Maximum/Minimum Consensus

Here we assume that each node has a local scalar value, referred to as *state*, and it is of interest to compute the average of these values. An averaging consensus algorithm [2], [3] of zero communication time-delay is to reach an agreement regarding the state x_i each node has with local adapting dynamics $u_i(t)$, which can be written in discrete-time as

$$x_i(t + 1) = x_i(t) + u_i(t) \tag{3}$$

where $t \in \mathbb{N}$ denotes the communication iteration, $x_i(0)$ and $x_i(t)$ denote the initial and updated state of node i after iteration t, respectively.

In most of the time, the dynamics $u_i(t)$ is defined as

$$u_i(t) = \sum_{j \in N_i} \omega_{j \to i} \big(x_j(t) - x_i(t)\big) \tag{4}$$

where $\omega_{j \to i}$ is neighboring weight from node j to node i [2]–[4].

The complete convergence of *averaging consensus:* at iteration t states that, for any $i, j \in V$,

$$x_i(t) = x_j(t) \tag{5}$$

and asymptotically convergence (in the sense that $t \to \infty$)

$$\|x_i(t) - x_j(t)\| \leq \epsilon \tag{6}$$

where $\|x - y\|$ is a measure of the discrepancy between x and y, and ϵ is an error bound or margin [10].

In contrast to the average, one might be only interested in the maximum/minimum state, namely maximum/minimum consensus, which defines the iteration as

$$x_i(t + 1) = \max / \min \big\{x_i(t), \{x_j(t)\}_{j \in N_i}\big\} \tag{7}$$

The averaging consensus was well investigated in the community of control and systems [2]–[5]. One major concern is with the convergence, for which the spectral properties of the graph Laplacian play a crucial role [2], [3], [10]-[12]. The second concern is raised by the information correlation/dependence among neighbors (especially when they own in part the same information). To account for this, fusion weights will be assigned to coordinate the nodes, e.g., covariance intersection [14],[15], which has inspired many strategies. In the absence of clear information about the correlation or relative quality of the information among nodes, two weighting methods have been proposed: one is to weight all nodes equally [16] and the other is to weight them according to the size of their neighborhood (named Metropolis weights [17]).

Particularly for the tracking problem, the sensors may updates their observation frequently, preventing sufficient peer-to-peer communicating to get the network converge. This necessitates limiting the number of communicating iterations to gain a trade-off or compromise between a high DoC and little missing or time-delay of sensor data.

C. Gossip and Diffusion

To save communication, one alternative is to apply gossip to randomly choose fewer neighbors at each time (rather than to all neighbors) for averaging. It turns out that under mild conditions this process converges over time asymptotically [18]. Gossip based distributed filtering has been reported in, e.g., [19],[20]. However, gossip experiences the same problem as inefficient/repeated computations, for example, the same set (or largely similar set) of nodes repeatedly fusing their information at different points in time.

As another alternative, diffusion [22]–[24] performs only one iteration of peer-to-peer communication (i.e., the sensing and consensus time scales are the same), avoiding the problem of repeated use of any information. Based on it, the distributed Kalman filter (DKF) [23],[24] does not only share sensor data, but also local intermediate estimates through a diffusion update step. By this, the one-iteration-only communication is actually carried out on two types of data: the sensor data for the incremental update, and the estimates for the diffusion update. Hybrid fusion has been previously studied in the network using a fusion center, e.g., [21].

III. Collecting Consensus and DSIF

A. Collecting Consensus

In this paper, we are interested in an information sharing protocol that does not repeatedly use any information, and that will converge to CC in a definite number of iterations. By CC, we mean that all the nodes have exactly the same information of interest. Such a consensus model in which each node aims to collect information from all reachable nodes via the shortest paths is referred to as collecting consensus. Different to averaging consensus, the information from different nodes remains conditionally independent (or more precisely stated, unfused) until the end of the communication, which requires that nodes have sufficient storage allowance.

To perform collecting consensus, the information that needs to be communicated is encapsulated as a set, and the DSIF algorithm defines the information set dynamics (in contrast to (3)) based on the union operation

$$I_i(t+1) = I_i(t) \cup u_i(t) \tag{8}$$

where $I_i(t)$ and $u_i(t)$ denote the existing and new incoming information set of node i at iteration $t \in \mathbb{N}$ respectively, $I_i(0)$ denotes the initial information set at node i with the size $|I_i(0)| = 1, \forall i \in V$, and $u_i(0)$ the initial dynamics.

As a result of CC, all nodes shall have exactly the same information set, i.e., $\forall i \in V, t \geq D_m$,

$$I_i(t) = \bigcup_{j \in V} I_j(0) \tag{9}$$

To this end, the DSIF algorithm consists of two stages:

1) In the starting iteration, each node collects information from all its in-neighbors

$$u_i(0) = \bigcup_{j \in N_i} I_j(0) \tag{10}$$

2) In the following iterations $t \in \mathbb{N}^+ = \{1, 2, \cdots\}$, each node collects the new information that its in-neighbors have received at the preceding iteration

$$u_i(t) = \bigcup_{j \in N_i} \{I_j(t) \setminus I_j(t-1)\} \tag{11}$$

where $A \setminus B$ is the set difference of A and B, namely the set of all elements that are members of A but not of B and when $t \geq D_m$, we will actually have $u_i(t) = \emptyset$.

As shown in (11), the receiving neighbors will sort out the new received data, which they then transmit to their out-neighbors in the next iteration. In this process, the same information may be repeatedly received over edges, leading to information over-use and communication power waste, which is one defect of the naive flooding protocol, named implosion [6]. To avoid this, we define the set-theoretic information flooding rules in the following.

B. Set-Theoretic Flooding Rules

Rule 1 (*data sending*): Each node only sends to its out-neighbors the new information that has never been flooded before, and does so no more than once in each iteration.

Rule 2 (*data accepting*): Each node will not repeatedly take in the same information either from different in-neighbors or from the same node at different iterations, but only accept the information at its first arrival.

For both rules above, the data from each node shall be associated with a unique ID for distinguishing. Given these two rules respected, we will have $\forall i \in V, t \in \mathbb{N}$,

$$|I_i(t)| = |N_i(\leq t)| \tag{12}$$

To combat time-increasing storage requirement and communication load (when $t < D_m$), each element of data (often called a tuple) may be somehow compressed via e.g., dimension reduction [25] and polynomial encoding [13], under the premise that little or even no information would be lost and the data from different nodes remain conditionally independent.

It is worth noting that in the case of maximum or minimum consensus, there is neither a problem of information set-size growing nor information overuse as the fusion result is always a single maximum or minimum value.

C. Convergence and Optimality of DSIF

To gain insights of the convergence of the proposed DSIF scheme, we need a metric to measure the DoC for collecting consensus, for which we propose a metric based on the size of the information set as follows.

Definition 3 (*DoC*): The DoC, denoted as C^o, of a network with N nodes, is defined as follows

$$C^o(t) = \frac{\sum_{i=1}^{N} |I_i(t)| - N}{N(N-1)} \tag{13}$$

where $t \in \mathbb{N}$ denotes the number of DSIF iterations that has been performed and in the following we limit it to $t \le D_m$. On the DoC, we have the following theorem, which states the convergence property of the DSIF protocol.

Theorem 1: $0 \le C^o(t_1) < C^o(t_2) \le 1, \forall 0 \le t_1 < t_2 \le D_m$.

Proof: Before performing DSIF, each node has its original one unit of data, i.e., $|I_i(0)| = 1, \forall i \in V$. That gives $C^o(0) = 0$. After $t \ge D_m$ DSIF iterations, CC will be reached as all nodes will have the same information, i.e., $|I_i(t)| = N, \forall i \in V$. Furthermore, we have the following two straightforward Claims (for which we omit any proof):

Claim 1: As stated by (12), the size of the information set owned by sensor $i \in V$ will not be reduced during flooding except that data fusion or removal is taken, i.e., $|I_i(t_1)| \le |I_i(t_2)|$ for any $0 \le t_1 < t_2$.

Claim 2: Supposing two nodes g and q are of distance D_m (namely $D(g-q) = D_m$), for any $0 < t \le D_m$, there must exist at least one node $j \in N_q(t)$ on $Path_{g-\ldots-j-\ldots-q}$ satisfying $D(j-q) = t$ whose information will arrive to node q exactly at iteration t and then, we have $|I_q(t-1)| + 1 \le |I_q(t)|$.

From these two claims, we may conclude that $C^o(t_1) < C^o(t_2), \forall 0 \le t_1 < t_2 \le D_m$, to accomplish the proof. ■

Theorem 2: In the sense of DoC as given in (13), the proposed DSIF achieves the highest converging efficiency among all distributed peer-to-peer communication schemes.

Proof: From the definition of the distance between nodes and the DSIF peer-to-peer communication rules, we have two additional straightforward Claims:

Claim 3: All nodes whose information can reach node i in t iterations of peer-to-peer communication belong to $N_i(\le t)$.

Claim 4: All nodes $q \in N_i(\le t)$ will surely flood their information to node i in t DSIF iterations.

A combination of Claims 3 and 4 indicates that the DSIF will gain the largest possible $|I_i(t)|$ for any $i \in V$ and $t \in \mathbb{N}$ as claimed, which entails the converging optimality. ■

D. Trade-off between DoC and Number of Iterations

For a given network topology, the DoC is uniquely determined by the number of DSIF iterations. In turn, one can also determine the required number of iterations for a desired DoC, e.g., $T_c = 0.5$. That is, the DSIF stops at iteration t once

$$C^o(t) \ge T_c \tag{14}$$

Algorithm 1: DSIF operations at node i for DoC T_c.

```
INITIALIZATION:
 t ← 1; C^o(0) ← 0
 I_i(t) = I_i(t−1) ∪ ⋃_{j∈N_i} I_j(t−1)
RECURSIVE FLOODING ITERATION:
 While C^o(t) < T_c
    t ← t + 1
    I_i(t) ← I_i(t−1) ∪ ⋃_{j∈N_i} {I_j(t−1) \ I_j(t−2)}
    C^o(t) ← (∑_{i=1}^{N} |I_i(t)| − N) / (N(N−1))
 End while
 Return: I_i(t)
```

Algorithm 1 summarizes the communicating operations that need to be performed on node i for a given DoC T_c.

For a constant network of a known topology, the minimum number of iterations can be determined a priori by (14). However for time-varying dynamic networks, it needs to be calculated online via a consensus algorithm. To facilitate the use in time-varying networks without burdening any consensus procedures, we define the local DoC metric as follows:

Definition 4 (*Local DoC*): The DoC of node $i \in V$, denoted as C_i^o, after t DSIF iterations, is given as

$$C_i^o(t) = \frac{|I_i(t)| - 1}{N_k - 1} \tag{15}$$

where N_k is number of nodes in the network at time k, which needs to be estimated if unknown. From here we derive the following theorem as the local-node version of Theorem 1.

Theorem 3: $0 \le C_i^o(t_1) \le C_i^o(t_2) \le 1, \forall 0 \le t_1 < t_2 \le D_m$

Proof: This theorem states the same content as Claim 1. As a key difference to Theorem 1, the equality $C_i^o(t_1) = C_i^o(t_2)$ holds when and only when the number of nodes that are of distance t_1 to node $i \in V$ is the same as that of distance t_2, i.e., $|N_i(t_1)| = |N_i(t_2)|$. ■

Setting a threshold, e.g., $T_c = 0.5$ which can be the same or different for different nodes, on the desired local DoC, the consensus updating at node i may stop at iteration t once

$$C_i^o(t) \ge T_c \tag{16}$$

Furthermore, based on DoC we can define the convergence speed (CoS), either globally or locally, to measure the change of the size of the information set at each iteration, which also indicates the local real-time communication bandwidth.

Definition 5 (*CoS*): At iteration $t \in \mathbb{N}$, the global CoS, denoted as C^s, is defined as,

$$C^s(t) = C^o(t) - C^o(t-1) \tag{17}$$

and the local CoS of node $i \in V$ is defined as

$$C_i^s(t) = C_i^o(t) - C_i^o(t-1) \tag{18}$$

Theorem 4: $C^s(t) > 0, C_i^s(t) \ge 0, \forall 1 \le t \le D_m$

Proof: The theorem is immediate from Theorems 1 and 3 as for any $1 \le t \le D_m, i \in V$, we have $C^s(t) = C^o(t) - C^o(t-1) > 0$ from Theorem 1, and $C_i^s(t) = C_i^o(t) - C_i^o(t-1) \ge 0$ from Theorem 3. ■

Theorems 1, 3 and 4 entail an appealing property of the DSIF protocol which will not only converge definitively, but also has a guaranteed converging speed that is globally positive and locally non-negative everywhere and at any iteration until CC is reached. We refer to this as *strong convergence*. It, however, also indicates a (non-negative) increasing storage requirement against communicating iterations. As an alternative to (16), we can build the predetermined threshold on the local CoS, e.g., $T_s = 0.1$, then the minimum number of iterations t needs to satisfy

$$C_i^s(t) \leq T_s \tag{19}$$

But it is critical to note that we do not have any monotonicity on the CoS, e.g., $C^s(t_2) \leq C^s(t_1)$ or $C_i^s(t_2) \leq C_i^s(t_1)$ for $1 \leq t_1 < t_2 \leq D_m$. Therefore, the CoS at iteration t does not say anything of the CoS at iteration $t+1$.

E. Comparison and Practical Consideration

Both metrics of DoC and CoS are clearly defined and easier to calculate than the one proposed for averaging consensus, e.g., convergence rate [10]–[12], steady-state mean-square deviation [4] or disagreement vector [2], [3]. As indicated by Theorem 2, no peer-to-peer communication protocols converge faster than DSIF in terms of DoC. This superiority, however, is achieved at the expense of higher node storage requirements and heavier communication bandwidths. If the size of the data set at one node exceeds its communication bandwidth, multiple iterations will then be needed for that data set, otherwise data fusion is required to control the data size. In the former case, the required number of iterations will increase, while in the latter case the information completeness or independence may not be kept. However, we will not address this issue further here, which is quite problem dependent. In brief, we have the following remark on the respective advantages of averaging consensus, diffusion and collecting consensus.

Remark 1: The averaging consensus takes the lowest communicating bandwidth (always one unit of data) but more iterations to reach any DoCs while the diffusion severely limits the number of iterations (to one only) which may insufficiently use the communication affordability (i.e., more iterations are actually allowed in real time communication). In contrast, the proposed DSIF protocol aims to get the best possible consensus in an real-time-allowed number of iterations, which is therefore particularly suited to small and moderate networks for which the nodes have sufficient storage and communicating power. A means to facilitate its use in large networks is to selectively apply data fusion such as averaging in every several flooding iterations in order to control the data-set size. This will lead to a hybrid protocol that iterates between flooding and averaging consensus, to gain a balance between benefiting from high communication efficiency and suffering from information overuse and slower convergence.

IV. Distributed Bayesian Filtering Using DSIF

A. State-of-the-art DPF Protocols

Before presenting our DPF framework based on DSIF, a brief revisit of the PF algorithm and existing DPF protocols is given below. Suppose that at time k, the local (marginal) posterior at sensor i is represented by a local PF

$$p(\mathbf{x}_k|\mathbf{z}_{i,1:k}) \approx \sum_{m=1}^{M_{i,k}} w_{i,k}^{(m)} \delta\big(\mathbf{x}_k - \mathbf{x}_{i,k}^{(m)}\big) \tag{20}$$

where $\delta(\mathbf{x}-\mathbf{y})$ is the Dirac delta impulse, which equals to one if $\mathbf{x}=\mathbf{y}$ and to zero otherwise, $\mathbf{x}_k$ is the true state vector, $\mathbf{z}_{i,1:k}$ is the observation serial, $\mathbf{x}_{i,k}^{(m)}$ and $w_{i,k}^{(m)}$ are the state and normalized weight of the mth particle respectively, $M_{i,k}$ is the total number of particles at filtering time k.

The essence of the PF is to assess how well each particle conforms to the state model and explains the observations, using this assessment to generate a weighted sample approximation to the Bayesian posterior, and thereby form sub-optimal state estimates. Given local measurement $\mathbf{z}_{i,k}, i \in V$, the weights of the particles are evaluated over time based on the sequential importance sampling (SIS) principle as

$$w_{i,k}^{(m)} \propto w_{i,k-1}^{(m)} \frac{p(\mathbf{z}_{i,k}|\mathbf{x}_{i,k}^{(m)})p(\mathbf{x}_{i,k}^{(m)}|\mathbf{x}_{i,k-1}^{(m)})}{\pi\big(\mathbf{x}_{i,k}^{(m)}|\mathbf{x}_{i,k-1}^{(m)}, \mathbf{z}_{i,1:k}\big)} \tag{21}$$

where $\pi(\cdot)$ is a proposal to generate particles, and in general its design shall take into account both the newest measurement $z_{i,k}$ and the prior in order to best match the posterior; see e.g. [19], [28], [31]. The use of the observation in the sampling proposal design is particularly helpful (and even necessary for avoiding sample degeneracy) when the observation is very accurate. However, caution should be exercised here since the repeated use of the observation (both for proposal design and in likelihood calculation) may not benefit the filter when the observation suffers from significant noise [42].

