AN EFFICIENT TWO-STAGE SAMPLING METHOD IN PARTICLE FILTER

Qi Cheng and Pascal Bondon

CNRS UMR 8506, Université Paris XI, France.

August 27, 2011

Abstract

We present a modified bootstrap filter to draw particles in the particle filter. The proposal distribution for each particle involves sampling from the state-space model a number of times, and then selecting the sample with the highest measurement likelihood. Numerical examples show that this filter outperforms the bootstrap filter with the same computational complexity when the state noise has a large variance.

Keywords: Particle filter, state estimation, Monte Carlo methods, importance sampling, nonlinear filter.

1 Introduction

Nonlinear filtering is a very active topic in signal processing and control theory. There is a vast literature on this subject, see Gelb (1974); Spall (1988); Tanizaki (1996); West and Harrison (1997); Doucet et al. (2001) for excellent references among others. Although the equations of the optimal nonlinear filter have been developed since the middle of the 1960s, the involved integrals are intractable. Hence, many suboptimal nonlinear filters have been introduced in the literature.

The simplest way to approximate a nonlinear state space model is to replace the state transition and the measurement equations by Taylor series expansions. The extended Kalman filter (EKF), the modified gain extended Kalman filter (MGEKF), the iterated EKF and the second order nonlinear filter are famous methods of this type. The EKF uses first-order Taylor series expansions, and then uses the Kalman filter to estimate the state. Although the state estimation given by the EKF is biased, this method is still widely used because of its simpleness. The convergence of the EKF was studied by Reif et al. (1999) where it was shown that the estimation errors of the EKF remain bounded in the mean-square sense when the initial estimation errors and the variances of the disturbing noises are small enough.

The unscented Kalman filter (UKF) uses several so called sigma points to calculate recursively the means and covariances used in the Kalman filter, see Van der Merwe (2004). The UKF implements the Kalman filter by quasi-Monte Carlo methods. Many algorithms are similar to the UKF. The central difference filter and the divided difference filter use Sterling polynomial interpolation to approximate the means and covariances which are used in the Kalman filter. As pointed out by Van der Merwe (2004), these two filters are essentially identical to the UKF. At first glance, UKF and EKF are different. However, both filters use the Gaussian distribution to approximate the true posterior distribution. Essentially, EKF and UKF are two different implementations of the linear update equation in the Kalman filter. When the variance of the observation noise is small, the UKF provides generally a more accurate state estimation than the EKF.

A different approach of the nonlinear filtering problem is the Bayesian approximation. The conditional posterior probability density of the state given the observation constitutes the solution to the statistical inference problem. In the linear Gaussian model, this density is Gaussian and therefore it depends only on the two first moments. In the general nonlinear state space model, the explicit expression of this density is unknown. Then, the key problem is to obtain a good approximation of this density. Many works have been done in this direction, see Sorenson and Alspach (1971); Masreliez (1975); Kitagawa (1987); Kramer and Sorenson (1988); Carlin et al. (1992); Gordon et al. (1993) among others. An excellent survey of the Bayesian filtering is given by Chen (2003).

Using Monte Carlo methods to approximate the recursive Bayesian filter is now popular, see for examples Kitagawa (1987); Kramer and Sorenson (1988); Tanizaki (1991); Carlin et al. (1992); Gordon et al. (1993). A numerical integration method was used by Kitagawa (1987), and a piecewise constant function to approximate the conditional posterior density was proposed by Kramer and Sorenson (1988). The importance sampling in nonlinear filtering was introduced by Tanizaki (1991) where a constant function was used as the importance function. The Gibbs sample was used by Carlin et al. (1992) to approximate the posterior density. An important progress in nonlinear filtering was achieved by Gordon et al. (1993) where a sequential sampling importance resampling method was developed. Since then, a lot of similar algorithms called particle filter (PF) or sequential Monte Carlo (SMC) methods have been developed, see Liu and Chen (1998); Doucet et al. (2000, 2001); Arulampalam et al. (2002); Hue et al. (2002); Cappé et al. (2007); Campillo and Rossi (2009).

