

A defensive marginal particle filtering method for data assimilation

Wen, Linjie; Wu, Jiangqi; Lu, Linjun; Li, Jinglai

DOI:

<https://doi.org/10.1137/19M1237430>

Document Version

Version created as part of publication process; publisher's layout; not normally made publicly available

Citation for published version (Harvard):

Wen, L, Wu, J, Lu, L & Li, J 2020, 'A defensive marginal particle filtering method for data assimilation', *SIAM/ASA Journal on Uncertainty Quantification*, vol. 3, no. 3, pp. 1215–1235.
<https://doi.org/10.1137/19M1237430>

[Link to publication on Research at Birmingham portal](#)

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

A Defensive Marginal Particle Filtering Method for Data Assimilation*

Linjie Wen[†], Jiangqi Wu[†], Linjun Lu[‡], and Jinglai Li[§]

Abstract. Particle filtering (PF) is an often used method to estimate the states of dynamical systems. 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. A key issue in the marginal PF method is to construct a good sampling distribution in the marginal space. When the posterior distribution is close to Gaussian, the ensemble Kalman filter (EnKF) method can usually provide a good sampling distribution; however the EnKF approximation may fail completely when the posterior is strongly non-Gaussian. In this work we propose for modest dimensional filtering problems a defensive marginal PF algorithm which constructs a sampling distribution in the marginal space by combining the standard PF and the EnKF approximation using a multiple importance sampling (MIS) scheme. An important feature of the proposed algorithm is that it can automatically adjust the relative weight of the PF and the EnKF components in the MIS scheme in each step, according to how non-Gaussian the posterior is. With numerical examples we demonstrate that the proposed method can perform well regardless of whether the posteriors can be well approximated by Gaussian.

Key words. ensemble Kalman filter, marginal particle filter, data assimilation, multiple importance sampling, particle filter, defensive importance sampling

AMS subject classifications. 62F15, 65C05

DOI. 10.1137/19M1237430

1. Introduction. Assimilation of data into mathematical models is an essential task in many fields of science and engineering, ranging from meteorology [13] to robotics [33]. Simply speaking, data assimilation is a method 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 [20]. The Kalman filter type of methods, which are based on linear control theory and optimization are a popular tool for data assimilation problems. Unfortunately, it is usually challenging to apply such methods to nonlinear systems, as they often require some linearization or approximation processes, e.g., the extended Kalman filter [17] or the ensemble Kalman filter [11]. Sometimes these methods can even fail [19, 30] when strong nonlinearity is present.

*Received by the editors January 10, 2019; accepted for publication (in revised form) February 21, 2020; published electronically DATE.

<https://doi.org/10.1137/19M1237430>

Funding: This work was supported by the NSFC under grants 11301337 and 5150080805.

[†]School of Mathematical Sciences and Institute of Natural Sciences, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai 200240, China (linjiewen@sjtu.edu.cn, jiangqiwu@sjtu.edu.cn).

[‡]Corresponding author. School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (linjunlu@sjtu.edu.cn).

[§]Corresponding author. School of Mathematics, University of Birmingham, Birmingham, UK (j.li.10@bham.ac.uk).

On the other hand, the sequential Monte Carlo (SMC) method (see, e.g., [8]), 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 (more details about the mathematical model are provided in section 2.1 with the specific formulation of the model given by (2.1)), and our goal is to estimate the hidden states $\{\mathbf{u}_t\}_{t=0}^T$ 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 of the hidden states is estimated by the Bayes' rule recursively [9]. 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, PF 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, 10, 4] and the references therein.

The PF method in its very basic form can be understood as drawing weighted samples according to the joint distribution $\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})$ 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) [10]. On the other hand, in practice one is often only interested in the marginal distribution $\pi(\mathbf{u}_t|\mathbf{y}_{0:t})$, which implies that it is unnecessary to sample the high-dimensional joint distribution $\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})$. Instead, one can perform IS only in the marginal space of $\pi(\mathbf{u}_t|\mathbf{y}_{0:t})$, and based on this idea, a method called marginal particle filter (MPF) was proposed in [18] to alleviate the degeneracy issue. The method later has found applications in the estimation of filter derivative [29] and robot localization [25]. A key in the MPF method is to construct an IS or proposal distribution that can approximate well the marginal posterior $\pi(\mathbf{u}_t|\mathbf{y}_{0:t})$ at each time step. One very natural idea is to construct the proposal distribution using the ensemble Kalman filter (EnKF). The basic idea behind EnKF is to assume the posterior distribution at each step follows a Gaussian distribution with the mean and covariance estimated from samples, and then update the samples according to the Kalman filter formulation. We emphasize here that, the difference between a direct use of EnKF and using it as an proposal in a PF/SMC scheme is that the SMC scheme can correct for the bias in the EnKF particles (due to the Gaussian approximation) by assigning an IS weight to each particle. The idea of using the EnKF approximation as a proposal distribution in PF is not new: for example, it has been used in [28] to construct an independent sequential IS distribution in the PF framework, and later its use in the MPF scheme is discussed in [26]. A limitation of the EnKF-based proposal distribution is that (just like the EnKF method itself) it may result in extremely poor estimates or even fail completely (see the example in section 4.1) when the posterior is strongly non-Gaussian; unfortunately it is almost impossible to know whether the posteriors are close to Gaussian in advance. The main contribution of the work is to propose defensive scheme to prevent such a failure of the estimation: it can automatically adjust between a proposal based on the EnKF approximation and a standard PF proposal and, as a result, the proposed algorithm may perform well regardless of whether the posteriors are close to Gaussian. Specifically our defensive scheme combines the EnKF-based proposal and the standard PF proposal using

the multiple IS method (also known as the deterministic mixture). In fact, combining several different IS distributions using multiple IS [34] to prevent the risk of failure of a single IS distribution is a very popular safeguard measure in the IS literature, e.g., [15, 27, 7], and the use of MPF here makes it possible to implement the method in the filtering problems. A key issue in the multiple IS method is to determine the weight of each component (in the present setting the two components are EnKF and PF, respectively). Ideally at each time step we want the method to choose the EnKF approximation if the posterior distribution is close to Gaussian and the standard PF otherwise. To achieve this goal, we provide an algorithm that can automatically determine the relative weight between the EnKF and the PF components by minimizing the variance of the resulting IS weight. We emphasize that such an ability to automatically adjust the weights of the EnKF and the PF components is a main feature that sets the proposed algorithm apart from many existing schemes combining EnKF and PF, notably [6, 12, 21, 28, 32]. Finally it is important to note that the proposed defensive MPF scheme does not depend on the EnKF approximation, and it can be used with any proposal distribution constructed in the marginal space.

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 these types 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. The problem setup. We consider the filtering problem in a generic form:

$$(2.1a) \quad \mathbf{u}_t \sim f_t(\cdot | \mathbf{u}_{t-1}), \quad \mathbf{u}_0 \sim \pi_0(\mathbf{u}_0),$$

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

where $\mathbf{u}_t \in \mathbb{R}^{n_u}$ denotes the state vector at time t , $\mathbf{y}_t \in \mathbb{R}^{n_v}$ are the observed data at time t , $f_t(\cdot | \mathbf{u}_{t-1})$ is the distribution of \mathbf{u}_t conditional on \mathbf{u}_{t-1} , H_t , an $n_v \times n_u$ matrix, is the observation operator at time t , and $\boldsymbol{\eta}_t$ is the observation noise. In this work, we shall assume that the observation noise $\boldsymbol{\eta}_t$ is Gaussian and the noise at different time steps is independent of each other. 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 , and the observation \mathbf{y}_{t+1} is independent of $\mathbf{y}_{0:t}$ conditional on \mathbf{u}_t :

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

which will be used in the derivation of the PF method. Note here that throughout this paper we use π as a generic notation of probabilistic distribution, the actual meaning of which is specified by its arguments.

