Particle Filtering for High-Dimensional Systems

Petar M. Djurić and Mónica F. Bugallo
Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY 11794-2350
Email: {petar.djuric, monica.bugallo}@stonybrook.edu

***Abstract*—Particle filtering methods aim at tracking probability distributions sequentially in time. One of the main challenges of these methods is their accuracy in high-dimensional state spaces. Namely, it can be shown that if the dimensions of these spaces are sufficiently high, the obtained results by particle filtering are practically useless. In this paper, we propose an approach for addressing this problem. It is based on breaking the high-dimensional distribution of the complete state into smaller dimensional (marginalized) distributions and attempting to track these distributions in a novel way as accurately as possible. We demonstrate the proposed approach with computer simulations.**

I. Introduction

Particle filtering (PF) is a methodology that aims at sequential tracking of distributions of interest that arise in state space models [1], [2]. This methodology is particularly popular due to its capacity to handle nonlinearities and non-Gaussian distributions. With PF the distributions are approximated by simulated samples (particles) of the unknown parameters or states and by weights assigned to the particles. The estimated distributions provide more information about the latent states than just point estimates of the states. Once a set of particles and weights are available, the end-user can construct various desired estimates and confidence intervals. Furthermore, one can use the particles and the weights to conduct tests or to choose models.

An important problem of PF is its quick deterioration in performance when the dimension of the state space becomes large. For example, it has been observed that the method when applied to high-dimensional geophysical data can collapse after a very few steps [3]. The poor performance of PF in high-dimensional spaces has also been discussed in [4]. In [5], it was shown that when the particle size is sub-exponential in the cube root of the system dimension, the maximum weight of the particles tends to one.

The root of the problem is easy to explain. Since PF is about intelligent exploration of the space of unknown states by sampling randomly from that space, it is obvious that it becomes increasingly difficult to draw good particles as the dimension of the space grows. We demonstrate it by a very simple example. In Fig. 1 on the left, we present the results of sampling from a uniform distribution on (0,1). Suppose that the set of interest is the interval (0.4, 0.6), which is a priori unknown. If the number of generated samples is M, the expected number of samples in the desired interval is $0.2M$. In the middle figure we see the results of sampling in a two-dimensional space, where now the two random variables x and y are independently sampled from uniform distributions on (0,1). Let now the region of interest be the square on the plot whose sides are equal to 0.2. This time the number of expected samples in the region of interest is $0.04M$. Finally, the plot on the right side shows the outcome of sampling in a three-dimensional space and where the set of interest is the plotted cube with sides equal to 0.2. This time, the expected number of good samples is only $0.008M$. In general, if the dimension of the space is d, the expected number of good samples is $0.2^d M$. Clearly, with increased d, it becomes very difficult to get samples from regions of interest from the state space. In the PF context, the desired region is where the posterior is high, and this posterior often gets very peaky in high-dimensional problems, which is why the propagation of particles to important areas is very challenging and often fails.

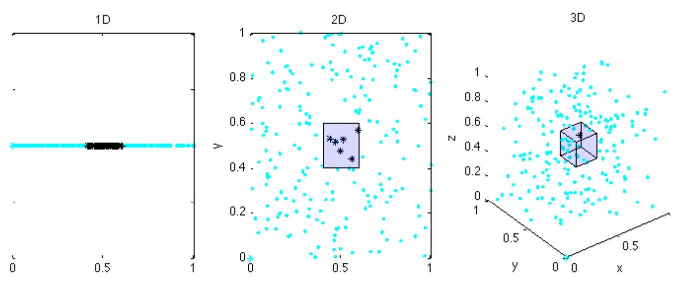

Fig. 1. Sampling in spaces of different dimensions.