Up to now, the PF is the most successful nonlinear filter. The PF has been used successfully in many domains such as guidance, signal and image processing and computer vision, but its performance depends heavily on the choice of the so called importance distribution (ID). Many works have been done to choose the ID, but no general rule seems to exist. The most popular

choice is to use the transition prior function as the ID, see Gordon et al. (1993). This method does not use the latest information of the observation. To overcome this problem, Doucet et al. (2000) and Van der Merwe et al. (2000) suggested respectively to use the EKF and the UKF to produce the ID. Pitt and Shephard (1999) proposed an auxiliary particle filter (APF) which uses an auxiliary variable to select the particles. These methods may provide a good approximation of the true posterior distribution, but they are valid in the environment of a small state noise variance. A comparison of some methods can be found in Šimandl and Straka (2007). In this paper, a new method which uses likelihood to choose the particles from the ID is introduced and compared with the standard PF.

The remainder of this paper is organized as follows. The principle of the PF is introduced in Section 2. In Section 3, a modified bootstrap filter (MBF) is proposed. Then, the effectiveness of our method is illustrated by numerical examples in Section 4. Finally, some conclusions are given in Section 5.

2 Particle filter

Consider a dynamic nonlinear discrete time system described by a state-space model

$$x_t = f(x_{t-1}) + u_t, \tag{1}$$

$$y_t = h(x_t) + v_t, \tag{2}$$

where x_t is the hidden state, y_t is the observation, and u_t, v_t are the state and observation noise. Both noises are independent and identically distributed sequences and are mutually independent. When we write (1), we always assume implicitly that u_t is independent of $\{x_{t-k}, k \geq 1\}$. This condition is natural when the process (x_t) is generated from the model in the increasing time order. Then, x_t is a homogeneous Markov chain, i.e., the conditional probability density of x_t given the past states $x_{0:t-1} = (x_0, \ldots, x_{t-1})$ depends only on x_{t-1} through the transition density $p(x_t|x_{t-1})$, and the conditional probability density of y_t given the states $x_{0:t}$ and the past observations $y_{1:t-1}$ depends only on x_t through the conditional likelihood $p(y_t|x_t)$. We further assume that the initial state x_0 is distributed according to a density function $p(x_0)$.

The objective of filtering is to estimate the posterior density of the state given the past observations $p(x_t|y_{1:t})$. A recursive update of the posterior density as new observations arrive is given by the recursive Bayesian filter defined by

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1},$$

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})},$$

where the conditional density $p(y_t|y_{1:t-1})$ can be calculated by $p(y_t|y_{1:t-1}) = \int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t$.

The difficulty to implement the recursive Bayesian filter is that the integrals are intractable, except for a linear Gaussian system in which case the closed-form solution of the integral equations is the well known Kalman filter introduced by Kalman (1960).

The PF uses Monte Carlo methods to calculate the integrals. The posterior density $p(x_{0:t}|y_{1:t})$ is represented by a set of N random samples $x^i_{0:t}$ (particles) drawn from $p(x_{0:t}|y_{1:t})$ with associated normalized positive weights ω^i_t ($\sum_i \omega^i_t = 1$). The posterior density is approximated by the discrete distribution, $\sum_{i=1}^N \omega^i_t \delta_{x^i_{0:t}}$, and the conditional expectation of any integrable function $g(x_{0:t})$ is approximated by the finite sum,

$$E[g(x_{0:t})|y_{1:t}] = \int g(x_{0:t})p(x_{0:t}|y_{1:t})dx_{0:t} \simeq \sum_{i=1}^{N} \omega^i_t g(x^i_{0:t}).$$

In general, it is difficult to sample directly from the full posterior density. To overcome this difficulty, the method of importance sampling is used, see e.g. Robert and Casella (2004). The particles $x^i_{0:t}$ are drawn from an easy sampling ID $q(x_{0:t}|y_{1:t})$ and we define the non-normalized weights as

$$\omega^i_t = \frac{p(x^i_{0:t}|y_{1:t})}{q(x^i_{0:t}|y_{1:t})}.$$

The ID is chosen to factorize such that

$$q(x_{0:t}|y_{1:t}) = q(x_t|x_{t-1}, y_t)q(x_{0:t-1}|y_{1:t-1}),$$

in order that the weights can be updated sequentially as

$$\omega^i_t \propto \omega^i_{t-1} \frac{p(y_t|x^i_t)p(x^i_t|x^i_{t-1})}{q(x^i_t|x^i_{t-1}, y_t)}. \tag{3}$$

