Nonlinear Processes in Geophysics

Merging particle filter for sequential data assimilation

S. Nakano[1,2], G. Ueno[1,2], and T. Higuchi[1,2]

[1]The Institute of Statistical Mathematics, Research Organization of Information and Systems, Japan
[2]Japan Science and Technology Agency, Japan

Abstract. A new filtering technique for sequential data assimilation, the merging particle filter (MPF), is proposed. The MPF is devised to avoid the degeneration problem, which is inevitable in the particle filter (PF), without prohibitive computational cost. In addition, it is applicable to cases in which a nonlinear relationship exists between a state and observed data where the application of the ensemble Kalman filter (EnKF) is not effectual. In the MPF, the filtering procedure is performed based on sampling of a forecast ensemble as in the PF. However, unlike the PF, each member of a filtered ensemble is generated by merging multiple samples from the forecast ensemble such that the mean and covariance of the filtered distribution are approximately preserved. This merging of multiple samples allows the degeneration problem to be avoided. In the present study, the newly proposed MPF technique is introduced, and its performance is demonstrated experimentally.

1 Introduction

Data assimilation is performed to obtain the best estimates of a state of a dynamic system or the evolution of a system by incorporating observation into a model of the system and is used as an important tool for modeling and prediction of geophysical processes. Data assimilation methods are classified into two categories: variational data assimilation and sequential data assimilation. While variational data assimilation is performed by fitting a dynamic model to all of the available observations during a period of interest, sequential data assimilation is an on-line approach that updates the estimation of a state at each observation time. In the present study, we focus on sequential data assimilation.

Correspondence to: S. Nakano
(shiny@ism.ac.jp)

Most sequential data assimilation techniques basically consider a probability density function (PDF) of a state of a dynamic system. An assimilation process is based on a prior PDF of the current state which is obtained using past data and a system model. This prior PDF is then updated to obtain the posterior PDF of the state by incorporating constraints based on observation. The procedure used to obtain the posterior PDF is called "filtering". The filtering procedure provides a PDF of the current state considering current and past observations, which should be a basis for accurate prediction of future states.

If a PDF of a state is Gaussian and the dynamics of the system is linear, then a filtering process can be described by the algorithm of the Kalman filter. However, since geophysical systems usually contain inherent nonlinearity, it is rare that the Kalman filter can be applied. The Kalman filter algorithm is sometimes extended by modifying the calculation of covariances of a state by linearizing a system model, and this extended algorithm is called the extended Kalman filter (EKF). However, for models with high nonlinearity, the EKF can make errors diverge (e.g., Evensen, 1992). Moreover, for a model with a large number of variables, the EKF requires a high computational cost. Although the computational cost could be reduced by using a variant of the EKF, the singular evolutive extended Kalman (SEEK) filter (Pham et al., 1998b), the SEEK filter also requires the linearization of a system model and it can provide an unstable result for cases with high nonlinearity.

In order to apply data assimilation to a system with nonlinear dynamics, it is practical to approximate a PDF of a state by an ensemble consisting of many realizations called "particles". The ensemble Kalman filter (EnKF) (Evensen, 1994; Burgers et al., 1998) is one of such methods, and several variants of this algorithm have also been proposed (e.g., Anderson, 2001; Whitaker and Hamill, 2002). The EnKF is applicable to data assimilation of nonlinear systems. In the EnKF, each particle in an ensemble is updated using a

Kalman gain calculated from the mean and the covariances of the prior ensemble. However, the EnKF basically assumes a linear relationship between a state and observed data in calculating a Kalman gain. Therefore, the EnKF does not provide good estimates of a state for cases in which linear approximation of the relationship between a state and observed data is invalid. In addition, the computational cost of each filtering step in the EnKF is large due to repetitive multiplications and additions of matrices. Pham et al. (1998a) have proposed another ensemble-based filtering method, the singular evolutive interpolated Kalman (SEIK) filter, which is derived as a variant of the SEEK filter. Although the SEIK filter can work more efficiently than the EnKF (Nerger et al., 2005), it is not applicable to cases with nonlinear observation as well.

The particle filter (PF) (Gordon et al., 1993; Kitagawa, 1993, 1996; Kitagawa and Gersch, 1996; Higuchi and Kitagawa, 2000; van Leeuwen, 2003), which is sometimes referred to as the sequential importance resampling (SIR) filter, is another method that is based on ensemble approximation of a PDF. In the PF, an estimation of a posterior PDF is obtained by resampling with replacement from a prior ensemble. As the PF does not require assumptions of linearity or Gaussianness, it is applicable to general nonlinear problems. In particular, the PF can be applied to cases in which the relationship between a state and observed data is nonlinear, to which the application of the ensemble Kalman filter (EnKF) is not appropriate. However, the PF often encounters a problem called "degeneration", which does not occur in the EnKF. Since resampling procedures are applied recursively, most of the particles are replaced by particles that fit the observed data better, and the posterior PDF is eventually represented by only a few of the particles among the members of the initial ensemble. This reduces the validity of ensemble approximation. This problem could be avoided by increasing the number of particles in the ensemble. However, in order to increase the number of particles, a prohibitive computational cost is often required at each forecast step.

One potential way to avoid the degeneration problem is to approximate a posterior distribution as a Gaussian distribution. This approach has been proposed by Kotecha and Djurić (2003) under the name of the Gaussian particle filter (GPF), and a similar algorithm was also presented by Anderson and Anderson (1999). In this technique, from an ensemble that represents a filtered posterior distribution, the mean and covariances are calculated to obtain a Gaussian distribution for approximating the filtered distribution. By drawing random samples from this Gaussian distribution, a filtered ensemble is newly generated. In the GPF, although the accuracy of an approximation of a filtered distribution is worse than in the PF because of the assumption of Gaussianness, no duplicate particles are contained in the ensemble and degeneration does not occur. However, in generating Gaussian random vectors to make a Gaussian ensemble, we must factorize the covariance matrix, which requires a high computa-

tional cost if the dimension of a state vector is large. In most practical cases, factorization of the covariance matrix with the dimension of a state vector is not realistic.

There is another way to avoid degeneration which is a variant of the PF referred to as the kernel filter (Hürzeler and Künsch, 1998; Anderson and Anderson, 1999) or the regularized particle filter (Musso et al., 2001). This technique approximates the filtered PDF by a sum of Gaussian functions with small standard deviations centered at the particle locations, and members of a filtered ensemble is drawn from the sum of Gaussian functions. However, in applying this technique to high-dimensional models, there is difficulty in designing a covariance matrix for each of the Gaussian functions. Although a covariance matrix could be made on the basis of the covariance matrix of an ensemble representing a prior or posterior PDF, this bring the same problem as the GPF; that is, the factorization of the covariance matrix is required and the computational cost would become prohibitive in cases that a state vector is high-dimensional.

Thus, there exists no practical method to allow sequential data assimilation with acceptable computational cost, except some methods such as the EnKF and the SEIK filter which also have a disadvantage in that it is not necessarily applicable to cases with nonlinear observations. To overcome this problem, another technique, the merging particle filter (MPF), is devised. The MPF is an improved algorithm of the PF, in which filtering is performed by merging several particles of a prior ensemble, which is rather similar to the genetic algorithm (e.g., Goldberg, 1989). This merging procedure allows the degeneration problem to be avoided and requires far fewer particles than the PF. The primary advantage of the PF over the EnKF is inherited; that is, the MPF is applicable even to cases in which the relationship between a state and observed data is nonlinear. Moreover, since the MPF does not require the calculation of an inverse matrix, the computational cost at each filtering step is lower than that of the EnKF. The PF algorithm, which the proposed algorithm is based on, is reviewed in Sect. 2, and the MPF algorithm is introduced in Sect. 3. In order to evaluate the performance of the MPF, the results of a number of experiments are described in Sect. 4. Finally, the effectiveness of the MPF is discussed and summarized in Sect. 5.

2 Particle filter

The following state space model is considered:

$$\boldsymbol{x}_k = F_k(\boldsymbol{x}_{k-1}, \boldsymbol{v}_k) \tag{1a}$$
$$\boldsymbol{y}_k = H_k(\boldsymbol{x}_k) + \boldsymbol{w}_k \tag{1b}$$

where the vectors $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ indicate the state of a system and observed data at a discrete time $T=t_k$ ($k=1, \ldots$), respectively, and the vectors $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ denote system noise and observation noise, respectively. The operator F_k represents

the temporal evolution of a state from time t_{k-1} to time t_k according to the system model based on the simulation, while H_k projects the state vector $\boldsymbol{x}_k$ to the observation space.