High-dimensional problems appear in various areas. In geophysical systems such as the atmosphere or the oceans, the state spaces are large [6] and often include more than one million variables [7]. An approach that attempts to keep the PF away from divergence employs backtrack PF, which is based on going back to the time when the weights of the particles showed low weights and on reprocessing the data [8]. Another approach is based on merged PF where at the measurement times, linear combinations of particles are taken in order to reduce the variance of the particle weights [9]. Computer vision is also an area where high-dimensional state spaces are rampant. There, one direction for tackling the problem is by partitioned sampling [10]. The underlying idea is to apportion the state space and exploit a decomposition of the dynamics

This work was supported by NSF under Awards CCF-0953316 and CCF-1320626.

978-1-4673-3146-3/13/$31.00 ©2013IEEE

of the states. For example, recently in [11], the problem of tracking human activity from video sequences was addressed based on ideas from [10] and using hierarchical particle filters.

In our own previous efforts, we proposed the concept of multiple particle filtering (MPF) in [12] and [13]. There, the state space is partitioned into subspaces and in each subspace the tracking is performed by separate particles filters. The particle filters exchange the estimates of their states with other particle filters, which are used for particle propagation in the subspaces and for computation of the particle weights. It has been reported that the MPF performs very well in a real time setting of device-free tracking [14], [15], in cognitive radar networks [16] and in automated tracking of sources of neural activity [17].

In this paper, we further investigate the concept of MPF and propose a novel class of such filters. Rather than exchanging point estimates or higher order moments, a particle filter uses particles from other particle filters that are necessary to carry out its operations of particle propagation and weight computation. It is important to note that the complexity of the proposed method grows linearly with the number of particle filters in the system. We demonstrate the improved performance of the filter with computer simulations.

II. Problem formulation

Let a dynamical system of interest be represented by the following state space model:

$$x_t = f_x(x_{t-1}, u_t), \qquad \text{state equation} \qquad (1)$$

$$y_t = f_y(x_t, v_t), \qquad \text{observation equation} \qquad (2)$$

where $t = 0, 1, 2, \cdots$ represents time index, $x_t \in \mathbb{R}^{d_x}$ is the latent state of the system at time instant t, $y_t \in \mathbb{R}^{d_y}$ are observations made about the system at time instant t, $f_x(\cdot)$ and $f_y(\cdot)$ are functions that can be nonlinear in their arguments, and $v_t \in \mathbb{R}^{d_v}$ and $u_t \in \mathbb{R}^{d_u}$ are noises in the state and observation equations, respectively. The dimensions of x_t, y_t, u_t, and v_t are d_x, d_y, d_u, and d_v, respectively. The noise distributions of u_t and v_t are parametric and known (where the parameters of the distributions may not be known).

Based on the given model and the observations $y_{1:t}$, the general objective is to extract complete information about the latent state x_t. More specifically, the goal is to estimate $p(x_t|y_{1:t})$, which is the filtering distribution of x_t. This has to be done sequentially where $p(x_t|y_{1:t})$ is computed from $p(x_{t-1}|y_{1:t-1})$. Related goals are to find the joint distribution $p(x_{0:t}|y_{1:t})$, or the predictive distributions $p(x_{t+\tau}|y_{1:t})$ and $p(y_{t+\tau}|y_{1:t})$, $\tau > 0$, or the smoothing distribution $p(x_t|y_{1:T})$, where $t < T$.

The emphasis here is on solutions of these problems when d_x is large. In this paper, we will discuss the case of filtering distribution. The proposed solution can be used similarly for estimating the predictive and smoothing distributions.

III. Particle filtering in high-dimensional spaces

We propose that we decompose the state space into subspaces of small dimensions and run one particle filter on each subspace and thereby have the particle filters operate in lower dimensional spaces. In mathematical terms, the space of x_t, Ω_{x_t}, is partitioned into n subspaces, $\Omega_{x_{i,t}}$, $i = 1, 2, \cdots, n$, where $\cup_{i=1}^{n} \Omega_{x_{i,t}} = \Omega_{x_t}$ and $\Omega_{x_{i,t}} \cap \Omega_{x_{j,t}} = \emptyset$, for $i \neq j$, $i, j = 1, 2, \cdots, n$. We intend to estimate sequentially the marginal filtering distributions, $p(x_{i,t}|y_{1:t})$, $i = 1, 2, \cdots, n$. Thus, with this approach we give up the goal of getting the full filtering distribution of the states and instead settle for tracking a set of marginalized filtering distributions.

