Hybrid three-dimensional variation and particle filtering for nonlinear systems*

Leng Hong-Ze(冷洪泽)† and Song Jun-Qiang(宋君强)

College of Computer, National University of Defense Technology, Changsha 410073, *China*

(Received 10 June 2012; revised manuscript received 7 September 2012)

This work addresses the problem of estimating the states of nonlinear dynamic systems with sparse observations. We present a hybrid three-dimensional variation (3DVar) and particle piltering (PF) method, which combines the advantages of 3DVar and particle-based filters. By minimizing the cost function, this approach will produce a better proposal distribution of the state. Afterwards the stochastic resampling step in standard PF can be avoided through a deterministic scheme. The simulation results show that the performance of the new method is superior to the traditional ensemble Kalman filtering (EnKF) and the standard PF, especially in highly nonlinear systems.

Keywords: three-dimensional variation (3DVar), particle piltering (PF), ensemble Kalman filtering (EnKF), chaos system

PACS: 05.45.–a, 02.60.–x **DOI:** 10.1088/1674-1056/22/3/030505

1. Introduction

Nonlinear filtering problems are present in many areas such as economics, statistical signal processing, robot controls, biostatistics, image processing, communications, target tracking, numerical weather prediction, and satellite navigation. These problems can be written in the form of so-called dynamic state space (DSS) modeling. Denoting the sequence of system states by $\{\boldsymbol{x}^k\}$, and the sequence of observation states by $\{\boldsymbol{y}^k\}$, where k is a discrete time index, then $\boldsymbol{x}^k \in \mathbb{R}^n$ and $\boldsymbol{y}^k \in \mathbb{R}^m$ represent the state vector and the observation vector, respectively. Assume that the dynamic system can be effectively modeled as a first-order Markov process, that is, the future state of the system only relies on the current state and is independent of old information, and therefore we have

$$\begin{cases} \boldsymbol{x}^k = f(\boldsymbol{x}^{k-1}, \boldsymbol{\mu}^k), \\ \boldsymbol{y}^k = h(\boldsymbol{x}^k, \boldsymbol{\nu}^k), \end{cases} \tag{1}$$

where $\boldsymbol{\mu}^k \in \mathbb{R}^n$ and $\boldsymbol{\nu}^k \in \mathbb{R}^m$ denote the model error and the observation error, respectively. A trajectory $\{\boldsymbol{x}^k\}$ is required which best fits the observations and the true posterior distribution.

One of the well-known methods for filtering problems is Kalman filtering (KF)[1] as well as its variants. For linear and Gaussian-distributed systems, KF provides optimal solutions. For weakly nonlinear systems, the extended Kalman filtering (EKF)[2,3] can be used to obtain a sub-optimal estimation. Applied to high-dimensional systems, however, EKF will lead to a large computational cost. Therefore, the ensemble Kalman filtering (EnKF)[4–6] is often used to simplify the background error covariance in the EKF and can ease the approximation of a probability distribution. Furthermore, unscented Kalman filtering (UKF)[7] is often used in nonlinear state estimation, especially in systems with a relatively low dimension.

Particle filtering (PF),[8] also known as a serial Monte Carlo method, is developed as a fully nonlinear and non-Gaussian approach, and has been studied extensively in recent years.[9–14] The PF aims at representing the posterior probability distribution of the system state by a set of weighted samples, the so-called particles. The PF enables the online processing of the observations, and the range of its application is very broad. However, the resampling step, which stochastically replicates high-weight particles while discarding those with lower weights, reduces the diversity of the particles and makes the parallel computation more challenging. Some new PF algorithms with no resampling have been proposed,[15–17] but the proposal distribution, also called the importance distribution, can potentially be improved.

Due to their respective advantages and disadvantages, the combination of Kalman filtering and PF has also been presented. Won *et al*.[18] built a position tracking system that uses a combination of Kalman filter and particle filter to synthesize the measurements from a measurement unit and a position sensor. A similar method was proposed in Ref. [19] which estimates orientation using a particle filter while it estimates position and velocity using a Kalman filter. In the particle Kalman filter (PKF),[20] the solution of a nonlinear filtering problem is expressed as the weighted average of an "ensemble of Kalman filters" operating in parallel. The standard weight-type correction in the PF is complemented by the Kalman-type correction with the associated covariance matrices in the Gaussian mixture. In the adaptive Gaussian mixture filter,[21] the posterior distribution is regarded as the sum of Gaussian kernels.

*Project supported by the National Natural Science Foundation of China (Grant No. 41105063).

†Corresponding author. E-mail: hzleng@nudt.edu.cn

© 2013 Chinese Physical Society and IOP Publishing Ltd

By introducing a tuning parameter for the importance weights, the Gaussian mixture filter has the virtue of both the weighting/resampling step of a particle filter and a local Kalman type correction.

In this paper, another approach named three-dimensional variation (3DVar)[22] is introduced, and it is combined with PF. In the new algorithm, the proposal distribution is improved by using ensemble 3DVar, which gives a better approximation of the posterior distribution, and when the new particles are generated through a deterministic scheme, the resampling step can be avoided.

