PARTICLE FILTERING IN HIGH-DIMENSIONAL SYSTEMS WITH GAUSSIAN APPROXIMATIONS

Mónica F. Bugallo and Petar M. Djurić

Department of Electrical and Computer Engineering Stony Brook University, Stony Brook, NY 11794 Email: {monica.bugallo,petar.djuric@stonybrook.edu}

ABSTRACT

In this paper we introduce a new multiple particle filtering approach for problems where the state-space of the system is of highdimension. We propose to break the space into subspaces and to perform separate particle filtering in each of them. 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 parametric distributions received from other subspaces. The proposed method is demonstrated by computer simulations and the results show an excellent performance when compared to other implementations of multiple particle filtering.

Index Terms— particle filtering, state-space models, highdimensional systems

1. INTRODUCTION

In many real applications, the systems in question are characterized by high-dimensional dynamical models exhibiting highly nonlinear behavior and with ensuing distributions that are distinctly non-Gaussian [1]. The non-linearity and non-Gaussianity features justify particle filtering (PF) as the methodology of choice for approximation of the posterior distributions of the parameters of interest [2]. The obtained approximations of these distributions consist of simulated samples (particles) drawn by a selected instrumental (importance or proposal) function and of weights assigned to the particles calculated by application of Bayes' rule [3, 4].

The accuracy of PF depends heavily on the choice of the instrumental function. The increase in dimensionality of the state-space creates a formidable challenge to generate particles in regions of non-negligible probability without an explosion in the number of particles that have to be sampled. How quickly the number of needed particles rises depends on the interrelationship of components of the state-space vector [5]. In [6], it is discussed that although PF does not avoid the curse-of-dimensionality in general, a carefully designed particle filter could mitigate it for certain problems. Also, in a review article on convergence of PF methods, it was claimed that under mild conditions, the mean square error (MSE) of the filter is upper-bounded and PF can beat the curse-of-dimensionality as the rate of convergence is independent of the state dimension [7]. Further, it was maintained that in ensuring a given precision measured by the MSE, the number of particles may be a function of the state-space dimension.

Approaches that attempt to keep PF away from divergence include backtracking 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], and merging 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 another area where high-dimensional state-spaces are rampant. There, one approach for tackling the problem is by partitioned sampling [10]. 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 previous work, we proposed the concept of multiple particle filtering (MPF) [12, 13]. The state-space is broken into subspaces and in each subspace filtering is performed by separate units. Therefore, the high-dimensional distribution of the complete state is divided into smaller dimensional (marginalized) distributions and we attempt to track these distributions as accurately as possible. To that end, the particle filters exchange the estimates of their states with other filters, which are used for particle propagation and for computation of the 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 a very recent advance on MPF, a novel approach was proposed where rather than exchanging point estimates, a particle filter uses particles from other particle filters that are necessary for propagation and weight computation of its particles [18]. The complexity of the proposed method grows linearly with the number of filters and the benefits are impressive.

In this paper, we further investigate the concept of MPF and propose an alternative to its implementation. Rather than exchanging point estimates or sets of particles (non parametric approximations), a particle filter uses parametric distributions from other particle filters to carry out its operations of particle propagation and weight computation. It is important to note that the complexity of the proposed method is minimal since only the parameters of the distributions are exchanged among filters. We demonstrate the performance of the new approach with computer simulations.

2. PROBLEM FORMULATION

The dynamical systems of interest are represented by

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

$$y_t = f_y(x_t, v_t),$$
 observation equation (2)

where $t = 1, 2, \cdots$ represents time index, $x_t \in \mathbb{R}^{d_x}$ is the d_x dimensional latent state of the system at time instant $t, y_t \in \mathbb{R}^{d_y}$ are d_y 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 $u_t \in \mathbb{R}^{d_u}$ and $v_t \in \mathbb{R}^{d_v}$ are noises in the state (of dimension d_u)

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

and observation (of dimension d_v) equations, respectively. The distributions of the noise terms u_t and v_t are parametric and known, even though the parameters of the distributions may be unknown. The focus is on systems where d_x is large.

Based on the given model and the observations of the system available at time instant $t, y_{1:t} \equiv \{y_1, y_2, \dots, y_t\}$, the objective is to obtain the complete information about the latent state x_t , which is given by the filtering distribution, $p(x_t|y_{1:t})$. Moreover, the aim is to estimate $p(x_t|y_{1:t})$ in a sequential manner, i.e., by computing $p(x_t|y_{1:t})$ from $p(x_{t-1}|y_{1:t-1})$.