At each time instant, the procedure is composed of several steps. They include (a) exchanges of a subset of particles from the most recent *filtering* random measures among relevant particle filters, (b) propagation of particles, (c) exchanges of particles from the current *predictive* random measures among relevant particle filters, (d) weight computation, and (e) resampling. A pictorial diagram with two particle filters is shown in Fig. 2. In this scheme, a particle filter has a full access to all the particles and weights of the particle filters needed for its particle propagation and weight computations.

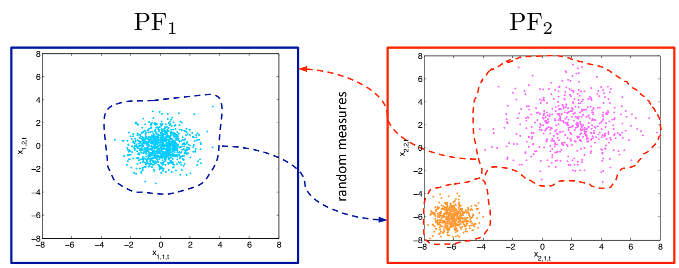

Fig. 2. The new multiple particle filter.

There are two crucial steps in the implementation of the method. The first is the propagation of the particles for the next time instant and the second is the computation of the weights of the propagated particles. Next we describe them in detail.

Suppose that at time $t-1$ the filtering random measures of each of the particle filters are given by

$$\chi_{i,t-1} = \{x_{i,t-1}^{(m)}, w_{i,t-1}^{(m)}\}_{m=1}^{M}, \quad i = 1, 2, \cdots, n,$$

where the $x_{i,t-1}^{(m)}$s are the particles and $w_{i,t-1}^{(m)}$s are the weights of the ith filter, and M is the total number of particles. We allow that each of the particle filters makes its measure available to the particle filters that need it. Let the ith particle filter for propagation of its particles use an instrumental function $\pi(x_{i,t}|x_{i,t-1}^{(m)}, y_{1:t})$. One choice of this function is the marginal predictive distribution,

$$\begin{aligned} &p(x_{i,t}|x_{i,t-1}^{(m)}, y_{1:t}) \\ &= \int p(x_{i,t}|x_{i,t-1}^{(m)}, x_{1,t-1}, \cdots, x_{i-1,t-1}, x_{i+1,t-1}, \cdots, x_{n,t-1}) \\ &\times p(x_{1,t-1}, \cdots, x_{n,t-1}|y_{1:t-1}) \mathrm{d}x_{1,t-1} \cdots \mathrm{d}x_{n,t-1}, \end{aligned} \qquad (3)$$

which means that the ith particle filter has to integrate out all its nuisance states (the ones that it is not tracking but are needed for propagation). Note that in (3) we formally assume that for the propagation of $x_{i,t-1}^{(m)}$, we need particles of all the other states. In the following, we use the approximation

$$\begin{aligned} &p(x_{1,t-1}, \cdots, x_{i-1,t-1}, x_{i+1,t-1}, \cdots, x_{n,t-1}|y_{1:t-1}) \\ &\approx \prod_{k=1,k\neq i}^{n} p(x_{k,t-1}|y_{1:t-1}). \end{aligned} \tag{4}$$

There are several ways to proceed. Here we present one.

