State-Space Partitioning Schemes in Multiple Particle Filtering for Improved Accuracy

Marija Iloska and Mónica F. Bugallo
Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, New York 11794-2350
Email: {marija.iloska, monica.bugallo}@stonybrook.edu

***Abstract*—Multiple particle filtering was proposed as an alternative to particle filtering when tracking states in systems of high dimensions. Multiple particle filters are comprised of a network of particle filters assigned to track subsets of the state, and require each other's obtained information to carry out the filtering. Many improvements of multiple particle filtering have been proposed, however, there have been only a few efforts studying the effects of state partitioning on the filtering performance. In this paper, we propose two novel partitioning schemes for improved accuracy of multiple particle filtering based on: i) random permutations, and ii) the connectedness of the states, i.e. the topology of the system. Computer simulations show that the filter significantly benefits from state permutations, especially when driven by the information in the topology.**

I. Introduction

A complex dynamical system can be often described as a network of interacting components which evolve the system as a whole. The components, also seen as the states of the system, are directly or indirectly observed and their interactions can be nonlinear. Complex systems are very abundant in nature and in engineering. Some examples include gene regulatory networks [1], sensor networks [2], target tracking [3], and chemical reactions [4].

Particle filters (PFs) have been traditionally used for inference and prediction of dynamical systems, especially in online applications. Their popularity rose relative to other filters (e.g., Kalman filter and its variations [5], [6], [7], [8]) as they can deal with systems of nonlinear and non-Gaussian nature. However, their applicability is limited to relatively small systems, as their performance collapses under the *curse of dimensionality* [9]. Unfortunately, many complex systems are characterized by high dimensions and cannot be handled by standard PFs. Additionally, the computational cost of PFs grows with the number of particles used, rendering them as an expensive methodology. Over a decade ago, the multiple particle filter (MPF) was introduced in [10] to tackle the curse of dimensionality. MPFs work by splitting the system into subsystems and assigning a separate PF to track the marginal density of the states representing the subsystem at hand. In order to carry out the filtering, the PFs need to communicate their respective estimates and/or predictions of the marginal densities to each other. The remarkable performance of the MPF has been demonstrated in [3], [11], [12], [13], [14], among others. Several improvements of the MPF have been proposed, such as the filters communicating only means and covariances of the states they track [15], using iterations for obtaining the best possible predictions [16], and other related alternatives (e.g., [17], [18], [19]). There have only been a few approaches addressing state-space partitioning for improved filtering in multiple particle filtering. The work in [20] proposes a probabilistic strategy on adaptive state partitioning based on the cross-correlation of the subsystems at each time instant, however, their focus mainly sits in state-space models (SSMs) in which the states are separable in the process equation, as is usually the case for multiple target tracking. Another approach in [21], proposes a *symbiotic* PF, where the state-space partitioning depends on the distance of the targets tracked. Two given targets are considered connected and can cooperate if they are travelling paths in close proximity. They share one PF until they part ways, in which case the filtering continues by each target getting assigned its own separate PF. The symbiotic PF considers cases in which the connectedness of the components is dynamic and determined based on the distances between them. However, there are scenarios where the relationships between components are well-defined in the model itself and remain fixed. It is also important to note that not all components of the system interact with each other. The connectedness of these components is referred to as the topology of the system, and can play a significant role in system partitioning. To the best of our knowledge, none of the existing approaches specifically target state-space partitioning based on the topology.

In this paper, we propose methods for partitioning of the system in MPFs for improved filtering. In particular, we explore two different partitioning schemes based on: i) uniform randomness, and ii) topology-informed randomness. Computer simulations demonstrate that the MPFs with the newly proposed partitioning schemes outperform the MPF with original partitioning. In fact, the MPF with the topology based partitioning scheme shows the most significant improvement.