We can implement recursively a basic sequential importance sampling (SIS) PF in the following steps, see Arulampalam et al. (2002) :

1. Sample the particles $x^i_t \sim q(x_t|x^i_{t-1}, y_t)$;
2. Update the weights according to (3).

The SIS algorithm suffers from a problem of degeneracy : the variance of the importance weights increases over time and only a few particles have non-zero normalized weights. Hence, the estimation becomes unreliable. The sampling importance resampling filter has been developed by Gordon et al. (1993) to overcome this drawback. The objective of resampling is to eliminate samples with low importance weights and multiply samples with high importance weights. This is achieved by resampling the particles $x^i_{0:t}$ with their weights ω^i_t. After resampling, the samples have equal weights ($\omega^i_t = \frac{1}{N}$). Many methods can be used to achieve this objective. The three most popular resampling algorithms are the multinomial resampling, the systematic resampling, and the residual resampling.

Choosing the ID is one of the key problems in the PF. The optimal ID satisfies $q(x_t|x_{0:t-1}, y_{1:t}) = p(x_t|x_{t-1}, y_t)$ and fully exploits the information in both x_{t-1} and y_t, see e.g. Doucet et al. (2000). In practice, this distribution is unknown for a general nonlinear model and therefore, it is impossible to sample from it. The second choice of ID is the transition density for its easiness to sample and leads to the standard bootstrap filter (BF).

The BF just uses the transition density as the ID and the information of the observation is not used. When the likelihood lies in the tails of the prior distribution or it is too narrow, most particles drawn from the prior distribution have small weights and the estimation is not reliable. In this case, the likelihood provides more information than the prior.

3 A modified bootstrap filter

The idea of the local linearization PF is to move the particles from the domain of low likelihood to the domain of high likelihood by using existing local linearization filters like EKF, UKF, iterated EKF and iterated UKF to produce the ID in the PF. The local linearization PF uses the Gaussian distribution to approximate the true posterior distribution. It was proposed by Doucet et al. (2000) to use the EKF to produce the ID in the PF (we called this algorithm PF-EKF). Similarly, the unscented particle filter (UPF) developed by Van der Merwe et al. (2000) uses the UKF to produce the ID and it was reported by Van der Merwe et al. (2000) that the UPF outperforms the PF-EKF. This method has some similarities with the Gaussian particle filter (Kotecha and Djurić, 2003) which also uses the Gaussian distribution to approach the posterior distribution. The difference between the local linearization PF and the Gaussian particle filter is that the latter uses a global Gaussian distribution to approximate the posterior distribution $p(x_t|y_t)$ while the former uses a local Gaussian distribution to approximate the posterior distribution of each particle $p(x_t|x_{t-1}, y_t)$. Since these filters use the latest information of y_t, the choice of the method of local linearization may improve the performance of the PF when the posterior distribution of the state can be closely approximated by a Gaussian distribution. When the variance of the state noise is large, the posterior distribution of the state is far from the Gaussian distribution in a general nonlinear system, and therefore these linearization PFs do not work well (see Section 4).

It was proposed by Gilks and Berzuini (2001) to integrate the Markov chain Monte Carlo (MCMC) techniques to the PF by choosing the particles according to a Markov chain transition kernel after the resampling step. This idea can effectively reduce the sample impoverishment in the PF. The Metropolis-Hastings algorithm can be used to construct the Markov kernel in which the invariant distribution is the posterior distribution of the state $p(x_{1:t}|y_{1:t})$. The particles are sampled from a proposal distribution and accepted with a probability calculated

by the Metropolis-Hastings algorithm, see Robert and Casella (2004). The advantage of the PF-MCMC is that it increases the diversity of the particles. But it is not easy to design a good proposal distribution in the Metropolis-Hastings algorithm.

