Sample Regenerating Particle Filter Combined With Unequal Weight Ensemble Kalman Filter for Nonlinear Systems

Li, Xiao; Cheng, Ai Jie; Lin, Hai Xiang

DOI
10.1109/ACCESS.2021.3100486

Publication date
2021

Document Version
Final published version

Published in
IEEE Access

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 1, 2021, accepted July 12, 2021, date of publication July 26, 2021, date of current version August 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3100486

Sample Regenerating Particle Filter Combined With Unequal Weight Ensemble Kalman Filter for Nonlinear Systems

XIAO LI[1,2], **AI JIE CHENG**[1], **AND HAI XIANG LIN**[2]
[1]School of Mathematics, Shandong University, Jinan, Shandong 250100, China
[2]Delft Institute of Applied Mathematics, Delft University of Technology, 2628 CD Delft, The Netherlands

Corresponding author: Hai Xiang Lin (h.x.lin@tudelft.nl)

The work of Xiao Li was supported in part by the Visitor's Grant of China Scholarship Council, and in part by the National Natural Science Foundation of China under Grant 11971272.

ABSTRACT We present an approach which combines the sample regenerating particle filter (SRGPF) and unequal weight ensemble Kalman filter (UwEnKF) to obtain a more accurate forecast for nonlinear dynamic systems. Ensemble Kalman filter assumes that the model errors and observation errors are Gaussian distributed. Particle filter has demonstrated its ability in solving nonlinear and non-Gaussian problems. The main difficulty for the particle filter is the curse of dimensionality, a very large number of particles is needed. We adopt the idea of the unequal weight ensemble Kalman filter to define a proposal density for the particle filter. In order to keep the diversity of particles, we do not apply resampling as the traditional particle filter does, instead we regenerate new samples based on a posterior distribution. The performance of the combined sample regenerating particle filter and unequal weight ensemble Kalman filter algorithm is evaluated using the Lorenz 63 model, the results show that the presented approach obtains a more accurate forecast than the ensemble Kalman filter and weighted ensemble Kalman filter under Gaussian noise with dense observations. It still performs well in case of sparse observations though more particles are required. Furthermore, for non-Gaussian noise, with an adequate number of particles, the performance of the approach is much better than the ensemble Kalman filter and more robust to noise with nonzero bias.

INDEX TERMS Particle filter, Monte Carlo method, nonlinear dynamic systems, Lorenz function.

I. INTRODUCTION

Particle filter (PF) is a Monte Carlo method which calculates the state estimation based on the samples generated from the prior (model) or proposal distribution and obtains the full posterior distribution by combining model states and observations using Bayes' theorem. Unlike the Kalman filter which is based on a linear assumption, particle filter can solve nonlinear and even non-Gaussian distribution problems. In real world, many problems are nonlinear, and their analytical solutions are rarely available, so particle filter is developed to deal with these problems [1], [2], [3]. And it has been applied in many areas, such as air traffic control [4], [5], meteorology [6], [7], aerospace [8], [9], oceanography [10], autonomous vehicles [11] and robotics [12], remote sensing [13], computer vision [14], [15] and biomedical research [16].

We consider the dynamic model of the form:

$$x_t = \mathcal{F}(x_{t-1}) + \xi_t, \tag{1}$$

where $\mathcal{F}(\cdot)$ is the deterministic dynamic operator, x_t is the unobserved state vector of interest in n-dimensional space at time t, ξ_t is the stochastic part, also called model error which means the model is not perfect. For simplicity of analysis, we assume the errors in the model are Gaussian distributed with zero mean and known covariance. In practice, initial values at time $t = 0$ are not exactly known, they are estimated using empirical knowledge and modeled with a mean and known covariance. Furthermore, we assume observations are obtained from the measurement equipment, the relation between the state variables and observations can be described as follows:

$$y_t = \mathcal{H}(x_t) + \eta_t, \tag{2}$$

The associate editor coordinating the review of this manuscript and approving it for publication was Liang Hu.

where $\mathcal{H}(\cdot)$ is the deterministic observable operator, y_t is the observed m-dimensional vector at time t, η_t represents the stochastic observation error.

PF is a sequential Monte-Carlo technique which produces a set of states drawn from the posterior distribution of the system. In practice, particle filter is implemented by calculating the trajectory sequentially with pairs of particles as $\{x_{t-1}^i, w_{t-1}^i\}, i = 1, 2, \ldots, N$. In which w_{t-1}^i is the weight of the particle and is proportional to the posterior distribution of the variable states x_{t-1}^i. The states x_t^i are calculated iteratively using the previously estimated states x_{t-1}^i and observations. For each iteration, the particle filter can be divided into three phases: prediction, update and resampling. In the prediction phase, we obtain the prior samples according to the prior density. Next, update the weights of particles according to the posterior distribution. Then, duplicate particles with high weights in the resampling phase and abandon those with low weights, which result in a set of uniformly weighted particles $\{x_t^i, w_t^i = 1/N\}$ [17]. Finally, the analysis of the states is calculated as the mean of these particles.

Particle filter has been applied to many problems, but there are still some challenges remaining. And one of them is sample degeneracy [1], [18]-[21], where one particle has a weight close to one while the weights of all other particles are close to zero. This leads to the collapse of the method. A lot of work has been done to tackle this problem and they can be mainly divided into two categories [22]: A. Selecting a suitable proposal distribution; B. Resampling. In the following paragraphs, these two methods will be briefly described.

Selecting a suitable proposal distribution is very important for the posterior estimation but in practice it is very challenging. There are many different ways of choosing a proposal density, an overview of some commonly used algorithms can be found in [1]. The simplest way is to use transition density $p(x_t|x_{t-1})$ as the proposal density directly. And we term this as standard proposal density. Another possible choice for the proposal distribution is the optimal proposal density, which incorporates the new observations y_t when generating the particles at t. The 'optimal' here does not refer to the performance of the particle filter. It refers to the variance of w_t^i over different particles x_t^i can reach the minimum value which is zero [23]. In [23], Snyder also gave a simple analytical example that the degeneracy of particle filter is reduced with the optimal proposal relative to the standard proposal for sufficiently high-dimensional systems.

The other way to deal with the degeneracy of the particle filter is resampling. The basic idea of resampling is to choose particles according to their weights. After resampling, the updated particles are more concentrated in domains with higher posterior probability, the outcomes of the filter improve in some extent. There are different resampling approaches, for example, multinomial resampling [24], stratified/systematic resampling [25], residual resampling [26]. Li *et al.* [27] gave an overview of resampling algorithms, and discussed undesired effects of resampling. One of the effects is sample impoverishment. With resampling, particles with high weights can be repeatedly selected while particles with low weights are abandoned, and, thereby, the diversity of the particles is reduced [28]. In the worst case, we have only a few different particles in the set after resampling. In this paper, we do not use resampling, instead we generate new particles based on the posterior distribution. We do this to preserve the diversity of particles with the optimization criterion of minimizing the variance of the particle weights. It will be presented in section 3.

Ensemble Kalman filter (EnKF) has been used for huge dimensional state spaces in e.g., weather forecast and PF has demonstrated the ability to deal with a strong non-linear problem without Gaussian distribution assumption in a reduced dimensional state space. Some researchers have tried to combine PF with EnKF [29], [30] to improve the filter performance for nonlinear dynamic systems. In [30], Papadakis *et al.* proposed a method named as the weighted ensemble Kalman filter (WEnKF). The WEnKF method uses EnKF to define a proposal density given the history measurements for particle filter. In fact, it is a hybrid filtering procedure which uses EnKF to generate samples and uses PF to approximate the importance weights for them. The experiment results in [30] show that it outperforms the traditional EnKF with a comparable computational cost for a high-dimensional non-linear problem. There was a mistake in the derivation of weights computation as pointed out by van Leeuwen *et al.* [31] and this method is not suitable for high-dimensional systems.

In this paper, we propose an approach which uses EnKF as proposal density in a different way. We consider different weights of ensemble members instead of giving them the same equal weight. We do this to obtain a better approximation of mean and variance of state variables especially when the dynamic system is highly nonlinear or noise distribution is non-Gaussian. In the next step, PF is used to update the weights of particles and calculate the mean and variance of the posterior distribution of states. Instead of resampling, we regenerate particles based on the mean and variance of the posterior distribution in order to keep the diversity of particles.

Section 2 gives a brief review of the EnKF and introduces the unequal weight ensemble Kalman filter (UwEnKF). We describe the basic idea of particle filter and the algorithm of sample regenerating particle filter (SRGPF) in section 3. In section 4, we combine the UwEnKF and SRGPF together to improve the accuracy of the estimation of the state variables. Comparison between our method and several existing methods is presented using the Lorenz 63 model in section 5. Conclusions and future work are summarized in section 6.