The rest of this paper is organized as follows. Background knowledge is given in Section 2. In Section 3, firstly the idea of ensemble 3DVar is presented, and then it is combined with PF by weighting the sub-ensembles of particles. In Section 4, the comparison results of the three methods applied to a low-dimensional chaos system are illustrated, and the conclusion is given in Section 5.

2. Background knowledge

2.1. 3DVar method

Assume that the background (a priori) and the observation are independent from each other, and we can obtain the possibility of the true state[22] as

$$L(\boldsymbol{x};\boldsymbol{x}^{\mathrm{b}},\boldsymbol{y}^{\mathrm{o}}) = p_{\mathrm{B}}(\boldsymbol{x}^{\mathrm{b}}|\boldsymbol{x})\cdot p_{\mathrm{R}}(\boldsymbol{y}^{\mathrm{o}}|\boldsymbol{x}), \tag{2}$$

$$p_{\mathrm{B}}(\boldsymbol{x}^{\mathrm{b}}|\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}|\boldsymbol{B}|^{1/2}} \times \exp\left\{-\frac{1}{2}\left[(\boldsymbol{x}^{\mathrm{b}}-\boldsymbol{x})^{\mathrm{T}}\boldsymbol{B}^{-1}(\boldsymbol{x}^{\mathrm{b}}-\boldsymbol{x})\right]\right\}, \tag{3}$$

$$p_{\mathrm{R}}(\boldsymbol{y}^{\mathrm{o}}|\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{R}|^{1/2}} \times \exp\left\{-\frac{1}{2}\left[(\boldsymbol{y}^{\mathrm{o}}-h(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}^{\mathrm{o}}-h(\boldsymbol{x}))\right]\right\}, \tag{4}$$

where $\boldsymbol{B}$ is the background error covariance based on the mean of the statistical results of the state in a period of time, $\boldsymbol{R}$ is the observation error covariance, which is usually a diagonal matrix, and $h(\cdot)$ is the observation operator.

The most probable state $\boldsymbol{x}$ is the one that maximizes $L(\boldsymbol{x};\boldsymbol{x}^{\mathrm{b}},\boldsymbol{y}^{\mathrm{o}})$, namely, the one that minimizes the following cost function:

$$J(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}^{\mathrm{b}}-\boldsymbol{x})^{\mathrm{T}}\boldsymbol{B}^{-1}(\boldsymbol{x}^{\mathrm{b}}-\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{y}^{\mathrm{o}}-h(\boldsymbol{x}))^{\mathrm{T}}\boldsymbol{R}^{-1}(\boldsymbol{y}^{\mathrm{o}}-h(\boldsymbol{x})). \tag{5}$$

By solving the cost function, one can obtain the optimal estimation if the system is linear and Gaussian-distributed.

2.2. Particle filtering

As a serial Monte Carlo method, PF makes use of a set of weighted samples, called "particles", to approach the distribution of the state.

Assume that there are N independent particles and we can approximate the probability density function (PDF) of $\boldsymbol{x}$ by a sum of Dirac functions[23]

$$p(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\boldsymbol{x}-\boldsymbol{x}_i). \tag{6}$$

Then the expectation of an arbitrary function $f(\boldsymbol{x})$ can be written as

$$E(f(\boldsymbol{x})) = \int f(\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \approx E_p(f(\boldsymbol{x})) = \frac{1}{N}\sum_{i=1}^{N}f(\boldsymbol{x}_i). \tag{7}$$

Unfortunately, it is very difficult to sample directly from the exact PDF of the state. Then the so-called proposal density function q is introduced, which approximates the posterior distribution, from which sampling can be made easier. The corresponding weight of each particle $\boldsymbol{x}_i^k$ is defined as

$$s_i = \frac{1}{A}\frac{p(\boldsymbol{y}^k|\boldsymbol{x}_i^k)p(\boldsymbol{x}_i^k|\boldsymbol{x}_i^{k-1})}{q(\boldsymbol{x}_i^k|\boldsymbol{x}_i^{k-1},\boldsymbol{y}^k)}, \tag{8}$$

where k is the time serial number, $q(\boldsymbol{x}_i^k|\boldsymbol{x}_i^{k-1},\boldsymbol{y}^k)$ is the proposal density, which is also called the importance density, $p(\boldsymbol{y}^k|\boldsymbol{x}_i^k)$ denotes the likelihood function, $p(\boldsymbol{x}_i^k|\boldsymbol{x}_i^{k-1})$ is a priori distribution, and A indicates the factor of normalization.

3. New algorithm

In standard PF, the proposal density $q(\boldsymbol{x}_i^k|\boldsymbol{x}_i^{k-1},\boldsymbol{y}^k)$ is set to be $p(\boldsymbol{y}^k|\boldsymbol{x}_i^k)$. It is easy to sample from the distribution, but it is not a good approximation of the posterior distribution since it has no new information about current observations, so it will result in filtering divergence. Finding a proper proposal density q is one of the key problems which needs to be resolved for the PF method.

Another scheme to avoid divergence is to add a resampling step after importance sampling. However, once the number of particles becomes significantly large, the computation cost of resampling cannot be ignored, meanwhile the parallelizability is limited.

