

# FULL ORDER NONLINEAR DISTRIBUTED ESTIMATION IN INTERMITTENTLY CONNECTED NETWORKS

Arash Mohammadi and Amir Asif

Computer Science and Engineering

York University, Toronto, ON, Canada M3J 1P3. Email: {marash, asif}@cse.yorku.ca

## ABSTRACT

The paper considers the problem of performing distributed particle filtering in intermittently connected networks with nonlinear state dynamics. In the context of large, geographically-distributed sensor networks, communication delays affect the convergence of the consensus algorithms used to derive the global state estimate from local estimates. We propose a non-linear fusion rule that relaxes the condition of requiring convergence of the consensus step between two successive iterations of the localized particle filters, thereby, allowing the consensus step to catch up with the localized filters in case of communication delays. Our Monte Carlo simulations illustrate the ability of the modified consensus/fusion based distributed implementation of the particle filter (MCF/DPF) to successfully handle intermittence in the network connectivity.

**Index Terms**— Consensus algorithms, Distributed estimation, Multi-sensor tracking, Intermittent networks, and Particle filters.

## 1. INTRODUCTION

Motivated by multisensor navigation and tracking applications, the paper considers distributed particle filter [1]-[3] implementations for intermittently connected networks [4] with nonlinear state dynamics. Distributed estimation in such sensor networks share some form of a consensus procedure [5, 6] typically based on the Chong-Mori-Chang track-fusion principle [7] that implicitly/explicitly fuses local estimates obtained from the localized filters to form the global estimate. Intermittent network connectivity results in information loss, significant delays in the convergence of the consensus algorithms, and loss in synchronization between the localized and fusion filters used to achieve the global consensus. In this paper, we study a generic framework for distributed estimation in intermittently connected networks from the consensus-convergence perspective. Here, the fundamental question is: *How can loss of synchronization between the localized and fusion filters be adequately resolved to compensate for delays in the convergence of the consensus algorithms?*

**Prior Work:** Distributed estimation of an unknown set of parameters by a network of randomly deployed sensors is of relevance in many nonlinear signal processing applications [8]-[18]. Most of the earlier work in distributed nonlinear estimation was based on message passing particle filter implementations [19, 20], where information is communicated across the network in a predefined orderly fashion. Though a fusion centre is not needed, the network topology is assumed known in message passing mechanisms. For arbitrarily configured networks with unknown topologies, consensus-based particle filter approaches [21]-[36] are more efficient alternatives. Sensors interchange information only with their immediate neighbours and iteratively refine the local estimates based on the information received. Among the types of information exchanged locally in

consensus based approaches, communicating local state posteriors [21]-[28] has been shown to be more resilient to packet losses in error prone networks as compared to sharing of local likelihoods [29]-[36]. In principle, any lost information should be contained in the following posteriors and, therefore, can be recovered. However, the main drawback of such methods is that their performance depends on the convergence [37] of the consensus step. In [21, 22], 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. In reality, communication channels are time-varying and unreliable with intermittent connectivity. Consequently, the fusion filter in the CF/DPF is unable to synchronize with the localized filters. Other consensus based estimation approaches [23]-[28] are also susceptible to such out-of-synchronization issues. Spurred by this consideration, the paper focuses on the design of a nonlinear, distributed estimation approach for unreliable, intermittently connected networks.

The main contribution of this paper is to extend the distributed estimation framework to unreliable networks where the localized and fusion filters lose synchronization due to delays in the convergence of the consensus algorithms used in the fusion filters. Though we explain the methodology in terms of the CF/DPF, generalization to other consensus based distributed estimation approaches is straightforward. We propose the modified CF/DPF (MCF/DPF) to handle such out of synchronization issues by implementing the modified fusion filter to run at a rate different from that of the local filters. The modified fusion filter is based on a non-linear fusion rule derived in the paper as an alternative to the Chong-Mori-Chang track-fusion principle. The condition for achieving consensus between two successive iterations of the localized particle filter is relaxed, which enables the consensus step to converge without strict time limitations. Our numerical simulations verify the efficiency of the proposed MCF/DPF in such intermittently connected networks.

The rest of the paper is organized as follows. Section 2 formulates the problem and introduces both the centralized and CF/DPF implementations. The modified fusion filter is discussed in Section 3, while Section 4 presents the simulation results.

