# DIFFUSIVE PARTICLE FILTERING FOR DISTRIBUTED MULTISENSOR ESTIMATION

Arash Mohammadi<sup> $\dagger$ </sup> and Amir Asif<sup> $\ddagger$ </sup>

<sup>†</sup>Concordia Institute for Information Systems Engineering, Concordia University, Montreal, QC, Canada <sup>‡</sup>Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada H3J 1P8

# ABSTRACT

The paper proposes an on-line distributed implementation of the particle filter (DPF) for applications, where the sensing and consensus time scales are the same. We are motivated by state estimation problems in large, geographically-distributed agent/sensor networks, where bandwidth constraints limit the number of information transfers between neighbouring nodes. As an alternative to consensus strategies often used by the DPF, we propose a diffusive framework to eliminate the need of running the consensus step. In our Monte Carlo simulations, the proposed diffusion based DPF (D/DPF) outperforms the state-of-the-art consensus based DPF approaches in environments with limited bandwidth or/and intermittent connectivity.

*Index Terms*— Consensus algorithms, Distributed particle filters, Diffusion algorithms, Target tracking, Intermittent connectivity.

# 1. INTRODUCTION

The paper designs distributed particle filter (DPF) implementations [1, 2] for multisensor navigation and tracking applications, where sensing and consensus time scales are the same, i.e., each node communicates its localized intermediate state estimates only once within its immediate neighbourhood between two successive observations. Existing DPF approaches are often based on two different time scales: (i) The sensing time scale for the collection of measurements in the sensor network, and; (ii) The consensus time scale to attain consistency in the local filters' estimates across the network. Such consensus-based distributed implementations [3, 4] require the consensus step to converge between two consecutive observations. In the context of large, geographically-distributed agent/sensor networks (AN/SN), communication delays and/or intermittence in network connectivity prevent the convergence of the consensus step leading to an accumulation of the observed data and causing these DPF implementations to fail. We propose an alternative DPF approach based on the diffusive fusion strategies [5], which eliminates the need of the consensus step. Another advantage of diffusive fusion is its robustness to changes in the underlying network topology. Reference [6] shows that diffusive strategies outperform consensus approaches for distributed estimation in adaptive AN/SN systems. Surprisingly, diffusive fusion is limited to distributed Kalman filter based estimators for systems with linear dynamics and have not yet been fully investigated for non-linear systems. The paper addresses this gap by introducing a diffusive framework for the DPF.

**Prior Work:** In systems with non-linear dynamics, the limitations of the Kalman filter (lack of optimality, linearization error, and slow convergence) are quite well known. Consequently, there is a surge of interest in developing DPF implementations [7]-[16] for AN/SN

with non-linear dynamics. Existing DPF approaches run localized filters to derive local state estimates. Consensus fusion [3, 4], used to achieve consistency in the local DPF estimates, is iterative in nature, where each node begins with a set of local information (e.g., local state estimates). At each consensus iteration, data is interchanged between neighbouring nodes, which is then used to update the local information. This assimilation process continues until the consensus parameters converge, e.g., to the average of the local values. The performance of the consensus-based approaches depends on the convergence of the consensus step [23] within successive observations, thus, precluding their application from real-time recursive estimation and adaptation [6] in AN/SN with fast sensing time scale.

In [13, 14], we have previously proposed a consensus/fusion based distributed implementation of the particle filter (CF/DPF) that introduces a separate consensus filter (referred to as the fusion filter) to derive the global posterior, thereby reducing the dependency of the CF/DPF on the convergence of the consensus step. In [15, 16], we developed a novel class of consensus + innovation DPF implementations to further reduce the filter's dependence on the convergence of the consensus step, i.e., each node runs a restricted number of consensus iterations between two consecutive observations without requiring the consensus step to converge. In both implementations, the estimation error remains bounded. In several practical applications, the consensus and sensing time steps are often equal [24]-[28], i.e., each node can communicate only once with its neighbouring nodes between two successive observations. Spurred by this consideration, the paper focuses on the design of a DPF implementation based on the diffusive strategies [29]-[32], which we refer to as the D/DPF. The main contribution of this paper is to incorporate diffusive fusion in the non-linear distributed estimation framework to eliminate the need of the consensus step. The condition for achieving consensus between successive iterations of the localized particle filter is no longer a requirement resulting in an on-line D/DPF implementation. Comprising of two steps, the proposed D/DPF approach is not just an averaging approach. First, the centralized particle filter is partitioned into localized filters used to compute intermediate localized estimates based on local observations. This step is called the localized filtering step. Second, local nodes cooperate distributively within their neighbourhoods to improve the accuracy of their intermediate localized estimates using a diffusive strategy focused on sharing local observations. The second step is called the diffusive fusion step that eliminates the need of achieving consensus between the local filters within two successive observations. Since references [5, 6] have derived the improvements possible with diffusive techniques as well as their convergence properties, the paper focuses on designing diffusive approaches for distributed nonlinear filters.