The PF considers a PDF of a state $\boldsymbol{x}_k$, and the PDF is approximated by an ensemble consisting of a large number of discrete samples called 'particles'. For example, a filtered distribution at time $T = t_{k-1}$, $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$, is approximated by particles $\{\boldsymbol{x}_{k-1|k-1}^{(1)}, \boldsymbol{x}_{k-1|k-1}^{(2)}, \cdots, \boldsymbol{x}_{k-1|k-1}^{(N)}\}$ as

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta\left(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1|k-1}^{(i)}\right) \tag{2}$$

where δ is Dirac's delta function, and N is the number of particles in the ensemble. Here we expressed $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{k-1})$ as $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$. From this ensemble approximation of $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{1:k-1})$, we obtain an ensemble approximation of the forecast distribution of the state at the next observation time $T = t_k$ as

$$p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_{k|k-1}^{(i)}\right). \tag{3}$$

Each particle of the forecast ensemble $\boldsymbol{x}_{k|k-1}^{(i)}$ is given by $F_k(\boldsymbol{x}_{k-1|k-1}^{(i)}, \boldsymbol{v}_k^{(i)})$ where $\boldsymbol{v}_k^{(i)}$ is a realization of the system noise. This procedure is called the forecast step.

From the forecast distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})$ and observed data $\boldsymbol{y}_k$, we obtain a filtered PDF $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ by using Bayes' theorem, as follows:

$$
\begin{aligned}
&p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \\
&= \frac{p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})\, p(\boldsymbol{y}_k|\boldsymbol{x}_k)}{\int p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1})\, p(\boldsymbol{y}_k|\boldsymbol{x}_k) d\boldsymbol{x}_k} \\
&\approx \frac{1}{\sum_j p\left(\boldsymbol{y}_k|\boldsymbol{x}_{k|k-1}^{(j)}\right)} \sum_{i=1}^{N} p\left(\boldsymbol{y}_k|\boldsymbol{x}_{k|k-1}^{(i)}\right) \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_{k|k-1}^{(i)}\right) \\
&= \sum_{i=1}^{N} w_i \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_{k|k-1}^{(i)}\right)
\end{aligned}
$$

$$\tag{4}$$

where $p(\boldsymbol{y}_k|\boldsymbol{x}_{k|k-1}^{(i)})$ is the likelihood of $\boldsymbol{x}_{k|k-1}^{(i)}$ given the data $\boldsymbol{y}_k$ and the weight w_i is defined as

$$w_i = \frac{p(\boldsymbol{y}_k|\boldsymbol{x}_{k|k-1}^{(i)})}{\sum_j p(\boldsymbol{y}_k|\boldsymbol{x}_{k|k-1}^{(j)})}. \tag{5}$$

This is called the filtering step.

Equation (4) shows that $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ is approximated using particles weighted by w_i. Based on Eq. (4), we obtain a new ensemble $\{\boldsymbol{x}_{k|k}^{(1)}, \cdots, \boldsymbol{x}_{k|k}^{(N)}\}$ which approximates $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ by resampling the forecast ensemble $\{\boldsymbol{x}_{k|k-1}^{(1)}, \cdots, \boldsymbol{x}_{k|k-1}^{(N)}\}$ with a weight of w_i for each i. The new ensemble may

contain multiple copies of $\boldsymbol{x}_{k|k-1}^{(i)}$ belonging to the forecast ensemble, and the number of copies m_i becomes

$$m_i \approx N w_i \quad \left(\sum m_i = N; \ m_i \geq 0\right) \tag{6}$$

for each $\boldsymbol{x}_{k|k-1}^{(i)}$. From Eqs. (4) and (6), we obtain an approximation of $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ using uniformly weighted particles, as follows:

$$
\begin{aligned}
p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) &\approx \sum_{i=1}^{N} w_i \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_{k|k-1}^{(i)}\right) \\
&\approx \sum_{i=1}^{N} \frac{m_i}{N} \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_{k|k-1}^{(i)}\right) \\
&= \frac{1}{N} \sum_{i=1}^{N} \delta\left(\boldsymbol{x}_k - \boldsymbol{x}_{k|k}^{(i)}\right).
\end{aligned}
\tag{7}
$$

Thus, the newly generated ensemble approximates the filtered PDF $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$. Equation (7) has the same form as Eq. (2), which allows us to recursively repeat the above procedure from Eq. (2) to Eq. (7). By repeating the procedure, a sequence of observed data is incorporated into the system model.

3 Merging particle filter

In the PF, a filtered ensemble generated through the resampling procedure contains multiple copies of particles with high likelihoods, and particles with low likelihoods are removed from the ensemble. Therefore, after repeating resampling several times, the diversity of the ensemble decreases and eventually becomes insufficient for validly representing a PDF. This problem can be avoided by increasing the number of particles. However, due to limited computational resources, it is often impossible to use a sufficient number of particles to repeat resampling several times. The MPF, which we propose in this section, allows us to remake a filtered ensemble while restraining the reduction of its diversity.

The MPF is a modification of the PF. In the MPF, a filtered ensemble is constructed based on samples from a forecast ensemble as in the PF. However, each particle of a filtered ensemble is generated as an amalgamation of multiple particles from the forecast ensemble, which is rather similar to the genetic algorithm. Although this does not ensure that the shape of the filtered PDF is preserved, the mean and covariance of the filtered PDF are approximately preserved (asymptotically preserved as the number of particles approaches infinity) in generating a filtered ensemble.

A filtered ensemble is obtained as follows. When the number of particles to be merged is assumed to be n, we draw $n \times N$ samples from the forecast ensemble with weights of w_i in Eq. (5), and we thus obtain an ensemble: $\{\hat{\boldsymbol{x}}_{k|k}^{(1,1)}, \cdots, \hat{\boldsymbol{x}}_{k|k}^{(n,1)}, \cdots, \hat{\boldsymbol{x}}_{k|k}^{(1,N)}, \cdots, \hat{\boldsymbol{x}}_{k|k}^{(n,N)}\}$. A subset

$\{\hat{x}_{k|k}^{(j,1)}, \cdots, \hat{x}_{k|k}^{(j,N)}\}$ from the $n \times N$ samples forms an ensemble approximating the filtered PDF, which satisfies

$$p(x_k|y_{1:k}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta\left(x_k - \hat{x}_{k|k}^{(j,i)}\right) \qquad (8)$$

because it consists of N samples drawn from the forecast ensemble with weights of w_i, as was the case in obtaining the filtered ensemble in the previous section. Next, we make a new ensemble consisting of N particles $\{x_{k|k}^{(1)}, \cdots, x_{k|k}^{(N)}\}$ to approximate $p(x_k|y_{1:k})$. Each particle in the new ensemble is generated as a weighted sum of n samples from the $n \times N$ sample set as:

$$x_{k|k}^{(i)} = \sum_{j=1}^{n} \alpha_j \hat{x}_{k|k}^{(j,i)}. \qquad (9)$$

In order to ensure that the newly generated ensemble preserves the mean and covariances of the filtered PDF for $N \to \infty$, the merging weights α_j are set to satisfy

$$\sum_{j=1}^{n} \alpha_j = 1 \qquad (10a)$$

$$\sum_{j=1}^{n} \alpha_j^2 = 1 \qquad (10b)$$

where each α_j is a real number. When the merging weights satisfy Eq. (10a), the mean of the PDF approximated by the new ensemble $\{x_{k|k}^{(1)}, \cdots, x_{k|k}^{(N)}\}$ becomes

$$\int x_k \frac{1}{N} \sum_{i=1}^{N} \delta\left(x_k - x_{k|k}^{(i)}\right) dx_k$$

$$= \frac{1}{N} \sum_{i=1}^{N} x_{k|k}^{(i)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{n} \alpha_j \hat{x}_{k|k}^{(j,i)}$$

$$= \sum_{j=1}^{n} \left[\alpha_j \int x_k \frac{1}{N} \sum_{i=1}^{N} \delta\left(x_k - \hat{x}_{k|k}^{(j,i)}\right) dx_k \right] \qquad (11)$$

$$\approx \sum_{j=1}^{n} \alpha_j \int x_k \, p(x_k|y_{1:k}) \, dx_k$$

$$= \int x_k \, p(x_k|y_{1:k}) \, dx_k = \mu_{k|k}$$

where $\mu_{k|k}$ is the mean of the filtered PDF $p(x_k|y_{1:k})$. In addition, if the merging weights α_j satisfy Eq. (10b), the co-

variances given by the new ensemble become

$$\int (x_k - \mu_{k|k})(x_k - \mu_{k|k})^T \frac{1}{N} \sum_{i=1}^{N} \delta\left(x_k - x_{k|k}^{(i)}\right) dx_k$$

