20th International Conference on Information Fusion
Xi'an, China - July 10-13, 2017

A Two-stage Particle Filter in High Dimension

Wenbo Wang
Department of Applied Mathematics
University of Twente
P.O. Box 217, 7500AE Enschede, Netherlands.
Email: w.wang@utwente.nl

Pranab K. Mandal
Department of Applied Mathematics
University of Twente
P.O. Box 217, 7500AE Enschede, Netherlands.
Email: p.k.mandal@utwente.nl

Abstract—**Particle Filter (PF) is a popular sequential Monte Carlo method to deal with non-linear non-Gaussian filtering problems. However, it suffers from the so-called *curse of dimensionality* in the sense that the required number of particle (needed for a reasonable performance) grows exponentially with the dimension of the system. One of the techniques found in the literature to tackle this is to split the high-dimensional state in to several lower dimensional (sub)spaces and run a particle filter on each subspace, the so-called multiple particle filter (MPF). It is also well-known from the literature that a good proposal density can help to improve the performance of a particle filter. In this article, we propose a new particle filter consisting of two stages. The first stage derives a suitable proposal density that incorporates the information from the measurements. In the second stage a PF is employed with the proposal density obtained in the first stage. Through a simulated example we show that in high-dimensional systems, the proposed two-stage particle filter performs better than the MPF with much fewer number of particles.**

Keywords: high-dimensional systems, particle filter, optimal proposal

I. Introduction

Particle filter (PF) [1] is now a popular and well-established sequential Monte Carlo method to deal with nonlinear systems with possibly non-Gaussian errors. It is, however, also well known that PF has difficulty coping with high-dimensional state space systems. In [2], for example, an approximate upper bound for the computational complexity for the PF in high dimension is given and the authors conclude afterwards that the PF does not avoid the *curse of dimensionality*. In [3], the authors consider the number of so-called *good samples* in the state space representation. Their conclusion is that propagating these good samples in the prediction phase and maintaining them as good samples becomes problematic in high-dimensional scenarios. Also, in general, PF has the tendency that after a few time steps, only one particle has the unity weight and the rest zero (often referred to as the *sample impoverishment* problem). The authors in [4] conclude that to avoid this problem the required number of particles grows exponentially with the dimension.

Recently, many efforts have been made to find a general solution for filtering in high-dimension. The block scheme is an important attempt to achieve a fair accuracy of estimation in high dimension while the computation complexity is still tolerable. For instance, the multiple particle filter (MPF) [5], [6] divides the state space into several subspaces and then runs one particle filter in each low-dimensional space. Another approach to avoid sample impoverishment is to optimise the importance or the proposal density used in the particle filter. The authors in [7] suggest the use of an optimal proposal density that minimizes the variance of the particle weights and thereby increasing the effective sample size of the particle representation. In fact, it is argued in [8] that the aforementioned optimal proposal density does postpone the weight degeneracy, though cannot avoid it completely. However, obtaining the optimal proposal is not always possible. In [9], the authors used approximated optimal proposal to show its effectiveness. Another similar approach is the adaptive importance sampling (AIS) technique. With AIS one obtains the particles from the desired posterior distribution through a few iterative steps, using an improved importance density at each iteration. Several such AIS techniques are reviewed in [10].

In this work, we propose a novel two-stage particle filter where the first stage is used to determine the proposal density using the knowledge from the measurements. On one hand, the new method can be thought of as an attempt to find the optimal proposal density, but completely differently from that of [9]. On the other hand, it is similar to the AIS with two iteration steps. However, the final proposal density is derived in a very different manner than that mentioned in [10]. In the AIS, a PF is run at every iteration and they are combined to come up with the final proposal. In our method, the first stage does not use a PF. It decouples the different components of the state vector and uses the measurements to find the best particle-component along every dimension-index and uses them to obtain the proposal density. The numerical results show that in high dimensional systems, the proposed filter performs better than the MPF approach, providing more accurate estimation using much less particles.

The rest of the paper is organised as follows. We introduce in Section II the state space model we are going to use. In Section III and Section IV, we review the basic ideas behind the PF and the MPF, respectively. The main contribution of this work, i.e., the details of the proposed two-stage particle filter is described in Section V. Section VI contains the comparison results for the three methods (PF, MPF and the newly proposed filter, what we call TPF) in the context of high dimension and based on simulated experiments. Finally, some conclusions are drawn in Section VII.

II. State-space Model

We consider a first-order Markov dynamic system described by the following state space model:

$$\textbf{State:} \qquad \mathbf{x}_t = \mathbf{F}(\mathbf{x}_{t-1}, \mathbf{u}_{t-1}), \tag{1}$$

$$\textbf{Measurement:} \; \mathbf{y}_t = \mathbf{G}(\mathbf{x}_t, \mathbf{v}_t), \quad t = 1, 2, \cdots, T, \tag{2}$$

where t denotes the discrete time, T is the length of the time-series, $\mathbf{u}_t$ and $\mathbf{v}_t$ are, respectively, the process and measurement noises and $\mathbf{x}_t, \mathbf{y}_t, \mathbf{u}_t, \mathbf{v}_t \in \mathbb{R}^D$. Furthermore, for the current exposition, we assume that the measurement function $\mathbf{G} : \mathbb{R}^D \to \mathbb{R}^D$ satisfies the following property. For $\mathbf{a} \equiv (a_1, \ldots, a_D) \in \mathbb{R}^D$,

$$\mathbf{G}(\mathbf{a}) = \left(G_1(a_1), \ldots, G_D(a_D)\right), \tag{3}$$

where $G_d : \mathbb{R} \to \mathbb{R}$ for $d = 1, 2, \ldots, D$. In other words, the measurement, y_d, along the d-th dimension depends only on the state, x_d, along the same dimension. The assumption (3) can, however, be easily relaxed. See Remark 2 at the end of Section V-C.

