

A defensive marginal particle filtering method for data assimilation

Linjie Wen · Jiangqi Wu · Linjun Lu ·
Jinglai Li

Received: date / Accepted: date

Abstract The Particle filtering (PF) method is often used to estimate the states of dynamical systems in a Bayesian framework. A major limitation of the standard PF method is that the dimensionality of the state space increases as the time proceeds and eventually may cause degeneracy of the algorithm. A possible approach to alleviate the degeneracy issue is to compute the marginal posterior distribution at each time step, which leads to the so-called marginal PF method. In this work we propose a defensive marginal PF algorithm which constructs a sampling distribution in the marginal space by combining the standard PF and the EnKF methods. With numerical examples we demonstrate that the proposed method has competitive performance against many existing algorithms.

Keywords Data assimilation · defensive importance sampling · ensemble Kalman filter · marginal particle filtering · particle filtering.

Mathematics Subject Classification (2000) 62F15 · 65C05

1 Introduction

Assimilation of data into mathematical models is an essential task in many fields of science and engineering, ranging from meteorology [11] to robotics [23].

This work was supported by the NSFC under grant number 11301337 and 5150080805.

L. Wen · J. Wu

School of Mathematical Sciences and Institute of Natural Sciences, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai 200240, China.

L. Lu

School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: linjunlu@sjtu.edu.cn

J. Li

Department of Mathematical Sciences, University of Liverpool, Liverpool, UK. E-mail: Jinglai.Li@liverpool.ac.uk

Simply speaking, data assimilation is to estimate the optimal prediction based on both the output of the mathematical model, which is only an approximation of the real-world system, and the observations that are subject to measurement noise. Many commonly used data assimilation methods, most notably, the Kalman filter [5, 19], are based on linear control theory and optimization, and their applications to nonlinear systems are usually challenging, which often require some linearization or approximation processes, e.g. the extended Kalman filter [14] or the ensemble Kalman filter [10]; sometimes they can even fail [16, 21] when strong nonlinearity is present.

On the other hand, the sequential Monte Carlo (SMC) method (see e.g., [2]), also known as particle filtering (PF), can deal with problems with strongly nonlinear models, without any linearization or approximation. The basic idea of PF is the following. Suppose that the mathematical model is a nonlinear stochastic dynamical system, and our goal is to estimate the hidden states $\{\mathbf{x}_t\}_{t=0}^T$ of the system from noisy partial observations $\{\mathbf{y}_t\}_{t=0}^T$ of the system. This can be done with the so-called Bayes filter (also known as the optimal filter), where the posterior probability density function (PDF) of the hidden states is estimated by the Bayes' rule recursively [8]. As the posterior distribution usually does not admit an analytical form, the PF method approximates the posterior distribution with Monte Carlo sampling (hence its name SMC). That is, the PF method employs a number of independent random realizations called particles, sampled directly from the state space, to represent the posterior probability: namely, at each time t , the method first generates particles and then updates the weight of each particle according to the observations \mathbf{y}_t . For further discussions on the PF method and its applications, we refer to [2, 3, 9, 6] and the references therein.

The PF method in its very basic form can be understood as to draw weighted samples according to the joint distribution $\pi(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$ using the importance sampling (IS) technique. When t is large, the method thus performs IS simulations in a high-dimensional state space, which may result in degeneracy of the particles (the IS weights becoming zero for all but one particle) [9]. On the other hand, often in practice one is only interested only in the marginal distribution $\pi(\mathbf{x}_t|\mathbf{y}_{0:t})$, which implies that it is unnecessary to sample the high-dimensional joint distribution $\pi(\mathbf{x}_{0:t}|\mathbf{y}_{0:t})$. Instead, one can perform IS only in the marginal space of $\pi(\mathbf{x}_t|\mathbf{y}_{0:t})$, and based on this idea, a method called marginal particle filter (MPF) was proposed in [15] to alleviate the degeneracy issue. The method later has found applications in the estimation of filter derivative [20] and robot localization [18]. A key step in this kind of methods is to construct an IS distribution that can well approximate the marginal posterior $\pi(\mathbf{x}_t|\mathbf{y}_{0:t})$ at each time step. To this end, the MPF method in [15] requires to do a kernel density estimation of the marginal posterior $\pi(\mathbf{u}_t|\mathbf{y}_{0:t})$ from the weighted samples at each time t , which can be a rather challenging task when the number of particles is limited. The main purpose of this work is to provide an alternative approach to construct the IS distribution in the marginal space. In particular, we consider the special case where the observation operator is linear and the observation noise is Gaussian, and we

propose a robust and efficiently method to compute the marginal IS distribution based on the ensemble Kalman filter (EnKF). A limitation of the EnKF based IS distribution is that (just like the EnKF method itself) it may result in poor performance when the posteriors are strongly non-Gaussian. To address the issue, we introduce a defensive scheme to construct an IS distribution by combining the EnKF and the standard PF methods, and with numerical examples, we demonstrate that the new method performs well even when the posteriors significantly deviate from a Gaussian distribution.

The rest of the paper is arranged as follows. In Section 2, we first introduce the basic setup of the filtering problem of dynamical models and then discuss the standard PF and EnKF methods for solving this type of problems. In Section 3, we present in detail our defensive MPF method. Numerical examples are provided in Section 4 to compare the performance of the proposed method and the existing ones, and finally Section 5 offers some concluding remarks.

2 The PF and the EnKF methods

We give a brief overview of the formulation of the PF and the EnKF methods in this section.

2.1 State-Space Models

We consider the filtering problem in a dynamic state-space form:

$$\mathbf{u}_t = f_t(\mathbf{u}_{t-1}) + \boldsymbol{\epsilon}_t, \quad \mathbf{u}_0 \sim \pi(\mathbf{u}_0), \quad (2.1a)$$

$$\mathbf{y}_t = H_t \mathbf{u}_t + \boldsymbol{\eta}_t, \quad (2.1b)$$

where $\mathbf{u}_t \in \mathbb{R}^{n_u}$ denotes the state vector at time t , and $\mathbf{y}_t \in \mathbb{R}^{n_v}$ represents the observed data at time t . In addition, $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\eta}_t$ are the propagation and the observation noise respectively, and H_t is the observation operator. In a filtering problem, the observation \mathbf{y}_t arrives sequentially in time and the goal is to estimate the true state \mathbf{u}_t , based on the prediction by (2.1a) and the measurement (2.1b). Finally we emphasize here that the dynamic model (2.1a) is Markovian, in that any future \mathbf{u}_{t+1} is independent of the past given the present \mathbf{u}_t :

$$\pi(\mathbf{u}_{t+1} | \mathbf{u}_{0:t}, \mathbf{y}_{0:t}) = \pi(\mathbf{u}_{t+1} | \mathbf{u}_t) \quad \text{and} \quad \pi(\mathbf{y}_{t+1} | \mathbf{u}_{0:t+1}, \mathbf{y}_{0:t}) = \pi(\mathbf{y}_{t+1} | \mathbf{u}_{t+1}), \quad (2.2)$$

which will be used in the derivation of the SMC method.

