

# Ensemble Kalman filter based sequential Monte Carlo sampler for sequential Bayesian inference

Wu, Jiangqi; Wen, Linjie; Green, Peter L; Li, Jinglai; Maskell, Simon

DOI:

[10.1007/s11222-021-10075-x](https://doi.org/10.1007/s11222-021-10075-x)

License:

Creative Commons: Attribution (CC BY)

*Document Version*

Publisher's PDF, also known as Version of record

*Citation for published version (Harvard):*

Wu, J, Wen, L, Green, PL, Li, J & Maskell, S 2022, 'Ensemble Kalman filter based sequential Monte Carlo sampler for sequential Bayesian inference', *Statistics and Computing*, vol. 32, no. 1, 20.  
<https://doi.org/10.1007/s11222-021-10075-x>

[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](mailto:UBIRA@lists.bham.ac.uk) providing details and we will remove access to the work immediately and investigate.



# Ensemble Kalman filter based sequential Monte Carlo sampler for sequential Bayesian inference

Jiangqi Wu<sup>1</sup> · Linjie Wen<sup>2</sup> · Peter L. Green<sup>3</sup> · Jinglai Li<sup>4</sup> · Simon Maskell<sup>5</sup>

Received: 15 December 2020 / Accepted: 13 December 2021  
© The Author(s) 2022

## Abstract

Many real-world problems require one to estimate parameters of interest, in a Bayesian framework, from data that are collected sequentially in time. Conventional methods for sampling from posterior distributions, such as Markov chain Monte Carlo cannot efficiently address such problems as they do not take advantage of the data's sequential structure. To this end, sequential methods which seek to update the posterior distribution whenever a new collection of data become available are often used to solve these types of problems. Two popular choices of sequential method are the ensemble Kalman filter (EnKF) and the sequential Monte Carlo sampler (SMCS). While EnKF only computes a Gaussian approximation of the posterior distribution, SMCS can draw samples directly from the posterior. Its performance, however, depends critically upon the kernels that are used. In this work, we present a method that constructs the kernels of SMCS using an EnKF formulation, and we demonstrate the performance of the method with numerical examples.

**Keywords** Ensemble Kalman filter · Parameter estimation · Sequential Bayesian inference · Sequential Monte Carlo sampler

**Mathematics Subject Classification** 62F15 · 65C05

## 1 Introduction

Bayesian inference is a popular method for estimating unknown parameters from data, largely due to its ability to

quantify uncertainty in the estimation results (Gelman 2013). In the current work we consider a special class of Bayesian inference problems where data have to be collected in a sequential manner. A typical example of this type of problems is the estimation of parameters, such as the initial states or the equation coefficients, in a dynamical system from observations related to the state vector at discrete times. Such problems arise from many real-world applications, ranging from weather prediction (Annan and Hargreaves 2004) to biochemical networks (Golightly and Wilkinson 2011). It should be emphasized that, unlike many data assimilation problems that seek to estimate the time-dependent states in dynamical systems, the parameters that we want to estimate here are assumed not to vary in time. To distinguish the two types of problems, we refer to the former as *state estimation* problems and the latter as *parameter estimation*. We should also note that in this work we focus on methods which use samples to represent the posterior distribution, and the approximation-based methods, such as the variational Bayes (Beal 2003) and the expectation propagation (Minka 2001) will not be discussed here. Conventional sampling methods, such as Markov chain Monte Carlo (MCMC) simulations (Gilks et al. 1995), use all the data in a single batch, are

---

This work was supported by the NSFC under Grant Number 111771289 and by the EPSRC under Grant Number EP/R018537/1.

