

Set-Membership Constrained Particle Filter: Distributed Adaptation for Sensor Networks

Shahrokh Farahmand, Stergios I. Roumeliotis, *Member, IEEE*, and Georgios B. Giannakis, *Fellow, IEEE*

Abstract—Target tracking is investigated using particle filtering of data collected by distributed sensors. In lieu of a fusion center, local measurements must be disseminated across the network for each sensor to implement a centralized particle filter (PF). However, disseminating raw measurements incurs formidable communication overhead as large volumes of data are collected by the sensors. To reduce this overhead and thus enable distributed PF implementation, the present paper develops a set-membership constrained (SMC) PF approach that: i) exhibits performance comparable to the centralized PF; ii) requires only communication of particle weights among neighboring sensors; and iii) can afford both consensus-based and incremental averaging implementations. These attractive attributes are effected through a novel adaptation scheme, which is amenable to simple distributed implementation using min- and max-consensus iterations. The resultant SMC-PF exhibits high gain over the bootstrap PF when the likelihood is peaky, but not in the tail of the prior. Simulations corroborate that for a fixed number of particles, and subject to peaky likelihood conditions, SMC-PF outperforms the bootstrap PF, as well as recently developed distributed PF algorithms, by a wide margin.

Index Terms—Particle filter, sensor network, adaptation, distributed, set-membership.

I. INTRODUCTION

Consider collaborating agents (e.g., robots) equipped with wireless sensors measuring distance and/or bearing from a target that they wish to track. The nonlinearities present in these measurements prevent sensors from employing the clairvoyant (linear) Kalman tracker. Linearized trackers such as the extended Kalman filter (EKF) [30, Chap. 13] or the unscented Kalman filter (UKF) [28], [31], have been typically used to approximate the minimum mean-square error (MMSE) state estimator. However, high-variance measurement and/or state noise combined with the aforementioned nonlinearities lead such linearized trackers to inconsistent estimates [2]. As an alternative, deterministic numerical methods can be

employed to evaluate the integrals associated with MMSE state estimates per time step, and yield accurate results [27]. Unfortunately, their complexity increases exponentially with the problem dimension.

Trading-off accuracy for complexity, non-parametric, non-linear estimators based on the particle filter (PF), improve estimation accuracy while maintaining a reasonable computational burden. In addition to complexity, when sensors are deployed to perform decentralized tracking, coping with the inter-sensor communication overhead presents an extra challenge. Especially when a fusion center (FC) is not available, sensors have to share raw measurements via flooding, which renders the communication overhead prohibitively high. These considerations motivate the context and objective of this paper, which is distributed PF with affordable complexity, and reduced overhead by communicating processed (as opposed to raw) measurements among neighboring sensors.

Tutorial treatment of PFs can be found in [6], [13], and [15]; see also [12], [38] for FC-based approaches. In-network (non-FC based) PF trackers are reported in [1], [5], [7], [19], [23], [24], [25], [32], [34], [36], [40], [42] and [43]. Specifically, local PFs are run at individual sensors in [1], and their estimates are spatially smoothed using a consensus filter. A query-response approach is advocated by [40] to exchange measurements among distributed robots. While easy to implement, the distributed schemes in [1] and [40] are ad-hoc, and can not perform close to a centralized PF tracker. Assuming a uniform prior, [5] adopts a mixture of local posteriors as the PF importance density, but does not account for past measurements in the current state estimate, which is tantamount to sub-optimality. Both [19] and [43] use a Gaussian mixture model (GMM) to approximate the posterior. They apply principal component analysis successively in an incremental loop to update GMM parameters. These two approaches as well as those in [7], [25] and [42] belong to the class of incremental algorithms that rely on GMM or similar parametric models to communicate partial posteriors or likelihoods.

Specifically, [7], [25] and [42] approximate the centralized PF by sequential training of a parametric model. While [25] and [42] sequentially approximate the posterior of interest by a GMM, whose parameters are communicated incrementally from sensor to sensor, [7] sequentially trains a parametric model to match the likelihood. To further improve performance, [25] incorporates measurements from the current sensor along with measurements from the next sensor which is transmitted back to the current sensor in its importance sampling step to generate more efficient particles. All these

Manuscript received July 19, 2010; revised February 9, 2011; accepted June 1, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Andrea Cavallaro. Work in this paper was supported through collaborative participation in the C&N Consortium sponsored by the U. S. ARL under the CTA Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. Support was also provided by the National Science Foundation (IIS-0643680). Copyright (c) 2011 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

S. Farahmand and G. B. Giannakis are with the Dept. of Electrical and Computer Engineering, Univ. of Minnesota, Minneapolis, MN, 55455 USA. e-mails: {shahrokh,georgios}@umn.edu; S. I. Roumeliotis is with the Dept. of Computer Science, Univ. of Minnesota, Minneapolis, MN, 55455 USA. e-mail: stergios@cs.umn.edu.

approaches have the following limitations: i) they require establishment of a Hamiltonian cycle that goes through all sensors in the network, which is an NP-hard problem; ii) affordable sub-optimal paths used in lieu of a Hamiltonian cycle can miss some of the sensors and their data; iii) they incur excessive delays due to sequential GMM training; and iv) they lack robustness to sensor failures e.g., if the sensor who is currently processing fails, the whole estimation task is compromised.

Such limitations can be avoided if consensus-based approaches are utilized instead [1], [23], [24], [34], [36]. In [23], sensor data are used to train a GMM in a distributed fashion based on which particles are generated per sensor. In [24], sensors consent on the average mean and covariance of their local particle filters and use them to generate particles. Support vector machines were utilized in [34] to consent on the average of local functions yielding a reduced subset of particles. Being suboptimal, the performance of [1], [23], [24], [34] cannot approach the performance of centralized PF. Re-formulating the posterior as the geometric mean of locally known Gaussian terms, [36] invokes consensus to make the parameters of the resultant Gaussian posterior available across sensors. While this approach has potential to approach the performance of centralized PF, it cannot cope with multi-modal posteriors.

Another major challenge in distributed PF is whether to communicate a GMM approximation of the posterior density or raw particles and weights. For low-dimensional data with a small number of modes, GMMs are reasonable. However, communicating GMMs amounts to transmitting covariance matrices which are proportional to the square of the state dimension. Therefore, GMMs can become costly for problems with large dimensions or a large number of peaks which can arise in multi-target tracking scenarios. Under such conditions, [32] proposes a Markov chain based approach, where particles and weights are communicated among sensors. The communication of such schemes can be greatly reduced if one can afford to communicate particle weights (that are scalars) only, and not the particles themselves. It will be demonstrated that this is possible at the cost of enforcing synchronism among sensors. The first contribution of this work is a synchronous non-parametric distributed PF operating either in an incremental- or a consensus-based mode. The proposed distributed PF can approach the performance of centralized PF, and requires communication of particle weights only.

Affordable inter-sensor communications are enabled through a novel distributed adaptation scheme, which considerably reduces the number of particles needed to achieve a given performance. Adaptation amounts to taking into account the current measurements in the importance density employed by the PF. The particles drawn according to a data-adapted importance density are more efficient, meaning they represent the posterior more accurately than particles drawn from a non-adapted density. Hence, one can afford fewer particles while alleviating the particle depletion problem.

Many works are available on adaptation methods for PFs [11], [14], [16], [18], [21], [22], [29], [35], [37], [39], [45]. However, none is developed to ensure affordable distributed

implementation. In the centralized PF setting, existing approaches fall under two categories. The first includes parametric schemes which fit a Gaussian to the true posterior, and aim to find the associated mean and covariance. The latter can be obtained by an EKF iteration [14] or a UKF iteration [35], after equating terms in a Taylor series expansion of the unnormalized posterior [37], or, via support vector regression [29]. Application-specific methods, such as the one in [4] for visual tracking, also fall under the first category. On the other hand, prior-editing [22], likelihood sampling [18] and its modifications [45], belong to the second category. Additional non-parametric approaches here include the particle flow using log-homotopy [11], where particles representing the prior are deterministically migrated to the region where the posterior has most of its probability mass; and its “stochastic” counterparts, which rely on a combination of bridging densities, and either Markov chain Monte Carlo methods [21], or, adaptive importance sampling [16]. The auxiliary PF benefits from adaptation as well [39]. The second contribution of the present paper is a novel adaptation method that does not assume Gaussian posteriors. From this vantage point, it belongs to the second category of centralized adaptation methods, but is particularly attractive for distributed implementation.

The rest of the paper is organized as follows. Section II outlines the basics of PF to establish context and notation, before formulating the problem of interest. Section III deals with distributed PF. The novel adaptation method is developed in Section IV, followed by performance analysis in Section V. Implementation issues are discussed in Section VI, and corroborating simulations are presented in Section VII. Section VIII concludes the paper.

II. PRELIMINARIES AND PROBLEM STATEMENT

Consider a network of N sensors distributed randomly on a field measuring their distance and/or bearing from a moving object. At time k , the object’s state vector $\mathbf{x}_k \in \mathbb{R}^{d_x}$ evolves as

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_k)$$

where \mathbf{w}_k denotes the state noise of known distribution, which dictates the predictor density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Sensor n measures the vector

$$\mathbf{y}_k^n = \mathbf{h}^n(\mathbf{x}_k) + \mathbf{v}_k^n$$

where $\mathbf{v}_k^n \in \mathbb{R}^{d_y}$ stands for measurement noise of known distribution, giving rise to the likelihood $p(\mathbf{y}_k^n|\mathbf{x}_k)$. Functions \mathbf{f} and \mathbf{h} are generally nonlinear, and \mathbf{v}_k^n is assumed independent across sensors. The network is connected, meaning that there is a (perhaps multi-hop) path connecting any two sensors.

Based on its own measurements $\{\mathbf{y}_k^n\}$ as well as those of others percolated through inter-sensor communications, sensor n wishes to obtain the MMSE optimal estimate of the state, namely the posterior mean $\hat{\mathbf{x}}_k := E[\mathbf{x}_k|\mathbf{y}_{1:k}]$, and its error covariance $\hat{\mathbf{C}}_k := E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T|\mathbf{y}_{1:k}]$, where $\mathbf{y}_{1:k} := [\mathbf{y}_1^T, \dots, \mathbf{y}_k^T]^T$ and $\mathbf{y}_k := [(\mathbf{y}_k^1)^T, \dots, (\mathbf{y}_k^N)^T]^T$. Treating these estimates together for brevity, the goal is to form in a distributed fashion the MMSE estimate of a function

$\phi(\mathbf{x}_k) : \mathbb{R}^{d_x} \rightarrow \mathbb{R}$, given by

$$I_\phi(k) := E[\phi(\mathbf{x}_k)|\mathbf{y}_{1:k}] = \int \phi(\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k}) d\mathbf{x}_k. \quad (1)$$

Clearly, (1) yields as a special case¹ $\hat{\mathbf{x}}_k$ when $\phi(\mathbf{x}_k) := \mathbf{x}_k(i)$, with $\mathbf{x}_k(i)$ denoting the i th entry of \mathbf{x}_k ; and also its error covariance entries $\hat{\mathbf{C}}_k(i, j)$ when $\phi(\mathbf{x}_k) := [\mathbf{x}_k(i) - \hat{\mathbf{x}}_k(i)][\mathbf{x}_k(j) - \hat{\mathbf{x}}_k(j)]$.

Since the likelihood and the predictor density are known, Bayes' rule allows in principle for recursive evaluation of the wanted posterior as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{\int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k} \quad (2a)$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}. \quad (2b)$$

Unfortunately, nonlinearity (and hence non-Gaussianity) of the processes involved render it impossible to evaluate exactly the integrals in (2). Among the available approximations, a popular one pursued here relies on particle filtering (PF) – the sequential version of importance sampling (IS) [3, Chapter 11].

IS refers to the process of estimating $E_{q_t}[\phi(\mathbf{x})]$, where \mathbf{x} is distributed according to a target (but inconvenient to sample from) density $q_t(\mathbf{x})$, based on samples (a.k.a. particles) $\{\mathbf{x}^{(m)}\}_{m=1}^M$ drawn from a surrogate density $q_{IS}(\mathbf{x})$, which is selected so that samples from it are “of importance” to capture $q_t(\mathbf{x})$ too. The resultant consistent estimate is a weighted average (with scale-invariant weights) given by

$$\hat{I}_\phi := \sum_{m=1}^M \bar{w}^{(m)} \phi(\mathbf{x}^{(m)}), \quad \bar{w}^{(m)} = \frac{w^{(m)}}{\sum_{m=1}^M w^{(m)}} \quad (3)$$

$$w^{(m)} = \frac{q_t(\mathbf{x}^{(m)})}{q_{IS}(\mathbf{x}^{(m)})}, \quad \forall m = 1, \dots, M.$$

Since densities can be expressed via moments, it turns out the same IS weights can be employed to estimate q_t as $\hat{q}_t(\mathbf{x}) = \sum_{m=1}^M \bar{w}^{(m)} \delta(\mathbf{x} - \mathbf{x}^{(m)})$, where $\delta(\cdot)$ denotes Dirac's delta. With identical IS weights one can also estimate marginals of q_t . As far as performance, it is known that the variance of \hat{I}_ϕ in (3) is minimized when the surrogate is selected as $q_{IS}(\mathbf{x}) \propto |\phi(\mathbf{x}) - E_{q_t}[\phi(\mathbf{x})]| q_t(\mathbf{x})$ [20]. This choice is obviously challenging to sample from, and also ϕ -dependent. The ϕ -independent selection ($q_{IS} \propto q_t$) on the other hand, emerges from a tractable approximation of the minimum IS variance given by [33]

$$\begin{aligned} \text{Var}_{q_{IS}} &:= E \left[\left(\hat{I}_\phi - E_{q_t}[\phi(\mathbf{x})] \right)^2 \right] \\ &\approx (1/M) \text{Var}_{q_t}(\phi(\mathbf{x})) E_{q_{IS}} \left[\left(\frac{q_t(\mathbf{x})}{q_{IS}(\mathbf{x})} \right)^2 \right]. \quad (4) \end{aligned}$$

Returning to PF, online estimation of $I_\phi(k)$ in (1) amounts to sequential IS with $q_t(\mathbf{x}_k) := p(\mathbf{x}_k|\mathbf{y}_{1:k})$. In fact, it is convenient to select as target density the joint posterior $q_t(\mathbf{x}_k) := p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, bearing in mind that identical weights can be

employed to estimate the marginal density of interest. With regards to the IS density $q_{IS}(\mathbf{x}_{0:k})$, consider the following convenient factorization of the joint posterior

$$\begin{aligned} p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) &= \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{\pi(\mathbf{x}_k|\mathbf{y}_k, \mathbf{x}_{k-1})} \\ &\times \frac{\pi(\mathbf{x}_k|\mathbf{y}_k, \mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \quad (5) \end{aligned}$$

where π here denotes an arbitrary density function (thus integrates to 1), for which both ratios in the right hand side (r.h.s.) of (5) are bounded away from infinity.