The APF is a popular variant of the standard PF. The essential of the APF is to reserve the most possible survival particles in the simulation. The APF uses the expectation $E[x_t|x_{t-1}^i]$ to characterize $p(x_t|x_{t-1}^i)$. When the variance of the state noise is small, this method performs very well. When the variance of the state noise is large, the distribution $p(x_t|x_{t-1}^i)$ can not be characterized by $E[x_t|x_{t-1}^i]$, and it was pointed out by Arulampalam et al. (2002) that the use of the APF can degrade the performance.

To improve the performance of the PF, we may either draw more particles or use the information of y_t and let the particles move towards the region of high likelihood. When the variance of the state noise is large, the prior function contains few information to predict the future state. In this case, as pointed out by Van der Merwe et al. (2000) : "It is therefore of paramount importance to move the particles towards the regions of high likelihood".

To use the observation information, we propose a new technique that may be used for any nonlinear system described by the state-space model (1)–(2) with large state noise variance. Our method uses a two-stage sampling technique :

1. For $j = 1, \ldots, M$, draw $x_t^{i,j} \sim p(x_t|x_{t-1}^i)$ and compute the conditional likelihood $p(y_t|x_t^{i,j})$.
2. Select the particle $x_t^{i,j^\star}$ whose conditional likelihood is maximum and set $x_t^i = x_t^{i,j^\star}$.

In the first step, the particles move randomly according to the prior information like in the BF, and in the second step, the information y_t is used to select the particle with high conditional likelihood. The corresponding MBF is described in Algorithm 1.

Remark 1. In the MBF, the ID is no longer the transition density and it seems difficult to obtain. This is why we resample from the x_t^i according to the weights $p(y_t|x_t^i)$ instead of using (3). As a consequence, the weights are biased. Nevertheless, the simulations in Section 4 show that the deviation of the weights can be ignored when M is not large.

Remark 2. In the MBF, we select particles from the prior density with high conditional likelihood. Essentially, this idea is the same as the resampling which chooses particles according to their likelihood.

Remark 3. How to choose integer M is a key point in the MBF. If the distribution of the M particles $x_t^{i,j}$, $j = 1, \ldots, M$, is sparse enough in comparison with the variance of the observation noise, the conditional likelihood $p(y_t|x_t^{i,j})$ differs notably among the M particles. Thus, only few particles contributes to the estimation of the posterior distribution $p(x_t|y_{1:t})$. In this case,

Algorithm 1 Modified bootstrap filter.

Initialization, $t = 0$

for $i = 0$ to N **do**

Draw particle $x_0^i \sim p(x_0)$ and set $t = 1$

end for

for $t = 1$ to T **do**

for $i = 1$ to N **do**

for $j = 1$ to M **do**

Draw particle $x_t^{i,j} \sim p(x_t|x_{t-1}^i)$

Compute the conditional likelihood $p(y_t|x_t^{i,j})$

end for

Select $x_t^{i,j^\star}$ such that $p(y_t|x_t^{i,j^\star})$ is maximum

Set $x_t^i = x_t^{i,j^\star}$

end for

Resample particle from the x_t^i according to the weights $p(y_t|x_t^i)$

end for

selecting particle $x_t^{i,j^\star}$ and discarding the $M-1$ particles with low conditional likelihood is not detrimental to the estimation of $p(x_t|y_{1:t})$. When M is small or the variance of the state noise is large, the sparsity is satisfied and the MBF performs well. In addition, the case of high-dimensional state variables may also be favorable for the MBF.

Remark 4. The MBF is different from the APF which chooses the particles whose conditional expectation has high conditional likelihood. The MBF is used when the state noise has large variance, while the APF is used when the state noise has small variance.

Remark 5. The MBF has some resemblance with a boosted BF (BBF) (Okuma et al., 2004) in which M samples $x_t^{i,j}$, $j = 1, \ldots, M$, are drawn from $p(x_t|x_{t-1}^i)$, $i = 1, \ldots, N$. These $N \times M$ particles approximate the density $p(x_t|y_{1:t-1})$. Then N particles are resampled from these $N \times M$ particles according to their weights $p(y_t|x_t^{i,j})$. The principle of the BBF is the same as the classic BF and the ID is the transition density. The difference between the MBF and the BBF is that a selection of good particles before resampling is implemented in the MBF. We show in Section 4 that MBF outperforms the BBF for a same computation time.