2.2. The PF. 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

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

The PF (or SMC) method allows one 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 PF method, and it is easier to start with a standard Monte Carlo (MC) estimation. Suppose that there is a real-valued function $h(\cdot) : \mathbf{R}^{T \times n_u} \rightarrow \mathbf{R}$ and we are interested in the expectation

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

which can be estimated with an 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 IS. The IS method introduces an importance distribution $q(\mathbf{u}_{0:T})$ and rewrites

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

with $w(\mathbf{u}_{0:T}) = \pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})/q(\mathbf{u}_{0:T})$ 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 $q(\mathbf{u}_{0:T})$, and it can also be verified that the IS estimator is also an unbiased one for I . The IS requires one 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 (2.2), we can write the posterior distribution $\pi(\mathbf{u}_{0:T}|\mathbf{y}_{0:T})$ in the form of

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

where $Z_T = \pi(\mathbf{y}_{0:T})$ is the normalization constant. Similarly, we can also assume that the importance distribution $q(\mathbf{u}_{0:T})$ is 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

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

and

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

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

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

We note that, in the formulation above, we do not have knowledge of the normalization constant Z_T . In the implementation, however, we can simply set the normalization constant to be 1, and renormalize the computed weights. 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 (2.4)–(2.6) (and taking $Z_T = 1$), and then renormalize the weights as

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

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

Algorithm 2.1 The PF algorithm.

1. At $t = 0$, sample $\{\mathbf{u}_0^m\}_{m=1}^M \sim q_0(\mathbf{u}_0)$ and compute $\{w_0^m = w_0(\mathbf{u}_0^m)\}_{m=1}^M$ using (2.4); renormalize the weights: $W_0^m = \frac{1}{\sum_{m'=1}^M w_0^{m'}} w_0^m$.
 2. At $t > 1$:
 3. prediction step: for each $m = 1, \dots, M$, draw $\mathbf{u}_t^m \sim q_t(\mathbf{u}_t|\mathbf{u}_{t-1}^m)$;
 4. updating step: for each $m = 1, \dots, M$, compute the incremental function α_t^m from (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 PF 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

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

In the PF algorithm, the variance of the importance weight $w_t(\mathbf{u}_{0:t})$ will increase over time, and thus as 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, the procedure of which can be found in, for example, [8, 10].

2.3. The EnKF. We now discuss the EnKF method which computes a Gaussian approximation of the posterior distribution $\pi(\mathbf{u}_t|\mathbf{y}_{0:t})$. For simplicity we assume that the observation operator H_t is linear and the observation noise $\boldsymbol{\eta}_t$ is Gaussian. Now 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 $\tilde{\Sigma}_t$. 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:

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

where I is the identity matrix and

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

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:

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

and, as is mentioned earlier, the prior distribution $\pi(\mathbf{u}_t|\mathbf{y}_{0:t-1})$ can be approximated by $N(\tilde{\boldsymbol{\mu}}_t, \tilde{\Sigma}_t)$. It is not hard to see that the posterior distribution is also Gaussian with mean $\boldsymbol{\mu}_t$ and covariance Σ_t given by (2.9). Moreover it can be verified that the samples

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

follow the distribution $N(\boldsymbol{\mu}_t, \Sigma_t)$, provided that $\tilde{\mathbf{u}}_t^m \sim N(\tilde{\boldsymbol{\mu}}_t, \tilde{\Sigma}_t)$ 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 $\tilde{\mathbf{u}}_t^m = f_t(\tilde{\mathbf{u}}_{t-1}^m) + \boldsymbol{\epsilon}_t^m$;
- updating step: for each $m = 1, \dots, M$, compute $\mathbf{u}_t^m = \tilde{\mathbf{u}}_t^m + K_t(\mathbf{y}_t - H_t \tilde{\mathbf{u}}_t^m - \boldsymbol{\eta}_t^m)$.

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 major limitation of the method.

3. The defensive MPF algorithm.

3.1. The MPF. As is discussed in section 2.2, the standard PF 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

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

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 $\pi(\mathbf{u}_t | \mathbf{y}_{0:t})$. We first note that a special choice of the IS distribution is

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

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 [18], a kernel-based IS distribution is suggested:

$$(3.3) \quad \pi(\mathbf{u}_t | \mathbf{y}_{0:t-1}) = \int \pi(\mathbf{u}_t | \mathbf{u}_{t-1}) \pi(\mathbf{u}_{t-1} | \mathbf{y}_{0:t-1}) d\mathbf{u}_{0:t-1},$$

that is, this integral is approximated by

$$(3.4) \quad \pi(\mathbf{u}_t | \mathbf{y}_{0:t-1}) \approx \sum_{m=1}^M w_{t-1}^m \pi(\mathbf{u}_t | \mathbf{u}_{t-1}^m),$$

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 , and each $\pi(\mathbf{u}_t | \mathbf{y}_{0:t-1})$ is obtained using a weighted Kernel density estimation (KDE) method. As a result the method requires one to perform a weighted KDE procedure at each time step, which can be computationally intensive even with some fast KDE algorithms (e.g., the dual-tree methods). We discuss an alternative approach to constructing the IS distribution in the next section.

3.2. The EnKF-based IS distribution. When the marginal posterior is close to Gaussian, the EnKF method can compute a good IS distribution in a very efficient 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 MPF. Specifically, letting $\{\mathbf{u}_t^m\}_{m=1}^M$ be the posterior ensemble at time t obtained with the EnKF formulation, we use the following procedure to compute the IS distribution:

Algorithm 3.1 Estimating the IS distribution from the ensemble.

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

$$(3.5) \quad \mu_{\text{En}} = \frac{1}{M} \sum_{m=1}^M \mathbf{u}_t^m, \quad \Sigma_{\text{En}} = \frac{1}{M-1} \sum_{m=1}^M (\mathbf{u}_t^m - \mu_{\text{En}})(\mathbf{u}_t^m - \mu_{\text{En}})^T;$$

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

2. draw M samples $\mathbf{u}_t^1, \dots, \mathbf{u}_t^M$ from q'_{EnKF} , and compute the weight of each sample, and renormalize it:

$$w_t^m = \frac{\pi(\mathbf{u}_t^m | y_{0:t})}{q'_{\text{EnKF}}(\mathbf{u}_t^m)}, \quad W_t^m = \frac{w_t^m}{\sum_{m'=1}^M w_t^{m'}};$$

3. estimate the mean and covariance of the weighted ensemble $\{(\mathbf{u}_t^m, W_t^m)\}$:

$$(3.6) \quad \mu_{\text{updated}} = \sum_{m=1}^M W_t^m \mathbf{u}_t^m, \quad \Sigma_{\text{updated}} = \sum_{m=1}^M W_t^m (\mathbf{u}_t^m - \mu_{\text{updated}})(\mathbf{u}_t^m - \mu_{\text{updated}})^T;$$

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

It is worth noting that the EnKF ensemble does not exactly follow the posterior distribution, and thus we choose not to directly use q'_{EnKF} in Algorithm 3.1, i.e., the Gaussian approximation estimated from the EnKF ensemble, as the IS distribution. Instead, we introduce additional steps (steps 2 and 3 in Algorithm 3.1), 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.