Usually, the state vector is hidden and cannot be observed directly. The main goal is to estimate the current state $\mathbf{x}_t$ from all the measurements, $\mathbf{y}_{1:t}$, up to the current time t. This is given by the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

III. Particle Filter

A particle filter is a sequential Monte Carlo method to approximate the joint posterior $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, recursively in time t. The desired posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is then obtained by marginalizing the joint posterior. The approximation is done by representing the density $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ by a set of weighted particles: $\{\mathbf{x}^i_{0:t}, w^i_t\}_{i=1}^{Np}$. More precisely ([11]),

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N_p} w^i_t \, \delta(\mathbf{x}_{0:t} - \mathbf{x}^i_{0:t}), \tag{4}$$

where $\mathbf{x}^i_{0:t}$ denotes the i-th particle, w^i_t is its associated weight, N_p is the number of particles, $\delta(\cdot)$ is the Dirac delta function.

Given the representation $\{\mathbf{x}^i_{0:t-1}, w^i_{t-1}\}_{i=1}^{Np}$ at time $(t-1)$ and the new measurement $\mathbf{y}_t$, one draws $\mathbf{x}^i_t$ from a so-called proposal density (also known as the importance density) $q(\cdot|\mathbf{x}^i_{0:t-1}, \mathbf{y}_{1:t})$ and updates the weights as

$$w^i_t \propto w^i_{t-1} \frac{p(\mathbf{x}^i_t|\mathbf{x}^i_{t-1}) \, p(\mathbf{y}_t|\mathbf{x}^i_t)}{q(\mathbf{x}^i_t|\mathbf{x}^i_{0:t-1}, \mathbf{y}_{1:t})} \tag{5}$$

where $p(\mathbf{x}^i_t|\mathbf{x}^i_{t-1})$ and $p(\mathbf{y}_t|\mathbf{x}^i_t)$ are determined by (1) and (2), respectively. To avoid unnecessarily carrying the particles with very small weights, often a resampling step is added.

In practice, the importance density $q(\mathbf{x}_t|\mathbf{x}^i_{0:t-1}, \mathbf{y}_{1:t})$ is often taken to be the transition density $p(\mathbf{x}_t|\mathbf{x}^i_{t-1})$. In that case the weight update equation (5) simplifies to

$$w^i_t \propto w^i_{t-1} p(\mathbf{y}_t|\mathbf{x}^i_t). \tag{6}$$

The optimal proposal density [7] that minimizes the variance of the weights is given by $q(\mathbf{x}_t|\mathbf{x}^i_{0:t-1}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t|\mathbf{x}^i_{t-1}, \mathbf{y}_t)$. However, it is usually difficult to obtain.

Though the PF performs quite well for general nonlinear non-Gaussian problems, the performance degrades rapidly as the dimension of the system increases. The MPF, which we describe in the next section, is an attempt to overcome this problem in high dimension.

IV. Multiple Particle Filter

Since the PF works well in low dimensional systems, the multiple particle filter divides the state space into several subspaces/blocks and apply the PF in the lower dimensional space. It thus runs multiple PF [6]. More precisely, MPF splits the state vector $\mathbf{x}_t$ into M blocks, $\mathbf{x}_t = \left(\mathbf{x}^{(m)}_t\right)_{m=1}^{M}$.

In implementing the PF on the m-th block, the proposal used to update the particles is the transition density (for that block), $p(\mathbf{x}^{(m)}_t|\mathbf{x}^{(m)}_{t-1})$. However, according to state equation (1), states of the m-th block at time t may depend on the states of the other blocks, $\mathbf{x}^{(-m)}_{t-1}$, as well. Subsequently, the proposal is taken to be

$$p\left(\mathbf{x}^{(m),i}_t|\mathbf{x}^{(m),i}_{t-1}\right) = p\left(\mathbf{x}^i_t|\mathbf{x}^{(m),i}_{t-1}, \hat{\mathbf{x}}^{(-m)}_{t-1}\right), \tag{7}$$

where $\hat{\mathbf{x}}^{(-m)}_{t-1}$ is the estimated mean of the posterior distribution, at time $(t-1)$, of all the blocks except the m-th one, obtained from the particle representations $\{\mathbf{x}^{(j),i}_{t-1}, w^{(j),i}_{t-1}\}_{i=1}^{N_p}$, $j \neq m$.

The weight update for the m-th block is performed as

$$w^{(m),i}_t \propto w^{(m),i}_{t-1} \, p\left(\mathbf{y}_t|\mathbf{x}^{(m),i}_t\right) \tag{8}$$

where the likelihood $p(\mathbf{y}_t|\mathbf{x}^{(m),i}_t)$ is calculated in a similar fashion as in (7):

$$p\left(\mathbf{y}_t|\mathbf{x}^{(m),i}_t\right) = p\left(\mathbf{y}_t|\mathbf{x}^{(m),i}_t, \hat{\mathbf{x}}^{(-m)}_{t|t-1}\right), \tag{9}$$

where $\hat{\mathbf{x}}^{(-m)}_{t|t-1}$ is the estimated mean of the predicted distribution, at time t, of all the blocks except the m-th one, obtained from the particle representations $\left\{\mathbf{x}^{(j),i}_t, w^{(j),i}_{t-1}\right\}_{i=1}^{N_p}$, $j \neq m$.

