Local particle filters: a unifying framework and novel parallelizable extensions

Andrea Quintanilla
School of Mathematics
University of Edinburgh
Edinburgh, UK
A.Quintanilla-Carranza@sms.ed.ac.uk

Víctor Elvira
School of Mathematics
University of Edinburgh
Edinburgh, UK
victor.elvira@ed.ac.uk

***Abstract*—Local particle filters (LPFs) mitigate weight degeneracy in high-dimensional particle filtering. They exploit the fact that, over short time intervals, certain regions of the state space evolve approximately independently, and not all observations inform every state dimension. In standard particle filters, particles deemed globally uninformative may be discarded even when they contain locally informative components.**

This paper makes two key contributions. First, we introduce a unifying framework that encompasses and categorizes a broad class of LPFs, explicitly stating the assumptions and approximations required to run them. Second, building on this framework, we propose a new class of LPFs that enhance parallelization by automatically detecting blocks in the latent space.

Numerical experiments on Gaussian-linear and nonlinear state-space models demonstrate that our approach improves both effective sample size and mean squared error compared to standard particle filters, particularly when the true model exhibits weak dependencies across state-space regions.

***Index Terms*—Particle filtering, high-dimensional state-space models, local particle filters, parallelisation.**

I. Introduction

State-space models (SSMs) are probabilistic models that allow to target challenging problems in science and engineering [1]. These models are composed by latent variables that are inferred through noisy observations and the so-called filtering distributions. However, these distributions are generally intractable and approximations need to be made. Particle filters (PF), a family of Monte Carlo methods that approximate filtering distributions by a set of weighted samples. PFs often suffer from *weight degeneracy*, where a single particle dominates the weight distribution, degrading the quality of the approximation [2], [3]. This phenomenon is especially a frequent challenge in high-dimensional spaces (in latent or observation space) [4], [5]. Local particle filters (LPFs) address this issue by modifying the PF to exploit local information, in lower dimensional subspaces, rather than global computations [6]. This localized approach not only mitigates weight degeneracy but also enables parallelization, improving computational efficiency. A complementary line of work interprets the transition distribution as a graph over the latent state, leveraging the sparsity typical of real-world systems composed of small interacting subsystems [7]–[9]. Although several methods aim to recover sparse transition matrices [8]–[12], which enhances robustness, efficiency, and interpretability, do not explicitly promote clustered structures.

In this paper, we present a unifying framework for LPFs and a new subfamily of them. First, we classify the methods according to how they modify the steps of the standard PF to take advantage of only local information. Next, we classify them according to whether some needed distributions in the LPF, that only have local information are assumed to be known or approximated.

We propose a novel class of methods to identify subsets of dimensions in the latent space -referred to as blocks- such that interactions between different blocks are weak in terms of the system dynamics. Within each block, we exploit this structural property through automated partitioning to exclude irrelevant information from other blocks and allow fully parallelized approach. Numerical experiments demonstrate that the proposed method outperforms standard particle filters, in models that have weak interactions between blocks. The underlying intuition is that, in high-dimensional state-space models, the error introduced by using a less accurate local approximation can be offset by operating within lower-dimensional blocks.

The paper is organized as follows. Section II formalizes the filtering problem. Section III introduces a comprehensive LPF framework, while Section IV details our proposed methods. The numerical experiments and conclusions are presented in Sections V and VI, respectively.

II. Problem statement and background

A. Problem statement

Consider a state-space model (SSM) defined by:

$$x_0 \sim p(x_0), \tag{1}$$

$$x_t | x_{t-1} \sim p(x_t | x_{t-1}), \tag{2}$$

$$y_t | x_t \sim p(y_t | x_t), \tag{3}$$

where $x_t \in \mathbb{R}^{d_x}$ is the state vector and $y_t \in \mathbb{R}^{d_y}$ is the observation vector at time t. All model parameters are assumed to be known. Bayesian filtering aims to compute the posterior distribution $p(x_t | y_{1:t})$, which is typically approximated.