The rest of the paper is organized as follows. Section 2 formulates the problem and reviews both centralized and Gaussian particle filters. Section 3 presents the proposed diffusive algorithm followed by Monte Carlo results in Section 4. Section 5 concludes the paper.

This research is supported in part by Grant No. 504310 from the Natural Science and Engineering Research Conference (NSERC), Canada.

#### 2. PROBLEM FORMULATION AND PARTICLE FILTER

(1)

(2)

 $\boldsymbol{\zeta}(k)$ 

The overall state-space model is given by

State Model:  

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1)) + \mathbf{\xi}(k)$$
Observation Model:
$$\begin{bmatrix} \mathbf{z}^{(1)}(k) \\ \vdots \\ \mathbf{z}^{(N)}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{g}^{(1)}(\mathbf{x}(k)) \\ \vdots \\ \mathbf{g}^{(N)}(\mathbf{x}(k)) \end{bmatrix} + \begin{bmatrix} \mathbf{\zeta}^{(1)}(k) \\ \vdots \\ \mathbf{\zeta}^{(N)}(k) \end{bmatrix}$$

 $\dot{z}(k)$ 

for a sensor network comprising N nodes and observing a set of  $n_x$  state variables  $\boldsymbol{x} = [X_1, X_2, \ldots, X_{n_x}]^T$ . The global observation vector is  $\boldsymbol{z} = [\boldsymbol{z}^{(1)T}, \ldots, \boldsymbol{z}^{(N)T}]^T$  with  $\boldsymbol{z}^{(l)}(k)$  denoting the observation at node l,  $(1 \leq l \leq N)$ , at time instant k. Symbol T denotes transposition and  $\{\boldsymbol{\xi}(\cdot), \boldsymbol{\zeta}(\cdot)\}$  are, respectively, the global non-Gaussian uncertainties in the process and observation models. Both state and observation dynamics  $\{\boldsymbol{f}(.), \boldsymbol{g}(\cdot)\}$  can potentially be non-linear functions.

 $g(\boldsymbol{x}(k))$ 

The optimal Bayesian filtering recursion for iteration k is

$$P(\boldsymbol{x}(k)|\boldsymbol{z}(1:k-1)) = (3)$$

$$\int P(\boldsymbol{x}(k-1)|\boldsymbol{z}(1:k-1))P(\boldsymbol{x}(k)|\boldsymbol{x}(k-1))d\boldsymbol{x}(k-1)$$

and 
$$P(\boldsymbol{x}(k)|\boldsymbol{z}(1:k)) = \frac{P(\boldsymbol{z}(k)|\boldsymbol{x}(k))P(\boldsymbol{x}(k)|\boldsymbol{z}(1:k-1))}{P(\boldsymbol{z}(k)|\boldsymbol{z}(1:k-1))}$$
 (4)

with  $P(\boldsymbol{x}(k)|\boldsymbol{x}(k-1))$  the transitional density based on (1). The particle filter is based on the principle of sequential importance sampling [1, 2], where the filtering distribution  $P(\boldsymbol{x}(k)|\boldsymbol{z}(1:k))$  is represented by its samples (particles)  $\{\mathbb{X}_i(k)\}_{i=1}^{N_s}$  derived from a proposal distribution  $q(\boldsymbol{x}(0:k)|\boldsymbol{z}(1:k))$  with normalized weights  $W_i(k) = P(\mathbb{X}_i(k)|\boldsymbol{z}(1:k))/q(\mathbb{X}_i(0:k)|\boldsymbol{z}(1:k))$  associated with the particles. It implements the filtering recursions by propagating the particles  $\mathbb{X}_i(k)$  and weights  $W_i(k), (1 \le i \le N_s)$ , as

$$\mathbb{X}_{i}(k) \sim q(\boldsymbol{x}_{i}(k)|\boldsymbol{x}_{i}(0:k-1),\boldsymbol{z}(1:k))$$
(5)

$$W_{i}(k) \propto W_{i}(k-1) \frac{P(\boldsymbol{z}(k)|\mathbb{X}_{i}(k))P(\mathbb{X}_{i}(k)|\mathbb{X}_{i}(k-1))}{q(\mathbb{X}_{i}(k)|\mathbb{X}_{i}(k-1), \boldsymbol{z}(1:k))}.$$
 (6)