It is worth noting here that the MPF provides only the marginalised filtering distribution of each block separately. In other words, it provides $p(\mathbf{x}^{(m)}_t|\mathbf{y}_{1:t})$ for each m, separately and not the full filtering posterior $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, which is the desired quantity. In the next section, we describe our method which does provide the full posterior.

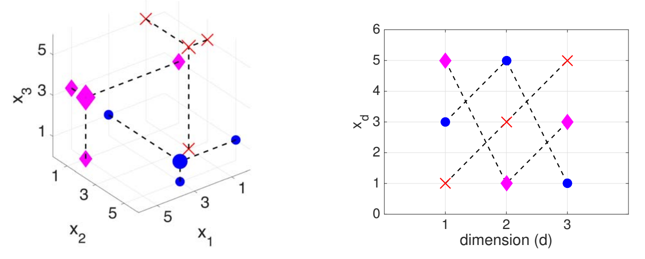
V. The Two-stage Particle Filter

In this section we describe the proposed two-stage PF. The first stage is used to determine a suitable proposal density and the second stage implements a PF with the proposal density of the first stage. To explain the intuition behind the new proposal an alternative visualization of vectors is helpful. Section V-A describes this alternative visualization. Then we give, in Section V-B, the motivation and intuition behind the proposal. The exact derivation of the new proposal is presented in Section V-C. Section V-D contains the pseudocode for the two-stage particle filter algorithm.

A. Alternative visualization of vector

In general, it is difficult to visualise a 4 or higher dimensional vector, due to the limit of perception of human beings. We can, however, represent a vector $\mathbf{a} \equiv (a_1, \ldots, a_D) \in \mathbb{R}^D$ as a piecewise linear curve, by joining the subsequent points $(1, a_1), (2, a_2), \ldots, (D, a_D)$ in a 2D-plot. Here the horizontal axis represents the element-index (or dimension-index) of the vector and the vertical axis represents the value of the vector elements (or equivalently, along the dimensions).

Figure 1 illustrates this alternative visualization for vectors in $\mathbb{R}^3$. The left plot (Figure 1a) provides the common visualization of three 3-dimensional vectors: $[1, 3, 5]$ (red cross), $[3, 5, 1]$ (blue dot) and $[5, 1, 3]$ (magenta diamond). The same three vectors are represented on the right plot (Figure 1b) by the piecewise-linear curves marked with red cross, blue dots and magenta diamonds, respectively.



(a) The common representation (b) The new representation

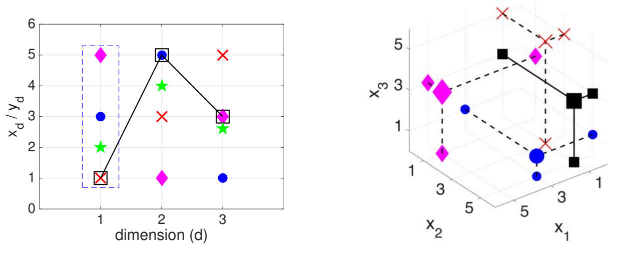
Fig. 1: Alternative visualization of vectors. In the righthand figure, the x-axis shows the dimension-index ($d = 1, 2, 3$) and the y-axis shows the value of the vector elements (i.e., x_d along the d-th dimension).

B. Motivation for the proposal

In a PF algorithm, a weight is assigned to a particle. If the state is multi-dimensional then the weight is for the whole state-vector $\mathbf{x}$. Unless all the components of the particle are good, i.e., come from the higher probability area of the posterior, the weight will be negatively affected by the bad components. Thus, a few bad components in the predicted vector, for example, from the transition density, will result in low likelihood for the particle (vector). Subsequently, a low unnormalized weight will be assigned to the particle vector as a whole and thus to the good components as well. In fact, as the dimension of the state vector increases, a particle will have fewer and fewer good components and thereby reducing the unnormalized weight (for the whole vector) to be close to zero.

In the first stage of our proposed filter, we decouple particles (vectors) into different dimension-indices, so that we can choose the best along every dimension-index. In choosing the best we make use of the measurements. We then combine all these "best" estimates to obtain a good (preliminary) point estimate, $\hat{\mathbf{x}}_t$, of the state-vector that has incorporated information from the measurement.

This can be best visualized with Figure 2. Suppose that the three vectors in Figure 1a are actually the predicted particles $\mathbf{x}_t^i,\ i = 1, \ldots, N_p = 3$. After representing the particles in the alternative manner, as in Figure 1b, we remove the connecting lines and consider each dimension-index d ($d = 1, \ldots, D$), separately. See Figure 2a. We consider only the vertical points along, for example, $d = 1$. Suppose $y_{1,t}$ is the measurement along dimension-index $d = 1$ (represented in the plot by the green star). From these vertical points, we choose the "best" in the sense of likelihood for the measurement $y_{1,t}$. This is represented by the square in Figure 2a. We do the same for all dimension-indices $d = 1, 2, \ldots, D$. The vector $\hat{\mathbf{x}}_t$ is then the vector corresponding to the piecewise-linear curve obtained by joining the squares by lines. The black square in Figure 2b represents $\hat{\mathbf{x}}_t$.