Suppose that at time step k , the particles and weights $\{\bar{w}_{k-1}^{(m)}, \mathbf{x}_{0:k-1}^{(m)}\}_{m=1}^M$ are available from step $k-1$. With $q_t(\mathbf{x}_{0:k}) := p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ and $q_{IS}(\mathbf{x}_{0:k}) := \pi(\mathbf{x}_k|\mathbf{y}_k, \mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$, the IS scheme at time k draws samples $\{\mathbf{x}_{0:k}^{(m)}\}_{m=1}^M$ from $q_{IS}(\mathbf{x}_{0:k})$, forms un-normalized weights $w_k^{(m)} = p(\mathbf{y}_k|\mathbf{x}_k^{(m)})p(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})/\pi(\mathbf{x}_k^{(m)}|\mathbf{y}_k, \mathbf{x}_{k-1}^{(m)})$ as well as their normalized versions $\{\bar{w}_k^{(m)}\}_{m=1}^M$, and relies on the set $\{\bar{w}_k^{(m)}, \mathbf{x}_{0:k}^{(m)}\}_{m=1}^M$ to estimate $I_\phi(k)$ as in (3). To draw samples from the chosen $q_{IS}(\mathbf{x}_{0:k})$ requires the estimate

$$\hat{p}(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) = \sum_{m=1}^M \bar{w}_{k-1}^{(m)} \delta(\mathbf{x}_{0:k-1} - \mathbf{x}_{0:k-1}^{(m)}). \quad (6)$$

With (6) available from time step $k-1$, the IS density approximant to sample from at step k is

$$\begin{aligned} \pi(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k) \hat{p}(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1}) \\ = \sum_{m=1}^M \bar{w}_{k-1}^{(m)} \pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) \delta(\mathbf{x}_{0:k-1} - \mathbf{x}_{0:k-1}^{(m)}) \quad (7) \end{aligned}$$

which amounts to re-sampling from $\{\bar{w}_{k-1}^{(m)}, \mathbf{x}_{0:k-1}^{(m)}\}$, followed by particle augmentation by sampling from $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$. While the product $\pi(\mathbf{x}_k|\mathbf{y}_k, \mathbf{x}_{k-1})p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$ is the IS density at step k , the factor $\pi(\mathbf{x}_k|\mathbf{y}_k, \mathbf{x}_{k-1})$ used to augment the re-sampled particles, is often referred to as the IS density. Note that there is no need to store the whole trajectory $\{\mathbf{x}_{0:k}^{(m)}\}_{m=1}^M$, but only the current set $\{\mathbf{x}_k^{(m)}\}_{m=1}^M$.

Summarizing, after initialization the PF implements these steps per time instant k ; see e.g., [13].

S1) Re-sample from the particles and weights $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$ available from time $k-1$ (cf. (6), (7));

S2) Draw the new sample $\mathbf{x}_k^{(m)} \sim \pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$ from the chosen (augmenting IS) density;

S3) Find weights through Bayes' rule, and normalize them to sum up to one, as (cf. (3), (5))

$$w_k^{(m)} = \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(m)})p(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)}, \quad \bar{w}_k^{(m)} = \frac{w_k^{(m)}}{\sum_{m=1}^M w_k^{(m)}}; \quad (8)$$

S4) Form $\hat{I}_\phi(k) := \sum_{m=1}^M \bar{w}_k^{(m)} \phi(\mathbf{x}_k^{(m)})$ as in (3), and output current state estimate and its covariance.

Selecting π is a performance-critical issue of the PF algorithm. A popular but not necessarily efficient choice is $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) := p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, which corresponds to the so-called bootstrap PF (B-PF). Setting the IS density equal to the

¹Likewise, any moment and thus the posterior density can be sequentially estimated, in par with the goal of PF.

prior, the B-PF weights in (8) become $w_k^{(m)} = p(\mathbf{y}_k | \mathbf{x}_k^{(m)})$. The attractive feature of B-PF is its simplicity, since it is easy to generate particles from the prior, and have their weights simply given by the likelihood. The optimum π in S2 in the sense of minimizing the variance of $w_k^{(m)}$ is $\pi(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) := p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$, and amounts to choosing the IS density equal to true posterior [14]. Since it is impossible to draw samples from the true posterior, sub-optimal choices of π accounting for the new data \mathbf{y}_k in the IS density are well motivated. They are referred to as data-adapted (or simply adapted) densities, to be differentiated from non-adapted choices such as the one in B-PF. The significance of adaptation in the present framework will become clear in the next section.

The described PF algorithm is *centralized* because it requires knowledge of \mathbf{y}_k to operate. Specifically, to generate particles in S2 and obtain weights in S3, it is necessary to disseminate \mathbf{y}_k throughout the network (e.g., via flooding), which is impractical especially for a large number of sensors N , or, a large size measurement vector (d_y). To cope with this challenge, the ensuing section presents a *distributed* PF that requires exchanging only particle weights, while the follow-up section introduces a novel adaptation scheme based on *set-membership* for reducing the number of particles.

III. DISTRIBUTED PF

At initialization ($k = 0$), each sensor draws new samples from $p(\mathbf{x}_0)$ and weighs them equally. Setting the seed of random number generators at all sensors to the same value ensures identical particles $\mathbf{x}_0^{(m)}$, $\forall m = 1, \dots, M$ generated at all sensors. Next, recall that after step $k - 1$, sensors have available $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$. Since the random number generators at all sensors are initialized with the same seed, re-sampled particles across all sensors will be identical; see also [7]. Thus, S1 can be performed locally per sensor provided that $\{\bar{w}_{k-1}^{(m)}, \mathbf{x}_{k-1}^{(m)}\}_{m=1}^M$ is known at all sensors. In the same spirit, S4 can be run locally provided that $\{\bar{w}_k^{(m)}, \mathbf{x}_k^{(m)}\}_{m=1}^M$ is commonly available to all sensors. If in addition a common non-adapted IS density is utilized (meaning $\pi(\cdot)$ is not dependent on \mathbf{y}_k), then S2 can also be run locally at each sensor. In a nutshell, it is possible to implement S1, S2, and S4 in a distributed fashion.

Focusing on S3, one can invoke the noise independence across sensors, and take logarithms on both sides of (8) to obtain

$$\begin{aligned} \log(w_k^{(m)}) &= \log(p(\mathbf{x}_k^{(m)} | \mathbf{x}_{k-1}^{(m)})) + \sum_{n=1}^N \log(p(\mathbf{y}_k^n | \mathbf{x}_k^{(m)})) \\ &\quad - \log(\pi(\mathbf{x}_k^{(m)} | \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)). \end{aligned} \quad (9)$$

Because sensors have available the particles $\{\mathbf{x}_{k-1}^{(m)}\}_{m=1}^M$ at time $k - 1$ from S1 as well as the newly generated ones $\{\mathbf{x}_k^{(m)}\}_{m=1}^M$ from S2, all terms except the sum on the r.h.s. of (9) are locally known when $\pi(\mathbf{x}_k^{(m)} | \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) = \pi(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)})$. This sum can become available per sensor either through incremental averaging or via consensus averaging, as described next.

Consider that each sensor n in the network has a scalar ψ_n . Through collaborative exchanges all sensors wish to find the sample mean $\bar{\psi}_N := (1/N) \sum_{n=1}^N \psi_n$, when no fusion center is available to receive and centrally average $\{\psi_n\}_{n=1}^N$. For the problem at hand, $\psi_n = \log(p(\mathbf{y}_k^n | \mathbf{x}_k^{(m)}))$. Operation without fusion centers is desirable for scalability, and also because isolated points of failure are avoided.

Incremental averaging relies on passing partial sums over a Hamiltonian path, which goes through every sensor exactly once. With sensors indexed by the order they appear in this Hamiltonian path, the algorithm commences with sensor 1 transmitting ψ_1 to sensor 2, which forms $\psi_1 + \psi_2$ and transmits it to sensor 3. Likewise, sensor n receives the partial sum $\psi_1 + \dots + \psi_{n-1}$ from sensor $n - 1$, adds ψ_n to it and communicates the sum to sensor $n + 1$. The desired $\bar{\psi}_N$ formed at sensor N is then percolated through the Hamiltonian path in order for all sensors to have a copy of the sample average. The incremental averaging scheme converges in finite time. But finding a Hamiltonian path is NP-hard, and requires perfect knowledge of the communication graph at every sensor [46]. Furthermore, the algorithm is not robust because a new Hamiltonian cycle must be established each time a sensor fails.

Alternatively, it is possible to make the sum in (9) available per sensor via consensus averaging, see e.g., [47]. Here sensors do not need to know the communication graph or establish Hamiltonian paths, but only need to communicate with their immediate neighbors. The desired $\bar{\psi}_N$ is obtained per sensor iteratively. After iteration $i - 1$, sensor n broadcasts $\psi_n(i - 1)$ to its neighbors. Having received $\{\psi_\ell(i - 1)\}_{\ell \in \mathcal{N}(n)}$ from its one-hop neighbors denoted by the set $\mathcal{N}(n)$, sensor n updates its iterate as

$$\psi_n(i) = \psi_n(i - 1) + \mu \sum_{\ell \in \mathcal{N}(n)} [\psi_\ell(i - 1) - \psi_n(i - 1)]$$

where $\psi_n(0) = \psi_n$ and μ is a constant stepsize. If μ is chosen small enough, and the communication graph remains connected, the iterates $\psi_n(i)$ converge asymptotically (as $i \rightarrow \infty$) to the desired sample mean [47]; that is, $\lim_{i \rightarrow \infty} \psi_n(i) = \bar{\psi}_N$, $\forall n$. Consensus-averaging is robust to sensor failures so long as the network remains connected. A remark is now due on the practical operation of consensus averaging.

Remark 1. Since only a finite number of consensus-averaging iterations can be afforded in practice, some residual error will be inevitable. To mitigate these residual weight mismatches, one can apply min- (max-) consensus iterations to form the $\min_n \psi_n(i)$ (correspondingly $\max_n \psi_n(i)$), after the consensus-averaging iterations are completed. In the simulations, both min- and max- consensus will be run on the $\psi_n(i)$ s and the final weight value will be set equal to their average.

Clearly from (9), either consensus or incremental averaging should be run per particle $\mathbf{x}_k^{(m)}$. Hence, the number of particles increases the communication overhead of S3 considerably, because M such consensus or incremental operations must be run in parallel. It will become evident that this requirement also limits the options for distributing S2, when adapted IS densities are employed. As mentioned earlier, performing S2

in a distributed fashion is possible when using a non-adapted IS density, such as the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ in B-PF, which does not require knowledge of \mathbf{y}_k . Such a selection of π however, requires a large number of particles to cope with particle depletion [6], which in turn increases the communication cost of S3 and renders the distributed algorithm inefficient. Indeed, recall from (4) that for M sufficiently large the error variance per PF step is $E[(\hat{I}_\phi(k) - E[\phi(\mathbf{x}_k)|\mathbf{y}_{1:k}])^2] \approx C/M$. Constant C , and the threshold value for M above which this per-step PF variance decreases as $\mathcal{O}(M^{-1})$, depend on the similarity between the true posterior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$, and the selected IS density $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$ [10], [14]. To ensure that p and π are “similar,” information in the observations \mathbf{y}_k must be exploited by the selected π .

This certainly advocates data adaptation when selecting the IS density. In non-adapted PF renditions, such as the B-PF, many particles $\mathbf{x}_k^{(m)}$ receive almost zero weights in S3, and have to be discarded since they are generated from a density far different from the true posterior. The latter leads to particle depletion, which compromises performance. In contrast, data-adapted IS densities ensure that the new measurements \mathbf{y}_k are accounted for during the particle generation phase, and thus fewer particles are needed to approximate the posterior accurately. Therefore, adaptation brings significant benefits because it reduces the number of required particles, which translates to a commensurate reduction in communication overhead of the proposed distributed PF. From the large list of sub-optimal adaptation methods for centralized PF mentioned in the Introduction, none of them is amenable to efficient distributed implementation.

To bypass this challenge, the next section introduces a novel adaptation method, which renders the distributed implementation of S2 affordable by reducing the overhead of inter-sensor communications.

IV. SET MEMBERSHIP-BASED ADAPTATION

The key idea behind the proposed low-overhead distributed adaptation scheme is to approximate the posterior density $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k)$ with an appropriately scaled (distorted) version of the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. To this end, suppose temporarily that a *local* set \mathcal{E}_k^n can be constructed per sensor n from the domain of $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$ to capture most of its probability mass. And based on these local sets, consider the *global* set $\mathcal{E}_k := \bigcap_n \mathcal{E}_k^n$, over which *all* sensors have high local posterior masses. With $\mathbb{1}_{\{\cdot\}}$ representing the indicator function, the proposed adapted IS density is

$$\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}) := \frac{\alpha_k \mathbb{1}_{\{\mathbf{x}_k \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\mathbf{x}_k \notin \mathcal{E}_k\}}}{c_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}) \quad (10)$$

where $\beta_k \ll \alpha_k$ (w.l.o.g. $\alpha_k = 1$), and c_k is a normalization constant so that $\pi_{\mathcal{E}_k}(\cdot)$ integrates to 1.

Before describing the process for selecting $\{\mathcal{E}_k^n\}_{n=1}^N$, which define the global set \mathcal{E}_k , the ensuing subsection explains why $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1})$ offers an attractive approximation of the global posterior.