In this section, we present a new algorithm combining 3DVar and PF, not only to give a better proposal distribution, but also to avoid the resampling steps.

3.1. Ensemble 3DVar

To minimize the cost function of 3DVar, one has to calculate the inverse of the background error covariance $\boldsymbol{B}$ in Eq. (5), which is very expensive in time.

An alternative is to rebuild the background covariance from a set of particles. We assume that there are N independent particles, and define

$$\bar{\boldsymbol{x}}^{\mathrm{b}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i^{\mathrm{b}}, \tag{9}$$

$$\delta \boldsymbol{x}_i^{\mathrm{b}} = \boldsymbol{x}_i^{\mathrm{b}} - \bar{\boldsymbol{x}}^{\mathrm{b}}, \tag{10}$$

$$\boldsymbol{X}^{\mathrm{b}} = [\delta \boldsymbol{x}_1^{\mathrm{b}}, \delta \boldsymbol{x}_2^{\mathrm{b}}, \ldots, \delta \boldsymbol{x}_N^{\mathrm{b}}], \tag{11}$$

$$\boldsymbol{P}^{\mathrm{b}} = \frac{1}{N-1} \boldsymbol{X}^{\mathrm{b}} (\boldsymbol{X}^{\mathrm{b}})^{\mathrm{T}}, \tag{12}$$

where $\bar{\boldsymbol{x}}^{\mathrm{b}}$ is the expected background state, and $\boldsymbol{P}^{\mathrm{b}}$ is the approximate background covariance matrix.

By introducing a new control variable $\boldsymbol{w}$ and defining

$$\boldsymbol{x} = \bar{\boldsymbol{x}}^{\mathrm{b}} + \boldsymbol{X}^{\mathrm{b}} \boldsymbol{w}, \tag{13}$$

$$h(\boldsymbol{x}) = h(\bar{\boldsymbol{x}}^{\mathrm{b}} + \boldsymbol{X}^{\mathrm{b}} \boldsymbol{w}) \approx \bar{\boldsymbol{y}}^{\mathrm{b}} + \boldsymbol{Y}^{\mathrm{b}} \boldsymbol{w}, \tag{14}$$

where

$$h(\bar{\boldsymbol{x}}^{\mathrm{b}}) \approx \bar{\boldsymbol{y}}^{\mathrm{b}} = \frac{1}{N} \sum_{i=1}^{N} h(\boldsymbol{x}_i^{\mathrm{b}}), \tag{15}$$

$$h(\boldsymbol{X}^{\mathrm{b}}) \approx \boldsymbol{Y}^{\mathrm{b}} = \left[h(\boldsymbol{x}_1^{\mathrm{b}}) - h(\bar{\boldsymbol{x}}^{\mathrm{b}}), \ldots, h(\boldsymbol{x}_N^{\mathrm{b}}) - h(\bar{\boldsymbol{x}}^{\mathrm{b}}) \right], \tag{16}$$

then we can obtain a new cost function with respect to the new control variable $\boldsymbol{w}$ as

$$\begin{aligned} \widehat{J}(\boldsymbol{w}) \;=\; & \frac{1}{2}(N-1)\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + \frac{1}{2}(\boldsymbol{y}^{\mathrm{o}} - \bar{\boldsymbol{y}}^{\mathrm{b}} - \boldsymbol{Y}^{\mathrm{b}}\boldsymbol{w})^{\mathrm{T}} \\ & \times \boldsymbol{R}^{-1}(\boldsymbol{y}^{\mathrm{o}} - \bar{\boldsymbol{y}}^{\mathrm{b}} - \boldsymbol{Y}^{\mathrm{b}}\boldsymbol{w}), \end{aligned} \tag{17}$$

and its gradient as

$$\nabla \widehat{J} = (N-1)\boldsymbol{w} - (\boldsymbol{Y}^{\mathrm{b}})^{\mathrm{T}} \boldsymbol{R}^{-1} (\boldsymbol{y}^{\mathrm{o}} - \bar{\boldsymbol{y}}^{\mathrm{b}} - \boldsymbol{Y}^{\mathrm{b}}\boldsymbol{w}). \tag{18}$$

If $\boldsymbol{w}^{\mathrm{a}}$ minimizes $\widehat{J}$, the corresponding vector $\boldsymbol{x}^{\mathrm{a}}$ must minimize J. The proof is shown in the appendix.

In this way, the calculation of the inverse matrix $\boldsymbol{B}$ is avoided, thus greatly reducing the computational expense. On the other hand, a better estimation of the state can be obtained with the flow-dependent forecasting covariance $\boldsymbol{P}^{\mathrm{b}}$. Now we find the linear combination of the trajectories to best fit the observations in 3DVar.

