PARTICLE FILTER ADAPTATION FOR DISTRIBUTED SENSORS VIA SET MEMBERSHIP

Shahrokh Farahmand[†], Stergios I. Roumeliotis[‡], and Georgios B. Giannakis[†]

ECE[†]/CS[‡] Dept., Univ. of Minnesota, 200 Union Street SE, Minneapolis, MN 55455

ABSTRACT

A distributed set-membership-constrained particle filter (SMC-PF) is developed for decentralized tracking applications using wireless sensor networks. Unlike existing PF alternatives, SMC-PF offers reduced overhead for inter-sensor communications because it requires only particle weights to be exchanged among sensors, instead of raw measurements or parameters of a Gaussian mixture model. SMC-PF relies on a novel distributed adaptation scheme based on successive set intersections that can afford reduced number of particles without sacrificing performance. Conditions are provided to quantify the variance reduction of the SMC-PF-based state estimator. Simulations corroborate the ability of the SMC-PF to considerably outperform the bootstrap PF for a fixed number of particles.

Index Terms— particle filtering, set-membership, distributed, adaptation.

1. INTRODUCTION

High-variance measurement noise and nonlinearities present in data models encountered with e.g., tracking of moving targets, often cause linearized estimators, such as the extended Kalman filter, to yield inconsistent estimates. Non-parametric and nonlinear estimators obtained using e.g., the particle filter (PF), can improve estimation accuracy at the price of increasing computational complexity. Besides complexity, when distributed sensors are deployed to perform decentralized tracking tasks, coping with the overhead associated with inter-sensor communications presents an additional challenge. This overhead can be prohibitively high in lieu of a fusion center (FC), where sensors need to utilize incremental or consensus algorithms.

Existing sub-optimum approaches to address these challenges include limiting the global (network-wide) information available to local (per-sensor) estimators [1], ad-hoc queries and responses [10], as well as Gaussian mixture model (GMM) approximations of either the likelihood or the posterior density to control the amount of inter-sensor communications [3, 12, 15]. GMM-based approaches entail covariance matrix exchanges at quadratic communication cost, and typically suffer from extensive delays caused by computations performed sequentially across sensors.

The present work develops an alternative approach requiring only particle weights to be exchanged among sensors. Different from centralized approaches [7, 9], a novel

distributed adaptation scheme is introduced based on set intersections [11] for reducing the number of particles, and hence the number of weights communicated, without sacrificing performance. In addition, conditions are provided to ensure that the proposed set membership constrained PF (SMC-PF) offers the highest accuracy improvement relative to the bootstrap PF.

Performance of SMC-PF is assessed using the one-step PF variance, that is the posterior variance when the particles accurately represent the current prior. This metric can be different from the exact PF variance but is chosen for two reasons: i) presumably the PF density accurately describes the density of interest, meaning that the one-step PF variance stays close to the true PF variance; and ii) the adopted metric is simple as it allows treating the current PF step as an importance sampling instance using the expression in [8]. Simulations demonstrate that the SMC-PF reduces the number of particles required by at least a half compared to the (non-adapted) bootstrap PF, while achieving comparable accuracy.

2. PROBLEM FORMULATION

Consider N sensors distributed randomly on a field to track their distance and bearing from a moving target. At time k, the target's state vector $\mathbf{x}_k$ evolves as $\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_k)$, where $\mathbf{w}_k$ is the process noise, yielding 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 $\mathbf{v}_k^n$ denotes observation noise, yielding the likelihood $p(\mathbf{y}_k^n|\mathbf{x}_k)$. Based on these state and measurement models, the goal is to estimate the target's state in a distributed fashion given sensor measurements.

If the sensors were able to communicate $\mathbf{y}_k^n$ amongst themselves, a centralized PF could be run per sensor based on the aggregate data $\mathbf{y}_k := [(\mathbf{y}_k^1)^T, \ldots, (\mathbf{y}_k^N)^T]^T$. After initialization, such a "centralized" PF would implement the following steps per time instant [2, 6].

S1) Re-sample from the particles/weights $\{\mathbf{x}_{k-1}^{(m)}, w_{k-1}^{(m)}\}_{m=1}^M$ at time $k-1$ (if needed), where M is the number of particles.

S2) Sample $\mathbf{x}_k^{(m)} \sim \pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$ with $\pi(.)$ denoting the importance density.