A computationally attractive approximation of Eqs. (5) and (6) for implementation is the Gaussian particle filter (GPF) [33], where the posterior distribution  $p(\boldsymbol{x}(k)|\boldsymbol{z}(1:k))$  is approximated with a Gaussian density whose mean and covariance are computed from a weighted average of the particles. Contrary to the standard particle filter [1, 2], the GPF does not require a resampling step reducing the local computational complexity of the particle filter. In this paper, we use diffusive strategies to implement a distributed GPF.

#### 3. DIFFUSIVE PARTICLE FILTER

In the D/DPF, each node at iteration k implements a localized filter to compute an intermediate local estimate based on observations limited to its immediate neighbourhood (*Localized Filtering Step*). Local nodes then cooperate distributively with each other to improve the accuracy of their intermediate localized state estimates (*Diffusive Fusion Step*). Below, we explain these steps in more details.

### 3.1. Local Filtering Step

In the D/DPF, the local filter at node l computes an intermediate state estimate of the entire state vector  $\boldsymbol{x}(k)$  by running one local-

ized GPF. In computing the localized state estimates, communication is limited to the local neighbourhoods, i.e., node l forms a localized state estimate by incorporating measurements recorded at those nodes to which it is connected. In the distributed estimation framework considered in this paper, two nodes are considered connected if they can communicate directly with each other. The set of nodes connected to node l is referred to as the neighbourhood of node l and is denoted by  $\aleph^{(l)}$ . Following Reference [29], beside having access to its local measurement, node l has access to the measurements of its neighbouring nodes<sup>1</sup>. The collective set of measurements available at Node l is denoted by

$$\boldsymbol{Z}^{(l)}(k) = \{ \mathbf{z}^{(i)}(k) : i \in \aleph^{(l)} \}.$$
(7)

The local filter at Node *l* computes a particle-based approximation of the local conditional posterior  $p(\boldsymbol{x}(k)|\boldsymbol{Z}^{(l)}(k))$  defined as follows

$$p(\boldsymbol{x}(k)|\boldsymbol{Z}^{(l)}(k)) = \sum_{i=1}^{n_p} W_i^{(l)} \delta\big(\boldsymbol{x}(k) - \mathbb{X}_i^{(l)}(k)\big),$$
(8)

where  $n_p$  is the number of individualized particles used by the local filters. In reality, the number of local particles may vary within nodes without affecting our implementation. The local intermediate state estimate denoted by  $\psi^{(l)}(k)$  at iteration  $(k \ge 1)$  is defined as the expected value of the posterior distribution  $p(\boldsymbol{x}(k)|\boldsymbol{Z}^{(l)}(k))$ , i.e.,

$$\boldsymbol{\psi}^{(l)}(k) = \mathbb{E}\left\{\boldsymbol{x}(k) | \boldsymbol{Z}^{(l)}(k)\right\} = \int \boldsymbol{x}(k) p(\boldsymbol{x}(k) | \boldsymbol{Z}^{(l)}(k)) d\boldsymbol{x}(k), \quad (9)$$

which is expressed in terms of the local particles as follows

1

$$\psi^{(l)}(k) = \sum_{i=1}^{n_p} W_i^{(l)}(k) \mathbb{X}_i^{(l)}(k).$$
(10)

Node *l* fuses its local intermediate state estimate  $\psi^{(l)}(k)$  with those of its neighbouring nodes using diffusive strategies to form its updated local state estimate, denoted by  $\hat{x}^{(l)}(k)$ . Assume all local filters are at steady-state at the end of iteration (k-1), i.e., node *l*, has computed  $\hat{x}^{(l)}(k)$  and its corresponding error covariance  $P^{(l)}(k)$ . At iteration *k*, the local filtering step is then completed at each node *l*,  $(1 \le l \le N)$  based on the following sub-steps:

Sub-Step L1. Observation collection: Node l collects observations made in its neighbourhood to form  $Z^{(l)}(k)$ , i.e., the collection of measurements available in the local neighbourhood  $\aleph^{(l)}$  of node l.