II. Problem Formulation

Consider a dynamical system characterized by the SSM:

$$\begin{aligned} \mathbf{x}_0 &\sim p(\mathbf{x}) \\ \mathbf{x}_t &= f_x(\mathbf{x}_{t-1}, \mathbf{u}_t) \\ \mathbf{y}_t &= f_y(\mathbf{x}_t, \mathbf{v}_t), \quad t = 1, \ldots, T, \end{aligned} \tag{1}$$

ISBN: 978-1-6654-6798-8 2026 EUSIPCO 2022

where $\mathbf{x}_t \in \mathbb{R}^{d_x}$ is the state of the system at time t, and $\mathbf{y}_t \in \mathbb{R}^{d_y}$ is a collection of direct and/or indirect observations of $\mathbf{x}_t$. The system evolves according to $f_x(\cdot,\cdot)$ with randomness captured by the process noise $\mathbf{u}_t$, and the states are observed through $f_y(\cdot,\cdot)$ with some observation noise $\mathbf{v}_t$. The system we consider is Markovian [22], and is initialized by sampling the state $\mathbf{x}_0$ from its prior $p(\mathbf{x})$.

III. Multiple Particle Filter

Standard PFs collapse when dealing with systems of high dimensions. MPF, which works on the principle of *divide and conquer* was introduced as a way to alleviate this issue. More formally, an MPF partitions the state at time t as $\mathbf{x}_t = [\mathbf{x}_{1,t}^\top, \mathbf{x}_{2,t}^\top, , \mathbf{x}_{K,t}^\top]^\top$, where $\mathbf{x}_{k,t} \in \mathbb{R}^{d_k}$ is the state vector of partition k. Each partition is addressed by a separate PF, and not all partitions need to be the same size. The kth PF approximates the marginal posterior $p(\mathbf{x}_{k,t}|\mathbf{y}_t)$ by generating a discrete random measure $\mathcal{X}_{k,t} = \left\{\mathbf{x}_{k,t}^{(m_k)}, w_{k,t}^{(m_k)}\right\}_{m_k=1}^{M_k}$, composed of M_k particles $\mathbf{x}_{k,t}$, and their associated weights $w_{k,t}$. The workflow of MPF for one time instant t can be summarized with the following steps:

1. At time t, each filter k proposes M_k particles by sampling

$$\mathbf{x}_{k,t}^{(m_k)} \sim p(\mathbf{x}_{k,t}|\mathbf{x}_{k,t-1}^{(m_k)}, \hat{\mathbf{x}}_{-k,t-1}), \quad m_k = 1,\dots,M_k, \quad (2)$$

where $\hat{\mathbf{x}}_{-k,t-1}$ are the estimates of all the partitions from the previous time step except the kth, i.e. $\hat{\mathbf{x}}_{-k,t-1}^\top = [\hat{\mathbf{x}}_{1,t-1},\dots,\hat{\mathbf{x}}_{k-1,t-1},\hat{\mathbf{x}}_{k+1,t-1},\dots,\hat{\mathbf{x}}_{K,t-1}]$. These estimates are made available by the exchange of information from the other filters at the end of the previous time step.

2. The proposed particles can be used to obtain predictions of the current state $\mathbf{x}_t$ as

$$\tilde{\mathbf{x}}_{k,t} = \frac{1}{M_k}\sum_{m_k=1}^{M_k} \mathbf{x}_{k,t}^{(m_k)}. \quad (3)$$

The kth filter sends these predictions to the rest of filters, and receives the predictions $\tilde{\mathbf{x}}_{-k,t}$ obtained from the rest of the filters, where $\tilde{\mathbf{x}}_{-k,t}$ is defined similarly as $\hat{\mathbf{x}}_{-k,t-1}$.

3. The filters use the obtained predictions and estimates to update the weights according to

$$\tilde{w}_{k,t}^{(m_k)} \propto \tilde{w}_{k,t-1}^{(m_k)} \frac{p(\mathbf{y}_t|\mathbf{x}_{k,t}^{(m_k)}, \tilde{\mathbf{x}}_{-k,t})p(\mathbf{x}_{k,t}^{(m_k)}|\hat{\mathbf{x}}_{-k,t-1})}{q(\mathbf{x}_{k,t}^{(m_k)}|\hat{\mathbf{x}}_{-k,t-1},\mathbf{y}_t)}. \quad (4)$$

Here, the numerator represents the likelihood and the transition density, and the denominator is the proposal distribution evaluated at the m_kth particle (see [10], [22] for thorough definitions and derivations of PF). The updating factor is an approximation as it uses estimates and predictions of all the state partitions excluding the kth element.