II. ENSEMBLE KALMAN FILTER

A. THE ENSEMBLE KALMAN FILTER

The EnKF is proposed by Evensen [32] and later clarified by Burgers *et al.* [33]. Unlike the Kalman filter, which is used for linear system model and calculates the error covariance analytically, EnKF approximates the covariance using a set of ensemble members. We use the following equations as the dynamic model in the discussion.

$$\mathbf{x}_k = \mathcal{F}(\mathbf{x}_{k-1}) + \boldsymbol{\xi}_k, \quad \boldsymbol{\xi}_k \sim N(0, \mathbf{Q}) \tag{3}$$

$$\mathbf{y}_k = \mathcal{H}(\mathbf{x}_k) + \boldsymbol{\eta}_k, \quad \eta_k \sim N(0, \mathbf{R}) \tag{4}$$

where $\mathbf{x}_k, \boldsymbol{\xi}_k \in \mathbf{R}^{n_x}$, $\mathbf{y}_k, \boldsymbol{\eta}_k \in \boldsymbol{R}^{n_y}$, $\boldsymbol{\xi}_k$ and $\boldsymbol{\eta}_k$ are Gaussian distributed noise with zero-mean and covariance matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively. Furthermore, $\mathbf{x}_0$, $\boldsymbol{\xi}_k$ and $\boldsymbol{\eta}_k$ are uncorrelated.

Denote an ensemble as $\mathbf{X}_k^f \in \mathbf{R}^{n_x \times N}$, $\mathbf{X}_k^f = (\mathbf{x}_{1,k}, \ldots, \mathbf{x}_{i,k}, \ldots, \mathbf{x}_{N,k})$, $i = 1, 2, \ldots, N$, N is the ensemble size [34].

$$\mathbf{x}_{i,k}^f = \mathcal{F}(\mathbf{x}_{i,k-1}^a) + \boldsymbol{\xi}_{i,k}, \tag{5}$$

the mean of the ensemble can be calculated as follows:

$$\overline{x_k^f} = \frac{1}{N}\sum_{i=1}^{N} x_{i,k}^f, \quad \overline{\mathcal{H}(x_k^f)} = \frac{1}{N}\sum_{i=1}^{N} \mathcal{H}(x_{i,k}^f). \tag{6}$$

And the covariance matrices can be approximated by

$$\mathbf{P}_{x_k x_k}^f = \frac{1}{N-1}\sum_{i=1}^{N}(x_{i,k}^f - \overline{x_k^f})(x_{i,k}^f - \overline{x_k^f})^T. \tag{7}$$

$$\mathbf{P}_{x_k \mathcal{H}(x_k)} = \frac{1}{N-1}\sum_{i=1}^{N}(x_{i,k}^f - \overline{x_k^f})(\mathcal{H}(x_{i,k}^f) - \overline{\mathcal{H}(x_k^f)})^T. \tag{8}$$

$$\mathbf{P}_{\mathcal{H}(x_k)\mathcal{H}(x_k)} = \frac{1}{N-1}\sum_{i=1}^{N}(\mathcal{H}(x_{i,k}^f) - \overline{\mathcal{H}(x_k^f)})(\mathcal{H}(x_{i,k}^f) - \overline{\mathcal{H}(x_k^f)})^T. \tag{9}$$

If $\mathcal{H}(\cdot)$ is linear, then the expression of the classical Kalman filter gain can be used for the ensemble filter gain.

$$\mathbf{K}_k = \mathbf{P}_{x_k x_k}^f \mathbf{H}^T (\mathbf{H}\mathbf{P}_{x_k x_k}^f \mathbf{H}^T + \mathbf{R})^{-1}, \tag{10}$$

If $\mathcal{H}(\cdot)$ is nonlinear,

$$\mathbf{K}_k = \mathbf{P}_{x_k \mathcal{H}(x_k)}(\mathbf{P}_{\mathcal{H}(x_k)\mathcal{H}(x_k)} + \mathbf{R})^{-1}. \tag{11}$$

In practice, a nonlinear observation operator $\mathcal{H}(\cdot)$ is often approximated by linearization, the Jacobian matrix of $\mathcal{H}(\cdot)$ is then used for $\mathbf{H}$ in (10),

$$\mathbf{H} = \left.\frac{\partial \mathcal{H}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x_k}} \tag{12}$$

For a strong nonlinear operator $\mathcal{H}(\cdot)$ or when the Jacobian is hard to obtain, (11) is used to compute the Kalman gain.

In the analysis step, the following update is performed for each ensemble member of the model state by

$$x_{i,k}^a = x_{i,k}^f + \mathbf{K}_k(y_k^o + \boldsymbol{\eta}_{i,k} - \mathcal{H}(x_{i,k}^f)). \tag{13}$$

The analysis state or best estimation of the state at time k is

$$x_k^a = \frac{1}{N}\sum_{i=1}^{N} x_{i,k}^a. \tag{14}$$

In the traditional EnKF, every sample has the same weight across all time steps. Consider in the initial step every sample is generated with Gaussian error distribution and each sample has the same weight. For a linear or weakly nonlinear system, the posterior density $p(x_k|y_{1:k})$ is or close to a Gaussian distribution, so it is just a scaling problem which transforms prior Gaussian distribution to posterior Gaussian distribution. Therefore, after normalization, the weights of ensemble members (samples) are the same as in the first step (initialization). While for nonlinear model, the ensemble posterior is non-Gaussian distributed. It is unlikely that all ensemble members still have the same weights as the previous step. Therefore, for strongly nonlinear problem, it is necessary to recalculate the weight of every ensemble member appropriately.

B. UNEQUAL WEIGHT ENSEMBLE KALMAN FILTER (UwEnKF)

In the following, we describe unequal weight ensemble Kalman filter and show how the weights are calculated. From (5), we have

$$p(x_{i,k}^f | x_{i,k-1}^a) \sim N(\mathcal{F}(x_{i,k-1}^a), \mathbf{Q}), \tag{15}$$

So,

$$p(x_{i,k}^f | x_{i,k-1}^a) \propto e^{-\frac{1}{2}(x_{i,k}^f - \mathcal{F}(x_{i,k-1}^a))\mathbf{Q}^{-1}(x_{i,k}^f - \mathcal{F}(x_{i,k-1}^a))^T}. \tag{16}$$

Normalize such that the sum of the weights equals 1,

$$p(x_{i,k}^f) = p(x_{i,k}^f | x_{i,k-1}^a) / \sum_{i=1}^{N} p(x_{i,k}^f | x_{i,k-1}^a). \tag{17}$$

Calculate the mean as follows instead of using (6),

$$\overline{x_k^f} = \sum_{i=1}^{N} p(x_{i,k}^f) x_{i,k}^f, \quad \overline{\mathcal{H}(x_k^f)} = \sum_{i=1}^{N} p(x_{i,k}^f)\mathcal{H}(x_{i,k}^f). \tag{18}$$

And replace (7)- (9) by

$$\mathbf{P}_{x_k x_k}^f = \sum_{i=1}^{N} p(x_{i,k}^f)(x_{i,k}^f - \overline{x_k^f})(x_{i,k}^f - \overline{x_k^f})^T, \tag{19}$$

$$\mathbf{P}_{x_k \mathcal{H}(x_k)} = \sum_{i=1}^{N} p(x_{i,k}^f)(x_{i,k}^f - \overline{x_k^f})(\mathcal{H}(x_{i,k}^f) - \overline{\mathcal{H}(x_k^f)})^T, \tag{20}$$

$$\mathbf{P}_{\mathcal{H}(x_k)\mathcal{H}(x_k)} = \sum_{i=1}^{N} p(x_{i,k}^f)(\mathcal{H}(x_{i,k}^f) - \overline{\mathcal{H}(x_k^f)})(\mathcal{H}(x_{i,k}^f) - \overline{\mathcal{H}(x_k^f)})^T. \tag{21}$$

The Kalman gain and the ensemble can still be calculated according to (10) and (13), respectively. We can rewrite (14) in section II-A as follows:

$$x_k^a = \overline{x_k^f} + \mathbf{K}_k(y_k^o - \overline{\mathcal{H}(x_k^f)}). \tag{22}$$

(22) gives the estimation of x_k, and the covariance of the analysis ensemble is calculated by

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{x_k x_k}^f. \tag{23}$$

III. PARTICLE FILTER

A. BASIC PARTICLE FILTER

Consider the problem of forecast, we want to get the estimation of x_k given the measurements $\{y_1, y_2, \ldots, y_k\}$. Under the Bayesian rule, we have [35]:

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)}{p(y_k)} \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}, \tag{24}$$

with the likelihood $p(y_k|x_k)$ and the prior distribution $p(x_k|x_{k-1})$. Considering the Monte Carlo method by drawing random samples from a given distribution,

$$p(x_k) = \sum_{i=1}^{N} w_{i,k}\delta(x_k - x_{i,k}), \tag{25}$$

where i represents the ith particle and N is the number of particles. The particle filter approximates the probability density (24) by a set of particles $\{x_{i,k}, w_{i,k}\}_{i=1}^{N}$.