3.3. The defensive MPF. As has been discussed earlier, when strong nonlinearity is present, the true posterior can no longer be approximated by a Gaussian distribution. In this case the IS distribution obtained with the EnKF method may deviate significantly from the true marginal posterior, leading to erroneous estimates of the states. To address the issue, we use the idea of *multiple IS* (MIS), also known as the deterministic mixture [27]. That is, 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 distribution, which in our case is the standard PF distribution, yielding

$$(3.7) \quad q_t(\mathbf{u}_t | a) = a q_{\text{EnKF}}(\mathbf{u}_t) + (1-a) q_{\text{PF}}(\mathbf{u}_t),$$

where q_{EnKF} is the Gaussian distribution computed with the EnKF procedure described above, q_{PF} is the distribution given by (3.2), which, as discussed earlier, is equivalent to the standard PF, and $a \in [0, 1]$ is the weight of the EnKF component, which will be referred to as the weight parameter hereafter. This mixture distribution 3.7 is the defensive IS scheme proposed in [15]. If this mixture IS distribution is used in (3.1) to estimate I , the estimation variance of the mixture is bounded by [15]

$$\sigma_{\text{DIS}}^2 \leq \frac{1}{1-a} \sigma_{\text{PF}}^2 + \frac{1-a}{a} I^2$$

with σ_{PF}^2 being the estimation variance of q_{PF} , regardless of how large the estimation variance of q_{EnKF} is. The MIS scheme is very similar to the mixture IS distribution in (3.7), except that it generates a fixed number of samples from each component, namely, aM samples from q_{EnKF} and $(1-a)M$ from q_{PF} and, hence, the name deterministic mixture. It has been shown that the use of MIS often yields more accurate and robust estimates than the standard mixture IS distribution [15, 27], and so here we adopt the MIS method as our defensive scheme.

An important issue here is how to compute the IS weight of each sample in the MIS scheme. In [34], the authors recommend the so-called balance heuristic weight:

$$(3.8) \quad w_t(\mathbf{u}_t) = \frac{\pi(\mathbf{u}_t|\mathbf{y}_{0:t})}{aq_{\text{EnKF}} + (1-a)q_{\text{PF}}} = \frac{1}{\frac{a}{w_{\text{EnKF}}} + \frac{(1-a)}{w_{\text{PF}}}},$$

where

$$(3.9) \quad w_{\text{EnKF}} = \frac{\pi(\mathbf{y}_t|\mathbf{u}_t)\pi(\mathbf{u}_t|\mathbf{y}_{0:t-1})}{\pi(\mathbf{y}_t|\mathbf{y}_{0:t-1})q_{\text{EnKF}}(\mathbf{u}_t)}, \quad w_{\text{PF}} = \pi(\mathbf{y}_t|\mathbf{u}_t)/\pi(\mathbf{y}_t|\mathbf{y}_{0:t-1}).$$

Computing w_{PF} is rather straightforward, but computing w_{EnKF} involves the evaluation of the integral in (3.3). In practice, we can use the weighted ensemble $\{(\mathbf{u}_{t-1}^m, w_{t-1}^m)\}_{m=1}^M$ follows the distribution $\pi(\mathbf{u}_{t-1}|\mathbf{y}_{0:t-1})$ to approximate (3.4).

We note that, as evaluating (3.4) requires summing over M particles, computing all the weights is of M^2 complexity, which can be highly intensive when the number of particles is large. However, by using the fast multipole method [14] one can reduce the computational cost to $M \log M$ (see [18] for more discussions). Another important matter in the proposed method is to determine the value of the weight parameter a , which is discussed in section 3.4. We hereby provide the complete defensive MPF (DMPF) algorithm in Algorithm 3.2.

Algorithm 3.2 The DMPF algorithm.

At $t = 0$:

Prediction: sample $\{\tilde{\mathbf{u}}_0^m\}_{m=1}^M$ from $\pi_0(\cdot)$;

Updating: $u_0^m = \tilde{u}_0^m + K_0(y_0 - H_0\tilde{\mathbf{u}}_0^m - \eta_t^m)$ for $m = 1, \dots, M$;

Compute q_{EnKF} using Algorithm 3.1 and particles $\{\hat{\mathbf{u}}_0^m\}_{m=1}^M$;

Draw M particles from q_0 from (3.7) for $t = 0$, and compute the weights using (3.9), yielding $\{(\mathbf{u}_0^m, w^m)\}_{m=1}^M$;

for $t=1, \dots, T$ **do**

 Prediction: for each $m = 1, \dots, M$, draw $\tilde{\mathbf{u}}_t^m = f_t(\tilde{\mathbf{u}}_{t-1}^m) + \epsilon_t^m$;

 Updating: $\mathbf{u}_t^m = \tilde{\mathbf{u}}_t^m + K_t(y_t - H_t\tilde{\mathbf{u}}_t^m)$ for $m = 1, \dots, M$;

 Compute q_{EnKF} using Algorithm 3.1 and particles $\{\hat{\mathbf{u}}_t^m\}_{m=1}^M$;

 Estimate the weight parameter a by solving the optimization problem (3.12);

 Draw M particles from q_t given by (3.7), and compute the weights using (3.9), yielding $\{(\mathbf{u}_t^m, w^m)\}_{m=1}^M$;

end

3.4. Optimizing the weight parameter. It is highly important to choose an appropriate value for the weight parameter a in the DMPF algorithm. In previous works on MIS, a fixed a is often used and, in particular, it is suggested in [15] that the parameter should be chosen between 0.1 – 0.5. In this section, we provide a method that can automatically determine the value of a at each time step. The method can assign more weight to the EnKF component if the posterior is close to Gaussian and to the PF component otherwise.

As is well known in the PF literature, the optimal IS distribution should yield equal IS weights, i.e., $w_t(\mathbf{u}_t) = 1$ for all \mathbf{u}_t , which is usually not possible in practice. Nevertheless, this gives us the idea that the variance of the weight function associated with distribution $q_t(\mathbf{u}_t|a)$ should be as small as possible, and so we can determine the value of a by minimizing the variance of $w_t(\mathbf{u}_t, a)$ (here we use the notation $w_t(\mathbf{u}_t, a)$ to emphasize the dependence of w_t on a):

$$(3.10) \quad \min_{a \in [0,1]} \text{Var}_{q_t(\mathbf{u}_t|a)}[w_t(\mathbf{u}_t, a)] = \min_{a \in [0,1]} \mathbb{E}_{q_t(\mathbf{u}_t|a)}[(w_t(\mathbf{u}_t, a) - 1)^2]$$

$$(3.11) \quad = \min_{a \in [0,1]} \int (w_t(\mathbf{u}_t, a) - 1)^2 q_t(\mathbf{u}_t|a) d\mathbf{u}_t.$$

Optimizing the variance directly is usually not feasible, and so a natural idea is to optimize its sample-average approximation:

$$\min_{a \in [0,1]} \frac{1}{M} \sum_{m=1}^M (w_t(\mathbf{u}_t^m, a) - 1)^2,$$

where $\{\mathbf{u}_t^m\}_{m=1}^M$ are drawn from distribution $q_t(\mathbf{u}_t|a)$. However, it is actually undesirable to use this sample average approximation, in that, whenever a is updated, we will have to generate new samples and compute the associated weights, which, as is discussed earlier, is of M^2 complexity. To address the issue, we choose a default value of a , say a_0 (in this work we choose $a_0 = 0.5$), and apply an IS simulation with distribution $q_t(\mathbf{u}_t|a_0)$ to estimate the variance, namely,

$$(3.12) \quad \min_{a \in [0,1]} \frac{1}{M} \sum_{m=1}^M (w_t(\mathbf{u}_t^m, a) - 1)^2 w_t(\mathbf{u}_t^m, a_0),$$

where samples are drawn from $q_t(\mathbf{u}_t|a_0)$. Solving the optimization program (3.12) does not affect the computational efficiency of the algorithm much and the reason is twofold. First this optimization problem is rather easy to solve as it is only a one-dimensional problem, specially as our method does not require high accuracy in the solution here. Second by design, solving this optimization problem does not require additional evaluations of the dynamical model f_t which is usually the most computationally intensive part of the problem, or the weights that are of M^2 computational complexity.