## 2. PROBLEM FORMULATION AND PARTICLE FILTER

The overall state-space model is given by

$$\text{State Model: } \mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1)) + \boldsymbol{\xi}(k) \quad (1)$$

$$\text{Observation Model: } \underbrace{\begin{bmatrix} \mathbf{z}^{(1)}(k) \\ \vdots \\ \mathbf{z}^{(N)}(k) \end{bmatrix}}_{\mathbf{z}(k)} = \underbrace{\begin{bmatrix} \mathbf{g}^{(1)}(\mathbf{x}(k)) \\ \vdots \\ \mathbf{g}^{(N)}(\mathbf{x}(k)) \end{bmatrix}}_{\mathbf{g}(\mathbf{x}(k))} + \underbrace{\begin{bmatrix} \boldsymbol{\zeta}^{(1)}(k) \\ \vdots \\ \boldsymbol{\zeta}^{(N)}(k) \end{bmatrix}}_{\boldsymbol{\zeta}(k)} \quad (2)$$

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

The optimal Bayesian filtering recursion for iteration  $k$  is

$$P(\mathbf{x}(k)|\mathbf{z}(1:k-1)) = \int P(\mathbf{x}(k-1)|\mathbf{z}(1:k-1))f(\mathbf{x}(k)|\mathbf{x}(k-1))d\mathbf{x}(k-1) \quad (3)$$

$$\text{and } P(\mathbf{x}(k)|\mathbf{z}(1:k)) = \frac{P(\mathbf{z}(k)|\mathbf{x}(k))P(\mathbf{x}(k)|\mathbf{z}(1:k-1))}{P(\mathbf{z}(k)|\mathbf{z}(1:k-1))}. \quad (4)$$

The particle filter is based on the principle of sequential importance sampling (SIS) [39], where the filtering distribution  $P(\mathbf{x}(k)|\mathbf{z}(1:k))$  is represented by its samples (particles)  $\{\mathbb{X}_i(k)\}_{i=1}^{N_s}$  derived from a proposal distribution  $q(\mathbf{x}(0:k)|\mathbf{z}(1:k))$  with normalized weights  $W_i(k) = \frac{P(\mathbb{X}_i(k)|\mathbf{z}(1:k))}{q(\mathbb{X}_i(k)|\mathbf{z}(1:k))}$  associated with the vector particles. The particle filter implements the filtering recursions by propagating the particles  $\mathbb{X}_i(k)$  and associated weights  $W_i(k)$ , ( $1 \leq i \leq N_s$ ), as

$$\mathbb{X}_i(k) \sim q(\mathbb{X}_i(k)|\mathbb{X}_i(0:k-1), \mathbf{z}(1:k)) \quad (5)$$

$$W_i(k) \propto W_i(k-1) \frac{P(\mathbf{z}(k)|\mathbb{X}_i(k))P(\mathbb{X}_i(k)|\mathbb{X}_i(k-1))}{q(\mathbb{X}_i(k)|\mathbb{X}_i(0:k-1), \mathbf{z}(1:k))}. \quad (6)$$

## 2.1. Distributed Particle Filter

Our distributed implementation is based on the following model

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1)) + \xi(k) \quad (7)$$

$$\mathbf{z}^{(l)}(k) = \mathbf{g}^{(l)}(\mathbf{x}(k)) + \zeta^{(l)}(k), \quad (8)$$

for sensor nodes ( $1 \leq l \leq N$ ). The entire state vector  $\mathbf{x}(k)$  is estimated by running one local filter at each node based only on its local observation  $\mathbf{z}^{(l)}(1:k)$ . By introducing a fusion filter (one per sensor node), the CF/DPF [21, 22] fuses the local filtering and prediction distributions to derive the global posterior distribution

$$P(\mathbf{x}(0:k)|\mathbf{z}(1:k)) \propto \frac{\prod_{l=1}^N P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k))}{\prod_{l=1}^N P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k-1))} \times P(\mathbf{x}(0:k)|\mathbf{z}(1:k-1)), \quad (9)$$

using the fusion rule [7]. Here,  $P^{(l)}(\cdot)$  is the relevant local posterior at node  $l$ . The  $k$ 'th iteration of the CF/DPF is based on the following steps. The weighted particles  $\{\mathbb{X}_i^{(l,LF)}(k-1), W_i^{(l,LF)}(k-1)\}$  for the local filter and particles  $\{\mathbb{X}_i^{(l,FF)}(k-1), W_i^{(l,FF)}(k-1)\}$  for the fusion filter are available from the previous iteration at each node.

**1. Local Filters:** The local filters are direct implementation of Eqs. (5)–(6) computing  $\{\mathbb{X}_i^{(l,LF)}(k), W_i^{(l,LF)}(k)\}$  from  $\{\mathbb{X}_i^{(l,LF)}(k-1), W_i^{(l,LF)}(k-1)\}$  and local observations  $\mathbf{z}^{(l)}(1:k)$ , followed by a re-sampling step if degeneracy occurs. The proposal distribution for the local filters is modeled as  $P^{(l)}(\mathbf{x}(k)|\mathbf{x}(k-1))$ .