2.2 The particle filter

In general, we can formulate the filtering problem in a Bayesian inference framework: i.e., we try to infer state parameters $\mathbf{u}_{0:T}$ from data $\mathbf{y}_{0:T}$ for some

positive integer T , and ideally we can compute the posterior distribution using the Bayes' formula:

$$\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T}) = \frac{\pi(\mathbf{y}_{0:T}|\mathbf{u}_{0:T})\pi(\mathbf{u}_{0:T})}{\pi(\mathbf{y}_{0:T})}.$$

As is mentioned earlier, the posterior distribution $\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})$ usually does not admit an analytical form, and the sequential Monte Carlo method can be used to address the issue. Simply speaking, SMC allows to generate (weighted) samples, called particles, from the posterior distribution $\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})$, which can be used to evaluate any quantities of interest associated with the posterior $\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})$.

We now give a brief overview of the SMC method, and it is easier to start with a standard MC estimation. Suppose that there is a real-valued function $h(\cdot) : \mathbf{R}^{t \times n_u} \rightarrow R$ and we are interested in the expectation

$$I = E_{u_{0:T}|\mathbf{y}_{0:T}}[h(\mathbf{u}_{0:T})] = \int h(\mathbf{u}_{0:T})\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})d\mathbf{u}_{0:T}$$

which can be estimated with a MC estimator:

$$\hat{I} = \frac{1}{M} \sum_{m=1}^M h(\mathbf{u}_{0:T}^m),$$

where $\{\mathbf{u}_{0:T}^m\}_{m=1}^M$ are samples drawn from $\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})$. It should be clear that the MC estimator \hat{I} is an unbiased estimator of I . In many practical problems, drawing samples from the target distribution $\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})$ can be a challenging task, and in this case, we can use the technique of importance sampling (IS). The IS method introduces an importance distribution $q_t(\mathbf{u}_{0:T})$ and rewrites

$$I = \int h(\mathbf{u}_{0:T})\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})d\mathbf{u}_n = \int h(\mathbf{u}_{0:T})w(\mathbf{u}_{0:T})q(\mathbf{u}_t|\mathbf{y}_{0:T})d\mathbf{u}_t$$

with $w_t(\mathbf{u}_{0:T}) = \pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})/q_t(\mathbf{u}_{0:T})$ is the IS weight. It yields directly an IS estimator of I :

$$\hat{I}_{IS} = \frac{1}{M} \sum_{m=1}^M h(\mathbf{u}_{0:T}^m)w(\mathbf{u}_{0:T}^m),$$

where the samples $\{\mathbf{u}_{0:T}^m\}_{m=1}^M$ are drawn from the importance distribution $q_t(\mathbf{u}_{0:T})$, and it can also be verified that the IS estimator is also an unbiased one for I . The IS requires to generate samples from $q(\mathbf{u}_{0:T})$ and to draw the joint sample $\mathbf{u}_{0:T}$ from a joint distribution $q(\mathbf{u}_{0:T})$. Using the Markovian property in Eq. (2.2), we can write the posterior distribution $\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})$ in the form of

$$\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T}) = \frac{1}{Z_T} \pi(\mathbf{y}_0|\mathbf{u}_0)\pi(\mathbf{u}_0) \prod_{t=1}^T \pi(\mathbf{y}_t|\mathbf{u}_t)\pi(\mathbf{u}_t|\mathbf{u}_{t-1}), \quad (2.3)$$

where Z_T is the normalization constant. Similarly, we can also assume that the importance distribution $q(\mathbf{u}_{0:T})$ is also given in such a sequential form:

$$q(\mathbf{u}_{0:T}) = q_0(\mathbf{u}_0) \prod_{t=1}^T q_t(\mathbf{u}_t | \mathbf{u}_{t-1}),$$

and the resulting IS weight function is

$$w_0(\mathbf{u}_0) = \frac{\pi(\mathbf{y}_0 | \mathbf{u}_0) \pi(\mathbf{u}_0)}{q_0(\mathbf{u}_0)}, \quad (2.4)$$

and

$$w_T(\mathbf{u}_{0:T}) = \frac{1}{Z} w_0(\mathbf{u}_0) \prod_{t=1}^T \alpha_t(\mathbf{u}_{0:t}), \quad (2.5)$$

for $t > 0$, where α_t is the incremental weight function:

$$\alpha_t(\mathbf{u}_{0:t}) = \frac{\pi(\mathbf{y}_t | \mathbf{u}_t) \pi(\mathbf{u}_t | \mathbf{u}_{t-1})}{q_t(\mathbf{u}_t | \mathbf{u}_{t-1})}. \quad (2.6)$$

We note that, in the formulation above, we do not have the knowledge of the normalization constant Z . In the implementation, however, we can simply set the normalization constant $Z = 1$, and renormalize the weights computed. Namely, suppose that we draw a set of samples $\{\mathbf{u}_{0:T}^m\}_{m=1}^M$ from the IS distribution $q(\mathbf{u}_{0:T})$, and we compute the weights $\{w_T^m\}_{m=1}^M$ of the samples using Eqs. (2.4)-(2.6) (and taking $Z = 1$), and then renormalize the weights as,

$$w_T^m = \frac{w_T^m}{\sum_{m'=1}^M w_T^{m'}}. \quad (2.7)$$

The SMC algorithm performs the procedure described above in a recursive manner:

- At $t = 0$, sample $\{\mathbf{u}_0^m\}_{m=0}^M \sim q_0(\mathbf{u}_0)$, and compute $\{w_0^m = w_0(\mathbf{u}_0^m)\}_{m=1}^M$ using Eq (2.4); renormalize the weights: $w_0^m = \frac{1}{\sum_{m'=1}^M w_0^{m'}} w_0^m$.
- At $t > 1$:
 - prediction step: for each $m = 1 \dots M$, draw $\mathbf{u}_t^m \sim q_t(\mathbf{u}_t | \mathbf{u}_{t-1}^m)$;
 - updating step: for each $m = 1 \dots M$, compute the incremental function α_t^m from Eq. (2.6); update the weights $w_t^m = \alpha_t^m w_{t-1}^m$, and renormalize them as $w_t^m = \frac{1}{\sum_{m'=1}^M w_t^{m'}} w_t^m$ for each $m = 1 \dots M$.

In the standard SMC method, one simply takes $q_0(\mathbf{u}_0) = \pi(\mathbf{u}_0)$ and

$$q_t(\mathbf{u}_t | \mathbf{u}_{t-1}) = \pi(\mathbf{u}_t | \mathbf{u}_{t-1}),$$

and as a result, $w_0 = \pi(\mathbf{y}_0 | \mathbf{u}_0)$, and the incremental weight function becomes

$$\alpha_t(\mathbf{u}_{0:t}) = \pi(\mathbf{y}_t | \mathbf{u}_t). \quad (2.8)$$

In the SMC algorithm, the variance of the importance weight $w_t(\mathbf{u}_{0:t})$ will increase over time, and thus at the time t increases, the IS weights will become negligibly small for all but one sample, an issue known as particle degeneracy. To address the issue, a resampling step is often performed to obtain a set of equally weighted particles: namely one first draws U_1 from uniform distribution $U[0, 1/M]$ and then defines $U_m = U_1 + (m - 1)/M$ for $m = 1 \dots M$.