4. Numerical examples. In this section we provide three numerical examples to compare the performance of the proposed DMPF method, with EnKF, PF, as well as two existing methods that combine EnKF and PF: the weighted EnKF (WEnKF) method in [28] and the

ensemble Kalman particle filter (EnKPF) in [12]. Here we provide a brief introduction to the two methods. The WEnKF method uses the EnKF to construct a proposal distribution in the PF framework, yielding weighted samples [28]. The EnKPF uses the progressive correction idea and introduces an “intermediate” posterior distribution at each time step; it then updates from the prior to the intermediate distribution using EnKF and from the intermediate distribution to the complete posterior using PF [12]. We note here that the EnKPF method is particularly similar to that proposed in the present work, as it also provides a continuous transition indexed by $a \in [0, 1]$ (γ in [12]) between EnKF and PF, which is determined automatically.

4.1. Bernoulli model. Our first example is the Bernoulli equation,

$$(4.1) \quad \dot{x} - x = -x^3, \quad x(t_0) = x_0,$$

which admits an analytical solution,

$$(4.2) \quad x(t) = M(x_0, \Delta t) = x_0 \times (x_0^2 + (1 - x_0^2)e^{-2\Delta t})^{-1/2},$$

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

$$(4.3) \quad \begin{aligned} x_0 &\sim N(\mu_0, \sigma_0), \\ x_k &= M(x_{k-1}, \Delta t) + \xi_k, \\ y_k &= x_k + \eta_k, \end{aligned}$$

where ξ_k and η_k are the model and observation noise, respectively. In this example we set $x_0 \sim N(-0.1, 0.2^2)$, $\Delta t = 0.3$, and the total number of steps to be 40. 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 with strongly non-Gaussian posteriors [1, 32].

We generate a true state and the associated data points from the model (4.3), which are shown in Figure 1. We first draw 5×10^5 particles using the standard PF method to represent the true posteriors. We then perform the five aforementioned methods to estimate the states, each with 10^4 particles. In EnKPF, the constrained diversity τ used to determine the weights (see [12] for details) is taken to be $[0.9, 1]$. We compare the posterior means and variances computed by all the methods in Figure 2. One can see from the plots that the DMPF method yields results in good agreement with the truth, while those of the EnKF significantly depart as the time proceeds. The results of the EnKPF method are better than EnKF, but still deviate evidently from those of PF and DMPF. The poor performance of the EnKF method in this example can be understood by examining the posterior distributions: in Figure 3, we plot the posterior distributions computed by all methods at steps 5 and 10, respectively, where the distributions of the PF and DMPF methods are obtained by performing a KDE 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 from the true posterior at $k = 10$ because of the cumulation of the non-Gaussianity as time increases. We also implement the WEnKF, the results of which are highly unstable and so are not plotted in these figures.

Moreover, recall that in both DMPF and EnKPF a scalar parameter $a \in [0, 1]$ is used to describe the relative strength of the EnKF and the PF components in the algorithm, and

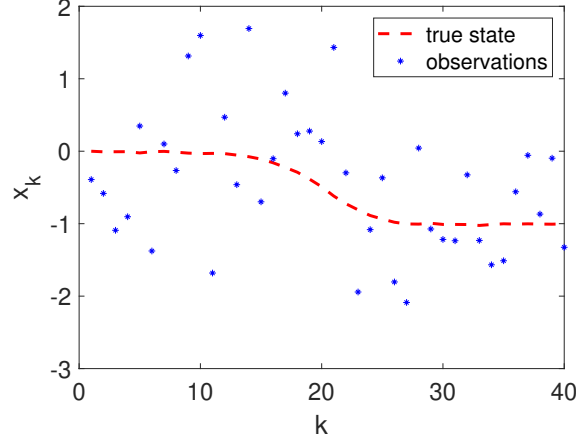


Figure 1. The true state (dashed lines) and the simulated observations (dots) of the Bernoulli model.

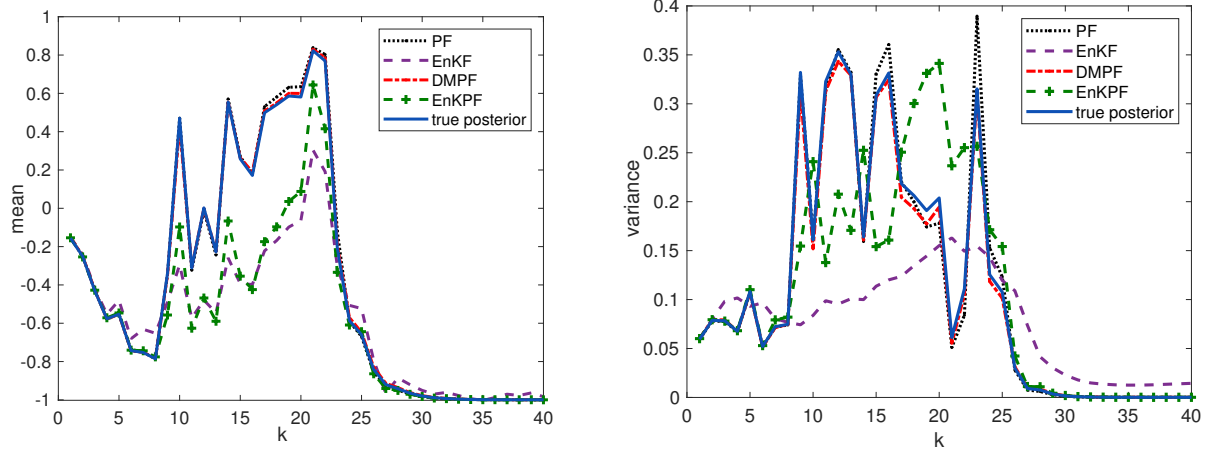


Figure 2. Left: the posterior means computed by the different methods. Right: the posterior variances computed by the different methods.

in both algorithms, the parameter is automatically determined. We plot the parameter a estimated in both methods at each time step in Figure 4. It can be seen from the figure that, in most of the steps, the weight parameter a is close to 0 in both methods, suggesting that both methods detect that the EnKF approximation is not a good approximation to the posterior in most steps and are able to choose suitable values for a accordingly. It should also be noted here that the EnKF method usually employs a rather small number of particles, e.g., several hundreds, and here we intentionally uses a rather large number of particles to demonstrate that the large bias in the EnKF approximation cannot be reduced by increasing the number of samples. In summary, this example demonstrates that, in the presence of strong non-Gaussianity, the Gaussian approximation computed by the EnKF method may fail completely, while our DMPF method can nevertheless produce accurate posterior estimates.