**2. Local Statistics:** Given  $\{\mathbb{X}_i^{(l,LF)}(k), W_i^{(l,LF)}(k)\}$ , the local filter at node  $l$ , ( $1 \leq l \leq N$ ), computes the minimum mean square error estimates (MMSE) of the mean  $\boldsymbol{\mu}^{(l)}(k)$  and error covariance  $\mathbf{P}^{(l)}(k)$  of its local filtering distribution  $P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k))$  as well as the MMSE of the mean  $\mathbf{v}^{(l)}(k)$  and error covariance  $\mathbf{R}^{(l)}(k)$  of its local prediction distribution  $P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k-1))$ .

**3. Consensus Step:** The two product densities in the first right hand term in Eq. (9) are approximated [42] as

$$\prod_{l=1}^N P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k)) \triangleq \prod_{l=1}^N \mathcal{N}(\boldsymbol{\mu}^{(l)}(k), \mathbf{P}^{(l)}(k)) \propto \mathcal{N}(\boldsymbol{\mu}(k), \mathbf{P}(k))$$

$$\prod_{l=1}^N P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k-1)) \triangleq \prod_{l=1}^N \mathcal{N}(\mathbf{v}^{(l)}(k), \mathbf{R}^{(l)}(k)) \propto \mathcal{N}(\mathbf{v}(k), \mathbf{R}(k))$$

where  $\mathcal{N}(\cdot, \cdot)$  denotes a Gaussian distribution. The statistics of the two product densities  $\{\boldsymbol{\mu}(k), \mathbf{P}(k)\}$  and  $\{\mathbf{v}(k), \mathbf{R}(k)\}$  are computed by running four vector consensus algorithms [21, 22].

**4. Global Estimates:** The fusion filter estimates the global posterior distribution  $P(\mathbf{x}(0:k)|\mathbf{z}(1:k))$ . Being a particle filter itself, implementation of the fusion filter requires the proposal distribution from which particles of the fusion filter are derived, i.e.,

$$\mathbb{X}_i^{(l,FF)}(k) \sim q(\mathbf{x}(k)|\mathbf{x}(k-1), \mathbf{z}(1:k)). \quad (10)$$

Please refer to [21, 22] for three different choices of the fusion filter's proposal distribution. The weight update equation is given by

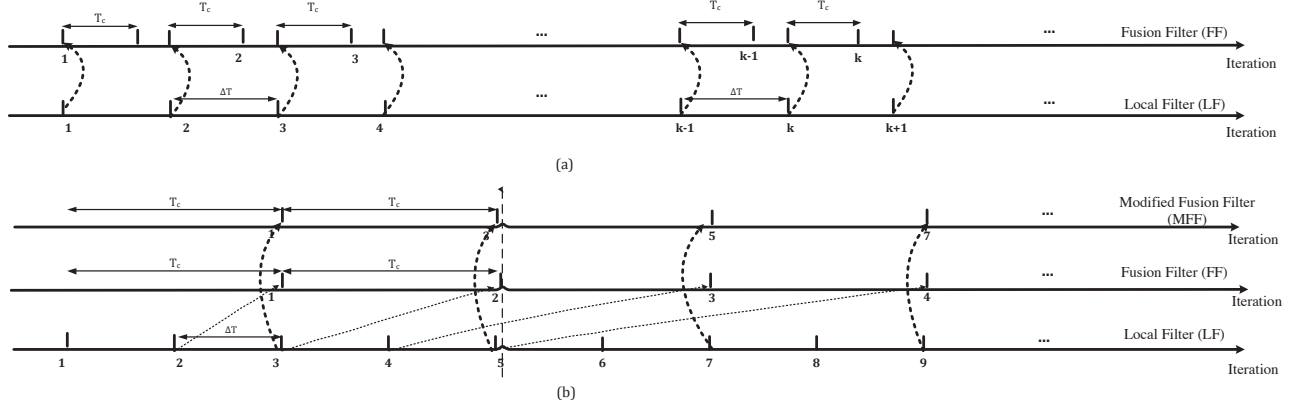
$$W_i^{(l,FF)}(k) \propto W_i^{(l,FF)}(k-1) \times \frac{\mathcal{N}(\mathbb{X}_i^{(l,FF)}(k); \boldsymbol{\mu}(k), \mathbf{P}(k))P(\mathbb{X}_i^{(l,FF)}(k)|\mathbb{X}_i^{(l,FF)}(k-1))}{\mathcal{N}(\mathbb{X}_i^{(l,FF)}(k); \mathbf{v}(k), \mathbf{R}(k))q(\mathbf{x}(k)|\mathbf{x}(k-1), \mathbf{z}(1:k))}. \quad (11)$$

Steps (2)–(4) describe the fusion filter at iteration  $k$ . For iteration  $k+1$ , the time index  $k$  is incremented and Steps (1)–(4) are repeated.