4. Normalize the weights as

$$w_{k,t}^{(m_k)} = \frac{\tilde{w}_{k,t}^{(m_k)}}{\sum_{n=1}^{M_k}\tilde{w}_{k,t}^{(n)}}, \quad for \quad m_k = 1,\dots,M_k. \quad (5)$$

5. Obtain the estimates of the states of each partition $\mathbf{x}_{k,t}$ using the weights, for e.g.

$$\hat{\mathbf{x}}_{k,t} = \sum_{m_k=1}^{M_k} w_{k,t}^{(m_k)}\mathbf{x}_{k,t}^{(m_k)}. \quad (6)$$

6. Resample when necessary to avoid *weight degeneracy* [22].

Note that not every filter needs to exchange information. This depends on the separability in the state and/or the observation equations, as we will elaborate in Section IV-B.

IV. Proposed Partitioning Schemes

Traditional MPFs fix the partition sizes before the start of the filter, and fix the chosen states within the partitions throughout the entire trajectory. In the following two subsections, we look at two ways of partitioning the state-space for improved filtering.

A. Random Partitioning

This scheme remains with fixed partition sizes as the original MPF, however, there are two key differences: i) the partitions are of equal size (with potentially only one partition of smaller size, depending on the total size of the system), and ii) at each time instant, the states occupying a given partition are chosen uniformly. For example, if the state at time t is $\mathbf{x}_t^\top = [x_{1,t}, x_{2,t}, x_{3,t}, x_{4,t}, x_{5,t}]$, we can obtain 3 partitions $\mathbf{x}_{k,t}$, for $k \in \{1,2,3\}$ of dimensions of $d_k \in \{2,2,1\}$. The dimensions d_k remain fixed, but the states in the partitions will change at each t (see Table I). Note that, because we are propagating the particles to new partitions, we are limited to using the same number of particles M_k for each partition k, i.e. $M_k = \frac{M}{k}$, where M is the total number of particles used for the MPF. Otherwise, the new partitions may be formed of states with different number of particles and a joint generation of new particles may be difficult or impossible. There are ways to alleviate this issue (e.g. replicate particles of some states to match others, and/or use estimates), but we will not consider this in this paper due to lack of space.

The random partitioning scheme was motivated by the idea of reducing the error introduced by the exclusion of

	Partition 1	Partition 2	Partition 3
⋮	⋮	⋮	⋮
Time $t-1$	$\mathrm{x}_{3,t-1}, x_{4,t-1}$	$\mathrm{x}_{5,t-1}, x_{1,t-1}$	$\mathrm{x}_{2,t-1}$
Time t	$\mathrm{x}_{2,t}, x_{3,t}$	$\mathrm{x}_{5,t}, x_{4,t}$	$\mathrm{x}_{1,t}$
Time $t+1$	$\mathrm{x}_{5,t+1}, x_{4,t+1}$	$\mathrm{x}_{2,t+1}, x_{3,t+1}$	$\mathrm{x}_{1,t+1}$
⋮	⋮	⋮	⋮

TABLE I: Example of random partitioning of a state with dimension $d_x = 5$.

states when partitioning. When separating two given states in different partitions and fixing this separation throughout the filtering, the information due to the coupling between the states is lost. By permuting the states in the partitions, every state "gets the chance" to be coupled with other states throughout a complete trajectory.