In addition to SIS, resampling is usually required to reduce the weight variance when it exceeds a certain threshold, so that all particles will have equal or approximate weights while the posterior distribution can be the best maintained [31], [32]. This has often been referred to as sampling importance resampling (SIR), which is the core of the majority of existing PFs. We assume the reader is familiar with the centralized PF and so limit ourselves hereafter to the distributed implementation, in which local nodes carry out PF calculations in parallel and meanwhile share information with their neighbors to assist their filters. For this, a variety of information sharing protocols have been proposed, which can be classified as follows:

1) *Sequential information passing*: Information transmits in a sequential, predefined manner from a node to one of its neighboring nodes via a cyclic path until the entire network is traversed [43]. The sequential realm is sensitive to the mobility and failure of nodes/edges and is time-consuming.
2) *Flooding*: As addressed, the flooding protocol provides the fastest albeit communication-intensive way to spread information over the network [7], [8], but, neither any clue to determine the number of communication iterations in order to compromise real time realization and DoC nor any convergence results has been shown.

3) *Averaging consensus*. There is a large body of work concerning averaging consensus-based distributed filtering. The data transmitted between neighboring nodes can be posterior statistics in the form of Gaussian component [33] /GM [29]–[30] or generalized probability densities [36]–[37], likelihood [26]–[28], particle set [34]–[35] or raw observations [38]. Excellent surveys are also available such as a taxonomy of DPFs [39], a comparison of several belief consensus algorithms [40] and a recent survey of convergence and error propagation of DPFs [28]. In summary, complete information sharing affords better accuracy but has higher communication requirements, such as [34]–[35] that exchange all particles. Parameter approximation [26]–[33] or random gossip [19]–[20] can significantly reduce the communication cost, but may lead to a deterioration in the filter performance.
4) *Diffusion*: The diffusion scheme addressed in Section II.C also provides a competitive alternative to the averaging consensus for DPF [7]-[8], [41].

We note that the sensor data can be either simple (e.g., range, bearing) or complex (e.g., image data). To avoid distracting from the key contribution of this paper on collecting consensus and the DSIF protocol, we only consider the former case for simplicity. For the latter case, one may consider compressing the sensor data, e.g., [25]-[26], [12]-[13] or transmitting the low-dimensional likelihood for replacement [26]–[27]. At the current stage, we have not considered complicated network issues such as communication constraints, e.g., [45]–[46], and asynchronous sensing, e.g., [47]–[48]. However, we note all of these issues are valuable to be investigated on the base of the proposed DSIF protocol.

B. DSIF on Sensor Data and on Local Posterior

In the proposed DPF framework, the DSIF scheme will be applied on the sensor data alone or jointly on local posteriors. In the latter, we propose parameterizing the posterior to save communication. Since a vast number of random numbers are required by the PF, it is communication intensive to run consensus on them, and it is not our intention to do so.

First, DSIF is implemented on the sensor data including the target-observations (and uncertainties) associated with the sensor ID, all as one unit. To note, the sensor position is often required for likelihood calculation and therefore can serve as the unique sensor ID for distinguishing. Then, the resultant consensus on sensor data with sensor profiles given a priori, is equivalent to collecting consensus on the likelihood which is required for PF updating. A likelihood function contains the information of both the sensor data and the sensor profile in a more compact manner. But for simplicity of understanding, we keep addressing consensus on sensor data.

The filtering posteriors obtained at different nodes, referred to as local posteriors, will be different, even if CC is reached on sensor data over the network where the difference attributes to the different random numbers. If DoC is low on sensor data, the difference between local posteriors will be relatively significant. As such, we may apply the second DSIF scheme to fuse local posteriors among neighbors as well as to get the local LMS (least mean squares) estimate; we refer to this step as diffusion, in parallel to [24]. By this, each node aims to improve their local estimate with regard to their neighbors' posterior. However, parameter approximation of local posteriors, typically via Gaussian or GM approximation, is needed (otherwise massive communication will be triggered if the complete posterior is communicated by transmitting the entire particle set), which will in turn introduce approximation errors to the posterior. This trade-off is much problem-dependent and will determine whether the second DSIF is worthwhile.

The operations that need to be conducted on each sensor in the proposed distributed PF is summarized in Algorithm 2. In it, steps 1-a and 1-b are independent of each other and therefore can be carried out in either order or in parallel. Sensor data DSIF and posterior DSIF have been implemented t_1 and t_2 iterations respectively, where t_1 and t_2 are not necessarily equal but are determined for respective desired or the largest affordable DoCs as addressed in Section III. They show complementary features and resemble the Incremental and Diffusion updates of the diffusion-based DKF [24]. But, there are obvious differences:

1) Our framework is developed for nonlinear models which releases the requirement of linear system functions and even Gaussian assumption of the posterior;
2) Our consensus protocol does not limit information sharing between neighbors to one iteration only but instead, the DoC will be pursued as much as the real time communication allows;
3) Our diffusion update (Step 5 in Algorithm 2) is an optional step, which is advocated for re-setting local posteriors only when local posteriors are significantly different (as a consequence of a low DoC on the sensor data achieved in the first DSIF implementation). When the difference between local posteriors is insignificant (because of a high DoC achieved on the sensor data), there will be less need to further fuse them and so it may be better not to diffuse local posteriors since the errors introduced due to parameterization can be more significant than the benefit. This is a critical point. We will demonstrate this in detail through simulations in Section V. In addition, we provide two easy-to-implement diffusion choices.
4) We point out that the proposed two DSIF procedures can be performed jointly, although this may not reduce the communication load and the storage requirement in total; see the following Remark 2.

Remark 2: Two DSIF implementations regarding the sensor data and the local filter estimates form the starting step and the end step of each filtering iteration, respectively. In the time series, they are adjacent. Therefore, they may be combined in one joint consensus scheme at some stages (which however does not necessarily indicate that $t_1 = t_2$), i.e., the local estimates obtained at filtering time k can be combined with sensor data received at time $k+1$ as one unit of data, both sharing the same node ID for DSIF. Then, the initial information set at node $i \in V$ can be defined as

$$I_{i,k}(0) := \{\hat{\mathbf{x}}_{i,k-1}, P_{i,k-1}\} \cup \mathbf{z}_{i,k} \tag{22}$$

Algorithm 2: Distributed PF calculation executed on node i.

Step 1-a Filter prediction: Propagate the particles $\mathbf{x}_{i,k-1}^{(m)}$ to $\mathbf{x}_{i,k}^{(m)}$ for all $m = 1, \cdots, M_{i,k}$, according to the proposal function $\pi\big(\mathbf{x}_{i,k}^{(m)}|\mathbf{x}_{i,k-1}^{(m)}, \mathbf{z}_{i,1:k}\big)$. At $k = 0$, particles are sampled from an initial proposal π_0 instead for filter initialization.

Step 1-b 1st *DSIF*: Perform t_1 DSIF iterations on sensor data as given in Algorithm 1, resulting in a combined measurement set $\mathbf{Z}_{i,k} = \{\mathbf{z}_{j,k}\}_{j\in N_i(\le t_1)}$. This step is carried out whenever new measurements become available.

Step 2 Filter updating: Re-weight all particles via (21) ($\mathbf{z}_{i,t}$ therein shall be replaced by $\mathbf{Z}_{i,k}$ obtained in the 1st DSIF) and then normalize them as follows

$$w_{i,k}^{(m)} \leftarrow w_{i,k}^{(m)} \bigg/ \sum_{m=1}^{M_{i,k}} w_{i,k}^{(m)} \tag{23}$$

Step 3 Estimate extraction: Extract local estimate $\hat{\mathbf{x}}_{i,k}$ and calculate their covariance $P_{i,k}$ from the local random measure $\chi_{i,k} = \big\{\mathbf{x}_{i,k}^{(m)}, w_{i,k}^{(m)}\big\}_{m=1,2,\cdots,M_{i,k}}$ as follows

$$\hat{\mathbf{x}}_{i,k} = \sum_{m=1}^{M_{i,k}} w_{i,k}^{(m)} \mathbf{x}_{i,k}^{(m)} \tag{24}$$

$$P_{i,k} = \sum_{m=1}^{M_{i,k}} w_{i,k}^{(m)} \left(\mathbf{x}_{i,k}^{(m)} - \hat{\mathbf{x}}_{i,k}^{(m)}\right)\left(\mathbf{x}_{i,k}^{(m)} - \hat{\mathbf{x}}_{i,k}^{(m)}\right)^T \tag{25}$$

Step 4 2nd *DSIF*: Perform t_2 DSIF on local estimates obtained in Step 3, resulting in a set of intermediate estimates $\{\hat{\mathbf{x}}_{j,k}, P_{j,k}\}_{j\in N_i(\le t_2)}$, which will be fused in the LMS sense as follows

$$\hat{\mathbf{x}}_{i,k}^{\text{LMS}} = \sum_{j\in N_i(\le t_2)} \hat{\mathbf{x}}_{j,k} P_{j,k}^{-1} \tag{26}$$

$$P_{i,k}^{\text{LMS}} = \left(\sum_{j\in N_i(\le t_2)} P_{j,k}^{-1}\right)^{-1} \tag{27}$$

This also offers the local filter output at sensor i.

Step 5 Diffusion: As an option, the shared filter estimates given in Step 4 can be used to re-set the local PF posterior $\chi_{i,k}$. If so, there are two choices (the second is expected to have a higher approximation accuracy than the first).

1) Re-set $\chi_{i,k}$ as the LMS fused Gaussian distribution, as is done in the Gaussian PF [49].

$$\chi_{i,k} \leftarrow \mathcal{N}\big(\hat{\mathbf{x}}_{i,k}^{\text{LMS}}, P_{i,k}^{\text{LMS}}\big) \tag{28}$$

2) Re-set $\chi_{i,k}$ as the shared GM before performing LMS fusion, as is done in the Gaussian sum PF [50]

$$\chi_{i,k} \leftarrow \sum_{j\in N_i(\le t_2)} \mathcal{N}\big(\hat{\mathbf{x}}_{j,k}, P_{j,k}\big) \tag{29}$$

Step 6 Resampling: Sample from the updated particle set [32] if the variance of weights exceeds a specified threshold and if Step 5 is not applied. If Step 5 is applied, sample from the diffused Gaussian or GM distribution $\chi_{i,k}$ given by Step 5 to generate a new particle set. Update $k \leftarrow k - 1$ and go to the next filtering iteration.

V. Simulations

In this section, we consider tracking a target that moves in the $x - y$ plane by using the proposed DPF based on a constant sensor network earlier appeared in [39] as given in Fig. 1. The network has totally 10 sensors and a diameter $D_m = 4$. The simulation models and parameters are the same to [27]. In specific, we have the initial state as $\mathbf{x}_0 = [4, 0.5, 4, 0.5]^T$. The Markov transition model that governs the target movement of nearly constant velocity is given by

$$\mathbf{x}_k = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix} \mathbf{u}_k \tag{30}$$

where $\mathbf{x}_k = [p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k}]^T$, $[p_{x,k}, p_{y,k}]^T$ gives the position and $[\dot{p}_{x,k}, \dot{p}_{y,k}]^T$ the velocity, $\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}_2, 0.00035\mathbf{I}_2)$.

The target emits an acoustic or radio signal with a known constant transmit power P_t that can be received by all sensors independently, i.e., the scalar measurement function of sensor i located at $[s_{i,x}, s_{i,y}]^T$ about target $\mathbf{x}_k$ is

$$z_{i,k} = \frac{\propto P_t}{\|[p_{x,k}, p_{y,k}]^T - [s_{i,x}, s_{i,y}]^T\|^\gamma} + v_k \tag{31}$$

where $\propto$ is a constant that depends on several factors such as fast and slow fading, and gains in the transmitter and receiver antennas, γ is the path loss exponent [44], and $v_k \sim \mathcal{N}(0, \sigma_v^2)$ is the measurement noise. In parallel to [27], we set simply $\propto P_t = 10, \gamma = 1, \sigma_v^2 = 0.001$.

When multiple synchronous observations are available, the weight of particles is updated by multiplying the likelihoods given by each available measurement. That is,

$$p\big(Z_{i,k}|x_{i,k}^{(m)}\big) = \prod_{j\in N_i(\le t)} p\big(z_{j,k}|x_{i,k}^{(m)}\big) \tag{32}$$

where $Z_{i,k} = \{z_{i,k}\}_{j\in N_i(\le t)}$ is the measurement set at sensor i gained in the first DSIF procedure of total t iterations.

For any sensor $i \in V$, the necessary and sufficient number of iterations, denoted as $D_{m,i}$, to receive the information from all the other sensors can be given by

$$D_{m,i} := \max_{j\in V} D(j - i) \tag{33}$$

We design three groups of simulations in the following three subsections that use the same ground truths to evaluate or compare the following five PF protocols, where the first three are distributed while the last two are centralized. All PFs use the same number of particles ($M = 1000$).

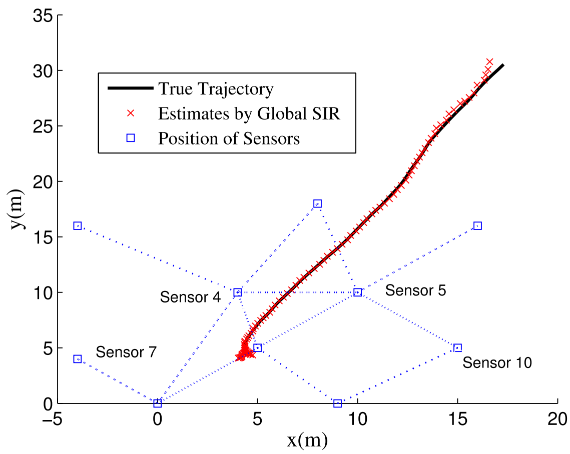


Fig. 1. The topology of the sensor network, the target trajectory and its estimate given a by a global SIR filter in one trial.

1) *C-SIR*: we apply DSIF only on the sensor data, named Consensus without Diffusion (i.e., Steps 4 and 5 are not applied in Algorithm 2). In this case, each local PF is a SIR filter that is free of any Gaussian assumption;
2) *CD-GMPF*: we apply DSIF on both sensor data and local estimates named Consensus with Diffusion (i.e. Steps 4 and 5 are applied in Algorithm 2). In this case, each local PF is a Gaussian sum PF that applies (29) for posterior approximation and fusion;
3) *L-C-SIR*: the Likelihood Consensus-based SIR filter [27] can be viewed as a special case of our C-SIR filter that applies sensor data averaging consensus (for likelihood multiplying) at each iteration. For fast converging, the Metropolis weights strategy [52] is employed for averaging in the L-C-SIR filter;
4) *Local-SIR/GMPF*: local SIR filter or GMPF that does not communicate with each other at all;
5) *Global-SIR*: a centralized SIR filter that is able to access all sensor observations at all times.

To mitigate the problem of sample impoverishment that is often caused by resampling in the SIR filters, the minimum-sampling-variance resampling [32] is applied when and only when the effective sample size is smaller than $M/2$ and if applied, a roughening noise that is equivalent to half of $\mathbf{u}_k$ will be used [31].

To measure the filtering accuracy, we calculate the root mean square error (RMSE) on both the position estimate and the velocity estimate, respectively, as follows

$$\text{RMSEpos}_k = \sqrt{\frac{1}{C}\sum_{c=1}^{C}(x_{k,c}-\hat{x}_{k,c})^2+(y_{k,c}-\hat{y}_{k,c})^2} \quad (34)$$

$$\text{RMSEvel}_k = \sqrt{\frac{1}{C}\sum_{c=1}^{C}(\dot{x}_{k,c}-\hat{\dot{x}}_{k,c})^2+(\dot{y}_{k,c}-\hat{\dot{y}}_{k,c})^2} \quad (35)$$

where $[\hat{x}_{k,c},\hat{y}_{k,c}]^T$ and $[\hat{\dot{x}}_{k,c},\hat{\dot{y}}_{k,c}]^T$ are the position-estimate and velocity-estimate given at filtering time k in trial c, respectively, and $C = 20$ is the total number of MC trials. Further, the average position RMSE is defined as the mean of RMSEpos_k over the entire simulation period of 100 filtering iterations. In each trial, the ground truth is independently generated (for generality). In all trials, the prior distribution of the particle set is initialized around the true state as $\mathcal{N}(x_0, P_0)$, with $P_0 = \text{diag}[2, 0.001, 2, 0.001]^T$.