4 Numerical Examples

It is difficult to compare the performances of nonlinear filters by a theoretical analysis since in general an explicit expression of the estimation error is not available. We compare the MBF

with the BF, the BBF, the PF-MCMC (PF with the MCMC resampling), the PF-EKF (PF with the EKF to produce the ID), the UPF (PF with the UKF to produce the ID) and the APF, through numerical simulations. We show that the MBF using N particles is more efficient than the BF using N particles, but the implementation time is slightly longer. When the same number of particles N and sub-sampling number M are used in the MBF and the BBF, the latter outperforms the former but the implementation time is longer. Lastly, for a same implementation time (which means using more than N particles in the BF and less than N particles in the BBF), we show that the MBF outperforms the BF and the BBF.

Three nonlinear models are considered in the following. For each model, we want to estimate the hidden states x_t for $t = 1, \ldots, T$, where T is a fixed horizon. Let $\hat{x}_t = \frac{1}{N}\sum_{i=1}^{N} x_t^i$ be the estimation of x_t obtained by the N particles x_t^i after resampling. To measure the performance of estimation of the states x_t for $t = 1, \ldots, T$, we introduce the root mean-squared error

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \|x_t - \hat{x}_t\|^2},$$

where $\|\cdot\|$ stands for the Euclidean norm. In the three examples, we take $T = 60$.

Model 1 is given by

$$x_t = 1 + \sin(w\pi(t-1)) + \frac{x_{t-1}}{2} + u_t,$$
$$y_t = \begin{cases} \frac{x_t^3}{5} + v_t, & t \leq 30, \\ \frac{x_t}{2} - 2 + v_t, & t > 30, \end{cases}$$

where $w = 4e - 2$, $u_t \sim N(0, \sigma_u^2)$ and $v_t \sim N(0, \sigma_v^2)$, where $N(a, b)$ is the normal distribution with mean a and variance b. We set $\sigma_u = 10$, $\sigma_v = \sqrt{5}$, and the initial state $x_0 \sim N(0, 5)$. The multinomial resampling is used for each filter. Each experiment is repeated 500 times independently.

First, we use 2000 particles in the seven PF and we set $M = 3$ in the MBF and the BBF. The results are listed in Table 1 where Mean and Variance denote the sample mean and the sample variance of RMSE calculated from the 500 realisations, and Time denotes the computation time of our Matlab code expressed in seconds. We see that for a same number of particles (2000), the BBF outperforms the BF, the PF-MCMC, the PF-EKF, the UPF, the APF and the MBF in terms of mean and variance of the RMSE, but uses more computation time. The MBF provides much better result than the BF but it requires more computation time. The performance of the PF-MCMC is better than the one of the BF because the diversity of the particles is increased by using MCMC, but of course the computation time is increased. The performance of the PF-MCMC is inferior to the one of the MBF, both in terms of RMSE and Time. The PF-EKF diverges and this is due to the large state noise variance. The UPF provides a better result than

Filter	Mean RMSE	Variance RMSE	Time
BF (2000)	3.60	1.15	1.43
PF-MCMC (2000)	3.53	1.11	4.52
PF-EKF (2000)	139.23	105812.03	19.28
UPF (2000)	3.41	0.68	67.42
APF (2000)	4.04	0.86	3.75
BBF (2000)	3.22	0.75	5.01
MBF (2000)	3.23	0.79	4.05

Table 1: Estimation results in Model 1 : 2000 particles, $M = 3$, multinomial resampling.