2.3 The ensemble Kalman filter

As is mentioned in Section 1, we consider in this work a special case where the observation operator H_t is linear and the observation noise $\boldsymbol{\eta}_t$ is Gaussian. In this case, the marginal posterior distributions $\pi(\mathbf{u}_t | \mathbf{y}_{0:t})$ can be approximated by the EnKF method. The basic idea of the EnKF method is the following. Suppose that at time t , the observation noise is $\boldsymbol{\eta}_t \sim N(0, R_t)$ and the prior $\pi(\mathbf{u}_t | \mathbf{y}_{0:t-1})$ can be approximated by a Gaussian distribution with mean $\tilde{\boldsymbol{\mu}}_t$ and covariance $\tilde{\boldsymbol{\Sigma}}_t$. It follows that the posterior distribution $\pi(\mathbf{u}_t | \mathbf{y}_{0:t})$ is also Gaussian and its mean and covariance can be obtained analytically:

$$\boldsymbol{\mu}_t = \tilde{\boldsymbol{\mu}}_t + K_t(\mathbf{y}_t - H_t \tilde{\boldsymbol{\mu}}_t), \quad \boldsymbol{\Sigma}_t = (I - K_t H_t) \tilde{\boldsymbol{\Sigma}}_t, \quad (2.9)$$

where I is the identity matrix and

$$K_t = \tilde{\boldsymbol{\Sigma}}_t H_t^T (H_t \tilde{\boldsymbol{\Sigma}}_t H_t^T + R_t)^{-1} \quad (2.10)$$

is the so-called Kalman gain matrix. Moreover, when the prior $\pi(\mathbf{u}_t | \mathbf{y}_{0:t-1})$ is exactly Gaussian, this formulation becomes the standard Kalman filter.

In the EnKF method, one avoids computing the mean and the covariance directly in each step. Instead, both the prior and the posterior distributions are represented with a set of samples, known as an ensemble. Specifically, let $\{\tilde{\mathbf{u}}_t^m\}_{m=1}^M$ be a set of samples drawn from the prior distribution $\pi(\mathbf{u}_t | \mathbf{y}_{0:t-1})$, and we shall compute a Gaussian approximation of $\pi(\mathbf{u}_t | \mathbf{y}_{0:t-1})$ from the samples. Namely we estimate the mean and the covariance of $\pi(\mathbf{u}_t | \mathbf{y}_{0:t-1})$ from the samples:

$$\tilde{\boldsymbol{\mu}}_t = \frac{1}{M} \sum_{m=1}^M \tilde{\mathbf{u}}_t^m, \quad \tilde{\boldsymbol{\Sigma}}_t = \frac{1}{M-1} \sum_{m=1}^M (\tilde{\mathbf{u}}_t^m - \tilde{\boldsymbol{\mu}}_t)(\tilde{\mathbf{u}}_t^m - \tilde{\boldsymbol{\mu}}_t)^T, \quad (2.11)$$

and as is mentioned earlier, the prior distribution $\pi(\mathbf{u}_t | \mathbf{y}_{0:t-1})$ can be approximated by $N(\tilde{\boldsymbol{\mu}}_t, \tilde{\boldsymbol{\Sigma}}_t)$. It follows immediately that the posterior distribution is also Gaussian with mean $\boldsymbol{\mu}_t$ and covariance $\boldsymbol{\Sigma}_t$ given by Eq. (2.9). Moreover it can be verified that the samples

$$\mathbf{u}_t^m = \tilde{\mathbf{u}}_t^m + K_t(\mathbf{y}_t - (H_t \tilde{\mathbf{u}}_t^m + \boldsymbol{\eta}_t^m)), \quad \boldsymbol{\eta}_t^m \sim N(0, R_t), \quad (2.12)$$

follow the distribution $N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, provided that $\tilde{\mathbf{u}}_t^m \sim N(\tilde{\boldsymbol{\mu}}_t, \tilde{\boldsymbol{\Sigma}}_t)$ for all $m = 1 \dots M$. That is, $\{\mathbf{u}_{t-1}^m\}_{m=1}^M$ is the (posterior) ensemble at step t . Given the ensemble $\{\mathbf{u}_{t-1}^m\}_{m=1}^M$ at time $t-1$, the EnKF algorithm performs the following two steps at time t :

- prediction step: for each $m = 1 \dots M$, draw $\tilde{\mathbf{u}}_t^m = f_t(\tilde{\mathbf{u}}_t | \mathbf{u}_{t-1}^m) + \epsilon_t^m$;
- updating step: for each $m = 1 \dots M$, compute $\mathbf{u}_t^m = \tilde{\mathbf{u}}_t^m + K_t(\mathbf{y}_t - H_t \tilde{\mathbf{u}}_t^m)$.

Finally we should note that, as the dynamical model is generally nonlinear, the EnKF method can only provide an approximation of the true posterior distribution, no matter how large the sample size is, which is certainly a significant limitation of the method.

3 The defensive marginal PF algorithm

As is discussed in Section 2.2, the standard SMC method aims to perform IS for the joint posterior distribution $\pi(\mathbf{u}_{0:t} | \mathbf{y}_{0:t})$, where the dimensionality of the state space grows as t increases. On the other hand, in many practical filtering problems, one is often only interested in the marginal posterior distribution at each step, $\pi(\mathbf{u}_t | \mathbf{y}_{0:t})$, rather than the whole joint distribution. This then yields a simple idea: if we perform IS in the marginal space, the dimensionality of the problem is thus fixed and much smaller than that of the joint parameter space. For any time t , suppose that there is a function defined on the marginal space: $h_t : R^{n_u} \rightarrow R$, and we are interested in the posterior expectation of $h_t(\mathbf{u}_t)$:

$$I = \int h_t(\mathbf{u}_t) \pi(\mathbf{u}_t | \mathbf{y}_{0:t}) d\mathbf{u}_t$$