The work of A. Q. is supported by the Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT). The work of V. E. is supported by ARL/ARO under W911NF-22-1-0235 and by the Advanced Research + Invention Agency (ARIA) under PROTECT project.

B. Particle filtering

Particle filters [13] approximate the filtering distribution as

$$p(x_t|y_{1:t}) \approx \sum_{m=1}^{M} w_t^{(m)} \delta_{x_t^{(m)}}(x_t), \tag{4}$$

where M is the number of particles, $x_t^{(m)}$ the m-th particle, $w_t^{(m)}$ its normalised weight and $\delta_{x_t^{(m)}}$ is the Dirac delta centred at $x_t^{(m)}$. The algorithm proceeds recursively: at each step, particles are first drawn from a proposal distribution $q(x_t|x_{t-1}^{(m)})$, then weighted using the ratio of target and proposal densities, and finally resampled according to these weights to reduce degeneracy. A common proposal choice is the transition distribution $p(x_t|x_{t-1}^{(m)})$, which yields the bootstrap particle filter and is used throughout this paper for simplicity.

C. Local particle filters

In this work, we refer to LPFs as a generic approach in which the latent space, $\mathcal{X} \equiv \mathbb{R}^{d_x}$, is partitioned into K disjoint blocks: $\mathcal{X} = [\mathcal{X}_1, \ldots, \mathcal{X}_K]$. These blocks are a key element of LPFs [6], [14], [15], as they will allow some degree of parallelization of the filtering task into spaces of lower dimensions, as we review in the next section. Let $x_{t,k}^{(m)}$ denote the particle m in block k at time t. We will use the index $-k$ to refer to the remaining blocks, e.g. $x_{t,-k} = [x_{t,1}, \ldots, x_{t,k-1}, x_{t,k+1}, \ldots, x_{t,K}]$.

III. A FRAMEWORK FOR LOCAL PARTICLE FILTERS

In this section, we propose a framework that encompasses the most relevant families of LPFs. Each LPF variant adapts the propagation and weighting steps differently, and we classify them based on whether the marginals are known exactly or approximated. We will see that the approximations are usually based on summaries of particles from the rest of the blocks, which adds computational complexity. We now discuss these strategies and partitioning methods.

A. Propagation step

The propagation step in bootstrap PF simulates M samples at each time t, according to the transition distribution $x_t^{(m)} \sim p(x_t|x_{t-1}^{(m)})$. In LPFs, propagation through a proposal is also used, but within each block, through different ways that we describe next.

a) Exact local approach: propagation can be performed by block, as in [16], by using a *local proposal*, which is a proposal of the form $p(x_{t,k}|\ x_{t-1,k})$. In this way, each subspace evolves independently. In order to use the bootstrap PF proposal and compute the weights easily, the LPFs that follow this local approach are applied only in SSMs where the subspaces evolve independently, that is, where

$$p(x_{t,k}|x_{t-1}) = p(x_{t,k}|x_{t-1,k}). \tag{5}$$

This local propagation only uses information from the same block, as the samples are drawn from $p(x_{t,k}|x_{t-1,k}^{(m)})$.

b) Approximated local approach: If block propagation is desired but the transition distribution does not factorize, an approximated local transition distribution can be used instead. An example of this approach is given in [14], where the local transition is approximated by

$$p(x_{t,k}|x_{t-1,k}) \approx p(x_{t,k}|x_{t-1,k}, \widehat{x}_{t-1,-k}), \tag{6}$$

where $\widehat{x}_{t-1,-k}$ is the particle approximation of the mean of the filtering distribution in other blocks at $t-1$.