$$= \frac{1}{N} \sum_{i=1}^{N} (x_{k|k}^{(i)} - \mu_{k|k})(x_{k|k}^{(i)} - \mu_{k|k})^T$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{j_1=1}^{n} \alpha_{j_1} \hat{x}_{k|k}^{(j_1,i)} - \mu_{k|k} \right) \left(\sum_{j_2=1}^{n} \alpha_{j_2} \hat{x}_{k|k}^{(j_2,i)} - \mu_{k|k} \right)^T$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left[\sum_{j_1=1}^{n} \alpha_{j_1} \left(\hat{x}_{k|k}^{(j_1,i)} - \mu_{k|k} \right) \right] \left[\sum_{j_2=1}^{n} \alpha_{j_2} \left(\hat{x}_{k|k}^{(j_2,i)} - \mu_{k|k} \right) \right]^T$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{n} \alpha_j^2 \left(\hat{x}_{k|k}^{(j,i)} - \mu_{k|k} \right) \left(\hat{x}_{k|k}^{(j,i)} - \mu_{k|k} \right)^T$$

$$= \sum_{j=1}^{n} \alpha_j^2 \int (x_k - \mu_{k|k})(x_k - \mu_{k|k})^T \frac{1}{N} \sum_{i=1}^{N} \delta\left(x_k - \hat{x}_{k|k}^{(j,i)}\right) dx_k$$

$$\approx \int (x_k - \mu_{k|k})(x_k - \mu_{k|k})^T p(x_k|y_{1:k}) \, dx_k = \Sigma_{k|k}$$

$$(12)$$

where $\Sigma_{k|k}$ is the covariance matrix of $p(x_k|y_{1:k})$. Here, we used an approximation as

$$\frac{1}{N} \sum_{i=1}^{N} (\hat{x}_{k|k}^{(j_1,i)} - \mu_{k|k})(\hat{x}_{k|k}^{(j_2,i)} - \mu_{k|k})^T \approx 0 \quad (\text{if } j_1 \neq j_2),$$

which is justified because the two sets of samples $\{\hat{x}_{k|k}^{(j_1,1)}, \cdots, \hat{x}_{k|k}^{(j_1,N)}\}$ and $\{\hat{x}_{k|k}^{(j_2,1)}, \cdots, \hat{x}_{k|k}^{(j_2,N)}\}$ are obtained through independent random sampling and would not correlate with each other. Therefore, the ensemble obtained using Eq. (9) affords an approximation of $p(x_k|y_{1:k})$ preserving the mean and covariances as

$$p(x_k|y_{1:k}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta\left(x_k - x_{k|k}^{(i)}\right). \qquad (13)$$

The number of merged particles n can be chosen almost arbitrarily. However, in order that the merging procedure makes sense, n must be equal to or greater than 3. If $n=1$, the weight α_1 must be 1 in order to satisfy both Eqs. (10a) and (10b), which is obviously equivalent to the normal PF. If $n=2$, then one of merging weights must be 1, and the other must be 0, so as to satisfy both Eqs. (10a) and (10b). This setting is also equivalent to the normal PF, which means that the merging procedure does not make sense. Although there is no upper limit for n, it is not necessary to set n to be large. As shown in the next section, if none of the merging weights are zero, we would greatly benefit by the merging procedure even when n is as small as 3.

When n is equal to or greater than 3, there are infinite allowable sets of the merging weights: $\{\alpha_1, \cdots, \alpha_n\}$. Although there is no definitive way to determine the values of
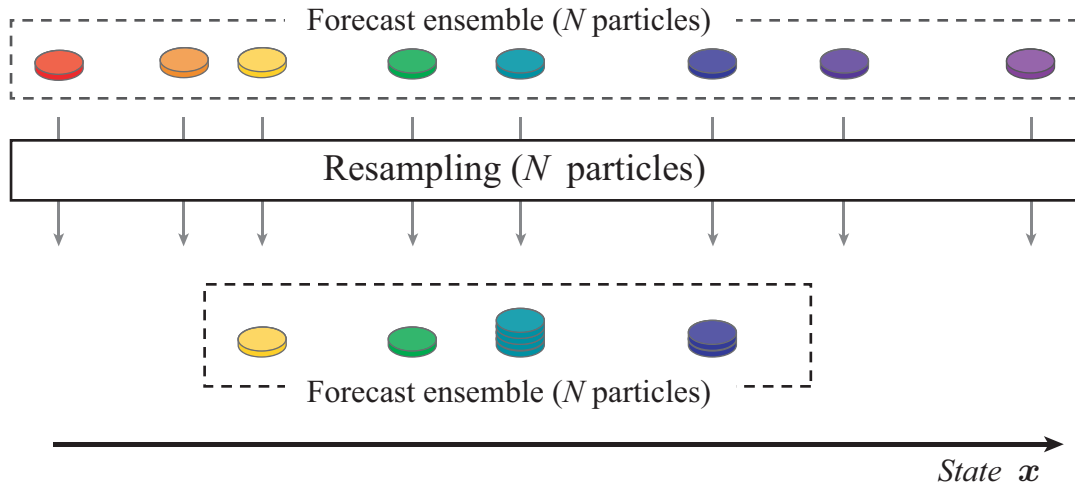
Fig. 1. PF scheme. The value of a state x is on the horizontal axis assuming that the state x is scalar.

the weights, it would be preferable to set them such that no two weights are equal to each other and that none of the weights become zero in order to reinforce the diversity of the filtered ensemble. Under this setting, two duplicate particles in the filtered ensemble $\{x_{k|k}^{(1)}, \cdots, x_{k|k}^{(N)}\}$ can be generated only from two identical sets of n merged particles drawn from the forecast ensemble $\{x_{k|k-1}^{(1)}, \cdots, x_{k|k-1}^{(N)}\}$, if duplicate particles are not contained in the forecast ensemble. When the probability that particle $x_{k|k-1}^{(i)}$ is drawn from the forecast ensemble is w_i ($0 \leq w_i < 1$), the probability that a sequence of n particles $\{x_{k|k-1}^{(i_1)}, \cdots, x_{k|k-1}^{(i_n)}\}$ is drawn is $\prod_{j=1}^{n} w_{i_j}$. Since $\prod_{j=1}^{n} w_{i_j} \leq (\max w_i)^n$, the number of duplicate particles contained in the filtered ensemble is, at most, approximately $N \times (\max w_i)^n$ for the MPF, while it is $N \times \max w_i$ for the PF.

Figures 1 and 2 show schematically the respective procedures of the PF and the MPF when the number of merging particles is set to be 3. In the PF, a filtered ensemble is simply obtained by resampling. In the MPF with 3 merging particles, after $3N$ particles are sampled from the forecast ensemble, the $3N$ particles are divided into N combinations of 3 particles, and the 3 particles in each combination are merged to obtain a new particle. Even from combinations of the same 3 particles, different particles can be made with different sets of weights. Thus, the filtered ensemble obtained with the MPF contains diverse particles in comparison with that obtained with the PF.

4 Numerical experiments

4.1 Lorenz 63 model

We performed a numerical experiment to test the MPF. Although this method is actually devised for data assimilation for high-dimensional models, we first used a simple model, the Lorenz 63 model (Lorenz, 1963), to investigate the behaviors of the method. The Lorenz 63 model is described by the following equations:

$$\frac{dx}{dt} = -s(x - y) \tag{14a}$$

$$\frac{dy}{dt} = rx - y - xz \tag{14b}$$

$$\frac{dz}{dt} = xy - bz. \tag{14c}$$

In the conventional parameter setting, the three parameters are set as follows: $s=10$, $r=28$, and $b=8/3$. One time step in integrating the system equation was set to be 0.01.

Initially, we ran this model to generate a sequence of measurement data for this test. The data were generated every 20 time step with errors of a standard deviation of 2.0. It was assumed that all of the components of the state vector, x, y, and z, could be observed. In this situation, the observation vector at each observation time resides in the same vector space as the state vector.