TABLE I
DoC Achieved at Each DSIF Iteration (Local and Global)

	Sensor 4	Sensor 5	Sensor 7	Sensor 10	Global
$t=0$	0	0	0	0	0
$t=1$	5/9	5/9	1/9	2/9	26/90
$t=2$	1	8/9	3/9	6/9	61/90
$t=3$	1	1	7/9	8/9	86/90
$t=4$	1	1	1	1	1

In particular, we will assess the filter performance at four representative sensors, marked in Fig. 1 as sensors 4, 5, 7 and 10. For them, we have $D_{m,4} = 2, D_{m,5} = 3, D_{m,7} = 4, D_{m,10} = 4$. This means that sensor 4 will achieve CC first (after 2 iterations) while sensors 7 and 10 will be the last (after 4 iterations). For different numbers of DSIF iterations, the global and local DoCs are given in Table I. Particularly, for $t = 1$, we have the global DoC determined as

$$C^o(1) = \frac{|E|}{N(N-1)} \quad (36)$$

where $|E|$ is the number of edges; (a, b) and (b, a) are counted as two different edges.

A. *Consensus without Diffusion*

In this case, each sensor operates a separate SIR filter. Sensors are assumed conditionally independent and use different random numbers. The posteriors obtained by sensors will be different from both each other and the global/local PF, even given that they all reach CC on sensor data.

For different numbers of DSIF iterations from 0 (no consensus at all) to 4 (D_m), the RMSEs of the position and velocity estimation of local C-SIR filters and the global SIR filter are given in Fig. 2.(a)-(e) respectively, corresponding to different DoCs. The average RMSEs over 100 filtering steps against the number of DSIF iterations are given in Fig. 2.(f). The results clearly demonstrate that:

1) A single passive sensor is not capable of delivering good tracking in this problem as the RMSEs given by local PFs are much higher than that provided by DPFs; this necessitates the collaboration of multiple geographically dispersed sensors;
2) The more informative sensor data used, the better the filter performance;
3) The larger DoC, the closer the local PF performance to the centralized PF, i.e., local filters converge to the global filter against iterations as the DoC increases;
4) Once CC is reached, the performance of the local PF is very close to that of the centralized PF (with regard to

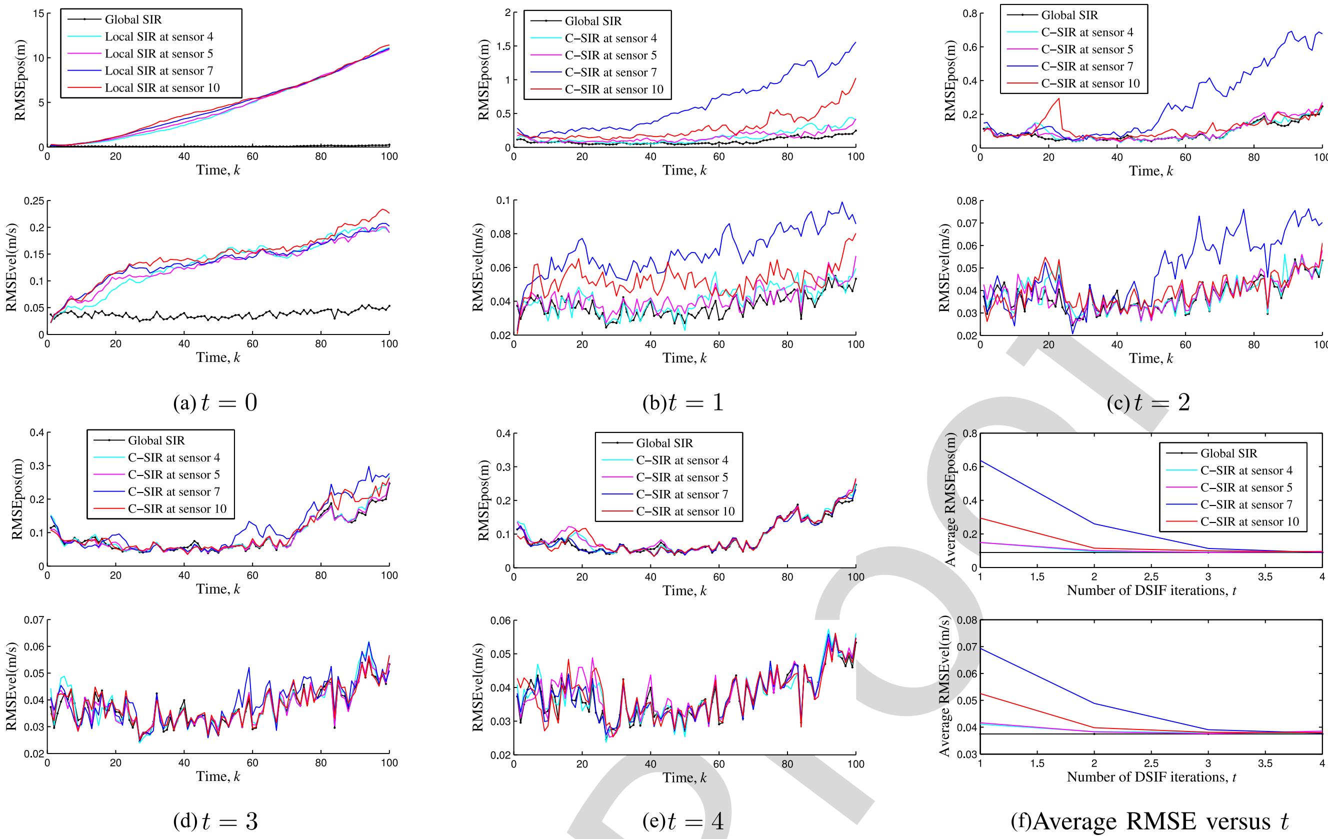


Fig. 2. Position and velocity RMSE of C-SIR filters with different numbers of DSIF iterations, comparing with the global SIR filter.

both position and velocity) but still not the same, since different random numbers are used.

Based on the measure of DoC, we are able to approximately determine how much information divergence different nodes will have and what payoff can be expected if one more or one less iteration of peer-to-peer communication is employed. For example, when the number of iterations is $t = 3$, the global DoC is as high as $86/90$, close to 1, which agrees with the slight difference between Fig. 3(a) and (b). This is a valuable part of the metric of DoC.

B. Consensus with Diffusion

In this case, each sensor runs a separate GMPF. Collecting consensus are applied on both the sensor data and intermediate estimates jointly in a single DSIF procedure (and set $t_1 = t_2$). Because of the GM diffusion of intermediate estimates, the local fused estimates are expected to be closer to each other. If CC is reached, they shall be exactly the same. In parallel to the last simulation, different numbers of DSIF iterations from 0 to 4 are employed to the CD-GMPFs, which are compared with the (centralized) global SIR PF in Fig. 3(a)–(e) respectively. The average RMSEs of these filters against the number of DSIF iterations are given in Fig. 3(f).

We use the same ground truth (20 MC trials) regarding the trajectories and sensor observations as the last simulation. Compared to the last simulation, we can find that

1) A single passive sensor can still hardly work well when the local SIR filters are replaced by local GMPFs;
2) Given the same number of DSIF iterations $t = 1, 2$, CD-GMPFs perform much better than C-SIR and are much closer to each other; this is because of the second DSIF scheme on the posteriors over the network which enhances the consensus to improve local estimates;
3) Given $t = 3, 4$ iterations, the local CD-GMPFs perform almost the same but different to the global SIR filter;
4) Given CC achieved, the RMSEs of all local GMPFs are exactly the same but are inferior to the global SIR filter, especially at the later stage in this tracking example. Analysis and discussion will be given next.

C. Comparison and Discussion

Finally, we compare both types of DSIF-based DPFs with the L-C-SIR filter [27], [39]. The key difference of the likelihood consensus to DSIF is that each node fuses information intermediately after receiving them and therefore the communication cost is lower, but it is exposed to repeated use of information and slower convergence.

First, for $t = 4$, the average (over all nodes) position RMSEs of the C-SIR, CD-GMPF and the L-C-SIR filters are given in Fig. 4. It shows that the C-SIR filter achieves the closest performance to that of the centralized filter. We further calculate the mean of these average RMSEs for $t = 0$ to 8 and the results are given in Fig. 5. It shows that these consensus protocols can all significantly improve the filter performance as compared to the local filter that applies no consensus and converges against communication iterations.

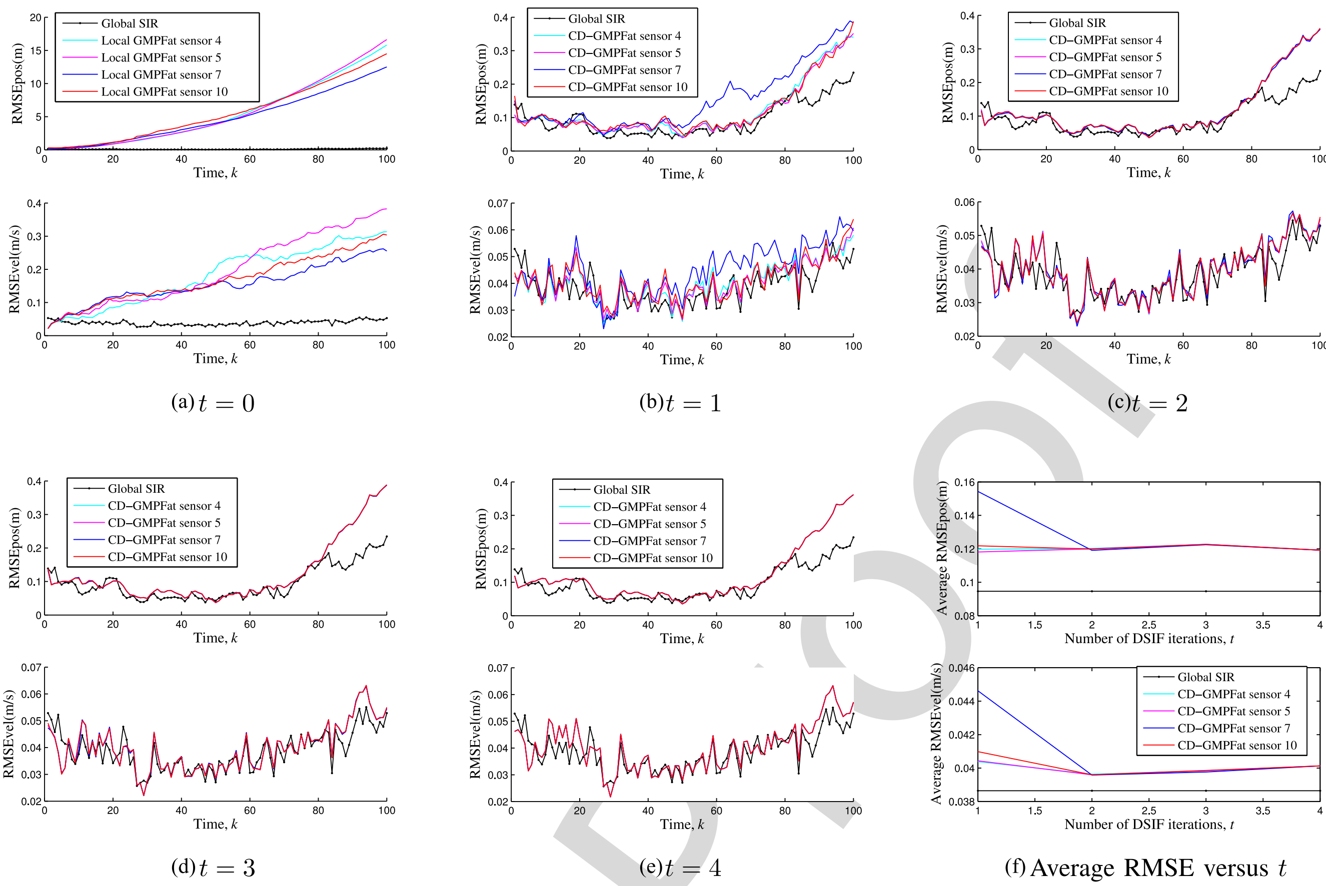


Fig. 3. Position and velocity RMSE of CD-GMPFs with different numbers of DSIF iterations, comparing to the global SIR filter.

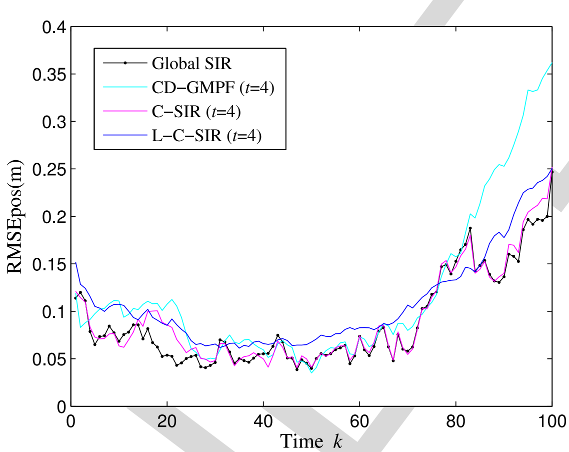


Fig. 4. Position RMSE of different DPFs applying 4 iterations of peer-to-peer communication.

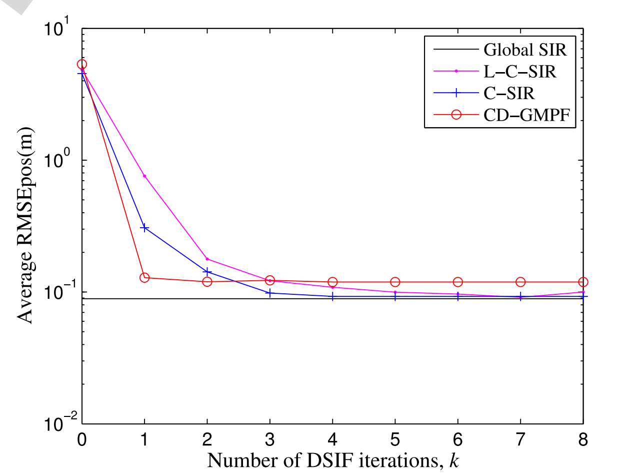


Fig. 5. Average position RMSE of different DPFs over 100 filtering steps against the number of peer-to-peer communication iterations.

Furthermore, we have the following observations, which show more insights of these three types of DPFs:

1) DSIF based C-SIR and CD-GMPF converge faster than the averaging consensus-based L-C-SIR filter at the expense of higher communication cost. CD-GMPF converges the fastest but it suffers from a larger RMSE at the end, all due to its diffusion step that shares information among nodes more thoroughly than without diffusion but also introduce errors;
2) For a relatively small number of iterations that correspond to a low DoC on observation (which may lead to a large discrepancy between local nodes' posteriors), the C-SIR filter is inferior to the CD-GMPF, as shown in Fig. 5 (also told by comparing between Figs. 2(f) and 3(f)). In this case, the diffusion update leads to earlier convergence and better performance for the filter. This is in line with the findings reported in [23];

3) For a large number of iterations that correspond to a high DoC on observation and consequently on posterior (leaving little space to benefit from posterior fusion), the diffusion update of the CD-GMPF is not so preferable; instead, the GM approximation error caused in the diffusion might be more significant than the benefit it can offer, resulting in an overall filter degradation. We must note that if the whole particle sets are transmitted for diffusion without any approximations, and also the dependence between the posteriors are accounted for properly in the diffusion update, it shall always be beneficial in theory regardless of the much greater cost in communication and local fusion calculation.

These results confirm our theoretical prediction and demonstrate further that, both approximation and data fusion during communication can be either beneficial or counterproductive. Generally speaking, parametric approximation can speed up the convergence but also introduces errors. Data fusion such as averaging will reduce communication costs but will also slow down the convergence (primarily because of repeated use of information in data fusion). In practice, we have to contend with a compromise between fast convergence, accurate information sharing and low storage and communication cost. Inspired by these findings, a problem-oriented hybrid protocol that takes the advantages of different approaches while minimizing the side-effects will be valuable.