We note that approximating the two product terms in the consensus step (Step 3 above) does not restrict the global posterior to be Gaussian as can be seen in Eq. (9). The last term in (9) is derived from the nonlinear state model and global posterior obtained from the previous CF/DPF iteration. Similarly, the local posteriors resulting from the localized particle filters are non-Gaussians by nature allowing the CF/DPF to retain its ability to handle non-Gaussian forcing and noise terms.

## 3. MODIFIED FUSION FILTER

In the CF/DPF, the local filters and the fusion filters can run out of synchronization due to intermittent network connectivity. The local filters are confined to their sensor node and unaffected by loss of connectivity. The fusion filters, on the other hand, run consensus algorithms. The convergence of these consensus algorithms is delayed if the communication bandwidth is reduced. In this section, we develop ways of dealing with such intermittent connectivity issues. First, let us introduce the notation. We assume that the observations arrive at constant time intervals of  $\Delta T$ . Each iteration of the local filters is performed within this interval, which we will refer to as the local filter's estimation interval. The duration (the fusion filter's estimation interval) of the update cycle of the fusion filter is denoted by  $T_c$ . Fig. 1 illustrates two scenarios dealing with different fusion filter's estimation intervals. Fig. 1(a) is the ideal scenario where  $T_c \leq \Delta T$  and the fusion filter's consensus step converges before the new iteration of the local filter. The timing diagram of the local filter is shown in the bottom subplot of Fig. 1(a) and the timing diagram of the fusion filter is shown in the top subplot of Fig. 1(a). In such a scenario, the local and fusion filters stay synchronized. Fig. 1(b) considers a more problematic scenario when  $T_c > \Delta T$ . Even with ideal connectivity, the fusion filter will continue to lag the localized filters with no hope of it catching up. The bottom two timing diagrams corresponding to the local and fusion filters in Fig. 1(b)



**Fig. 1.** Multirate implementations for the local and fusion filters: (a) ( $T_c \leq \Delta T$ ); (b) ( $T_c > \Delta T$ ). In (b), the CF/DPF lag grows exponentially.

refer to this scenario with  $T_c = 2\Delta T$ . As illustrated, the lag between the fusion and localized filters grows exponentially with time in this scenario. An improvement to the fusion filter is suggested in the top subplot of Fig. 1(b), where the modified fusion filter derives the global posterior at every alternate (1, 3, 5, ...) iteration of the local filter by using the most recently available local filtering density of the localized filters. This allows the fusion filter to catch up with the localized filter even for cases  $T_c > \Delta T$ . Such a modified fusion implementation requires an updated fusion rule for the global posterior density as fusion is not conducted at every local filter iteration. This is considered next in the following discussion.

In our explanation, we use the notation that the fusion filter takes up to  $m$  iterations of the local filters to converge. At iteration  $k + m$ , this implies that node  $l$ , for ( $1 \leq l \leq N$ ), has a particle-based approximation of the local filtering distributions  $P^{(l)}(\mathbf{x}(k + m)|\mathbf{z}^{(l)}(1:k + m))$ , while its fusion filter has a particle-based approximation of the global posterior distribution  $P(\mathbf{x}(0:k)|\mathbf{z}(1:k))$  for iteration  $k$ . In the conventional fusion filter, the statistics of  $P^{(l)}(\mathbf{x}(k+1)|\mathbf{z}^{(l)}(1:k+1))$ , for ( $1 \leq l \leq N$ ), are used in the next consensus step of the fusion filter which then computes the global posterior  $P(\mathbf{x}(0:k+1)|\mathbf{z}(1:k+1))$  based on Eq. (9). The modified fusion filter uses the most recent local filtering distributions  $P^{(l)}(\mathbf{x}(k+m)|\mathbf{z}^{(l)}(1:k+m))$  according to Theorem 1.

**Theorem 1.** Conditioned on the state variables, assume that the observations made at node  $l$  are independent of the observations made at node  $j$ , ( $j \neq l$ ). The global posterior distribution for a  $N$ -sensor network at iteration  $k+m$  is then given by

$$P(\mathbf{x}(0:k+m)|\mathbf{z}(1:k+m)) \propto \prod_{l=1}^N \frac{\prod_{k'=k+1}^{k+m} P^{(l)}(\mathbf{x}(k')|\mathbf{z}^{(l)}(1:k'))}{\prod_{k'=k+1}^{k+m} P^{(l)}(\mathbf{x}(k')|\mathbf{z}^{(l)}(1:k'-1))} \times \prod_{k'=k+1}^{k+m} P(\mathbf{x}(k')|\mathbf{x}(k'-1)) \times P(\mathbf{x}(0:k)|\mathbf{z}(1:k)). \quad (12)$$