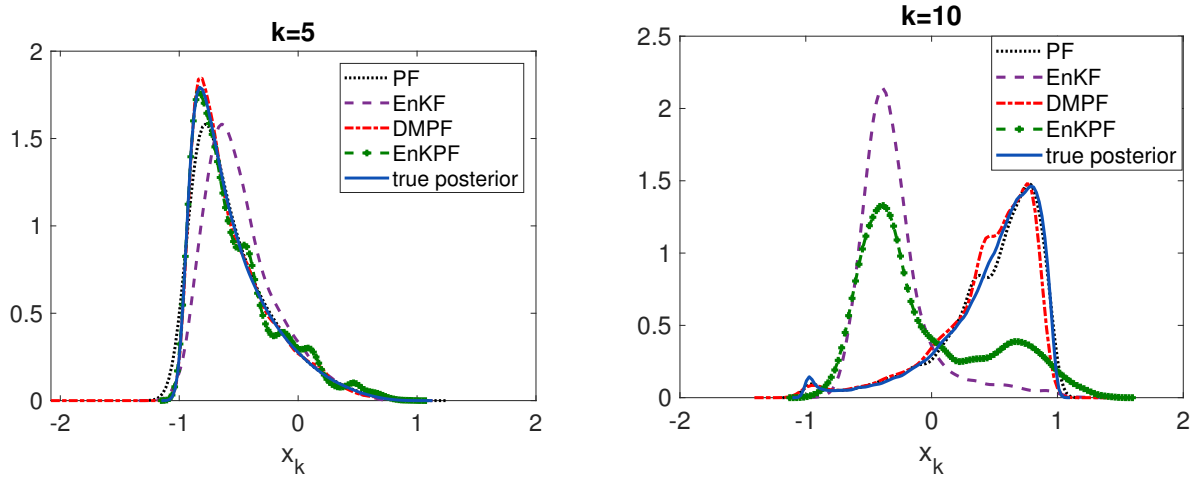


Figure 3. Left: the posterior distributions at $k = 5$. Right: the posterior distributions at $k = 10$. In both plots, the solid lines are the true posteriors and the dashed ones are the EnKF approximations.

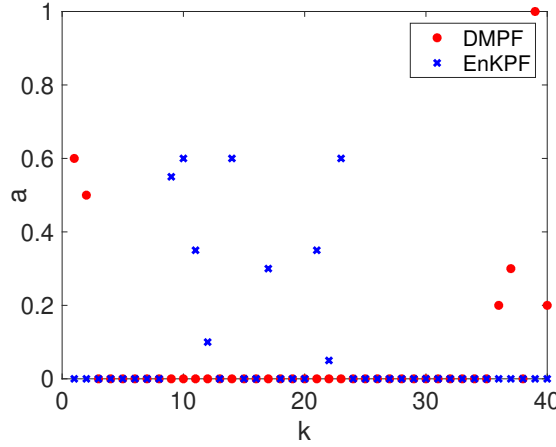


Figure 4. The weight parameter a computed at each time step in DMPF and EnKPF in the Bernoulli model.

4.2. Lorenz 63 system. Our second example is the classical Lorenz 63 system [24], an often used benchmark problem for testing data assimilation algorithms. Specifically the system is described by,

$$(4.4) \quad \begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = x(\rho - z) - y, \\ \dot{z} = xy - \beta z. \end{cases}$$

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

$$(4.5) \quad \begin{cases} x_{t+1} = x_t + \sigma(y_t - x_t)\Delta t + \xi_t^x, \\ y_{t+1} = y_t + (x_t(\rho - z_t) - y_t)\Delta t + \xi_t^y, \\ z_{t+1} = z_t + (x_t y_t - \beta z_t)\Delta t + \xi_t^z, \end{cases}$$

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 independent and identically distributed (i.i.d.) zero-mean Gaussian with standard deviation σ_ξ . Moreover at each time t the observed data are 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 $\sigma = 10$, $\rho = 28$, $\beta = 8/3$, $\Delta t = 0.03$, and 150 steps, 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 $\sigma_\eta = 1$. In this example we also use a simulated true state and generate noisy data from it, where both of them are shown in Figure 5. We use this example to quantitatively compare the performance of these methods. We repeat the simulations 1000 times for all methods with 10^4 particles each time, and so we can examine the statistical behavior of the methods. Also, as a comparison reference, we also perform a PF with 5×10^5 particles to represent the true posteriors. In Figure 6, we show the weight parameter a computed in one of the DMPF and EnKPF simulations, and one can see from the figure that, unlike the first example, the parameter a calculated in DMPF is close to 1 in most of steps, indicating that the posteriors in those steps can be well approximated by the EnKF approximation.

To quantitatively compare the performance of the methods, we compute the root mean squared error (RMSE) for posterior mean and variance, namely, in the j th simulation, we estimate the posterior mean $(\hat{\mathbf{x}}_j, \hat{\mathbf{y}}_j, \hat{\mathbf{z}}_j)$, and the posterior variance $(\mathbf{V}_j^x, \mathbf{V}_j^y, \mathbf{V}_j^z)$, and the RMSE is then calculated as,

$$\begin{aligned} \text{RMSE}_{\text{mean}} &= \frac{1}{1000} \sum_{j=1}^{1000} [(\hat{\mathbf{x}}_j - \mathbf{x})^2 + (\hat{\mathbf{y}}_j - \mathbf{y})^2 + (\hat{\mathbf{z}}_j - \mathbf{z})^2]^{\frac{1}{2}}, \\ \text{RMSE}_{\text{var}} &= \frac{1}{1000} \sum_{j=1}^{1000} [(\mathbf{V}_j^x - \mathbf{V}^x)^2 + (\mathbf{V}_j^y - \mathbf{V}^y)^2 + (\mathbf{V}_j^z - \mathbf{V}^z)^2]^{\frac{1}{2}}, \end{aligned}$$

where $(\mathbf{x}, \mathbf{y}, \mathbf{z})$, and $(\mathbf{V}^x, \mathbf{V}^y, \mathbf{V}^z)$ are the true values of the posterior mean and variance (computed with 5×10^5 PF particles). It should be clear that the RMSE is a measure of the estimation error: smaller RMSE usually indicates lower estimation error. In Figure 7, we plot the RMSE of the posterior mean (left) and variance (left) for all methods, and in Table 1 we show the RMSE averaged over all time steps. Some conclusions can be drawn from the figures and the table. First the EnKF performs significantly better than PF, and this can be understood as that the posteriors in this problem can be well approximated by the EnKF (Gaussian) approximation, and the PF method does not take advantage of that. On the other hand, the DMPF and the EnKPF methods perform similarly to the EnKF, suggesting that both methods can detect the fact that EnKF approximates the posteriors well and adjust the algorithm accordingly to take advantage of it.

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.