$$\begin{aligned} p(x_k|y_{1:k}) &= \sum_{i=1}^{N} w_{i,k}\delta(x_k - x_{i,k}) \\ &= \sum_{i=1}^{N} w_{i,k-1}\frac{p(y_k|x_{i,k})}{p(y_k)}p(x_{i,k}|x_{i,k-1})\delta(x_k - x_{i,k-1}), \end{aligned} \tag{26}$$

and we denote

$$w_{i,k} = w_{i,k-1}\frac{p(y_k|x_{i,k})}{p(y_k)}p(x_{i,k}|x_{i,k-1}). \tag{27}$$

Sometimes we cannot draw samples from the prior distribution directly, in that case a proposal transition density $q(x_k|x_{k-1}, y_k)$ is introduced,

$$\begin{aligned} p(x_k|y_{1:k}) = \frac{p(y_k|x_k)}{p(y_k)} \int & \frac{p(x_k|x_{k-1})}{q(x_k|x_{k-1}, y_k)} \\ & \times q(x_k|x_{k-1}, y_k)p(x_{k-1}|y_{1:k-1})dx_{k-1}, \end{aligned} \tag{28}$$

The optimal proposal density function $q(x_k|x_{k-1}, y_k)$ is equal to $p(x_k|x_{k-1}, y_k)$. So, we draw random samples from the proposal density distribution $p(x_k|x_{k-1}, y_k)$. Equation (27) for the weight then becomes,

$$w_{i,k} = w_{i,k-1}\frac{p(y_k|x_{i,k})p(x_{i,k}|x_{i,k-1})}{p(y_k)p(x_{i,k}|x_{i,k-1}, y_k)}. \tag{29}$$

Now, we have got the pairs of particle with the associated weight, $\{x_{i,k}, w_{i,k}\}_{i=1}^{N}$, the next step is resampling. In resampling, a new set of N particles with equal weight is generated from the posterior distribution. The basic idea is that particles with high weights are multiplied while the ones with low weights are abandoned. The total number of particles after the resampling remains N each with a weight of $1/N$.

Algorithm 1: Sample Importance Resampling Particle Filter (SIRPF)

```
Result: $\left[\{x_{i,k}, w_{i,k}\}_{i=1}^{N}\right] =$
        SIRPF$\left[\{x_{i,k-1}, w_{i,k-1}\}_{i=1}^{N}, y_k\right]$
1. Importance sampling
for i = 1:N
    sample $x_{i,k} \sim q(x_k|x_{i,k-1}, y_k)$,
    evaluate the importance weight:
    $w_{i,k} \propto w_{i,k-1}\frac{p(y_k|x_{i,k})p(x_{i,k}|x_{i,k-1})}{q(x_{i,k}|x_{i,k-1}, y_k)}$,
end for
for i = 1:N
    normalize importance weight:
    $w_{i,k} = w_{i,k}/\sum_{i=1}^{N} w_{i,k}$,
end for
2.Resampling
effective particle set size $\hat{N_{eff}} = \frac{1}{\sum_{i=1}^{N}(w_{i,k})^2}$,
if $\hat{N_{eff}} < N_{thr}$ then
    $(x_{i,k}, w_{i,k})_{i=1}^{N} = resample(x_{i,k}, w_{i,k})_{i=1}^{N}$,
    $\hat{x_k} = \sum_{i=1}^{N} w_{i,k}x_{i,k}$,
end if
```

B. SAMPLE REGENERATING PARTICLE FILTER (SRGPF)

Degeneracy is a serious deficiency of the particle filter. The main reason for this is that in (29) the weight at time t_k depends on that at the previous time step. When t increases, most $w_{i,t}$ tend to approach zero and only one particle is left with a weight approaching 1, this leads to the collapse of the method.

Resampling is sometimes used to moderate the deficiency of PF as in Algorithm 1. An alternative approach to avoid this weight degeneracy is the following. Instead of using particles directly from time step t_k to time step t_{k+1}. The particles are regenerated after each filter step. The sample regenerating particle filter (SRGPF) is shown in Algorithm 2.

C. WHY NOT RESAMPLING?

Resampling is an efficient way to reduce the sample degeneracy, but it introduces bias in the estimation. There will always be errors when a set of samples is used to represent the entire distribution, especially when the set is small. Consequently when particles are sampled from this set, bias between estimation and truth of states occurs. The aim of resampling is to make every particle to have the same weight, so why don't we choose the particles from the posterior distribution instead of from the sample set?

However, sometimes it is not easy to draw samples from the posterior distribution. But we can approximate the mean and variance of the posterior distribution, so we can choose to sample from a Gaussian distribution which has the same mean and covariance as the posterior distribution. Note that we do not assume the posterior distribution of the state is Gaussian. Our aim is to regenerate particles according to the known mean and variance of state variables to avoid sample

Algorithm 2: Sample Regenerating Particle Filter (SRGPF)

```
Result: $\left[\{x_{i,k}, w_{i,k}\}_{i=1}^{N}\right] = \text{SRGPF}\left[\{x_{i,k-1}, w_{i,k-1}\}_{i=1}^{N}, y_k\right]$
1. Importance sampling
for i = 1:N
    sample $x_{i,k} \sim q(x_k|x_{i,k-1}, y_k)$,
    evaluate the importance weight:
    $w_{i,k} \propto w_{i,k-1}\frac{p(y_k|x_{i,k})p(x_{i,k}|x_{i,k-1})}{q(x_{i,k}|x_{i,k-1},y_k)}$,
end for
for i = 1:N
    normalize importance weight:
    $w_{i,k} = w_{i,k}/\sum_{i=1}^{N} w_{i,k}$,
end for
2. Regenerate particles
    Calculate the mean and variance of the posterior distribution:
    $\hat{x_k} = \sum_{i=1}^{N} w_{i,k}x_{i,k}$,
    $\mathbf{P}_k = \sum_{i=1}^{N} w_{i,k}(x_{i,k} - \hat{x_k})(x_{i,k} - \hat{x_k})^T$,
    sample $(x_{i,k}, w_{i,k})_{i=1}^{N} \sim N(\hat{x_k}, \mathbf{P}_k)$
```

impoverishment. And Gaussian distribution has the highest probability around the mean value, i.e., the analysis state ($\hat{x_k}$ in section 3.2).

IV. COMBINING UNEQUAL WEIGHT ENSEMBLE KALMAN FILTER WITH SAMPLE REGENERATING PARTICLE FILTER (UwEnKF-SRGPF)

A. REVIEW OF WEIGHTED ENSEMBLE KALMAN FILTER

The weighted ensemble Kalman filter (WEnKF) was proposed by Papadakis *et al.* in [30]. The algorithm uses ensemble Kalman filter as proposal density for the particle filter. Experiments with a synthetic 2-D turbulence model show that when observation is given at every time step, the results of WEnKF and EnKF are very close to each other. As the observations become sparser, e.g., one observation at every 5 timesteps, WEnKF has a faster convergence and a more accurate estimation than EnKF. The WEnKF filter is more suitable for the non-Gaussian distribution of ensemble members, which may be caused by nonlinear stochastic dynamics over time [36]. The algorithm can be divided into two parts. First, use the EnKF to define a proposal density. Second, use particle filter to calculate the weight of every ensemble member (particle) and approximate the analysis state.

The weighted ensemble Kalman filter algorithm is as follows:

Algorithm 3: Weighted Ensemble Kalman Filter (WEnKF)

```
Result: $\left[\{x_{i,k}, w_{i,k}\}_{i=1}^{N}\right] = \text{WEnKF}\left[\{x_{i,k-1}, w_{i,k-1}\}_{i=1}^{N}, y_k\right]$
1. Initialization ($k = 0$).
Generating initial values with N ensemble members
    $[x_{1,0}, x_{2,0}, \ldots, x_{N,0}] \sim N(x_0, \mathbf{P}_0)$.
2. For k = 1, 2, ...
  a) Prediction step
  i. Evolve each ensemble member forward using (5).
  ii. Get the expectation and covariance of the state
  variables according to (6) - (9).
  b) Update step
  i. Calculate the Kalman gain according to (10).
  ii. Update the ensemble according to (13) and use them
  as input for Algorithm 1.
```

B. THE ALGORITHM OF UwEnKF-SRGPF

If the posterior PDF is known and it is easy to draw samples from this PDF, then independent samples with equal weights can be drawn. For example, a Gaussian prior combined with a linear Gaussian likelihood, such as EnKF does [22]. In reality we do not know the exact posterior distribution, so we draw samples from a proposal density function. There are many ways to choose the proposal density, such as a relaxation scheme [22], weighted ensemble Kalman filter [30], optimal proposal density [37], implicit particle filter [38], equivalent-weights particle filter [39] and implicit equal-weights particle filter [10]. In this paper, we use the unequal weight ensemble Kalman filter (UwEnKF) as the proposal density to generate samples. Then, we use the particle filter to correct the weight of these samples.