*Proof.* Theorem 1 is obtained using: (i) The Markovian property of the state variables; (ii) Assuming that the local observations made at two sensor nodes conditioned on the state variables are independent of each other, and; (iii) Using the Bayes' rule. First, by applying the Bayes' rule to Eq. (9), the posterior distribution at iteration  $k$  can be represented as follows

$$P(\mathbf{x}(0:k)|\mathbf{z}(1:k)) \propto P(\mathbf{z}(k)|\mathbf{x}(k))P(\mathbf{x}(0:k)|\mathbf{z}(1:k-1)) \quad (13)$$

Now, using the Markovian property of the state variables, and assuming that the local observations made at two sensor nodes conditioned on the state variables are independent of each other (13) becomes

$$P(\mathbf{x}(0:k)|\mathbf{z}(1:k)) \propto \left( \prod_{l=1}^N P^{(l)}(\mathbf{z}^{(l)}(k)|\mathbf{x}(k)) \right) \times P(\mathbf{x}(k)|\mathbf{x}(k-1))P(\mathbf{x}(0:k-1)|\mathbf{z}(1:k-1)). \quad (14)$$

At node  $l$ , for ( $1 \leq l \leq N$ ), using the Bayes' rule

$$P^{(l)}(\mathbf{z}^{(l)}(k)|\mathbf{x}(k)) = \frac{P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k))}{P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k-1))} \times P^{(l)}(\mathbf{z}^{(l)}(k)|\mathbf{z}^{(l)}(1:k-1)). \quad (15)$$

Next, we write the posterior density at iteration  $k+m$ , i.e.,

$$P(\mathbf{x}(0:k+m)|\mathbf{z}(1:k+m)) \propto P(\mathbf{x}(0:k+m)|\mathbf{z}(1:k+m-1)) \times \frac{\prod_{l=1}^N P^{(l)}(\mathbf{x}(k+m)|\mathbf{z}^{(l)}(1:k+m))}{\prod_{l=1}^N P^{(l)}(\mathbf{x}(k+m)|\mathbf{z}^{(l)}(1:k+m-1))}. \quad (16)$$

Then the first term on the right hand side of (16) is factorized as

$$P(\mathbf{x}(0:k+m)|\mathbf{z}(1:k+m-1)) = P(\mathbf{x}(k+m)|\mathbf{x}(k+m-1)) P(\mathbf{x}(0:k+m-1)|\mathbf{z}(1:k+m-1)). \quad (17)$$

As in Eq. (16), we continue to expand  $P(\mathbf{x}(0:k+m-1)|\mathbf{z}(1:k+m-1))$  (i.e., the posterior distribution at iteration  $k+m-1$ ) all the way back to iteration  $k+1$  to prove Eq. (12).  $\square$

In the consensus step, two average consensus algorithms compute

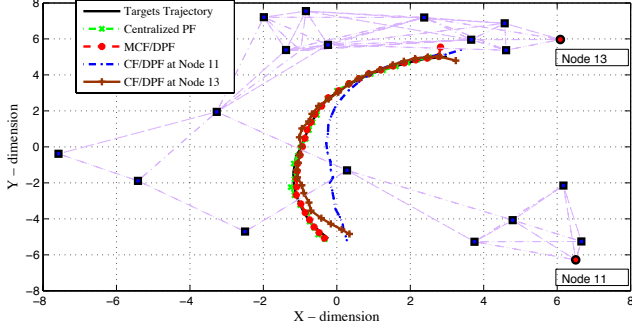
$$\prod_{l=1}^N \prod_{k'=k+1}^{k+m} P^{(l)}(\mathbf{x}(k')|\mathbf{z}^{(l)}(1:k')) \propto \quad (18)$$

$$\prod_{l=1}^N \mathcal{N}(\boldsymbol{\mu}^{(l)}(k+1:k+m), \mathbf{P}^{(l)}(k+1:k+m))$$

and

$$\prod_{l=1}^N \prod_{k'=k+1}^{k+m} P^{(l)}(\mathbf{x}(k')|\mathbf{z}^{(l)}(1:k'-1)) \propto \quad (19)$$

$$\prod_{l=1}^N \mathcal{N}(\mathbf{v}^{(l)}(k+1:k+m), \mathbf{R}^{(l)}(k+1:k+m)),$$



**Fig. 2.** Robot trajectories estimated from the centralized, MCF/DPF, and CF/DPF implementations in a 20-node network. Trajectories from centralized and MCF/DPF overlap and are indistinguishable.