Sub-Step L2. Local state estimation: Node l computes the local state estimate  $\psi^{(l)}(k)$  based on the collective observations available in its neighbourhood as described next. First, the local GPF generates  $n_p$  random particles from its local proposal distribution, i.e.,

$$\tilde{\mathbb{X}}_{i}^{(l)}(k) \sim \pi(\boldsymbol{x}(k) | \boldsymbol{Z}^{(l)}(1:k)),$$
(11)

where  $\mathbf{Z}^{(l)}(1:k) = [\mathbf{Z}^{(l)}(1)^T, \dots, \mathbf{Z}^{(l)}(k)^T]^T$  is the collection of local observations from iteration (k = 1) to the current iteration. Node *l* computes the mean  $\bar{\boldsymbol{\mu}}^{(l)}(k)$  and covariance  $\bar{\boldsymbol{\Sigma}}^{(l)}(k)$  of its predictive particles as follows

$$\bar{\boldsymbol{\mu}}^{(l)}(k) = \frac{1}{n_p} \sum_{i=1}^{n_p} \tilde{\mathbb{X}}_i^{(l)}(k)$$
(12)

and 
$$\bar{\boldsymbol{\Sigma}}^{(l)}(k) = \frac{1}{n_p} \sum_{i=1}^{n_p} (\bar{\boldsymbol{\mu}}^{(l)}(k) - \tilde{\mathbb{X}}_i^{(l)}(k)) (\bar{\boldsymbol{\mu}}^{(l)}(k) - \tilde{\mathbb{X}}_i^{(l)}(k))^T$$
(13)

<sup>1</sup>Based on the unscented particle filter, an enhancement of the D/DPF can be developed that eliminates the need for sharing localized observations within local neighbourhoods. This will be the focus of our future work.

Node l then updates the corresponding weights of its predictive particles as follows

$$\tilde{W}_{i}^{(l)} = \frac{p(\boldsymbol{Z}^{(l)}(k)|\tilde{\mathbb{X}}^{(l)}(k)) \overbrace{\mathcal{N}[\tilde{\mathbb{X}}_{i}^{(l)}(k); \boldsymbol{\bar{\mu}}^{(l)}(k), \boldsymbol{\bar{\Sigma}}^{(l)}(k)]}^{p(\boldsymbol{x}(k)|\boldsymbol{x}(k-1))}}{\pi(\tilde{\mathbb{X}}_{i}^{(l)}(k)|\boldsymbol{Z}^{(l)}(1:k))}, \quad (14)$$

and normalize them as

$$W_i^{(l)} = \tilde{W}_i^{(l)} / \sum_{i=1}^{n_p} \tilde{W}_i^{(l)}.$$
(15)

In Eq. (14),  $\mathcal{N}[\cdot]$  denotes the Gaussian distribution with mean and covariance specified within its parenthesis. Further, Node *l* updates its local intermediate state estimate and corresponding covariance as

$$\psi^{(l)}(k) = \sum_{i=1}^{n_p} W_i^{(l)} \tilde{\mathbb{X}}_i^{(l)}(k)$$
(16)

and 
$$\mathbf{P}^{(l)}(k) = \sum_{i=1}^{n_p} W_i^{(l)} (\boldsymbol{\psi}^{(l)}(k) - \tilde{\mathbb{X}}_i^{(l)}(k)) (\boldsymbol{\psi}^{(l)}(k) - \tilde{\mathbb{X}}_i^{(l)}(k))^T (17)$$

Implemented at node l,  $(1 \le l \le N)$ , the local GPF approximates the localized filtering density with a single Gaussian as follows

$$p(\boldsymbol{x}(k)|\boldsymbol{Z}^{(l)}(1:k)) = \mathcal{N}(\boldsymbol{x}(k); \boldsymbol{\psi}^{(l)}(k), \boldsymbol{P}^{(l)}(k)).$$
(18)

This completes the local filtering step of the proposed D/DPF. Next, we present our diffusive fusion strategy where each node updates its local state estimates by collaborating with its neighbouring nodes.