We shall construct an IS distribution $q_t(\mathbf{u}_t | \mathbf{y}_{0:t})$, and estimate I as

$$\hat{I}_{IS} = \frac{1}{M} \sum_{m=1}^M h_t(\mathbf{u}_t^m) w_t(\mathbf{u}_t^m), \quad (3.1)$$

where \mathbf{u}_t^m are drawn from $q_t(\mathbf{u}_t)$ and $w_t(\mathbf{u}_t^m) = \pi(\mathbf{u}_t | \mathbf{y}_{0:t}) / q_t(\mathbf{u}_t)$. The key issue here is certainly to find a good IS distribution $q_t(\mathbf{u}_t)$, and ideally this IS distribution should approximate the marginal posterior $\pi(\mathbf{u}_t | \mathbf{y}_{0:t})$. In [15], a kernel-based IS distribution is suggested:

$$q_t(\mathbf{u}_t) = \sum_{m=1}^M w_{t-1}(\mathbf{u}_{t-1}^m) Q_m(\mathbf{u}_t | \mathbf{u}_{t-1}^m),$$

where each Q_m is obtained using a weighted Kernel density estimation (KDE) method. As a result the method requires to perform a weighed KDE procedure at each time step, which can be computationally intensive even with some fast KDE algorithms (e.g. the dual-tree methods). We also note that another special choice of the IS distribution is

$$q_t(\mathbf{u}_t) = \pi(\mathbf{u}_t | \mathbf{y}_{0:t-1}), \quad (3.2)$$

and it should be clear that the associated weight becomes $w_t(\mathbf{u}_t) = \pi(\mathbf{y}_t | \mathbf{u}_t)$ and the algorithm is essentially equivalent to the standard PF method.

In our setting, the EnKF method can naturally yield such an approximate marginal posterior distribution in a very efficient and effective manner. Loosely speaking, at a given time, we first compute an ensemble of the marginal posterior distribution using the EnKF scheme, estimate the associated Gaussian approximation from the ensemble, and use it as the IS distribution in the marginal SMC. Specifically, let $\{\mathbf{u}_t^m\}_{m=1}^M$ be the posterior ensemble at time t obtained with the EnKF formulation, we use the following procedure to compute the IS distribution:

Algorithm 1: Estimating the IS distribution from the ensemble

1. estimate the mean and covariance from the posterior ensemble $\{u_t^m\}_{m=1}^M$:

$$\mu_{\text{En}} = \frac{1}{M} \sum_{m=1}^M u_t^m, \quad \Sigma_{\text{En}} = \frac{1}{M-1} \sum_{m=1}^M (u_t^m - \mu_{\text{En}})(u_t^m - \mu_{\text{En}})^T; \quad (3.3)$$

let $q'_{\text{EnKF}}(u) = N(\mu_{\text{En}}, \Sigma_{\text{En}})$;

2. draw M samples u_t^1, \dots, u_t^M from q'_{EnKF} , and compute the weight of each sample:

$$w^m = \frac{\pi(u_t | y_{0:t})}{q(u_t)}$$

3. estimate the mean and covariance of the weighted ensemble $\{(u_t^m, w^m)\}$:

$$\boldsymbol{\mu}_{\text{updated}} = \sum_{m=1}^M w^m u_t^m, \quad \Sigma_{\text{updated}} = \sum_{m=1}^M w^m (u_t^m - \boldsymbol{\mu}_{\text{updated}})(u_t^m - \boldsymbol{\mu}_{\text{updated}})^T; \quad (3.4)$$

let $q_{\text{EnKF}} = N(\boldsymbol{\mu}_{\text{updated}}, \Sigma_{\text{updated}})$.

It is worth noting that, the EnKF ensemble does not exactly follow the posterior distribution, and so in the procedure above, instead of using q_{EnKF} , i.e., the Gaussian approximation estimated directly from the EnKF ensemble, we introduce an additional step, in which we first generate a weighted ensemble according to the true posterior, and then update the Gaussian approximation according to this weighted ensemble. By doing so we ensure that the Gaussian approximation is constructed with respect to the true posterior ensemble.

A well-known issue in the EnKF method is that, due to the nonlinearity of the model, the ensemble computed by the method may become increasingly inaccurate as the time increases; as a result the IS distribution obtained with the EnKF method may deviate significantly from the true marginal posterior, leading to poor performance or even failure of the IS estimator in Eq. (3.1). To address the issue, we use the idea of *defensive importance sampling* (DIS) [13]. The basic idea behind DIS is quite straightforward: namely, to prevent the failure of the IS distribution computed with the EnKF method, one uses a mixture of the Gaussian approximation computed by EnKF and a safe or defensive distribution, which in our case is the standard PF distribution. Namely,

our defensive IS distribution is:

$$q_t(\mathbf{u}_t) = aq_{\text{EnKF}}(\mathbf{u}_t) + (1 - a)q_{\text{PF}}(\mathbf{u}_t), \quad (3.5)$$

where q_{EnKF} is the Gaussian distribution computed with the EnKF procedure described above, q_{PF} is the distribution given by Eq. (3.2), which, as discussed earlier, is equivalent to the standard PF, and $a \in [0, 1]$ is the weight of the EnKF component. An important issue here is how to compute the IS weight of each sample. It is easy to see that the weight function is

$$w_t(\mathbf{u}_t) = \frac{\pi(\mathbf{u}_t|\mathbf{y}_{0:t})}{aq_{\text{EnKF}} + (1 - a)q_{\text{PF}}} = \frac{1}{\frac{a}{w_{\text{EnKF}}} + \frac{(1-a)}{w_{\text{PF}}}}, \quad (3.6)$$

where

$$w_{\text{EnKF}} = \frac{\pi(\mathbf{y}_t|\mathbf{u}_t)\pi(\mathbf{u}_t|\mathbf{y}_{0:t-1})}{\pi(\mathbf{y}_t)q_{\text{EnKF}}(\mathbf{u}_t)}, \quad w_{\text{PF}} = \pi(\mathbf{y}_t|\mathbf{u}_t). \quad (3.7)$$

Computing w_{PF} is rather straightforward, but computing w_{EnKF} involves the evaluation of the integral:

$$\pi(\mathbf{u}_t|\mathbf{y}_{0:t-1}) = \int \pi(\mathbf{u}_t|\mathbf{u}_{t-1})\pi(\mathbf{u}_{t-1}|\mathbf{y}_{0:t-1})d\mathbf{u}_{0:t-1}. \quad (3.8)$$

In practice, this integral is approximated by

$$\pi(\mathbf{u}_t|\mathbf{y}_{0:t-1}) \approx \sum_{m=1}^M w_{t-1}^m \pi(\mathbf{u}_t|\mathbf{u}_{t-1}^m),$$

where $\{\mathbf{u}_{t-1}^m\}_{m=1}^M$ are the samples generated in the previous step and w_{t-1}^m is the associated weight of each sample \mathbf{u}_{t-1}^m (namely, the weighted ensemble $\{(\mathbf{u}_{t-1}^m, w_{t-1}^m)\}_{m=1}^M$ follows the distribution $\pi(\mathbf{u}_{t-1}|\mathbf{y}_{0:t-1})$). Finally we provide the complete defensive marginal PF (DMPF) algorithm in Algorithm 2.