We start with introducing the method of WEnKF which uses EnKF to define the proposal density, as this forms the basis of our UwEnKF-SRGPF method. For WEnKF, an ensemble is generated as follows:

$$x_{i,k}^{a} = x_{i,k}^{f} + \mathbf{K}_k(y_k^o + \boldsymbol{\eta}_{i,k} - \mathcal{H}(x_{i,k}^{f})). \tag{30}$$

Following this, we use PF to update the weights of the ensemble. Assume the model and observation errors are Gaussian distributed and independent, when $\mathcal{H}$ is a linear operator we have [31]

$$p(x_{i,k}^{a}|x_{i,k-1}, y_k) \propto e^{-1/2(x_{i,k}^{a}-\mu_{i,k})\Sigma_k^{-1}(x_{i,k}^{a}-\mu_{i,k})^T}, \tag{31}$$

$$\mu_{i,k} = \mathcal{F}(x_{i,k-1}^{a}) + \mathbf{K}_k(y_k - \mathbf{H}(\mathcal{F}(x_{i,k-1}^{a}))), \tag{32}$$

$$\Sigma_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{Q}(\mathbf{I} - \mathbf{K}_k\mathbf{H})^T + \mathbf{K}_k\mathbf{R}\mathbf{K}_k^T. \tag{33}$$

Slightly different from WEnKF, when we use the UwEnKF to define the proposal density, we generate the ensemble $x_{i,k}^{a}$ from the distribution $N(x_k^a, P_k^a)$ as we have obtained in section II-B. Next, update the weights of particles according to (29). The advantage is that we do not need to calculate the proposal density as WEnKF does. Since it is not easy to obtain the proposal density as it needs the inverse of Σ_k which incurs complex computations. Now the weight can be calculated as:

$$w_{i,k} \propto w_{i,k-1} \frac{p(y_k^o|x_{i,k}^a)p(x_{i,k}^a|x_{i,k-1}^a)}{p(y_k)}. \tag{34}$$

The following relation between the likelihood and the prior yields,

$$\begin{aligned} p(y_k^o|x_{i,k}^a) &\propto e^{-1/2(y_k^o-\mathcal{H}(x_{i,k}^a))\mathbf{R}^{-1}(y_k^o-\mathcal{H}(x_{i,k}^a))^T}, \\ p(x_{i,k}^a|x_{i,k-1}^a) &\propto e^{-1/2(x_{i,k}^a-\mathcal{F}(x_{i,k-1}^a))\mathbf{Q}^{-1}(x_{i,k}^a-\mathcal{F}(x_{i,k-1}^a))^T}. \end{aligned} \tag{35}$$

According to (34) - (35), we get

$$w_{i,k} \propto w_{i,k-1} p(y_k^o|x_{i,k}^a)p(x_{i,k}^a|x_{i,k-1}^a), \tag{36}$$

$$w_{i,k} = w_{i,k} / \sum_{i=1}^{N} w_{i,k}, \tag{37}$$

The final analyzed estimation and covariance of the system are

$$\hat{x}_k = \sum_{i=1}^{N} w_{i,k} x_{i,k}^a, \tag{38}$$

$$\mathbf{P}_k = \sum_{i=1}^{N} w_{i,k}(x_{i,k}^a - \hat{x}_k)(x_{i,k}^a - \hat{x}_k)^T. \tag{39}$$

For sequential importance resampling (SIR) method, after resampling, $x_{i,k}^a$ is directly used as prior samples in the next step. Because some particles have duplications, so these particles have a higher weight than others. We regenerate particles to make sure at every time step each ensemble member has the same weight of $1/N$.

$$x_{i,k} \sim N(\hat{x}_k, \mathbf{P}_k) \tag{40}$$

Our UwEnKF-SRGPF algorithm (Algorithm 4) can be described as follows:

V. ASSIMILATION EXPERIMENTS

The Lorenz 63 model equation was developed by Edward Lorenz in 1963 as a simplified mathematical model for atmospheric convection.

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), & (41a) \\ \frac{dy}{dt} = x(\rho - z) - y, & (41b) \\ \frac{dz}{dt} = xy - \beta z, & (41c) \end{cases}$$

With the parameters $\sigma = 10$, $\beta = 8/3$, $\rho = 28$, this model has the famous chaotic behavior and is known as the Butterfly Attractor orbit.

Let $\Delta t = 0.01$, $\mathbf{x} = (x, y, z)^T$, $\mathbf{y} = (x_{obs}, y_{obs}, z_{obs})^T$, use a fourth-order Runge-Kutta scheme to integrate the model (41a)-(41c). Denote the model error at every time step as ξ. Furthermore, a noise term η is added to the observation. The dynamic system becomes:

$$\begin{cases} x_{k+1} = \mathbf{F}_k(\mathbf{x}_k) + \xi, & (42a) \\ \mathbf{y}_{k+1} = \mathbf{H}\mathbf{x}_{k+1} + \eta, & (42b) \end{cases}$$

Algorithm 4: Unequal Weight Ensemble Kalman Filter - Sample Regenerating Particle Filter (UwEnKF-SRGPF)

```
Result: $\left[\{x_{i,k}, w_{i,k}\}_{i=1}^N\right] =$
    UwEnKF-SRGPF$\left[\{x_{i,k-1}, w_{i,k-1}\}_{i=1}^N, y_k\right]$
1. Initialization ($k = 0$).
Generate initial values with N ensemble members
    $[x_{1,0}, x_{2,0}, \ldots, x_{N,0}] \sim N(x_0, \mathbf{P}_0)$.
2. For k = 1, 2, ...
  a) Prediction step
  i. Evolve the ensemble forward using (5).
  ii. Calculate and normalize the weight $p(x_{i,k}^f)$ according to (16) and (17).
  iii. Compute the mean and covariance of the state variables according to (18) and (19).
  b) Update step
  i. Calculate the Kalman gain according to (10) or (11).
  ii. Calculate the ensemble analysis by (22).
  iii. Generate samples from $N(x_k^a, \mathbf{P}_k^a)$ which has the mean and covariance as defined by (22) and (23).
  iv. Calculate the weights and normalize according to (35), (36) and (37).
  v. Obtain the analyzed state $\hat{x}_k$ and covariance $\mathbf{P}_k$ according to (38) and (39).
  c) Regenerate samples
  i. Regenerate samples $x_{i,k}^a \sim N(\hat{x}_k, \mathbf{P}_k)$ as input for $k + 1$ step.
```

Assume ξ and η have a diagonal covariance matrix with the diagonal elements $\sigma_{model} = \sigma_{obs} = 2$. $\mathbf{H}$ is an identity matrix. The real starting point is $(x_0, y_0, z_0) = (1.50887, -1.531271, 25.46091)$ while the initial guess is $\hat{\mathbf{x}}_0 = (1.0, -1.0, 27.0)^T$ and $\sigma_{initial} = 2$. The root mean square error (RMSE) of the analysis is defined as:

$$E_a = \sqrt{\frac{1}{T} \sum_{k=1}^{T} \left(\sum_{j=1}^{m} \left(x_{k,j}^{true} - x_{k,j}^a \right)^2 / m \right)}. \tag{43}$$

where $x_{k,j}$ is the jth variable of $\mathbf{X}_k$, the index m represents the number of assimilation steps, k represents the k-th time step, T represents the total number of time steps and we choose $T = 1000$ in the following experiments.

A. RESULTS

We evaluate the performance of our method UwEnKF-SRGPF and compare with that of EnKF and WEnKF. In [30], the authors smooth the ensemble members distribution by adding a zero mean Gaussian perturbation whose covariance depends on the mean discrepancy between the estimate and the current measurement. Without smoothing, for a small number of particles and large measurements latency the WEnKF was unstable and diverged sometimes. For a proper comparison, we do not use smoothing. And moreover,

TABLE 1. RMSE of analysis states under Gaussian noise with different observation frequencies.

Number	1-timestep observations			5-timesteps observations			10-timestep observations			20-timesteps observations		
	EnKF	WEnKF	U-SRGPF	EnKF	WEnKF	U-SRGPF	EnKF	WEnKF	U-SRGPF	EnKF	WEnKF	U-SRGPF
5	1.5334	1.5435	1.3573	2.2295	2.1552	2.3904	2.6765	2.6768	2.9626	3.7118	3.7920	4.1640
10	1.3759	1.3718	1.1531	1.8739	1.8650	1.9470	2.1094	2.0921	2.3470	2.9560	2.9740	3.2060
20	1.3330	1.3329	1.1023	1.7203	1.7040	1.7603	1.8920	1.8804	2.0165	2.6664	2.5981	2.8518
30	1.3208	1.3165	1.0965	1.6638	1.6596	1.7011	1.8014	1.7899	1.9245	2.4223	2.5045	2.6254
40	1.3158	1.3161	1.0900	1.6495	1.6474	1.6646	1.7606	1.7704	1.8574	2.3469	2.3944	2.6020
50	1.3132	1.3078	1.0935	1.6437	1.6353	1.6415	1.7258	1.7140	1.8301	2.3620	2.3314	2.5466
100	1.3069	1.3086	1.0894	1.6126	1.6031	1.5938	1.6820	1.6729	1.7185	2.2828	2.2495	2.3763
200	1.3063	1.3062	1.0893	1.5980	1.5970	1.5744	1.6636	1.6438	1.6715	2.2482	2.2333	2.1875
500	1.3051	1.3048	1.0883	1.5908	1.5830	1.5601	1.6391	1.6353	1.6204	2.2015	2.1832	2.1303

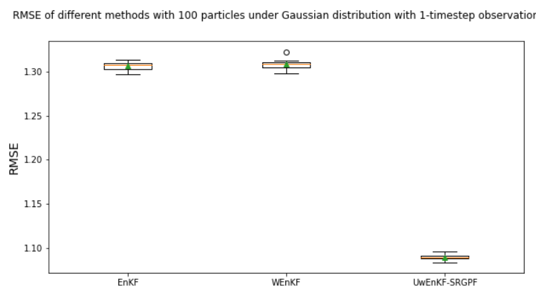


FIGURE 1. RMSE of EnKF, WEnKF, and UwEnKF-SRGPF with 100 particles and observations at every timestep under Gaussian noise.