3. MULTIPLE PARTICLE FILTERING WITH GAUSSIAN EXCHANGES OF INFORMATION

In this section, we first briefly summarize the notion of PF and recap its difficulty when dealing with high-dimensional state-spaces. Then, we describe the concept of MPF and a novel *parametric* way of implementing it without incurring explosion in the needed number of exchanged particles among the different filters.

3.1. Particle filtering and the curse-of-dimensionality

We use the PF methodology for estimating the distributions of interest. We recall that with PF we approximate the distributions of interest with random measures composed of particles and their associated weights. The particles represent possible values of the states and their weights are probability masses assigned to the particles. In brief, at time t - 1, the posterior distribution, $p(x_{t-1}|y_{1:t-1})$, is represented by the random measure χ_{t-1} defined by

$$\chi_{t-1} = \left\{ x_{t-1}^{(m)}, w_{t-1}^{(m)} \right\}_{m=1}^{M},$$
(3)

where $x_{t-1}^{(m)}$ and $w_{t-1}^{(m)}$ are the *m*th particle and weight, respectively, and *M* is the total number of particles. Alternatively, we say that the distribution of interest is approximated according to

$$p(x_{t-1}|y_{1:t-1}) \approx p^{M}(x_{t-1}|y_{1:t-1})$$
$$= \sum_{m=1}^{M} w_{t-1}^{(m)} \delta_{x_{t-1}^{(m)}}(\mathrm{d}x_{t-1}), \qquad (4)$$

where $p^M(x_{t-1}|y_{1:t-1})$ is the approximating discrete distribution and $\delta_{x_{t-1}^{(m)}}(dx_{t-1})$ is the unit delta measure concentrated at $x_{t-1}^{(m)}$. With PF we essentially estimate $p(x_t|y_{1:t})$ from the estimate of $p(x_{t-1}|y_{1:t-1})$. More specifically, we first generate particles $x_t^{(m)}$ that will represent the support of $p^M(x_t|y_{1:t})$ by using the particles $x_{t-1}^{(m)}$ of $p^M(x_{t-1}|y_{1:t-1})$ and then compute the weights $w_t^{(m)}$ of $x_t^{(m)}$, thereby completing the description of $p^M(x_t|y_{1:t})$. At every time instant, thus, we implement particle propagation and weight computation [19]. There is a third step that also needs to be applied and is known as resampling [5], where we basically remove particles that have low weights and replicate ones with large weights. With this step, we prevent the deterioration of the PF with time.

The main reason for the curse-of-dimensionality of PF is the difficulty in drawing "good" particles in high-dimensional spaces. One can readily see how the problem arises. Suppose that the space of interest is a hypercube with side 2 units. Let the important region where probability mass is significant be an inscribed sphere with radius 1. In a one-dimensional space, if we draw uniformly, particles come from the "sphere" with probability one. In a two-dimensional

space it drops to $\pi/4$. In general for a *d*-dimensional space, the probability is given by

$$P(d) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)2^d}.$$
 (5)

Figure 1 displays the previous function. It is clear that with the increase of d, it becomes extremely challenging to sample from the parts of the state-space with non-negligible probability masses. Thus, the onus is on devising smart schemes that guarantee the generation of "good" particles.



Fig. 1. Probability of success.

3.2. Multiple particle filtering

In curtailing the curse-of-dimensionality, we propose to 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 split into *n* subspaces, $\Omega_{x_{i,t}}$, $i = 1, 2, \cdots, n$, where $\Omega_{x_t} = \times_{i=1}^n \Omega_{x_{i,t}}$.

As a result of breaking the state-space into subspaces, instead of estimating the joint posterior, we settle for less, and are interested in estimating various posteriors of subsets of the states, $p(x_{i,t}|y_{1:t})$, $i = 1, 2, \dots, n$. Thus, with this approach we give up the goal of getting the joint distribution of the state and instead settle for tracking a set of marginalized filtering distributions. This should not be necessarily viewed as a setback because when we deal with systems with a very large number of states, it does not even make much sense to insist on tracking all of them jointly. We also point out that in problems with a large number of unknowns, one usually exploits the independence properties of distributions to represent high-dimensional distributions with lower ones compactly [20]. For example, in a system where the dynamic states describe physical quantities related to geographical locations, as in geophysical systems, states corresponding to locations that are far apart are practically independent of each other. In problems related to target tracking based on received signal strength measurements, the targets can be tracked independently for as long as their spatial separation is big enough.