B. Topologically Weighted Random Partitioning

The topology based partitioning scheme was designed to make use of the connectedness between the states in time. We considered the separability in the state equation, i.e., we partitioned the states based on their contributions (and the strengths of their contributions) to a given state in the next time instant. To illustrate this separability, we give a short example with $d_x = 4$. Consider the state equation

$$\begin{aligned} x_{1,t} &= c_{11}x_{1,t-1} + c_{13}x_{3,t-1} + c_{14}x_{4,t-1} + u_{1t} \\ x_{2,t} &= c_{21}x_{1,t-1} + c_{22}x_{2,t-1} + u_{2t}, \\ x_{3,t} &= c_{32}x_{2,t-1} + c_{33}x_{3,t-1} + c_{34}x_{4,t-1} + u_{3t}, \\ x_{4,t} &= c_{41}x_{2,t-1} + c_{44}x_{4,t-1} + u_{4t}, \end{aligned} \tag{7}$$

where c_{ij}, for $i \in \{1, \ldots, 4\}$, and for $j \in \{1, \ldots, 4\}$ are (in this case) known parameters signifying the contribution of the state $x_{j,t-1}$ to the state $x_{i,t}$. Let us consider the first two states $x_{1,t}$ and $x_{2,t}$. Notice that c_{12}, c_{23}, and c_{24} are equal to 0. In that sense, we can say that the state $x_{1,t}$ is separable from $x_{2,t-1}$, and the state $x_{2,t}$ from $x_{3,t-1}$, and $x_{4,t-1}$. A similar statement can be made for all states $x_{i,t}$. In practice, we would be dealing with much higher dimensions where the effect of these connections is more pronounced. The topology weighted random partitioning scheme can be summarized in the following steps:

1. Initialize the set of state indices $S_j \in \{1, \ldots, d_x\}$ with number of elements $d_j = d_x$.

2. Use multinomial sampling to choose a random state index N_i with $p(N_i = i) = \frac{1}{d_j}$.

3. Compute, what we term, the weights of connectedness of the states in the ith row as

$$\omega_i^{(j)} = \frac{|c_{ij}|}{\sum_{\ell \in S_j} |c_{i\ell}|}, \quad for \quad j \in S_j \tag{8}$$

We take the absolute value of the coefficients as we are only interested in the strength of the connection.

4. Sample d_k indices from the set S_j without replacement based on the computed weights, i.e., $P(N_j = j) = \omega_i^{(j)}$. If $d_k \leq d_j$, then take all the rest of the indices.

5. Form a partition of the states whose indices were obtained in step 4, including state i obtained in step 2.

6. Update S_j and its dimension d_j by removing the indices used to form a partition in steps 4-5.

State dimension	Partition size	Total number of particles
$d_x = 100$	$d_k \in \{5, 10, 25\}$	$M \in \{500, 1500, 2500, 4000, 5000\}$
$d_x = 300$	$d_k \in \{10, 30, 60\}$	$M \in \{10000, 25000, 40000, 65000, 90000\}$

TABLE II: Conditions under which the MPFs were run.

7. Repeat steps $2 - 6$ until all the rest of the states have been placed in partitions.

Note that if a given state $\tilde{j}$ has been placed in a partition in a previous iteration, its parameter $c_{i\tilde{j}}$ will not be used in the computation of the weights in step 3 in the following iterations.

V. Simulations

We ran simulations on synthetic data to obtain some initial results using the proposed as well as the original partitioning schemes. For notational convenience, we abbreviate the MPF which uses random partitioning as RP-MPF, and the MPF which uses topologically weighted random partitioning as TWRP-MPF. We consider the following SSM:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{C}g(\mathbf{x}_{t-1}) + \mathbf{u}_t, \\ \mathbf{y}_t &= \mathbf{H}\mathbf{x}_t + \mathbf{v}_t, \end{aligned} \tag{9}$$

where $\mathbf{C} \in \mathbb{R}^{d_x \times d_x}$ is the coefficient matrix whose elements c_{ij} were generated as $c_{ij} \sim \mathcal{U}(-2, 2), \forall i, j$. Additionally, approximately about $\frac{1}{3}$ of them were made 0s to ensure some degree of separability. The observation matrix $\mathbf{H} \in \mathbb{R}^{d_x \times d_x}$ was generated similarly to $\mathbf{C}$, with elements $h_{ij} \sim \mathcal{U}(-1, 1), \forall i \neq j$, and setting the diagonal elements to 1. The function $g(\cdot)$ can be nonlinear and is assumed known. In this case, we used the sigmoid function i.e.,