Filter	Mean RMSE	Variance RMSE	Time
BF (5000)	3.27	0.82	4.33
BBF (1700)	3.27	0.81	4.45
MBF (2000)	3.23	0.79	4.05

Table 2: Estimation results in Model 1 : different numbers of particles, $M = 3$, multinomial resampling.

the BF and the PF-MCMC with a huge computation burden. The variance of the RMSE is lower for the UPF than for the MBF, but this is the contrary for the mean of the RMSE. As mentioned above, the APF is inferior to the BF, the PF-MCMC and the UPF in the scenario of large state noise variance. The APF is valid only when the state noise variance is small.

According to Table 1, the BBF performs better than the BF and the MBF, but it requires more computation time. In Table 2, we use more particles in the BF and less particles in the BBF, and we compare the results with the MBF using 2000 particles. When 5000 particles are used in the BF, the performance of the BF is improved but it is still inferior to the one of the MBF, both in terms of RMSE and Time. When 1700 particles are used in the BBF, the performance of the BBF is reduced and it is worse than that of the MBF, both in terms of RMSE and Time. Therefore, for a same computation time, the MBF outperforms the BF and the BBF.

In Table 3, we take $M = 6$ and we give the estimation results of the BF with 8000 particles, the PF-MCMC with 4000 particles, the BBF with 2000 and 1500 particles, and the MBF with 2000 particles. We see that the MBF has better performance than the BF while using less calculation time. The BBF with 2000 particles is more precise than the MBF, but its computation time is double. Lastly, the performances in terms of RMSE of the MBF and the BBF with 1500 particles are similar, but again the BBF is more time consuming.

Filter	Mean RMSE	Variance RMSE	Time
BF (8000)	3.18	0.72	8.01
PF-MCMC (4000)	3.34	0.84	9.45
BBF (2000)	3.10	0.52	15.55
BBF (1500)	3.16	0.57	9.42
MBF (2000)	3.15	0.57	7.65

Table 3: Estimation results in Model 1 : different numbers of particles, $M = 6$, multinomial resampling.

In Figure 1, we plot the mean of the RMSE obtained with the MBF using 2000 particles and different values of M. For $3 \leq M \leq 6$, the mean of the RMSE decreases as M increases. But for $M = 7$ and $M = 10$, the RMSE increases and this may be due to the apparition of some bias in the weights when M is too large.

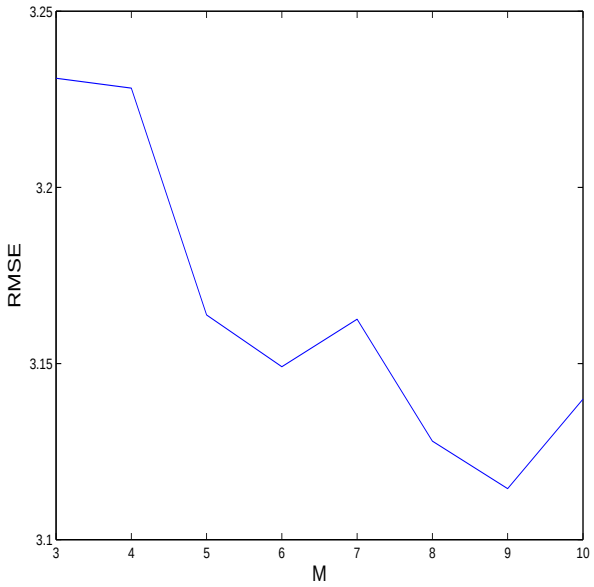

Figure 1: Sample mean of the RMSE in Model 1 for different values of M : 2000 particles in the MBF, multinomial resampling.

The computational cost of the PF depends on the resampling method. We have repeated the above simulations using the residual resampling and the systematic resampling. The results with the residual resampling and the multinomial resampling are similar, while the systematic resampling is more efficient. In Table 4, we take $M = 6$ and we give the estimation results of the BF with 2000 and 8000 particles, and the BBF and the MBF with 2000 particles, when the systematic resampling is used in each filter. Comparing the results in Tables 1 and 4 for the BF with 2000 particles, and the results in Tables 3 and 4 for the other filters, we see that a lower

Filter	Mean RMSE	Variance RMSE	Time
BF (2000)	3.39	0.76	1.38
BF (8000)	3.05	0.41	7.55
PF-MCMC (4000)	3.29	0.89	9.12
BBF (2000)	2.97	0.31	14.92
BBF (1500)	3.16	0.71	9.09
MBF (2000)	3.03	0.31	7.45

Table 4: Estimation results in Model 1 : different numbers of particles, $M = 6$, systematic resampling.

computation time is obtained for each filter with the systematic resampling. Again, the MBF outperforms the other filters when a same computation time is fixed.