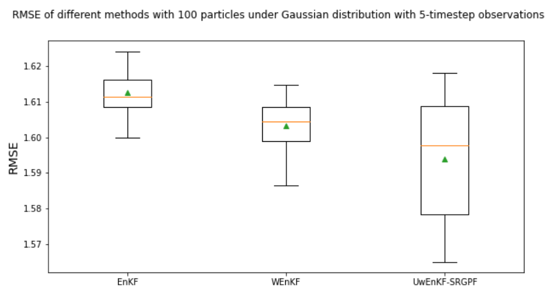


FIGURE 2. RMSE of EnKF, WEnKF, and UwEnKF-SRGPF with 100 particles and observations at every 5-timesteps under Gaussian noise.

for WEnKF, the proposal distribution is Gaussian which has the mean and covariance as defined in (22) and (23) respectively. We perform experiments with different noise distribution and observations at different time steps or frequencies.

1) GAUSSIAN DISTRIBUTED NOISE

Assume the errors of the initial value, the dynamic model and observations are all Gaussian distributed with zero means. For EnKF, it samples from the posterior distribution directly and gives all the ensemble members the same weight. For WEnKF, it uses EnKF to define a proposal density and corrects the weights of ensemble members (particles) by PF. While UwEnKF-SRGPF considers the weight of ensemble members to get a more accurate approximation of Kalman gain and regenerates particles instead of resampling. In the experiments, we performed test with different number of particles, namely $N = 5, 10, 20, 30, 40, 50, 100, 200, 500$, and with different observation frequencies. The RMSE is computed on 10 independent trials.

In Table 1, U-SRGPF denotes UwEnKF-SRGPF to save space. We can see when there is observation at every time step, UwEnKF-SRGPF performs the best. UwEnKF-SRGPF performs better than EnKF because the error caused by the linearization step of EnKF is higher than the error of UwEnKF-SRGPF which mainly comes from the representation error of the distribution by a small amount of samples. For WEnKF, resampling will lose diversity of the samples and samples with useful information may be discarded. When the observation frequency decreases to one at every 5 timesteps, UwEnKF-SRGPF needs more than 100 particles to compete with the other two methods. The reason is that for small number of particles the variance of weights is large, the distribution cannot be well represented under this condition. When the observations become sparser with only an observation at every 20 timesteps, UwEnKF-SRGPF can obtain better performance than the other two methods when the ensemble has 200 or more particles. This can be explained that UwEnKF-SRGPF needs more samples to approximate the posterior distribution, but with a sufficient large ensemble this approximation error becomes smaller than the linearization error in the EnKF and WEnKF methods under sparse observations. So, with UwEnKF-SRGPF there is a strong increase in computational effort when the observation frequency decreases. In the following, we will further analyze the performance of the three filter methods for cases when observations are at every 1, 5 and 10 timesteps, and the ensemble size is 100, and the case of non-Gaussian distributed errors will also be investigated using the exponential distributed noise.

Fig. 1 and 2 give RMSE of all three methods for 100 particles with 1 or 5 timestep observations under Gaussian noise. The red line is the median and the green triangle is the mean of the RMSE for ten trials. When the observation is at 1 timestep, the RMSE of the UwEnKF-SRGPF has a smaller variation over the trials as the interquartile range is smaller than EnKF and WEnKF. When the observation is at every 5 timestep, the variation of RMSE for UwEnKf-SRGPF is larger than the other two methods. But UwEnKf-SRGPF still performs well as the mean of RMSE for the 10 trials is lower than that of EnKF and WEnKF.

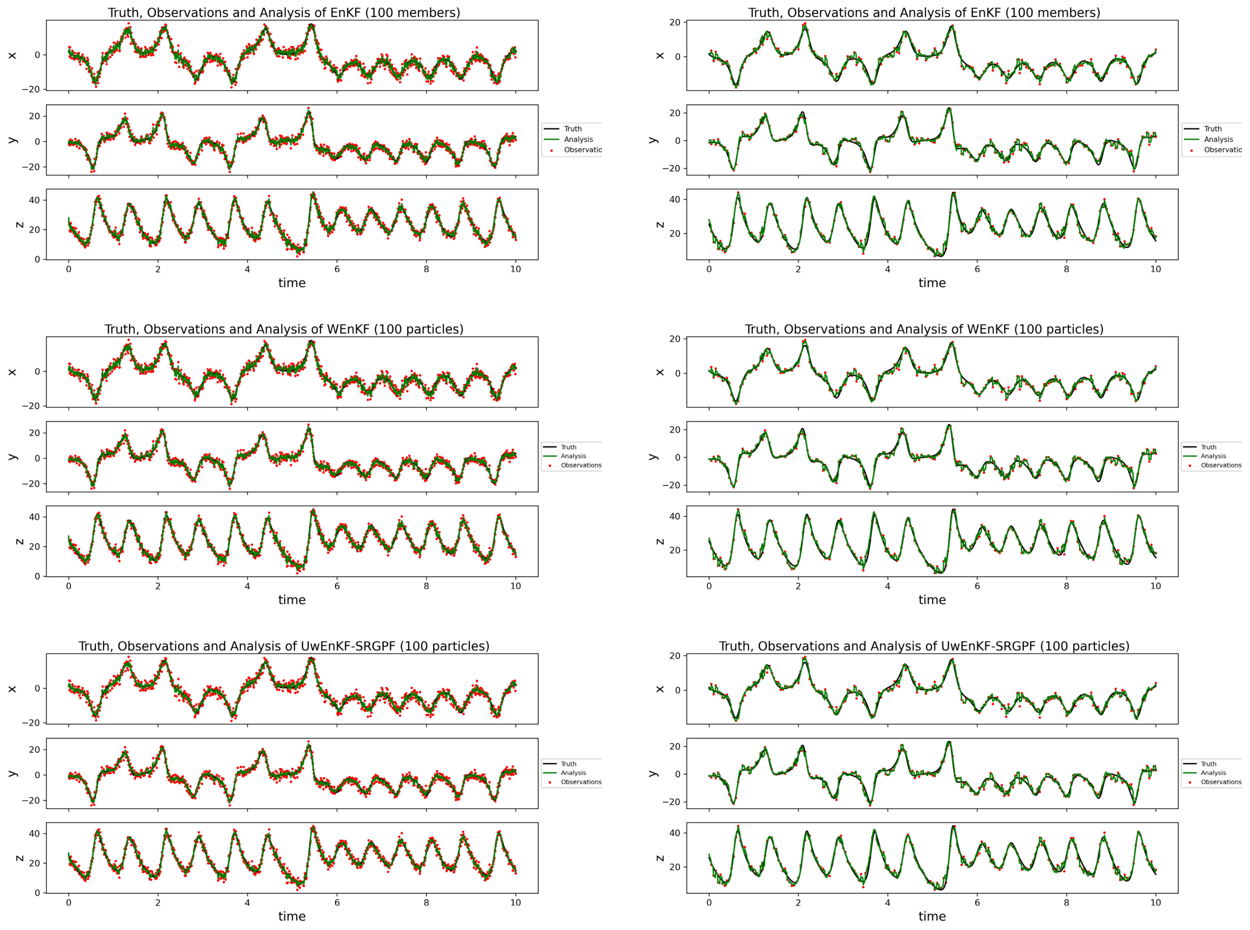


FIGURE 3. Assimilation experiments using EnKF, WEnKF, and UwEnKF-SRGPF with 100 particles by observation at every timestep (left) and every 5 timesteps (right) under Gaussian noise.

In Fig. 3, we present analysis and truth for all the three methods. We can see when the observation is at every timestep, the analysis trajectories of EnKF, WEnKF and UwEnKF-SRGPF can closely follow the truth. When the observation is at 5 timesteps, all the methods show small deviations from the truth. Whereas the analysis trajectories of UwEnKF-SRGPF are much closer to the truth. It implies that when the observation becomes sparse, the error caused by nonlinear transition dynamics becomes large and the performance of DA methods degrades.