S3) Update weights through Bayes' rule as

$$w_k^{(m)} \propto w_{k-1}^{(m)} \frac{p(\mathbf{y}_k|\mathbf{x}_k^{(m)})p(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)})}{\pi(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)}$$

and normalize them to sum up to one.

S4) Estimate the state as $\hat{\mathbf{x}}_k = \sum_{m=1}^M w_k^{(m)} \mathbf{x}_k^{(m)}$.

To update the weights in S3, centralized PFs require disseminating $\mathbf{y}_k$ throughout the entire network (e.g., by flooding), which is impractical especially for large N. To avoid this, we first present a distributed PF that requires exchanging only particle weights, and then introduce an adaptation scheme based on set membership for reducing their number.

2.1. Distributed Particle Filtering

Considering that all sensors are initialized with the same random seed [3], the random operations performed per sensor yield the exact same results, provided that they use the same particles $\mathbf{x}_{k-1}^{(m)}$. Thus, S1 and S4 can be performed locally without any communication. Furthermore, invoking the noise independence across sensors, and taking logarithms on both sides of the weight update equation yields (cf. S3)

$$\log\left(w_k^{(m)}\right) \propto \log\left(w_{k-1}^{(m)}\right) + \sum_{n=1}^{N} \log\left(p(\mathbf{y}_k^n|\mathbf{x}_k^{(m)})\right)$$
$$- \log\left(\pi(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)\right) . \quad (1)$$

Because sensors have available the particles $\mathbf{x}_{k-1}^{(m)}$ and their weights, $w_{k-1}^{(m)}$, from the previous time step, the first and last terms on the right hand side are already known for a non-adapted importance density (i.e. $\pi(\mathbf{x}_k^{(m)}|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k) = \pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$). The summation can be efficiently computed via consensus averaging [13]. However, the number of particles increases the communication overhead of S3 considerably because M such consensuses must be run in parallel. As it will become evident, this requirement also limits the options for distributing S2. A straightforward approach to performing S2 in a distributed fashion is to use a non-adapted importance density, such as the prior $\pi(.) = p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})$ (bootstrap PF), which does not require knowledge of $\mathbf{y}_k$. Such a selection of $\pi(.)$ however, requires a large number of particles, which in turn increases the communication cost of S3.

Specifically, an approximate performance analysis of importance sampling, which represents the major operation in PF steps S2 and S3, suggests that in the asymptotic regime the variance $E[(\hat{\mathbf{x}}_k - E[\mathbf{x}_k|\mathbf{y}_{1:k}])^2] \approx C/M$ reduces as M grows [8]. The values of C and M for the asymptotic variance to hold depend on the similarity between the true posterior density $p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$, and the selected importance sampling density $\pi(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$ [5, 7]. To ensure that $p(.)$ and $\pi(.)$ are similar, information in the observations $\mathbf{y}_k$ must be exploited by the selected $\pi(.)$ density, a process which is referred to as *adaptation*. In non-adapted schemes, such as the bootstrap PF, many of the generated particles receive almost zero weights in S3, and have to be discarded since they are generated from a density far different from the true posterior. This often leads to particle depletion, which compromises performance. In contrast, adaptation ensures 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.

In most cases in practice, the importance density $\pi(.) := p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathbf{y}_k)$ (perfect adaptation) is hard to compute or draw samples from, even for centralized PF implementations. To bypass this challenge, the next section introduces an approximate adaptation method, which renders the distributed implementation of S2 efficient.

3. SET-MEMBERSHIP-BASED ADAPTATION

The key idea behind the proposed low-communication cost distributed adaptation scheme is to approximate the posterior density $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k)$ with an appropriately scaled (dis-
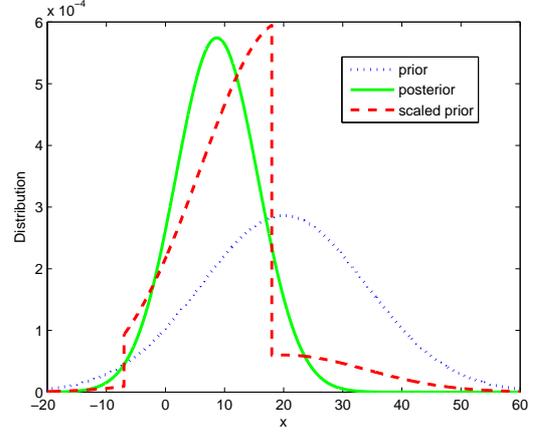