$$g(s) = \frac{1}{1 + e^{-s}}.$$

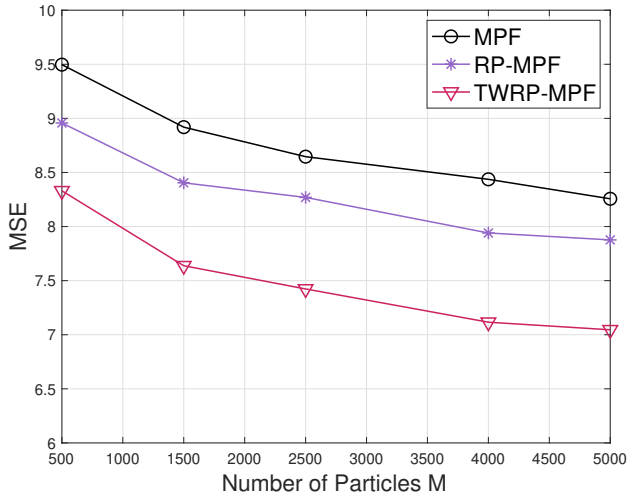
The noise parameters were taken to be Gaussian as $\mathbf{u}_t \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \sigma_x^2 \mathbb{I})$, and $\mathbf{v}_t \overset{iid}{\sim} \mathcal{N}(\mathbf{0}, \sigma_y^2 \mathbb{I})$, with $\sigma_x^2 = 0.1$ and $\sigma_y^2 = 0.1$.

The three MPFs were applied on two systems, one of dimension $d_x = 100$, and the other of $d_x = 300$. For each case, we tested the performance of the methods with different partition sizes, and using a range of particles. The details of the conditions for the simulations can be seen in Table II. The performance of the MPFs was evaluated using the mean squared error (MSE) as a metric, defined as

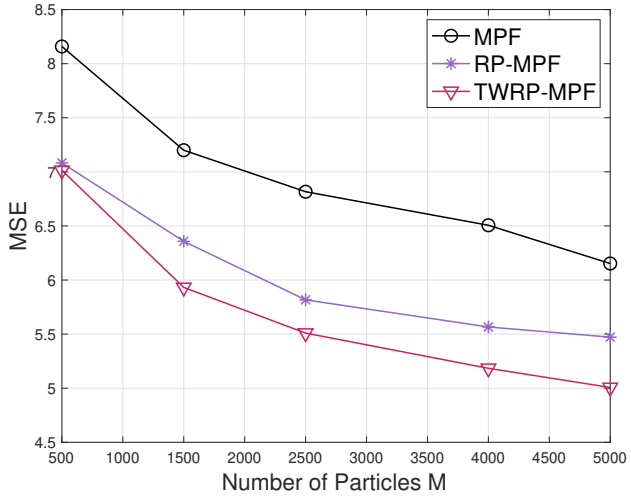
$$MSE(\hat{\mathbf{x}}) = \frac{1}{d_x}\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{d_x}(\hat{x}_{j,t} - x_{j,t})^2 \tag{10}$$

and the results were averaged over $R = 500$ independent runs for each system.