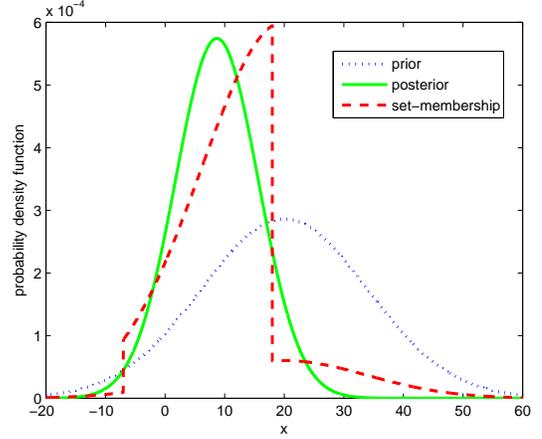


Fig. 1. Similarity between scaled prior (set-membership approximation) and posterior densities

A. Posterior Density Approximation

Consider that each sensor n has determined the local set \mathcal{E}_k^n containing most of the probability mass of its local posterior $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$, and approximate the latter using $\pi_{\mathcal{E}_k^n}(\mathbf{x}_k|\mathbf{x}_{k-1})$. Since the noise across sensors is independent, the global posterior can be expressed as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k) &\propto \left[\prod_{n=1}^N \frac{p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)}{p(\mathbf{x}_k|\mathbf{x}_{k-1})} \right] p(\mathbf{x}_k|\mathbf{x}_{k-1}) \\ &\approx \left[\prod_{n=1}^N \frac{\pi_{\mathcal{E}_k^n}(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{x}_k|\mathbf{x}_{k-1})} \right] p(\mathbf{x}_k|\mathbf{x}_{k-1}) \\ &\approx \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}) \end{aligned} \quad (11)$$

where the product $\prod_{n=1}^N \frac{\pi_{\mathcal{E}_k^n}(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{x}_k|\mathbf{x}_{k-1})}$ is approximated using the function $(\alpha_k \mathbb{1}_{\{\mathbf{x}_k \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\mathbf{x}_k \notin \mathcal{E}_k\}}) / c_k$, which is valid for $\beta_k \ll 1$ since $\mathcal{E}_k := \bigcap_n \mathcal{E}_k^n$. Compared to the prior itself, the scaled prior in (10), later referred to as the set-membership approximation, can be much closer to the posterior, which is the density of interest; see also Fig. 1.

B. Local Set Selection

In view of (11), the local set \mathcal{E}_k^n at sensor n must be constructed so that $\pi_{\mathcal{E}_k^n}(\mathbf{x}_k|\mathbf{x}_{k-1}) \approx p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$. Note though that it is impossible to specify directly the set \mathcal{E}_k^n that contains a high probability mass of $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$ because the latter is conditioned on \mathbf{x}_{k-1} , which is not available. One way around this obstacle is to construct a separate local \mathcal{E}_k^n for every $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k^n)$ corresponding to each particle $m = 1, \dots, M$. However, the computational burden of this construction defeats the purpose of introducing the local sets at the first place. (Indeed, instead of computing a single \mathcal{E}_k^n per sensor which requires consensus on a single \mathcal{E}_k , multiple local sets necessitate consenting on one \mathcal{E}_k per particle m , which increases the communication overhead prohibitively.)

The remedy at an intuitive level is to construct \mathcal{E}_k^n so that it contains a high probability mass of $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$

on the *average* performed over \mathbf{x}_{k-1} . This motivates selecting \mathcal{E}_k^n to contain a large probability mass of the density $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$. Such a choice is prudent because

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)d\mathbf{x}_{k-1}$$

where \mathbf{x}_{k-1} is averaged over the posterior $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$ that is estimated using particles and their weights at step $k-1$. Through this density, averaging out \mathbf{x}_{k-1} takes into account not only the locally available current measurement \mathbf{y}_k^n , but also all past measurements $\mathbf{y}_{1:k-1}$ through $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$.

In the remainder of this subsection, three schemes will be developed to select local sets \mathcal{E}_k^n containing most of the probability mass under $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$. Their efficacy will be tested via simulations.

1) *Particle Bounding Box*: The crux of this construction is reliance on local B-PF to form the corresponding local set \mathcal{E}_k^n . Toward this objective, consider (re-) sampling per sensor as in S1-S2 from the prior $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n) := p(\mathbf{x}_k|\mathbf{x}_{k-1})$, which is the IS density of B-PF. Since this local prior is non-adapted, it is expected that many less than M particles re-sampled as in S1-S2 will fall in the region where the likelihood (and hence the local posterior) is high. Thus, each sensor must over-sample particles as in S1 by a factor $L > 1$ ($L \in \mathbb{N}$), so that a few of the LM particles will fall in the region that the set \mathcal{E}_k^n is sought to capture. Recall also that with the same seed across sensors, these LM particles at all sensors will be identical. Once the local likelihood $p(\mathbf{y}_k^n|\mathbf{x}_k^{(m)})$ is weighted in, the sensors generate different weights, which along with the LM particles yield an estimate of $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$. Re-sampling M particles from the latter, and fitting an axes-aligned box around them yields the desired \mathcal{E}_k^n , which represents the region where $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$ contains most of its mass.

Letting $\{\bar{\mathbf{x}}_k^{n,(m)}\}_{m=1}^M$ denote these re-sampled particles, the axes-aligned bounding box describing the local set at sensor n at time k is specified by the following polyhedron:

$$\begin{aligned} \mathcal{E}_k^n &:= \left\{ \mathbf{x} : \min_m \bar{\mathbf{x}}_k^{n,(m)} \preceq \mathbf{x} \preceq \max_m \bar{\mathbf{x}}_k^{n,(m)} \right\} \\ &:= \left\{ \mathbf{x} : \mathbf{x}_{k,\min}^n \preceq \mathbf{x} \preceq \mathbf{x}_{k,\max}^n \right\} \end{aligned} \quad (12)$$

where the minimum, maximum, and inequalities (\preceq) should be understood component-wise. Fig. 2 depicts the min and max operations for $d_x = 2$.

In summary, to arrive at the set in (12) each sensor runs the following three local steps.

LS1) Over re-sample particles and weights $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$ to obtain $\{\tilde{\mathbf{x}}_{k-1}^{(m')}\}_{m'=1}^{LM}$ with $L \in \mathbb{N}$.

LS2) Generate LM new particles from the prior $\tilde{\mathbf{x}}_k^{(m')} \sim p(\mathbf{x}_k|\tilde{\mathbf{x}}_{k-1}^{(m')})$; form their weights using the local likelihood $\tilde{w}_k^{n,(m')} = p(\mathbf{y}_k^n|\tilde{\mathbf{x}}_k^{(m')})$; and normalize them to obtain $\bar{w}_k^{n,(m')}$.

LS3) Re-sample M particles from $\{\tilde{\mathbf{x}}_k^{(m')}, \bar{w}_k^{n,(m')}\}_{m'=1}^{LM}$, and construct \mathcal{E}_k^n as in (12).

Note in closing that the weights in LS1 and LS2 are not used subsequently, since the aim here is not PF but construction of the local sets, which are specified only by the (min and max per entry) particles re-sampled at LS3. Fig. 3 depicts

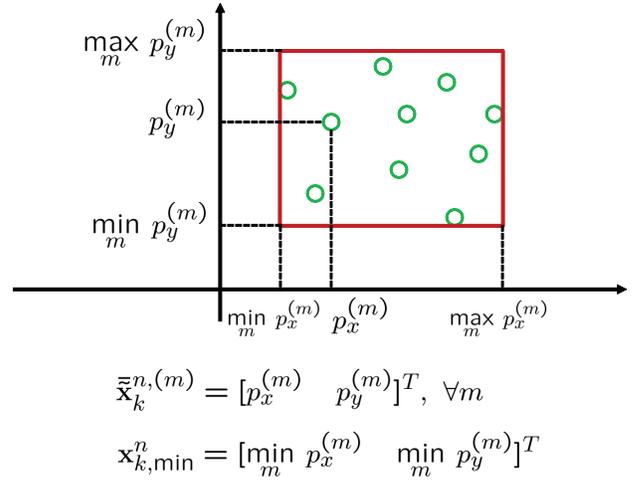


Fig. 2. Particle bounding box approach

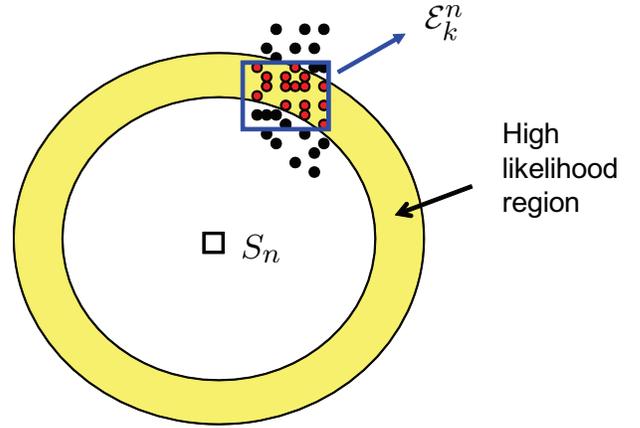


Fig. 3. Graphical depiction of selecting the local set \mathcal{E}_k^n

the particle bounding box approach for a sensor measuring its distance from the target.

2) *UKF-based Local Set*: This construction capitalizes on the unscented particle filter (UPF) [35], which enables fitting to the local posterior, namely $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$, a Gaussian density specified by its mean and covariance. Per time step k , the latter can be found through a UKF update. Given the obtained mean and covariance, each local set is constructed to encapsulate a prescribed amount of probability mass from the corresponding local posterior of interest. For the Gaussian approximating density, this set takes the form of an ellipsoid. The local set \mathcal{E}_k^n is then specified as the smallest axes-aligned box containing this ellipsoid. Specifically, the following local steps are performed at each sensor n .

LS1') Given the particles and weights $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$ from step $(k-1)$, find the mean and covariance matrix of the Gaussian density approximating the posterior at time step $(k-1)$, that is

$$\begin{aligned} \hat{\mathbf{x}}_{k-1} &= \sum_{m=1}^M \bar{w}_{k-1}^{(m)} \mathbf{x}_{k-1}^{(m)} \\ \hat{\mathbf{C}}_{k-1} &= \sum_{m=1}^M \bar{w}_{k-1}^{(m)} (\mathbf{x}_{k-1}^{(m)} - \hat{\mathbf{x}}_{k-1})(\mathbf{x}_{k-1}^{(m)} - \hat{\mathbf{x}}_{k-1})^T. \end{aligned}$$

LS2') With input the estimates $\hat{\mathbf{x}}_{k-1}$ and $\hat{\mathbf{C}}_{k-1}$ found in LS1', run a UKF prediction-correction iteration as in [31]. Incorporating the local measurement \mathbf{y}_k^n in the correction step, the UKF output yields the conditional mean $\hat{\mathbf{x}}_k^n$, and covariance matrix $\hat{\mathbf{C}}_k^n$ per sensor n at time step k .

LS3') Based on $\hat{\mathbf{x}}_k^n$ and $\hat{\mathbf{C}}_k^n$, form \mathcal{E}_k^n as the smallest axes-aligned box containing the ellipsoid

$$(\mathbf{x}_k - \hat{\mathbf{x}}_k^n)^T \left(\hat{\mathbf{C}}_k^n \right)^{-1} (\mathbf{x}_k - \hat{\mathbf{x}}_k^n) \leq \eta^2 \quad (13)$$

where η is set depending on the probability mass the designer chooses to capture by the ellipsoid.

As an example, suppose one requires 99% of the probability mass to be contained in the ellipsoid. The linear transformation $\mathbf{x}_{\text{new}} = \left(\hat{\mathbf{C}}_k^n \right)^{-1/2} (\mathbf{x}_{\text{old}} - \hat{\mathbf{x}}_k^n)$ expresses the quadratic form in (13) as a sum of d_x independent, zero-mean, and unit-variance Gaussian-squared random variables, which amounts to a random variable distributed according to a centralized $\chi_{d_x}^2$ density with d_x degrees of freedom. Parameter η^2 is the point for which the integral of this density over $[0, \eta^2]$ equals 0.99.

To visualize the smallest axes-aligned box containing the ellipsoid, consider as an example the two-dimensional case ($d_x = 2$), illustrated graphically in Fig. 4. The smallest axes-aligned bounding box is completely determined by the four boundary points x_{\min} , x_{\max} , y_{\min} , and y_{\max} . These points are obtained by projecting the ellipsoid on the x - and y -axes. To generalize for any d_x and specify analytically the smallest axes-aligned box in LS3', let \mathbf{e}_d denote the elementary $d_x \times 1$ vector having its d th entry equal to 1, and all other entries equal to 0. Clearly, the projection of a point \mathbf{x} on the ellipsoid over the d th side of the polyhedral box is given by the inner product $\mathbf{e}_d^T \mathbf{x}$. Hence, the minimum $[\mathbf{x}_k^n(d)]_{\min}$ along the d th coordinate axis can be obtained as the solution of the following constrained optimization problem:

$$\begin{cases} \min_{\mathbf{x}} & \mathbf{e}_d^T \mathbf{x} \\ \text{subject to} & (\mathbf{x} - \hat{\mathbf{x}}_k^n)^T \left(\hat{\mathbf{C}}_k^n \right)^{-1} (\mathbf{x} - \hat{\mathbf{x}}_k^n) \leq \eta^2 \end{cases}$$

This problem is convex, and admits the following closed-form solution

$$[\mathbf{x}_k^n(d)]_{\min} = \hat{\mathbf{x}}_k^n(d) - \eta \sqrt{\hat{\mathbf{C}}_k^n(d, d)}.$$

Clearly, solving the related optimization problem with \mathbf{e}_d replaced by $-\mathbf{e}_d$ will yield the maximum $[\mathbf{x}_k^n(d)]_{\max}$ along the d th coordinate axis. Finding likewise the minima and maxima for all $d = 1, \dots, d_x$, specifies the desired box.

Remark 2. While an EKF update can be employed instead of the UKF in LS2', the superior performance of UKF confirmed by simulations suggests that not much is gained when adopting the EKF. Notwithstanding, UKF is not used here as a tracker, but only to specify the local sets $\{\mathcal{E}_k^n\}_{n=1}^N$.