From the above discussions, we conclude that UwEnKF-SRGPF have a better performance than EnKF and WEnKF with enough particles. When observation is at every timestep, the conclusion is obvious as shown in the table 1 since the RMSE of UwEnKF-SRGPF is the smallest among all the methods. When observations is at every 5 timestep, we cannot filter out the noise when observation is not available. For small ensemble sizes (less than or equal to 50 in Table 1), WEnKF and EnKF performs better than UwEnKF-SRGPF. It is because when the ensemble size is small, the variance of the weights is large, which leads to very inaccurate ensemble mean and covariance. Furthermore, UwEnKF-SRGPF encounters this problem twice as it also requires weights

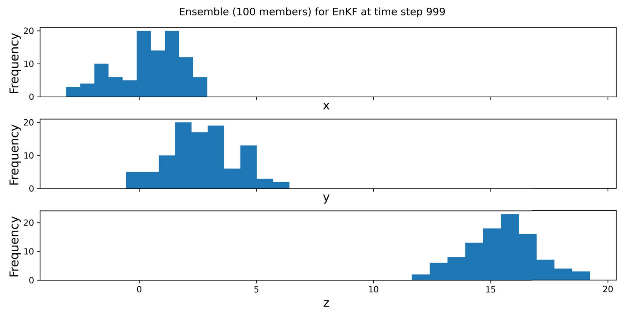


FIGURE 4. Histogram of posterior distribution for EnKF at $t = 9.99$ under Gaussian noise with 100 particles and 1-timestep observations.

calculation when using UwEnKF as the proposal density. As the ensemble sizes increase, the variance of weights is reduced and UwEnKF-SRGPF becomes the one with the best performance.

Fig. 4 - 6 show the ensemble distributions of the three methods at the 1000-th timestep (t = 10) with 1-timestep observations. The number of samples is 100 and they are generated from the posterior distribution at previous (999-th) timesteps which is t = 9.99. And the truth of Lorenz

TABLE 2. RMSE of analysis states under exponential noise with different observation frequencies.

Ensemble size	1-timestep observations		5-timestep observations		10-timestep observations	
	EnKF	UwEnKF-SRGPF	EnKF	UwEnKF-SRGPF	EnKF	UwEnKF-SRGPF
5	3.2616	7.2715	5.7180	9.1940	8.2173	9.8097
10	2.2430	4.0414	4.5524	6.4603	6.5415	7.4444
20	2.0510	1.9261	4.1295	4.8884	5.7389	6.5268
30	1.9299	1.3979	3.9185	4.0612	5.8055	5.3278
40	1.8893	1.2209	3.8380	3.6139	5.7150	5.0990
50	1.8955	1.0947	3.8699	3.7179	5.4706	5.2879
100	1.8329	0.9132	3.7706	3.1388	5.3561	4.7951
200	1.8049	0.8308	3.7399	2.8253	5.2756	4.5887
500	1.7923	0.7795	3.7220	2.6297	5.2826	4.3328
1000	1.7850	0.7690	3.7202	2.5445	5.2583	4.1619

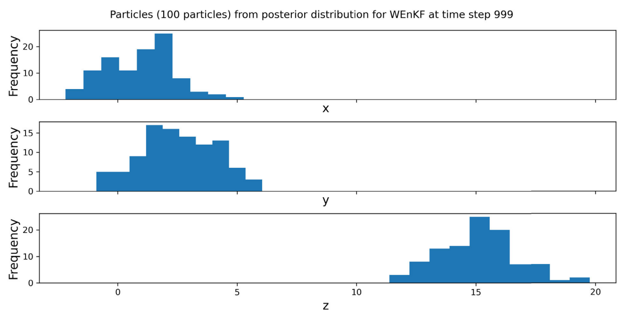


FIGURE 5. Histogram of posterior distribution for WEnKF at $t = 9.99$ under Gaussian noise with 100 particles and 1-timestep observations.

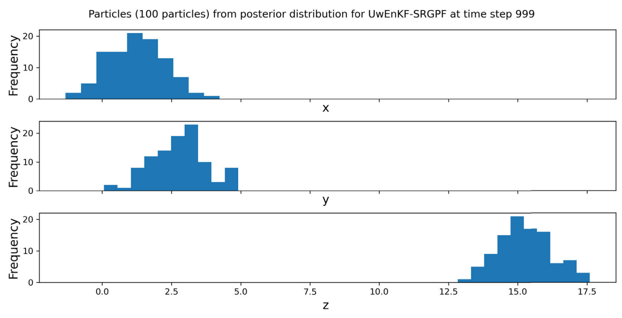


FIGURE 6. Histogram of posterior distribution for UwEnKF-SRGPF at $t = 9.99$ under Gaussian noise with 100 particles and 1-timestep observations.

63 at t $= 9.99$ is $(x, y, z) = (2.0735, 3.4608, 15.9068)$. From the figures, we can see that the posterior distributions at t $= 9.99$ of the three methods are different but they all cover the truth value of the model. The posterior distribution of UwEnKF-SRGPF is basically symmetrical around the truth value. For WEnKF, it is a little skewed and there is a bias between the mean of posterior distribution and the truth at the 999-th timesteps. The similarity between the results of UwEnKF-SRGPF and WEnKF is that both are unimodal distribution and basically symmetric, whereas for EnKF, the variance of posterior distribution is much higher. All ensemble members have equal weights in EnKF, occasionally some samples far from the truth are also kept resulting in a wide distribution as shown in the Fig. 4. UwEnKF-SRGPF and WEnKF update the weights according to a posterior distribution, the analysis is more accurate than EnKF. And

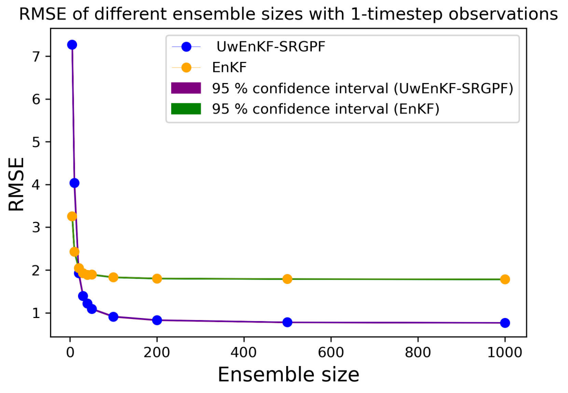


FIGURE 7. RMSE of different ensemble sizes with 1-timestep observations and exponential noise.

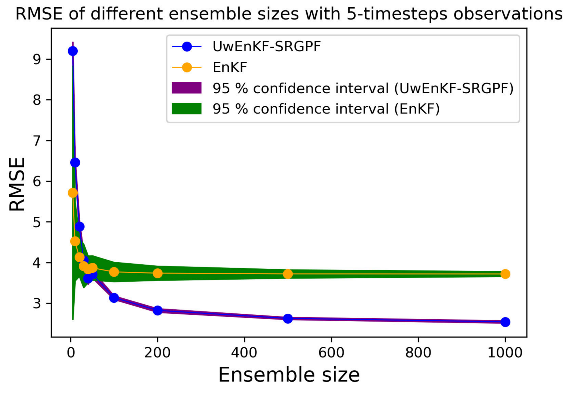


FIGURE 8. RMSE of different ensemble sizes with 5-timesteps observations and exponential noise.

UwEnKF-SRGPF is more accurate than WEnKF since the posterior distribution is more consistent with the system.

2) EXPONENTIAL DISTRIBUTED NOISE

Although Gaussian noise is usually used to model errors in data assimilation applications, in some cases it cannot properly represent the actual error which might be non-Gaussian. In order to investigate the performance of UwEnKF-SRGPF for non-Gaussian distributed errors, we will test our UwEnKF-SRGPF method and compare with EnKF by assuming the error in the dynamic model is exponential distributed. We use the same errors for the initial value and observations as in the experiment 1) with Gaussian noise. The