VI. Conclusion

Flooding is an efficient albeit simple solution for information sharing over networks and is the basis of many other networking protocols. In this paper, we formulated it from a set-theoretic perspective, named distributed set-theoretic information flooding (DSIF). This led to a novel consensus protocol for networking referred to as collecting consensus, which has significant both advantages and disadvantages over averaging consensus and diffusion. We have analyzed the explicit convergence and optimality of DSIF based on a novel metric of DoC (degree of consensus). Practical solutions have been proposed either to determine the minimum number of iterations required for any desired DoC or to calculate the DoC that can be achieved by an actual number of iterations. It has also been noted that to save communication, data fusion (such as averaging) can be employed during flooding, which however may cause repeated information use and slower convergence. This trade-off has been analyzed.

Based on the theoretical results, a distributed particle filter framework is proposed and implemented for nonlinear target tracking which applies DSIF on sensor data alone or jointly with intermediate estimates. Simulations have demonstrated the convergence of the DSIF (faster than averaging consensus), the relationship between the filter performance and the DoC, and the advantage and disadvantage of applying parameterized approximation and data fusion for networking.

Acknowledgment

The authors would like to thank the insightful discussions with Prof. F. Hlawatsch, and Dr. F. Meyer to this paper as well as the likelihood consensus code from Dr. O. Hlinka. The authors would also like to acknowledge one anonymous reviewer's idea for a protocol which 1) applies DSIF for a certain number of time intervals, then 2) reduces the amount of stored information through a weighted averaging — like step, and 3) restarts with DSIF approach. We believe that this is a valuable research topic.

References

[1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003

[2] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007

[4] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[6] W. Dargie, C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2010.

[7] S. S. Dias and M. G. S. Bruno, "Cooperative target tracking using decentralized particle filtering and RSS Sensors," *IEEE Trans. Signal Process.*, vol. 61, no. 14, pp. 3632–3646, Jul. 2013

[8] M. G. S. Bruno and S. S. Dias, "Cooperative emitter tracking using Rao-Blackwellized random exchange diffusion particle filters," *EURASIP J. Adv. Signal Process.*, vol. 19, pp. 1–18, 2014.

[9] H. Liu, X. Jia, P.-J. Wan, X. Liu, and F. F. Yao, "A distributed and efficient flooding scheme using 1-Hop information in mobile ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 5, 658–671, May 2007.

[10] V. Yadav and M. V. Salapaka, "Distributed protocol for determining when averaging consensus is reached," in *Proc. 45th Annu. Allerton Conf.*, Sep. 2007, pp. 715–720.

[11] A. Olshevsky and J. N. Tsitsiklis. "Convergence speed in distributed consensus and averaging," *SIAM Rev.*, vol. 53, no. 4, pp. 747–772, 2011.

[12] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, "Average consensus on networks with quantized communication," *Int. J. Robust Nonlinear Control*, vol. 19, no. 16, pp. 1787–1816, 2009.

[13] Y. Minsky, A. Trachtenberg, and R. Zippel, "Set reconciliation with nearly optimal communication complexity," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2213–2218, Sep. 2003.

[14] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. Amer. Control Conf.*, 1997, vol. 4, pp. 2369–2373.

[15] T. Bailey, S. Julier, and G. Agamennoni, "On conservative fusion of information with unknown non-Gaussian dependence," in *Proc. 2012 15th Int. Conf. Inf. Fusion*, Jul. 2012, pp. 1876–1883.

[16] D. Ciuonzo, A. De Maio, and P. Salvo Rossi, "A systematic framework for composite hypothesis testing of independent Bernoulli Trials," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1249–1253, Sep. 2015.

[17] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, 2005, pp. 63–70.

[18] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[19] D. Ustebay, M. Coates, and M. Rabbat, "Distributed auxiliary particle filters using selective gossip," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, May 2011, pp. 3296–3299.

[20] S. D. Gupta, M. Coates, and M. Rabbat, "Error Propagation in gossip-based distributed particle filters," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 3, pp. 148–163, Sep. 2015.

[21] X. R. Li, Y. Zhu, J. Wang, and C. Han, "Optimal linear estimation fusion—Part I: Unified fusion rules," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2192–2208, Sep. 2003.

[22] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[23] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

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

[25] K. Zhang, X. R. Li, P. Zhang, and H. Li, "Optimal linear estimation fusion—Part VI: Sensor data compression," in *Proc. 6th Int. Conf. Inf. Fusion*, Cairns, Qld, Australia, 2003, pp. 221–228.

[26] M. Coates, "Distributed particle filters for sensor networks," in *Proc. Int. Symp. Info. Process. Sens. Netw.*, Berkeley, CA, USA, Apr. 2004, pp. 99–107.

[27] O. Hlinka, O. Slucciak, F. Hlawatsch, P. M. Djuric, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4334–4349, Aug. 2012.

[28] O. Hlinka, F. Hlawatsch, and P. M. Djuri, "Consensus-based distributed particle filtering with distributed proposal adaptation," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3029–3041, Jun. 2014

[29] X. Sheng, Y. H. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," in *Proc. Int. Symp. Inf. Process. Sens. Netw.*, Los Angeles, CA, USA, Apr. 2005, pp. 181–188.

[30] D. Gu, "Distributed particle filter for target tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 3856–3861.

[31] T. Li, M. Bolic, and P. Djuric, "Resampling methods for particle filtering: Classification, implementation, and strategies," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015.

[32] T. Li, G. Villarrubia, S. Sun, J. M. Corchado, and J. Bajo, "Resampling methods for particle filtering: identical distribution, a new method and comparable study," *Frontiers Inf. Technol. Electron. Eng.*, vol. 16, no. 11, pp. 969–984, 2015.

[33] A. Mohammadi and A. Asif, "Distributed particle filter implementation with intermittent/irregular consensus convergence," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2572–2587, May 2013.

[34] S. H. Lee and M. West, "Convergence of the Markov chain distributed particle filter (MCDPF)," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 801–812, Feb. 2013.

[35] I. L. Manuel and A. N. Bishop, "Distributed Monte Carlo information fusion and distributed particle filtering," in *Proc. 19th World Congr. Int. Fed. Autom. Control*, Cape Town, South Africa, Aug. 2014, pp. 8681–8688.

[36] M. Uney, D. E. Clark, and S. J. Julier, "Distributed fusion of PHD filters via exponential mixture densities," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 521–531, Jun. 2013

[37] G. Battistelli and L. Chisci, "Kullback-Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[38] P. M. Djuric, J. Beaudeau, and M. F. Bugallo, "Non-centralized target tracking with mobile agents," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, 2011, pp. 5928–5931.

[39] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 61–81, Jan. 2013.

[40] V. Savic, H. Wymeersch, and S. Zazo, "Belief consensus algorithms for fast distributed target tracking in wireless sensor networks," *Signal Process.*, vol. 95, 149–160, 2014.

[41] A. Mohammadi and A. Asif, "Diffusive particle filtering for distributed multisensory estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 3801–3805.

[42] T. Li, J. M. Corchado, J. Bajo, S. Sun, and J. F. Paz, "Effectiveness of Bayesian Filters: An information fusion perspective," *Inf. Sci.*, vol. 329, pp. 670–689, 2016.

[43] T. Zhao and A. Nehorai, "Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1511–1524, Apr. 2007.

[44] J. Prieto, S. Mazuelas, A. Bahillo, P. Fernndez, R. M. Lorenzo, and E. J. Abril, "Adaptive data fusion for wireless localization in harsh environments," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1585–1596, Apr. 2012.

[45] A. Mohammadi and A. Asif, "Distributed consensus + innovation particle filtering for bearing/range tracking with communication constraints," *IEEE Trans. Signal Process*, vol. 63, no. 3, pp. 620–635, Feb. 2015.

[46] S. Seifzadeh, B. Khaleghi, and F. Karray, "Distributed soft-data-constrained multi-model particle filter," *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 384–394, Mar. 2015.

[47] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Distributed sequential estimation in asynchronous wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1965–1969, Nov. 2015.

[48] G. Zhu, F. Zhou, L. Xie, R. Jiang, and Y. Chen, "Sequential asynchronous filters for target tracking in wireless sensor networks," *IEEE Sens. J.*, vol. 14, no. 9, pp. 3174–3182, Sep. 2014.

[49] J. Kotecha and P. M. Djuri, "Gaussian particle filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[50] J. Kotecha and P. M. Djuri, "Gaussian sum particle filtering," *IEEE Trans. Signal Process.*, vol 51, no. 10, pp. 2602–2612, Oct. 2003.

Tiancheng Li received the Bachelor's degree in mechanical and electrical engineering, with a minor degree in Business Administration, from Harbin Engineering University, Harbin, China, in 2008, the first Ph.D. degree in electrical and electronic engineering from London South Bank University, London, UK., in 2013, and the second Doctoral degree in mechatronics engineering from Northwestern Polytechnical University, Xian Shi, China, in 2015. Since June 2014, he has been a Research Associate with the BISITE Research Group, University of Salamanca, Spain and presently, he holds a Marie Skodowska-Curie Individual Fellowship H2020-MSCA-IF-2015 with the Host University of Salamanca and the Partner Vienna University of Technology, Austria.

His research interests include the general area of statistical signal processing, information fusion, and distributed computing, with particular interest in novel model-free solutions (by means of data clustering, fitting and mining) for realistic multisensor multiobject detection, tracking and forecasting.

Juan M. Corchado received the Ph.D. degree in computer sciences from the University of Salamanca, Salamanca, Spain, in 1998 and the Ph.D. degree in artificial intelligence from the University of the West of Scotland, Paisley, U.K., in 2000.

He is the Vice President for Research and Technology Transfer since December 2013 and a Full Professor with Chair at the University of Salamanca. He is also the Director of the Science Park of the University and Director of the Doctoral School of the University. Prior to this, he had been twice elected as a Dean of the School of Science. He has been a research collaborator with the Plymouth Marine Laboratory, U.K. since 1993 and been a Visiting Professor in Osaka Institute of Technology, Japan since January 2015. He is the Director of the BISITE (Bioinformatic, Intelligent Systems and Educational Technology) Research Group, which he created in 2000, the President of the IEEE Systems, Man and Cybernetics Spanish Chapter. He is the coauthor of more than 300 books, book chapters, papers, technical reports, etc. and holds 24 patents and intellectual properties, most of which are related to bioinformatics, artificial intelligence and intelligent system. He has served as the President of the Organizing or Scientific Committee of more than 20 international conferences, such as SAC, DCAI, and PAAMS.

Javier Prieto received the telecommunication engineer degree, the marketing research and techniques degree, and the Ph.D. degree in information and communication technologies and the Extraordinary Performance Award for Doctorate Studies from the University of Valladolid, Valladolid, Spain, in 2008, 2010, and 2012, respectively.

Since 2007, he has been working in different public and private research centers, such as the University of Valladolid, and as a Visiting Researcher in the Massachusetts Institute of Technology, USA. He currently serves as a Postdoctoral Researcher at StageMotion under a Torres Quevedo grant and as an Assistant Lecturer at the University of Salamanca, where he is member of the BISITE research group. He has authored more than 40 academic papers and 2 Spanish patents, and participated in more than 20 research projects. His research interests include artificial intelligence, localization and navigation technologies, and Bayesian inference techniques, for improving social welfare and environmental sustainability. He serves as an Editor for the IEEE Communications Letters and for the *Wireless Communications and Mobile Computing* journal, as a Guest Editor for the *International Journal of Distributed Sensor Networks* in the special issue on "Energy and Spectrum Efficient Wireless Sensor Networks," and as a Reviewer in numerous international journals.

QUERIES

Q1. Authors: When accessing and uploading your corrections at the Author Gateway, please note we cannot accept new source files as corrections for your paper. Do not send new Latex, Word, or PDF files, as we cannot simply “overwrite” your paper. Please submit your corrections as an annotated PDF or as clearly written list of corrections, with location in paper. You can also upload revised graphics to the Author Gateway.

Q2. Authors: If you need an invoice or have any other billing questions, please contact reprints@ieee.org as they handle these requests.

Q3. Author: Please provide associate editor name in the place of xxx.

Convergence of Distributed Flooding and Its Application for Distributed Bayesian Filtering

Tiancheng Li, Juan M. Corchado, *Member, IEEE*, and Javier Prieto, *Member, IEEE*

***Abstract*—Distributed flooding is a fundamental information sharing method to get network consensus via peer-to-peer communication. However, a unified consensus-oriented formulation of the algorithm and its convergence performance are not yet explicitly available in the literature. To fill this void in this paper, set-theoretic flooding rules are defined by encapsulating the information of interest in finite sets (one set per node), namely distributed set-theoretic information flooding (DSIF). This leads to a new type of consensus referred to as "collecting consensus," which aims to ensure that all nodes get the same information. Convergence and optimality analyses are provided based on a consistent measure of the degree of consensus of the network. Compared with the prevailing averaging consensus, the proposed DSIF protocol benefits from avoiding repeated use of any information and offering the highest converging efficiency for network consensus while being exposed to increasing node-storage requirements against communication iterations and higher communication load. The protocol has been advocated for distributed nonlinear Bayesian filtering, where each node operates a separate particle filter, and the collecting consensus is pursued on the sensor data alone or jointly with intermediate local estimates. Simulations are provided in detail to demonstrate the theoretical findings.**

***Index Terms*—Consensus, diffusion, distributed tracking, particle filter, sensor network.**

I. Introduction

DSTRIBUTED computation has gained immense attention in the past decade, accompanying the rapid development and popularity of wireless sensor networks. In the successful networking operation, it is often of high interest that each node iteratively shares information with its intermediate neighbors (namely peer-to-peer communication) and consequently the entire network tends to reach a global alignment [1]/consensus [2]–[4] (to a certain degree). Compared to the centralized networking solutions based on a fusion center, distributed networking offers several advantages regarding scalability to adding or removing nodes, immunity to node failure, and dynamic adaptability to network topology changes.

However, there is a significant conflict between the degree of consensus (DoC) and communication requirement, as a higher DoC requires more communication, either more communicating iterations or higher communicating bandwidth which are limited by real time implementation and the communicating affordability of the nodes, respectively. Therefore, it is of paramount significance to seek a good balance for the trade-off so that the network achieves a satisfactory consensus in real-time and with affordable communication costs, which forms the majority of the research in the literature.

One of the most fundamental solutions for network information sharing is the flooding carried out in a distributed manner, by which all nodes synchronously broadcast their information to neighbors from the near to the distant. This protocol is well known in a few areas such as the communications [6]. Given that the network is strongly connected, it is able to achieve complete consensus (CC, i.e., all nodes have exactly the same set of information) after a certain number of iterations of peer-to-peer communication. This is quite appealing in theory but poses crucial challenges to the storage and communication requirement for large networks in practice. In fact, flooding has rarely been investigated in literature for distributed filtering (except for a few works, e.g., [7], [8]) even for small networks, regardless of its fast convergence (to be explicitly demonstrated in this paper) and ease of implementation. To note, there is one work that also refers to distributed flooding [9] which transmits the information of one single node over the network for routing. It is very different from the distributed protocol we consider here.

To date, a unified consensus-oriented formulation of the flooding algorithm and its convergence analysis are still missing in the literature. On one hand, there are cases in which DoC is required in the first priority while the sensor nodes have sufficient node-storage and communicating affordability to do so, for which flooding or even CC is simply preferable. On the other hand, instead of CC, it is more desired to perform flooding in a fewer, affordable, number of iterations for real time realization. What then can be expected and how can the number of iterations be properly determined?

To address the void and to answer the questions above, this paper aims to contribute from three aspects:

1) Formulate the flooding algorithm by encapsulating the information of interest in a finite set at each node and define set-theoretic flooding rules, namely *distributed set-theoretic information flooding* (DSIF), raising a new type of consensus termed *collecting consensus*.
2) Analysis the convergence and optimality of the DSIF protocol, based on a novel, consistent, metric of the DoC.

Manuscript received March 11, 2016; revised July 4, 2016 and October 4, 2016; accepted November 5, 2016. Date of publication; date of current version. This work was supported in part by the Marie Skodowska-Curie Individual Fellowship (H2020-MSCA-IF-2015) and in part by the Spanish Ministry of Economy and Competitiveness under Torres Quevedo Grant PTQ-14-06928. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Vincenzo Matta.

T. Li is with the University of Salamanca, BISITE Research Group, Salamanca 37002, Spain (e-mail: t.c.li@usal.es).

J. M. Corchado is with the University of Salamanca, BISITE Research Group, Salamanca 37002, Spain, and also with the Osaka Institute of Technology, Osaka, Japan (e-mail: corchado@usal.es).

J. Prieto is with the University of Salamanca, BISITE Research Group, Salamanca 37002, Spain, and also with the StageMotion, R&D Department, Palencia 34005, Spain (e-mail: javierp@usal.es).