3) *UKF with Gaussian Mixture Model:* When the local posterior $p(\mathbf{x}_k | \mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$ is multi-modal, UKF can not approximate it well with a single Gaussian bell, which is uni-modal. To cope with multi-modality, a GMM can be trained from the particles representing the prior, using the expectation-maximization (EM) algorithm; see e.g., [3, Chapter 9]. Then, the UKF-based algorithm of the uni-modal case can be applied

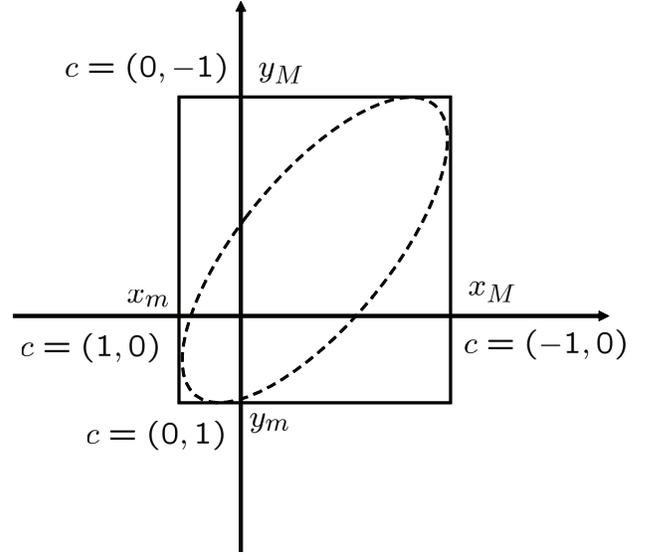


Fig. 4. The smallest axis-aligned bounding box for a 2-D ellipsoid

to each mixture component separately. This amounts to one axes-aligned box per mixture component. The local set \mathcal{E}_k^n is then constructed as the smallest box containing the union of these boxes. While the number of components (modes) can be preselected based on complexity considerations, it is also possible to fit a GMM with unknown model order. In this case, one of the criteria outlined in e.g., [44, Appendix C] can be utilized for model order determination.

C. Global Set Determination

Once sensor n processes its own \mathbf{y}_k^n to obtain the local set $\mathcal{E}_k^n = \left\{ \mathbf{x} : \mathbf{x}_{k,\min}^n \preceq \mathbf{x} \preceq \mathbf{x}_{k,\max}^n \right\}$, the global set \mathcal{E}_k can be constructed as

$$\mathcal{E}_k := \bigcap_{n=1}^N \mathcal{E}_k^n = \left\{ \mathbf{x} : \max_n \mathbf{x}_{k,\min}^n \preceq \mathbf{x} \preceq \min_n \mathbf{x}_{k,\max}^n \right\}. \quad (14)$$

Since boxes are used to represent local sets, distributing their intersection is very simple because it only requires finding minima and maxima of scalar quantities available per sensor; see also Fig. 5. Distinct from the distributed B-PF algorithm of Section III that requires consensus averaging iterations, constructing the global set \mathcal{E}_k in a distributed fashion entails consenting on the minimum and maximum points of the local set boundaries.

It is possible to find, say these maximum points, using a max-consensus approach along the lines of the consensus averaging scheme outlined in Section III. Except that instead of $\bar{\psi}_N$, sensors now wish to compute $\psi_{\max} := \max_n \psi_n$ in a distributed fashion. Clearly, one scalar $\psi_n = \mathbf{x}_{k,\min}^n(d)$ is involved here per coordinate d . In iteration i of the max-consensus algorithm, sensor n communicates $\psi_n(i-1)$ to its immediate neighborhood $\mathcal{N}(n)$, and updates the local auxiliary variables $\psi_n(i)$ as

$$\psi_n(i) = \max\{\psi_n(i-1), \max_{l \in \mathcal{N}(n)} \psi_l(i-1)\}, \quad \psi_n(0) = \psi_n.$$

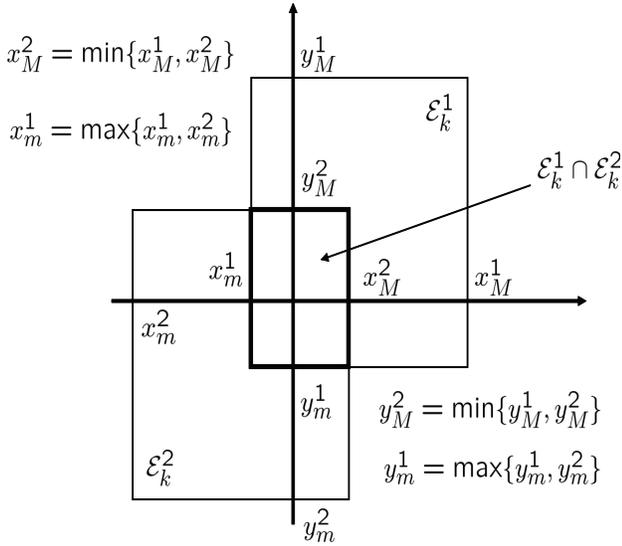


Fig. 5. Intersection of two boxes amounts to computing minima and maxima

Most importantly when compared to consensus averaging, the iterates $\psi_n(i)$ converge to the exact ψ_{\max} in finite time, and the number of required iterations does not exceed the diameter of the communication graph; see e.g., [8]. The min-consensus scheme operates similarly with the obvious substitutions.

Because min- and max-consensus algorithms converge in finite iterations bounded by the diameter of the communication graph, a distributed implementation becomes available at much lower communication cost compared to propagating (or performing consensus of) raw measurements across sensors.

Remark 3. When the intersection \mathcal{E}_k in (14) is empty, local sets are stretched out by a constant factor before re-computing the intersection. This is repeated until the process arrives at a non-empty intersection. Simulations indicate that two repetitions are typically sufficient, while the first intersection is non-empty most of the time.

D. Distributed SMC-PF

Once \mathcal{E}_k is available to all sensors, samples are drawn per sensor (cf. S2) from the scaled prior density $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$. To generate these samples, rejection sampling (RS) is employed, see e.g., [3, Chapter 11], with IS density $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ and target density $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, upper-bounded as: $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) \leq (\alpha_k/c_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$. Specifically, samples are drawn first from the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$. Then, they are accepted with probability $c_k\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})/[\alpha_k p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})] = \mathbb{1}_{\{\mathbf{x}_k \in \mathcal{E}_k\}} + (\beta_k/\alpha_k)\mathbb{1}_{\{\mathbf{x}_k \notin \mathcal{E}_k\}}$. Therefore, if a sample belongs to \mathcal{E}_k , it is accepted with probability 1; otherwise, it is rejected with high probability. In other words, the accept probability satisfies $\beta_k/\alpha_k \ll 1$.

Now all pieces of the novel distributed SMC-PF can be put together. Sensors start with a common seed, and (re-)sample identical particles as per S1 locally. For S2, RS-based sampling from $\pi_{\mathcal{E}_k}(\cdot)$ is run also locally after \mathcal{E}_k becomes available to every sensor via consensus on the set boundaries. Incremental or consensus averaging distributes the weights in

S3, which is all every sensor needs to perform S4. Specifically for consensus averaging, one uses $\pi(\cdot) = \pi_{\mathcal{E}_k}(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})$ in (9) to obtain the weights as

$$\log(w_k^{(m)}) = \sum_{n=1}^N \log(p(\mathbf{y}_k^n|\mathbf{x}_k^{(m)})) + \log(c_k^{(m)}) - \log(\alpha_k \mathbb{1}_{\{\mathbf{x}_k^{(m)} \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\mathbf{x}_k^{(m)} \notin \mathcal{E}_k\}}). \quad (15)$$

The SMC-PF algorithm is summarized in Table I.

Remark 4. The novel approach is referred to as set membership constrained (SMC) PF for two reasons: i) successive set intersections are computed, which is a trade-mark of set-membership approaches [41]; and, ii) rejection sampling relies on an accept-reject criterion assessing membership in the set \mathcal{E}_k .

Remark 5. (Communication Cost) Let κ denote the number of consensus averaging iterations in either B-PF or SMC-PF, and D_{graph} the diameter of the communication graph. Then, B-PF requires $\kappa M + 2D_{\text{graph}}M$ scalars to be communicated per time step and sensor, while SMC-PF requires $2d_x D_{\text{graph}} + \kappa M + 2D_{\text{graph}}M$ scalars; hence, for the same number of particles M , SMC-PF needs communication of $2d_x D_{\text{graph}}$ more scalars than B-PF. Note that κM denotes the number of scalars that are transmitted per time step and sensor for consensus-averaging iterations. Afterwards, min- and max-consensuses should be run, as a consequence of Remark 1, which requires communication of $2D_{\text{graph}}M$ scalars. Finally, SMC-PF needs to run min- and max- consensuses to find the global set which requires transmission of $2d_x D_{\text{graph}}$ scalars.

V. PERFORMANCE ANALYSIS

This section analyzes the performance of SMC-PF using the state estimator's mean-square error (MSE) as figure of merit. First, the MSE is expressed as the superposition of its minimum MSE (MMSE) value plus the PF-related error variance denoted by Var_{PF} . Subsequently, a per-step tractable approximation of the Var_{PF} is introduced to allow for comparing SMC-PF with the B-PF. This finite-sample approximate analysis is also complemented with the asymptotic MSE performance of SMC-PF as the number of particles grows large to establish that the MMSE is indeed attained asymptotically; see also [9] and [26].

Consider first the MSE at iteration k , namely

$$\begin{aligned} \text{MSE}(k) &:= E \left[\left(\hat{I}_\phi(k) - \phi(\mathbf{x}_k) \right)^2 \right] \\ &= E \left[\left(\hat{I}_\phi(k) - I_\phi(k) + I_\phi(k) - \phi(\mathbf{x}_k) \right)^2 \right] \\ &= E \left[\left(\hat{I}_\phi(k) - I_\phi(k) \right)^2 \right] + E \left[\left(I_\phi(k) - \phi(\mathbf{x}_k) \right)^2 \right] \\ &\quad + 2E \left[\left(\hat{I}_\phi(k) - I_\phi(k) \right) \left(I_\phi(k) - \phi(\mathbf{x}_k) \right) \right] \\ &= \text{Var}_{PF}(k) + \text{MMSE}(k) + \text{Cross Term}. \quad (16) \end{aligned}$$

The cross term vanishes because

$$\begin{aligned} \text{Cross Term} &= 2E_{\mathbf{y}_{1:k}} \left[E \left[\left(\hat{I}_\phi(k) - I_\phi(k) \right) \left(I_\phi(k) - \phi(\mathbf{x}_k) \right) \middle| \mathbf{y}_{1:k} \right] \right] \end{aligned}$$

Table I. SMC-PF Algorithm

Initialization. Draw $\mathbf{x}_0^{(m)} \sim p(\mathbf{x}_0)$, $\forall m = 1, \dots, M$. Set $\bar{w}_0^{(m)} = \frac{1}{M}$. Repeat for time $k \geq 1$ Repeat for sensors $n = 1, \dots, N$ (can be implemented in parallel) Given $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$ and \mathbf{y}_k^n , run either LS1-LS3 or LS1'-LS3' to obtain \mathcal{E}_k^n . End for Given $\{\mathcal{E}_k^n\}_{n=1}^N$, construct \mathcal{E}_k as in (14) in a distributed fashion. At each sensor run in parallel Re-sample from $\{\mathbf{x}_{k-1}^{(m)}, \bar{w}_{k-1}^{(m)}\}_{m=1}^M$, [S1]. Draw the new sample $\mathbf{x}_k^{(m)} \sim \pi(\mathbf{x}_k \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) := \pi_{\mathcal{E}_k}(\mathbf{x}_k \mathbf{x}_{k-1}^{(m)})$ [S2]. Update the weights as in (15) [S3]. Use incremental or consensus averaging. Obtain $\{\mathbf{x}_k^{(m)}, \bar{w}_k^{(m)}\}_{m=1}^M$ at the output of S2-S3. Form $\hat{I}_\phi(k) := \sum_{m=1}^M \bar{w}_k^{(m)} \phi(\mathbf{x}_k^{(m)})$ as the final estimate [S4]. End for

$$\begin{aligned}
 &= 2E_{\mathbf{y}_{1:k}} \left[E \left[\hat{I}_\phi(k) - I_\phi(k) | \mathbf{y}_{1:k} \right] E \left[I_\phi(k) - \phi(\mathbf{x}_k) | \mathbf{y}_{1:k} \right] \right] \\
 &= 2E_{\mathbf{y}_{1:k}} \left[E \left[\hat{I}_\phi(k) - I_\phi(k) | \mathbf{y}_{1:k} \right] \times 0 \right] = 0 \quad (17)
 \end{aligned}$$

where the second equality holds because $I_\phi(k)$ is constant when conditioned on $\mathbf{y}_{1:k}$ and the operations required to obtain the PF estimate $\hat{I}_\phi(k)$, namely re-sampling in S1 and drawing new particles in S2, do not depend on the true state \mathbf{x}_k when conditioned on $\mathbf{y}_{1:k}$.

Equations (16) and (17) imply that $\text{MSE}(k) = \text{Var}_{PF}(k) + \text{MMSE}(k)$. The $\text{MMSE}(k)$ that lower bounds the $\text{MSE}(k)$ depends on the way the latent state variables and measurements are related - a dependency specified by the nonlinear functions \mathbf{f} and \mathbf{h} of the model. Therefore, it is prudent to minimize $\text{Var}_{PF}(k)$ per step k since this amounts to minimizing the $\text{MSE}(k)$.

In the next subsection, $\text{Var}_{PF}(k)$ will be approximated using the IS variance in (4) in order to obtain a performance metric, which will be useful to assess the finite-sample efficiency of the SMC-PF tracker.