(a) Best particle-element along each dimension-index (b) Best state-vector

Fig. 2: Derivation of $\hat{\mathbf{x}}_t$. For every dimension-index d in the left plot, consider the vertical points (red cross, blue dot and magenta diamond). The one enclosed in a square has the highest likelihood, corresponding to the measurement indicated by green star. The piecewise-linear curve marked by the squares on the left, represents $\hat{\mathbf{x}}_t$, the black square on the right plot.

Finally, to determine the proposal density we modify the transition density by pushing the center of the distribution towards the newly found estimate $\hat{\mathbf{x}}_t$. We now give the exact derivation of the proposal density.

C. Derivation of the new proposal

Suppose, at time $(t-1)$, the weighted particle representation is given by $\left\{\mathbf{x}_{t-1}^i, w_{t-1}^i\right\}_{i=1}^{N_p}$. We propagate the particles using the state equation (1) and predict the particles at time t as $\mathbf{x}_{t-1}^i \sim p(\mathbf{x}_t | \mathbf{x}_{t-1})$. Note that using assumption (3) on $\mathbf{G}$, we can rewrite the D-dimensional measurement equation as D equations given by

$$y_{t,d} = G_d(x_{t,d}) + v_{t,d}, \quad 1 \leq d \leq D, \tag{10}$$

where $\mathbf{x}_t = \left(x_{t,d}\right)_{d=1}^D$. We now consider

$$\hat{\mathbf{x}}_t = \left(x_{t,d}^{i^*}\right)_{d=1}^D, \quad i^* = \arg \max_{1 \leq i \leq N_p} p\left(y_{t,d} | x_{t,d}^i\right), \tag{11}$$

where the likelihood $p(y_{t,d}|x_{t,d}^i)$ is corresponding to (10).

The proposal density $q(\cdot)$ is then obtained by pushing the center of the prediction density corresponding to (1) towards $\hat{\mathbf{x}}_t$. More precisely, samples $\mathbf{x}_t^i \sim q(\cdot)$ are generated as follows.

$$\begin{aligned}&\text{(a) Draw } \tilde{\mathbf{x}}_t^i \sim p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i),\ \tilde{\tilde{\mathbf{x}}}_t^i \sim \mathcal{N}(\hat{\mathbf{x}}_t, \sigma^2\,\mathbf{I}),\ \sigma > 0.\\ &\text{(b) Take } \mathbf{x}_t^i = \beta\,\tilde{\tilde{\mathbf{x}}}_t^i + (1-\beta)\,\tilde{\mathbf{x}}_t^i,\quad \beta \in [0,1].\end{aligned} \tag{12}$$

If the state noise is additive Gaussian with $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \sigma_u^2\,\mathbf{I})$, then the new proposal density is equivalent to

$$\begin{aligned}&q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})\\ &= \mathcal{N}\Big(\beta\hat{\mathbf{x}}_t + (1-\beta)\mathbf{F}(\mathbf{x}_{t-1}), (\beta^2\sigma^2 + (1-\beta)^2\sigma_u^2)\,\mathbf{I}\Big).\end{aligned} \tag{13}$$

In general, the proposal density is given by the convolution of the $\mathcal{N}(\beta\,\hat{\mathbf{x}}_t, \beta^2\,\sigma^2\,\mathbf{I})$–density and the conditional density of $(1-\beta)\,\mathbf{x}_t$ given $\mathbf{x}_{t-1}$.

Remark 1: We note here that by pushing the center of the transition density towards $\hat{\mathbf{x}}_t$ can be considered as an attempt to incorporate the information from the measurement into the importance density. Though this is not exactly the optimal importance function given in [7].

Remark 2: We like to state here that the two-stage particle filter can be applied even without the restriction (3) on the measurement model. The proposed technique can still be applied if, for example, the state and the measurement vector can be divided into equal number of blocks and one block of measurements relates (only) to one unique block of the state vector. The block-sizes need not be the same. So, the dimensions of the state and measurements can also be different. We will discuss the general case in a future exposition.

D. The pseudocode for the proposed filter

The proposed two-stage particle filter (TPF) can be described through Algorithm 1.

Algorithm 1 Two-stage Particle Filter

```
Require: Initial distribution $p(\mathbf{x}_0)$, measurement $\mathbf{y}_{1:T}$
 1: Initialize
 2: Draw $\mathbf{x}_0^i \sim p(\mathbf{x}_0)$ and set $w_0^i = \frac{1}{N_p},\ i = 1, \ldots, N_p$
 3: for $t = 1 : T$ do
 4:     First stage (proposal density)
 5:     Draw $\mathbf{x}_t^i \sim p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$
 6:     Determine $\hat{\mathbf{x}}_t$ according to (11) and the proposal $q(\cdot)$.
 7:     Second stage (particle filter)
 8:     Draw, afresh, $\mathbf{x}_t^i \sim q(\cdot)$ according to (12)
 9:     Assign $\mathbf{x}_t^i$ with weight $w_t^i$ according to (5)
10:     Resample, if necessary
11: end for
```

VI. Numerical Results

In this section we present the simulation results for the proposed two-stage filter (TPF). We compare the performance of the TPF with those of the standard particle filter (SPF) and the multiple particle filter (MPF).