In most cases of interest, the propagation of the states of the *i*th particle filter depends on states of some other filters. Furthermore, the computation of the weights of the particle filter depends on states of other filters. Thus, the particle filters are coupled in that they need to exchange information so that they can proceed with accurate sequential estimation of the distributions they are tasked to track. In our previous efforts [12, 13, 18], we proposed that coupled filters would exchange either point estimates, higher order moments or particles when necessary. These approaches could be referred to as *non-parametric* multiple particle filters.

3.3. Parametric multiple particle filtering

In this paper we propose an alternative to the *non-parametric* approach to MPF. As in the *non-parametric* scheme, the main hurdle in the implementation of the multiple particle filter is the propagation of the particles and the computation of their weights. We propose

that each particle filter, after obtaining its discrete random measures of the filtering and predictive distributions at t-1, computes their approximation by parametric distributions with parameters $\phi_{i_f,t-1}$ and $\phi_{i_p,t}$, respectively. In other words, we seek parametric distributions $p(x_{i,t-1}|\phi_{i_f,t-1})$ and $p(x_{i,t}|\phi_{i_p,t})$ that approximate the filtering (needed for particle generation) and predictive (needed for weight computation) distributions of $x_{i,t-1}$ and $x_{i,t}$, respectively, i.e.,

$$p(x_{i,t-1}|y_{1:t-1}) \approx p(x_{i,t-1}|\phi_{i_f,t-1}),$$
 (6)

$$p(x_{i,t}|y_{1:t-1}) \approx p(x_{i,t}|\phi_{i_p,t}),$$
 (7)

where the forms of the approximating distributions are known but their parameters $\phi_{i_f,t-1}$ and $\phi_{i_p,t}$ are unknown. We emphasize that the approximating distributions are obtained from the filters' random measures. This approach is similar to that used in formulating Gaussian particle filtering [21] and Gaussian sum particle filtering [22].

We proceed by way of example. For simplicity, we assume that the distributions $p(x_{i,t-1}|\phi_{i_f,t-1})$ and $p(x_{i,t}|\phi_{i_p,t})$ are Gaussians. This means that the parameters $\phi_{i,t-1}$ are the mean and covariance of $x_{i,t-1}$ given $y_{1:t-1}$, i.e., $\mu_{i_f,t-1}and\Sigma_{i_f,t-1}$, whereas $\phi_{i_p,t}$ represents the mean and covariance of $x_{i,t}$ given $y_{1:t-1}$, i.e., $\mu_{i_p,t}and\Sigma_{i_p,t}$. In brief, and considering that at time t-1 the *i*th filter approximates its marginal posterior distribution by the random measure $\chi_{i,t-1} = \left\{ x_{i,t-1}^{(m)}, w_{i,t-1}^{(m)} \right\}_{m=1}^{M}$, the algorithm proceeds as follows:

1. Obtain the filtering distribution $\mathcal{N}_f(\mu_{i_f,t-1}, \Sigma_{i_f,t-1})$

$$\mu_{i_f,t-1} = \sum_{m=1}^{M} w_t^{(m)} x_{i,t-1}^{(m)},$$

$$\Sigma_{i_f,t-1} = \sum_{m=1}^{M} w_t^{(m)} \left(x_{i,t-1}^{(m)} - \mu_{i,t-1} \right) \left(x_{i,t-1}^{(m)} - \mu_{i,t-1} \right)^{\top}$$

2. Propagate particles $x_{i,t-1}^{(m)}$

A particle is drawn from the set $x_{i,t-1}^{(m)}, m = 1, \dots, M$ (resampling) and it is propagated J times. If the propagation of the particle requires particles from other filters, the *i*th particle filter uses the Gaussian approximations of these filters to generate the needed particles $x_{i,t-1}^{(j)}$, i.e.,

$$x_{i,t}^{(m,j)} \sim p\left(x_{i,t} | x_{i,t-1}^{(m)}, x_{i_f,t-1}^{(j)}\right),$$

where $x_{i_f,t-1}^{(j)}$ is a particle generated from all the filtering Gaussian distributions of the other filters and that is needed for propagation of $x_{i,t-1}$. At the end of this step, the filter has $M \times J$ particles, all with equal weights. These particles and their equal weights form $p^{MJ}(x_{i,t}|y_{1:t-1})$.