Fig. 1. Comparison of scaled prior to posterior density

torted) version of the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, defined as

$$u(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathcal{E}_k) = \frac{\alpha_k I_{(\mathbf{x}_k \in \mathcal{E}_k)} + \beta_k I_{(\mathbf{x}_k \notin \mathcal{E}_k)}}{c_k} p(\mathbf{x}_k|\mathbf{x}_{k-1})$$
$$(2)$$

where $\beta_k \ll \alpha_k = 1$, $I_{(.)}$ is the indicator function, and c_k represents the normalization constant for $u(.)$ to be a proper density. In this expression, $\mathcal{E}_k := \bigcap_n \mathcal{E}_k^n$ denotes the *global set*, where all sensors have high local posterior masses, while $\mathcal{E}_k^n$ is the *local set*, which contains most of the probability mass of the local posterior $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$.

Before describing the process for selecting $\mathcal{E}_k^n$ and determining the global set $\mathcal{E}_k$, the ensuing sub-section shows that $u(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathcal{E}_k)$ indeed approximates the global posterior.

3.1. Posterior Density Approximation

Suppose that each sensor determines a local set $\mathcal{E}_k^n$ containing most of the probability mass of the local posterior $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)$. The local posterior can then be approximated as $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n) \approx u(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathcal{E}_k^n)$, with $u(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathcal{E}_k^n)$ as in (2). Since the noise across sensors is independent, the global posterior is

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k) \propto \left[\prod_{n=1}^{N} \frac{p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n)}{p(\mathbf{x}_k|\mathbf{x}_{k-1})}\right] p(\mathbf{x}_k|\mathbf{x}_{k-1}) \quad (3)$$

$$\approx \left[\prod_{n=1}^{N} \frac{u(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathcal{E}_k^n)}{p(\mathbf{x}_k|\mathbf{x}_{k-1})}\right] p(\mathbf{x}_k|\mathbf{x}_{k-1}) \approx u(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathcal{E}_k)$$

where the multi-step function $\prod_{n=1}^{N} \frac{u(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathcal{E}_k^n)}{p(\mathbf{x}_k|\mathbf{x}_{k-1})}$ is approximated using the bi-step function $u(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathcal{E}_k)$, which is valid for $\beta_k \ll 1$. Compared to the prior itself, the scaled prior in (2) can be much closer to the posterior, which is the density of interest; see also Fig. 1.

3.2. Local Set Selection

This step is critical since a good choice of the global set hinges on judicious selection of local sets. Here, $\mathcal{E}_k^n$ will be heuristically chosen so that $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_k^n) \approx u(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathcal{E}_k^n)$. Specifically, the following local steps (LS1-LS3) are performed in parallel at each sensor indexed by n.

LS1) Oversample particles and weights $\{\mathbf{x}_{k-1}^{(m)}, w_{k-1}^{(m)}\}_{m=1}^{M}$ to obtain $\{\tilde{\mathbf{x}}_{k-1}^{(m')}, \tilde{w}_{k-1}^{(m')}\}_{m'=1}^{LM}$ with $L \in \mathcal{N}$.
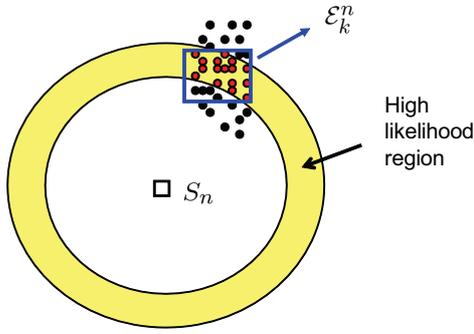
Fig. 2. Graphical depiction of the selection of the local set $\mathcal{E}_k^n$.

LS2) Generate new samples $\tilde{\mathbf{x}}_k$ from the prior $\tilde{\mathbf{x}}_k^{(m')} \sim p(\mathbf{x}_k|\tilde{\mathbf{x}}_{k-1}^{(m')})$. Update the weights by the local likelihood $\tilde{w}_k^{n,(m')} \propto \tilde{w}_{k-1}^{(m')} p(\mathbf{y}_k^n|\tilde{\mathbf{x}}_k^{(m')})$, and normalize them.

LS3) Re-sample M particles from $\{\tilde{\mathbf{x}}_k^{(m')}, \tilde{w}_k^{n,(m')}\}_{m'=1}^{LM}$. Define $\mathcal{E}_k^n$ as the smallest box containing the re-sampled particles.