The generated data were assimilated into the model using the PF and the MPF. In this and the following experiments, we assume additive system noise, and thus Eqs. (1a) and (1b) are rewritten as follows.

$$x_k = F(x_{k-1}) + v_k \tag{15a}$$
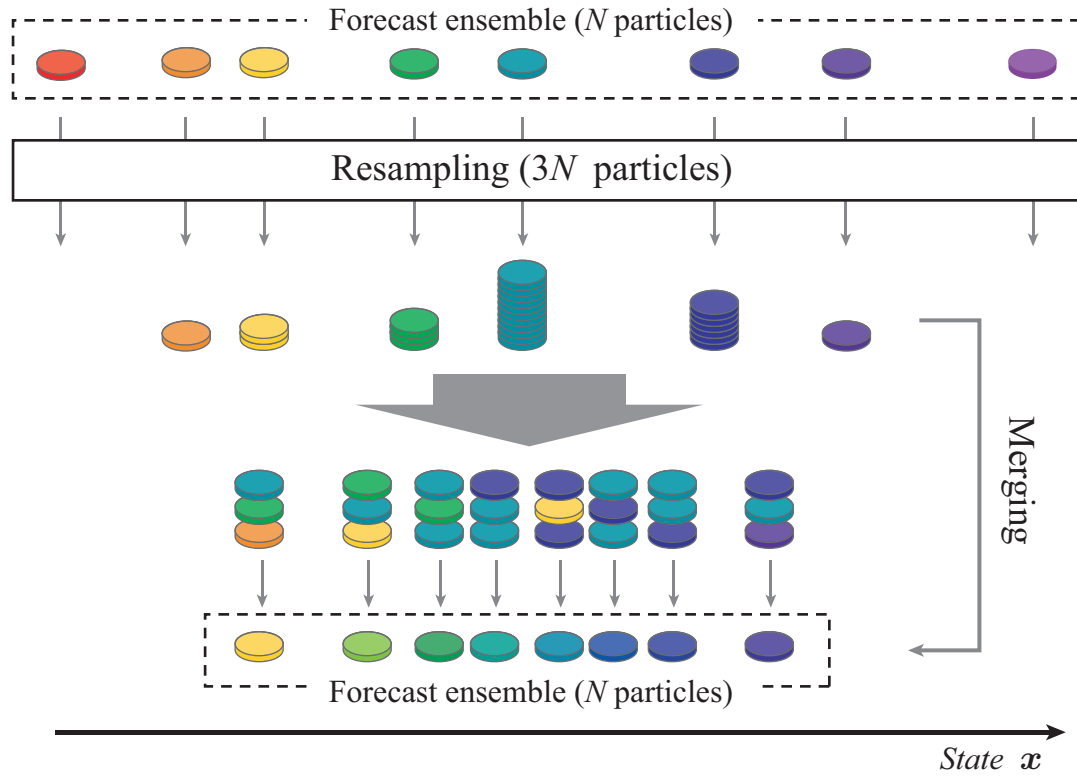
$$y_k = H(x_k) + w_k \tag{15b}$$

Fig. 2. Scheme of the MPF, in which the number of merging particles is set to be 3. The value of a state x is on the horizontal axis assuming that the state x is scalar.

where the subscript k in F_k and H_k is omitted because the system and observation models considered here are time-independent. In applying the MPF, the number of merged particles was set to $n=3$, and the weights α_j were set as follows:

$$\alpha_1 = \frac{3}{4} \tag{16a}$$

$$\alpha_2 = \frac{\sqrt{13}+1}{8} \tag{16b}$$

$$\alpha_3 = -\frac{\sqrt{13}-1}{8} \tag{16c}$$

which satisfies Eqs. (10a) and (10b). In both the PF and the MPF, we need to calculate the likelihood $p(\boldsymbol{y}_k|\boldsymbol{x}_k)$ where $\boldsymbol{y}_k$ is the observation vector (x_k^o, y_k^o, z_k^o), and $\boldsymbol{x}_k$ is the state vector (x_k, y_k, z_k) at time $T=t_k$. Assuming that observation noise $\boldsymbol{w}_k$ obeys a Gaussian distribution with zero mean and a diagonal covariance as diag$(\sigma^2, \sigma^2, \sigma^2)$, the likelihood becomes

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{||\boldsymbol{y}_k - \boldsymbol{x}_k||^2}{2\sigma^2}\right] \tag{17}$$

where we set $\sigma=3$. The system noise was assumed to be a Gaussian noise with zero mean and a diagonal covariance as

diag$(0.01, 0.01, 0.01)$. Particles of the forecast ensemble at the initial time step $(T=t_1)$ were generated from a Gaussian distribution where the mean was given by the value of the data at the same time step and the standard deviation was 4.0 for each component.

Figure 3 shows the x-component of the state vector $\boldsymbol{x}_k$ as estimated by the MPF, where the number of particles was set to $N=64$, and Fig. 4 shows that estimated by the PF, where the number of particles was also set to $N=64$. Here, the estimate was given by the average over the ensemble members. In each figure, the black squares indicate the test data that were assimilated into the model, and the red line indicates the true trajectory of the state. Finally, the blue line indicates the state estimated through data assimilation. As seen in these figures, the MPF successfully estimated the state, while the estimate by the PF largely deviated from the true state after around time step 6360. Figures 5 and 6 show the same data as shown in Figs. 3 and 4, respectively, but are focused on the period from time step 6000 to time step 7000. While the true state began to decrease after time step 6360, the estimate by the PF began to increase, and the PF failed to trace the true trajectory thereafter. Estimates by the MPF also increased after time step 6360. However, this result was improved by the filtering at time step 6380, after which the MPF again successfully traced again the true state.
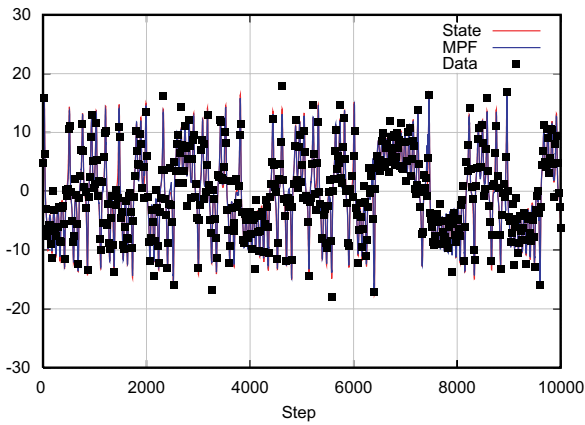
Fig. 3. Result of the experiment of data assimilation by the MPF for the Lorenz 63 model. The number of particles was set to $N=64$. The black squares indicate the test data that were assimilated into the model. The red line indicates the true state of x, and the blue line indicates the estimation of x as a result of the data assimilation.
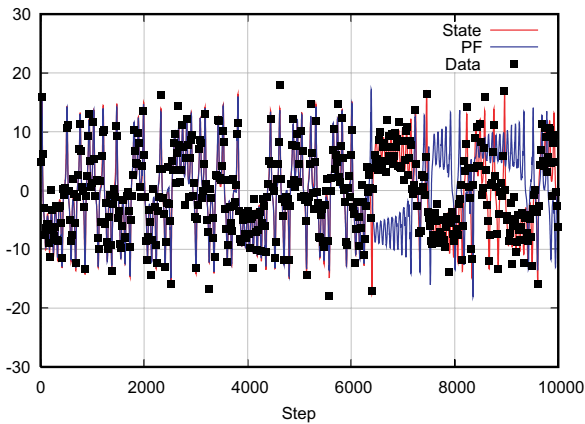

Fig. 5. Result of the experiment of data assimilation by the MPF for the Lorenz 63 model from time step 6000 to time step 7000 for the x-component.

Fig. 4. Result of the experiment of data assimilation by the PF for the Lorenz 63 model. The number of particles was set to $N=64$. The black squares indicate the test data that were assimilated into the model. The red line indicates the true state of x and the blue line indicates the estimation of x as a result of the data assimilation.

Fig. 6. Result of the experiment of data assimilation by the PF for the Lorenz 63 model from time step 6000 to time step 7000 for the x-component.