## 3.2. Diffusion Step

The second step is based on local collaboration, where node l,  $(1 \le l \le N)$ , fuses its local intermediate estimate  $\psi^{(l)}(k)$  with that of its neighbouring nodes as follows

$$\hat{\boldsymbol{x}}^{(l)}(k) = \sum_{j \in \aleph^{(l)}} \alpha^{(j,l)} \boldsymbol{\psi}^{(j)}(k),$$
(19)

such that if we collect the nonnegative weights  $\alpha^{(j,i)}$  into a  $N \times N$  matrix  $\boldsymbol{A}$ , the weights  $\alpha^{(j,l)}$  satisfy the following properties

(i) 
$$\alpha^{(j,l)} \ge 0$$
; (ii)  $\mathbf{A}^T \mathbf{1} = \mathbf{1}$ , and; (iii)  $\alpha^{(j,l)} = 0$  if  $j \notin \aleph^{(l)}$ , (20)

where 1 is a vector of size N with all entries equal to one. Eq. (20) implies that the weights on the links arriving at a single node add up to one, which is equivalent to saying that the matrix is left-stochastic. Moreover, if two nodes are not connected, then their corresponding entry is zero. Diffusive matrix A can be designed using covariance intersection [31] or updated adaptively as explained in [32]. A simple approach for choosing the diffusion matrix is to assign a weigh to each node according to the cardinality of its neighbourhood, i.e.,

$$\alpha^{(j,l)} = \begin{cases} \alpha^{(l)} \Delta^{(l)} & \text{if } j \in \aleph^{(l)} \\ 0 & \text{otherwise} \end{cases}$$
(21)

where  $\Delta^{(l)}$  is the connectivity degree of node l (i.e., cardinality of  $\aleph^{(l)}$ ) and  $\alpha^{(l)}$  is a normalization constant to ensure that  $A^T \mathbf{1} = \mathbf{1}$ . Through diffusive fusion, the GPF implemented at node l,  $(1 \le l \le N)$ , forms a Gaussian approximation of the posterior distribution as

$$p(\boldsymbol{x}(k)|\boldsymbol{z}(k)) = \mathcal{N}(\boldsymbol{x}(k); \hat{\boldsymbol{x}}^{(l)}(k), \boldsymbol{P}^{(l)}(k)).$$
(22)

Algorithm 1 D/DPF IMPLEMENTATION

Input:  $\{\mathbb{X}_{i}^{(l)}(k-1), W_{i}^{(l)}(k-1)\}_{i=1}^{n_{p}}$  and  $\boldsymbol{z}^{(l)}(k)$ . Output:  $\{\mathbb{X}_{i}^{(l)}(k), W_{i}^{(l)}(k)\}_{i=1}^{n_{p}}, \hat{\boldsymbol{x}}^{(l)}(k)$  and  $\boldsymbol{P}^{(l)}(k)$ .

- **Localized GPFs:** At iteration k, Node l,  $(1 \le l \le N)$ , updates its particle set as follows:
- L1. Local Observation Collection: Form  $Z^{(l)}(k)$  by collecting  $z^{(j)}(k), (j \in \aleph^{(l)})$ .
- L2. Local Predictive Particle Generation: Sample a new predicted particle  $\tilde{\mathbb{X}}_{i}^{(l)}(k)$  using Eq. (11).
- L3. Gaussian Modeling Predictive Distribution: Compute mean  $\bar{\mu}^{(l)}(k)$  and covariance  $\bar{\Sigma}^{(l)}(k)$  of a Gaussian approximation of the local predictive density using  $\tilde{\mathbb{X}}_{i}^{(l)}(k)$  and Eqs. (12)-(13).
- L4. Weight Update: Compute the weights associated with  $\tilde{\mathbb{X}}_{i}^{(l)}(k)$  using Eq. (14).
- L5. Intermediate Local Estimates: Approximate the local state estimate  $\boldsymbol{\psi}^{(l)}(k)$  and its corresponding error covariance  $\boldsymbol{P}^{(l)}(k)$  from  $\{\tilde{\mathbb{X}}_{i}^{(l)}, W_{i}^{(l)}(k)\}_{i=1}^{N_{p}^{(l)}}$  and Eqs. (16)-(17).
- **Diffusive Fusion**: Node *l* updates its local state estimate  $\hat{x}^{(l)}(k)$  using  $\psi^{(j)}(k), (j \in \aleph^{(l)})$  and Eq. (19).
- L6. Diffusive Particle Generation: Node l generates a set of  $n_p$  updated particles  $\mathbb{X}_i^{(l)}(k)$  by sampling from the Gaussian approximation of the posterior distribution given by Eq. (22).