We implement the filters on a system, where the state is is linear-Gaussian and the measurement is exponential. More precisely, the state equation (1) is given by

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{u}_{t-1}, \tag{14}$$

$$\text{with } \mathbf{A} = \begin{bmatrix} 0.1 & 0 & 0 & \cdots & 0 & 0.9\\ 0.9 & 0.1 & 0 & \cdots & 0 & 0\\ 0 & 0.9 & 0.1 & \ddots & 0 & 0\\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots\\ 0 & 0 & \ddots & 0.9 & 0.1 & 0\\ 0 & 0 & \cdots & 0 & 0.9 & 0.1 \end{bmatrix} \tag{15}$$

and the noise $\mathbf{u}_t \sim \mathcal{N}(\mathbf{0}, \sigma_u^2\,\mathbf{I})$. The measurement equation (2) is given in the form (3) as

$$y_{d,t} = e^{\frac{x_{d,t}}{2}} + v_{d,t}, \quad d = 1, 2, \ldots, D, \tag{16}$$

where $\mathbf{v}_t \sim \mathcal{N}(0, \sigma_v^2\,\mathbf{I})$. We generate a synthetic time series with length $T = 100$ starting from the initial state $\mathbf{x}_t = \mathbf{0}$, with $\sigma_u^2 = 1$ and $\sigma_v^2 = 0.1$. In our experiments we vary the dimension of the system to be 3, 10, 30, 100, 300 and 500.

During the implementation of the particle filters we assume the noise parameters σ_u^2 and σ_v^2 are known. We take the initial distribution $p(\mathbf{x}_0)$ to be $\mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{I})$ with two possible values: $\hat{\mathbf{x}}_0 = \mathbf{0}$ and $\hat{\mathbf{x}}_0 = \mathbf{5}$. The idea behind considering these two values is to check to what extent the knowledge about the initial state affects the performance of a filter. Recall that the true $\mathbf{x}_0 = \mathbf{0}$. So, with the first choice we assume that the truth is approximately known whereas the second choice is far form the truth.

For the MPF we divide the D-dimensional state-vector into B blocks of equal sizes. For $D = 3, 10$ and 30 we use $B = 3, 5$ and 6, respectively. In the other cases we use $B = 10$.

For the proposed TPF we set the design parameters $\beta = 0.2$ and $\sigma^2 = 0.1$.

As for the number of particles, N_p, we vary them as $10^3, 10^4, 10^5$ for the SPF, $100, 1000, 5000$ for each block in the MPF and $50, 100, 200$ for the TPF.

For comparison purpose we run every algorithm (with a particular set of parameters) $R = 70$ times and compare the averages over these Monte Carlo runs.

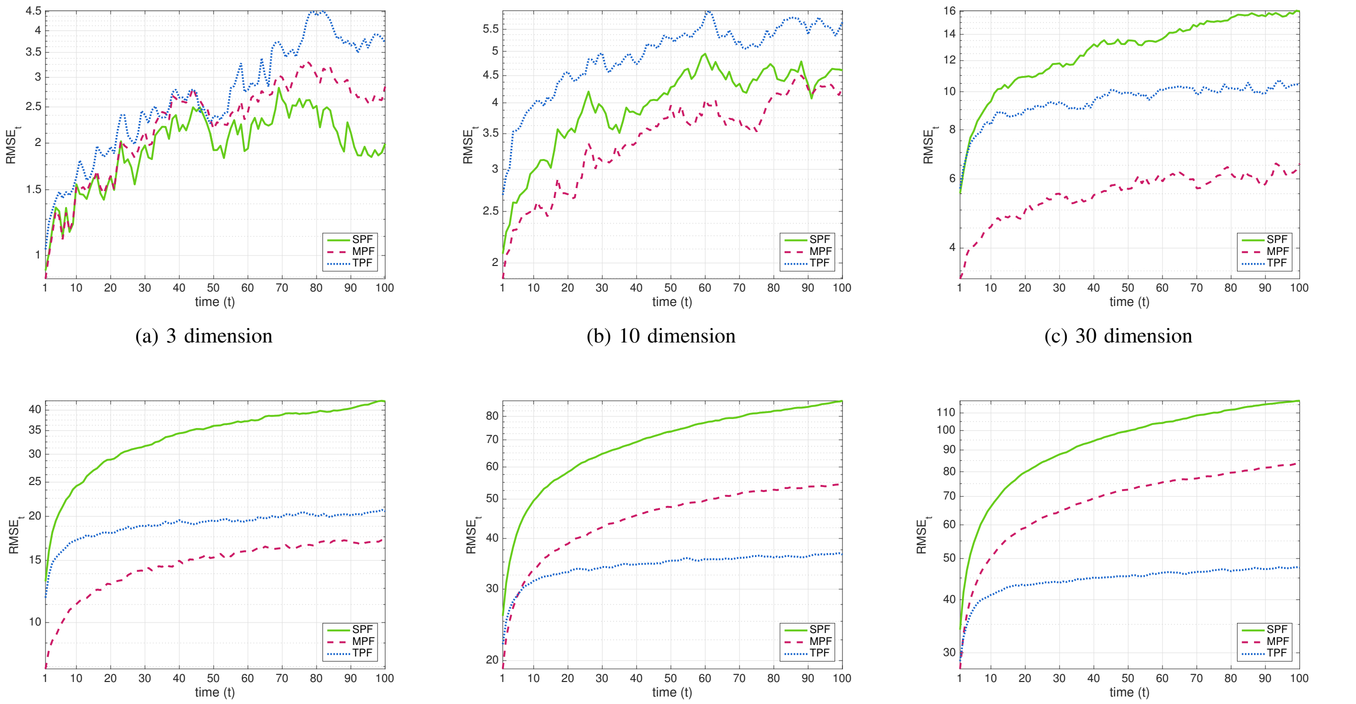
Table I summarizes the (design) parameters used in the implementations of different filters.