In order to clarify why the PF failed to trace the true trajectory, histograms of the ensemble for x around time step 6360 are shown in Fig. 7. At time step 6340, a gap appeared around $-1<x<0$ in the filtered ensemble in the result by the PF. This gap expanded remarkably at the next forecast step, resulting in a large gap in the forecast ensemble at time step 6360. While the x value of the true state was -0.125 at this time step, as indicated by the dashed line in each panel, no members of the ensemble were distributed around the true state. In contrast, no distinct gap appeared in the filtered ensemble at time step 6340 in the result obtained by the MPF. Thus, there were only small gaps in the forecast ensemble at the next time step.
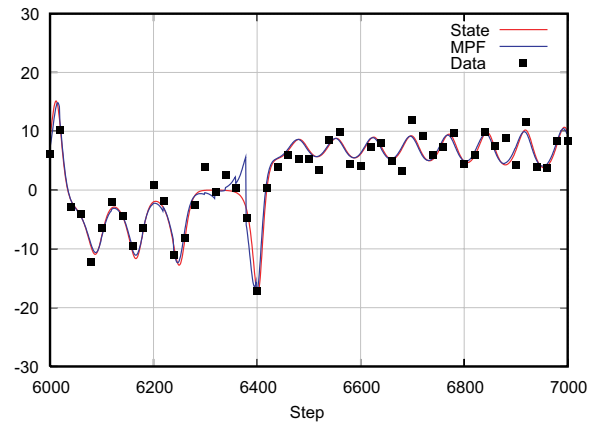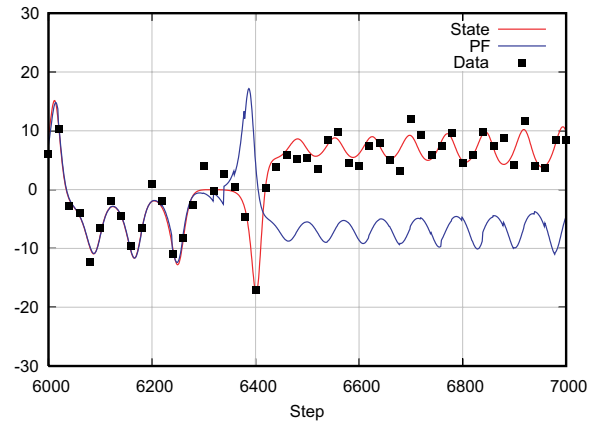
We conducted experiments with various numbers of particles using the PF and the MPF. Table 1 shows the root-mean-square of deviations from the true state over 50 000 time steps for all of the components for each experiment. In this table, the results obtained using the EnKF (Evensen, 1994; Burgers et al., 1998), which is widely used for data assimilation, are also displayed for reference. Even if the number of particles was increased into 128, the PF provided a worse estimation than the MPF. When the number of particles was set to $N=256$, the estimates by the PF became as good as those by the MPF. Although Kivman (2003) has pointed out that the PF tends to provide better estimations than the EnKF, this table shows that the EnKF yields lower errors when the number of particles is small. However, even in such cases, the MPF gives better estimations than the EnKF.
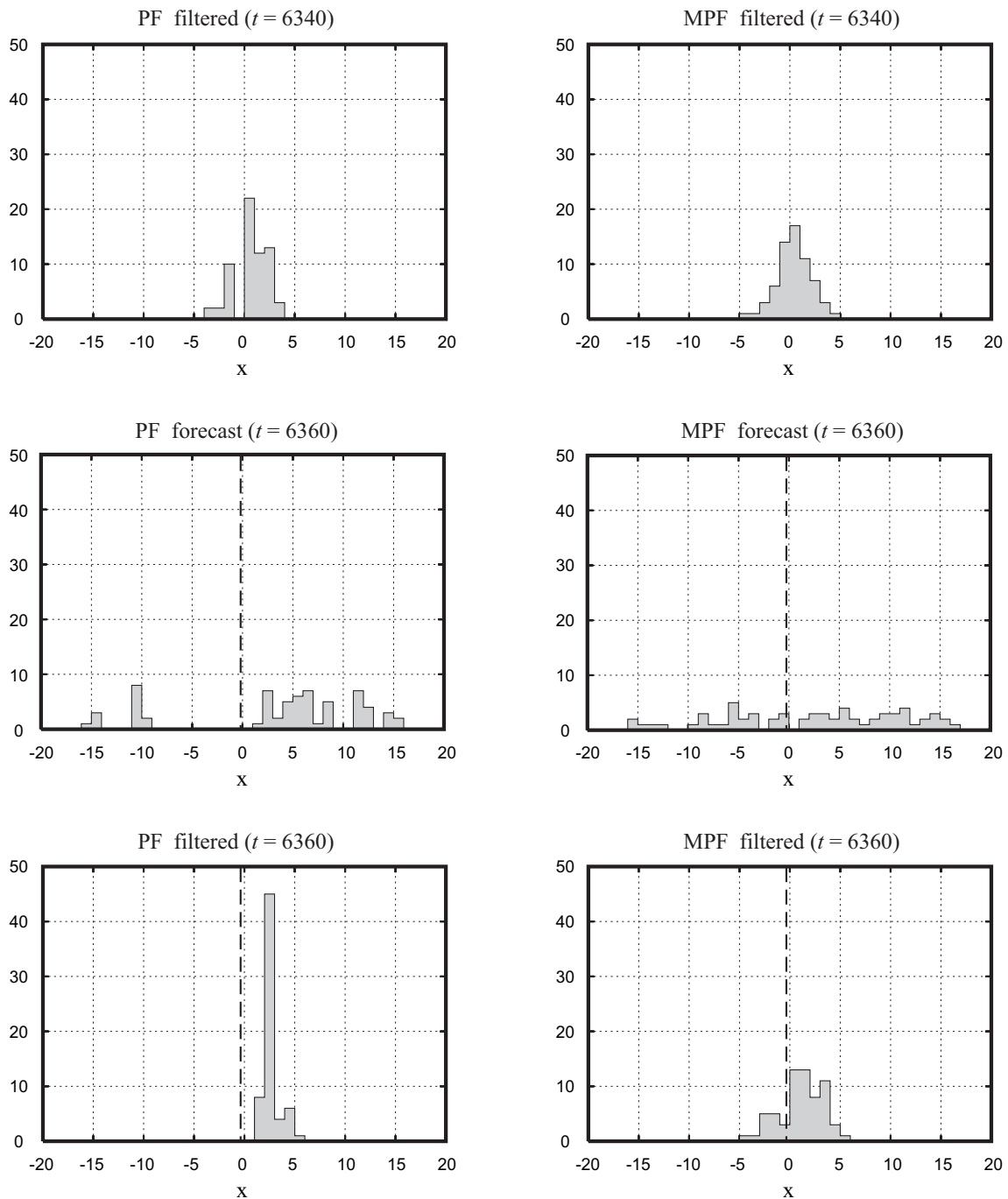
Fig. 7. Histograms of the distribution of x in the ensemble around time step 6360. The left-hand panels show the distributions for the results obtained by the PF, and the right-hand panels show the distributions for the results obtained by the MPF. The upper panels show the filtered distributions at time step 6340. The middle panels show the forecast distributions at time step 6360. The lower panels show the filtered distribution at time step 6360. In the middle and lower panels, for reference, the true state of x is indicated with dashed lines.

4.2   Lorenz 96 model

In order to evaluate the performance of the MPF for models on higher dimension, we performed another experiment us- ing the Lorenz 96 model (Lorenz and Emanuel, 1998), which is described by the following equations:

$$\frac{dx_j}{dt} = (x_{j+1} - x_{j-2})x_{j-1} - x_j + f \tag{18}$$

Table 1. Root-mean-square deviations from the true state over 50 000 time steps for an experiment using the Lorenz 63 model.

	PF	MPF	EnKF
$N{=}64$	4.55	1.00	1.34
$N{=}128$	3.87	0.91	1.29
$N{=}256$	0.87	0.92	1.29
$N{=}512$	0.86	0.91	1.29

Table 2. Root-mean-square deviations from the true state from time step 3000 to time step 20 000 for an experiment using the Lorenz 96 model. Since the result has converged to the limit, we omitted to calculate the deviations for $N{>}8192$ for the EnKF and those for $N{>}65\,536$ for the MPF.

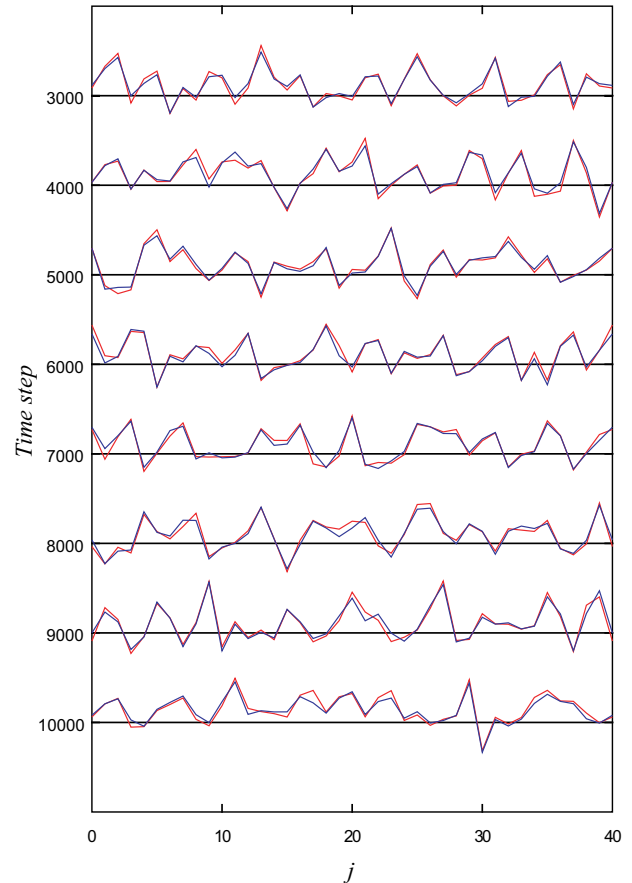|	PF	MPF	EnKF
$N{=}128$	3.47	1.74	0.91
$N{=}256$	3.10	1.03	0.88
$N{=}512$	2.94	0.90	0.87
$N{=}1024$	2.26	0.84	0.87
$N{=}2048$	1.60	0.83	0.86
$N{=}4096$	1.29	0.81	0.86
$N{=}8192$	1.08	0.81	0.86
$N{=}16\,384$	0.96	0.80	–
$N{=}32\,768$	0.84	0.80	–
$N{=}65\,536$	0.83	0.80	–
$N{=}131\,072$	0.79	–	–
$N{=}262\,144$	0.77	–	–