Algorithm 2: The DMPF algorithm

- 1 At $t = 0$:
 - 2 Prediction: sample $\{\tilde{u}_0^m\}_{m=1}^M$ from $\pi_0(\cdot)$;
 - 3 Updating: $u_0^m = \tilde{u}_0^m + K_0(y_0 - H_0\tilde{u}_0^m)$ for $m = 1 \dots M$;
 - 4 Compute q_{EnKF} using Algorithm 1 and particles $\{u_0^m\}_{m=1}^M$;
 - 5 Draw M particles from q_0 from Eq. (3.5) for $t = 0$, and compute the weights using Eq. (3.7), yielding $\{(u_0^m, w^m)\}_{m=1}^M$;
 - 6 **for** $t=1 \dots T$ **do**
 - 7 Prediction: for each $m = 1 \dots M$, draw $\tilde{u}_t^m = f_t(\tilde{u}_t|u_{t-1}^m) + \epsilon_t^m$;
 - 8 Updating: $u_t^m = \tilde{u}_t^m + K_t(y_t - H_t\tilde{u}_t^m)$ for $m = 1 \dots M$;
 - 9 Compute q_{EnKF} using Algorithm 1 and particles $\{u_t^m\}_{m=1}^M$;
 - 10 Draw M particles from q_t given by Eq. (3.5), and compute the weights using Eq. (3.7), yielding $\{(u_t^m, w^m)\}_{m=1}^M$;
 - 11 **end**
-

4 Numerical examples

In this section we provide several numerical examples to demonstrate the performance of the proposed DMPPF method. In all these examples, we also implement the standard PF and the EnKF method for comparison purposes.

4.1 Lorenz 63 system

Our first example is the classical Lorenz 63 system [17], an often used benchmark problem for testing data assimilation algorithms. Specifically the system is described by,

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = x(\rho - z) - y, \\ \dot{z} = xy - \beta z. \end{cases} \quad (4.1)$$

For simplicity, we consider a discrete-time version of the system with additive noise:

$$\begin{cases} x_{t+1} = x_t + \sigma(y_t - x_t)\Delta t + \xi_t^x, \\ y_{t+1} = y_t + (x_t(\rho - z_t) - y_t)\Delta t + \xi_t^y, \\ z_{t+1} = z_t + (x_t y_t - \beta z_t)\Delta t + \xi_t^z, \end{cases} \quad (4.2)$$

where Δt is the discrete-time step size. Here we assume that the model noise ξ_t^x , ξ_t^y and ξ_t^z are all i.i.d zero-mean Gaussian with standard deviation σ_ξ . Moreover at each time t the observed data is taken to be $x_t^d = x_t + \eta_t^x$, $y_t^d = y_t + \eta_t^y$ and $z_t^d = z_t + \eta_t^z$, where the observation noise η_t^x , η_t^y and η_t^z are once again assumed to be i.i.d. zero-mean Gaussian with standard deviation σ_η .

In our numerical tests we take the parameters to be $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, $\Delta t = 0.01$, and the initial condition to be

$$[x_0, y_0, z_0] = [1.51, -1.53, 25.46].$$

The noise standard deviations are $\sigma_\epsilon = 0.5$ and $\sigma_\eta = 1$. We generated a true state and the associated data points from the model, which are shown in Fig. 1.

As is mentioned at the beginning of Section 4, we estimate the states from the simulated data with the proposed DMPPF method, as well as the standard PF and the EnKF methods. We note here that in all the three examples we use $a = 0.6$ in the DMPPF method. With each method, we generate 10^4 particles. We also perform a PF simulation with 10^5 particles and regard the results as the true posterior. We first compare the posterior mean computed with the different methods in Fig. 2, and note here that all these results for posterior mean agree very well with each other and so all the plots in Fig. 2 are indistinguishable. We then compare the posterior variances computed by all the methods and show the results in Fig. 3. One can see from the figure that, for the variances, the results of both the EnKF and the DMPPF methods agree reasonably with the true posterior variance, while those of the standard PF (with 10^4 particles) are subject to much stronger fluctuations than the

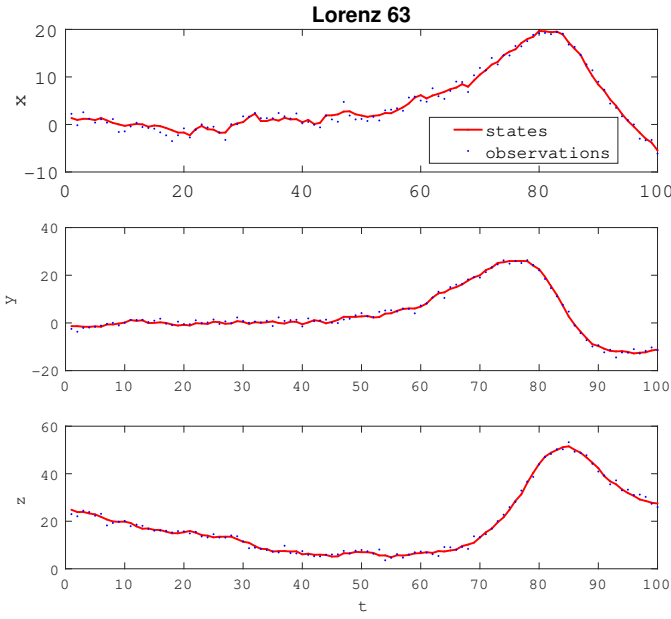


Fig. 1 The true state (dashed lines) and the simulated observations (dots) of the Lorenz 63 model.

other ones. These results suggest that for problems where the posterior distribution can be well approximated by the EnKF approximation, our DMPF method which utilises the EnKF approximation can substantially improve of the performances over the standard PF.

4.2 Bernoulli model

Our second example is the the Bernoulli equation,

$$\frac{dx}{dt} - x = -x^3, \quad x(t_0) = x_0, \quad (4.3)$$

which admits an analytical solution,

$$x(t) = M(x_0) = x_0 \times (x_0^2 + (1 - x_0^2)e^{-2\Delta t})^{-1/2}, \quad (4.4)$$

where $\Delta t = t - t_0$. Here for simplicity we use the analytical solution to construct the discrete-time model:

$$\begin{aligned} x_0 &\sim \mathcal{N}(\mu_0, \sigma_0), \\ x_k &= M(x_{k-1}) + \xi_k, \\ y_k &= x_k + \eta_k, \end{aligned} \quad (4.5)$$