A. Finite-sample analysis

Consider specializing (4) for the following target and IS densities:

$$\begin{aligned}
 q_t(\mathbf{x}_k) &:= p(\mathbf{x}_{0:k} | \mathbf{y}_{1:k}) \\
 &= \frac{p(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{0:k-1} | \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})} \quad (18a)
 \end{aligned}$$

$$\begin{aligned}
 q_{IS}(\mathbf{x}_k) &:= \pi(\mathbf{x}_{0:k} | \mathbf{y}_{1:k}) \\
 &= \pi(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_k) p(\mathbf{x}_{0:k-1} | \mathbf{y}_{1:k-1}) \quad (18b)
 \end{aligned}$$

and express the per-step variance of PF as

$$\text{Var}_{PF}(k) = E_{\mathbf{y}_{1:k}} \left[E \left[\left(\hat{I}_\phi(k) - I_\phi(k) \right)^2 | \mathbf{y}_{1:k} \right] \right] \quad (19)$$

where the inner expectation in (19) corresponds to Var_{IS} in (4) with the specific choices in (18). Substituting (4) into (19) one arrives at

$$\begin{aligned}
 \text{Var}_{PF}(k) &\approx (1/M) \times \\
 &E_{\mathbf{y}_{1:k}} \left[\text{Var}_p(\phi(\mathbf{x}_k)) E_\pi \left[\left(\frac{p(\mathbf{x}_{0:k} | \mathbf{y}_{1:k})}{\pi(\mathbf{x}_{0:k} | \mathbf{y}_{1:k})} \right)^2 | \mathbf{y}_{1:k} \right] \right]. \quad (20)
 \end{aligned}$$

Recall that the choice of π is what differentiates B-PF from SMC-PF. Using (18), the π -dependent inner expectation in (20) can be written as

$$\begin{aligned}
 &E_\pi \left[\left(\frac{p(\mathbf{x}_{0:k} | \mathbf{y}_{1:k})}{\pi(\mathbf{x}_{0:k} | \mathbf{y}_{1:k})} \right)^2 | \mathbf{y}_{1:k} \right] \\
 &= \int \frac{p^2(\mathbf{y}_k | \mathbf{x}_k) p^2(\mathbf{x}_k | \mathbf{x}_{k-1})}{p^2(\mathbf{y}_k | \mathbf{y}_{1:k-1}) \pi(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{y}_k)} p(\mathbf{x}_{0:k-1} | \mathbf{y}_{1:k-1}) d\mathbf{x}_{0:k} \\
 &\approx \frac{1}{p^2(\mathbf{y}_k | \mathbf{y}_{1:k-1})} \sum_{m=1}^M \bar{w}_{k-1}^{(m)} \int \frac{p^2(\mathbf{y}_k | \mathbf{x}_k) p^2(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k \quad (21)
 \end{aligned}$$

where for the approximation $p(\mathbf{x}_{0:k-1} | \mathbf{y}_{1:k-1})$ was replaced by its estimate in (6).

For comparison with B-PF, set $\pi(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) := p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)})$ to specialize (21) to

$$\begin{aligned}
 &\int \frac{p^2(\mathbf{y}_k | \mathbf{x}_k) p^2(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k \\
 &= \left(\int p^2(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \right) \times 1 \\
 &= \left(\int_{\mathcal{E}_k} p^2(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \right. \\
 &\quad \left. + \int_{\bar{\mathcal{E}}_k} p^2(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \right) \\
 &\quad \times \left(\int_{\mathcal{E}_k} p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k + \int_{\bar{\mathcal{E}}_k} p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \right) \\
 &= A^{(m)}(\mathcal{E}_k) B^{(m)}(\mathcal{E}_k) + A^{(m)}(\mathcal{E}_k) B^{(m)}(\bar{\mathcal{E}}_k) \\
 &\quad + A^{(m)}(\bar{\mathcal{E}}_k) B^{(m)}(\mathcal{E}_k) + A^{(m)}(\bar{\mathcal{E}}_k) B^{(m)}(\bar{\mathcal{E}}_k) \quad (22)
 \end{aligned}$$

where $\bar{\mathcal{E}}_k$ denotes the complement set of \mathcal{E}_k , and $A^{(m)}$, $B^{(m)}$ are defined as

$$\begin{aligned}
 A^{(m)}(\mathcal{E}_k) &:= \int_{\mathcal{E}_k} p^2(\mathbf{y}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \\
 B^{(m)}(\mathcal{E}_k) &:= \int_{\mathcal{E}_k} p(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k.
 \end{aligned}$$

For the SMC-PF, set $\pi(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) := \pi_{\mathcal{E}_k}(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)})$, to obtain

$$\int \frac{p^2(\mathbf{y}_k | \mathbf{x}_k) p^2(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k | \mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k$$

$$\begin{aligned}
&= c_k^{(m)} \int \frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\alpha_k \mathbb{1}_{\{\mathbf{x}_k \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\mathbf{x}_k \in \bar{\mathcal{E}}_k\}}} d\mathbf{x}_k \\
&= \frac{c_k^{(m)}}{\alpha_k} A^{(m)}(\mathcal{E}_k) + \frac{c_k^{(m)}}{\beta_k} A^{(m)}(\bar{\mathcal{E}}_k) \quad (23)
\end{aligned}$$

where the normalization constant $c_k^{(m)}$ is given by

$$\begin{aligned}
c_k^{(m)} &= \int (\alpha_k \mathbb{1}_{\{\mathbf{x}_k \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\mathbf{x}_k \in \bar{\mathcal{E}}_k\}}) p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \\
&= \alpha_k B^{(m)}(\mathcal{E}_k) + \beta_k B^{(m)}(\bar{\mathcal{E}}_k).
\end{aligned}$$

Substituting $c_k^{(m)}$ back into (23) one arrives at

$$\begin{aligned}
&\int \frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k \\
&= A^{(m)}(\mathcal{E}_k)B^{(m)}(\mathcal{E}_k) + \frac{\beta_k}{\alpha_k} A^{(m)}(\mathcal{E}_k)B^{(m)}(\bar{\mathcal{E}}_k) \\
&\quad + \frac{\alpha_k}{\beta_k} A^{(m)}(\bar{\mathcal{E}}_k)B^{(m)}(\mathcal{E}_k) + A^{(m)}(\bar{\mathcal{E}}_k)B^{(m)}(\bar{\mathcal{E}}_k). \quad (24)
\end{aligned}$$

Comparison of (22) with (24) reveals that SMC-PF is a generalization of B-PF, which allows for more flexibility by adjusting the ratio α_k/β_k to minimize $\text{Var}_{PF}(k)$. For a given set \mathcal{E}_k , it is possible to plug (24) back into (21), differentiate with respect to this ratio, and solve to find the optimal

$$\left(\frac{\alpha_k}{\beta_k} \right)^* = \sqrt{\frac{\sum_{m=1}^M \bar{w}_{k-1}^{(m)} A^{(m)}(\mathcal{E}_k) B^{(m)}(\bar{\mathcal{E}}_k)}{\sum_{m=1}^M \bar{w}_{k-1}^{(m)} A^{(m)}(\bar{\mathcal{E}}_k) B^{(m)}(\mathcal{E}_k)}}.$$

Unfortunately, this optimal ratio is given in terms of intractable integrals emerging in the definitions of $A^{(m)}(\mathcal{E}_k)$ and $B^{(m)}(\mathcal{E}_k)$. While numerical methods or additional IS estimates can in principle be considered per step to approximate these integrals, this is hardly justified from a complexity perspective. In practice, it is thus reasonable to adjust the ratio α_k/β_k heuristically. Simulations will confirm that the $\text{MSE}(k)$ is robust to changes of α_k/β_k , and a heuristic selection yields satisfactory performance.

It is of interest now to investigate the conditions under which the SMC-PF outperforms the B-PF by the largest margin. To this end, consider the following operating assumptions. **A1.** Let $\mathcal{E}_k \subseteq \mathbb{R}^{d_x}$ be a set in the state-space of \mathbf{x}_k such that

$$p^2(\mathbf{y}_k|\mathbf{x}_k) \leq \epsilon_1 \ll 1, \quad \forall \mathbf{x}_k \in \bar{\mathcal{E}}_k.$$

A2. For the set \mathcal{E}_k in A1, it holds that

$$\int_{\mathcal{E}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \leq \epsilon_2 \ll 1, \quad \forall m = 1, \dots, M.$$

A3. The likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$ does not have most of its mass in the tail of the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$; thus,

$$\int_{\mathcal{E}_k} p^2(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k := C^{(m)} \gg \epsilon := \max\{\epsilon_1, \epsilon_2\}$$

for all $m = 1, \dots, M$.

A4. Given $\epsilon_0 \ll \epsilon$, the number of particles M is selected large enough so that the approximation errors in (20) and (21) are bounded by ϵ_0 .

For notational brevity, the dependence of ϵ_1, ϵ_2 and $C^{(m)}$ on k is suppressed. Condition A1 ensures that the likelihood takes high values for \mathbf{x}_k lying inside \mathcal{E}_k ; while A2 guarantees that the same set \mathcal{E}_k contains only a small probability mass of the prior. Together, A1 and A2 enforce the so-called *peaky likelihood* condition. A3 asserts that the likelihood does not fall in the tail of the prior density by requiring the relevant integral of their product to be large. Combination of A1-A3 suggests that the likelihood is peaky, but does not fall in the tail of the prior. A4 is needed to ensure that (21) offers an accurate approximation of the $\text{MSE}(k)$. Under conditions A1-A4, the SMC-PF provides considerable MSE improvement per step k when compared to the B-PF, as summarized in the following proposition.

Proposition 1. *Under A1-A4, for a fixed number of particles M , and with $\beta_k/\alpha_k := \epsilon$, the SMC-PF error variance in (20) lowers that of B-PF by a factor $\mathcal{O}(\epsilon)$.*

Proof: Condition A2 implies that $B^{(m)}(\mathcal{E}_k) \leq \epsilon_2$, and hence $1 - \epsilon_2 \leq B^{(m)}(\bar{\mathcal{E}}_k) \leq 1$; while A3 implies that $A^{(m)}(\mathcal{E}_k) = C^{(m)}$. On the other hand, A1 yields

$$\begin{aligned}
A^{(m)}(\bar{\mathcal{E}}_k) &= \int_{\bar{\mathcal{E}}_k} p^2(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \\
&\leq \int_{\bar{\mathcal{E}}_k} \epsilon_1 p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \leq \epsilon_1.
\end{aligned}$$

For the B-PF it thus follows that (cf. (22))

$$\begin{aligned}
\int \frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k &\geq A^{(m)}(\mathcal{E}_k)B^{(m)}(\bar{\mathcal{E}}_k) \\
&\geq C^{(m)}(1 - \epsilon_2) \quad (25)
\end{aligned}$$

while for the SMC-PF it holds that (cf. (24))

$$\begin{aligned}
\int \frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k &\leq C^{(m)}\epsilon_2 + \frac{\beta_k}{\alpha_k} C^{(m)} \\
&\quad + \frac{\alpha_k}{\beta_k} \epsilon_1 \epsilon_2 + \epsilon_1.
\end{aligned}$$

Upon selecting $\beta_k/\alpha_k = \epsilon$, the π -dependent factor in the PF error variance can be bounded as

$$\begin{aligned}
\int \frac{p^2(\mathbf{y}_k|\mathbf{x}_k)p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)} d\mathbf{x}_k &\leq C^{(m)}\epsilon_2 + \epsilon C^{(m)} + \frac{\epsilon_1 \epsilon_2}{\epsilon} + \epsilon_1 \\
&\leq 2(C^{(m)} + 1)\epsilon. \quad (26)
\end{aligned}$$

Consider two cases. In the first case $C^{(m)} \geq 1$, which implies that the upper bound in (26) becomes $4C^{(m)}\epsilon$. Comparing this against (25), shows a factor $4\epsilon/(1 - \epsilon_2)$ reduction in the evaluated integral for SMC-PF compared to B-PF. In the second case $C^{(m)} \leq 1$, the upper bound in (26) becomes 4ϵ . Comparing this against (25), shows a factor $4\epsilon/(C^{(m)}(1 - \epsilon_2))$ reduction in the evaluated integral for SMC-PF compared to B-PF. With $C^{(m)} \leq 1$, there is at least a reduction of $4\epsilon/(1 - \epsilon_2)$ in the second case as well. For both cases, the improvement of SMC-PF over B-PF is $4\epsilon/(1 - \epsilon_2) = \mathcal{O}(\epsilon)$. This gain is achieved for every summand and thus for the overall sum in (21). Hence, under A4 a factor $\mathcal{O}(\epsilon)$ improvement is established for the per-step variance in (20), and the proof is complete. \square

B. Asymptotic analysis

Proposition 1 is important because it provides an approximate performance analysis for finite M . Even though PFs in practice always entail a finite M , it is also critical to ensure that the novel SMC-PF achieves zero error asymptotically as $M \rightarrow \infty$. For B-PF, this line of convergence results can be found in [9] and [26]. In order to tailor related theorems for the SMC-PF, it is necessary to define an equivalent system with a properly modified likelihood, and state-transition kernel. To this end, consider factorizing the posterior as

$$\begin{aligned} p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) &\propto p(\mathbf{x}_0) \prod_{\tau=1}^k p(\mathbf{y}_\tau|\mathbf{x}_\tau)p(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) \\ &= p(\mathbf{x}_0) \prod_{\tau=1}^k \frac{p(\mathbf{y}_\tau|\mathbf{x}_\tau)p(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})}{\pi_{\mathcal{E}_\tau}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})} \\ &\quad \times \pi_{\mathcal{E}_\tau}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) \\ &:= p(\mathbf{x}_0) \prod_{\tau=1}^k \tilde{p}(\mathbf{y}_\tau|\mathbf{x}_\tau)\tilde{p}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) \end{aligned} \quad (27)$$

where $\tilde{p}(\mathbf{y}_\tau|\mathbf{x}_\tau) := p(\mathbf{y}_\tau|\mathbf{x}_\tau)p(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})/\pi_{\mathcal{E}_\tau}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})$ and $\tilde{p}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}) := \pi_{\mathcal{E}_\tau}(\mathbf{x}_\tau|\mathbf{x}_{\tau-1})$. To guarantee that $\tilde{p}(\mathbf{y}_k|\mathbf{x}_k)$ and $\tilde{p}(\mathbf{x}_k|\mathbf{x}_{k-1})$ satisfy the conditions for convergence in [9] and [26], it will be necessary to invoke the following additional assumptions.