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

Digital Object Identifier 10.1109/TSIPN.2016.2631944

It is shown that DSIF enjoys the highest efficiency for network consensus among all distributed peer-to-peer communication schemes while suffering from heavier storage and communicational costs.

3) Show how the proposed DSIF scheme can be applied for and can benefit distributed Bayesian filtering, in which each node runs a separate particle filter (PF). Local PFs share sensor data alone or jointly with intermediate posteriors via DSIF. The latter usually needs to be parameterized in order to to reduce the communication cost. The gain and loss to do so are analyzed and demonstrated in simulations.

The reminder of this paper is organized as follows. Notations and definitions regarding networking and three prevailing distributed information sharing protocols are given in Section II. As the main theoretical contribution, the collecting consensus-oriented DSIF is formulated in Section III with an analysis of its convergence and optimality. Section IV shows how to apply DSIF for distributed PF (DPF), with a brief literature review for DPF also given. Simulations are given in Section V and we conclude in Section VI.

II. Notation, Definition and Background

A. Notation

The network topology is represented by a directed graph $G = (V, E)$ with the set of nodes $V = \{1, 2, \cdots, N\}$ and the set of edges $E \subseteq V \times V$. In the directed graph, any edge is denoted by an ordered pair of nodes $(i, j) \in E$, which means node j is directly reachable from node i, where i is called the in-neighbor of j while j is the out-neighbor of i. For any $j \in V$, denote $N_j := \{i \in V | (i, j) \in E, i \neq j\}$, which is the set of all the in-neighbors of node j excluding node j itself. Undirected graph is a special type of directed graph where for any $(i, j) \in E$, we must have $(j, i) \in E$. If there exists a sequence of connected edges as follows

$$\{(i, \cdot), \cdots, (\cdot, j)\} \subseteq E \tag{1}$$

This sequence of edges is called a path from node i to node j (denoted as $Path_{i-\ldots-j}$) and node j is said to be "reachable" from node i. A digraph is said to be "strongly connected" (SC) if any node is reachable from all the other nodes, which is the digraph that can reach CC. For undirected graphs, SC is the same as connectivity [3].

The length of a path is given by the number of edges on that path. The length of the shortest path (perhaps, not unique) from node i to node j is called the distance from node i to node j, denoted as $D(i - j)$. Particularly, $D(i - i) = 0$ and if $(i, j) \in E$, $D(i - j) = 1$. We denote the set of all the nodes that are of distance $t \in \mathbb{N} = \{0, 1, 2, ...\}$ to j as $N_j(t)$ and that of distance $t \in \mathbb{N}$ or smaller to j as $N_j(\leq t)$, namely $N_j(t) := \{n \in V | D(n - j) = t\}, N_j(\leq t) := \{n \in V | D(n - j) \leq t\}$. Obviously, we have $N_j(1) = N_j, N_j(0) = j$.

The largest distance between any two nodes, denoted as D_m, is called the diameter of the graph which is given by

$$D_m = \max_{i,j \in V} D(i - j) \tag{2}$$

Clearly, D_m only exist in SC networks. For any SC networks of at least two nodes, we have $D_m \in [1, N - 1]$, where the left bound corresponds to the fully connected network in which all nodes are in-neighbors of the others, while the right bound corresponds to the weakest connected network where all nodes are on a single chain in order, each having no more than two intermediate neighbors.

With particular regard to the distributed filtering problem, we assume that each node has independent abilities for: (i) filtering calculation, (ii) sensing to collect observations and (iii) communicating to neighbors. The communication is carried out in recursive iterations between neighboring nodes, each iteration consisting of sending no more than one data packet and receiving no more than one data packet. In addition, we need to clarify the following two definitions.

Definition 1 (*real time communication*): Communication is fully carried out between two successive observations and causes no sensor data missing or time-delay to the filter.

Definition 2 (*communication bandwidth*): The maximum size of the data packet that one node can send to or receive from its neighbor per communication.

B. Averaging/Maximum/Minimum Consensus

Here we assume that each node has a local scalar value, referred to as *state*, and it is of interest to compute the average of these values. An averaging consensus algorithm [2], [3] of zero communication time-delay is to reach an agreement regarding the state x_i each node has with local adapting dynamics $u_i(t)$, which can be written in discrete-time as

$$x_i(t + 1) = x_i(t) + u_i(t) \tag{3}$$

where $t \in \mathbb{N}$ denotes the communication iteration, $x_i(0)$ and $x_i(t)$ denote the initial and updated state of node i after iteration t, respectively.

In most of the time, the dynamics $u_i(t)$ is defined as

$$u_i(t) = \sum_{j \in N_i} \omega_{j \to i} \big(x_j(t) - x_i(t)\big) \tag{4}$$

where $\omega_{j \to i}$ is neighboring weight from node j to node i [2]–[4].

The complete convergence of *averaging consensus:* at iteration t states that, for any $i, j \in V$,

$$x_i(t) = x_j(t) \tag{5}$$

and asymptotically convergence (in the sense that $t \to \infty$)

$$\|x_i(t) - x_j(t)\| \leq \epsilon \tag{6}$$

where $\|x - y\|$ is a measure of the discrepancy between x and y, and ϵ is an error bound or margin [10].

In contrast to the average, one might be only interested in the maximum/minimum state, namely maximum/minimum consensus, which defines the iteration as

$$x_i(t + 1) = \max / \min \big\{x_i(t), \{x_j(t)\}_{j \in N_i}\big\} \tag{7}$$

The averaging consensus was well investigated in the community of control and systems [2]–[5]. One major concern is with the convergence, for which the spectral properties of the graph Laplacian play a crucial role [2], [3], [10]-[12]. The second concern is raised by the information correlation/dependence among neighbors (especially when they own in part the same information). To account for this, fusion weights will be assigned to coordinate the nodes, e.g., covariance intersection [14],[15], which has inspired many strategies. In the absence of clear information about the correlation or relative quality of the information among nodes, two weighting methods have been proposed: one is to weight all nodes equally [16] and the other is to weight them according to the size of their neighborhood (named Metropolis weights [17]).

Particularly for the tracking problem, the sensors may updates their observation frequently, preventing sufficient peer-to-peer communicating to get the network converge. This necessitates limiting the number of communicating iterations to gain a trade-off or compromise between a high DoC and little missing or time-delay of sensor data.

C. Gossip and Diffusion

To save communication, one alternative is to apply gossip to randomly choose fewer neighbors at each time (rather than to all neighbors) for averaging. It turns out that under mild conditions this process converges over time asymptotically [18]. Gossip based distributed filtering has been reported in, e.g., [19],[20]. However, gossip experiences the same problem as inefficient/repeated computations, for example, the same set (or largely similar set) of nodes repeatedly fusing their information at different points in time.

As another alternative, diffusion [22]–[24] performs only one iteration of peer-to-peer communication (i.e., the sensing and consensus time scales are the same), avoiding the problem of repeated use of any information. Based on it, the distributed Kalman filter (DKF) [23],[24] does not only share sensor data, but also local intermediate estimates through a diffusion update step. By this, the one-iteration-only communication is actually carried out on two types of data: the sensor data for the incremental update, and the estimates for the diffusion update. Hybrid fusion has been previously studied in the network using a fusion center, e.g., [21].

III. Collecting Consensus and DSIF

A. Collecting Consensus

In this paper, we are interested in an information sharing protocol that does not repeatedly use any information, and that will converge to CC in a definite number of iterations. By CC, we mean that all the nodes have exactly the same information of interest. Such a consensus model in which each node aims to collect information from all reachable nodes via the shortest paths is referred to as collecting consensus. Different to averaging consensus, the information from different nodes remains conditionally independent (or more precisely stated, unfused) until the end of the communication, which requires that nodes have sufficient storage allowance.

To perform collecting consensus, the information that needs to be communicated is encapsulated as a set, and the DSIF algorithm defines the information set dynamics (in contrast to (3)) based on the union operation

$$I_i(t+1) = I_i(t) \cup u_i(t) \tag{8}$$

where $I_i(t)$ and $u_i(t)$ denote the existing and new incoming information set of node i at iteration $t \in \mathbb{N}$ respectively, $I_i(0)$ denotes the initial information set at node i with the size $|I_i(0)| = 1, \forall i \in V$, and $u_i(0)$ the initial dynamics.

As a result of CC, all nodes shall have exactly the same information set, i.e., $\forall i \in V, t \geq D_m$,

$$I_i(t) = \bigcup_{j \in V} I_j(0) \tag{9}$$

To this end, the DSIF algorithm consists of two stages:

1) In the starting iteration, each node collects information from all its in-neighbors

$$u_i(0) = \bigcup_{j \in N_i} I_j(0) \tag{10}$$

2) In the following iterations $t \in \mathbb{N}^+ = \{1, 2, \cdots\}$, each node collects the new information that its in-neighbors have received at the preceding iteration

$$u_i(t) = \bigcup_{j \in N_i} \{I_j(t) \setminus I_j(t-1)\} \tag{11}$$

where $A \setminus B$ is the set difference of A and B, namely the set of all elements that are members of A but not of B and when $t \geq D_m$, we will actually have $u_i(t) = \emptyset$.

As shown in (11), the receiving neighbors will sort out the new received data, which they then transmit to their out-neighbors in the next iteration. In this process, the same information may be repeatedly received over edges, leading to information overuse and communication power waste, which is one defect of the naive flooding protocol, named implosion [6]. To avoid this, we define the set-theoretic information flooding rules in the following.

B. Set-Theoretic Flooding Rules

Rule 1 (*data sending*): Each node only sends to its out-neighbors the new information that has never been flooded before, and does so no more than once in each iteration.

Rule 2 (*data accepting*): Each node will not repeatedly take in the same information either from different in-neighbors or from the same node at different iterations, but only accept the information at its first arrival.

For both rules above, the data from each node shall be associated with a unique ID for distinguishing. Given these two rules respected, we will have $\forall i \in V, t \in \mathbb{N}$,

$$|I_i(t)| = |N_i(\leq t)| \tag{12}$$

To combat time-increasing storage requirement and communication load (when $t < D_m$), each element of data (often called a tuple) may be somehow compressed via e.g., dimension reduction [25] and polynomial encoding [13], under the premise that little or even no information would be lost and the data from different nodes remain conditionally independent.

It is worth noting that in the case of maximum or minimum consensus, there is neither a problem of information set-size growing nor information overuse as the fusion result is always a single maximum or minimum value.

C. Convergence and Optimality of DSIF

To gain insights of the convergence of the proposed DSIF scheme, we need a metric to measure the DoC for collecting consensus, for which we propose a metric based on the size of the information set as follows.

Definition 3 (*DoC*): The DoC, denoted as C^o, of a network with N nodes, is defined as follows

$$C^o(t) = \frac{\sum_{i=1}^{N} |I_i(t)| - N}{N(N-1)} \tag{13}$$

where $t \in \mathbb{N}$ denotes the number of DSIF iterations that has been performed and in the following we limit it to $t \leq D_m$. On the DoC, we have the following theorem, which states the convergence property of the DSIF protocol.

Theorem 1: $0 \leq C^o(t_1) < C^o(t_2) \leq 1, \forall 0 \leq t_1 < t_2 \leq D_m$.

Proof: Before performing DSIF, each node has its original one unit of data, i.e., $|I_i(0)| = 1, \forall i \in V$. That gives $C^o(0) = 0$. After $t \geq D_m$ DSIF iterations, CC will be reached as all nodes will have the same information, i.e., $|I_i(t)| = N, \forall i \in V$. Furthermore, we have the following two straightforward Claims (for which we omit any proof):

Claim 1: As stated by (12), the size of the information set owned by sensor $i \in V$ will not be reduced during flooding except that data fusion or removal is taken, i.e., $|I_i(t_1)| \leq |I_i(t_2)|$ for any $0 \leq t_1 < t_2$.

Claim 2: Supposing two nodes g and q are of distance D_m (namely $D(g-q) = D_m$), for any $0 < t \leq D_m$, there must exist at least one node $j \in N_q(t)$ on $Path_{g-\ldots-j-\ldots-q}$ satisfying $D(j-q) = t$ whose information will arrive to node q exactly at iteration t and then, we have $|I_q(t-1)| + 1 \leq |I_q(t)|$.

From these two claims, we may conclude that $C^o(t_1) < C^o(t_2), \forall 0 \leq t_1 < t_2 \leq D_m$, to accomplish the proof. ∎

Theorem 2: In the sense of DoC as given in (13), the proposed DSIF achieves the highest converging efficiency among all distributed peer-to-peer communication schemes.

Proof: From the definition of the distance between nodes and the DSIF peer-to-peer communication rules, we have two additional straightforward Claims:

Claim 3: All nodes whose information can reach node i in t iterations of peer-to-peer communication belong to $N_i(\leq t)$.

Claim 4: All nodes $q \in N_i(\leq t)$ will surely flood their information to node i in t DSIF iterations.

A combination of Claims 3 and 4 indicates that the DSIF will gain the largest possible $|I_i(t)|$ for any $i \in V$ and $t \in \mathbb{N}$ as claimed, which entails the converging optimality. ∎

D. Trade-off between DoC and Number of Iterations

For a given network topology, the DoC is uniquely determined by the number of DSIF iterations. In turn, one can also determine the required number of iterations for a desired DoC, e.g., $T_c = 0.5$. That is, the DSIF stops at iteration t once

$$C^o(t) \geq T_c \tag{14}$$

Algorithm 1: DSIF operations at node i for DoC T_c.

INITIALIZATION:
$t \leftarrow 1; C^o(0) \leftarrow 0$
$I_i(t) = I_i(t-1) \cup \bigcup_{j \in N_i} I_j(t-1)$
RECURSIVE FLOODING ITERATION:
While $C^o(t) < T_c$
  $t \leftarrow t+1$
  $I_i(t) \leftarrow I_i(t-1) \cup \bigcup_{j \in N_i} \{I_j(t-1) \setminus I_j(t-2)\}$
  $C^o(t) \leftarrow \frac{\sum_{i=1}^{N} |I_i(t)| - N}{N(N-1)}$
End while
Return: $I_i(t)$

Algorithm 1 summarizes the communicating operations that need to be performed on node i for a given DoC T_c.

For a constant network of a known topology, the minimum number of iterations can be determined a priori by (14). However for time-varying dynamic networks, it needs to be calculated online via a consensus algorithm. To facilitate the use in time-varying networks without burdening any consensus procedures, we define the local DoC metric as follows:

Definition 4 (*Local DoC*): The DoC of node $i \in V$, denoted as C_i^o, after t DSIF iterations, is given as

$$C_i^o(t) = \frac{|I_i(t)| - 1}{N_k - 1} \tag{15}$$

where N_k is number of nodes in the network at time k, which needs to be estimated if unknown. From here we derive the following theorem as the local-node version of Theorem 1.

Theorem 3: $0 \leq C_i^o(t_1) \leq C_i^o(t_2) \leq 1, \forall 0 \leq t_1 < t_2 \leq D_m$

Proof: This theorem states the same content as Claim 1. As a key difference to Theorem 1, the equality $C_i^o(t_1) = C_i^o(t_2)$ holds when and only when the number of nodes that are of distance t_1 to node $i \in V$ is the same as that of distance t_2, i.e., $|N_i(t_1)| = |N_i(t_2)|$. ∎

Setting a threshold, e.g., $T_c = 0.5$ which can be the same or different for different nodes, on the desired local DoC, the consensus updating at node i may stop at iteration t once

$$C_i^o(t) \geq T_c \tag{16}$$

Furthermore, based on DoC we can define the convergence speed (CoS), either globally or locally, to measure the change of the size of the information set at each iteration, which also indicates the local real-time communication bandwidth.

Definition 5 (*CoS*): At iteration $t \in \mathbb{N}$, the global CoS, denoted as C^s, is defined as,

$$C^s(t) = C^o(t) - C^o(t-1) \tag{17}$$

and the local CoS of node $i \in V$ is defined as

$$C_i^s(t) = C_i^o(t) - C_i^o(t-1) \tag{18}$$

Theorem 4: $C^s(t) > 0, C_i^s(t) \geq 0, \forall 1 \leq t \leq D_m$

Proof: The theorem is immediate from Theorems 1 and 3 as for any $1 \leq t \leq D_m, i \in V$, we have $C^s(t) = C^o(t) - C^o(t-1) > 0$ from Theorem 1, and $C_i^s(t) = C_i^o(t) - C_i^o(t-1) \geq 0$ from Theorem 3. ∎

Theorems 1, 3 and 4 entail an appealing property of the DSIF protocol which will not only converge definitively, but also has a guaranteed converging speed that is globally positive and locally non-negative everywhere and at any iteration until CC is reached. We refer to this as *strong convergence*. It, however, also indicates a (non-negative) increasing storage requirement against communicating iterations. As an alternative to (16), we can build the predetermined threshold on the local CoS, e.g., $T_s = 0.1$, then the minimum number of iterations t needs to satisfy

$$C_i^s(t) \leq T_s \tag{19}$$

But it is critical to note that we do not have any monotonicity on the CoS, e.g., $C^s(t_2) \leq C^s(t_1)$ or $C_i^s(t_2) \leq C_i^s(t_1)$ for $1 \leq t_1 < t_2 \leq D_m$. Therefore, the CoS at iteration t does not say anything of the CoS at iteration $t+1$.