For the new system with control variable $\boldsymbol{w}$, we know the background value $\bar{\boldsymbol{w}}^{\mathrm{b}} = 0$, and its error covariance $\boldsymbol{P}^{\mathrm{w,b}} = (N-1)^{-1}\boldsymbol{I}$ from Eq. (17), and from gradient function (18) we have

$$\bar{\boldsymbol{w}}^{\mathrm{a}} = \boldsymbol{P}^{\mathrm{w,a}} \cdot \left[(\boldsymbol{Y}^{\mathrm{b}})^{\mathrm{T}} \boldsymbol{R}^{-1} (\boldsymbol{y}^{\mathrm{o}} - \bar{\boldsymbol{y}}^{\mathrm{b}}) \right], \tag{19}$$

$$\boldsymbol{P}^{\mathrm{w,a}} = \left[(N-1)\boldsymbol{I} + (\boldsymbol{Y}^{\mathrm{b}})^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{Y}^{\mathrm{b}} \right]^{-1}, \tag{20}$$

where $\boldsymbol{P}^{\mathrm{w,a}}$ denotes the analysis error covariance of control variable $\boldsymbol{w}$. Therefore, $\bar{\boldsymbol{x}}^{\mathrm{a}} = \bar{\boldsymbol{x}}^{\mathrm{b}} + \boldsymbol{X}^{\mathrm{b}}\bar{\boldsymbol{w}}^{\mathrm{a}}$, $\boldsymbol{P}^{\mathrm{a}} = \boldsymbol{X}^{\mathrm{b}}\boldsymbol{P}^{\mathrm{w,a}}(\boldsymbol{X}^{\mathrm{b}})^{\mathrm{T}}$.

By defining $\boldsymbol{W}^{\mathrm{a}} = [(N-1)\boldsymbol{P}^{\mathrm{w,a}}]^{1/2}$, one can decompose the analysis covariance as follows:

$$\boldsymbol{P}^{\mathrm{w,a}} = (N-1)^{-1}\boldsymbol{W}^{\mathrm{a}}(\boldsymbol{W}^{\mathrm{a}})^{\mathrm{T}}, \quad \boldsymbol{P}^{\mathrm{a}} = (N-1)^{-1}\boldsymbol{X}^{\mathrm{a}}(\boldsymbol{X}^{\mathrm{a}})^{\mathrm{T}},$$

where $\boldsymbol{X}^{\mathrm{a}} = \boldsymbol{X}^{\mathrm{b}}\boldsymbol{W}^{\mathrm{a}}$. Then, the new particles are written as

$$\boldsymbol{x}_i^{\mathrm{a}} = \bar{\boldsymbol{x}}^{\mathrm{a}} + \boldsymbol{X}^{\mathrm{a}}(i) = \bar{\boldsymbol{x}}^{\mathrm{b}} + \boldsymbol{X}^{\mathrm{b}}(\bar{\boldsymbol{w}}^{\mathrm{a}} + \boldsymbol{W}^{\mathrm{a}}(i)), \tag{21}$$

where $\boldsymbol{X}^{\mathrm{a}}(i)$ and $\boldsymbol{W}^{\mathrm{a}}(i)$ denote the i-th column of $\boldsymbol{X}^{\mathrm{a}}$ and $\boldsymbol{W}^{\mathrm{a}}$, respectively.

To avoid the reverse of matrix in Eq. (20), one can use a decomposition equation[24]

$$\begin{aligned} \boldsymbol{W}^{\mathrm{a}} \;&=\; [(N-1)\boldsymbol{P}^{\mathrm{w,a}}]^{1/2} = \left[\frac{1}{N-1} (\boldsymbol{P}^{\mathrm{w,a}})^{-1} \right]^{-1/2} \\ &=\; \left[\boldsymbol{I} + \frac{1}{N-1} (\boldsymbol{Y}^{\mathrm{b}})^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{Y}^{\mathrm{b}} \right]^{-1/2} \\ &=\; \boldsymbol{V} \cdot (\boldsymbol{I} + \boldsymbol{D})^{-1/2} \cdot \boldsymbol{V}^{\mathrm{T}}, \end{aligned} \tag{22}$$

where $\boldsymbol{V}$ and $\boldsymbol{D}$ are respectively the eigenvector matrix and the eigenvalue diagonal matrix of $1/(N-1) \cdot (\boldsymbol{Y}^{\mathrm{b}})^{\mathrm{T}} \boldsymbol{R}^{-1} \boldsymbol{Y}^{\mathrm{b}}$.

The problem of ensemble 3DVar has been studied before,[25] but our algorithm is presented from a different point of view, and the implementation details are also distinguishable.

3.2. Combining ensemble 3DVar and particle filtering

Considering the shortcoming of linearization of the observation operator when it is nonlinear, we introduce the idea of PF, which is fully nonlinear. We split up the ensemble with NG particles into G sub-ensembles, with N members each.

For each sub-ensemble g, we define

$$\boldsymbol{P}_g^{\mathrm{b}} = \frac{1}{N-1} \boldsymbol{X}_g^{\mathrm{b}} (\boldsymbol{X}_g^{\mathrm{b}})^{\mathrm{T}}, \quad \boldsymbol{P}_g^{\mathrm{a}} = \frac{1}{N-1} \boldsymbol{X}_g^{\mathrm{a}} (\boldsymbol{X}_g^{\mathrm{a}})^{\mathrm{T}},$$

where $\boldsymbol{X}_g^{\mathrm{a}} = \boldsymbol{X}_g^{\mathrm{b}} \boldsymbol{W}_g^{\mathrm{a}}$, and $\boldsymbol{W}_g^{\mathrm{a}} = \left[(N-1)\boldsymbol{P}_g^{\mathrm{w,a}} \right]^{1/2}$.