B. Weighting

In PF, the weights correct for the discrepancy between the proposal distribution and the target posterior. Each particle receives a single weight in standard PFs. LPFs address weight collapse by assigning multiple weights per particle, one for each block, allowing the weight to vary across dimensions. This allows a particle to remain influential where it matches the target distribution and to have less impact where it does not. We now classify the local weighting schemes.

a) Exact local approach: Many local approaches [15], [17] assume that the likelihood factorizes across blocks:

$$p(y_t \mid x_t) = \prod_{k=1}^{K} p(y_{t,k} \mid x_{t,k}). \tag{7}$$

This assumption ensures that the local likelihood, $p(y_{t,k}|x_{t,k})$, is known and used to compute the joint weights:

$$w_{1:t,k} = p\left(y_{t,k}|x_{t,k}\right) w_{1:t-1,k}. \tag{8}$$

b) Approximated local approach: In other approaches, such as in [18], the assumption (7) is dropped, allowing greater flexibility in modelling dependencies. However, it requires estimation methods to compute *approximated local* weights. A concrete example is given in the multiple particle filter framework (multiple PF) [14], [19]. It assumes the factorization of the transition

$$p(x_t|x_{t-1}) = \prod_{k=1}^{K} p(x_{t,k}|x_{t-1,k}). \tag{9}$$

Next, a first-order approximation of $p(y_t|x_{t,k})$ is made as

$$p(y_t|x_{t,k}) \approx p\left(y_t|x_{t,k}, \widetilde{x}_{t,-k}\right), \tag{10}$$

where $\widetilde{x}_{t,-k}$ is equal to the prediction of PF, in the rest of the subspaces. Finally, using the bootstrap proposal, as in $p(x_{t,k}|x_{t-1,k})$, [16], results in the following local weights

$$w_{1:t,k} \approx p\left(y_t|x_{t,k}, \widetilde{x}_{t,-k}\right) w_{1:t-1,k}.$$

Observe that performance of the LPF depends on the first-order estimation (10). Second-order approximations are proposed in [20], [21].

C. Partitioning step

LPFs typically require a state-space partition as input. Many approaches assume spatial structure, for instance when latent space and observations are related to physical locations, and a distance is defined in these locations [6]. In contrast, we will be interested in adaptive partitioning methods when components are not necessarily tied to physical locations. In [22], a method for partitioning SSMs is based on the structure of the transition distribution. Other approaches with adaptive partitioning are presented in [23], which employ multiple partitions with parallel local particle filter instances, and in [24], which construct partitions based on covariance between dimensions. An approach based on Granger causality is introduced in [25].

IV. Proposed method

We propose a parallelizable, *non-interacting* approach, where local propagation steps run independently across blocks without information exchange. The key idea is to design partitions such that the loss of cross-block information has less impact on estimation accuracy than the degradation caused by the curse of dimensionality in global methods. This is particularly effective when inter-block dependencies are weak, allowing fully parallel execution with accuracy comparable to more communicative local methods.

We assume a separable likelihood and focus on the case where the transition is not. As seen, LPFs typically (1) presuppose a partition of the state space and (2) construct local approximations of either the transition or the likelihood by sharing information across local filters. In contrast, we propose a fully parallel LPF. We first partition the state into blocks that are approximately conditionally independent:

$$p(x_{t,k} \mid x_{t-1,k}, x_{t-1,-k}) \approx p(x_{t,k} \mid x_{t-1,k}). \tag{11}$$

Independent particle filters are then run in each block. Under (11), sampling from $p(x_{t,k} \mid x_{t-1,k})$ is achieved by drawing from the global distribution, $x_t^m \sim p(x_t \mid x_{t-1}^m)$, decomposing x_t^m into $(x_{t,k}^m, x_{t,-k}^m)$, and discarding $x_{t,-k}^m$. Moreover, if (11) holds, then for any $x_{t-1,-k}^{(m)} \neq x'_{t-1,-k}$:

$$p(x_{t,k} \mid x_{t-1,k}^{(m)}, x_{t-1,-k}^{(m)}) \approx$$
$$p(x_{t,k} \mid x_{t-1,k}^{(m)}, {x'}_{t-1,-k}).$$

Thus, propagation in block k may fix the remaining entries to a constant, $x'_{t-1,-k} = c$. In SSMs with polynomial dynamics and Gaussian noise (as in our experiments), we set $c = 0$, which removes contributions from $x_{t-1,-k}^{(m)}$ to $x_{t,k}^{(m)}$ during propagation.