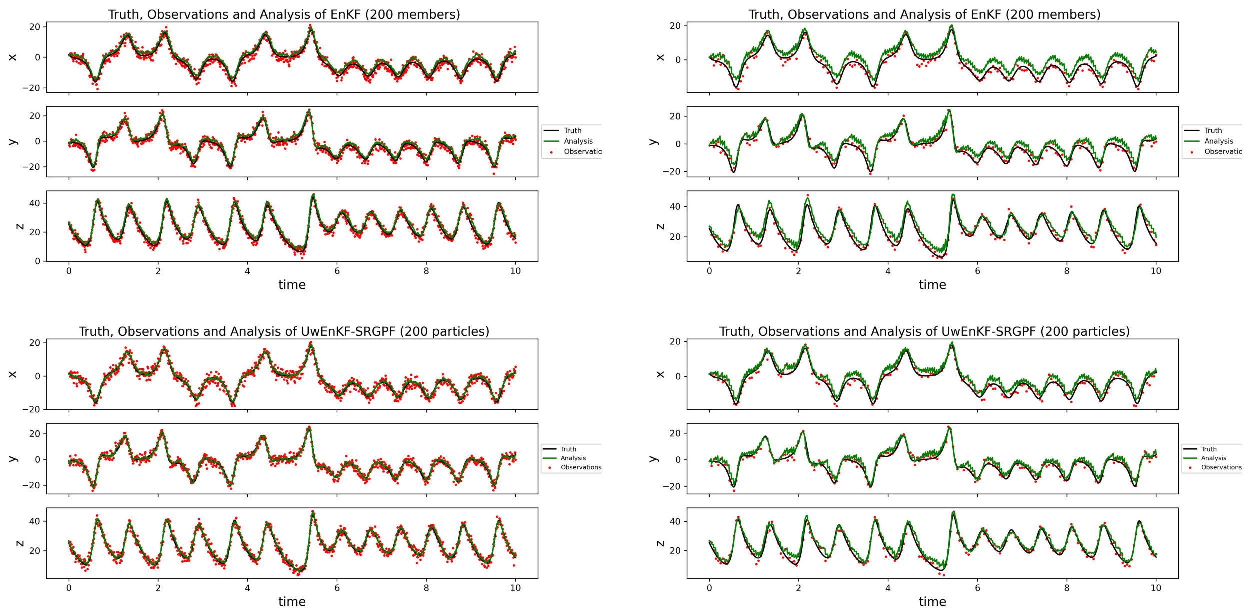


FIGURE 9. **Assimilation experiments using EnKF and UwEnKF-SRGPF with 200 particles by 1-timestep (left) and 5-timesteps (right) observations. The model noise is exponential and the observation noise is Gaussian.**

probability density function of exponential distribution is

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{44}$$

We choose $\lambda = 0.5$, the variance $1/\lambda^2 = 4$ is the same as the variance of errors in section V-A1. The mean is $1/\lambda = 2$, so the bias of model error is non-zero. Note that for exponential distribution, we cannot obtain the probability density function of the proposal distribution which is required for implementing WEnKF, so we do not include WEnKF in this comparison. For UwEnKF-SRGPF, we obtain the ensemble from the proposal density (UwEnKF), and generate particles from the Gaussian with the same mean and covariance as the ensemble. Since we cannot get the analytic pdf of the proposal density now. The reason we choose Gaussian distribution is that it has the high probability around the mean value and it is symmetric. In addition, UwEnKF-SRGPF will update the weights of particles according to their certainty.

Next, we do 10 independent experiments with an ensemble size of 5, 10, 20, 30, 40, 50, 100, 200, 500 and 1000. And no results are shown when observation frequency is at every 20 timestes since the estimations are not accurate for both methods. We use the average RMSE as a performance indicator for the methods. Results are shown in Table 2.

From Table 2, it can be seen that UwEnKF-SRGPF outperforms EnKF in almost all cases, except when the ensemble is very small. This is because that for a very small ensemble, a large representation error of the posterior distribution as explained in the previous section. When there is an observation at every timestep, the performance of UwEnKF-SRGPF increases with the ensemble size and is better than EnKF. When observations are getting sparser, at every 5 or 10 timesteps, the RMSE of UwEnKF-SRGPF and EnKF are much larger than that of 1-timestep observation. Since the model error distribution is exponential, the error mean is greater than zero. When observations are not available, errors may accumulate and cannot be corrected, and the performance deteriorates.

Fig. 7 and Fig. 8 show the variation of RMSE for different ensemble sizes under 1-timestep and 5-timesteps observations respectively. When there is observation at every timestep, the differences of the results of the ten different runs are very small, so the curves of the 95% confidence interval are hardly visible in Fig. 7. For 5-timesteps observations as shown in Fig. 8, the 95% confidence interval for RMSE of EnKF is much wider than that of UwEnKF-SRGPF. With the increasing of ensemble members, the interval gets narrower. The main reason is that the bias for EnKF accumulates between two observation moments and more ensemble members are needed to get stable estimations. While UwEnKF-SRGPF performs much better as it is more stable and more accurate than EnKF. We can also see when there is observation at every timestep, the steepest decline in RMSE occurs between ensemble of size 20 and 50, after that the curve becomes flat. The optimal choice for ensemble size is 100, considering the performance and computational cost. When the observation is at every 5 timesteps, for performance of EnKF fluctuates much more and is less stable compared to UwEnKF-SRGPF. The performance of UwEnKF-SRGPF is not only better than EnKF, it continues to improve until reaching the ensemble size of 200, whereas EnKF shows relatively little improvement after the ensemble size of 50.

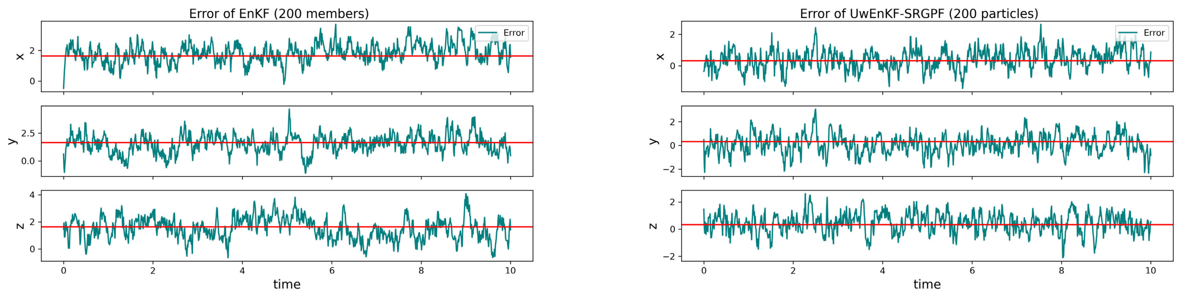


FIGURE 10. Errors of EnKF and UwEnKF-SRGPF with 200 particles by 1-timestep under exponential noise. The error is between the analysis and the truth, the red line is the mean of error.

For comparison under different observation frequencies, we give the analysis trajectories with 200 particles in the following.

From Fig. 9, we can see that the accuracy of the results are less than the case with Guassian noise (Fig. 3). The analysis trajectories now deviate from the truth except for UwEnKF-SRGPF with observations at every timestep. The analysis trajectories of EnKF lie above the truth since the noise mean is greater than zero, especially for 5-timesteps observations. Fig. 10 gives more details of the difference between the analysis and the truth. For EnKF, the mean of error is large than zero which is caused by the nonzero bias in the exponential noise. While for UwEnKF-SRGPF, the mean of error is close to zero, because UwEnKF-SRGPF calculates the weights of particles and it can correct the nonzero bias in the noise distribution. When observation is at every 5 timesteps, the analysis curves of both methods have little spikes departing from the truth. The reason is that the noise distribution has non-zero bias and errors accumulated between two adjacent observations. The error of UwEnKF-SRGPF is smaller than EnKF showing it is more robust for non-Gaussian distributed noises.

VI. CONCLUSION

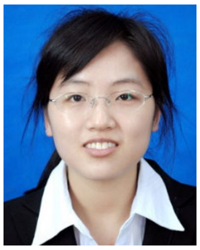
In this paper, we proposed the UwEnKF-SRGPF approach which combines sample regenerating particle filter (SRGPF) and unequal weight ensemble Kalman filter (UwEnKF). There are two known ways for dealing with the degeneracy of the particle filter, one is finding a good proposal density and the other is resampling. Our approach tries to solve the degeneracy problem by using the UwEnKF as proposal density function for the particle filter and by regenerating particles instead of resampling to preserve the diversity. In this way, the estimation at time t_k reduces the dependence on the previous time step. The traditional approach uses particles at time t_{k-1} as input for model at time t_k, while our method tries to generate particles according to a certain assumption of the posterior distribution of states and the diversity of particles is kept.

We used the Lorenz 63 model to evaluate the performance of our approach and compare with EnKF and WEnKF under Gaussian noise. UwEnKF-SRGPF performs the best as the ensemble size increases, WEnKF is in the second place. Although not as good as the other two methods, EnKF also gives a comparable performance. When the noise is exponential distributed, we compared UwEnKF-SRGPF with EnKF, as WEnKF cannot be directly implemented. It shows that UwEnKF-SRGPF performs much better than EnKF and can filter out the noise when observation is available, whereas EnKF has a bias since the noise distribution has a long tail with a nonzero mean. UwEnKF-SRGPF has demonstrated to be more robust in dealing with noises with nonzero biases. One of the problems requiring further investigation is to improve the accuracy of analysis when observation is sparse, as we have seen that when the observation frequency further decreases to one observation at 10 or 20 timesteps, the performance of all the methods degrades dramatically.