A. Comparison over the timeline

To start with, we apply the algorithms SPF, MPF with the highest number of particles in our design (10^5 and 5000, respectively) and the proposed TPF with 100 particles. For each algorithm, we calculate the RMSE for each $t = 1, \ldots, T$ (based on $R = 70$ Monte Carlo runs) as:

$$RMSE_t = \sqrt{\frac{1}{R}\sum_{r=1}^{R}\sum_{d=1}^{D}\left(\hat{x}_{d,t}^{[r]} - x_{d,t}\right)^2}, \tag{17}$$

where $\hat{x}_{d,t}^{[r]}$ is the point estimate (the average of the particles along dimension-index d) of $x_{d,t}$ in the r-th Monte Carlo run.



(a) 3 dimension (b) 10 dimension (c) 30 dimension

(d) 100 dimension (e) 300 dimension (f) 500 dimension

Fig. 3: $RMSE$ over time. The red dashed line represents the MPF, the green solid line the SPF, and the blue dotted line the proposed TPF. All filters use the initial distribution with mean $\hat{\mathbf{x}}_0 = \mathbf{0}$.

TABLE I: Parameters for different PF's

Parameters		**value(s)**
Dimension (D)		$3, 10, 30, 100, 300, 500$
Initial distribution: $p(\mathbf{x}_0) = \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{I})$		$\hat{\mathbf{x}}_0 = [0, \cdots, 0]_D$, or $[5, \cdots, 5]_D$
State noise ($\mathbf{u}$)		$\sigma_u^2 = 1$
Measurement noise ($\mathbf{v}$)		$\sigma_v^2 = 0.1$
Length time series (T)		100
Monte Carlo run (R)		70
# Particles (N_p)	in SPF	1×10^3, 1×10^4, 1×10^5
(for each block)	in MPF	1×10^2, 1×10^3, 5×10^3
	in TPF	5×10^1, 1×10^2, 2×10^2
For proposal in TPF		$\beta = 0.2$, $\sigma^2 = 0.1$
Number of blocks in MPF		$3, 5, 6, 10$

Figure 3 presents the $RMSE_t$ values for different algorithms and for different system-dimensions D. These results correspond to the initial distribution with mean $\hat{\mathbf{x}}_0 = \mathbf{0}$, i.e., with some knowledge about the initial distribution.

We see that as the dimension of the system increases, the performance of SPF (even with 10^5 particles) degrades as compared to the other two algorithms. For relatively low dimensional systems ($D = 3 - 100$), similar RMSE values indicate similar performance of the MPF and TPF. However, for high-dimensional systems, especially for $D = 500$, the TPF has much lower RMSE, and thus outperforms the MPF.

To test how important the knowledge about the initial state is, we perform a similar analysis of the algorithms with the initial distribution having mean $\hat{\mathbf{x}}_0 = \mathbf{5}$. We notice that at the beginning (i.e., for small t's) the RMSE values corresponding to $\hat{\mathbf{x}}_0 = \mathbf{5}$ is larger than those corresponding to $\hat{\mathbf{x}}_0 = \mathbf{0}$. However, as time progresses, the RMSE's are similar, no matter what the dimension D is. In other words, a good knowledge of the initial state is not so important in the long run. Also, we notice that the proposed TPF suffers the least from inaccurate initial state. We present the comparison for only $D = 500$ in Figure 4.

B. Stability of the algorithms

In order to judge how stable (over different Monte Carlo runs) the performance of an algorithm is, we compare the time averaged error given by

$$TAE_r = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\sum_{d=1}^{D}\left(\hat{x}_{d,t}^{[r]} - x_{d,t}\right)^2} \qquad (18)$$

for each Monte Carlo run $r = 1, \ldots, 70$.

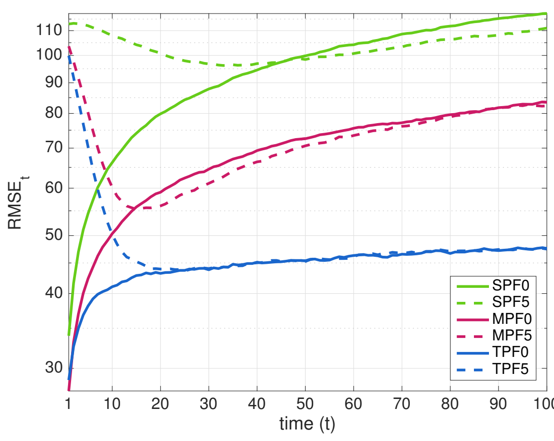


Fig. 4: Effect of (incorrect) initial guess. In the legends, '0' implies that the method uses the initial distribution with mean $\hat{\mathbf{x}}_0 = \mathbf{0}$, i.e., we roughly know the initial state, whereas '5' corresponds to $\hat{\mathbf{x}}_0 = \mathbf{5}$, indicating that we do not use any specific knowledge about the initial state.

The TAE's for 15 different runs are plotted in Figure 5 for $D = 30, 100$ and 500. Table II contains the mean and standard deviation of all the TAE's for all D's. The standard deviations are given within parentheses. Low standard deviations indicate that all the algorithms are quite stable. Also, the superior performance of the TPF in high dimension is clearly seen from the lower mean TAE than those of MPF and SPF. We also notice that when we do not assume the precise knowledge of the initial state ($\hat{\mathbf{x}}_0 = \mathbf{5}$), the TAE is slightly larger than than that with $\hat{\mathbf{x}}_0 = \mathbf{0}$. That is mainly due to the differences in the beginning of the time series. As observed in Figure 4, the estimates do not differ much as time progresses.