The method is summarised in Algorithm 1. After partitioning the state space (line 1), initial particles are drawn from the prior (line 2). For each time step and block k, particles are extended to global space by padding other dimensions with zeros (lines 5–6). These extended particles are propagated with the global model (line 7), then dimensions in $-k$ are dropped, to get a particle of the block k (line 8). Local weights are computed (line 9) and used to resample particles (line 10). As shown in Section V, partition choice strongly influences performance and should reflect model dynamics. The algorithm 2 introduces a static partitioning strategy inspired by [24], adapted to our setting. Unlike [24], which updates the partition at each time step based on the particles, we apply the partitioning once based on the mean of the estimations from a bootstrap PF on independent training data.

Algorithm 1 Parallel LPF

```
 1: Obtain a partition $\{B_k\}_{k=1}^K$ using Algorithm 2.
 2: Sample initial particles $\widetilde{x}_0^{(m)} \sim p(x_0)$ for $m = 1, \dots, M$.
 3: for $t = 1, 2, \dots$ do
 4:     for $k = 1, \dots, K$ in parallel do
 5:         Extend local particles to global space
 6:         $\tilde{x}_{t-1}^{(m)} = (\tilde{x}_{t-1,k}^{(m)}, \tilde{x}_{t-1,-k}^{(m)})$, with $\tilde{x}_{t-1,-k}^{(m)} = 0$.
 7:         Propagate using the global transition $x_t^{(m)} \sim p(x_t | \widetilde{x}_{t-1}^{(m)})$
 8:         Keep $x_{t,k}^{(m)}$
 9:         Compute local weights $w_{t,k}^{(m)} \propto p(y_{t,k} | x_{t,k}^{(m)})$ and
10:         normalize by $\sum_{n=1}^M w_{t,k}^{(n)}$.
11:         Resample local particles using the normalised
12:         local weights.
13:     end for
14: end for
```

Algorithm 2 Fixed spectral clustering (F-SC) for latent space partitioning

```
1: Run a bootstrap PF in previous data using the same SSM as in the
   proposed Algorithm 1.
2: Obtain the resulting estimated means at each time step and store them in
   the design matrix $X \in \mathbb{R}^{T,d_x}$.
3: Estimate the correlation matrix $C$, of $X$.
4: Apply spectral clustering to $C$ to obtain the partition $\{B_{t,k}\}_{k=1}^K$.
```

V. Numerical experiments

We compare the bootstrap PF and the proposed parallel LPF using three partition strategies: the proposed F-SC, an oracle partition, and a random partition. The experiments use a linear-Gaussian SSM and a modified stochastic Lorenz 96 model, both designed with weakly interacting blocks. The linear-Gaussian case, a simple benchmark where dependence is detectable via correlation, serves to test the different partitions. The Lorenz 96 model, a non-linear chaotic system representative of the target applications, is used for a more challenging evaluation, where we also compare against other local PFs; block and multiple PF [14], [15]. Performance is measured by the mean squared error (MSE),

$$\mathrm{MSE}(\widehat{x}) = \frac{1}{d_x T} \sum_{t=1}^{T} \sum_{q=1}^{d_x} \left(\hat{x}_{q,t} - x_{q,t}\right)^2,$$

where $\widehat{x}$ is the mean trajectory; $x_{q,t}$ is the Kalman filter mean for the linear model and the true state for Lorenz 96, since a closed form solution is not available. The results, averaged over $S = 25$ runs, report MSE with standard deviations. We also report the mean effective sample size across blocks, $\mathrm{ESS} = \frac{1}{KT} \sum_{t=1}^{T} \sum_{k=1}^{K} \frac{1}{\sum_{m=1}^{N} w_{t,k}^{(m)}}$, and the Adjusted Rand Index (ARI), to measure the similarity between any cluster and the oracle cluster, random partitions tend to ARI equal 0, while oracle partitions have ARI equal 1.