For each sampled $x_{i,t-1}^{(m)}$ for propagation, we generate J particles $x_{i,t}^{(m,j)}$, $j = 1, 2, \cdots, J$. We do this by first randomly drawing the indices of the particles from the other state spaces according to their respective weights, and then using them for regular particle propagation. Since each $x_{i,t-1}^{(m)}$ has J children, we end up with $M \times J$ particles. The obtained particles and weights (all equal) represent the predictive random measure.

The next step is the computation of the weights for the generated particles $x_{i,t}^{(m,j)}$. With this step one evaluates the weights $w_{i,t}^{(m,j)}$ by

$$w_{i,t}^{(m,j)} \quad \propto \quad p(y_t|x_{i,t}^{(m,j)}). \tag{5}$$

This expression is due to the fact that the instrumental distribution is the same as the transition distribution. Again, the computation in (5) requires integration because the observations are, in general, functions of states that are tracked by other particle filters, i.e.,

$$\begin{aligned} &p(y_t|x_{i,t}^{(m,j)}, y_{1:t-1}) \\ &= \int p(y_t|x_{i,t}^{(m,j)}, x_{1,t}, \cdots, x_{i-1,t}, x_{i+1,t}, \cdots, x_{n,t}) \\ &\times p(x_{1,t}, \cdots, x_{i-1,t}, x_{i+1,t}, \cdots, x_{n,t}|y_{1:t-1}) \mathrm{d}x_{1,t} \cdots \mathrm{d}x_{n,t}. \end{aligned} \tag{6}$$

We approximate the evaluation of the integral in (6) by drawing particles L times from the predictive measures of the other filters and evaluating the average likelihood of $x_{i,t}^{(m,j)}$ by

$$w_{i,t}^{(m,j)} \quad \dot{\propto} \quad \frac{1}{L} \sum_{l=1}^{L} p\big(y_t|x_{i,t}^{(m,j)}, x_{1,t}^{(\lambda_{1,l})}, \cdots, x_{n,t}^{(\lambda_{n,l})}\big), \tag{7}$$

where $\lambda_{r,l}$ is the index of the lth particle from the rth random measure.

There is one more step and its objective is to bring the number of MJ particles back to M so that we finally obtain the filtering distributions at time t. This is readily achieved by resampling. In one approach, one draws the M particles by using their weights $w_{i,t}^{(m,j)}$ and in another, one samples one particle from each set of children (there are M such sets). With the former resampling, the particles have equal weights and with the latter, they are different. The surviving particles and their weights are used to form $p^M(x_{i,t}|y_{1:t})$.

IV. Simulation results

In this section we present results for a system whose state space dimension is $d_x = 30$. In particular, we considered a system with the following state equations:

$$\begin{aligned} x_{1,t} &= .8x_{1,t-1} + .2x_{d_x,t-1} + u_{1,t} \\ x_{2,t} &= .8x_{2,t-1} + .2x_{1,t-1} + u_{2,t} \\ &\vdots \\ x_{d_x,t} &= .8x_{d_x,t-1} + .2x_{d_x-1,t-1} + u_{d_x,t}, \end{aligned} \tag{8}$$

where $u_{i,t}, i = 1, \cdots, d_x$ are independent and identically distributed zero-mean Gaussian perturbations with variance $\sigma_{u_i}^2 = 1$. For simplicity, we assumed that there was one observation per state. The observations were highly nonlinear with respect to the states and given by

$$y_{i,t} = e^{\frac{x_{i,t}}{2}} v_{i,t}, \qquad i = 1, \cdots, d_x, \tag{9}$$

with $v_{i,t}$ denoting independent zero-mean Gaussian random variables with variance $\sigma_{v_i}^2 = 1$.