TABLE II: Mean and standard deviation of the TAE's. Standard deviations are given within parentheses.

	Dimension (D)					
	3	10	30	100	300	500
SPF0	1.576 (1.303)	3.951 (0.9571)	13.05 (1.913)	34.70 (2.802)	71.40 (3.589)	96.97 (3.519)
SPF5	1.619 (1.288)	4.166 (0.8590)	13.59 (2.079)	35.31 (2.651)	73.85 (3.405)	103.3 (3.68)
MPF0	1.694 (1.734)	3.216 (1.473)	5.526 (0.8356)	14.88 (1.066)	46.30 (2.003)	70.19 (2.442)
MPF5	1.732 (1.725)	3.307 (1.450)	5.899 (0.7721)	15.78 (0.9546)	47.35 (1.832)	72.13 (2.393)
TPF0	2.194 (1.974)	4.869 (1.168)	9.537 (0.7067)	19.06 (0.8192)	34.42 (0.9271)	44.82 (0.8334)
TPF5	2.527 (2.395)	5.272 (1.443)	10.22 (0.7166)	20.72 (0.8677)	37.57 (0.8698)	49.22 (0.8212)

C. *Performance with varying number of particles*

In this section, we compare the performances of the three algorithms with varying number of particles. In general, increasing the number of particle will increase the accuracy of the filter. However, increasing the number of particles will also require more computational power. In this section, we would like to analyze the relationship between the number of particles and the accuracy for varying system-dimensions. As the measure of accuracy we use the mean time averaged error defined by

$$MTAE = \frac{1}{R}\sum_{r=1}^{R} TAE_r, \tag{19}$$

where TAE_r is given by (18). For this analysis we only consider the filters using initial state distribution having mean $\hat{\mathbf{x}}_0 = \mathbf{0}$. The number of particles used are as shown in Table I. The results are plotted in Figure 6.

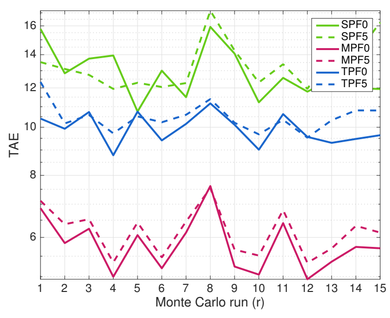
We see that indeed as the number of particles increases the performance of all the algorithms increases (the solid line is below the dashed line which is below the dotted line). For 10 or higher-dimensional systems performance of SPF with 10^5 particles is worse than that of MPF with 100 particles (for each block) or TPF with only 50 particles. For very high dimensional systems ($D = 300, 500$), the new filter TPF even with 50 particles performs better than the MPF with 5000 particles (for each block). We could conclude from here that the new filter is an efficient one for the high-dimensional problems.

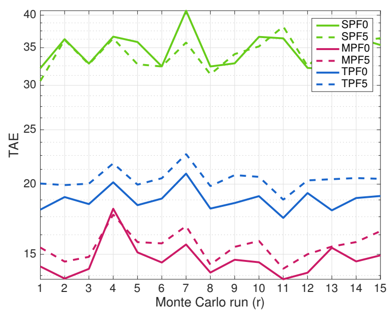
To compare the computational complexity of the MPF and the TPF we have compared the execution times of each algorithm for each run. Figure 7 plots the execution times for 15 different runs for $D = 300$. The statistics (means and standrad deviations) of all the execution times for $D = 300$ and $D = 500$ are presented in Table III.

TABLE III: Comparison of the execution times (in seconds)

	MPF			TPF		
$D \setminus N_p$	100	1000	5000	50	100	200
300	9.377 (0.8635)	84.17 (6.694)	424.5 (39.79)	6.466 (0.9055)	12.67 (1.908)	27.27 (3.649)
500	21.04 (0.7241)	179.8 (13.55)	891.8 (76.87)	17.34 (2.426)	36.63 (4.983)	72.06 (9.772)

We see in Figure 7 that the MPF usually consumes more time than the TPF, when $D = 300$. A very similar behaviour can be seen in Table III for $D = 500$, except that the difference in execution times are now higher. The MPF with 100 particles (for each block) executes slightly faster than the TPF with 100 particles. However, taking into consideration that the estimation performance of the TPF is far better than the MPF (see Figure 6), we conclude that in high dimensional systems the proposed TPF provides more accurate estimation by consuming less time than the MPF.


(a) 30 dimension


(b) 100 dimension

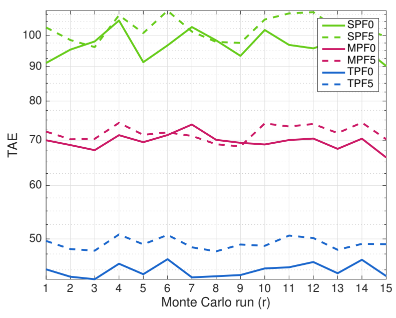

(c) 500 dimension

Fig. 5: TAE for 15 Monte Carlo runs. The red line represents the MPF, the green line the SPF, and the blue line the proposed TPF. The solid line corresponds to the initial distribution having mean $\hat{\mathbf{x}}_0 = \mathbf{0}$, and the dashed line corresponds to $\hat{\mathbf{x}}_0 = \mathbf{5}$.