3. Obtain the predictive distribution $\mathcal{N}_p(\mu_{i_p,t}, \Sigma_{i_p,t})$

$$\mu_{i_{p},t} = \frac{1}{MJ} \sum_{m=1}^{M} \sum_{j=1}^{J} x_{i,t}^{(m,j)},$$

$$\Sigma_{i_{p},t} = \frac{1}{MJ} \sum_{m=1}^{M} \sum_{j=1}^{J} \left(x_{i,t}^{(m,j)} - \mu_{i_{p},t} \right) \left(x_{i,t}^{(m,j)} - \mu_{i_{p},t} \right)^{\top}$$

4. Computation of the weights of $x_{i,t}^{(m,j)}$

We approximate the evaluation of the weights by first drawing L particles from the approximations of the predictive Gaussian distributions of the filters whose particles are needed. We form L different sets of such particles $x_{i_{p,t}}^{(l)}$, $l = 1, 2, \cdots, L$. Then we evaluate the average likelihood of $x_{i,t}^{(m,j)}$ by

 $w_{i,t}^{(m,j)} \quad \dot{\alpha} \quad \frac{1}{L} \sum_{l=1}^{L} p\left(y_t | x_{i,t}^{(m,j)}, x_{i_p,t}^{(l)}\right). \tag{8}$

5. Downsampling to M particles

After the computation of the weights, each filter still has $M \times J$ particles. To bring the number of particles from $M \times J$ back to M we propose that one samples one particle from each set of children (there are M such sets) and assigns to the drawn particle a non-normalized weight equal to the sum of non-normalized weights of the particles from the respective sets of children. The surviving particles and their weights are used for forming $p(x_{i,t}|y_{1:t})$.

We point out that the choice of the parametric distribution may allow in some cases for closed-form solutions of the computation of the weights, which should always be favored over the Monte Carlo computation of them.

4. SIMULATION RESULTS

We evaluated the proposed *parametric* multiple particle filter for a system of dimension $d_x = 50$ generated according to:

$$\begin{aligned}
x_{1,t} &= .7x_{1,t-1} + .3x_{d_x,t-1} + u_{1,t} \\
x_{2,t} &= .7x_{2,t-1} + .3x_{1,t-1} + u_{2,t} \\
\vdots \\
x_{d_x,t} &= .7x_{d_x,t-1} + .3x_{d_x-1,t-1} + u_{d_x,t},
\end{aligned} \tag{9}$$

where $u_{i,t}$, $i = 1, \dots, d_x$ were independent and identically distributed zero-mean Gaussian perturbations with variance $\sigma_{u_i}^2 = 1$. For simplicity, we assumed that there was only one very non-linear observation per state given by

$$v_{i,t} = e^{\frac{x_{i,t}}{2}} v_{i,t}, \qquad i = 1, \cdots, d_x,$$
 (10)

with $v_{i,t}$ being an independent zero-mean Gaussian random variable with variance $\sigma_{v_i}^2 = 1$. We note that there was no coupling of the observations, and therefore, with the proposed algorithm the *i*th filter only needed information from other filters for generation of particles but not for calculation of weights.

y

We let the system evolve for T = 50 time units and all the filters used the same amount of particles (1,000) when calculating the estimate of the state. We compared the following implementations:

- The standard PF (SPF) algorithm that generated 1,000 particles of dimension 50. We denoted this filter as SPF $1 \times 1000 \times 1$, where the first index denotes the number of filters (in this case one), the second index indicates the number of particles per filter (M = 1,000), and the third index represents the number of children in the propagation step (J = 1).
- The MPF algorithm from [12] where we used 50 filters (one per dimension) and for each of them generated 20 particles of dimension 1. To deal with the coupling of the states given in (9), the filters exchanged the means of their particles that



Fig. 2. Left: Average MSE of the different methods. Right: Average MSE of various implementations of the new parametric MPF.

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 np-MPF $50 \times 20 \times 1$. We point out that this filter used the same amount of particles as the SPF, a total of 1,000. However, in order to obtain the approximation of the marginal distribution of a particular state, it only used 20 particles, so in that respect it was in a clear disadvantage. Also, it is important to point out that this filter belongs to the family of *non-parametric* MPF.