LS1 and the first part of LS2 produce the same result for all sensors since random operations are performed with the same random seed across sensors. Once the local likelihood is weighted in, the sensors generate different results, which explains why the index n appears. Note that the re-sampled particles at step LS3 come from the distribution $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$, which represents the local posterior per sensor n. We fit a box around these samples, and expect this box to represent the region where $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{y}_k^n)$ contains most of its mass. Fig. 2 depicts the local set selection process for a sensor measuring its distance to a target.

3.3. Global Set Determination

Once each sensor processes its local information $\mathbf{y}_k^n$ to obtain the local set $\mathcal{E}_k^n$, the global set $\mathcal{E}_k$ is computed in a distributed fashion using min- and max-consensus operations. In particular, when boxes are used to represent local sets, computing their intersection is very simple because it only requires computing minima and maxima. Moreover, min- and max-consensus algorithms converge in finite iterations equal to the diameter of the communication graph. Thus, a distributed implementation becomes available at much lower communication cost compared to propagating (or performing consensus of) raw measurements across sensors.

Remark 1. When the intersection is empty, local sets are stretched out by a constant factor, and the intersection is re-computed. This is repeated until the process arrives at a non-empty intersection. Simulations indicate that two repetitions are needed at most with a judicious choice of the scaling factor, while the first intersection is non-empty most of the time.

3.4. Distributed Importance Density Sampling

Once $\mathcal{E}_k$ is available to all sensors, samples are drawn per sensor (cf. S2) from the scaled prior density $\pi(.) = u(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)}, \mathcal{E}_k)$ in (2). Specifically, each sensor first draws samples from the prior $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ itself. If the sample belongs to $\mathcal{E}_k$, it is accepted with high probability ($\alpha_k = 1$). Otherwise, the sample is discarded with high probability; that is the accept probability satisfies $\beta_k \ll 1$. Note that all

sensors draw identical particles due to their initialization with the same random seed. Finally, the remaining steps of the distributed SMC-PF are carried out as outlined in Section 2.1.

Remark 2. The novel approach is referred to as set membership constrained PF for two reasons: i) successive set intersections are computed, which is a trade-mark of set-membership approaches [11], and ii) rejection sampling relies on an accept/reject criterion assessing membership in a set $\mathcal{E}_k$.

4. PERFORMANCE ANALYSIS

Analysis in this section relies on the following assumptions.

A1. Particles and weights $\{\mathbf{x}_{k-1}^{(m)}, w_{k-1}^{(m)}\}_{m=1}^M$ provide an accurate representation of $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$.

A2. There exists a set $\mathcal{E}_k$ such that $p^2(\mathbf{y}_k|\mathbf{x}_k) \leq \epsilon_1$, $\forall \mathbf{x}_k \notin \mathcal{E}_k$ with $\epsilon_1 \ll 1$.

A3. For the set $\mathcal{E}_k$ in **A2**, it holds with $\epsilon_2 \ll 1$ that

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

A4 There exists a constant $C \gg \max\{\epsilon_1, \epsilon_2\}$ such that

$$\int_{\mathcal{E}_k} p^2(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1}^{(m)})d\mathbf{x}_k \geq C, \ \forall m = 1, \ldots, M.$$

A1 is needed since the one-step PF variance is sought to measure performance. The one-step PF variance is equivalent to the importance sampling variance, and can be expressed as in [8]. Together, A2 and A3 represent the so called "peaky likelihood" condition, where the likelihood is significantly more peaky than the prior density. A4 is invoked to ensure that the likelihood is not located at the tail of the prior. Under these assumptions, the performance improvement of the SMC-PF compared to a bootstrap PF is maximized. The pertinent result can be formally stated as follows.

Proposition 1. *Under A1-A4, for a fixed number of particles M, and with $\beta_k = \max\{\epsilon_1, \epsilon_2\}$, the one-step PF variance in estimating $\mathbf{x}_k$ given by $E[(\hat{\mathbf{x}}_k - E[\mathbf{x}_k|\mathbf{y}_{1:k}])^2]$ reduces by a factor of $\max\{\epsilon_1, \epsilon_2\}$ for the SMC-PF compared to the bootstrap PF.* (See [14] for the proof)