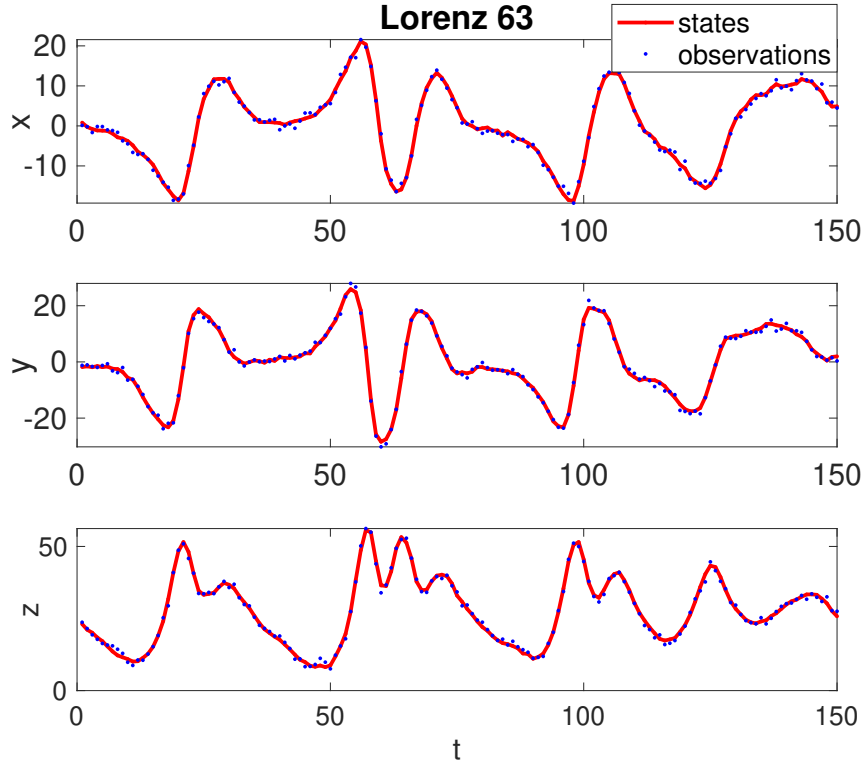


Figure 5. The true state (dashed lines) and the simulated observations (dots) of the Lorenz 63 model.

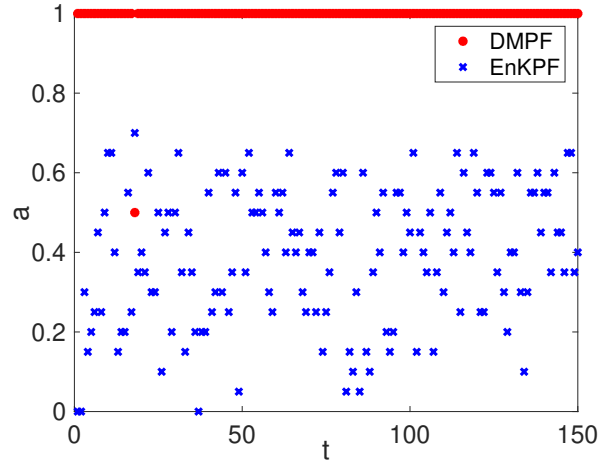


Figure 6. Parameter a in the DMPF and the EnKPF methods for the Lorenz 63 model.

The kinematic model of the car-like robot is described by the following nonlinear system [18]:

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

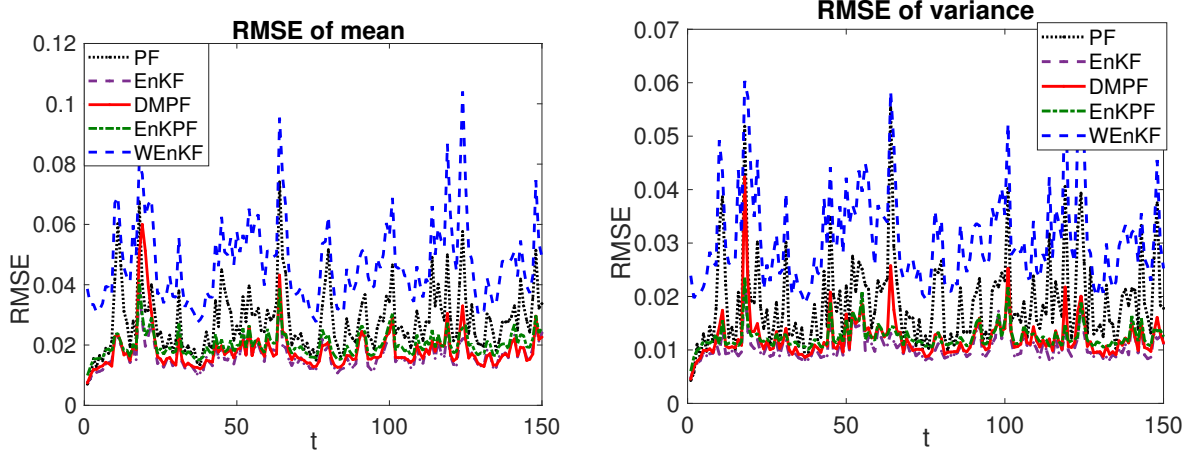


Figure 7. The RMSE of the posterior mean (left) and of the posterior variance (right) for the Lorenz 63 model.

Table 1

RMSE averaged over all time steps in the Lorenz 63 model.

| - | mean | variance |
|--------------|-------|----------|
| PF | 0.028 | 0.019 |
| EnKF | 0.017 | 0.010 |
| DMPF | 0.018 | 0.012 |
| EnKPF | 0.020 | 0.012 |
| WEnKF | 0.047 | 0.031 |

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 Figure 8. In this problem, we assume the linear and the angular velocities v and ω are controlled as follows:

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

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

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

where M represents the standard fourth-order Runge–Kutta solution of (4.6) with $\Delta t = 0.05$. In (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

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

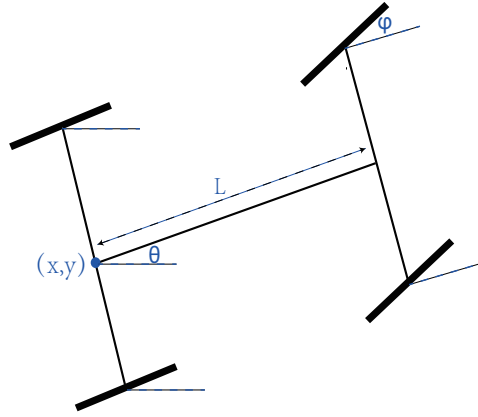


Figure 8. The schematic illustration of the car-like robot model.

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 discretized 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 Figure 9. We emphasize here that no observations are made on the front-wheel angle ϕ and so only the true states of it are plotted in Figure 9. The constrained diversity τ in EnKPF is taken to be $[0.25, 0.5]$.

We implement all five methods to estimate the states of the four parameters in this problem, each with 10^4 particles. Once again we repeat the simulations of each method 1000 times and calculate the RMSE of the posterior mean and variance. In the calculation of RMSE the true posterior mean and variance are obtained by using the standard PF with 5×10^5 particles. We show the results in several figures. First, in Figure 10 (left) we show the weight parameter a computed in one trial for both DMPF and EnKPF. We observe rather mixed results in DMPF: the weight a is estimated to be around 1 in about 40 steps and significantly less than 1 in the rest. To further understand the behavior of the DMPF method, we consider the time step at $t = 72$, where a is estimated to be 0, which indicates that the algorithm detects strong non-Gaussianity in the step. We show the scatter plots of the particles (the particles are resampled so they all have the same weights) at this step, and one can see from the plots that the posterior samples are distributed very differently from Gaussian.

We then plot the RMSE of the mean and the variance in Figure 11 for all methods, and we summarize the RMSE averaged over all time steps in Table 2. The figure shows that the EnKF works well for time steps before $t = 70$, and its results become significantly inaccurate shortly after that, which makes its average RMSE over all time steps the worst among all methods. From Table 2, we can see that DMPF and EnKPF have the best performance among the five methods, where the EnKPF yields lower variance RMSE. It should be noted here that a limitation of the EnKPF is that, its performance depends critically on the constrained diversity τ , which has to be specified by the user. On the other hand, the DMPF method does not have such free parameters that need to be tuned.