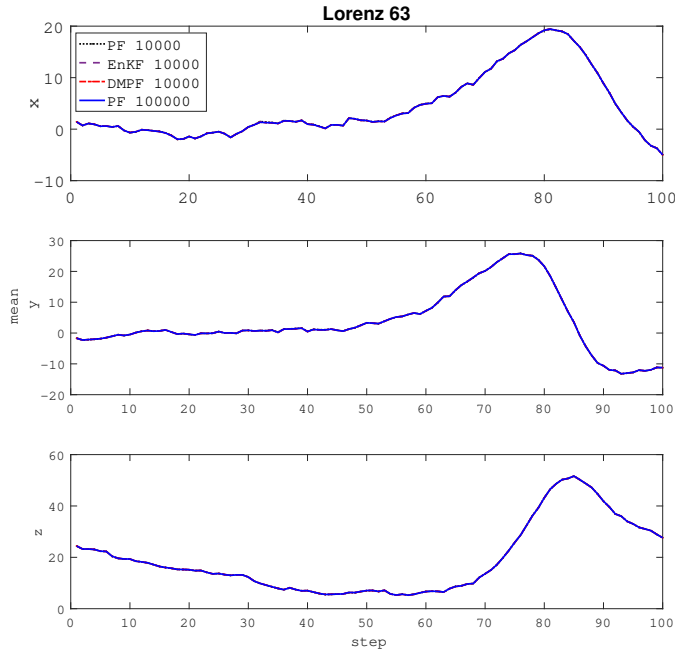


Fig. 2 The posterior means computed by the different methods for the Lorenz 63 model.

where ξ_k and η_k are the model and observation noise respectively. In this example we set $x_0 \sim \mathcal{N}(-0.1, 0.2^2)$, $\Delta t = 0.3$ and the total number of steps to be 50. Moreover, we assume that both ξ_k and η_k follow zero-mean Gaussian distributions with standard deviation 0.01 and 0.8. This is an often used example which admits strongly non-Gaussian posteriors [1, 22].

In this example we also use a simulated true state and generate noisy data from it, where both of them are shown in Fig. 4. We first estimate the states using the PF method with 5×10^5 particles which we use to represent the true posteriors. We then perform the DMPF, standard PF and the EnKF methods to obtain the posterior statistics, with 3×10^4 particles for each method. We compare the posterior means and variances computed by all the methods in Fig. 5. One can see from the plots that, as both the PF and the DMPF methods yield results in a good agreement with the truth, those of the EnKF significantly depart as the time proceeds. The poor performance of the EnKF method in this example can be understood by examining the posterior distributions: in Fig. 6, we plot the posterior distributions computed by all methods at steps 5 and 10 respectively (the distributions of the PF and DMPF methods are obtained by performing a kernel density estimation with the particles). As one can see here, while at $k = 5$ the EnKF approximation remains rather close to the true posterior distribution, it significantly deviates

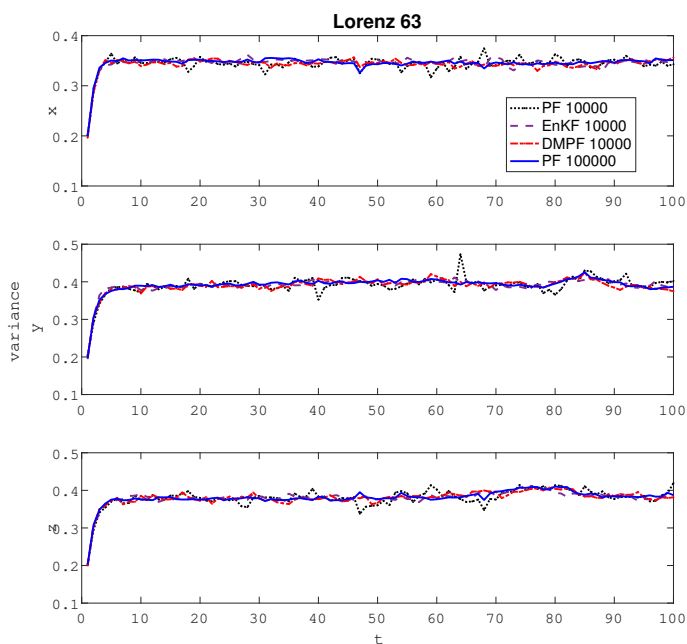


Fig. 3 The posterior variances computed by the different methods for the Lorenz 63 model.

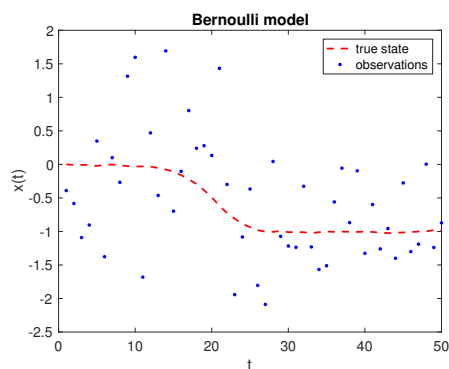


Fig. 4 The true state (dashed lines) and the simulated observations (dots) of the Bernoulli model.

from the the true posterior at $k = 10$ because of the cumulation of the non-Gaussianity as time increases. It is important to note that our DMPF method can nevertheless produce results agreeing with the true posteriors, even though the EnKF method fails.

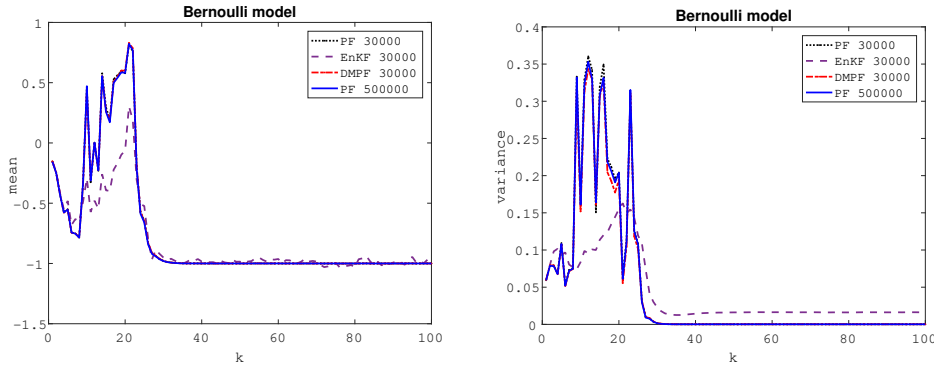


Fig. 5 Left: the posterior means computed by the different methods. Right: the posterior variances computed by the different methods.

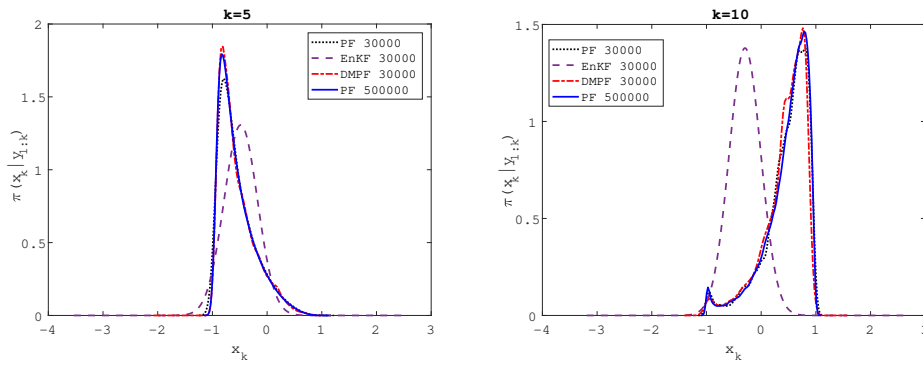


Fig. 6 Left: the posterior distributions at $k = 5$. Right: the posterior distributions at $k = 10$. In both plots, the solid lines are the true posteriors (results of PF with 50,000 particles) and the dashed ones are the EnKF approximations.