Fig. 8. Result of the experiment of data assimilation by the MPF for the Lorenz 96 model for every 1000 times step from 3000 to 10 000. In this experiment, the number of particles was set to $N{=}512$. The red and blue lines indicate the true state and the estimate by the MPF, respectively.

for $j{=}1,\ldots,J$. Here, $x_{-1}{=}x_{J-1}$, $x_0{=}x_J$, and $x_{J+1}{=}x_1$. In this study, J was set to be 40; that is, the dimension of a state vector is 40. The forcing term f was set to be 8. One time step was set to be 0.005. In order to generate data for the experiment, we ran this model from the initial condition as

$$x_j = 8.0 \qquad \text{(for } j \neq 20\text{)} \qquad (19\text{a})$$
$$x_j = 8.008 \qquad \text{(for } j = 20\text{)}. \qquad (19\text{b})$$

After we iterated the model through 2000 time steps to allow fluctuations in the system to develop sufficiently, the data were generated every 10 time steps with errors having a standard deviation of 1.5. It was assumed that we can observe x_j if j is an even number ($j{=}2,\ldots,40$); that is, if half of the state variables are observed. In assimilating these test data, the system noise was assumed to be a Gaussian noise with zero mean and a diagonal covariance as $\mathrm{diag}(0.25,\ldots,0.25)$. Particles of the forecast ensemble at the initial time step ($T{=}t_1$) were generated from a Gaussian distribution with mean 2.0 and variance 2.0 for each component. Again, in applying the MPF, the number of merged particles was set to $n{=}3$, and the weights α_j were set according to Eq. (16). The likelihood was calculated as follows:

$$p(\mathbf{y}_k|\mathbf{x}_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{||\mathbf{y}_k - H\mathbf{x}_k||^2}{2\sigma^2}\right] \qquad (20)$$

where $\mathbf{y}_k$ is the observation vector ($y_{1,k},\ldots,y_{20,k}$) and σ was set to be 3. The operator H extracts the observable components from the state vector $\mathbf{x}_k$. Since we assume that we can observe x_j for an even number of j, $H\mathbf{x}_k{=}(x_{2,k}x_{4,k}\ldots x_{40,k})^T$.

Figures 8 and 9 show the estimation by the MPF and that by the PF, respectively. In the experiments shown in these figures, the number of particles was set to $N{=}512$. The abscissa indicates j, and the value of x_j for each j for every 1000 time step from 3000 to 10 000 is shown in these figures. As shown in Fig. 9, the PF often deviates from the true state (e.g., at time step 9000). On the other hand, the MPF successfully estimates the state over the period shown here. Table 2 shows the root-mean-square of the deviations from
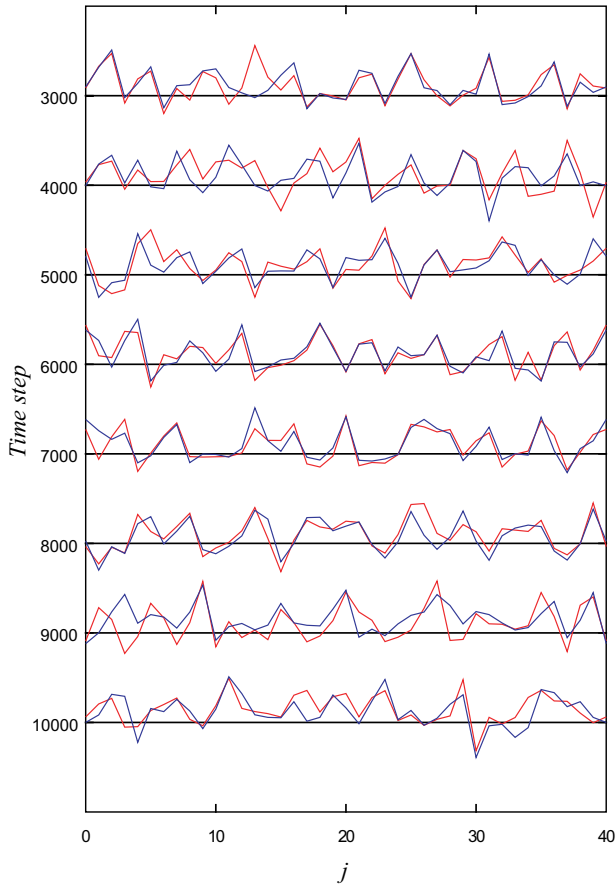
Fig. 9. Result of the experiment of data assimilation by the PF for the Lorenz 96 model for every 1000 time steps from 3000 to 10 000. In this experiment, the number of particles was set to $N=512$. The red and blue lines indicate the true state and the estimate by the PF, respectively.

the true state from time step 3000 to time step 20 000 for various numbers of particles. Again, for reference purposes, the results using the EnKF are also shown in this table. We omitted the calculation of the deviations for $N>8192$ for the EnKF and those for $N>65\,536$ for the MPF which requires much computational resources and cost, because the value of the root-mean-square deviation has converged to the limit and the estimate would not be improved any more even if N increased.

When N is small, the MPF fails to estimate the state, while the EnKF achieves a robust estimation of the state. However, the estimation accuracy of the MPF is remarkably improved when $N=256$, and it becomes better than that of the EnKF when $N\geq1024$. In comparison with the PF, the MPF provides good estimates without requiring a large number of particles. In this experiment, the MPF requires only 1024 particles to obtain as good accuracy as the PF with 32 768 particles. As the number of ensemble members N increases, the

result using the PF is gradually improved, and the root-mean-square of the deviations for the PF seems to converge to a slightly better value than that for the MPF, probably because the MPF does not preserve the shape of the PDF while the PF can faithfully preserve the shape of the filtered PDF with abundant particles. For cases that N is larger than 262 144, we did not perform experiments because they need too much computational resources, and we could not confirm the value which the root-mean-square deviation for the PF converged to. Thus, the result of the PF with a further large ensemble size possibly converges to a further good value than that for $N=262\,144$. However, the use of such an enormous number of particles is not realistic, and it seems to provide only minor improvement of the estimation accuracy even if it were possible. For practical applications to high-dimensional systems, the use of the MPF or the EnKF with much fewer particles would be effectual.

4.3 Lorenz 96 model with nonlinear observation

Another experiment was performed to examine whether the MPF works for the Lorenz 96 model with nonlinear observations. In this experiment, we assumed that we can observe only an absolute value $|x_j|$ if j is an even number ($j=2,\ldots,40$). The data were generated every 10 time steps by taking the absolute values of x_j containing errors with a standard deviation of 1.5. As in the previous experiment, the system noise was assumed to be a Gaussian noise with zero mean and a diagonal covariance as $\mathrm{diag}(0.25,\ldots,0.25)$, and particles of the forecast ensemble at the initial time step ($T=t_1$) were generated from a Gaussian distribution with mean 2.0 and variance 2.0 for each component. The number and the weights of the merged particles in applying the MPF were also the same as in the previous experiment. The likelihood was calculated as follows:

$$p(\boldsymbol{y}_k|\boldsymbol{x}_k) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{||\boldsymbol{y}_k - H(\boldsymbol{x}_k)||^2}{2\sigma^2}\right] \quad (21)$$

where $H(\boldsymbol{x}_k)=(|x_{2,k}|\;|x_{4,k}|\;\ldots\;|x_{40,k}|)^T$ and $\sigma=3$.