We use now another nonlinear model to test the performance of our algorithm. Model 2 is given by

$$x_t = \frac{x_{t-1}}{2} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + u_t,$$
$$y_t = \frac{x_t^3}{80} + v_t,$$

where $u_t \sim N(0, 81)$, $v_t \sim N(0, 4)$ and the initial state $x_0 \sim N(0, 10)$. The multinomial resampling is used for each filter. Each experiment is repeated 500 times independently. We use 3000 and 7000 particles in the BF, 3000 particles in the PF-MCMC, the PF-EKF, the UPF, the APF, the BBF and the MBF, and we take $M = 3$. The state estimation results are listed in Table 5 and we obtain the same conclusions as for model 1 : for a fixed computation time, the MBF outperforms the other nonlinear filters.

To test the performance of the MBF in the case of a multidimensional model, a classical bearings-only tracking problem is considered now. The state vector is composed of the target's position and velocity in the plane, $x_t = (z_{1t}, z_{2t}, \dot{z}_{1t}, \dot{z}_{2t})'$, where (z_{1t}, z_{2t}) are the coordinates of the position and $(\dot{z}_{1t}, \dot{z}_{2t})$ the coordinates of the velocity. Assuming constant velocity, Model 3 is given by

$$x_t = Ax_{t-1} + Bu_t,$$
$$y_t = \arctan\left(\frac{z_{1t}}{z_{2t}}\right) + v_t,$$

Filter	Mean RMSE	Variance RMSE	Time
BF (3000)	2.26	0.74	2.32
BF (7000)	2.20	0.64	6.74
PF-MCMC (3000)	2.25	0.74	7.01
PF-EKF (3000)	23.81	597.00	29.79
UPF (3000)	2.31	0.48	103.81
APF (3000)	2.77	0.46	3.71
BBF (3000)	2.12	0.38	10.53
MBF (3000)	2.18	0.46	6.33

Table 5: Estimation results in Model 2 : different numbers of particles, $M = 3$, multinomial resampling.

where

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$u_t = (u_{1t}, u_{2t})'$ and noises u_{1t}, u_{2t}, v_t are independent and identically distributed zero-mean Gaussian sequences which are mutually independent with $\sigma^2_{u_1} = \sigma^2_{u_2} = 1$, $\sigma^2_v = 0.01$, and y_t is the sensor measurement, see e.g. Gordon et al. (1995). The residual resampling is used for each filter. Each experiment is repeated 500 times independently. We use 2000 and 6000 particles in the BF, 2000 particles in the PF-MCMC, the PF-EKF, the UPF, the APF, the BBF and the MBF, and we take $M = 3$. The results for the RMSE of the position (z_{1t}, z_{2t}) and the computation time are given in Table 6. Again, the MBF outperforms the other filters in terms of computation time.

Some conclusions can be drawn from the simulations in the three models :

- When a same number of particles is used, the MBF outperforms the BF, the PF-MCMC, the PF-EKF, the UPF and the APF in terms of mean and variance of the RMSE, but the computation time is larger than for the BF.
- When a same computation time is fixed, the MBF outperforms the BF, the BBF, the PF-MCMC, the PF-EKF, the UPF and the APF. When more particles are used in the BF, its performance increases but is less than the performance of the MBF for a fixed computation time.
- We recommend to set M less than 10.

Filter	Mean RMSE	Variance RMSE	Time
BF (2000)	0.0959	0.0790	1.85
BF (6000)	0.0657	0.0647	4.66
PF-MCMC (2000)	0.2046	0.1214	5.91
PF-EKF (2000)	0.3623	0.3299	100.2400
UPF (2000)	0.3742	0.2517	83.5500
APF (2000)	0.1745	0.1074	3.38
BBF (2000)	0.0659	0.0621	5.08
MBF (2000)	0.0752	0.0711	3.65

Table 6: Estimation results in Model 3 : different numbers of particles, $M = 3$, residual resampling.