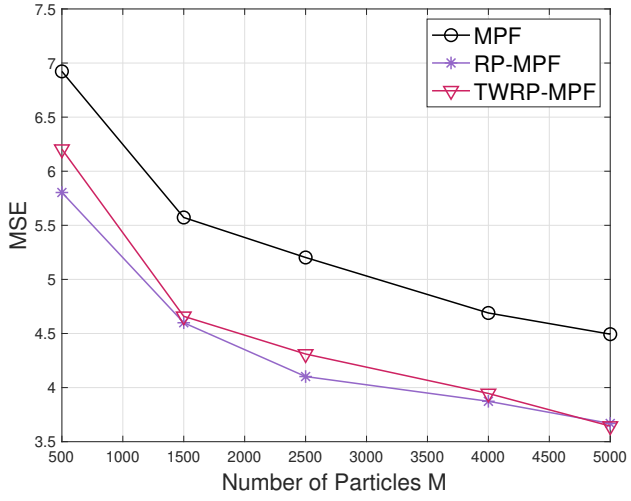
At first glance of Fig. 1, the most noticeable difference is that all filters perform significantly better with smaller partition sizes. This is because a higher number of filters are employed in tracking, and each filter handles lower dimensional states. Such results have been demonstrated in previous works [10], [16]. However, in one run, the partition sizes were kept same for all three filters, thus the difference in their performance is not due to partition size. The RP-MPF makes it evident (see Fig. 1) that the filtering benefits from at least some kind



(a) $d_x = 100$, $d_k = 25$.



(b) $d_x = 100$, $d_k = 10$.



(c) $d_x = 100$, $d_k = 5$.

Fig. 1: Performance of MPFs on a system of dimension $d_x = 100$. The results are averaged over $R = 500$ independent runs.

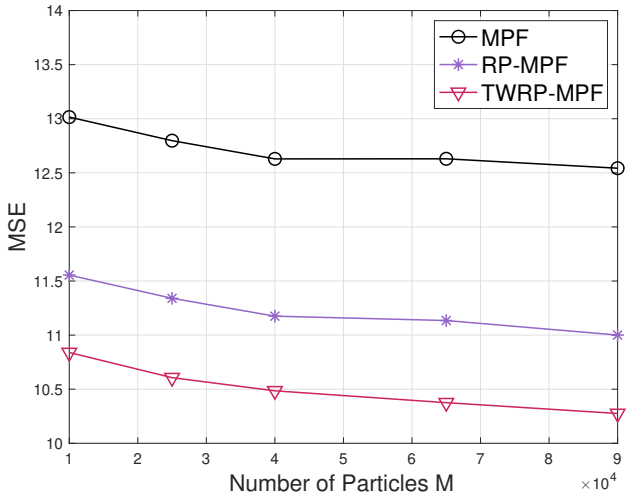


Fig. 2: Performance of MPFs on a system of dimension $d_x = 300$ with partition size $d_k = 60$. The results are averaged over $R = 500$ independent runs.

of permutation of the states in partitions. The TWRP-MPF can be seen as a more principled version of the RP-MPF, where the random partitioning scheme is informed by the connectedness between the states. We can see in Fig. 1c for $d_k = 5$ that there is little to no difference in performance between the RP-MPF and the TWRP-MPF, however, that is not the case for the larger partition sizes in Fig. 1a - 1b. This is thought to be due to the fact that $d_k = 5$ is too small of a partition size to allow for the effect of the topology to manifest, and the net effect of the topology-based partitioning reduces to that of just random partitioning. While having smaller partition sizes is computationally less expensive, it can limit the opportunity of state coupling, which in some cases is more relevant for physical interpretation. Even in that case, a less obvious remark worth noting is that the MPF requires more than three times the number of particles to achieve the same performance as the newly proposed filters. In fact, in the simulations with $d_k = 25$ in Fig. 1a, the original MPF with $M = 5000$ is outperformed by the TWRP-MPF with $M = 500$, and the RP-MPF with $M = 1500$. Similar behavior was observed in the case of $d_x = 300$. We feature only the result of $d_k = 60$ in Fig. 2 to preserve space. The difference in performance is more pronounced as error accumulates with the increase of state dimensions, thus, such partitioning can be more useful for very high dimensions.

VI. Conclusion

In this paper, we proposed novel state-space partitioning schemes for improved accuracy of multiple particle filtering. In particular, we looked at the effects of partitioning the states purely randomly, and based on the topology of the system. The simulations show that a multiple particle filter benefits from permutations in state partitioning, and particularly extracts more information when the states are randomly partitioned when influenced by the topology. This, of course, works where

we assume the topology is known, which is not always the case. Interesting future work should address finding ways to jointly learn the topology while tracking states. Further, reassignment of states every few time steps may be of use to cut down computational cost. However, some topologies are time-varying and an extensive study of such systems with the proposed or improved partitioning schemes can be an interesting line of research.