instead of computing  $\prod_{l=1}^N P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k))$  and  $\prod_{l=1}^N P^{(l)}(\mathbf{x}(k)|\mathbf{z}^{(l)}(1:k-1))$  as was the case for the conventional fusion filter. The modified fusion filter starts with a set of particles  $\mathbb{X}_i^{(\text{MFF},l)}(k)$ ,  $W_i^{(\text{MFF},l)}(k)$  approximating  $P(\mathbf{x}(0:k)|\mathbf{z}(1:k))$  and generates updated particles  $\mathbb{X}_i^{(\text{MFF},l)}(k+m)$ ,  $W_i^{(\text{MFF},l)}(k+m)$  for  $P(\mathbf{x}(0:k+m)|\mathbf{z}(1:k+m))$  using the following weight update equation

$$W_i^{(l,\text{MFF})}(k+m) \propto W_i^{(l,\text{MFF})}(k) \times \frac{\prod_{k'=k+1}^{k+m} P^{(l)}(\mathbb{X}_i^{(l,\text{MFF})}(k')|\mathbb{X}_i^{(l,\text{MFF})}(k'-1))}{\mathcal{N}(\mathbb{X}_i^{(l,\text{MFF})}(k+m); \mathbf{v}(k+1:k+m), \mathbf{R}(k+1:k+m))}, \quad (20)$$

which is obtained directly from (12). Note that the normal approximation in Eqs. (18)-(20) is similar to the one used in the conventional fusion filter. Furthermore, we note that the modification requires prediction of the particles from iteration  $k$  all the way to  $k+m$  to evaluate the second term on the right hand side of (20).

#### 4. EXPERIMENTAL RESULTS

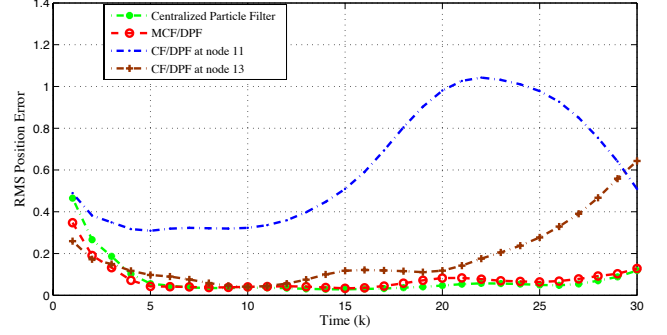
In this paper, we consider a distributed mobile robot localization problem [26, 40] based on angle-only measurements [39]. This is a good benchmark since the underlying dynamics is nonlinear and the state is unobservable to individual sensors, which justifies the need for some communication between them. The state variables of the unicycle robot is defined as  $\mathbf{x} = [X, Y, \theta]^T$ , where  $X$  and  $Y$  are 2D coordinates of the robot while  $\theta$  is its orientation. The linear velocity and angular velocity are denoted by  $\tilde{V}(k)$  and  $\tilde{\Omega}(k)$ , respectively. The nonlinear, time-invariant dynamical model for the robot follows the following discrete-time unicycle model [40]

$$X(k+1) = X(k) + \frac{\tilde{V}(k)}{\tilde{\Omega}(k)} \left( \sin(\theta(k) + \tilde{\Omega}(k)\Delta T) - \sin(\theta(k)) \right) \quad (21)$$

$$Y(k+1) = Y(k) + \frac{\tilde{V}(k)}{\tilde{\Omega}(k)} \left( \cos(\theta(k) + \tilde{\Omega}(k)\Delta T) - \cos(\theta(k)) \right) \quad (22)$$

$$\theta(k+1) = \theta(k) + \tilde{\Omega}(k)\Delta T + \xi_\theta \Delta T, \quad (23)$$

with  $\Delta T$  the sampling time and  $\xi_\theta$  the orientation noise term. Terms  $\tilde{\Omega}(k) = \Omega(k) + \xi_\Omega(k)$  and  $\tilde{V}(k) = V(k) + \xi_v(k)$ , where  $\{\xi_v, \xi_\Omega\}$  are the noise terms in these control inputs. The design parameters are:  $\Delta T = 1$  and mean velocity of 30 cm/s with a standard deviation of 5 cm/s. The random noises are Gaussian, i.e.,  $\xi_v(k) \sim$



**Fig. 3.** RMS errors from Monte Carlo simulations for the centralized, CF/DPF (fusion filter's outputs from two randomly selected nodes  $\{11, 13\}$  since CF/DPF does not converge), and MCF/DPF.