A5. It holds that $\alpha_k, \beta_k > 0$, and $p(\mathbf{y}_k|\mathbf{x}_k)$ is a bounded function of \mathbf{x}_k for the given \mathbf{y}_k .

A6. The following lower bound (ν_k) is valid:

$$\int_{\mathcal{E}_k} p(\mathbf{y}_k|\mathbf{x}_k) \left(\int_{\mathcal{E}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1} \right) d\mathbf{x}_k > \nu_k > 0.$$

A7. The prior density is finite, that is $p(\mathbf{x}_k|\mathbf{x}_{k-1}) < \infty$.

A8. With $p(\mathbf{y}_k|\mathbf{x}_k)$ satisfying A5, let \mathcal{L}_k^4 denote the class of all Borel-measurable functions $\phi : \mathbb{R}^{d_x} \rightarrow \mathbb{R}$ upper-bounded by a constant function $\bar{B}(\mathbf{y}_{1:k})$, that is

$$\sup_{\mathbf{x}_k} |\phi(\mathbf{x}_k)|^4 p(\mathbf{y}_k|\mathbf{x}_k) < \bar{B}(\mathbf{y}_{1:k}). \quad (28)$$

Corollary 1. Given A5 and for any bounded Borel-measurable function $\phi : \mathbb{R}^{d_x} \rightarrow \mathbb{R}$, the SMC-PF estimator has its per-step error variance bounded as

$$E \left[\left(\hat{I}_\phi(k) - I_\phi(k) \right)^2 \right] \leq C(k) \frac{\|\phi\|^2}{M} \quad (29)$$

where $C(k)$ is a constant, and $\|\phi\| := \sup_{\mathbf{x}} |\phi(\mathbf{x})|$.

Proof: If A5 holds, then the required conditions for [9, Theorem 2] are satisfied for the SMC-PF, which readily establishes the validity of (29). \square

Corollary 1 asserts convergence in the mean-square sense of the SMC-PF estimate $\hat{I}_\phi(k)$ to the MMSE estimate $I_\phi(k)$. It implies that as $M \rightarrow \infty$, the per-step error variance of SMC-PF vanishes, and MMSE(k) is achieved. However, Corollary 1 applies only to bounded ϕ 's, which excludes many functions of interest such as $\phi(\mathbf{x}_k) = \mathbf{x}_k(i)$. Those are accommodated under Corollary 2.

To apply the next corollary, two minor modifications of SMC-PF are needed. Let $\{\xi_m^l\}_{m=1:M}^{l=1:M}$ denote a set of non-negative weights such that $\sum_{m=1}^M \xi_m^l = \sum_{l=1}^M \xi_m^l = 1$. Instead of drawing $\mathbf{x}_k^{(m)} \sim \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, consider that particles in S2 are generated from a mixture IS density as follows

$$\mathbf{x}_k^{(l)} \sim \sum_{m=1}^M \xi_m^l \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}), \quad \forall l = 1, \dots, M. \quad (30)$$

For each l , drawing particles as in (30) amounts to re-sampling one particle from the IS density described by $\{\xi_m^l, \mathbf{x}_{k-1}^{(m)}\}_{m=1}^M$, and subsequently generating one particle from the corresponding $\pi_{\mathcal{E}_k}$. Selecting $\xi_m^m = 1$ and $\xi_m^l = 0$ with $l \neq m$, one arrives at the unmodified SMC-PF. Once all the M samples are generated as in (30), the second modification proceeds to check whether the generated particles satisfy

$$\frac{1}{M} \sum_{m=1}^M \tilde{p}(\mathbf{y}_k|\mathbf{x}_k^{(m)}) \geq \check{\nu}_k > 0 \quad (31)$$

where $\tilde{p}(\cdot)$ is the density defined after (27), and $\check{\nu}_k$ denotes an estimate of ν_k in A6.

If (31) is satisfied, the weights can be obtained as in S3 of the unmodified SMC-PF; otherwise, the generated particles are discarded, and a new set of particles is drawn according to (30). This process is repeated until (31) is satisfied. It is known that iterating between (30) and (31) is not an infinite loop, so long as $\check{\nu}_k$ is chosen close to or smaller than ν_k in A6 [26].

Corollary 2. Suppose that A5-A8 hold, and the SMC-PF is modified as in (30) and (31). For any function $\phi \in \mathcal{L}_k^4$ the SMC-PF estimator thus satisfies

$$E \left[\left(\hat{I}_\phi(k) - I_\phi(k) \right)^4 \right] \leq C(k) \frac{\|\phi\|_{k,4}^4}{M^2}$$

$$\|\phi\|_{k,4} := \max \left\{ 1, \left[\int |\phi(\mathbf{x}_s)|^4 p(\mathbf{x}_s|\mathbf{y}_{1:s}) d\mathbf{x}_s \right], s \in [1, k] \right\}.$$

It then follows from [26, Corollary 6.1] that

$$\lim_{M \rightarrow \infty} \hat{I}_\phi(k) = I_\phi(k), \quad \text{almost surely.}$$

Proof: A5-A8 ensure that SMC-PF satisfies the conditions H0-H2 of [26, Theorem 6.1]. \square

Corollary 2 considerably broadens the range of allowable ϕ 's by allowing for unbounded functions. It only requires the product of ϕ raised to the fourth power with the likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$ to be bounded away from infinity as a function of \mathbf{x}_k (cf. (28)). All such ϕ 's belong to \mathcal{L}_k^4 . When the measurement noise is Gaussian, the likelihood decays exponentially as $\mathbf{x}_k(i)$ grows. Therefore, all polynomial functions of $\mathbf{x}_k(i)$ will belong to \mathcal{L}_k^4 . For any such function $\phi \in \mathcal{L}_k^4$, Corollary 2 asserts that the fourth-order moment of the per-step error of the SMC-PF vanishes as $M \rightarrow \infty$. Furthermore, the SMC-PF estimator converges to the MMSE one with probability 1.

It should be stressed that Corollaries 1 and 2 are applicable only when exact weights are known in SMC-PF. This pertains to the incremental version of SMC-PF or the consensus-based version with the number of consensus iterations approaching infinity. For consensus with finite number of iterations, a

residual error in particle weights remains, and more sophisticated analysis would be necessary to derive results similar to Corollaries 1 and 2.

VI. EFFICIENT SAMPLING FROM THE SMC IS DENSITY

This section deals with an efficient means of sampling from the adapted IS density $\pi_{\mathcal{E}_k}(\cdot)$. Recall from Section IV-D that sampling from $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ in S2 relies on RS. The well-known problem with RS is its variable (and possibly large) delay before a sample is accepted [3, Chapter 11]. If $\beta_k \ll 1$ in the present setup and $B^{(m)}(\mathcal{E}_k) \ll 1$, then RS incurs unreasonably large delays because almost all samples fall outside \mathcal{E}_k , and are discarded with high probability. Implementing SMC-PF would thus benefit from bounding the maximum delay of RS by allowing only a prescribed maximum of P samples to be rejected. If P samples are rejected and no sample is yet accepted, the recommendation is to resort to IS with a properly chosen surrogate density.

To perform IS, choose as target density $q_t(\mathbf{x}_k) := \pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$, and select the IS one as the mixture

$$q_{IS}(\mathbf{x}_k) := \gamma U(\mathbf{x}_k|\mathcal{E}_k) + (1 - \gamma)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) \quad (32)$$

where $\gamma \in [0, 1]$ is a design parameter, and $U(\mathbf{x}_k|\mathcal{E}_k)$ denotes the density which is uniform over \mathcal{E}_k , and zero outside \mathcal{E}_k . Although γ and q_{IS} are not indexed by m for brevity, these as well as other quantities in this subsection are for a given m , but the results apply to all particles. Since \mathcal{E}_k represents an axes-aligned box, drawing samples uniformly inside \mathcal{E}_k amounts to drawing one independent uniform sample per coordinate. Therefore, sampling from the surrogate $q_{IS}(\mathbf{x}_k)$ in (32) is easy.

Next, draw M' samples $\check{\mathbf{x}}_k^{(m')}$ from $q_{IS}(\mathbf{x}_k)$, and weigh them as

$$\check{w}_k^{(m')} \propto \frac{(\alpha_k \mathbb{1}_{\{\check{\mathbf{x}}_k^{(m')} \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\check{\mathbf{x}}_k^{(m')} \in \bar{\mathcal{E}}_k\}}) p(\check{\mathbf{x}}_k^{(m')}|\mathbf{x}_{k-1}^{(m)})}{\gamma U(\check{\mathbf{x}}_k^{(m')}|\mathcal{E}_k) + (1 - \gamma)p(\check{\mathbf{x}}_k^{(m')}|\mathbf{x}_{k-1}^{(m)})}.$$

If a particle is now re-sampled from $\{\check{\mathbf{x}}_k^{(m')}, \check{w}_k^{(m')}\}_{m'=1}^{M'}$, it will be approximately coming from $\pi_{\mathcal{E}_k}(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$.

To optimize this IS process, γ will be selected to minimize the mean-square of IS weights, which is equivalent to minimizing Var_{IS} (cf. (4)). This mean-square is given by

$$\begin{aligned} E[\check{w}_k^2] &= \int \frac{(\alpha_k \mathbb{1}_{\{\mathbf{x}_k \in \mathcal{E}_k\}} + \beta_k \mathbb{1}_{\{\mathbf{x}_k \in \bar{\mathcal{E}}_k\}})^2 p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\gamma U(\mathbf{x}_k|\mathcal{E}_k) + (1 - \gamma)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})} d\mathbf{x}_k \\ &= \int_{\mathcal{E}_k} \frac{\alpha_k^2 p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{\gamma U(\mathbf{x}_k|\mathcal{E}_k) + (1 - \gamma)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})} d\mathbf{x}_k \\ &\quad + \int_{\bar{\mathcal{E}}_k} \frac{\beta_k^2 p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})}{(1 - \gamma)} d\mathbf{x}_k \\ &\approx \frac{\alpha_k^2 \text{Vol}(\mathcal{E}_k)}{\gamma} \int_{\mathcal{E}_k} p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \\ &\quad + \frac{\beta_k^2}{1 - \gamma} \int_{\bar{\mathcal{E}}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k := \frac{C_1}{\gamma} + \frac{C_2}{1 - \gamma} \quad (33) \end{aligned}$$

where in obtaining the last approximation it was assumed that $U(\mathbf{x}_k|\mathcal{E}_k) \gg p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ for all $\mathbf{x}_k \in \mathcal{E}_k$; and used

the fact that $U(\mathbf{x}_k|\mathcal{E}_k) = 1/\text{Vol}(\mathcal{E}_k)$, where Vol represents the space volume. The aforementioned assumption is justified because performing IS presumes that the RS stage has failed. Hence, $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ has a very low probability mass inside \mathcal{E}_k . Minimizing (33) with respect to γ leads to

$$\gamma^* = \frac{\sqrt{C_1}}{\sqrt{C_1} + \sqrt{C_2}}.$$

What remains is to obtain, at least approximately, the constants C_1 and C_2 . To specify C_2 , it suffices to recognize that most of $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ mass should lie outside \mathcal{E}_k ; hence,

$$C_2 := \beta_k^2 \int_{\bar{\mathcal{E}}_k} p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \approx \beta_k^2.$$

For C_1 , note that out of P samples drawn from $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ none falls inside \mathcal{E}_k ; hence, \mathcal{E}_k contains less than $1/P$ of the probability mass. In essence, the approximation $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) \approx U(\mathbf{x}_k|\mathcal{E}_k)/P$ is invoked inside \mathcal{E}_k . Thus, C_1 can be expressed as

$$\begin{aligned} C_1 &:= \alpha_k^2 \text{Vol}(\mathcal{E}_k) \int_{\mathcal{E}_k} p^2(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}) d\mathbf{x}_k \\ &\approx \alpha_k^2 \text{Vol}(\mathcal{E}_k) \int_{\mathcal{E}_k} \frac{U^2(\mathbf{x}_k|\mathcal{E}_k)}{P^2} d\mathbf{x}_k = \frac{\alpha_k^2}{P^2}. \end{aligned}$$

Upon substituting C_1 and C_2 back into γ^* , one arrives at

$$\gamma^* = \frac{\alpha_k}{\alpha_k + P\beta_k}.$$

Remark 6. (Computational Cost) Due to the RS step, the exact computational complexity of SMC-PF is a random quantity. However, the IS algorithm proposed in this section provides a deterministic upper bound on complexity. Denoting the complexity of B-PF with M particles as $\text{bpf}(M)$, which is $\mathcal{O}(M)$, the worst-case complexity per time step and sensor of SMC-PF is approximately $\text{bpf}(LM) + M \text{bpf}(M') + M \text{Sample}_{\text{Comp}}(P) + \text{Weight}_{(15)}(M)$, where $\text{Sample}_{\text{Comp}}(P)$ represents the complexity of sampling P particles from the prior and checking if they fall in the set \mathcal{E}_k , and $\text{Weight}_{(15)}(M)$ stands for the complexity of updating the weights in (15). Note that at the beginning of each step sensors run a local B-PF with LM particles; hence, the term $\text{bpf}(LM)$, which is $\mathcal{O}(M)$, appears in the complexity. After consenting on the global set, sensors sample from the set-membership density. In the worst case, the RS steps fail and one has to resort to the IS method described in this section. For the failed RS steps, the complexity is $M \text{Sample}_{\text{Comp}}(P)$, that is also $\mathcal{O}(M)$; while for the IS step, the complexity is $M \text{bpf}(M')$, which is $\mathcal{O}(M)$ as well. Finally, one has to update the weights of the sampled particles whose complexity is given by $\text{Weight}_{(15)}(M)$ that is also $\mathcal{O}(M)$. Since each individual summand in the complexity of SMC-PF is $\mathcal{O}(M)$, the sum is also $\mathcal{O}(M)$. In a nutshell, SMC-PF is computationally more demanding than B-PF, but the complexity of both is linear with respect to the number of particles M .