If we assume that the error is Gaussian-distributed, the weight s_g of each sub-ensemble g or each "centre" particle $\bar{\boldsymbol{x}}_g^{\mathrm{a}}$ can be written as

$$\begin{aligned} s_g^k \;&\propto\; s_g^{k-1} p(\boldsymbol{y}^{\mathrm{o}} | \bar{\boldsymbol{x}}_g^{\mathrm{a}}) p(\bar{\boldsymbol{x}}_g^{\mathrm{b}} | \bar{\boldsymbol{x}}_g^{\mathrm{a}}) \\ &\propto\; s_g^{k-1} \exp \left\{ -\frac{1}{2} \left[(\bar{\boldsymbol{x}}_g^{\mathrm{a}} - \bar{\boldsymbol{x}}_g^{\mathrm{b}})^{\mathrm{T}} (\boldsymbol{P}_g^{\mathrm{b}})^{-1} (\bar{\boldsymbol{x}}_g^{\mathrm{a}} - \bar{\boldsymbol{x}}_g^{\mathrm{b}}) \right] \right. \\ &\quad \left. -\frac{1}{2} \left[(\boldsymbol{y}^{\mathrm{o}} - h(\bar{\boldsymbol{x}}_g^{\mathrm{a}}))^{\mathrm{T}} \boldsymbol{R}^{-1} (\boldsymbol{y}^{\mathrm{o}} - h(\bar{\boldsymbol{x}}_g^{\mathrm{a}})) \right] \right\} \\ &\propto\; s_g^{k-1} \exp \left[-J(\bar{\boldsymbol{x}}_g^{\mathrm{a}}) \right]. \end{aligned} \tag{23}$$

The weights should then be normalized. It is noted that $(\bar{\boldsymbol{x}}_g^{\mathrm{a}} - \bar{\boldsymbol{x}}_g^{\mathrm{b}})^{\mathrm{T}} (\boldsymbol{P}_g^{\mathrm{b}})^{-1} (\bar{\boldsymbol{x}}_g^{\mathrm{a}} - \bar{\boldsymbol{x}}_g^{\mathrm{b}})$ is equal to $(N-1)\boldsymbol{w}_g^{\mathrm{T}}\boldsymbol{w}_g$. The expected value of the state is

$$\bar{\boldsymbol{x}}^{\mathrm{a}} = \sum_{g=1}^{G} s_g \boldsymbol{x}_g^{\mathrm{a}}. \tag{24}$$

In Eq. (23), the primary cost function J is adopted instead of $\widehat{J}$, since it includes the full information about nonlinearity.

Now giving the expected value, we need to construct the new ensemble of particles. From the old ensemble, the departure of each particle can be obtained from the expected value as follows:

$$\delta \boldsymbol{x}_{g,i} \;=\; \boldsymbol{x}_{g,i}^{\mathrm{a}} - \bar{\boldsymbol{x}}^{\mathrm{a}} = [\bar{\boldsymbol{x}}_g^{\mathrm{a}} + \boldsymbol{X}_g^{\mathrm{a}}(i)] - \bar{\boldsymbol{x}}^{\mathrm{a}}$$

$$= (\bar{\boldsymbol{x}}_g^{\mathrm{a}} - \bar{\boldsymbol{x}}^{\mathrm{a}}) + \boldsymbol{X}_g^{\mathrm{a}}(i). \tag{25}$$

The new analysis error covariance is defined by the linear combination of covariances of sub-ensembles, that is,

$$\begin{aligned}\boldsymbol{P}^{\mathrm{a}} &= \sum_{g=1}^{G} s_g \boldsymbol{P}_g^{\mathrm{a}} = \sum_{g=1}^{G} \left(s_g \sum_{i=1}^{N} \frac{1}{N-1} \delta \boldsymbol{x}_{g,i} (\delta \boldsymbol{x}_{g,i})^{\mathrm{T}} \right) \\ &= 1/(NG-1) \cdot \boldsymbol{X}^{\mathrm{a}} \cdot (\boldsymbol{X}^{\mathrm{a}})^{\mathrm{T}}, \end{aligned} \tag{26}$$

where

$$\begin{aligned}\boldsymbol{X}^{\mathrm{a}} = \sqrt{NG-1} \Bigg[&\sqrt{\frac{s_1}{N-1}} (\delta \boldsymbol{x}_{1,1}, \delta \boldsymbol{x}_{1,2}, \ldots, \delta \boldsymbol{x}_{1,N}), \ldots, \\ &\sqrt{\frac{s_G}{N-1}} (\delta \boldsymbol{x}_{G,1}, \delta \boldsymbol{x}_{G,2}, \ldots, \delta \boldsymbol{x}_{G,N}) \Bigg]. \end{aligned} \tag{27}$$

Thus,

$$\delta \boldsymbol{x}_{g,i}^{\mathrm{new}} = \sqrt{\frac{NG-1}{N-1}} s_g \delta \boldsymbol{x}_{g,i}. \tag{28}$$

Actually, the weight of each particle can be comprehended as follows: on one side, if $s_g < (N-1)/(NG-1)$, the old sub-ensemble can be regarded as being far from the centre, thus it should be "pulled" closer to the centre; on the other side, if $s_g > (N-1)/(NG-1)$, the old sub-ensemble should be "pushed" farther away in order to keep diversity.