$\mathcal{N}(0.3, .05)$ ,  $\xi_\Omega(k) \sim \mathcal{N}(0.08, .01)$  and  $\xi_\theta \sim \mathcal{N}(0, .01)$ . Measurements at each node  $l$  in a 20-node network are the target's bearings

$$Z^{(l)}(k) = \text{atan} \left( \frac{X(k) - X^{(l)}}{Y(k) - Y^{(l)}} \right) + \zeta^{(l)}(k), \quad (24)$$

where  $(X^{(l)}, Y^{(l)})$  are the coordinates of node  $l$ . Three particle filter implementations are tested: (i) Centralized approach (benchmark); (ii) CF/DPF (since consensus is not reached, the fusion estimates in the CF/DPF are different from one node to another. Results from nodes  $\{11, 13\}$  are included), and; (iii) MCF/DPF. We consider  $N_s = 20,000$  particles in the centralized implementation and  $N_s = 1000$  particles at each node for implementations (ii)-(iii) to keep the total particles same in (i) to (iii). Fig. 2 plots one realization of the sensor placement along with the estimated robot's trajectories obtained from implementations (i) to (iii). In (ii)-(iii), the fusion filters take up to two iterations of the localized particle filters. The CF/DPF does not reach consensus, while the MF/DPF performs consensus at alternate iterations of the local filters. Fig. 2 illustrates that the estimates of the modified fusion filter are much closer to its centralized counterpart. Fig. 3 shows the RMS error curves for the target's position based on a Monte Carlo simulation of 100 runs. The performance of the proposed MCF/DPF remains close to its centralized counterpart. In other nonlinear tracking scenarios with non-Gaussian forcings that we tested, the MCF/DPF successfully handled intermittence in network connectivity with performance close to its centralized counterpart with no intermittance issues.

#### 5. SUMMARY

The paper proposes a distributed estimation framework for intermittently connected networks with nonlinear dynamics. To illustrate the framework, a consensus-based distributed particle filter implementation, MCF/DPF, is implemented to cope with intermittent communication connectivity. At each node, the MCF/DPF runs two particle filters: (a) The local filter is based only on the node's local observations and recursively derives the local estimates, and; (b) The modified fusion filter (similar in concept to the channel filter [41]) extracts new information from the local estimates obtained from the neighboring nodes. The MCF/DPF allows the modified fusion filter to run at a rate different from that of the local filters, therefore, enabling the consensus step to converge without strict time limitations. Future work will consider multitarget tracking, which is likely to introduce a multinomial posterior.