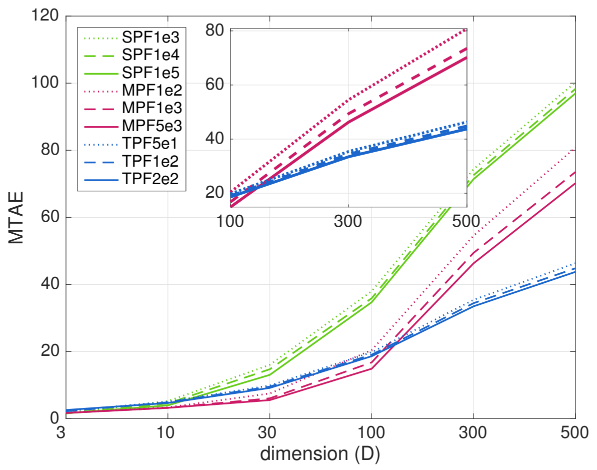


Fig. 6: Comparison with different number of particles. The green colour represents the SPF, the red colour the MPF and the blue colour the TPF. In the legend, for example, SPF1e5 means the SPF with 1×10^5 particles. For MPF it is the number of particles for each block. All plots correspond to the initial distribution with mean $\hat{\mathbf{x}}_0 = \mathbf{0}$. The inset plot is the zoomed in version of the MPF and the TPF for higher dimensions.

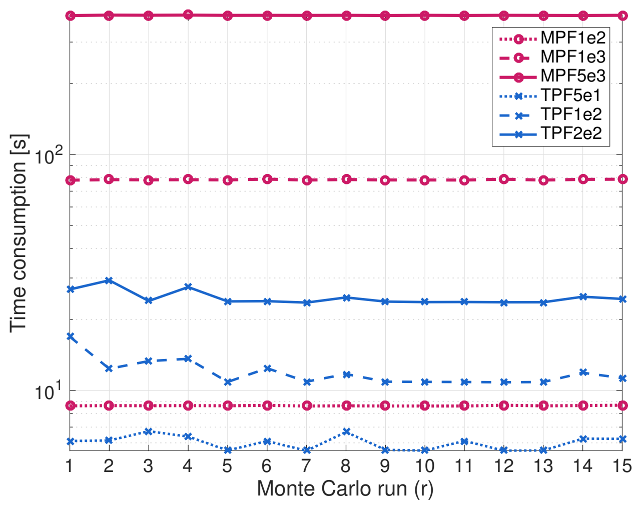


Fig. 7: Execution times (in log scale) for 15 Monte Carlo runs, for $D = 300$. The red circles represent the MPF and the blue crosses the TPF. In the legend, for example, TPF2e2 means the TPF method with 2×10^2 particles. For MPF it is the number of particles for each block.

VII. Conclusion

In this article, we have proposed a new particle filter (PF) that can be used with high-dimensional systems. The proposed filter is a two-stage process. In the first stage a suitable proposal density is determined. In the second stage a PF is implemented using the proposal density found in the first stage. With the help of a simulated numerical example we have demonstrated that the proposed two-stage particle filter (TPF) outperforms the multiple particle filter (MPF) in high-dimensional scenarios. The TPF provides more accurate estimates with less number of particles.

In the current form, the TPF requires the model to satisfy that every component of the state-vector has its own observation dependent only on it, see condition (3). However, as mentioned in the Remark 2 this condition can be relaxed. TPF can be implemented if we can split both the state and the measurement into same number of blocks and the r-th block of measurements depends only on the r-th block of the state. The block sizes do not need to be the same. Also, the current implementation is done on a unimodal system. As immediate future work we would like to focus on multimodal distribution and flexible relationships between the observation-components and the state-components.

Acknowledgment

The authors would like to thank the support from the EC Seventh Framework Programme [FP7 2013-2017] TRAcking in compleX sensor systems (TRAX) Grant agreement no.: 607400.

References

[1] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113, 1993.

[2] F. Daum and J. Huang. Curse of dimensionality and particle filters. In *Proceedings of 2003 IEEE Aerospace Conference*, volume 4, pages 1979–1993. IEEE, March 2003.

[3] P. M. Djurić and M. F. Bugallo. Particle filtering for high-dimensional systems. In *2013 IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 352–355. IEEE, Dec 2013.

[4] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.

[5] P. M. Djurić, T. Lu, and M. F. Bugallo. Multiple particle filtering. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'07, Honolulu, Hawaii, USA*, volume 3, pages 1181–1184. IEEE, April 2007.

[6] M. F. Bugallo, T Lu, and P. M. Djurić. Target tracking by multiple particle filtering. In *2007 IEEE aerospace conference*, pages 1–7. IEEE, 2007.

[7] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

[8] C. Snyder. Particle filters, the "optimal" proposal and high-dimensional systems. In *Proceedings of the ECMWF Seminar on Data Assimilation for atmosphere and ocean*, 2011.

[9] S. Saha, P. K. Mandal, Y. Boers, H. Driessen, and A. Bagchi. Gaussian proposal density using moment matching in smc methods. *Statistics and Computing*, 19(2):203–208, 2009.

[10] M. F. Bugallo, L. Martino, and J. Corander. Adaptive importance sampling in signal processing. *Digital Signal Processing*, 47:36–49, 2015.

[11] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.