E. Comparison and Practical Consideration

Both metrics of DoC and CoS are clearly defined and easier to calculate than the one proposed for averaging consensus, e.g., convergence rate [10]–[12], steady-state mean-square deviation [4] or disagreement vector [2], [3]. As indicated by Theorem 2, no peer-to-peer communication protocols converge faster than DSIF in terms of DoC. This superiority, however, is achieved at the expense of higher node storage requirements and heavier communication bandwidths. If the size of the data set at one node exceeds its communication bandwidth, multiple iterations will then be needed for that data set, otherwise data fusion is required to control the data size. In the former case, the required number of iterations will increase, while in the latter case the information completeness or independence may not be kept. However, we will not address this issue further here, which is quite problem dependent. In brief, we have the following remark on the respective advantages of averaging consensus, diffusion and collecting consensus.

Remark 1: The averaging consensus takes the lowest communicating bandwidth (always one unit of data) but more iterations to reach any DoCs while the diffusion severely limits the number of iterations (to one only) which may insufficiently use the communication affordability (i.e., more iterations are actually allowed in real time communication). In contrast, the proposed DSIF protocol aims to get the best possible consensus in an real-time-allowed number of iterations, which is therefore particularly suited to small and moderate networks for which the nodes have sufficient storage and communicating power. A means to facilitate its use in large networks is to selectively apply data fusion such as averaging in every several flooding iterations in order to control the data-set size. This will lead to a hybrid protocol that iterates between flooding and averaging consensus, to gain a balance between benefiting from high communication efficiency and suffering from information overuse and slower convergence.

IV. Distributed Bayesian Filtering Using DSIF

A. State-of-the-art DPF Protocols

Before presenting our DPF framework based on DSIF, a brief revisit of the PF algorithm and existing DPF protocols is given below. Suppose that at time k, the local (marginal) posterior at sensor i is represented by a local PF

$$p(\mathbf{x}_k|\mathbf{z}_{i,1:k}) \approx \sum_{m=1}^{M_{i,k}} w_{i,k}^{(m)} \delta\big(\mathbf{x}_k - \mathbf{x}_{i,k}^{(m)}\big) \tag{20}$$

where $\delta(\mathbf{x}-\mathbf{y})$ is the Dirac delta impulse, which equals to one if $\mathbf{x}=\mathbf{y}$ and to zero otherwise, $\mathbf{x}_k$ is the true state vector, $\mathbf{z}_{i,1:k}$ is the observation serial, $\mathbf{x}_{i,k}^{(m)}$ and $w_{i,k}^{(m)}$ are the state and normalized weight of the mth particle respectively, $M_{i,k}$ is the total number of particles at filtering time k.

The essence of the PF is to assess how well each particle conforms to the state model and explains the observations, using this assessment to generate a weighted sample approximation to the Bayesian posterior, and thereby form sub-optimal state estimates. Given local measurement $\mathbf{z}_{i,k}, i \in V$, the weights of the particles are evaluated over time based on the sequential importance sampling (SIS) principle as

$$w_{i,k}^{(m)} \propto w_{i,k-1}^{(m)} \frac{p(\mathbf{z}_{i,k}|\mathbf{x}_{i,k}^{(m)})p(\mathbf{x}_{i,k}^{(m)}|\mathbf{x}_{i,k-1}^{(m)})}{\pi\big(\mathbf{x}_{i,k}^{(m)}|\mathbf{x}_{i,k-1}^{(m)},\mathbf{z}_{i,1:k}\big)} \tag{21}$$

where $\pi(\cdot)$ is a proposal to generate particles, and in general its design shall take into account both the newest measurement $z_{i,k}$ and the prior in order to best match the posterior; see e.g. [19], [28], [31]. The use of the observation in the sampling proposal design is particularly helpful (and even necessary for avoiding sample degeneracy) when the observation is very accurate. However, caution should be exercised here since the repeated use of the observation (both for proposal design and in likelihood calculation) may not benefit the filter when the observation suffers from significant noise [42].

In addition to SIS, resampling is usually required to reduce the weight variance when it exceeds a certain threshold, so that all particles will have equal or approximate weights while the posterior distribution can be the best maintained [31], [32]. This has often been referred to as sampling importance resampling (SIR), which is the core of the majority of existing PFs. We assume the reader is familiar with the centralized PF and so limit ourselves hereafter to the distributed implementation, in which local nodes carry out PF calculations in parallel and meanwhile share information with their neighbors to assist their filters. For this, a variety of information sharing protocols have been proposed, which can be classified as follows:

1) *Sequential information passing*: Information transmits in a sequential, predefined manner from a node to one of its neighboring nodes via a cyclic path until the entire network is traversed [43]. The sequential realm is sensitive to the mobility and failure of nodes/edges and is time-consuming.
2) *Flooding*: As addressed, the flooding protocol provides the fastest albeit communication-intensive way to spread information over the network [7], [8], but, neither any clue to determine the number of communication iterations in order to compromise real time realization and DoC nor any convergence results has been shown.

3) *Averaging consensus*. There is a large body of work concerning averaging consensus-based distributed filtering. The data transmitted between neighboring nodes can be posterior statistics in the form of Gaussian component [33] /GM [29]–[30] or generalized probability densities [36]–[37], likelihood [26]–[28], particle set [34]–[35] or raw observations [38]. Excellent surveys are also available such as a taxonomy of DPFs [39], a comparison of several belief consensus algorithms [40] and a recent survey of convergence and error propagation of DPFs [28]. In summary, complete information sharing affords better accuracy but has higher communication requirements, such as [34]–[35] that exchange all particles. Parameter approximation [26]–[33] or random gossip [19]–[20] can significantly reduce the communication cost, but may lead to a deterioration in the filter performance.

4) *Diffusion*: The diffusion scheme addressed in Section II.C also provides a competitive alternative to the averaging consensus for DPF [7]-[8], [41].

We note that the sensor data can be either simple (e.g., range, bearing) or complex (e.g., image data). To avoid distracting from the key contribution of this paper on collecting consensus and the DSIF protocol, we only consider the former case for simplicity. For the latter case, one may consider compressing the sensor data, e.g., [25]-[26], [12]-[13] or transmitting the low-dimensional likelihood for replacement [26]–[27]. At the current stage, we have not considered complicated network issues such as communication constraints, e.g., [45]–[46], and asynchronous sensing, e.g., [47]–[48]. However, we note all of these issues are valuable to be investigated on the base of the proposed DSIF protocol.

B. DSIF on Sensor Data and on Local Posterior

In the proposed DPF framework, the DSIF scheme will be applied on the sensor data alone or jointly on local posteriors. In the latter, we propose parameterizing the posterior to save communication. Since a vast number of random numbers are required by the PF, it is communication intensive to run consensus on them, and it is not our intention to do so.

First, DSIF is implemented on the sensor data including the target-observations (and uncertainties) associated with the sensor ID, all as one unit. To note, the sensor position is often required for likelihood calculation and therefore can serve as the unique sensor ID for distinguishing. Then, the resultant consensus on sensor data with sensor profiles given a priori, is equivalent to collecting consensus on the likelihood which is required for PF updating. A likelihood function contains the information of both the sensor data and the sensor profile in a more compact manner. But for simplicity of understanding, we keep addressing consensus on sensor data.

The filtering posteriors obtained at different nodes, referred to as local posteriors, will be different, even if CC is reached on sensor data over the network where the difference attributes to the different random numbers. If DoC is low on sensor data, the difference between local posteriors will be relatively significant. As such, we may apply the second DSIF scheme to fuse local posteriors among neighbors as well as to get the local LMS (least mean squares) estimate; we refer to this step as diffusion, in parallel to [24]. By this, each node aims to improve their local estimate with regard to their neighbors' posterior. However, parameter approximation of local posteriors, typically via Gaussian or GM approximation, is needed (otherwise massive communication will be triggered if the complete posterior is communicated by transmitting the entire particle set), which will in turn introduce approximation errors to the posterior. This trade-off is much problem-dependent and will determine whether the second DSIF is worthwhile.

The operations that need to be conducted on each sensor in the proposed distributed PF is summarized in Algorithm 2. In it, steps 1-a and 1-b are independent of each other and therefore can be carried out in either order or in parallel. Sensor data DSIF and posterior DSIF have been implemented t_1 and t_2 iterations respectively, where t_1 and t_2 are not necessarily equal but are determined for respective desired or the largest affordable DoCs as addressed in Section III. They show complementary features and resemble the Incremental and Diffusion updates of the diffusion-based DKF [24]. But, there are obvious differences:

1) Our framework is developed for nonlinear models which releases the requirement of linear system functions and even Gaussian assumption of the posterior;
2) Our consensus protocol does not limit information sharing between neighbors to one iteration only but instead, the DoC will be pursued as much as the real time communication allows;
3) Our diffusion update (Step 5 in Algorithm 2) is an optional step, which is advocated for re-setting local posteriors only when local posteriors are significantly different (as a consequence of a low DoC on the sensor data achieved in the first DSIF implementation). When the difference between local posteriors is insignificant (because of a high DoC achieved on the sensor data), there will be less need to further fuse them and so it may be better not to diffuse local posteriors since the errors introduced due to parameterization can be more significant than the benefit. This is a critical point. We will demonstrate this in detail through simulations in Section V. In addition, we provide two easy-to-implement diffusion choices.
4) We point out that the proposed two DSIF procedures can be performed jointly, although this may not reduce the communication load and the storage requirement in total; see the following Remark 2.

Remark 2: Two DSIF implementations regarding the sensor data and the local filter estimates form the starting step and the end step of each filtering iteration, respectively. In the time series, they are adjacent. Therefore, they may be combined in one joint consensus scheme at some stages (which however does not necessarily indicate that $t_1 = t_2$), i.e., the local estimates obtained at filtering time k can be combined with sensor data received at time $k+1$ as one unit of data, both sharing the same node ID for DSIF. Then, the initial information set at node $i \in V$ can be defined as

$$I_{i,k}(0) := \{\hat{\mathbf{x}}_{i,k-1}, P_{i,k-1}\} \cup \mathbf{z}_{i,k} \tag{22}$$

Algorithm 2: Distributed PF calculation executed on node i.

Step 1-a Filter prediction: Propagate the particles $\mathbf{x}_{i,k-1}^{(m)}$ to $\mathbf{x}_{i,k}^{(m)}$ for all $m = 1, \cdots, M_{i,k}$, according to the proposal function $\pi\big(\mathbf{x}_{i,k}^{(m)}|\mathbf{x}_{i,k-1}^{(m)}, \mathbf{z}_{i,1:k}\big)$. At $k = 0$, particles are sampled from an initial proposal π_0 instead for filter initialization.

Step 1-b 1st *DSIF*: Perform t_1 DSIF iterations on sensor data as given in Algorithm 1, resulting in a combined measurement set $\mathbf{Z}_{i,k} = \{\mathbf{z}_{j,k}\}_{j\in N_i(\leq t_1)}$. This step is carried out whenever new measurements become available.

Step 2 Filter updating: Re-weight all particles via (21) ($\mathbf{z}_{i,t}$ therein shall be replaced by $\mathbf{Z}_{i,k}$ obtained in the 1st DSIF) and then normalize them as follows

$$w_{i,k}^{(m)} \leftarrow w_{i,k}^{(m)} \Big/ \sum_{m=1}^{M_{i,k}} w_{i,k}^{(m)} \tag{23}$$

Step 3 Estimate extraction: Extract local estimate $\hat{\mathbf{x}}_{i,k}$ and calculate their covariance $P_{i,k}$ from the local random measure $\chi_{i,k} = \big\{\mathbf{x}_{i,k}^{(m)}, w_{i,k}^{(m)}\big\}_{m=1,2,\cdots,M_{i,k}}$ as follows

$$\hat{\mathbf{x}}_{i,k} = \sum_{m=1}^{M_{i,k}} w_{i,k}^{(m)} \mathbf{x}_{i,k}^{(m)} \tag{24}$$

$$P_{i,k} = \sum_{m=1}^{M_{i,k}} w_{i,k}^{(m)} \left(\mathbf{x}_{i,k}^{(m)} - \hat{\mathbf{x}}_{i,k}^{(m)}\right)\left(\mathbf{x}_{i,k}^{(m)} - \hat{\mathbf{x}}_{i,k}^{(m)}\right)^T \tag{25}$$

Step 4 2nd *DSIF*: Perform t_2 DSIF on local estimates obtained in Step 3, resulting in a set of intermediate estimates $\{\hat{\mathbf{x}}_{j,k}, P_{j,k}\}_{j\in N_i(\leq t_2)}$, which will be fused in the LMS sense as follows

$$\hat{\mathbf{x}}_{i,k}^{\mathrm{LMS}} = \sum_{j\in N_i(\leq t_2)} \hat{\mathbf{x}}_{j,k} P_{j,k}^{-1} \tag{26}$$

$$P_{i,k}^{\mathrm{LMS}} = \left(\sum_{j\in N_i(\leq t_2)} P_{j,k}^{-1}\right)^{-1} \tag{27}$$

This also offers the local filter output at sensor i.

Step 5 Diffusion: As an option, the shared filter estimates given in Step 4 can be used to re-set the local PF posterior $\chi_{i,k}$. If so, there are two choices (the second is expected to have a higher approximation accuracy than the first).

1) Re-set $\chi_{i,k}$ as the LMS fused Gaussian distribution, as is done in the Gaussian PF [49].

$$\chi_{i,k} \leftarrow \mathcal{N}\big(\hat{\mathbf{x}}_{i,k}^{\mathrm{LMS}}, P_{i,k}^{\mathrm{LMS}}\big) \tag{28}$$

2) Re-set $\chi_{i,k}$ as the shared GM before performing LMS fusion, as is done in the Gaussian sum PF [50]

$$\chi_{i,k} \leftarrow \sum_{j\in N_i(\leq t_2)} \mathcal{N}\big(\hat{\mathbf{x}}_{j,k}, P_{j,k}\big) \tag{29}$$

Step 6 Resampling: Sample from the updated particle set [32] if the variance of weights exceeds a specified threshold and if Step 5 is not applied. If Step 5 is applied, sample from the diffused Gaussian or GM distribution $\chi_{i,k}$ given by Step 5 to generate a new particle set. Update $k \leftarrow k-1$ and go to the next filtering iteration.

V. Simulations

In this section, we consider tracking a target that moves in the $x-y$ plane by using the proposed DPF based on a constant sensor network earlier appeared in [39] as given in Fig. 1. The network has totally 10 sensors and a diameter $D_m = 4$. The simulation models and parameters are the same to [27]. In specific, we have the initial state as $\mathbf{x}_0 = [4, 0.5, 4, 0.5]^T$. The Markov transition model that governs the target movement of nearly constant velocity is given by

$$\mathbf{x}_k = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \begin{bmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{bmatrix} \mathbf{u}_k \tag{30}$$

where $\mathbf{x}_k = [p_{x,k}, \dot{p}_{x,k}, p_{y,k}, \dot{p}_{y,k}]^T$, $[p_{x,k}, p_{y,k}]^T$ gives the position and $[\dot{p}_{x,k}, \dot{p}_{y,k}]^T$ the velocity, $\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}_2, 0.00035\mathbf{I}_2)$.

The target emits an acoustic or radio signal with a known constant transmit power P_t that can be received by all sensors independently, i.e., the scalar measurement function of sensor i located at $[s_{i,x}, s_{i,y}]^T$ about target $\mathbf{x}_k$ is

$$z_{i,k} = \frac{\propto P_t}{\|[p_{x,k}, p_{y,k}]^T - [s_{i,x}, s_{i,y}]^T\|^\gamma} + v_k \tag{31}$$

where $\propto$ is a constant that depends on several factors such as fast and slow fading, and gains in the transmitter and receiver antennas, γ is the path loss exponent [44], and $v_k \sim \mathcal{N}(0, \sigma_v^2)$ is the measurement noise. In parallel to [27], we set simply $\propto P_t = 10, \gamma = 1, \sigma_v^2 = 0.001$.