It is important to note that although the notation  $\mathbf{P}^{(l)}(k)$  has been used in Eq. (22), it does not represent the covariance of the diffusive state estimate  $\hat{\mathbf{x}}^{(l)}(k)$ . This is because the diffusion update is only performed on the intermediate state estimates  $\psi^{(l)}(k)$  and not performed on the covariance matrices. Algorithm 1 lists the steps involved in our diffusive particle filter implementation.

**Comparison with Consensus Fusion**: The diffusive fusion used in the proposed D/DPF is similar in nature to the update step of consensus based algorithms with an important difference that is elaborated in terms of the conventional Laplacian-based consensus algorithm [4], where each node updates its local state estimate as follows

$$\hat{\boldsymbol{x}}^{(l)}(k) = \boldsymbol{\psi}^{(l)}(k) + \epsilon \sum_{j \in \aleph^{(l)}} \left( \boldsymbol{\psi}^{(l)}(k) - \boldsymbol{\psi}^{(j)}(k) \right)$$
(23)
$$= \left( 1 - (n^{(l)} - 1)\epsilon \right) \boldsymbol{\psi}^{(l)}(k) + \sum_{j \in \aleph^{(l)} - \{l\}} \boldsymbol{\psi}^{(j)}(k),$$

where  $\epsilon \in (0, 1/\Delta_{\max}^{(l)})$  with  $\Delta_{\max}^{(l)}$  the maximum degree of local neighbourhoods within the AN/SN. In consensus-type algorithms, the weights are  $\epsilon$  and  $(1 - (n^{(l)} - 1)\epsilon)$ , respectively. In the D/DPF, a convex combination of the intermediate estimates of the neighbours are used with more general weights. This is a key difference between the D/DPF and existing distributed particle filters, which results in the improved performance observed in our simulations presented in Section 4. The diffusive weights  $\alpha^{(j,l)}$  can change with time resulting in an adaptive, time-varying approach.

**Communication Complexity**: At each iteration k, node l communicates: (i) its vector observation  $z^{(l)}$  of length  $n_z^{(l)}$  within its neighbourhood during the local filtering step, and; (ii) the intermediate state estimates  $\psi$  of dimension  $n_x$  during the diffusion update step. The communication overhead for node l is  $(n_x + n_z^{(l)})$ . Please note that the D/DPF involves only one iteration of diffusive fusion between two successive observations within local neighbourhoods,



**Fig. 1**. RMS errors in the target's track estimated using the centralized particle filter, the proposed D/DPF, and two consensus-based DPF implementations, namely the CI/DUPF [16] and Gu *et al.* [36].

while the consensus-based approaches require several consensus iterations across the network for the consensus to converge. The mutual transfer of observations within local neighbourhoods in D/DPF is a fairly common observation strategy [6] used in distributed particle filter implementations.

### 4. EXPERIMENTAL RESULTS

A distributed bearing-only target tracking application [35] is simulated to test the proposed D/DPF, where the trajectory of a maneuvering target (i.e., its position [X, Y] and velocity  $[\dot{X}, \dot{Y}]$ ) is estimated. A sensor network of N = 20 nodes is considered with sensors distributed randomly in a square region. Each sensor communicates with its neighbours within a connectivity radius of  $\sqrt{2 \log(N)/N}$  units. In this paper, we consider a non-linear clockwise coordinated turn kinematic motion model [35] with Gaussian observation noise. The process excitation is Gaussian,  $\mathcal{N}(0, \sigma_v = 0.016)$ . Measurement at node l is the target's bearing with respect to the node's platform (referenced clockwise positive to the *y*-axis), i.e.,

$$\mathbf{z}^{(l)}(k) = \operatorname{atan}\left(\frac{X(k) - X^{(l)}}{Y(k) - Y^{(l)}}\right) + \boldsymbol{\zeta}^{(l)}(k),$$
 (24)

where  $(X^{(l)}, Y^{(l)})$  are the coordinates of node l. The bearing noise variance  $\sigma^{(l)^2}(k)$  at node l is dependent on the distance between the observer and the target  $r^{(l)}(k)$ , and is given by

$$\sigma^{(l)^2}(r^{(l)}(k)) = 0.01r^{(l)^2}(k) + 0.0115r^{(l)}(k).$$
<sup>(25)</sup>