## 6. REFERENCES

- [1] 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 Trans. on Signal Proc.*, vol. 50, pp. 174-187, 2002.
- [3] M.E. Liggins II, C-Y Chong, I. Kadar, M.G. Alford, V. Vannicola, and S. Thomopoulos. "Distributed fusion architectures and algorithms for target tracking," in *Proc. of the IEEE*, vol. 85, no. 1, pp. 95-107, 1997.
- [4] S. Kar, B. Sinopoli, and J. Moura, "Kalman Filtering with Intermittent Observations: Weak Convergence to a Stationary Distribution," to appear in *IEEE Transactions on Automatic Control*.
- [5] R. Olfati-Saber, A. Fax, and R.M. Murry, "Consensus and cooperation in networked multi-agent systems," in *Proceedings of the IEEE*, 2007.
- [6] 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.
- [7] C.Y. Chong, S. Mori, and K.C. Chang, *Multi-target Multi-sensor Tracking*, chapter Distributed Multi-target Multi-sensor Tracking, Artech House, pp. 248-295, 1990.
- [8] 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.
- [9] 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," *Signal Process., IEEE Trans.*, vol. 60, no. 9, pp. 4719-4733, Sept. 2012.
- [10] 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.
- [11] T. Zhao and A. Nehorai, "Distributed Sequential Bayesian Estimation of a Diffusive Source in Wireless Sensor Networks," *IEEE Tran. Sig. Proc.*, vol. 55, no. 4, pp. 1511-1524, 2007.
- [12] F. Sawo "Nonlinear State and Parameter Estimation of Spatially Distributed Systems," *PhD thesis*, University of Karlsruhe, 2009.
- [13] T.V. Waterschoot and G. Leus, "Distributed estimation of static fields in wireless sensor networks using the finite element method",
- [14] H. Pasula, S. Russell, M. Ostland, and Y. Ritov, "Tracking many objects with many sensors," in *Proc. 12'th Int. Joint Conference on Artificial Intelligence*, 1999.
- [15] M. Bolic, P.M. Djuric, and Sangjin Hong, "Resampling algorithms and architectures for distributed particle filters," In *IEEE Tran. on Sig. Proc.*, vol. 53, no. 7, pp. 2442-2450, 2005.
- [16] B. Balasingam, M. Bolic, P.M. Djuric, and J. Miguez "Efficient distributed resampling for particle filters," In *IEEE Int. Con. on Acoustics, Speech and Sig. Proc.*, pp. 3772-3775, 2011.
- [17] S. Challa, M. Palaniswami and A. Shilton, "Distributed data fusion using support vector machines," in *Fifth International Conference on Information Fusion, (FUSION)*, vol. 2, p. 881-885, 2002.
- [18] M. Rosencrantz, G. Gordon, and S. Thrun, "Decentralized data fusion with distributed particle filters," in *Conference on Uncertainty in AI (UAI)*, 2003.
- [19] M. Coates, "Distributed particle filters for sensor networks," *ISPN Sensor Networks*, pp. 99-107, 2003.
- [20] X. Sheng, Y. Hu, and P. Ramanathan, "GMM approximation for multiple targets localization and tracking in wireless sensor network," in *International Symposium on Information Processing in Sensor Networks*, pp. 181-188, 2005.
- [21] A. Mohammadi and A. Asif, "Consensus-based Particle Filter Implementations for Distributed Non-linear Systems", in *Nonlinear Est. & Applications to Industrial Sys. Control*, Editor G. Rigatos, 2011.
- [22] A. Mohammadi and A. Asif, "A Consensus/Fusion based Distributed Implementation of the Particle Filter," *IEEE CAMSAP*, 2011.
- [23] A. Mohammadi and A. Asif, "Distributed Particle Filters for Intermittent Connections: Feedback Between Fusion and Local Filters improves Performance," *IEEE SSP*, 2012.
- [24] A. Mohammadi and A. Asif, "Distributed State Estimation for Large-scale Nonlinear Systems: A Reduced Order Particle Filter Implementation," *IEEE SSP*, 2012.
- [25] A. Mohammadi and A. Asif, "Consensus-Based Distributed Unscented Particle Filter," in *IEEE SSP*, pp. 237-240, 2011.
- [26] A. Simonetto, T. Keviczky, R. Babuska, "Distributed Non-linear Estimation for Robot Localization using Weighted Consensus," In *IEEE Inter. Con. on Robotics and Automation*, pp. 3026-3031, 2010.
- [27] M.J. Coates and B.N. Oreshkin, "Asynchronous distributed particle filter via decentralized evaluation of gaussian products," in *Proc. ISIF Int. Conf. Info. Fusion*, 2010.
- [28] D. Gu, S. Junxi, H. Zhen and L. Hongzuo, "Consensus based distributed particle filter in sensor networks," in *IEEE Int. Con. on Information and Automation*, pp. 302-307, 2008.
- [29] A. Mohammadi and A. Asif, "A Constraint Sufficient Statistics based Distributed Particle Filter for Bearing Only Tracking", *IEEE international conference on communications (ICC)*, 2012.
- [30] O. Hlinka, F. Hlawatsch, P.M. Djuric, "Likelihood consensus-based distributed particle filtering with distributed proposal density adaptation," *IEEE ICASSP*, pp. 3869-3872, 2012.
- [31] O. Hlinka, O. Sluciak, F. Hlawatsch, P.M. Djuric, M. Rupp, "Distributed Gaussian particle filtering using likelihood consensus," *IEEE ICASSP*, pp. 3756-3759, 2011.
- [32] D. Ustebay, M. Coates, and M. Rabbat, "Distributed auxiliary particle filters using selective gossip," In *IEEE ICASSP*, pp. 3296-3299, 2011.
- [33] C.J. Bordin, M.G.S. Bruno, "Consensus-based distributed particle filtering algorithms for cooperative blind equalization in receiver networks," *ICASSP 2011*, pp. 3968-3971, 2011.
- [34] Sh. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, "Particle filter adaptation for distributed sensors via set membership," in *Int. Con. on Acoustics, Speech, & Sig. Proc.*, pp. 3374-3377, 2010.
- [35] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, "Set-membership constrained particle filter: Distributed adaptation for sensor networks," *IEEE Trans. on Signal Processing*, vol. 59, no. 9, pp. 4122-4138, Sept. 2011.
- [36] D. Gu, "Distributed particle filter for target tracking," in *IEEE Int. Conference on Robotics and Automation*, pp. 3856-3861, 2007.
- [37] 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.
- [38] Y. Zhu, Z. You, J. Zhao, K. Zhang, and X. Li, "The optimality for the distributed Kalman filtering fusion with feedback", *Automatica*, vol. 37, no. 9, pp. 1489-1493, 2001.
- [39] M. S. Arulampalam, B. Ristic, N. Gordon, and T. Mansell, "Bearings-only tracking of maneuvering targets using particle filters," *EURASIP Appl. Sig. Process.*, vol. 15, pp. 2351-2365, 2004.
- [40] S. Thrun, W. Burgard, and D. Fox, "Probabilistic Robotics," *The MIT Press*, 2005.
- [41] S. Grime and H. F. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Cont. Eng. Prac.*, pp. 849-863, 1994.
- [42] A. Mohammadi and A. Asif, Distributed Particle "Filter Implementation with Intermittent/Irregular Consensus Convergence," accepted in *IEEE Tran. on Sig. Proc.*, 2013. Manuscript of 30 pages.