The new ensemble is obtained for the next generation and all the particles have the same weight again. Here we have avoided the resampling step and have selected the new particles in a deterministic way.

3.3. Overview of the algorithm

Step 1 Divide the given ensemble into several sub-ensembles. Different divisions may be needed for different issues.

Step 2 Utilize a certain gradient-based unconstrained method to minimize the new cost function $\widehat{J}(\boldsymbol{w}_g)$ for each sub-ensemble, and then calculate $\bar{\boldsymbol{x}}_g^{\mathrm{a}} = \bar{\boldsymbol{x}}_g^{\mathrm{b}} + \boldsymbol{X}_g^{\mathrm{b}} \bar{\boldsymbol{w}}_g^{\mathrm{a}}$.

Step 3 Calculate the weight s_g for each "centre" particle $\bar{\boldsymbol{x}}_g^{\mathrm{a}}$, and the square root $\boldsymbol{X}^{\mathrm{a}}$ of the analysis error covariance $\boldsymbol{P}^{\mathrm{a}}$.

Step 4 Calculate the estimation of the state and construct the new particles deterministically for the next generation.

4. Experiments and results

In this section, we apply the new algorithm (hereafter referred to as 3DPF) to a low-dimensional chaos system and compare it with the standard EnKF and standard PF.

The Lorenz63 system is often used as a benchmark to test the performance of filtering algorithms, and it can be written as

$$\frac{\mathrm{d}x}{\mathrm{d}t} = a(y-x), \frac{\mathrm{d}y}{\mathrm{d}t} = (b-z)x - y, \frac{\mathrm{d}z}{\mathrm{d}t} = xy - cz, \tag{29}$$

where $a = 10$, $b = 28$, and $c = 8/3$. The time step $\mathrm{d}t$ is set as 0.01, and the simulation window T as 1000.

4.1. Linear observation system

Suppose that the model error covariance is

$$\boldsymbol{Q} = [0.10, 0.01, 0.02; 0.01, 0.20, 0.01; 0.02, 0.01, 0.30].$$

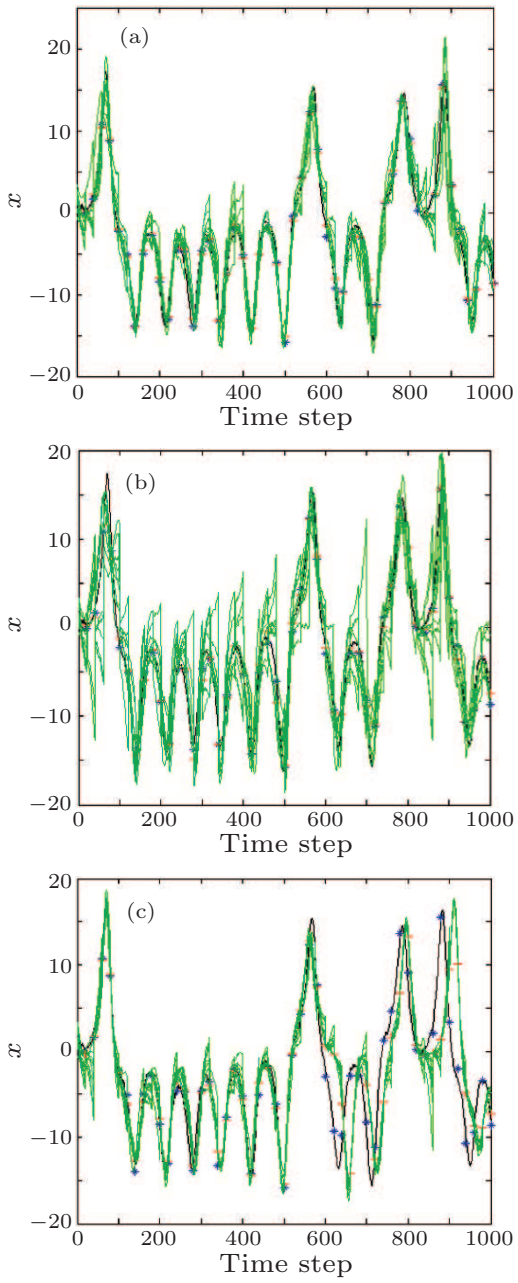


Fig. 1. (color online) Comparison of three methods for the Lorenz-63 model: (a) 3DPF, (b) EnKF, (c) PF. The black line is the true solution, the plus signs represent the analysis values, the stars denote the perturbed observations, and the shaded lines (green in the online paper) depict the trajectories of the observed variable in time. It is noted that the truth can be followed by only 9 particles with 3DPF.