Figures 10 and 11 show the estimation by the MPF and that by the PF, respectively. In the experiments shown in these figures, the number of particles was set to $N=1024$. The abscissa indicates j, and, again, the value of x_j for each j for every 1000 time step from 3000 to 10 000 is shown. As shown in Fig. 10, the MPF successfully estimates the state. On the other hand, the PF failed to estimate the state. Table 3 shows the root-mean-square of deviations from the true state from time step 3000 to time step 20 000 for various numbers of particles. The results using the EnKF are also shown in this table again. Here, it should be noted that the algorithm of the EnKF must be modified to apply to cases with nonlinear observations because the EnKF basically assumes a linear relationship between a state and observed data. In applying the EnKF to this particular experiment, according to Evensen
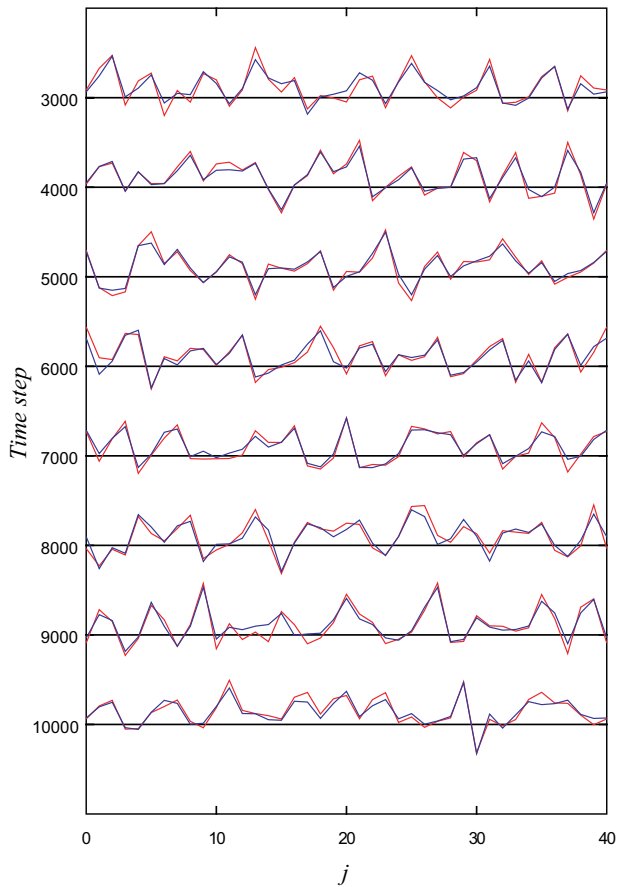
Fig. 10. Result of the experiment of data assimilation by the MPF for the Lorenz 96 model with nonlinear observations for every 1000 time steps from 3000 to 10 000. In this experiment, the number of particles was set to $N=1024$. The red and blue lines indicate the true state and the estimate by the MPF, respectively.

Fig. 11. Result of the experiment of data assimilation by the PF for the Lorenz 96 model with nonlinear observations for every 1000 time steps from 3000 to 10 000. In this experiment, the number of particles was set to $N=1024$. The red and blue lines indicate the true state and the estimate by the PF, respectively.
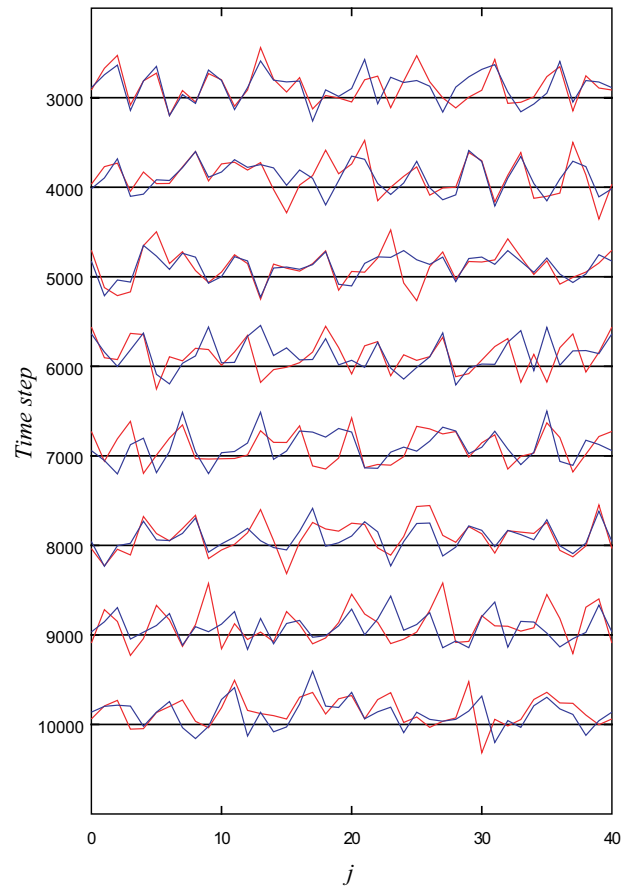
(2003), we define a new state vector $\boldsymbol{x}'_k=[\boldsymbol{x}_k^T, (H(\boldsymbol{x}_k))^T]^T$ such that the observation model becomes linear, and the state space model in Eqs. (15a) and (15b) is accordingly rewritten into a new state space model as follows:

$$\boldsymbol{x}'_k = F'(\boldsymbol{x}'_{k-1}, \boldsymbol{v}_k) \tag{22a}$$

$$\boldsymbol{y}_k = H'\boldsymbol{x}'_k + \boldsymbol{w}_k. \tag{22b}$$

Here the operators F' and H' are defined as:

$$F'(\boldsymbol{x}'_{k-1}, \boldsymbol{v}_k) = F'\big(\boldsymbol{x}_{k-1}, H(\boldsymbol{x}_{k-1}), \boldsymbol{v}_k\big)$$
$$= \begin{pmatrix} F(\boldsymbol{x}_{k-1}) + \boldsymbol{v}_k \\ H\big(F(\boldsymbol{x}_{k-1}) + \boldsymbol{v}_k\big) \end{pmatrix} \tag{23a}$$

$$H' = (O_{\dim\boldsymbol{x}_k}\ I_{\dim\boldsymbol{y}_k}) \tag{23b}$$

where $O_{\dim\boldsymbol{x}_k}$ is a zero matrix whose dimension is the same as $\boldsymbol{x}_k$ and $I_{\dim\boldsymbol{y}_k}$ is an identity matrix whose dimension is

the same as $\boldsymbol{y}_k$, and thus H' extracts $H(\boldsymbol{x}_k)$ from the vector $\boldsymbol{x}'_k$. The EnKF is then applied to this new state space model. Since the results have converged to the limit, we omitted to calculate the deviations for $N>8192$ for the EnKF and those for $N>65\,536$ for the MPF.

As shown in this table, if an enormous number of particles are not allowed, the MPF provides much better results than the PF. The MPF requires only 1024 particles to achieve better accuracy than the PF with 32 768 particles, as well as the previous experiment. With an enormous number of particles, the PF apparently provides better results than the MPF. In this experiment in which only absolute values are allowed to be observed, the filtered PDF may often have multiple modes and moments of higher order than the second moment could then be significant. This situation would limit the accuracy of the MPF, which preserves only the first two moments, even if infinite ensemble members are used. However, the PF requires more than at least 65 536 particles to obtain better

Table 3. Root-mean-square deviations from the true state from time step 3000 to time step 20 000 for an experiment using the Lorenz 96 model with nonlinear observation. Since the result has converged to the limit, we omitted to calculate the deviations for $N > 8192$ for the EnKF and those for $N > 65\,536$ for the MPF.

	PF	MPF	EnKF
$N=128$	4.17	3.56	1.75
$N=256$	4.01	2.47	1.94
$N=512$	3.66	1.50	1.93
$N=1024$	3.70	1.20	1.98
$N=2048$	3.15	1.19	1.99
$N=4096$	2.65	1.14	1.99
$N=8192$	2.07	1.14	1.99
$N=16\,384$	1.80	1.13	–
$N=32\,768$	1.23	1.13	–
$N=65\,536$	1.19	1.13	–
$N=131\,072$	1.04	–	–
$N=262\,144$	1.00	–	–

accuracy than the MPF. Thus, as far as the number of particles is not allowed to increased to more than at least 65 536, the degeneration problem of the PF is more serious than the problem concerning high order moments of the MPF in this experiment. In comparison between the MPF and the EnKF, when N is small, the EnKF provides better estimations again, although estimations by the EnKF are not so good. When $N \geq 512$, the estimation accuracy of the MPF is remarkably improved to be much better than that of the EnKF. Actually, the EnKF does not effectively work in this experiment. Figure 12 shows the estimation by the EnKF, where the number of particles was set to $N=1024$. It is indicated that the estimates by the EnKF often significantly deviate from the true state which means that the EnKF fails to capture the variation of the true state. Thus, for this experiment, the use of the MPF would be the most effectual.