When multiple synchronous observations are available, the weight of particles is updated by multiplying the likelihoods given by each available measurement. That is,

$$p\big(Z_{i,k}|x_{i,k}^{(m)}\big) = \prod_{j\in N_i(\leq t)} p\big(z_{j,k}|x_{i,k}^{(m)}\big) \tag{32}$$

where $Z_{i,k} = \{z_{i,k}\}_{j\in N_i(\leq t)}$ is the measurement set at sensor i gained in the first DSIF procedure of total t iterations.

For any sensor $i \in V$, the necessary and sufficient number of iterations, denoted as $D_{m,i}$, to receive the information from all the other sensors can be given by

$$D_{m,i} := \max_{j\in V} D(j-i) \tag{33}$$

We design three groups of simulations in the following three subsections that use the same ground truths to evaluate or compare the following five PF protocols, where the first three are distributed while the last two are centralized. All PFs use the same number of particles ($M = 1000$).

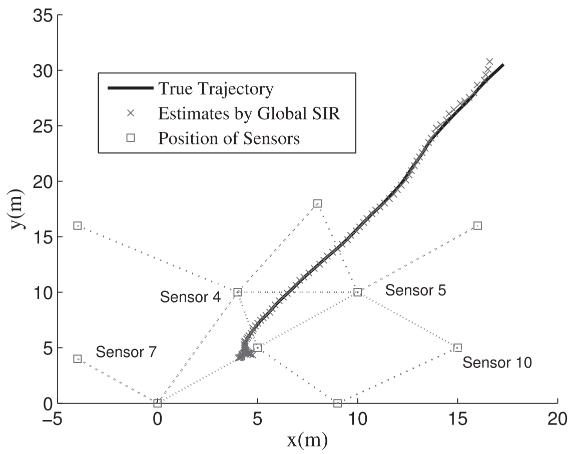


Fig. 1. The topology of the sensor network, the target trajectory and its estimate given a by a global SIR filter in one trial.

1) *C-SIR*: we apply DSIF only on the sensor data, named Consensus without Diffusion (i.e., Steps 4 and 5 are not applied in Algorithm 2). In this case, each local PF is a SIR filter that is free of any Gaussian assumption;
2) *CD-GMPF*: we apply DSIF on both sensor data and local estimates named Consensus with Diffusion (i.e. Steps 4 and 5 are applied in Algorithm 2). In this case, each local PF is a Gaussian sum PF that applies (29) for posterior approximation and fusion;
3) *L-C-SIR*: the Likelihood Consensus-based SIR filter [27] can be viewed as a special case of our C-SIR filter that applies sensor data averaging consensus (for likelihood multiplying) at each iteration. For fast converging, the Metropolis weights strategy [52] is employed for averaging in the L-C-SIR filter;
4) *Local-SIR/GMPF*: local SIR filter or GMPF that does not communicate with each other at all;
5) *Global-SIR*: a centralized SIR filter that is able to access all sensor observations at all times.

To mitigate the problem of sample impoverishment that is often caused by resampling in the SIR filters, the minimum-sampling-variance resampling [32] is applied when and only when the effective sample size is smaller than $M/2$ and if applied, a roughening noise that is equivalent to half of $\mathbf{u}_k$ will be used [31].

To measure the filtering accuracy, we calculate the root mean square error (RMSE) on both the position estimate and the velocity estimate, respectively, as follows

$$\text{RMSEpos}_k = \sqrt{\frac{1}{C}\sum_{c=1}^{C}(x_{k,c}-\hat{x}_{k,c})^2+(y_{k,c}-\hat{y}_{k,c})^2} \quad (34)$$

$$\text{RMSEvel}_k = \sqrt{\frac{1}{C}\sum_{c=1}^{C}(\dot{x}_{k,c}-\hat{\dot{x}}_{k,c})^2+(\dot{y}_{k,c}-\hat{\dot{y}}_{k,c})^2} \quad (35)$$

where $[\hat{x}_{k,c},\hat{y}_{k,c}]^T$ and $[\hat{\dot{x}}_{k,c},\hat{\dot{y}}_{k,c}]^T$ are the position-estimate and velocity-estimate given at filtering time k in trial c, respectively, and $C = 20$ is the total number of MC trials. Further, the average position RMSE is defined as the mean of RMSEpos_k over the entire simulation period of 100 filtering iterations. In each trial, the ground truth is independently generated (for generality). In all trials, the prior distribution of the particle set is initialized around the true state as $\mathcal{N}(x_0, P_0)$, with $P_0 = \text{diag}[2, 0.001, 2, 0.001]^T$.

TABLE I
DoC Achieved at Each DSIF Iteration (Local and Global)

	Sensor 4	Sensor 5	Sensor 7	Sensor 10	Global
$t = 0$	0	0	0	0	0
$t = 1$	5/9	5/9	1/9	2/9	26/90
$t = 2$	1	8/9	3/9	6/9	61/90
$t = 3$	1	1	7/9	8/9	86/90
$t = 4$	1	1	1	1	1

In particular, we will assess the filter performance at four representative sensors, marked in Fig. 1 as sensors 4, 5, 7 and 10. For them, we have $D_{m,4} = 2, D_{m,5} = 3, D_{m,7} = 4, D_{m,10} = 4$. This means that sensor 4 will achieve CC first (after 2 iterations) while sensors 7 and 10 will be the last (after 4 iterations). For different numbers of DSIF iterations, the global and local DoCs are given in Table I. Particularly, for $t = 1$, we have the global DoC determined as

$$C^o(1) = \frac{|E|}{N(N-1)} \quad (36)$$

where $|E|$ is the number of edges; (a, b) and (b, a) are counted as two different edges.

A. Consensus without Diffusion

In this case, each sensor operates a separate SIR filter. Sensors are assumed conditionally independent and use different random numbers. The posteriors obtained by sensors will be different from both each other and the global/local PF, even given that they all reach CC on sensor data.

For different numbers of DSIF iterations from 0 (no consensus at all) to 4 (D_m), the RMSEs of the position and velocity estimation of local C-SIR filters and the global SIR filter are given in Fig. 2.(a)-(e) respectively, corresponding to different DoCs. The average RMSEs over 100 filtering steps against the number of DSIF iterations are given in Fig. 2.(f). The results clearly demonstrate that:

1) A single passive sensor is not capable of delivering good tracking in this problem as the RMSEs given by local PFs are much higher than that provided by DPFs; this necessitates the collaboration of multiple geographically dispersed sensors;
2) The more informative sensor data used, the better the filter performance;
3) The larger DoC, the closer the local PF performance to the centralized PF, i.e., local filters converge to the global filter against iterations as the DoC increases;
4) Once CC is reached, the performance of the local PF is very close to that of the centralized PF (with regard to

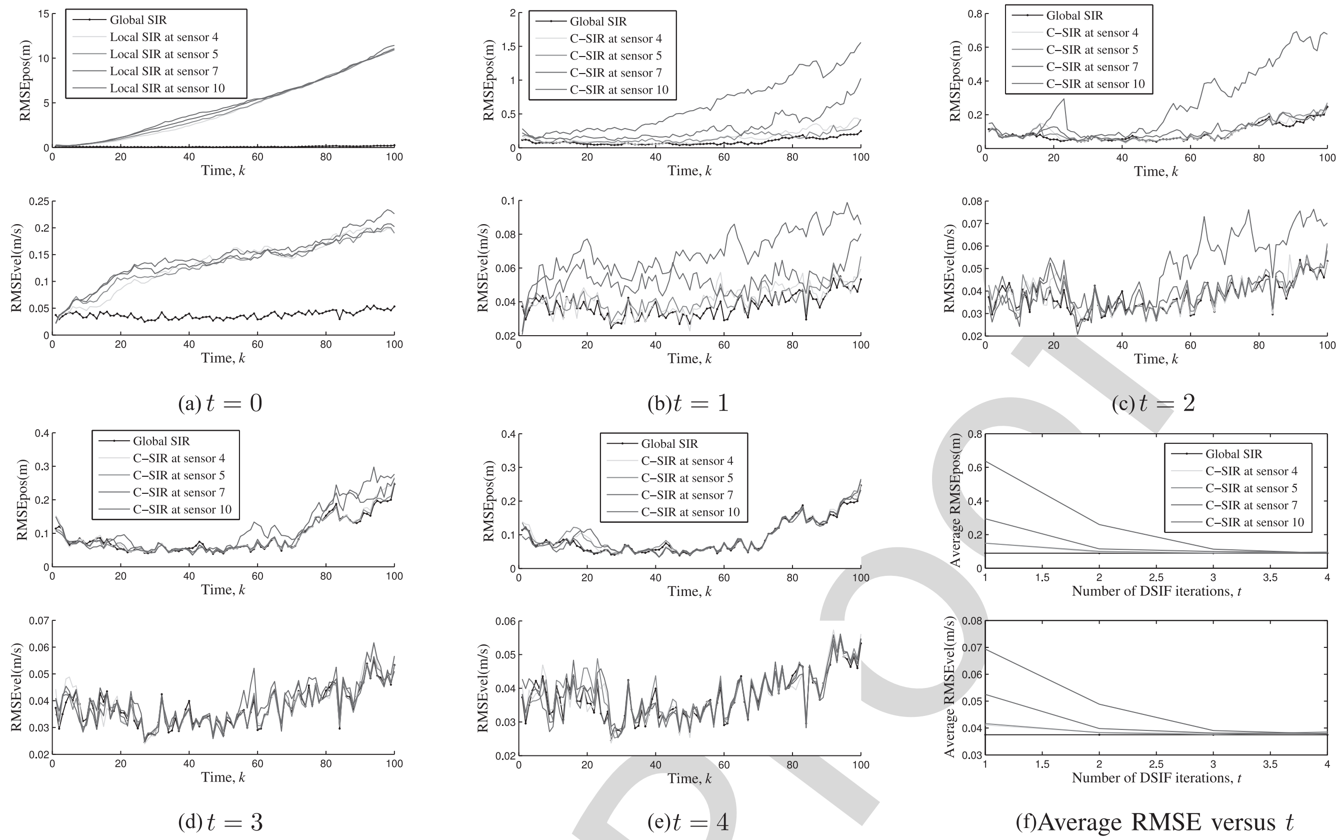


Fig. 2. Position and velocity RMSE of C-SIR filters with different numbers of DSIF iterations, comparing with the global SIR filter.

both position and velocity) but still not the same, since different random numbers are used.

Based on the measure of DoC, we are able to approximately determine how much information divergence different nodes will have and what payoff can be expected if one more or one less iteration of peer-to-peer communication is employed. For example, when the number of iterations is $t = 3$, the global DoC is as high as $86/90$, close to 1, which agrees with the slight difference between Fig. 3(a) and (b). This is a valuable part of the metric of DoC.

B. Consensus with Diffusion

In this case, each sensor runs a separate GMPF. Collecting consensus are applied on both the sensor data and intermediate estimates jointly in a single DSIF procedure (and set $t_1 = t_2$). Because of the GM diffusion of intermediate estimates, the local fused estimates are expected to be closer to each other. If CC is reached, they shall be exactly the same. In parallel to the last simulation, different numbers of DSIF iterations from 0 to 4 are employed to the CD-GMPFs, which are compared with the (centralized) global SIR PF in Fig. 3(a)–(e) respectively. The average RMSEs of these filters against the number of DSIF iterations are given in Fig. 3(f).

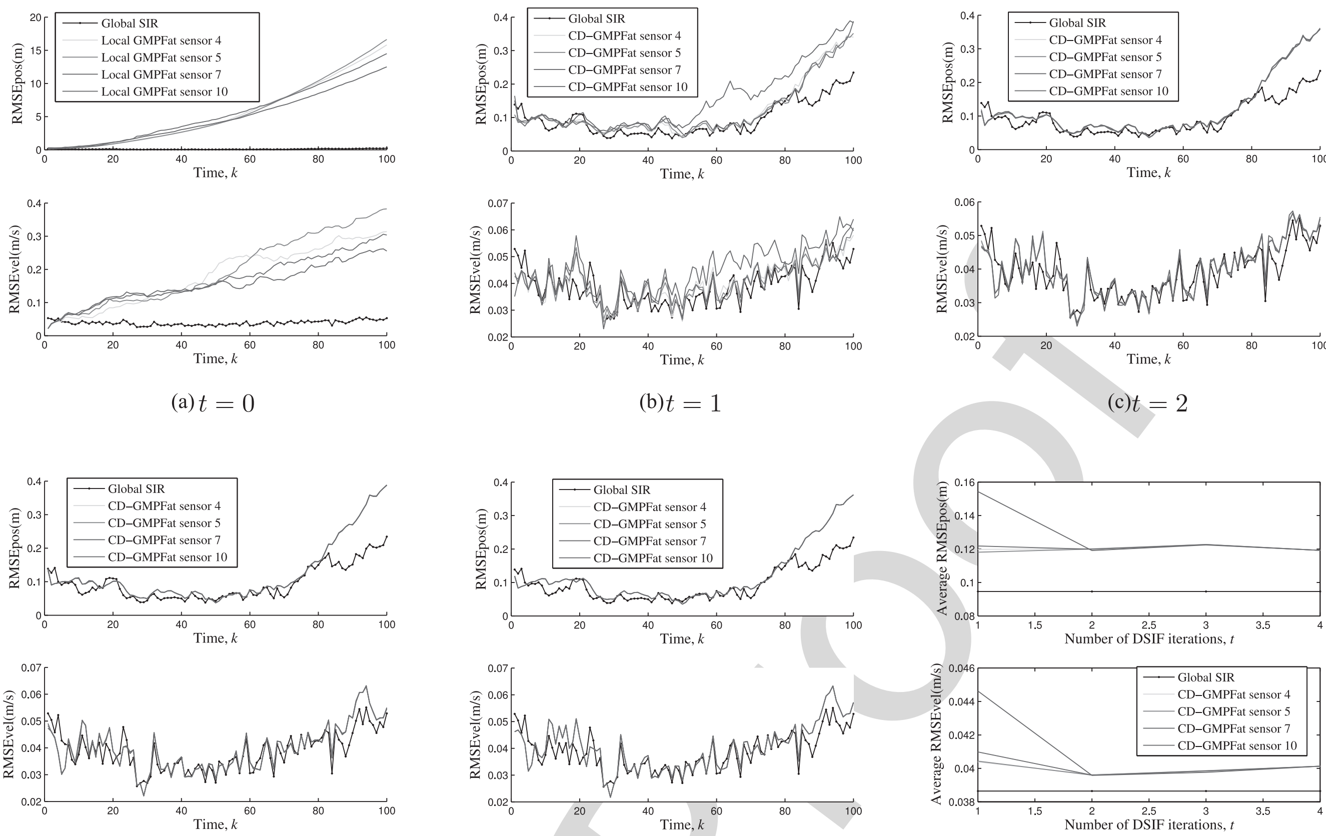
We use the same ground truth (20 MC trials) regarding the trajectories and sensor observations as the last simulation. Compared to the last simulation, we can find that

1) A single passive sensor can still hardly work well when the local SIR filters are replaced by local GMPFs;
2) Given the same number of DSIF iterations $t = 1, 2$, CD-GMPFs perform much better than C-SIR and are much closer to each other; this is because of the second DSIF scheme on the posteriors over the network which enhances the consensus to improve local estimates;
3) Given $t = 3, 4$ iterations, the local CD-GMPFs perform almost the same but different to the global SIR filter;
4) Given CC achieved, the RMSEs of all local GMPFs are exactly the same but are inferior to the global SIR filter, especially at the later stage in this tracking example. Analysis and discussion will be given next.

C. Comparison and Discussion

Finally, we compare both types of DSIF-based DPFs with the L-C-SIR filter [27], [39]. The key difference of the likelihood consensus to DSIF is that each node fuses information intermediately after receiving them and therefore the communication cost is lower, but it is exposed to repeated use of information and slower convergence.

First, for $t = 4$, the average (over all nodes) position RMSEs of the C-SIR, CD-GMPF and the L-C-SIR filters are given in Fig. 4. It shows that the C-SIR filter achieves the closest performance to that of the centralized filter. We further calculate the mean of these average RMSEs for $t = 0$ to 8 and the results are given in Fig. 5. It shows that these consensus protocols can all significantly improve the filter performance as compared to the local filter that applies no consensus and converges against communication iterations.



(a) $t = 0$ (b) $t = 1$ (c) $t = 2$

(d) $t = 3$ (e) $t = 4$ (f) Average RMSE versus t

Fig. 3. Position and velocity RMSE of CD-GMPFs with different numbers of DSIF iterations, comparing to the global SIR filter.