VII. SIMULATIONS

A network with 100 sensors is considered organized in 20 clusters. Each sensor communicates its distance measurement

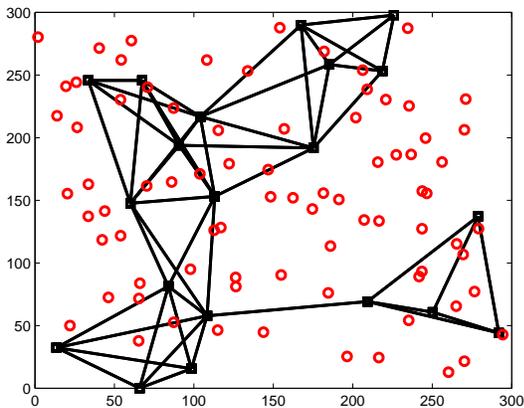


Fig. 6. Sensor network and its communication graph

from the target to its cluster-head sensor. Cluster-head sensors collaboratively perform tracking. For a fixed number of clusters, as the number of sensors increases so does the dimension of data available to cluster-head sensors. In this case, it is prohibitive to communicate raw measurements across the network and distributed approaches should be utilized instead. Fig. 6 depicts the network setup along with the associated communication graph. Sensors are depicted as circles and cluster-heads as squares.

The target moves according to a white Gaussian acceleration model [2, p. 273]. The continuous-time state vector $\mathbf{x}(t) := [x_1(t), x_2(t), \dot{x}_1(t), \dot{x}_2(t)]^T$ comprises the target coordinates in two dimensions along with their derivatives. After sampling the continuous model with period T_s (set throughout the simulations to $T_s = 1$), the state equation in discrete time becomes

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{w}_k$$

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{bmatrix}.$$

The measurement equation expressing the distance from sensor n (located at position \mathbf{s}_n) to the target in the presence of zero-mean, unit-variance, additive white Gaussian noise v_k^n , is given by

$$y_k^n = \frac{c_n}{250 + \|\mathbf{s}_n - \mathbf{x}_k(1:2)\|^2} + v_k^n \quad (34)$$

where the target position is denoted by the vector $\mathbf{x}_k(1:2)$ using MATLAB notation; and the constant c_n is adjusted so that the signal-to-noise ratio (SNR) in (34) is 10dB at a 60 meter distance. These numbers are selected to ensure a sufficiently peaky likelihood function highlighting the gains of SMC-PF relative to B-PF. On the other hand, these numbers and specifically 250 in the denominator of (34) prevent a very peaky likelihood, which may lead to divergence of all filters.

The time-averaged root MSE (RMSE) of $\hat{\mathbf{x}}_k(1:2)$ is used

to assess performance and is defined as

$$\text{Average RMSE} := \left(\frac{1}{J} \sum_{j=1}^J \frac{1}{(K_{\max} - K_{\min})} \times \sum_{k=K_{\min}+1}^{K_{\max}} \|\mathbf{x}_k^{(j)}(1:2) - \hat{\mathbf{x}}_k^{(j)}(1:2)\|_2^2 \right)^{1/2}$$

where j indexes Monte Carlo runs, $J = 100$ represents the total number of Monte Carlo runs, and $K_{\min} = 8$, $K_{\max} = 16$ specify the time interval over which the MSE is averaged. A non-zero K_{\min} is selected to ensure that choice of initialization does not greatly impact the average RMSE, while $K_{\max} = 16$ is chosen to ensure that the target stays in the sensing area for most realizations. Furthermore, the state noise standard deviation is $\sigma_w = 10$. SMC-PF parameters are $\beta_k = 0.001$, $P = 200$, $L = 10$, and $M' = 100$. Unless stated otherwise, the particle bounding box method is employed for SMC-PF. Furthermore, the efficient sampling approach of Section VI is utilized for SMC-PF. The simulation run-time for SMC-PF directly depends on the choice of parameters β_k and P . A larger value of β_k increases simulation speed as particles are accepted with higher probability, at the price of a possible performance degradation. On the other hand, a smaller P also increases simulation speed as transition from the RS stage to the IS stage is faster, but can also incur loss in performance. The values selected for these parameters lead to a reasonable trade-off between run-time and performance.

A. Comparison with B-PF

Fig. 7 compares the performance of SMC-PF with that of B-PF and a benchmark centralized PF, which comprises the combination of the auxiliary particle filter (APF) and the unscented particle filter (UPF). In the APF - UPF combination, the two-stage weighting approach of APF is combined with the importance density obtained from UPF. The consensus averaging required for distributed implementation of B-PF and SMC-PF is assumed to be perfect in these figures. Clearly, SMC-PF outperforms B-PF and its performance comes very close to that of the benchmark filter.

The effect of finite number of consensus-averaging iterations is demonstrated in Fig. 8. The approach described in Remark 1 is utilized to mitigate residual mismatch between particle weights. With as few as 6 consensus-averaging iterations (note that 6 is also the diameter of the communication graph), the performance of SMC-PF comes close to that with perfect consensus averaging.

Finally, RMSE is plotted versus communication cost in Fig. 9. To generate each square in this plot, a single data point corresponding to a given number of particles and given number of consensus iterations is selected from Fig. 8. Then, the communication cost for this data point is computed and the result is plotted as a single square in Fig. 9. Repeating this process for every data point in Fig. 8 yields all the squares (and similarly triangles) in Fig. 9. Thanks to its distributed adaptation, SMC-PF entails lower communication overhead relative to B-PF for a given RMSE in Fig. 9. The difference is considerable especially for small RMSE values.

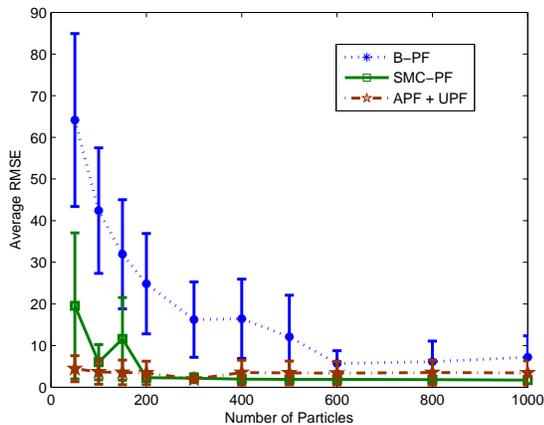


Fig. 7. Comparison (mean \pm one standard deviation) of B-PF, SMC-PF, and APF combined with UPF as benchmark

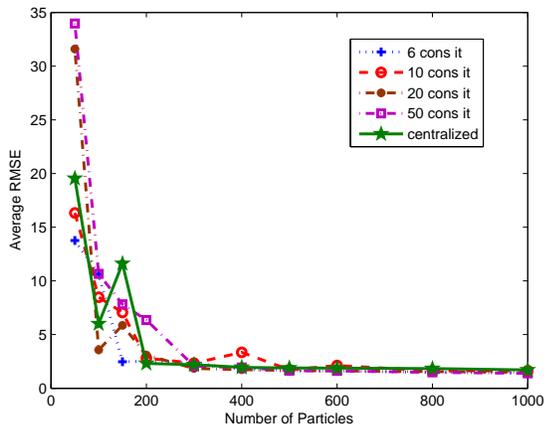


Fig. 8. SMC-PF with different number of consensus iterations

As the number of particles grows large, both SMC-PF (all variants) and B-PF converge to the true MMSE estimator $E[\mathbf{x}_k | \mathbf{y}_{1:k}]$. However, these asymptotic results can be deceiving as the finite-sample performance of the two approaches can be dramatically different; see Fig. 7. In addition, the threshold at which the asymptotic results “kick in” can vary considerably depending on the choice of the IS density $\pi(\cdot)$ [10]. Our simulations corroborate that in a practical setup with a finite number of particles, the SMC-PF markedly outperforms the B-PF, while also offering an affordable distributed implementation.

B. Comparison with Distributed GMM-based PFs

SMC-PF is compared here against the consensus-based algorithm in [36], which will be referred to as the distributed PF-1 (DPF-1), and the incremental algorithm in [25], which will be referred to as DPF-2. While comparing an incremental algorithm with a consensus-based one might be unfair as they belong to different classes, the sensor selection approach introduced in [25] to establish the incremental loop allows one to conduct a fair comparison. The proposed method does not require a Hamiltonian cycle. In fact, each sensor in the

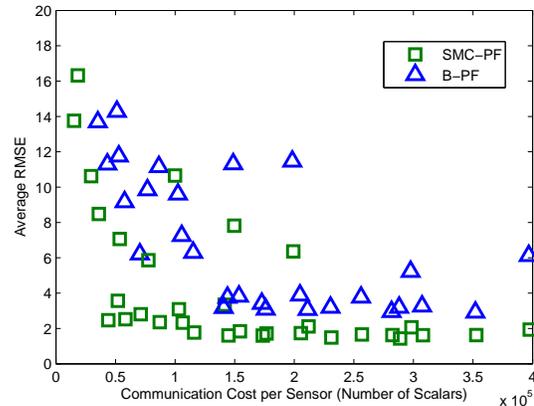


Fig. 9. RMSE versus communication complexity for B-PF and SMC-PF

incremental loop checks its neighbors and determines the ones who have not participated in the incremental loop per round. Among these, the “current” sensor selects the one closest to the target as the next sensor in the loop. When one arrives at a sensor whose neighbors have already participated, the incremental loop ends and the loop for the next round of measurements begins. This approach may lead to sensors left out.

Fig. 10 compares SMC-PF with DPF-1 and DPF-2. Here, 200 consensus averaging iterations are performed for DPF-1, and the number of GMM components in DPF-2 is set to 4. It can be seen that SMC-PF outperforms both methods in terms of RMSE. The communication cost of all methods is plotted in Fig. 11 versus RMSE. One observes that DPF-2 incurs the lowest communication overhead, but its performance is worse than what is achievable by DPF-1 and SMC-PF for higher communication cost. SMC-PF also outperforms DPF-1 with smaller communication complexity. Note that the stars corresponding to DPF-2 in Fig. 11 are obtained from their respective data points in Fig. 10 for different number of particles. Since DPF-2 communicates GMMs rather than particles, its communication complexity is independent of the number of particles and thus all stars fall on the same vertical line in Fig. 11. Similarly, the data point for DPF-1 is obtained from Fig. 10. Only one data point is plotted as all other ones yielded much higher RMSEs. Finally, we should note that while 6 consensus-averaging iterations seem to be enough for SMC-PF, DPF-1 requires close to 200 or more iterations to yield sufficiently small RMSEs. The main reason for this difference is that consenting on the average of values that are spread, requires more iterations than that of concentrated values. Due to the re-sampling phase, the weights are all given by $1/M$ prior to multiplication by the likelihood. After multiplication by the local likelihood, these weights take values in the interval $[0, 1/M]$; hence, consensus-averaging is run among very concentrated values. On the other hand, the means and covariances that DPF-1 aims to consent on, can have values spread far away from each other; hence, consensus-averaging requires more iterations to converge.

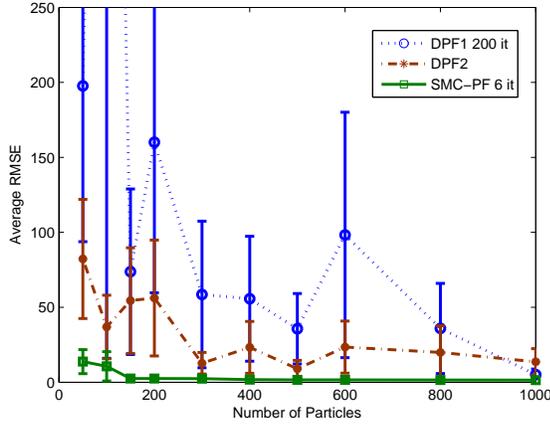


Fig. 10. RMSE comparison (mean \pm one standard deviation) of different DPF algorithms

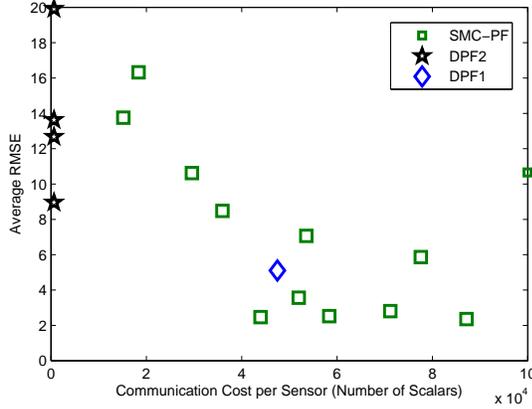


Fig. 11. RMSE versus communication complexity comparison of different DPF algorithms

C. Different SMC-PF Implementations

In this section, 20 sensors are considered without cluster heads, collecting measurements

$$y_k^n = \frac{c_n}{1 + \|\mathbf{s}_n - \mathbf{x}_k(1:2)\|^2} + v_k^n \quad (35)$$

where c_n is chosen to have an SNR of 10 dB at distance equal to 150 meters. Due to the peaky-likelihood conditions, divergent estimated tracks appear and affect the RMSE. To assess performance of each algorithm, the percentage of accurate tracks is plotted as a metric. The latter is defined as the fraction of Monte Carlo runs in which the estimated position error stays below a given RMSE threshold (set equal to 10 in the simulated tests)

$$\left\| \hat{\mathbf{x}}_k^{(j)}(1:2) - \mathbf{x}_k^{(j)}(1:2) \right\| \leq \text{RMSE}_{thr}$$

for all k such that $K_{\min} \leq k \leq K_{\max}$.

Fig. 12 demonstrates the effect of β_k on performance. SMC-PF exhibits robustness to changes in β_k , and only when β_k increases beyond 0.1 its performance begins to deteriorate.