5   Summary and discussion

We proposed a new algorithm, the MPF, for realizing practical sequential data assimilation. The MPF provides an ensemble-based approximation of the filtered PDF such that the mean and covariance are approximately preserved. The MPF allows the problem of degeneration, which occurs in the PF, to be avoided. It must be noted that the MPF does not preserve the shape of the filtered PDF while the PF can faithfully preserve the shape of the filtered PDF with abundant particles. Therefore, if a sufficient number of particles is used, the PF should provide a better estimation than the MPF. In particular, in cases that the filtered PDF is significantly non-Gaussian, the MPF possibly provides a rather bad estimate. In application to a high-dimensional system, however, it is not realistic to use a sufficient particles to avoid de-
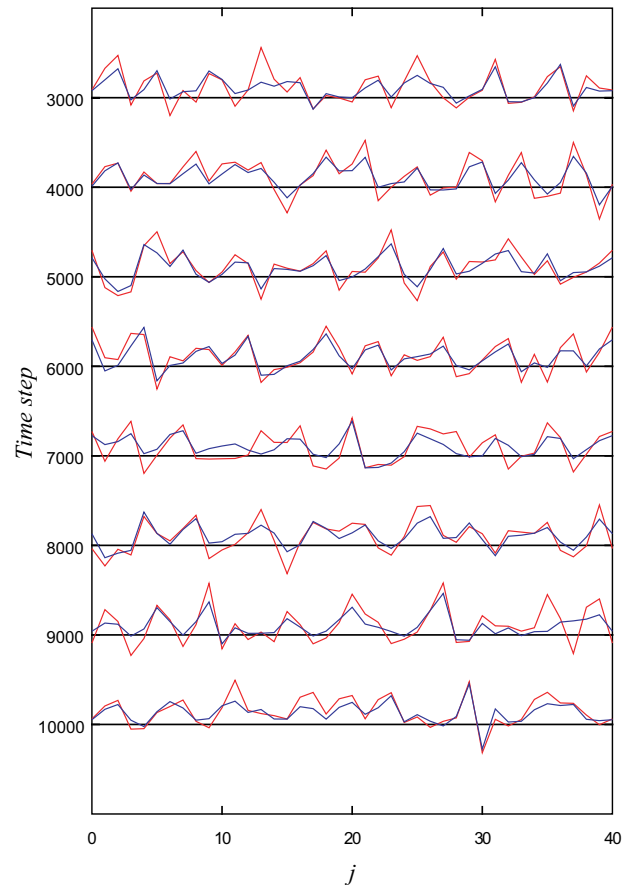

Fig. 12. Result of the experiment of data assimilation by the EnKF for the Lorenz 96 model with nonlinear observations for every 1000 time steps from 3000 to 10 000. In this experiment, the number of particles was set to $N=1024$. The red and blue lines indicate the true state and the estimate by the EnKF, respectively.

generation, and therefore the PF should fail to approximate the filtered PDF. Indeed, as illustrated in Sect. 4.2, the PF provides a worse estimation of the state than the MPF for the Lorenz 96 model, until the number of particles in the ensemble was increased to at least 65 536. Since usual geophysical models are of much higher dimension than the Lorenz 96 model, although they could be less nonlinear, a hopelessly large number of particles would be required in order to use the PF. The MPF requires far fewer particles than the PF and thus would be a more effectual algorithm.

In addition, the MPF is applicable to cases in which the relationship between a state and observed data is nonlinear. For cases with nonlinear observations, the EnKF does not necessarily provide a good estimation of the state. As illustrated in Sect. 4.3, even if the number of particles is increased, the estimation by the EnKF is not improved, whereas that by the MPF is remarkably improved. Therefore, the MPF would be the best method of sequential data assimilation with nonlinear observations.

Table 4. Comparison among the algorithms for sequential data assimilation with a high-dimensional nonlinear system.

	PF	MPF	EnKF
Nonlinear observation	OK	OK	Ineffectual for some cases
Necessary number of particles	Exceedingly many	Medium	Relatively few
Cost of filtering	Low	Low	High

For cases in which the relationship between a state in the system and observed data is linear, the EnKF basically provides a good estimation without a large number of particles. However, the EnKF tends to require a higher computational cost at each filtering step in applying to a high-dimensional model, because it involves many multiplications and additions between matrices. In addition, even if the number of particles is taken to be small, estimates using the EnKF can be affected by spurious correlations between distant locations, and thus localization on the covariance matrix (Ott et al., 2004) might be required to avoid this problem. On the other hand, a computational cost at each filtering step is not serious in the MPF, because neither iterative calculations of inverse matrices nor numerous multiplications between matrices are required. Therefore, for cases in which a system model does not require a great deal of computational time, the MPF may perform better than the EnKF.

Table 4 summarizes the characteristics of the algorithms of the PF, the MPF, and the EnKF. In the cases of a nonlinear relationship between a state and observed data, the EnKF does not necessarily work, whereas the PF or the MPF can be applied. The PF requires an exceedingly large number of particles, which imposes prohibitive computational cost at each forecast step. The MPF requires far fewer particles than the PF, although the EnKF requires fewer particles than the MPF. As for the computational cost at each filtering step, the EnKF requires a larger computational cost than the PF and the MPF. The high computational cost at each filtering step would become serious in the case that the number of assimilated data is large. On the other hand, the increase in the number of particles causes a high computational cost at each forecasting step, which becomes serious for the case in which a system model requires a great deal of computational time. Therefore, for the case in which only linear observations are used, the choice between the MPF and the EnKF should be made based on the considerations of the dimension of the observation vector and the computational cost required by the system model.

References

Anderson, J. L.: A ensemble adjustment Kalman filter for data assimilation, Mon. Wea. Rev., 129, 2884–2903, 2001.

Anderson, J. L. and Anderson, S. L.: A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts, Mon. Wea. Rev., 127, 2741–2758, 1999.

Burgers, G., van Leeuwen, P. J., and Evensen, G.: Analysis scheme in the ensemble Kalman filter, Mon. Wea. Rev., 126, 1719–1724, 1998.

Evensen, G.: Using the extended Kalman filter with a multilayer quasi-geostrophic model, J. Geophys. Res., 97(C11), 17 905–17 924, 1992.

Evensen, G.: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, J. Geophys. Res., 99(C5), 10 143–10 162, 1994.

Evensen, G.: The ensemble Kalman filter: theoretical formulation and practical implementation, Ocean Dynam., 53, 343–367, doi:10.1007/s10236-003-0036-9, 2003.

Goldberg, D. E.: Genetic algorithms in search, optimization and machine learning, Addison-Wesley, Reading, 1989.

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

Higuchi, T. and Kitagawa, G.: Knowledge discovery and self-organizing state space model, IEICE Transactions on Information and Systems, E83-D, 36–43, 2000.

Hürzeler, M. and Künsch, H. R.: Monte Carlo approximations for general state space models, J. Comp. Graph. Statist., 7, 175–191, 1998.

Kitagawa, G.: Monte Carlo filtering and smoothing method for non-Gaussian nonlinear state space model, Inst. Statist. Math. Res. Memo., 1993.

Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, J. Comp. Graph. Statist., 5, 1–25, 1996.

Kitagawa, G. and Gersch, W.: Smoothness priors analysis of time series, chap. 6, Springer-Verlag, New York, 1996.

Kivman, G. A.: Sequential parameter estimation for stochastic systems, Nonlin. Process. Geophys., 10, 253–259, 2003.

Kotecha, J. H. and Djurić, P. M.: Gaussian particle filtering, IEEE Trans. Signal Processing, 51, 2592–2601, 2003.

Lorenz, E. N.: Deterministic nonperiodic flow, J. Atmos. Sci., 20, 130–141, 1963.

Lorenz, E. N. and Emanuel, K. A.: Optimal sites for supplementary weather observations: Simulations with a small model, J. Atmos. Sci., 55, 399–414, 1998.

Musso, C., Oudjane, N., and Le Gland, F.: Improving regularized particle filters, in: Sequential Monte Carlo methods in practice, edited by Doucet, A., de Freitas, N., and Gordon, N., chap. 12, p. 247, Springer-Verlag, New York, 2001.

Nerger, L., Hiller, W., and Schröter, J.: A comparison of error subspace Kalman filters, Tellus, 57A, 715–735, 2005.

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, D. J., and Yorke, J. A.: A local ensemble Kalman filter for atmospheric data assimilation, Tellus, 56A, 415–428, 2004.

Pham, D. T., Verron, J., and Gourdeau, L.: Singular evolutive Kalman filters for data assimilation in oceanography, C. R. Acad. Sci. Ser. II, 326, 255–260, 1998a.

Pham, D. T., Verron, J., and Roubaud, M. C.: A singular evolutive extended Kalman filter for data assimilation in oceanography, J. Mar. Syst., 16, 323–340, 1998b.

van Leeuwen, P. J.: A variance-minimizing filter for large-scale applications, Mon. Wea. Rev., 131, 2071–2084, 2003.

Whitaker, J. S. and Hamill, T. M.: Ensemble data assimilation without perturbed observations, Mon. Wea. Rev., 130, 1913–1924, 2002.