- The MPF algorithm from [18] that used 50 filters, each of them operating with 20 particles of dimension 1 and with 6 children per particle corresponding to the 6 particles received by their coupled filters according to equation (9). For example, filter 1 used 6 particles from filter 10 for each of its *M* particles to be propagated. The required particles from the other filters were obtained by resampling. At each step, the number of particles per filter was brought back to 20 using resampling. Therefore, this filter was also using 20 particles for approximation of the marginal distribution and estimation of the state. Note that this filter did not need to exchange particles when calculating the weights of the particles because the observation equations were decoupled. We denoted this filter as np-MPF 50×20×4 and it also belongs to the family of non-parametric MPF.
- The new MPF algorithm that used 50 filters, each of them operating with 20 particles of dimension 1 and with J = 6children per particle corresponding to the 6 particles that were generated according to the parametric distribution received by the coupled filters according to equation (9). Again, in the case of filter 1 for instance, to propagate its own particles the filter used the filtering distribution transmitted by filter 10 to generate 6 particles for each of its M particles. At each step, the number of particles per filter was brought back to 20 by randomly selecting from the J children of each particle one of them according to their weights. The final weight assigned to a particle was the sum of the weights of its 6 children. Once more this filter was only using 20 particles for approximation of the marginal distribution and estimation of the state. We also note that when calculating the weights according to (8), there was no need for exchanging information between the filters because the observation equations were decoupled. We denoted this filter as p-MPF 50×20×4 and it belongs to

the family of parametric MPF.

Figure 2 (left) shows the average mean square error (MSE) of all the states calculated from 500 runs of the system. It is obvious that the traditional PF suffers from the large dimension of the state. The performance of the most basic *non-parametric* MPF where the filters exchanged point estimates considerably improved with respect to the SPF even though the former was using only 20 particles for approximation of the marginal distributions. The best performance corresponded to the *non-parametric* MPF algorithm with the filters exchanging actual particles. The newly proposed *parametric* MPF algorithm performed closely but it is important to remark the savings in communication of this filter since only the parameters of the distributions need to be exchanged among the filters as opposed to sets of particles.

Finally, in Fig. 2 (right) we present a comparison of different parameter combinations for the new *parametric* MPF algorithm in terms of number of particles per individual filter and number of children generated per particle. The experiment was run 500 times. For the considered 50-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, p-MPF $50 \times 20 \times 6$ from the previous experiment had almost as good performance as the other new MPFs that used either more or less children.

5. CONCLUSION

Particle filtering suffers from the curse-of-dimensionality. Namely, it can be shown that its accuracy critically degrades to the point of complete failure in high-dimensional state-spaces. In this paper, we proposed an approach for addressing this problem based on breaking the high-dimensional distribution of the complete state-space of the system into smaller dimensional (marginalized) distributions and attempting to track these distributions as accurately as possible in a novel way. The filters in the system exchange parametric distributions, which are used to perform generation of particles and weight computation. We demonstrated the proposed approach with computer simulations, which revealed an excellent performance of the proposed method.

6. REFERENCES

- 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.
- [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] B. D. Ripley, Stochastic Simulation, John Wiley & Sons, 1987.
- [4] D. B. Rubin, "A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm," *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 543–546, 1987.
- [5] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings-F*, vol. 140, no. 2, pp. 107–113, 1993.
- [6] F. Daum and J. Huang, "Curse of dimensionality and particle filters," in *the Proceedings of IEEE Aerospace Conference*, Big Sky, MO, 2003, vol. 4, pp. 1979–1993.
- [7] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions* on *Processing*, vol. 50, no. 3, pp. 736–746, 2002.
- [8] E. T. Spiller, A. Budhiraja, K. Ide, and C. K. R. T. 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.
- [18] P. M. Djurić and M. F. Bugallo, "Particle filtering for highdimensional systems," in the Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 2013.
- [19] A. Doucet, N. De Freitas, and N. Gordon, Eds., Sequential Monte Carlo methods in Practice, Springer, 2001.
- [20] D. Kollar and N. Friedman, Probabilistic Graphical Models: Principles and Techniques, The MIT Press, 2009.
- [21] J. H. Kotecha and P. M. Djurić, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592–2601, 2003.
- [22] J. H. Kotecha and P. M. Djurić, "Gaussian sum particle filtering," *EEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602–2612, 2003.