In order to show the robustness of the D/DPF to noise, we considered a tracking scenario with state-dependent observation noise and large initialization uncertainty [35]. Due to state-dependent noise variance, we note that the signal to noise ratio (SNR) is time-varying with the mean SNR of 16dB and variance of 4.05 (averaged across all nodes and time). Initialization of the filter is performed according to [35]. For all nodes, the standard deviation of the initial observation error is set to  $7.5^{\circ}$ . The target starts its track from coordinates (10, 10) with the initial course set at  $-150^{\circ}$ . In our Monte Carlo simulation, we test the following four particle filter implementations.



Fig. 2. Target's track estimated using the proposed D/DPF.

- 1. The centralized particle filter used as the benchmark.
- 2. The proposed diffusive D/DPF described in the paper.
- 3. The distributed consensus + innovation particle filter (CI/DUPF) [16], which has been specially designed for networks with communication constraints and limited consensus iterations. In the CI/DUPF, the number of consensus iterations in between two successive iterations of the DPF is limited to 3. Convergence of the consensus step is not guaranteed.
- Distributed particle filter [36], referred to as Gu *et al.*, where the consensus algorithms are allowed to converge.

The total number  $n_p$  of the vector particles in all implementations is set to 1000 with the particles evenly divided between the local filters in the distributed implementations. Based on a Monte-Carlo simulation of 100 runs, Fig. 1 plots the root mean square (RMS) position error between the target's true and estimated positions as it evolves over time for the aforementioned distributed implementations of the particle filter. See [16] for the definition of the RMS position error. The RMS error corresponding to the D/DPF is lower than its counterparts showing that the diffusive fusion is more robust to initialization uncertainties and noise. In the context of linear systems using the Kalman filter, the superiority of the diffusive fusion strategies over consensus strategies has already been reported in [6]. We confirm the result for the distributed particle filter implementations developed for state estimation in non-linear systems. A realization of the sensor placement is shown in Fig. 2, where we see that the estimated trajectory of the D/DPF overlaps the actual trajectory of the target. Additional simulations corroborated these findings.

#### 5. SUMMARY

The paper proposes the diffusion-based distributed implementation of the particle filter (D/DPF) based on our diffusive-fusion framework for intermittently connected networks with non-linear dynamics. Our main contribution is to extend the distributed non-linear estimation framework to incorporate diffusive fusion and to eliminate the need of consensus of any form. The condition for achieving consensus between successive iterations of the local particle filters is no longer a requirement, resulting in an *on-line* D/DPF implementation.

#### 6. REFERENCES

- P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, J. Miguez, "Particle Filtering", *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19-38, 2003.
- [2] N. Gordon, M. Sanjeev, S. Maskell, and T. Clapp, "A tutorial on particle filters for online non-linear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–187, 2002.
- [3] R. Olfati-Saber, A. Fax, and R.M. Murry, "Consensus and coopration in networked multi-agent systems," *Proceedings of the IEEE*, 2007.
- [4] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, pp. 1847–1864, 2010.
- [5] A.H. Sayed, "Adaptive Networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460-497, 2014.
- [6] S.Y. Tu and A.H. Sayed, "Diffusion Strategies Outperform Consensus Strategies for Distributed Estimation Over Adaptive Networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217-6234, 2012.
- [7] X. Zhong and A. B. Premkumar, "Particle filtering approaches for multiple acoustic source detection and 2-D direction of arrival estimation using a single acoustic vector sensor," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4719-4733, Sept. 2012.
- [8] Y. Huang, W. Liang, H. Yu, and Y. Xiao, "Target tracking based on a distributed particle filter in underwater sensor networks," *Wireless Comm. and Mobile Comp.*, pp. 1023–1033, 2008.
- [9] T. Zhao and A. Nehorai, "Distributed Sequential Bayesian Estimation of a Diffusive Source in Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 4, pp. 1511-1524, 2007.
- [10] C.J. Bordin and M.G.S. Bruno, "Consensus-based distributed particle filtering algorithms for cooperative blind equalization in receiver networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3968-3971, 2011.
- [11] G.G. Rigatos and P. Siano, "Distributed state estimation for condition monitoring of non-linear electric power systems," In *IEEE Int. Sym.* on *Industrial Electronics*, pp. 1703–1708, 2011.
- [12] O. Hlinka, F. Hlawatsch and P.M. Djuric, "Consensus-based Distributed Particle Filtering With Distributed Proposal Adaptation," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3029-3041, 2014.
- [13] A. Mohammadi and A. Asif, "Distributed Particle Filter Implementation with Intermittent/Irregular Consensus Convergence," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2572-2587, May 2013.
- [14] A. Mohammadi and A. Asif, "Full order non-linear distributed estimation in intermittently connected networks." *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6327-6331, 2013.
- [15] A. Mohammadi and A. Asif, "Application of Constraint Sufficient Statistics to Distributed Particle Filters: Bearing/Range Tracking," accepted in *IEEE Transactions on Signal Processing*, 2014.
- [16] A. Mohammadi and A. Asif, "A distributed consensus plus innovation particle filter for networks with communication constraints," *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP), pp. 1105-1109, 2014.
- [17] S.H. Lee and M. West, "Convergence of the Markov Chain Distributed Particle Filter (MCDPF)," *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 801-812, 2013.
- [18] S.S. Dias and M.G.S. Bruno, "Cooperative Target Tracking Using Decentralized Particle Filtering and RSS Sensors," *IEEE Transactions on Signal Processing*, vol. 61, no. 14, pp. 3632-3646, 2013.