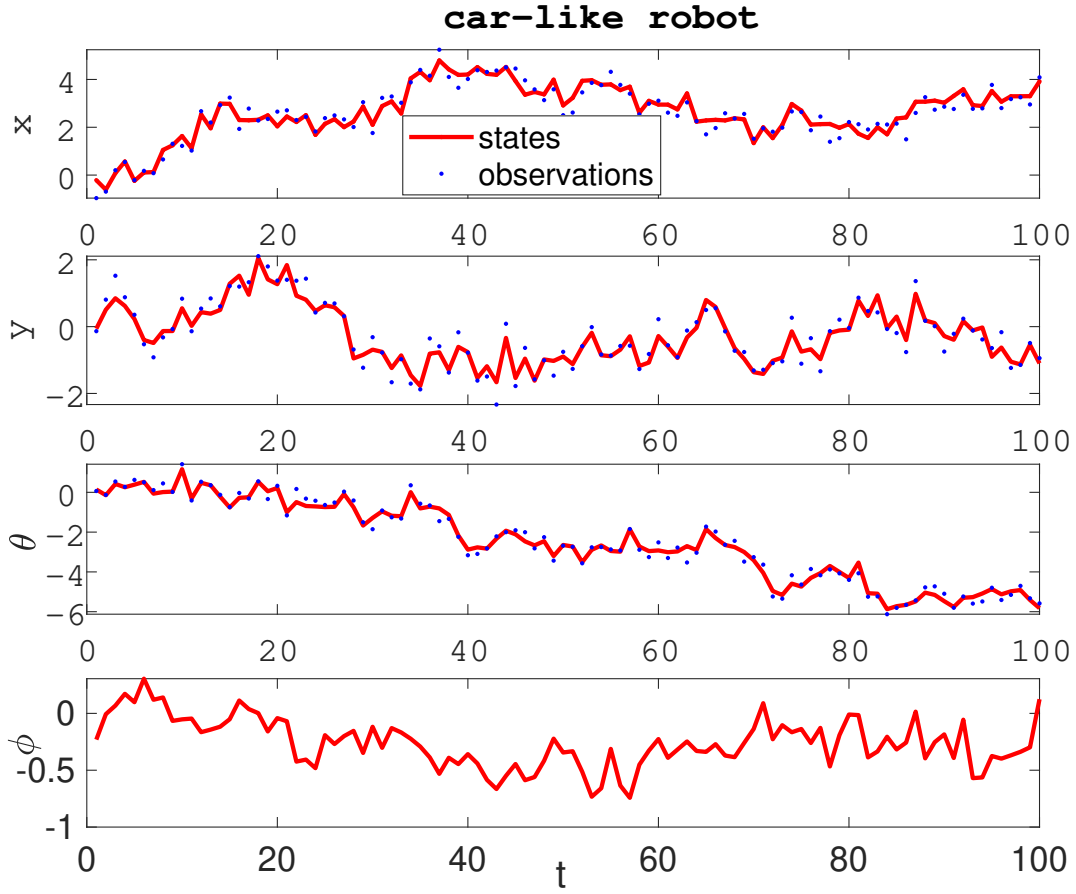


Figure 9. The true state (dashed lines) and the simulated observations (dots) of the car-like robot model.

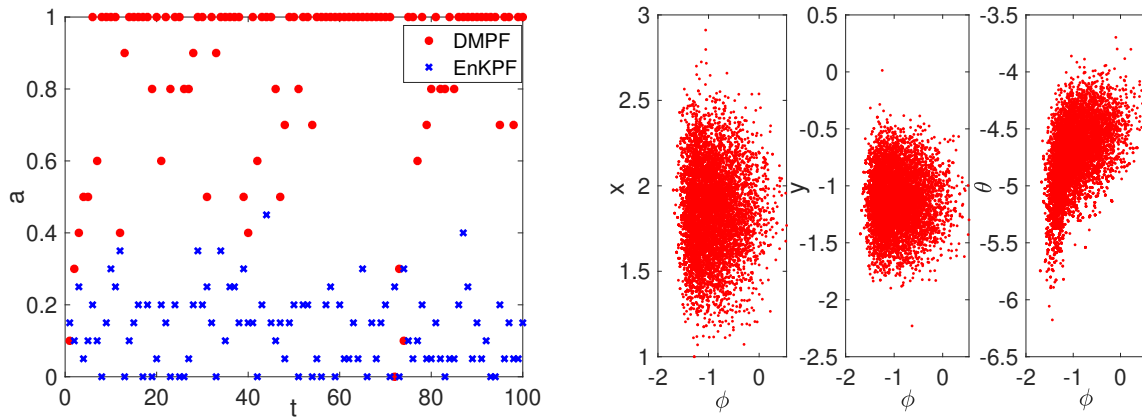


Figure 10. Left: the parameter a in DMPF and EnKPF for the car-like robot model. The scatter plots of the particles at $t=72$ in the DMPF method.

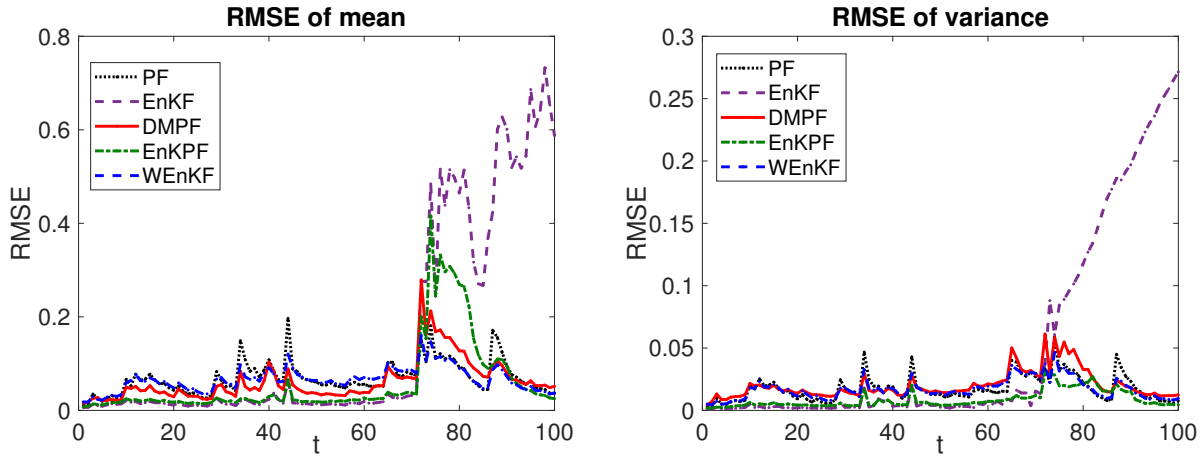


Figure 11. The RMSE of the posterior mean (left) and of the posterior variance (right) for the car-like robot model.

Table 2

RMSE averaged over all time steps in the car-like robot model.

| - | mean | variance |
|--------------|-------|----------|
| PF | 0.069 | 0.017 |
| EnKF | 0.15 | 0.052 |
| DMPF | 0.061 | 0.019 |
| EnKPF | 0.059 | 0.013 |
| WEnKF | 0.066 | 0.017 |

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 a proposal distribution in the marginal space by combining the standard PF and the EnKF-based proposals, which ensures that the algorithm performs well even when the posterior is strongly non-Gaussian. The proposed method can automatically adjust the relative weight of the PF and the EnKF components. We provide three examples to demonstrate the performance of the proposed method: in the first example we show that the DMPF method performs well when the standard EnKF fails due to the strong non-Gaussianity of the posterior distribution; the second example demonstrates that the DMPF method can significantly outperform PF when the posterior is close to Gaussian; finally it was illustrated by the third example that the defensive scheme used in our method can provide better results than simply using the EnKF as the IS distribution when a significant fraction of the time steps cannot be well approximated by Gaussian. The three examples demonstrate that the DMPF method has a good performance regardless of whether or not the posteriors are close to Gaussian. We believe that the method can be useful in a wide range of practical data assimilation problems.