- Their is no significant long-term effect of the approximation of the weights in the MBF.

5 Conclusion

We have proposed a modification of the standard PF for nonlinear filtering in the scenario of a large state noise variance. The idea is to select particles with high conditional likelihood. Our algorithm outperforms the PF with a same computational complexity. In the future, it will be interesting to investigate how to choose the number M of sub-sampling adaptively.

Acknowledgements

The authors are grateful to two anonymous referees for their helpful comments.

References

S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filter for on-line nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.*, 50(2):174–188, 2002.

F. Campillo and V. Rossi. Convolution particle filter for parameter estimation in general state-space models. *IEEE Transactions on Aerospace and Electronic Systems*, 45(3):1063–1072, 2009.

O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proceedings*, 95(5):899–924, 2007.

B. P. Carlin, N. G. Polson, and D. S. Stoffer. A Monte Carlo approach to nonnormal and nonlinear state space modeling. *J. Amer. Statist. Assoc.*, 87(418):493–500, 1992.

Z. Chen. *Bayesian filtering: from Kalman filters to particle filters and beyond*. Communication Research laboratory, McMaster University, 2003.

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

A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer-Verlag, New York, 2001.

A. Gelb, editor. *Applied optimal estimation*. The MIT Press, Cambridge, 1974.

W. Gilks and C. Berzuini. Following a moving target - Bayesian inference for dynamic Bayesian models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 63(1):127–146, 2001.

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

N. J. Gordon, D. J. Salmond, and C. Ewing. Bayesian state estimation for tracking and guidance using the bootstrap filter. *Journal of Guidance, Control, and Dynamics*, 18(6):1434–1443, 1995.

C. Hue, J.-P. Le Cadre, and P. Perez. Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 38(3):791–812, 2002.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, Series D, J. Basic Eng.*, 82:35–45, 1960.

G. Kitagawa. Non-Gaussian state-space modeling of nonstationary time series. *J. Amer. Statist. Assoc.*, 82(400):1032–1063, 1987.

J. H. Kotecha and Djurić. Gaussian particle filter. *IEEE Trans. Signal Process.*, 51(10):2592–2601, 2003.

S. C. Kramer and H. W. Sorenson. Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, 24(6):789–801, 1988.

J. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *J. Amer. Statist. Assoc.*, 93(443):1032–1044, 1998.

C. J. Masreliez. Approximate non-Gaussian filtering with linear state and observation relations. *IEEE Trans. Automat. Control*, 20(1):107–110, 1975.

K. Okuma, A. Taleghani, N. De Freitas, J. Little, and D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proceedings of European Conference on Computer Vision*, Prague, Czech, 2004. IEEE.

M. K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. *J. Amer. Statist. Assoc.*, 94(446):590–599, 1999.

K. Reif, S. Gunther, E. Yaz, and R. Unbehauen. Stochastic stability of the discrete-time extended Kalman filter. *IEEE Trans. Automat. Control*, 44(4):714–728, 1999.

C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer-Verlag, New York, second edition, 2004.

H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using gaussian sums. *Automatica*, 7(4):465–479, 1971.

J. C. Spall, editor. *Bayesian Analysis of Time Series and Dynamic Models*. Marcel Dekker, New York, 1988.

H. Tanizaki. *Nonlinear Filters: Estimation and Applications*. PhD thesis, University of Pennsylvania, 1991.

H. Tanizaki. *Nonlinear Filters: Estimation and Applications*. Springer-Verlag, New York, second edition, 1996.

M. Šimandl and O. Straka. Sampling densities of particle filter: a survey and comparison. In *Proceedings of the American Control Conference, New York, USA*, 2007.

R. Van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, Oregon Health and Science University, 2004.

R. Van der Merwe, N. de Freitas, A. Doucet, and E. Wan. The unscented particle filter. In *Advances in Neural Information Processing Systems*, Nov 2000.

M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models*. Springer-Verlag, New York, second edition, 1997.