REFERENCES

[1] P. J. van Leeuwen, "Particle filtering in geophysical systems," *Monthly Weather Rev.*, vol. 137, no. 12, pp. 4089–4114, Dec. 2009.
[2] P. J. van Leeuwen, "Nonlinear data assimilation in geosciences: An extremely efficient particle filter," *Quart. J. Roy. Meteorolog. Soc.*, vol. 136, no. 653, pp. 1991–1999, Oct. 2010.
[3] P. J. van Leeuwen, "Efficient nonlinear data-assimilation in geophysical fluid dynamics," *Comput. Fluids*, vol. 46, no. 1, pp. 52–58, Jul. 2011.
[4] R. Karlsson and N. Bergman, "Auxiliary particle filters for tracking a maneuvering target," in *Proc. 39th IEEE Conf. Decis. Control*, Dec. 2000, pp. 3891–3895.
[5] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002.
[6] I. Hoteit, D.-T. Pham, G. Triantafyllou, and G. Korres, "A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography," *Monthly Weather Rev.*, vol. 136, no. 1, pp. 317–334, Jan. 2008.
[7] J. Poterjoy, R. A. Sobash, and J. L. Anderson, "Convective-scale data assimilation for the weather research and forecasting model using the local particle filter," *Monthly Weather Rev.*, vol. 145, no. 5, pp. 1897–1918, Apr. 2017.
[8] M. S. Grewal and A. P. Andrews, "Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]," *IEEE Control Syst.*, vol. 30, no. 3, pp. 69–78, Jun. 2010.
[9] D. H. Won, S. Chun, S. Sung, Y. J. Lee, J. Cho, J. Joo, and J. Park, "INS/vSLAM system using distributed particle filter," *Int. J. Control, Autom. Syst.*, vol. 8, no. 6, pp. 1232–1240, Dec. 2010.
[10] H. H. Holm, M. L. Sætra, and P. J. van Leeuwen, "Massively parallel implicit equal-weights particle filter for ocean drift trajectory forecasting," *J. Comput. Phys., X*, vol. 6, Mar. 2020, Art. no. 100053.
[11] J. Jeong, Y. Cho, and A. Kim, "HDMI-loc: Exploiting high definition map image for precise localization via bitwise particle filter," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6310–6317, Oct. 2020.

[12] C. N. Amarnath, M. I. Momtaz, and A. Chatterjee, "Encoded check driven concurrent error detection in particle filters for nonlinear state estimation," in *Proc. IEEE 26th Int. Symp. Line Test. Robust Syst. Design (IOLTS)*, Jul. 2020, pp. 1–6.
[13] B. Cluzet, M. Lafaysse, E. Cosme, C. Albergel, L.-F. Meunier, and M. Dumont, "CrocO_v1.0: A particle filter to assimilate snowpack observations in a spatialised framework," *Geosci. Model Dev.*, vol. 14, pp. 1595–1614, Jul. 2020.
[14] J. Zhang, Z. Liu, and Y. Lin, "Correlation Gaussian particle filter for robust visual tracking," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Aug. 2020, pp. 4854–4857.
[15] W. Ma and F. Xu, "Study on computer vision target tracking algorithm based on sparse representation," *J. Real-Time Image Process.*, vol. 18, no. 2, pp. 407–418, Jul. 2020.
[16] S. K. Veeramalla and V. K. H. R. Talari, "Multiple dipole source localization of EEG measurements using particle filter with partial stratified resampling," *Biomed. Eng. Lett.*, vol. 10, no. 2, pp. 205–215, Feb. 2020.
[17] M. Speekenbrink, "A tutorial on particle filters," *J. Math. Psychol.*, vol. 73, pp. 140–152, Aug. 2016.
[18] C. Snyder, T. Bengtsson, and M. Morzfeld, "Performance bounds for particle filters using the optimal proposal," *Monthly Weather Rev.*, vol. 143, no. 11, pp. 4750–4761, Oct. 2015.
[19] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high-dimensional particle filtering," *Monthly Weather Rev.*, vol. 136, no. 12, pp. 4629–4640, Dec. 2008.
[20] M. Zhu, P. J. V. Leeuwen, and J. Amezcua, "Implicit equal-weights particle filter," *Quart. J. Roy. Meteorolog. Soc.*, vol. 142, no. 698, pp. 1904–1919, May 2016.
[21] Y. Chen, W. Zhang, and M. Zhu, "A localized weighted ensemble Kalman filter for high-dimensional systems," *Quart. J. Roy. Meteorolog. Soc.*, vol. 146, no. 726, pp. 438–453, Dec. 2019.
[22] P. J. Leeuwen, H. R. Künsch, L. Nerger, R. Potthast, and S. Reich, "Particle filters for high-dimensional geoscience applications: A review," *Quart. J. Roy. Meteorolog. Soc.*, vol. 145, no. 723, pp. 2335–2365, May 2019.
[23] C. Snyder, "Particle filters, the 'optimal' proposal and high-dimensional systems," in *Proc. ECMWF Seminar Data Assimilation Atmos. Ocean*, Sep. 2011, pp. 1–10.
[24] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F Radar Signal Process.*, vol. 140, no. 2, p. 107, 1993.
[25] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *J. Comput. Graph. Statist.*, vol. 5, no. 1, pp. 1–25, Mar. 1996.
[26] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *J. Amer. Stat. Assoc.*, vol. 93, no. 443, pp. 1032–1044, Aug. 1998.
[27] L. Tiancheng, B. Miodrag, and D. M. Petar, "Resampling methods for particle filtering: Classification, implementation, and strategies," *Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015.
[28] M. Bolić, P. M. Djurić, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 15, pp. 1–11, Dec. 2004.
[29] A. S. Stordal, H. A. Karlsen, G. Nævdal, H. J. Skaug, and B. Vallès, "Bridging the ensemble Kalman filter and particle filters: The adaptive Gaussian mixture filter," *Comput. Geosci.*, vol. 15, no. 2, pp. 293–305, Sep. 2010.
[30] N. Papadakis, E. Mémin, A. Cuzol, and N. Gengembre, "Data assimilation with the weighted ensemble Kalman filter," *Tellus A, Dyn. Meteorol. Oceanogr.*, vol. 62, no. 5, pp. 673–697, 2010.
[31] P. J. van Leeuwen, Y. Cheng, and S. Reich, *Nonlinear Data Assimilation* (Frontiers in Applied Dynamical Systems: Reviews and Tutorials), vol. 2. Cham, Switzerland: Springer, 2015, p. xii and 118, doi: 10.1007/978-3-319-18347-3.
[32] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *J. Geophys. Res.*, vol. 99, no. C5, 1994, Art. no. 10143.
[33] G. Burgers, P. Jan van Leeuwen, and G. Evensen, "Analysis scheme in the ensemble Kalman filter," *Monthly Weather Rev.*, vol. 126, no. 6, pp. 1719–1724, Jun. 1998.
[34] P. L. Houtekamer and L. Herschel Mitchell, "Ensemble Kalman filtering," *Quart. J. Roy. Meteorolog. Soc.*, vol. 131, no. 613, pp. 3269–3289, Oct. 2005.
[35] P. van Leeuwen, "Particle filters for nonlinear data assimilation in high-dimensional systems," *Annales de la Faculté des Sci. de Toulouse Mathématiques*, vol. 26, no. 4, pp. 1051–1085, 2017.
[36] S. Beyou, A. Cuzol, S. S. Gorthi, and E. Mémin, "Weighted ensemble transform Kalman filter for image assimilation," *Tellus A, Dyn. Meteorol. Oceanogr.*, vol. 65, no. 1 Dec. 2013, Art. no. 18803,.
[37] A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice* (Statistics for Engineering and Information Science). New York, NY, USA: Springer-Verlag, 2001, p. xxviii and 581, doi: 10.1007/978-1-4757-3437-9.
[38] J. Skauvold, J. Eidsvik, P. J. V. Leeuwen, and J. Amezcua, "A revised implicit equal-weights particle filter," *Quart. J. Roy. Meteorolog. Soc.*, vol. 145, no. 721, pp. 1490–1502, Mar. 2019.
[39] M. Ades and P. J. van Leeuwen, "An exploration of the equivalent weights particle filter," *Quart. J. Roy. Meteorolog. Soc.*, vol. 139, no. 672, pp. 820–840, Aug. 2012.



XIAO LI received the B.E. degree in statistics and the M.S. degree in computational mathematics from Shandong University, China, in 2011 and 2014, respectively, where she is currently pursuing the Ph.D. degree. From 2014 to 2017, she was a Technical Staff with the Public Meteorological Service Center (PMSC), China Meteorological Administration (CMA), mainly worked on weather forecast correction and extreme weather warning information system. From 2019 to 2021, she is visiting TU Delft as an exchange student.

AI JIE CHENG is currently a Professor at the School of Mathematics, Shandong University. His research interests include numerical solutions of partial differential equations, scientific and engineering computing, and reservoir simulation.

HAI XIANG LIN is currently an Associate Professor with the Mathematical Physics Group, Delft Institute of Applied Mathematics (DIAM), TU Delft. He is also appointed as a Professor of data analytics for environmental modeling at the Institute of Environmental Sciences, Leiden University, on behalf of the R. Timman Foundation. In addition, he is teaching master's and Ph.D. courses in high performance computing, parallel computing, scientific programming, and computational aspects of stochastic differential equations. He has published more than 150 peer-reviewed journals and international conference papers. His research interests include parallel numerical algorithms (sparse matrix computation), parallel data assimilation algorithms, simulation, and control of power systems, modeling and simulation of environmental pollution, machine learning, and big data.