We let the system evolve for $T = 60$ time units and we compared the following algorithms:

- The standard PF (SPF) that generated 600 particles of dimension 30. We denoted this filter as `SPF 1×600×1`, where the first number denotes the number of filters (in this case one), the particles per filter ($M = 600$), and the number of children in the propagation step ($J = 1$).
- The SPF that generated 600×4 particles of dimension 30. This meant that for each of the 600 particles we drew $J = 4$ children and later we resampled the set of 2400 particles back to 600 particles. We denoted this filter as `SPF 1×600×4`.
- The MPF from [12] where we used 30 filters and for each of them generated 20 particles of dimension 1. To deal with the coupling of the states given in (8), the filters exchanged the means of their particles that were used in the corresponding instrumental function. Therefore, all the particles of a given filter were propagated using the same information (mean) from the coupled state. We denoted this filter as `MPF 30×20×1`. We point out that this filter uses the same amount of particles as the first SPF, a total of 600. However, in order to obtain the approximation of the marginal distribution of a particular state, it only uses 20 particles.
- The MPF that used 30 filters and where each of them generated 600 particles of dimension 1. As in the previous MPF, the individual filters again exchanged the means of their particles. We denoted this filter as `MPF 30×600×1`. The rationale for using 600 particles per filter was to compare it to the first SPF, as the latter uses 600 particles for its approximation of the marginal distribution of a given state.
- The new MPF that used 30 filters, each of them operating with 20 particles of dimension 1 but with 4 generated children per particle. At each step, the number of particles per filter was brought back to 20 using resampling. For the propagation of its own particles, a given particle filter used 4 particles from a filter responsible for the needed state for the propagation. The required particles from the other filter were obtained by resampling. Note that this filter does not need to apply (7) because the observation equations are decoupled. We denoted this filter as `New MPF 30×20×4`.

Figure 3 shows the average mean square error (MSE) of all the states calculated from 500 realizations of the system. It is obvious that the traditional PF suffers from the large

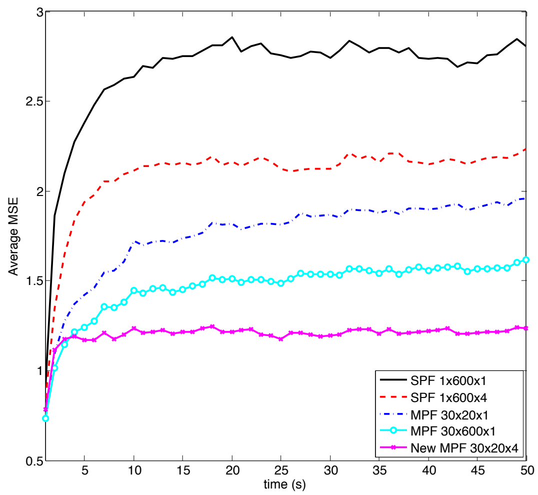

Fig. 3. Average MSE comparison of the different methods.

dimension of the state. Even the improvement obtained by generating 4 children per particle is insufficient to reach the performance of the most basic MPF. We note the large difference in performance between the `SPF 1×600×1` and the `MPF 20×30×1` when the latter only uses 20 particles per filter to approximate the marginals. The comparison clearly shows that the best performance was obtained by the newly proposed MPF. We emphasize that this performance was achieved by only 20 particles per filter.

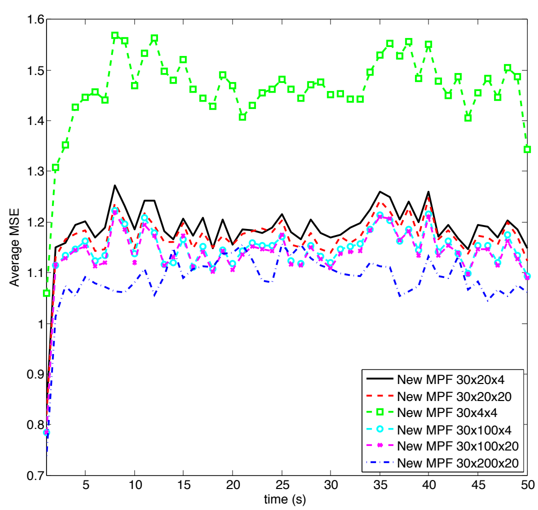

Fig. 4. Average MSE comparison of various implementations of the new MPF.