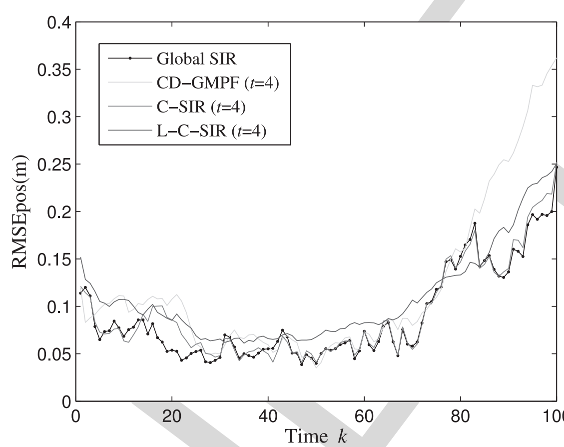


Fig. 4. Position RMSE of different DPFs applying 4 iterations of peer-to-peer communication.

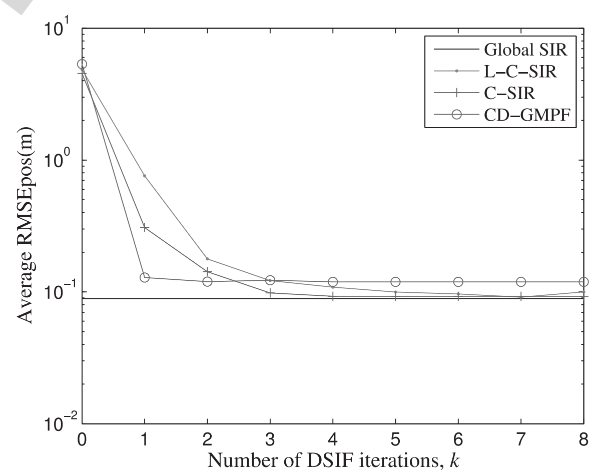


Fig. 5. Average position RMSE of different DPFs over 100 filtering steps against the number of peer-to-peer communication iterations.

Furthermore, we have the following observations, which show more insights of these three types of DPFs:

1) DSIF based C-SIR and CD-GMPF converge faster than the averaging consensus-based L-C-SIR filter at the expense of higher communication cost. CD-GMPF converges the fastest but it suffers from a larger RMSE at the end, all due to its diffusion step that shares information among nodes more thoroughly than without diffusion but also introduce errors;
2) For a relatively small number of iterations that correspond to a low DoC on observation (which may lead to a large discrepancy between local nodes' posteriors), the C-SIR filter is inferior to the CD-GMPF, as shown in Fig. 5 (also told by comparing between Figs. 2(f) and 3(f)). In this case, the diffusion update leads to earlier convergence and better performance for the filter. This is in line with the findings reported in [23];

3) For a large number of iterations that correspond to a high DoC on observation and consequently on posterior (leaving little space to benefit from posterior fusion), the diffusion update of the CD-GMPF is not so preferable; instead, the GM approximation error caused in the diffusion might be more significant than the benefit it can offer, resulting in an overall filter degradation. We must note that if the whole particle sets are transmitted for diffusion without any approximations, and also the dependence between the posteriors are accounted for properly in the diffusion update, it shall always be beneficial in theory regardless of the much greater cost in communication and local fusion calculation.

These results confirm our theoretical prediction and demonstrate further that, both approximation and data fusion during communication can be either beneficial or counterproductive. Generally speaking, parametric approximation can speed up the convergence but also introduces errors. Data fusion such as averaging will reduce communication costs but will also slow down the convergence (primarily because of repeated use of information in data fusion). In practice, we have to contend with a compromise between fast convergence, accurate information sharing and low storage and communication cost. Inspired by these findings, a problem-oriented hybrid protocol that takes the advantages of different approaches while minimizing the side-effects will be valuable.

VI. Conclusion

Flooding is an efficient albeit simple solution for information sharing over networks and is the basis of many other networking protocols. In this paper, we formulated it from a set-theoretic perspective, named distributed set-theoretic information flooding (DSIF). This led to a novel consensus protocol for networking referred to as collecting consensus, which has significant both advantages and disadvantages over averaging consensus and diffusion. We have analyzed the explicit convergence and optimality of DSIF based on a novel metric of DoC (degree of consensus). Practical solutions have been proposed either to determine the minimum number of iterations required for any desired DoC or to calculate the DoC that can be achieved by an actual number of iterations. It has also been noted that to save communication, data fusion (such as averaging) can be employed during flooding, which however may cause repeated information use and slower convergence. This trade-off has been analyzed.

Based on the theoretical results, a distributed particle filter framework is proposed and implemented for nonlinear target tracking which applies DSIF on sensor data alone or jointly with intermediate estimates. Simulations have demonstrated the convergence of the DSIF (faster than averaging consensus), the relationship between the filter performance and the DoC, and the advantage and disadvantage of applying parameterized approximation and data fusion for networking.

Acknowledgment

The authors would like to thank the insightful discussions with Prof. F. Hlawatsch, and Dr. F. Meyer to this paper as well as the likelihood consensus code from Dr. O. Hlinka. The authors would also like to acknowledge one anonymous reviewer's idea for a protocol which 1) applies DSIF for a certain number of time intervals, then 2) reduces the amount of stored information through a weighted averaging — like step, and 3) restarts with DSIF approach. We believe that this is a valuable research topic.

References

[1] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003

[2] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007

[4] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[6] W. Dargie, C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2010.

[7] S. S. Dias and M. G. S. Bruno, "Cooperative target tracking using decentralized particle filtering and RSS Sensors," *IEEE Trans. Signal Process.*, vol. 61, no. 14, pp. 3632–3646, Jul. 2013

[8] M. G. S. Bruno and S. S. Dias, "Cooperative emitter tracking using Rao-Blackwellized random exchange diffusion particle filters," *EURASIP J. Adv. Signal Process.*, vol. 19, pp. 1–18, 2014.

[9] H. Liu, X. Jia, P.-J. Wan, X. Liu, and F. F. Yao, "A distributed and efficient flooding scheme using 1-Hop information in mobile ad hoc networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 5, 658–671, May 2007.

[10] V. Yadav and M. V. Salapaka, "Distributed protocol for determining when averaging consensus is reached," in *Proc. 45th Annu. Allerton Conf.*, Sep. 2007, pp. 715–720.

[11] A. Olshevsky and J. N. Tsitsiklis. "Convergence speed in distributed consensus and averaging," *SIAM Rev.*, vol. 53, no. 4, pp. 747–772, 2011.

[12] P. Frasca, R. Carli, F. Fagnani, and S. Zampieri, "Average consensus on networks with quantized communication," *Int. J. Robust Nonlinear Control*, vol. 19, no. 16, pp. 1787–1816, 2009.

[13] Y. Minsky, A. Trachtenberg, and R. Zippel, "Set reconciliation with nearly optimal communication complexity," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2213–2218, Sep. 2003.

[14] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proc. Amer. Control Conf.*, 1997, vol. 4, pp. 2369–2373.

[15] T. Bailey, S. Julier, and G. Agamennoni, "On conservative fusion of information with unknown non-Gaussian dependence," in *Proc. 2012 15th Int. Conf. Inf. Fusion*, Jul. 2012, pp. 1876–1883.

[16] D. Ciuonzo, A. De Maio, and P. Salvo Rossi, "A systematic framework for composite hypothesis testing of independent Bernoulli Trials," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1249–1253, Sep. 2015.

[17] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symp. Inf. Process. Sensor Netw.*, 2005, pp. 63–70.

[18] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[19] D. Ustebay, M. Coates, and M. Rabbat, "Distributed auxiliary particle filters using selective gossip," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, May 2011, pp. 3296–3299.

[20] S. D. Gupta, M. Coates, and M. Rabbat, "Error Propagation in gossip-based distributed particle filters," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 3, pp. 148–163, Sep. 2015.

[21] X. R. Li, Y. Zhu, J. Wang, and C. Han, "Optimal linear estimation fusion—Part I: Unified fusion rules," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2192–2208, Sep. 2003.

[22] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.

[23] S.-Y. Tu and A. H. Sayed, “Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks,” *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[24] F. S. Cattivelli and A. H. Sayed, “Diffusion Strategies for distributed kalman filtering and smoothing,” *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[25] K. Zhang, X. R. Li, P. Zhang, and H. Li, “Optimal linear estimation fusion—Part VI: Sensor data compression,” in *Proc. 6th Int. Conf. Inf. Fusion*, Cairns, Qld, Australia, 2003, pp. 221–228.

[26] M. Coates, “Distributed particle filters for sensor networks,” in *Proc. Int. Symp. Info. Process. Sens. Netw.*, Berkeley, CA, USA, Apr. 2004, pp. 99–107.

[27] O. Hlinka, O. Slucciak, F. Hlawatsch, P. M. Djuric, and M. Rupp, “Likelihood consensus and its application to distributed particle filtering,” *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4334–4349, Aug. 2012.

[28] O. Hlinka, F. Hlawatsch, and P. M. Djuri, “Consensus-based distributed particle filtering with distributed proposal adaptation,” *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3029–3041, Jun. 2014

[29] X. Sheng, Y. H. Hu, and P. Ramanathan, “Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network,” in *Proc. Int. Symp. Inf. Process. Sens. Netw.*, Los Angeles, CA, USA, Apr. 2005, pp. 181–188.

[30] D. Gu, “Distributed particle filter for target tracking,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 3856–3861.

[31] T. Li, M. Bolic, and P. Djuric, “Resampling methods for particle filtering: Classification, implementation, and strategies,” *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015.

[32] T. Li, G. Villarrubia, S. Sun, J. M. Corchado, and J. Bajo, “Resampling methods for particle filtering: identical distribution, a new method and comparable study,” *Frontiers Inf. Technol. Electron. Eng.*, vol. 16, no. 11, pp. 969–984, 2015.

[33] A. Mohammadi and A. Asif, “Distributed particle filter implementation with intermittent/irregular consensus convergence,” *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2572–2587, May 2013.

[34] S. H. Lee and M. West, “Convergence of the Markov chain distributed particle filter (MCDPF),” *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 801–812, Feb. 2013.

[35] I. L. Manuel and A. N. Bishop, “Distributed Monte Carlo information fusion and distributed particle filtering,” in *Proc. 19th World Congr. Int. Fed. Autom. Control*, Cape Town, South Africa, Aug. 2014, pp. 8681–8688.

[36] M. Uney, D. E. Clark, and S. J. Julier, “Distributed fusion of PHD filters via exponential mixture densities,” *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 521–531, Jun. 2013

[37] G. Battistelli and L. Chisci, “Kullback-Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability,” *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[38] P. M. Djuric, J. Beaudeau, and M. F. Bugallo, “Non-centralized target tracking with mobile agents,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, 2011, pp. 5928–5931.

[39] O. Hlinka, F. Hlawatsch, and P. M. Djuric, “Distributed particle filtering in agent networks: A survey, classification, and comparison,” *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 61–81, Jan. 2013.

[40] V. Savic, H. Wymeersch, and S. Zazo, “Belief consensus algorithms for fast distributed target tracking in wireless sensor networks,” *Signal Process.*, vol. 95, 149–160, 2014.

[41] A. Mohammadi and A. Asif, “Diffusive particle filtering for distributed multisensory estimation,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 3801–3805.

[42] T. Li, J. M. Corchado, J. Bajo, S. Sun, and J. F. Paz, “Effectiveness of Bayesian Filters: An information fusion perspective,” *Inf. Sci.*, vol. 329, pp. 670–689, 2016.

[43] T. Zhao and A. Nehorai, “Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks,” *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1511–1524, Apr. 2007.

[44] J. Prieto, S. Mazuelas, A. Bahillo, P. Fernndez, R. M. Lorenzo, and E. J. Abril, “Adaptive data fusion for wireless localization in harsh environments,” *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1585–1596, Apr. 2012.

[45] A. Mohammadi and A. Asif, “Distributed consensus + innovation particle filtering for bearing/range tracking with communication constraints,” *IEEE Trans. Signal Process*, vol. 63, no. 3, pp. 620–635, Feb. 2015.

[46] S. Seifzadeh, B. Khaleghi, and F. Karray, “Distributed soft-data-constrained multi-model particle filter,” *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 384–394, Mar. 2015.

[47] O. Hlinka, F. Hlawatsch, and P. M. Djuric, “Distributed sequential estimation in asynchronous wireless sensor networks,” *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1965–1969, Nov. 2015.

[48] G. Zhu, F. Zhou, L. Xie, R. Jiang, and Y. Chen, “Sequential asynchronous filters for target tracking in wireless sensor networks,” *IEEE Sens. J.*, vol. 14, no. 9, pp. 3174–3182, Sep. 2014.

[49] J. Kotecha and P. M. Djuri, “Gaussian particle filtering,” *IEEE Trans. Signal Process.*, vol. 51, no. 10, pp. 2592–2601, Oct. 2003.

[50] J. Kotecha and P. M. Djuri, “Gaussian sum particle filtering,” *IEEE Trans. Signal Process.*, vol 51, no. 10, pp. 2602–2612, Oct. 2003.

Tiancheng Li received the Bachelor’s degree in mechanical and electrical engineering, with a minor degree in Business Administration, from Harbin Engineering University, Harbin, China, in 2008, the first Ph.D. degree in electrical and electronic engineering from London South Bank University, London, UK., in 2013, and the second Doctoral degree in mechatronics engineering from Northwestern Polytechnical University, Xian Shi, China, in 2015. Since June 2014, he has been a Research Associate with the BISITE Research Group, University of Salamanca, Spain and presently, he holds a Marie Skodowska-Curie Individual Fellowship H2020-MSCA-IF-2015 with the Host University of Salamanca and the Partner Vienna University of Technology, Austria.

His research interests include the general area of statistical signal processing, information fusion, and distributed computing, with particular interest in novel model-free solutions (by means of data clustering, fitting and mining) for realistic multisensor multiobject detection, tracking and forecasting.

Juan M. Corchado received the Ph.D. degree in computer sciences from the University of Salamanca, Salamanca, Spain, in 1998 and the Ph.D. degree in artificial intelligence from the University of the West of Scotland, Paisley, U.K., in 2000.

He is the Vice President for Research and Technology Transfer since December 2013 and a Full Professor with Chair at the University of Salamanca. He is also the Director of the Science Park of the University and Director of the Doctoral School of the University. Prior to this, he had been twice elected as a Dean of the School of Science. He has been a research collaborator with the Plymouth Marine Laboratory, U.K. since 1993 and been a Visiting Professor in Osaka Institute of Technology, Japan since January 2015. He is the Director of the BISITE (Bioinformatic, Intelligent Systems and Educational Technology) Research Group, which he created in 2000, the President of the IEEE SYSTEMS, MAN AND CYBERNETICS Spanish Chapter. He is the coauthor of more than 300 books, book chapters, papers, technical reports, etc. and holds 24 patents and intellectual properties, most of which are related to bioinformatics, artificial intelligence and intelligent system. He has served as the President of the Organizing or Scientific Committee of more than 20 international conferences, such as SAC, DCAI, and PAAMS.

Javier Prieto received the telecommunication engineer degree, the marketing research and techniques degree, and the Ph.D. degree in information and communication technologies and the Extraordinary Performance Award for Doctorate Studies from the University of Valladolid, Valladolid, Spain, in 2008, 2010, and 2012, respectively.

Since 2007, he has been working in different public and private research centers, such as the University of Valladolid, and as a Visiting Researcher in the Massachusetts Institute of Technology, USA. He currently serves as a Postdoctoral Researcher at StageMotion under a Torres Quevedo grant and as an Assistant Lecturer at the University of Salamanca, where he is member of the BISITE research group. He has authored more than 40 academic papers and 2 Spanish patents, and participated in more than 20 research projects. His research interests include artificial intelligence, localization and navigation technologies, and Bayesian inference techniques, for improving social welfare and environmental sustainability. He serves as an Editor for the IEEE COMMUNICATIONS LETTERS and for the *Wireless Communications and Mobile Computing* journal, as a Guest Editor for the *International Journal of Distributed Sensor Networks* in the special issue on “Energy and Spectrum Efficient Wireless Sensor Networks,” and as a Reviewer in numerous international journals.

QUERIES

Q1. Authors: When accessing and uploading your corrections at the Author Gateway, please note we cannot accept new source files as corrections for your paper. Do not send new Latex, Word, or PDF files, as we cannot simply "overwrite" your paper. Please submit your corrections as an annotated PDF or as clearly written list of corrections, with location in paper. You can also upload revised graphics to the Author Gateway.

Q2. Authors: If you need an invoice or have any other billing questions, please contact reprints@ieee.org as they handle these requests.

Q3. Author: Please provide associate editor name in the place of xxx.