- [19] O. Hlinka, F. Hlawatsch, and P.M. Djuric, "Distributed Particle Filtering in Agent Networks: A Survey, Classification, and Comparison," IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 61-81, 2013.
- [20] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, "'Setmembership constrained particle filter: Distributed adaptation for sensor networks," IEEE Transactions on Signal Processing, vol. 59, no. 9, pp. 4122-4138, Sept. 2011.
- [21] D. Ustebay, R. Castro, M. Coates, and M. Rabbat, "Distributed Approximation and Tracking Using Selective Gossip," *Compressed Sensing and Sparse Filtering*, Springer Berlin Heidelberg, pp. 325-355, 2014.
- [22] D. Ustebay, M. Coates, and M. Rabbat, "Distributed auxiliary particle filters using selective gossip," In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3296–3299, 2011.
- [23] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," In SIAM J. Control Optim., vol. 48, no. 1, pp. 33–56, 2009.
- [24] S. Kar and J.M.F. Moura, "Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 99-109, 2013.
- [25] U.A. Khan and A. Jadbabaie, "Networked estimation under information constraints," arXiv preprint arXiv:1111.4580, 2011.
- [26] S. Kar and J.M.F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 674-690, 2011.
- [27] M. Doostmohammadian and U.A. Khan, "On the Genericity Properties in Distributed Estimation: Topology Design and Sensor Placement," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 195-204, 2013.
- [28] S. Das and J. M. F. Moura, "Distributed state estimation in multi-agent networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4246-4250, 2013.
- [29] F. Cattivelli and A.H. Sayed, "Diffusion Strategies for Distributed Kalman Filtering and Smoothing," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069-2084, 2010.
- [30] J.W. Lee; S.E. Kim, W.J. Song, and A.H. Sayed, "Spatio-Temporal Diffusion Strategies for Estimation and Detection Over Networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4017-4034, 2012.
- [31] J. Hu, L. Xie, and C. Zhang, "Diffusion Kalman Filtering Based on Covariance Intersection," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 891-902, 2012.
- [32] F. Cattivelli and A.H. Sayed, "Diffusion distributed Kalman filtering with adaptive weights," *Asilomar Conference on Signals, Systems and Computers*, pp. 908-912, 2009.
- [33] J.H. Kotecha and P.M. Djuric, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592-2601, 2003.
- [34] O. Hlinka, O. Sluciak, F. Hlawatsch, P.M. Djuric, and M. Rupp, "Distributed Gaussian particle filtering using likelihood consensus," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3756-3759, 2011.
- [35] M. S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell, "Bearingsonly tracking of maneuvering targets using particle filters," *EURASIP Applied Signal Processing*, vol. 15, pp. 2351-2365, 2004.
- [36] D. Gu, S. Junxi, H. Zhen and L. Hongzuo, "Consensus based distributed particle filter in sensor networks," in *IEEE International Conference on Information and Automation*, pp. 302–307, 2008.