The first variable x is observed every 20 steps and the operation function is defined as $h(x) = x^{\alpha} + \varepsilon$, where $\alpha = 1$ and the error variance of ε is $\boldsymbol{R} = 0.5$.

Suppose that there are only 9 particles and they are divided into sub-ensembles with $G = 3$ and $N = 3$. There is no inflation to the error covariance in the procedure.

In Fig. 1, the truth can be followed very well by only 9 particles in the method of 3DPF, whereas the particles in EnKF diverge severely. This shows that the deterministic method can outperform the classical perturbed observation method, especially when the number of particles is restricted. In Fig. 1(c), the standard PF performs badly after a period of time due to its lack of diversity of particles.

4.2. Nonlinear observation system

If the observation system becomes nonlinear, the filtering problem will be more challenging. In function $h(x) = x^{\alpha} + \varepsilon$, we set $\alpha > 1$ and ε is still Gaussian-distributed with variance $\boldsymbol{R} = 0.5$. Applied to nonlinear systems, the three methods yield very different performances as shown in Fig. 2. Here, the root mean square (RMS) is defined as

$$\mathrm{RMS} = \sqrt{\frac{1}{JT}\sum_{j=1}^{J}\sum_{i=1}^{\mathrm{T}}\left(x_{ij}^{\mathrm{a}} - x_{ij}^{\mathrm{t}}\right)^2}, \tag{30}$$

where J is the number of variables in the model system, T is the simulation time window, a indicates the analysis, t denotes the truth, and x_{ij} represents the value of j-th variable at time step i. We set the model error covariance $\boldsymbol{Q}' = 0.3\boldsymbol{Q}$, and the number of particles is augmented to 20 ($N = 4, G = 5$).

From Fig. 2, it can be shown that with α increasing, the performance difference among the three methods becomes bigger. The new algorithm deals very well with highly nonlinear observation systems.

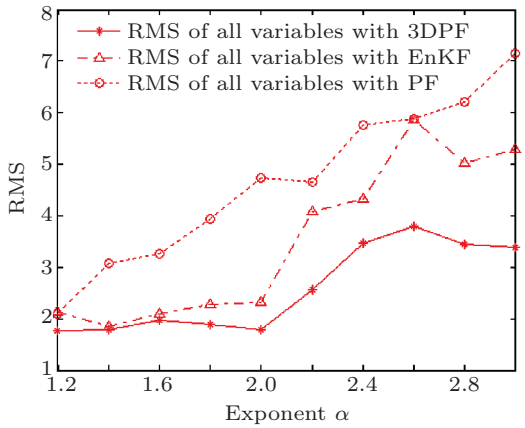


Fig. 2. (color online) Variations of RMS with exponent in the observation function h.

Figure 3 illustrates the performances of the methods when the model error covariance varies. It shows that when the observation system is highly nonlinear, here $\alpha = 3$, 3DPF performs much better than EnKF and PF in most cases, which indicates that our algorithm is very robust, and the smaller the model covariance, the more remarkable its superiority is. However, the performance difference between 3DPF and EnKF lessens as the observation error covariance increases.

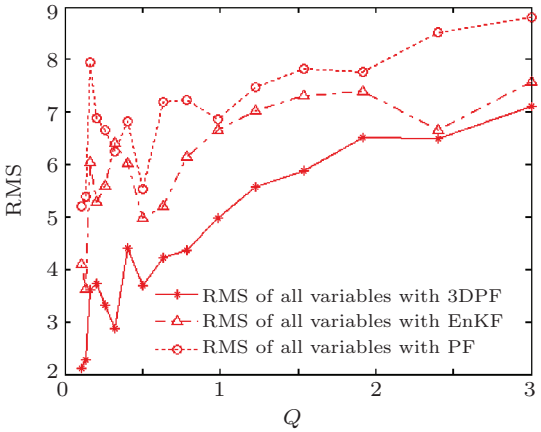


Fig. 3. (color online) Variations of RMS with model error covariance Q for three methods. The exponent in the observation function is fixed at 3.

It should be noted that if the analysis or forecast error covariance in 3DPF is multiplied by a proper inflation factor, or the ensemble is more appropriately divided into subensembles, the performance of the algorithm can be improved.

5. Conclusion

In this paper, we propose a new approach that combines the 3DVar and PF. Applied to a highly nonlinear chaos system, it performs much better than the traditional EnKF and the standard PF, with a computation cost similar to that of EnKF.

The simulation system used in this paper has low dimensions. In future work we wish to apply the new algorithm to a more realistic system, and make the distributions of the model error and the observation error non-Gaussian. Extending from 3DPF to 4DPF (with an additional dimension of time) is also challenging.