The effect of measurement noise is illustrated in Fig. 13, where the SNR is reduced by a factor of 9, and approaches

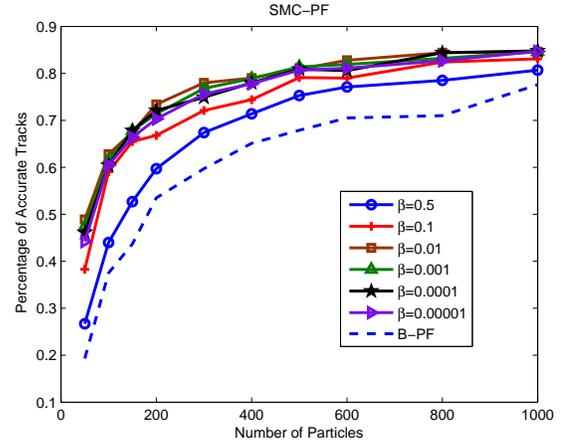


Fig. 12. SMC-PF percentage of accurate tracks for different values of β_k

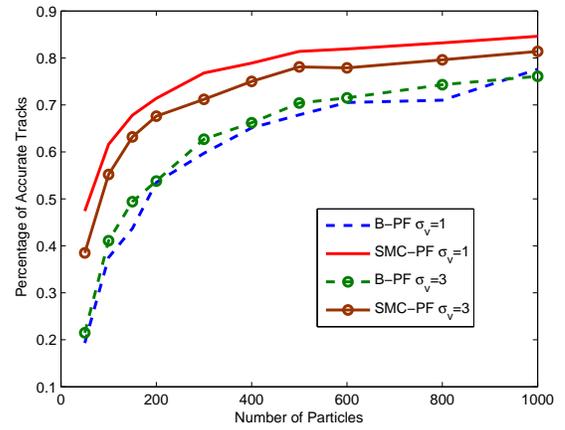


Fig. 13. SNR effect on performance: Percentage of accurate tracks

0 dB at 150 meters distance. In this scenario, an interesting feature emerges. B-PF at SNR= 0 dB outperforms the one at SNR= 10 dB. This is a manifestation of particle depletion, which is more pronounced at 10 dB (more accurate measurements). However, this trend is not observed in SMC-PF whose performance degrades as the SNR decreases.

Different methods for computing the local sets are compared in Fig. 14, where the percentages of accurate tracks are plotted. This figure compares the particle bounding box, the UKF-based set selection, and the UKF with a 4-component GMM for $\beta_k = 0.001$, and $\sigma_w = 10$. Parameter $\eta^2 = 13.277$ is chosen to ensure that 99% of the probability mass is contained in the ellipsoid defined in LS3'. It is observed that the particle bounding box and UKF with GMM approaches outperform the UKF-only by a small margin, which increases as the number of particles grows. In terms of computational complexity, UKF with GMM is the most demanding while UKF-only and the particle bounding box methods exhibit similar and noticeably lower complexity. Overall, for these simulations, the particle bounding box approach offers the best performance-complexity trade-off.

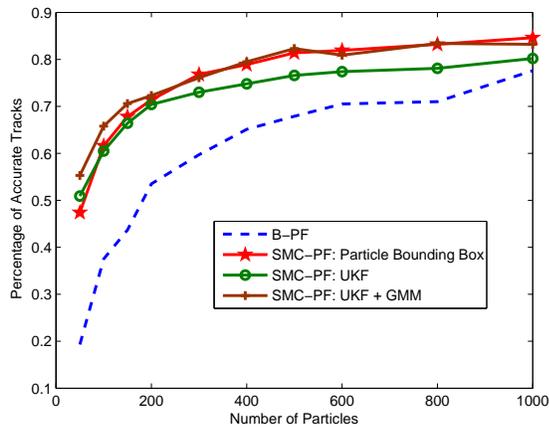


Fig. 14. Local set selection methods: Percentage of accurate tracks

VIII. CONCLUSIONS

A distributed PF scheme was developed to approximate the practically infeasible optimum MMSE state estimator involved when tracking targets using a network of wireless sensors. The novel non-parametric approach offers a decentralized tracker with performance approaching that of centralized PF, affordable complexity, and inter-sensor communications at reduced overhead.

Instrumental to these desirable features was a data-adapted IS density introduced to draw particles per sensor. The novel IS density relies on a judiciously constructed global set, which becomes available to all sensors using a finite number of min- or max-consensus iterations. Membership to this set allows for distributed PF iterations to be run in a connected network by exchanging only particle weights (but not particles) among neighboring sensors.

Asymptotic analysis established consistency of the distributed PF tracker, while finite-sample analysis based on the reduction of the state-estimator variance per PF iteration revealed conditions under which the novel SMC-PF tracker improves upon a bootstrap PF alternative. Simulated tests confirmed that the novel SMC-PF outperforms the bootstrap PF by a wide margin, which becomes more pronounced as measurements become more accurate, and the peaky likelihood conditions are satisfied.

REFERENCES

- [1] A. Ahmad and M. R. Gani, "Distributed estimation for nonlinear stochastic signals over sensor networks," *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, Las Vegas, NV, April 2008.
- [2] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*, New York: Wiley, 2001.
- [3] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [4] N. Bouaynaya and D. Schonfeld, "Complete system with head tracking using motion-based particle filter and randomly perturbed active contour," *Proc. of SPIE*, vol. 5685, pp. 864-873, 2005.
- [5] M. Borkar, V. Cevher, and J. H. McClellan, "Estimating target state distributions in a distributed sensor network using a Monte Carlo approach," *Proc. of Wrksp. on Machine Learning for Signal Processing*, Mystic, CT, Sep. 2005.
- [6] O. Cappe, S. J. Godsill, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, pp. 899-924, May 2007.

- [7] M. Coates, "Distributed particle filters for sensor networks," *Proc. of Intl. Conf. on Information Processing in Sensor Networks*, Berkeley, CA, April 2004.
- [8] J. Cortes, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, pp. 1993-2000, Nov. 2006.
- [9] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. on Signal Processing*, vol. 50, no. 3, pp. 736-746, March 2002.
- [10] F. Daum and J. Huang, "Curse of dimensionality and particle filters," *Proc. of Aerosp. Conf.*, Big Sky, MT, March 2003.
- [11] F. Daum and J. Huang, "Nonlinear filters with generalized particle flow," *Proc. of Aerosp. Conf.*, Big Sky, MT, March 2010.
- [12] P. M. Djuric, M. Vemula, and M. F. Bugallo, "Target tracking by particle filtering in binary sensor networks," *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp. 2229-2238, June 2008.
- [13] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.
- [14] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *J. of Stat. and Computing*, vol. 10, no. 3, pp. 197-208, 2000.
- [15] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Ed. Oxford University Press, 2010.
- [16] M. Evans, "Chaining via annealing," *Annals of Statistics*, vol. 19, no. 1, pp. 382-393, 1991.
- [17] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, "Particle filter adaptation for distributed sensors via set-membership," *Proc. of Intl. Conf. on Acoustics, Speech, and Signal Processing*, Dallas, TX, March 2010.
- [18] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, "Particle filters for mobile robot localization," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Ed. Springer, 2001, pp. 405-428.
- [19] W. Gao, H. Zhao, C. Song, and J. Xu, "A new distributed particle filtering for WSN target tracking," *Proc. of Intl. Conf. on Signal Processing Systems*, Singapore, May 2009.
- [20] J. Geweke, "Bayesian inference in econometric models using Monte Carlo integration," *Econometrica*, vol. 57, no. 6, pp. 1317-1339, Nov. 1989.
- [21] S. Godsill and T. Clapp, "Improvement strategies for Monte Carlo particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Ed. Springer, 2001, pp. 139-158.
- [22] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140, no. 2, pp. 107-113, April 1993.
- [23] D. Gu, "Distributed particle filter for target tracking," *Proc. of Intl. Conf. on Robotics and Automation*, Rome, Italy, April 2007.
- [24] D. Gu, J. Sun, Z. Hu, and H. Li, "Consensus based distributed particle filter in sensor networks," *Proc. of Intl. Conf. on Information and Automation*, Zhangjiajie, P. R. China, June 2008.
- [25] O. Hlinka, P. M. Djuric, and F. Hlawatsch, "Time-space-sequential particle filtering with low-rate communications," *Proc. of Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, CA, Nov. 2009.
- [26] X.-L. Hu, T. B. Schon, and L. Ljung, "A basic convergence result for particle filtering," *IEEE Trans. on Signal Processing*, vol. 56, no. 4, pp. 1337-1348, April 2008.
- [27] K. Ito and K. Xiong, "Gaussian filters for nonlinear filtering problems," *IEEE Trans. on Automatic Control*, vol. 45, no. 5, pp. 910-927, May 2000.
- [28] S. Julier, J. Uhlmann, and H.F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. on Automatic Control*, vol. 45, no. 3, pp. 477-482, March 2000.
- [29] N. Kabaoglu and H. A. Cirpan, "Wideband target tracking by using SVR-based sequential Monte Carlo method," *Signal Processing*, vol. 88, no. 11, pp. 2804-2816, Nov. 2008.
- [30] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1993.
- [31] T. Lefebvre, H. Bruyninckx, and J. De Schutter, "Comment on 'A new method for nonlinear transformation of means and covariances in filters and estimators,'" *IEEE Trans. on Auto. Control*, vol. 47, pp. 1406-1408, Aug. 2002.

- [32] S. Lee and M. West, "Markov chain distributed particle filters (MCDPF)," *Proc. of Conf. on Decision and Control*, Shanghai, P. R. China, Dec. 2009.
- [33] J. S. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statistics and Computing*, vol. 6, no. 2, pp. 113-119, June 1996.
- [34] H. Liu, H. So, F. Chan, and K. Lui, "Distributed particle filtering for target tracking in sensor networks," *Progress in Electromagnetics Research C*, vol. 11, pp. 171-182, 2009.
- [35] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," *Technical Report CUED/F-INFENG/TR 380*, Cambridge Univ., 2000.
- [36] B. N. Oreshkin and M. J. Coates, "Asynchronous distributed particle filter via decentralized evaluation of Gaussian products," *Proc. of Intl. Conf. on Information Fusion*, Edinburgh, Scotland, July 2010.
- [37] M. Orton and W. Fitzgerald, "A Bayesian approach to tracking multiple targets using sensor arrays and particle filters," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 216-223, Feb. 2002.
- [38] O. Ozdemir, R. Niu, and P. K. Varshney, "Tracking in wireless sensor networks using particle filtering: Physical layer considerations," *IEEE Trans. on Signal Processing*, vol. 57, no. 5, pp. 1987-1999, May 2009.
- [39] M. K. Pitt and N. Shephard, "Auxiliary variable based particle filters," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Ed. Springer, 2001, pp. 273-293.
- [40] M. Rosencrantz, G. Gordon, and S. Thrun, "Decentralized sensor fusion with distributed particle filters," *Proc. of Conf. on Uncertainty in Artificial Intelligence*, Acapulco, Mexico, Aug. 2003.
- [41] F. C. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Trans. on Automatic Control*, vol. AC-13, no. 1, pp. 22-28, Feb. 1968.
- [42] Z. Sheng, Y.-H. Hu, and P. Ramanathan, "Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network," *Proc. of Intl. Conf. on Information Processing in Sensor Networks*, Los Angeles, CA, April 2005.
- [43] C. Song, H. Zhao, and W. Jing, "Asynchronous distributed PF algorithm for WSN target tracking," *Proc. of Intl. Conf. on Wireless Communications and Mobile Computing*, Leipzig, Germany, June 2009.
- [44] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Pearson Prentice Hall, 2005.
- [45] P. Torma and C. Szepesvari, "Local importance sampling: A novel technique to enhance particle filtering," *Journal of Multimedia*, vol. 1, no. 1, pp. 32-43, April 2006.
- [46] J. Wu and H. Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *Telecommun. Syst. J.*, vol. 3, pp. 6384, Sep. 2001.
- [47] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Sys. and Control Ltrs.*, pp. 65-78, Sept. 2004.



Stergios I. Roumeliotis (M'02) received the Diploma in Electrical Engineering from the National Technical University of Athens, Greece, in 1995, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Southern California, CA in 1999 and 2000 respectively. From 2000 to 2002 he was a Postdoctoral Fellow at the California Institute of Technology, CA. Between 2002 and 2008 he was an Assistant Professor with the Department of Computer Science and Engineering, University of Minnesota, MN, where he is currently an Associate

Professor. Since 2009, S.I. Roumeliotis is the Associate Director for Research of the Digital Technology Center (DTC). His research interests include vision-aided inertial navigation of aerial and ground autonomous vehicles, distributed estimation under communication and processing constraints, and active sensing for reconfigurable networks of mobile sensors.

S.I. Roumeliotis is the recipient of the Guillermo E. Borja Award (2009), the National Science Foundation (NSF) Presidential Early Career Award for Scientists and Engineers (PECASE) (2008), the NSF CAREER award (2006), the McKnight Land-Grant Professorship award (2006-08), the ICRA Best Reviewer Award (2006), and he is the co-recipient of the One NASA Peer award (2006), and the One NASA Center Best award (2006). Papers he has co-authored have received the King-Sun Fu Best Paper Award of the IEEE Transactions on Robotics (2009), the Robotics Society of Japan Best Journal Paper award (2007), the ICASSP Best Student Paper award (2006), the NASA Tech Briefs award (2004), and three of them were Finalists for the RSS Best Paper Award (2009), the ICRA Best Student Paper Award (2009) and the IROS Best Paper Award (2006). S.I. Roumeliotis served as Associate Editor for the IEEE Transactions on Robotics between 2006 and 2010.



Georgios B. Giannakis (Fellow'97) received his Diploma in Electrical Engr. from the Ntl. Tech. Univ. of Athens, Greece, 1981. From 1982 to 1986 he was with the Univ. of Southern California (USC), where he received his MSc. in Electrical Engineering, 1983, MSc. in Mathematics, 1986, and Ph.D. in Electrical Engr., 1986. Since 1999 he has been a professor with the Univ. of Minnesota, where he now holds an ADC Chair in Wireless Telecommunications in the ECE Department, and serves as director of the Digital Technology Center.

His general interests span the areas of communications, networking and statistical signal processing - subjects on which he has published more than 300 journal papers, 500 conference papers, 20 book chapters, two edited books and two research monographs. Current research focuses on compressive sensing, cognitive radios, network coding, cross-layer designs, wireless sensors, social and power grid networks. He is the (co-) inventor of twenty patents issued, and the (co-) recipient of seven paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award in Wireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, and the G. W. Taylor Award for Distinguished Research from the University of Minnesota. He is a Fellow of EURASIP, and has served the IEEE in a number of posts, including that of a Distinguished Lecturer for the IEEE-SP Society.



Shahrokh Farahmand received the B.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran in 2003, and his M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Minnesota at Twin Cities, in 2006 and 2011, respectively.

His general area of interest lies in estimation theory with specific focus on target tracking algorithms and dynamic state estimation.