4.3 Localization of a car-like robot

Finally we consider a real-world problem, in which the position of a remotely controlled car-like robot is inferred from the on-board GPS data. The kinematic model of the car-like robot is described by the following non-linear system [15]:

$$\begin{aligned}
 \dot{x} &= v \cos(\theta), \\
 \dot{y} &= v \sin(\theta), \\
 \dot{\theta} &= \frac{v}{L} \tan(\phi), \\
 \dot{\phi} &= \omega,
 \end{aligned} \tag{4.6}$$

where (x, y) are the position coordinates of the vehicle, L is its length, θ is the steering orientation angle, ϕ is the front wheel orientation angle, and v and ω are the linear and angular velocities respectively. The schematic illustration of the model is shown in Fig. 7. In this problem, we assume the linear and the

angular velocities v and ω are controlled as follows:

$$\begin{aligned} v &= 0.7|\sin(t)| + 0.1, \\ \omega &= 0.08 \cos(t), \end{aligned} \quad (4.7)$$

The discrete-time version of the model is described by:

$$\begin{pmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \\ \phi_{t+1} \end{pmatrix} = M(t, \begin{pmatrix} x_t \\ y_t \\ \theta_t \\ \phi_t \end{pmatrix}) + \begin{pmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_\theta \\ \epsilon_\phi \end{pmatrix} \quad (4.8)$$

where M stands for the standard fourth-order Runge-Kutta solution of Eq. (4.6) with $\Delta t = 0.05$. In Eq. (4.8), ϵ_x , ϵ_y , ϵ_θ and ϵ_ϕ are the errors in the state process. In particular all these errors are taken to be zero mean Gaussian with standard deviation 0.3.

On the other hand, the GPS makes measurements of the pose (x, y, θ) of the vehicle, and specifically these measurements are taken to be

$$\hat{x} = x + \eta_x, \quad \hat{y} = y + \eta_y, \quad \hat{\theta} = \theta + \eta_\theta,$$

where η_x , η_y and η_θ are the observation noise following $N(0, 0.3^2)$. We shall estimate x , y , θ and ω from these measurements for a time period $T = 5$ that is discretised into 100 steps. The true states of the system are randomly generated and the measurement data are simulated from the generated true states using the prescribed model; both the true states and the associated measurements are plotted in Fig. 8. We emphasise here that no observations are made on the front-wheel angle ϕ and so only the true states of it are plotted in Fig. 8. Once again we apply the three methods to estimate the states of the four parameters in this problem, and with each method we generate 10^4 particles. We then compare the results of the three methods with the true posterior statistics, which are obtained by using the standard PF with 5×10^5 particles. We show the comparison of the results in Fig. 9 (posterior mean) and Fig. 10 (posterior variance). From Fig. 9 we can see that all the posterior means computed by all the methods agree very well with the true posterior mean for parameters x , y and θ ; for parameter ϕ on which we do not have direct observation, the results of the EnKF method deviate significantly from those of the others after around 70 steps. For the posterior variance shown in Fig. 10, we observe that for all four parameters, the results of the standard PF are subject to much fluctuations than those of the DMPF and the EnKF methods; moreover similar to the posterior mean, the posterior variance estimated by the EnKF method also deviate from the true result after about 70 steps. Thus for this problem, the proposed DMPF method has the best overall performance in all the three methods.

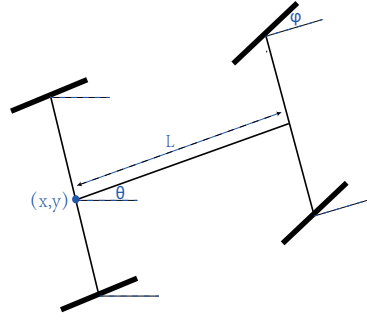


Fig. 7 The schematic illustration of the car-like robot model.

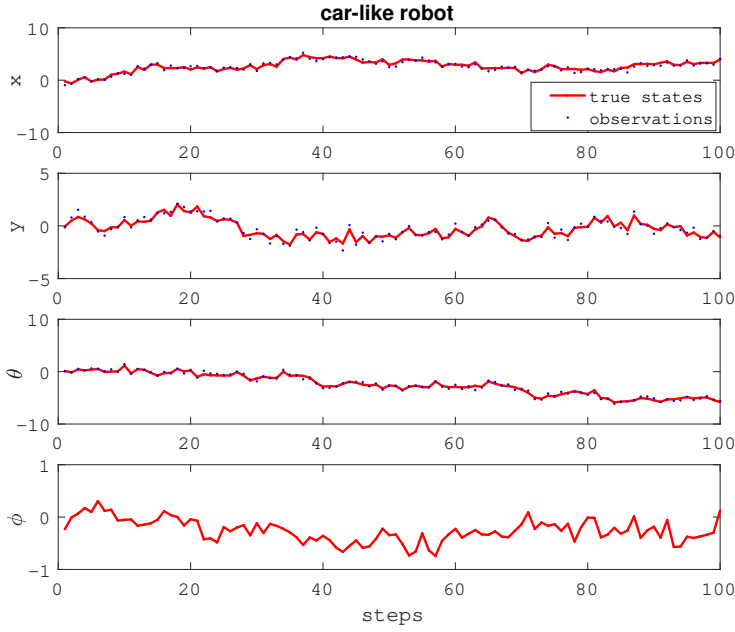


Fig. 8 The true state (dashed lines) and the simulated observations (dots) of the car-like robot model.

5 Conclusions

In summary, we have presented a marginal particle filtering method that samples the posterior distribution in the marginal state space. In particular, we propose a defensive scheme to construct an IS distribution in the marginal space by combining the PF and the EnKF methods, which ensures that the algorithm performs well even when the posterior is strongly non-Gaussian. With numerical examples, we demonstrate that the proposed method has competitive performance against the PF and the EnKF methods. We believe that the

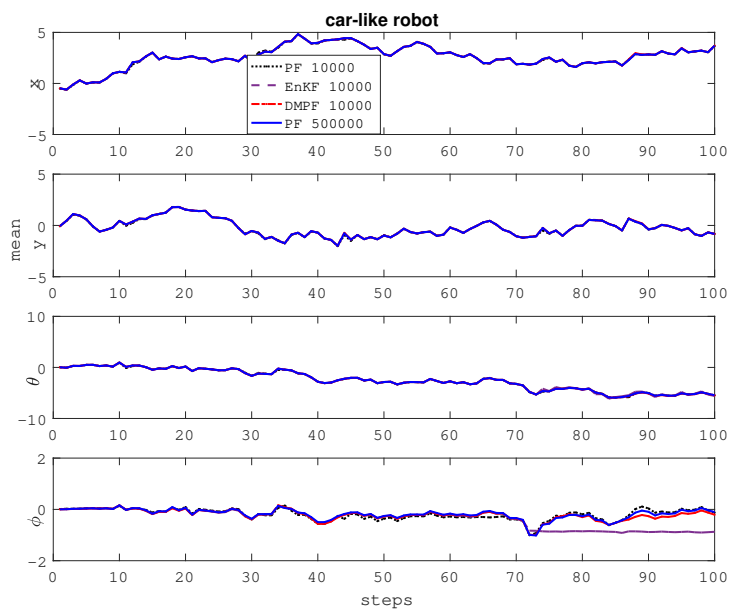


Fig. 9 Left: the true posterior distribution. Right: the KL distance between p_{n-1} and p_n , plotted against the number of iterations.