Appendix A

Substituting Eq. (12) into Eq. (5) and linearizing the observation function like in Eq. (14) yield the identity

$$\begin{aligned} &J(\boldsymbol{x}^{\mathrm{b}} + \boldsymbol{X}^{\mathrm{b}}\boldsymbol{w}) \\ &= \widehat{J}(\boldsymbol{w}) - (N-1)\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{I} - (\boldsymbol{X}^{\mathrm{b}})^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{b}}(\boldsymbol{X}^{\mathrm{b}})^{\mathrm{T}})^{-1}\boldsymbol{X}^{\mathrm{b}})\boldsymbol{w}. \end{aligned} \tag{A1}$$

Assume $\boldsymbol{X}^{\mathrm{b}}$ to be a linear transformation from an $N-$ dimensional space $\tilde{\mathrm{S}}$ to an $(N-1)$-dimensional space S spanned by the columns of $\boldsymbol{X}^{\mathrm{b}}$.

From Eqs. (19) and (20), we know that the row vectors in $\boldsymbol{P}^{\mathrm{w,a}}(\boldsymbol{Y}^{\mathrm{b}})^{\mathrm{T}}\boldsymbol{R}^{-1}$ are linearly dependent on each other, thus the first term on the right-hand side of Eq.(A1) depends only on the components of $\boldsymbol{w}$ in S, while the second term depends only on its components in the null space orthogonal to S, because of the following equation:

$$\boldsymbol{X}^{\mathrm{b}}(\boldsymbol{I} - (\boldsymbol{X}^{\mathrm{b}})^{\mathrm{T}}(\boldsymbol{X}^{\mathrm{b}}(\boldsymbol{X}^{\mathrm{b}})^{\mathrm{T}})^{-1}\boldsymbol{X}^{\mathrm{b}}) = \boldsymbol{X}^{\mathrm{b}} - \boldsymbol{X}^{\mathrm{b}} = \boldsymbol{0}. \tag{A2}$$

Generally, the null space has only one dimension, and it is spanned by the vector $(1,1,\ldots,1)^{\mathrm{T}}$.

Thus, if we obtain the solution $\bar{\boldsymbol{w}}$ by minimizing $\widehat{J}(\boldsymbol{w})$, we have $\bar{\boldsymbol{w}}^{\mathrm{T}} \cdot (1,1,\ldots,1)^{\mathrm{T}} = 0$, that is, the second term on the right-hand side in Eq. (A1) should be zero. Then the corresponding $\boldsymbol{x}^{\mathrm{a}}$ minimizes the cost function $J(\boldsymbol{x})$.

References

[1] Eubank R L 2006 *A Kalman Filter Primer* (Boca Raton, Fla.: Chapman & Hall/CRC)

[2] Speyer J L and Chung W H 2008 *Stochastic Processes, Estimation, and Control* (Philadelphia: Society for Industrial and Applied Mathematics)

[3] Psiaki M L 2005 *Journal of Guidance, Control, and Dynamics* **28** 885

[4] Houtekamer P L and Mitchell H L 2005 *Quart. J. R. Meteorol. Soc.* **131** 3269

[5] Hunt B R, Kostelich E J and Szunyogh I 2007 *Physica D* **230** 112

[6] Reich S 2012 *Quart. J. R. Meteorol. Soc.* **138** 222

[7] Zhang Z T and Zhang J S 2010 *Chin. Phys. B* **19** 104601

[8] Arulampalam M S, Maskell S, Gordon N and Clapp T 2002 *IEEE Trans. Signal Process.* **50** 174

[9] Schon T, Gustafsson F and Nordlund P 2005 *IEEE Trans. Signal Process.* **53** 2279

[10] Eyink G L and Kim S 2006 *J. Stat. Phys.* **123** 1071

[11] Nakano S, Ueno G and Higuchi T 2007 *Nonlin. Processes Geophys.* **14** 395

[12] Vanleeuwen P J 2010 *Quart. J. R. Meteorol. Soc.* **136** 1991

[13] Bi J, Guan W and Qi L T 2012 *Chin. Phys. B* **21** 068901

[14] Kunihama T, Omori Y and Zhang Z 2012 *J. Time Series Anal.* **33** 61

[15] Kotceha J H and Djuric P M 2003 *IEEE Trans. Signal Process.* **51** 2592

[16] Merwe R V D and Wan E 2003 *Proceedings of the IEEE International Conference on Acoustics Speech, and Signal Processing* (Hongkong) p. 701

[17] Míguez J 2006 *Digital Signal Processing* **9** 1

[18] Won S P, Melek W W and Golnaraghi F 2010 *IEEE Trans. Industrial Electron.* **57** 1787

[19] Won S P, Melek W W and Golnaraghi F 2011 *Meas. Sci. Technol.* **22** 125108

[20] Hoteit I, Luo X and Pham D 2012 *Mon. Wea. Rev*. **140** 528

[21] Stordal A, Karlsen H, Nævdal G, Skaug H and Vallès B 2011 *Computational Geosciences* **15** 293

[22] Edwards AW F 1972 *Likelihood* (Cambridge: Cambridge University Press)

[23] Vanleeuwen P J 2009 *Mon. Wea. Rev.* **137** 4089

[24] Zupanski M, Navon I M and Zupanski D 2008 *Q. J. R. Meteorol. Soc.* **134** 1039

[25] Zupanski M 2005 *Mon. Wea. Rev.* **133** 1710