REFERENCES

[1] G. Karlebach and R. Shamir, "Modelling and analysis of gene regulatory networks," *Nature reviews Molecular cell biology*, vol. 9, no. 10, pp. 770–780, 2008.

[2] X. Sheng and Y. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE transactions on signal processing*, vol. 53, no. 1, pp. 44–53, 2004.

[3] M. F Bugallo, T. Lu, and P. M Djuric, "Target tracking by multiple particle filtering," in *2007 IEEE aerospace conference*. IEEE, 2007, pp. 1–7.

[4] P. J Hansen and P. C Jurs, "Chemical applications of graph theory. part i. fundamentals and topological indices," *Journal of Chemical Education*, vol. 65, no. 7, pp. 574, 1988.

[5] Rudolph E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[6] K. Fujii, "Extended kalman filter," *Reference Manual*, pp. 14–22, 2013.

[7] S. J Julier and J. K Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal processing, sensor fusion, and target recognition VI*. International Society for Optics and Photonics, 1997, vol. 3068, pp. 182–193.

[8] E. A Wan, R. Van Der Merwe, and S. Haykin, "The unscented kalman filter," *Kalman filtering and neural networks*, vol. 5, no. 2007, pp. 221–280, 2001.

[9] T. Bengtsson, P. Bickel, B. Li, et al., "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," in *Probability and statistics: Essays in honor of David A. Freedman*, pp. 316–334. Institute of Mathematical Statistics, 2008.

[10] P. M Djuric, T. Lu, and M. F Bugallo, "Multiple particle filtering," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*. IEEE, 2007, vol. 3, pp. III–1181.

[11] B. Ait-El-Fquih and I. Hoteit, "An efficient multiple particle filter based on the variational bayesian approach," in *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2015, pp. 252–257.

[12] Ç. Taşdemir, M. F Bugallo, and P. M Djurić, "A particle-based´ approach for topology estimation of gene networks," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2017, pp. 1–5.

[13] P. Renzulli, R. Restaino, and P. Addesso, "A computationally efficient approach to wlan localization based on multiple filters," in *2015 International Conference on Localization and GNSS (ICL-GNSS)*, 2015, pp. 1–6.

[14] S. Schlupkothen and G. Ascheid, "Multiple particle filtering for tracking wireless agents via monte carlo likelihood approximation," *EURASIP Journal on Advances in Signal Processing*, vol. 2019, no. 1, pp. 1–20, 2019.

[15] P. M Djurić and M. F Bugallo, "Multiple particle filtering with improved efficiency and performance," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4110–4114.

[16] P. Closas and M. F Bugallo, "Improving accuracy by iterated multiple particle filtering," *IEEE Signal Processing Letters*, vol. 19, no. 8, pp. 531–534, 2012.

[17] B. Ait-El-Fquih and I. Hoteit, "A variational bayesian multiple particle filtering scheme for large-dimensional systems," *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5409–5422, 2016.

[18] L. Úbeda-Medina, Ángel F García-F., and J. Grajal, "Sigma-point multiple particle filtering," *Signal Processing*, vol. 160, pp. 271–283, 2019.

[19] M. F Bugallo and P. M Djuric, "Complex systems and particle filtering," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*. IEEE, 2008, pp. 1183–1187.

[20] S Pérez-Vieites, J Vilà-Valls, M F Bugallo, J Míguez, and P Closas, "Second order subspace statistics for adaptive state-space partitioning in multiple particle filtering," in *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2019, pp. 609–613.

[21] M. F Bugallo and P. M Djurić, "Target tracking by symbiotic particle filtering," in *2010 IEEE Aerospace Conference*. IEEE, 2010, pp. 1–7.

[22] S. Särkkä, *Bayesian filtering and smoothing*, vol. 3, Cambridge University Press, 2013.