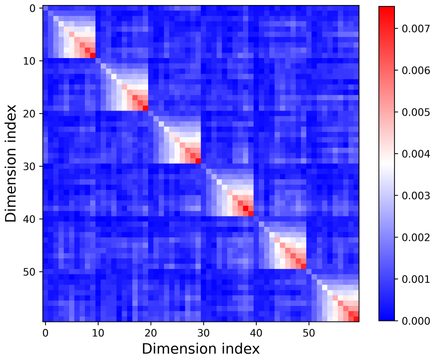

Fig. 1: A visualization of the matrix A with $\alpha = 0.3$ and $\rho = 0.2$.

TABLE I: MSE performance of proposed LPF with different partitions and bootstrap PF, in Gaussian-linear SSM for different settings.

Setting	$\alpha = 0.1$ $\rho = 0.1$	$\alpha = 0.1$ $\rho = 0.3$	$\alpha = 0.3$ $\rho = 0.1$	$\alpha = 0.3$ $\rho = 0.3$
MSE performance				
Bootstrap PF	8.58 (0.62)	9.41 (0.58)	10.85 (1.10)	11.25 (1.14)
Oracle	2.30 (0.07)	3.84 (0.60)	3.35 (0.26)	15.33 (0.42)
F-SC	2.54 (3.18)	2.60 (2.78)	4.03 (3.80)	12.69 (2.23)
Random	79.25 (0.82)	81.51 (1.42)	75.33 (1.06)	72.06 (1.01)
ESS performance				
Bootstrap PF	21.7 (0.5)	19.7 (0.5)	20.4 (0.5)	18.4 (0.4)
Oracle	239.9 (0.2)	241.7 (0.3)	235.7 (0.3)	217.1 (0.3)
F-SC	240.6 (2.7)	242.3 (3.2)	238.6 (2.5)	225.4 (2.8)
Random	336.9 (1.3)	333.2 (1.9)	330.4 (1.7)	299.1 (1.4)

A. Gaussian-linear SSM

We use a linear-Gaussian SSM, with $d_x = d_y = 60$, $K = 6$, $M = 400$, a block-diagonal transition matrix A and block-diagonal noise covariance Q. Weak inter-block interactions are controlled by parameters α and ρ, which set the strength and density of off-block entries (see Figure 1). Observations follow the linear transformation H equal to the block-diagonal version of A helping to regulate observation noise as dimensionality increases; and covariance $R = I$, ensuring factorized likelihood.

Tables I show that the bootstrap PF fails in high dimensions. The local method improves performance, but only with partitions close to the oracle and when α, $\rho < 0.3$. ARI remained high for F-SC, 0.98 (0.06). All local variants achieve significantly higher ESS than bootstrap PF.

B. Modified Lorenz 96 SSM

Lorenz 96 is widely used in the PF literature for high-dimensional inference and climate applications [26], [27]. We modify it to create blocks and control their dependences. The state space is divided into K blocks, and the deterministic dynamics are modified as in Eq. 12, with coefficients $\beta^q_{q'}$ set to 1 within a block and α at boundaries,

$$\frac{\mathrm{d}x_q}{\mathrm{d}t} = \left(\beta^q_{q+1} x_{q+1} - \beta^q_{q-2} x_{q-2}\right) \beta^q_{q-1} x_{q-1} - x_q + F, \tag{12}$$

for $q = 1, \dots, d_x$, with periodic boundary conditions. $\alpha = 0$ indicates disjoint blocks, while $\alpha = 1$ recovers the original, connected model. The SSM is defined by

$$x_{t+1,q} = x_{t,q} + \Delta t \frac{\mathrm{d}x_{t,q}}{\mathrm{d}t} + w_{t+1,q}, \tag{13}$$