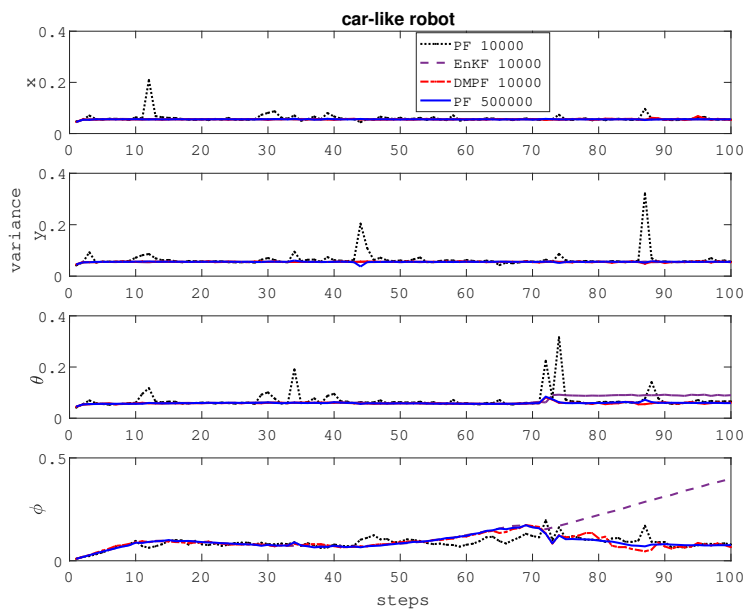


Fig. 10 Left: the true posterior distribution. Right: the KL distance between p_{n-1} and p_n , plotted against the number of iterations.

method can be useful in a wide range of practical data assimilation problems where the other popular existing methods may not perform well.

The proposed method can be improved in several aspects. First, in this work we have mainly considered problems where the marginal state space is of rather low dimensions. On the other hand, for problems of high dimensions, it becomes very challenging to accurately estimate the IS weights in each step. This issue needs to be addressed and so the MPF type of methods can apply to high dimensional problems. Second, as has been discussed in [15], computing the IS weights in each time step is of M^2 complexity where M is the number of particles, and as a result the method become prohibitively expensive for problems requiring a large number of particles. It has been suggested in [15] that some approximation techniques such as the fast multipole method [12] can be used to reduce the computational cost, but further improvement of the efficiency is still needed to make the method useful in large scale problems. Finally in the present form of our method, the EnKF approximation is taken to be a Gaussian distribution, but this restriction can be relaxed by using, for example, mixtures to represent the marginal posteriors [4, 22, 7]. We plan to study these issues and improve the MPF method in the future.

References

1. Apte, A., Hairer, M., Stuart, A., Voss, J.: Sampling the posterior: An approach to non-gaussian data assimilation. *Physica D: Nonlinear Phenomena* **230**(1-2), 50–64 (2007)
2. Arnaud Doucet Nando de Freitas, N.G. (ed.): *Sequential Monte Carlo Methods in Practice*. Springer Science & Business Media (2001)
3. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing* **50**(2), 174–188 (2002)
4. Bengtsson, T., Snyder, C., Nychka, D.: Toward a nonlinear ensemble filter for high-dimensional systems. *Journal of Geophysical Research: Atmospheres* **108**(D24) (2003)
5. Bishop, G., Welch, G., et al.: An introduction to the kalman filter. *Proc of SIGGRAPH, Course* **8**(27599-3175), 59 (2001)
6. Cappé, O., Godsill, S.J., Moulines, E.: An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE* **95**(5), 899–924 (2007)
7. Chen, R., Liu, J.S.: Mixture kalman filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **62**(3), 493–508 (2000)
8. Doucet, A., Godsill, S., Andrieu, C.: On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing* **10**(3), 197–208 (2000)
9. Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering* **12**(656-704), 3 (2009)
10. Evensen, G.: *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media (2009)
11. Ghil, M., Malanotte-Rizzoli, P.: *Data assimilation in meteorology and oceanography*. In: *Advances in geophysics*, vol. 33, pp. 141–266. Elsevier (1991)
12. Greengard, L., Rokhlin, V.: A fast algorithm for particle simulations. *Journal of computational physics* **73**(2), 325–348 (1987)
13. Hesterberg, T.: Weighted average importance sampling and defensive mixture distributions. *Technometrics* **37**(2), 185–194 (1995)
14. Jazwinski, A.H.: *Stochastic processes and filtering theory*. Courier Corporation (2007)
15. Klaas, M., de Freitas, N., Doucet, A.: Toward practical n2 monte carlo: the marginal particle filter. In: *Uncertainty in Artificial Intelligence (UAI)*. AUAI Press (2005)

16. Kuznetsov, L., Ide, K., Jones, C.: A method for assimilation of lagrangian data. *Monthly Weather Review* **131**(10), 2247–2260 (2003)
17. Lorenz, E.N.: Deterministic nonperiodic flow. *Journal of the atmospheric sciences* **20**(2), 130–141 (1963)
18. Martinez-Cantin, R., de Freitas, N., Castellanos, J.A.: Analysis of particle methods for simultaneous robot localization and mapping and a new algorithm: Marginal-slam. In: ICRA, pp. 2415–2420 (2007)
19. Musoff, H., Zarchan, P.: *Fundamentals of Kalman filtering: a practical approach*. American Institute of Aeronautics and Astronautics (2009)
20. Poyiadjis, G., Doucet, A., Singh, S.S.: Particle methods for optimal filter derivative: Application to parameter estimation. In: *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05)*. IEEE International Conference on, vol. 5, pp. v–925. IEEE (2005)
21. Salman, H., Kuznetsov, L., Jones, C., Ide, K.: A method for assimilating lagrangian data into a shallow-water-equation ocean model. *Monthly Weather Review* **134**(4), 1081–1101 (2006)
22. Stordal, A.S., Karlsen, H.A., Nævdal, G., Skaug, H.J., Vallès, B.: Bridging the ensemble kalman filter and particle filters: the adaptive gaussian mixture filter. *Computational Geosciences* **15**(2), 293–305 (2011)
23. Thrun, S., Burgard, W., Fox, D.: *Probabilistic robotics*. MIT press (2005)