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 challenging to accurately estimate

the IS weights in each step. This issue needs to be addressed so that the MPF type of methods can apply to high-dimensional problems. Second, as has been discussed in [18], 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 [18] that some approximation techniques such as the fast multipole method [14] 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. Third we reinstate it here that the proposed DMPF scheme does not require the IS distribution to be the EnKF approximation, and rather it can be used with any desired IS distribution. For example, one can design mixtures to approximate the marginal posteriors [3, 32, 5] and use them as the IS distribution in the defensive scheme. Finally, a difficulty in the particle-based methods is that the underlying dynamical system is often computationally intensive, and considerable efforts have been devoted to accelerating the computation, including surrogate models [22, 23], multilevel methods [16], and dimension reduction techniques [31], just to name a few. We expect that these techniques can be used to accelerate the DMPF algorithm as well. We hope to study these issues and improve the DMPF method in the future.

REFERENCES

- [1] A. APTE, M. HAIRER, A. STUART, AND J. VOSS, *Sampling the posterior: An approach to non-Gaussian data assimilation*, Phys. D, 230 (2007), pp. 50–64.
- [2] M. S. ARULAMPALAM, S. MASKELL, N. GORDON, AND T. CLAPP, *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*, IEEE Trans. Signal Process., 50 (2002), pp. 174–188.
- [3] T. BENGTTSSON, C. SNYDER, AND D. NYCHKA, *Toward a nonlinear ensemble filter for high-dimensional systems*, J. Geophys. Res. Atmos., 108 (2003), # 8775.
- [4] O. CAPPÉ, S. J. GODSILL, AND E. MOULINES, *An overview of existing methods and recent advances in sequential Monte Carlo*, Proc. IEEE, 95 (2007), pp. 899–924.
- [5] R. CHEN AND J. S. LIU, *Mixture Kalman filters*, J. Roy. Statist. Soc. Ser. B, 62 (2000), pp. 493–508.
- [6] N. CHUSTAGULPROM, S. REICH, AND M. REINHARDT, *A hybrid ensemble transform particle filter for nonlinear and spatially extended dynamical systems*, SIAM/ASA J. Uncertain. Quantif., 4 (2016), pp. 592–608.
- [7] J.-M. CORNUET, J.-M. MARIN, A. MIRA, AND C. P. ROBERT, *Adaptive multiple importance sampling*, Scand. J. Stat., 39 (2012), pp. 798–812.
- [8] A. DOUCET, N. DE FREITAS, AND N. GORDON, *An introduction to sequential Monte Carlo methods*, in Sequential Monte Carlo Methods in Practice, Springer, New York, 2001, pp. 3–14.
- [9] A. DOUCET, S. GODSILL, AND C. ANDRIEU, *On sequential Monte Carlo sampling methods for Bayesian filtering*, Statist. Comput., 10 (2000), pp. 197–208.
- [10] A. DOUCET AND A. M. JOHANSEN, *A tutorial on particle filtering and smoothing: Fifteen years later*, in Oxford Handbook of Nonlinear Filtering, Vol. 12, 2009, Chapter 3.
- [11] G. EVENSEN, *Data Assimilation: The Ensemble Kalman Filter*, Springer, Dordrecht, the Netherlands, 2009.
- [12] M. FREI AND H. R. KÜNSCH, *Bridging the ensemble Kalman and particle filters*, Biometrika, 100 (2013), pp. 781–800.
- [13] M. GHIL AND P. MALANOTTE-RIZZOLI, *Data assimilation in meteorology and oceanography*, in Advances in Geophysics, Vol. 33, Elsevier, Burlington, MA, 1991, pp. 141–266.
- [14] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
- [15] T. HESTERBERG, *Weighted average importance sampling and defensive mixture distributions*, Technometrics, 37 (1995), pp. 185–194.

- [16] A. JASRA, K. KAMATANI, K. J. H. LAW, AND Y. ZHOU, *Multilevel particle filters*, SIAM J. Numer. Anal., 55 (2017), pp. 3068–3096.
- [17] A. H. JAZWINSKI, *Stochastic Processes and Filtering Theory*, Courier Corporation, North Chelmsford, MA, 2007.
- [18] M. KLAAS, N. DE FREITAS, AND A. DOUCET, *Toward practical N2 Monte Carlo: The marginal particle filter*, in Uncertainty in Artificial Intelligence (UAI), AUAI Press, Corvallis, OR, 2005, pp. 308–315.
- [19] L. KUZNETSOV, K. IDE, AND C. JONES, *A method for assimilation of Lagrangian data*, Month. Weather Rev., 131 (2003), pp. 2247–2260.
- [20] K. LAW, A. STUART, AND K. ZYGALAKIS, *Data Assimilation*, Springer, Cham, Switzerland, 2015.
- [21] J. LEI AND P. J. BICKEL, *A moment matching ensemble filter for nonlinear non-Gaussian data assimilation*, Month. Weather Rev., 139 (2011), pp. 3964–3973.
- [22] J. LI AND D. XIU, *A generalized polynomial chaos based ensemble Kalman filter with high accuracy*, J. Comput. Phys., 228 (2009), pp. 5454–5469.
- [23] W. LI, G. LIN, AND D. ZHANG, *An adaptive ANOVA-based PCKF for high-dimensional nonlinear inverse modeling*, J. Comput. Phys., 258 (2014), pp. 752–772.
- [24] E. N. LORENZ, *Deterministic nonperiodic flow*, J. Atmos. Sci., 20 (1963), pp. 130–141.
- [25] R. MARTINEZ-CANTIN, N. DE FREITAS, AND J. A. CASTELLANOS, *Analysis of particle methods for simultaneous robot localization and mapping and a new algorithm: Marginal-SLAM*, in ICRA, IEEE, Piscataway, NJ, 2007, pp. 2415–2420.
- [26] M. MORZFELD, D. HODYSS, AND C. SNYDER, *What the collapse of the ensemble Kalman filter tells us about particle filters*, Tellus A, 69 (2017), 1283809.
- [27] A. OWEN AND Y. ZHOU, *Safe and effective importance sampling*, J. Amer. Statist. Assoc., 95 (2000), pp. 135–143.
- [28] N. PAPADAKIS, É. MÉMIN, A. CUZOL, AND N. GENGEMBRE, *Data assimilation with the weighted ensemble Kalman filter*, Tellus A, 62 (2010), pp. 673–697.
- [29] G. POYIADJIS, A. DOUCET, AND S. S. SINGH, *Particle methods for optimal filter derivative: Application to parameter estimation*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP’05), 2005, Vol. 5, IEEE, Piscataway, NJ, 2005, pp. 925–928.
- [30] H. SALMAN, L. KUZNETSOV, C. JONES, AND K. IDE, *A method for assimilating Lagrangian data into a shallow-water-equation ocean model*, Month. Weather Rev., 134 (2006), pp. 1081–1101.
- [31] A. SOLONEN, T. CUI, J. HAKKARAINEN, AND Y. MARZOUK, *On dimension reduction in Gaussian filters*, Inverse Problems, 32 (2016), 045003.
- [32] 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., 15 (2011), pp. 293–305.
- [33] S. THRUN, W. BURGARD, AND D. FOX, *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2005.
- [34] E. VEACH AND L. J. GUIBAS, *Optimally combining sampling techniques for Monte Carlo rendering*, in Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, ACM, 1995, pp. 419–428.