$$y_{t+1} = x_{t+1} + v_t, \tag{14}$$

where $w_{t+1,q} \sim \mathcal{N}(0, \Delta t)$, $v_t \sim \mathcal{N}(0, I)$, and integration uses with $\Delta t = 0.05$. We use particles $d_y = d_x = 80$, $K = 2$, $F = 8$, and $M = 600$. Table II shows that all local PFs, including our proposed LPF with a high-quality partition, outperform the bootstrap PF in all scenarios. Our LPF attains its best performance for weakly coupled blocks. The similar performance across all metrics of F-SC and random partitions (the latter omitted for clarity) highlights the challenge of capturing non-linear dependencies for clustering. In this setting, block PF and multiple PF demonstrate greater robustness to partition choice, underscoring the importance of accurate block discovery for our method.

TABLE II: MSE performance of proposed LPF, multiple PF and block PF with different partitions and the bootstrap PF, in Modified Lorenz 96 SSM for different settings.

Setting	$\alpha = 0.0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 1.0$
MSE performance				
Bootstrap PF	8.66 (1.81)	9.31 (1.67)	10.54 (1.43)	10.32 (1.45)
Proposed LPF with different partitions				
Oracle	2.45 (1.05)	4.31 (1.60)	5.96 (1.20)	7.25 (1.53)
F-SC	14.06 (0.99)	15.78 (0.92)	16.17 (0.90)	15.76 (0.92)
Block PF with different partitions				
Oracle	2.66 (1.27)	2.56 (1.37)	3.24 (1.59)	3.22 (1.25)
F-SC	3.70 (1.53)	3.86 (1.93)	4.11 (1.41)	4.55 (1.92)
Multiple PF with different partitions				
Oracle	2.84 (1.32)	2.44 (1.15)	3.45 (1.72)	3.20 (1.55)
F-SC	3.79 (1.72)	4.27 (1.81)	4.40 (1.47)	4.34 (1.23)
ESS performance				
Bootstrap PF	5.81 (0.95)	5.61 (0.79)	5.19 (0.63)	4.98 (0.58)
Proposed LPF with different partitions				
Oracle	30.33 (5.06)	23.04 (4.44)	17.57 (2.45)	14.59 (2.16)
F-SC	8.87 (1.19)	7.98 (0.88)	8.11 (0.76)	8.13 (0.45)
Block PF with different partitions				
Oracle	30.10 (6.00)	29.87 (5.87)	27.18 (6.36)	25.43 (5.11)
F-SC	24.37 (6.77)	23.75 (6.10)	22.43 (4.73)	20.81 (4.97)
Multiple PF with different partitions				
Oracle	29.39 (6.55)	30.49 (5.39)	26.24 (5.58)	26.29 (6.36)
F-SC	25.01 (5.91)	22.96 (4.99)	22.22 (3.96)	21.22 (4.26)

VI. Conclusion

We proposed a unifying framework for LPFs that classifies existing methods, and we also introduced a novel LPF that enables fully parallel propagation through adaptive block partitioning. Experiments on linear–Gaussian models show improvements in ESS and MSE. On modified Lorenz 96 models, the method performs well with oracle partitions. Future work will investigate alternative partitioning strategies for finding partitions in non-linear models.

References

[1] S. Särkkä and L. Svensson, *Bayesian Filtering and Smoothing*, Cambridge University Press, June 2023, Google-Books-ID: utXBEAAAQBAJ.

[2] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to High-Dimensional Particle Filtering," *Monthly Weather Review*, Dec. 2008.

[3] S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart, "Importance Sampling: Intrinsic Dimension and Computational Cost," *Statistical Science*, vol. 32, no. 3, pp. 405–431, 2017, Publisher: Institute of Mathematical Statistics.

[4] P. J. Van Leeuwen, "Nonlinear Data Assimilation for high-dimensional systems: - with geophysical applications -," in *Nonlinear Data Assimilation*, vol. 2, pp. 1–73. Springer International Publishing, Cham, 2015, Series Title: Frontiers in Applied Dynamical Systems: Reviews and Tutorials.