5. SIMULATIONS

Simulations are carried out for a field of 100×100 meters with $N = 6$ sensors measuring their distances from a moving target. The SMC-PF always outperforms the bootstrap PF by a clear margin. Figs. 3 and 4 plot, respectively, the time-averaged MSE, and the percentage of accurate tracks versus the number of particles. A track is taken as accurate when the maximum MSE across time of a given track stays below a pre-selected threshold. For a given performance level, the SMC-PF requires at most half of the particles that the bootstrap PF needs, and in many cases considerably less. This is the advantage adaptation offers because particles are drawn from a density much closer to the posterior which is the optimum density. Hence, fewer particles are enough to accurately represent the posterior after updating the weights (c.f. S3). Consequently, communication overhead is cut in half because only half of the consensus averages are required when compared to the bootstrap PF. Similar results were obtained for sensor fields of up to $N = 40$ sensors.
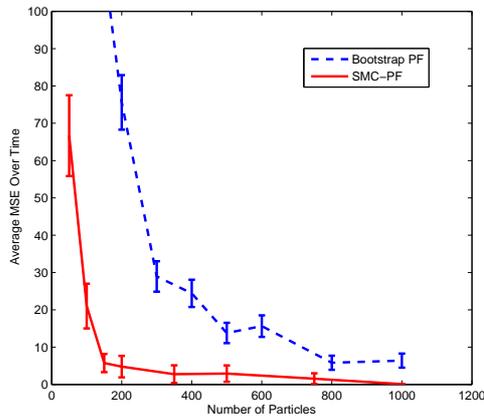
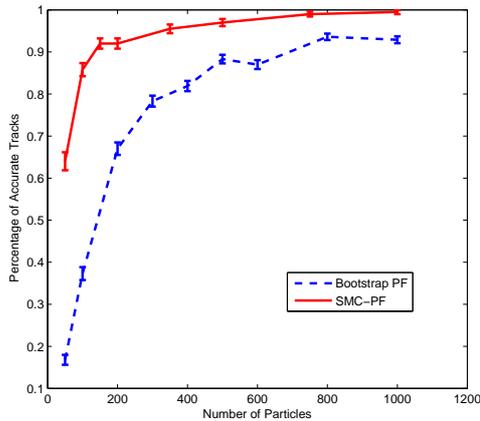**Fig. 3**. Time averaged MSE vs. number of particles

Fig. 4. Percentage of accurate tracks vs. number of particles

Asymptotically, as the number of particles goes to infinity both methods 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 [5]. In addition, the threshold at which the asymptotic results become valid can vary considerably depending on the choice of the importance sampling density $\pi(.)$ [5]. Our simulations corroborated that in a practical setup with finite samples, the SMC-PF markedly outperforms the bootstrap PF, while it can also afford a simpler distributed implementation.

The local sets are chosen heuristically in Section 3.2. Ongoing work focuses on determining optimal ways of selecting the local sets by minimizing the one-step PF variance[1].

6. REFERENCES

[1] A. Ahmad and M. R. Gani, "Distributed estimation for nonlinear stochastic signals over sensor networks," *Proc. of ICASSP*, Las Vegas, NV, April 2008.

[2] O. Cappe, S. J. Godsil, and E. Moulines, "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, May 2007.

[3] M. Coates, "Distributed particle filters for sensor networks," *Proc. of IPSN*, Berkeley, CA, April 2004.

[4] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Trans. on Signal Processing,* March 2002.

[5] F. Daum and J. Huang, "Curse of dimensionality and particle filters," *Proc. of Aerospace Conf.*, Big Sky, MT, March 2003.

[6] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[7] 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.

[8] J. S. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Stat. and Computing*, 1996.

[9] 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.

[10] 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.

[11] F. C. Schweppe, "Recursive state estimation: Unknown but bounded errors and system inputs," *IEEE Trans. on Automatic Control*, pp. 22-28, Feb. 1968.

[12] 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 IPSN*, Los Angeles, CA, April 2005.

[13] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. and Control Letters*, pp. 65-78, Sept. 2004.

[14] S. Farahmand, S. I. Roumeliotis, and G. B. Giannakis, "Particle Filter Adaptation for Distributed Sensors via Set Membership," *IEEE Transactions on Signal Processing*, Jan. 2010 (submitted).

[15] L. Zuo, K. Mehrotra, P. Varshney, and C. Mohan, "Bandwidth-efficient target tracking in distributed sensor networks using particle filters," *Proc. of Intl. Conf. on Info. Fusion*, Florence, Italy, July 2006.

---

[1]The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.