In Figure 4 we compared different parameter combinations for the new MPF in terms of number of particles per individual filter and number of children generated per particle. The experiment was run 100 times. For the stated 30-dimensional problem, we observe that if we dramatically decrease the number of particles per filter ($M = 4$), there will be a big loss in performance. However, `New MPF 30×20×4` from the previous experiment had almost as good performance as the other new MPFs that used either more particles or more children or both.

V. Conclusion

We have addressed a particle filtering methodology for problems where the state space is of high dimension. We proposed to break the space into subspaces and to perform separate particle filtering in each of the subspaces. The two critical operations of particle filtering, the particle propagation and weight computation of each particle filter are performed wherever necessary with the aid of particles from other subspaces. The proposed method was demonstrated by computer simulations. The obtained results are promising and encourage further study of the method.

References

[1] A. Doucet and A. M. Johansen, "A tutorial on particle filtering and smoothing: Fifteen years later," in *Handbook of Nonlinear Filtering*. Oxford University Press, 2009.

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

[3] J. L. Anderson and S. L. Anderson, "A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts," *Monthly Weather Review*, vol. 127, no. 12, pp. 2741–2758, 1999.

[4] F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *the Proceedings of IEEE Aerospace Conference*, vol. 4, Big Sky, MO, 2003, pp. 1979–1993.

[5] T. Bengtsson, P. Bickel, and B. Li, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," *Probability and Statistics: Essays in honor of David A. Freedman*, vol. 2, pp. 316–334, 2008.

[6] A. J. Majda and J. Harlim, *Filtering Complex Turbulent Systems*. Cambridge University Press, 2012.

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

[8] E. T. Spiller, A. Budhiraja, K. Ide, and C. K. Jones, "Modified particle filter methods for assimilating Lagrangian data into a point-vortex model," *Physica D: Nonlinear Phenomena*, vol. 237, no. 10, pp. 1498–1506, 2008.

[9] S. Nakano, G. Ueno, and T. Higuchi, "Merging particle filter for sequential data assimilation," *Nonlinear Processes in Geophysics*, vol. 14, no. 4, pp. 395–408, 2007.

[10] J. MacCormick and M. Isard, "Partitioned sampling, articulated objects, and interface-quality hand tracking," in *the Proceedings of the European Conference on Computer Vision*, 2000, pp. 3–19.

[11] Z. L. Husz, A. M. Wallace, and P. R. Green, "Tracking with a hierarchical partitioned particle filter and movement modelling," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 41, no. 6, pp. 1571–1584, 2011.

[12] P. M. Djurić, T. Lu, and M. F. Bugallo, "Multiple particle filtering," in *the Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007, pp. III–1181.

[13] M. F. Bugallo, T. Lu, and P. M. Djurić, "Target tracking by multiple particle filtering," in *the Proceedings of the Aerospace Conference*, 2007.

[14] S. Nannuru, Y. Li, M. Coates, and B. Yang, "Multi-target device-free tracking using radio frequency tomography," in *the Proceedings of the Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2011, pp. 508–513.

[15] S. Nannuru, Y. Li, Y. Zeng, M. Coates, and B. Yang, "Radio frequency tomography for passive indoor multi-target tracking," *IEEE Transactions on Mobile Computing*, 2012.

[16] P. Chavali and A. Nehorai, "Scheduling and power allocation in a cognitive radar network for multiple-target tracking," *IEEE Transactions on Signal Processing*, vol. 60, no. 2, pp. 715–729, 2012.

[17] L. Miao, J. J. Zhang, C. Chakrabarti, and A. Papandreou-Suppappola, "Efficient Bayesian tracking of multiple sources of neural activity: Algorithms and real-time FPGA implementation," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 633–647, 2013.