[5] P. J. Van Leeuwen, "Particle filtering in geophysical systems," *Monthly Weather Review*, vol. 137, no. 12, pp. 4089–4114, 2009.

[6] A. Farchi and M. Bocquet, "Comparison of local particle filters and new implementations," *Nonlinear Processes in Geophysics*, vol. 25, no. 4, pp. 765–807, 2018, Publisher: Copernicus Publications Göttingen, Germany.

[7] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[8] V. Elvira and É. Chouzenoux, "Graphical inference in linear-gaussian state-space models," *IEEE Transactions on Signal Processing*, vol. 70, pp. 4757–4771, 2022.

[9] E. Chouzenoux and V. Elvira, "Sparse graphical linear dynamical systems," *Journal of Machine Learning Research*, vol. 25, no. 223, pp. 1–53, 2024.

[10] B. Cox and V. Elvira, "Sparse bayesian estimation of parameters in linear-gaussian state-space models," *IEEE Transactions on Signal Processing*, vol. 71, pp. 1922–1937, 2023.

[11] E. Chouzenoux and V. Elvira, "Graphit: Iterative reweighted l1 algorithm for sparse graph inference in state-space models," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[12] B. Cox, E. Chouzenoux, and V. Elvira, "Graphgrad: Efficient estimation of sparse polynomial representations for general state-space models," *IEEE Transactions on Signal Processing*, vol. 73, pp. 1562–1576, 2025.

[13] P. Djuric, J. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, Sept. 2003.

[14] 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*. 2007, vol. 3, pp. III–1181, IEEE.

[15] P. Rebeschini and R. Van Handel, "Can local particle filters beat the curse of dimensionality?," *Ann. Appl. Probab.*, 2015.

[16] 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)*, Apr. 2015, pp. 4110–4114, ISSN: 2379-190X.

[17] J. Poterjoy, "A localized particle filter for high-dimensional nonlinear systems," *Monthly Weather Review*, vol. 144, no. 1, pp. 59–76, 2016.

[18] 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, Publisher: IEEE.

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

[20] L. Ubeda-Medina, A. F. Garcia-Fernández, and J. Grajal, "Sigma-point multiple particle filtering," *Signal Processing*, vol. 160, pp. 271–283, 2019, Publisher: Elsevier.

[21] J. P. Beaudeau, M. F. Bugallo, and P. M. Djurić, "RSSI-Based Multi-Target Tracking by Cooperative Agents Using Fusion of Cross-Target Information," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5033–5044, Oct. 2015.

[22] M. Iloska and M. F. Bugallo, "State-space partitioning schemes in multiple particle filtering for improved accuracy," in *2022 30th European Signal Processing Conference (EUSIPCO)*. 2022, pp. 2026–2030, IEEE.

[23] R. Min, C. Garnier, F. Septier, and J. Klein, "Parallel block particle filtering," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*. 2021, pp. 86–90, IEEE.

[24] R. Min, C. Garnier, F. Septier, and J. Klein, "State space partitioning based on constrained spectral clustering for block particle filtering," *Signal Processing*, vol. 201, pp. 108727, 2022, Publisher: Elsevier.

[25] A. Quintanilla and V. Elvira, "State-space partitioning via granger causality for local particle filters," in *33rd European Signal Processing Conference (EUSIPCO 2025)*, Palermo, Italy, September 2025.

[26] B. Cox, E. Chouzenoux, and V. Elvira, "GraphGrad: Efficient Estimation of Sparse Polynomial Representations for General State-Space Models," *IEEE Transactions on Signal Processing*, vol. 73, pp. 1562–1576, 2025.

[27] D. S. Wilks, "Effects of stochastic parametrizations in the Lorenz '96 system," *Quarterly Journal of the Royal Meteorological Society*, vol. 131, no. 606, pp. 389–407, 2005, _eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1256/qj.04.03.