---

✉ Jinglai Li  
j.li.10@bham.ac.uk  
Peter L. Green  
plgreen@liverpool.ac.uk  
Simon Maskell  
S.Maskell@liverpool.ac.uk

- <sup>1</sup> Equipment Reliability Institute, Nuclear Power Operations Research Institute, 50 Shiboguan Rd, Shanghai 200120, China
- <sup>2</sup> School of Mathematical Sciences, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai 200240, China
- <sup>3</sup> School of Engineering, University of Liverpool, Liverpool L69 7XL, UK
- <sup>4</sup> School of Mathematics, University of Birmingham, Birmingham B15 2TT, UK
- <sup>5</sup> Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 7XL, UK

unable to take advantage of the sequential structure of the problems. On the other hand, the sequential methods utilize the sequential structure of the problem and update the posterior whenever a new collection of data become available, which makes them particularly convenient and efficient for sequential inference problems.

A popular sequential method for parameter estimation is the ensemble Kalman filter (EnKF), which was initially developed to address the dynamical state estimation problems (Evensen 2009). The EnKF method was extended to estimate parameters in many practical problems, e.g., Annan and Hargreaves (2004), Annan et al. (2005), and more recently, it was generically formulated as a derivative-free optimization based parameter estimation method in Iglesias et al. (2013). EnKF for parameter estimation was further developed and analyzed in Arnold et al. (2014), Iglesias (2016), Schillings and Stuart (2017), etc. The basic idea of EnKF for parameter estimation is to construct an artificial dynamical system, turning the parameters of interest into the states of the constructed dynamical system, before applying the standard EnKF procedure to estimate the states of the system. A major limitation of the EnKF method is that, just like the original version for dynamical state estimation, it can only compute a Gaussian approximation of the posterior distribution, and thus methods directly sampling the posterior distribution are certainly desirable. To this end, the sequential Monte Carlo sampler (SMCS) method (Del Moral et al. 2006), can draw samples directly from the posterior distribution. The SMCS algorithm is a generalization of the particle filter (Sanjeev Arulampalam et al. 2002; Doucet and Johansen 2009) for dynamic state estimation, generating weighted samples from the posterior distribution. Since the SMCS algorithm was proposed in Del Moral et al. (2006), considerable improvements and extensions of the method have been proposed, such as Fearnhead and Taylor (2013), Beskos et al. (2017), Heng et al. (2020), Everitt et al. (2020), and more information on the developments of the SMCS methods can be found in the recent reviews (Dai et al. 2020; Chopin and Papaspiliopoulos 2020). We also note here that there are other parameter estimation schemes also based on particle filtering, e.g., Gilks and Berzuini (2001), Chopin (2002), and the differences and connections between SMCS and these schemes are discussed in Del Moral et al. (2006). As will be discussed later, a key issue in the implementation of SMCS is to choose suitable forward and backward kernels, as the performance of SMCS depends critically on such choices. As has been shown in Del Moral et al. (2006), the optimal forward and backward kernels exist in principle, but designing effective kernels for specific problems is nevertheless a highly challenging task. In dynamic state estimation problems, often the EnKF approximation is used as the proposal distribution in the particle filtering algorithm (Papadakis et al. 2010; Wen et al. 2020), especially for prob-

lems in which the posteriors are only modestly non-Gaussian. Building upon similar ideas, we propose in this work to construct the kernels in SMCS by using an EnKF framework. Specifically, the forward kernel is obtained directly from an EnKF approximation, and the backward kernel is derived by making a Gaussian approximation of the optimal backward kernel.

The remaining work is organized as follows. In Sect. 2, we present the generic setup of the sequential inference problems that we consider in this work. In Sects. 3 and 4, we, respectively, review the SMCS and the EnKF methods for solving sequential inference problems. In Sect. 5, we present the proposed EnKF-SMCS method, and in Sect. 6, we provide several numerical examples to illustrate the performance of the proposed method. Finally Sect. 7 offers some concluding remarks.

## 2 Problem setup

We consider a sequential inference problem formulated as follows. Suppose that we want to estimate the parameter  $x \in \mathbb{R}^{n_x}$  from data  $y_1, \dots, y_t, \dots, y_T$  which become available sequentially in time. In particular the data  $y_t \in \mathbb{R}^{n_y}$  is related to the parameter of interest  $x$  via the follow model,

$$y_t = G_t(x) + \eta_t, \quad t = 1 \dots T,$$

where each  $G_t(\cdot)$  is a mapping from  $\mathbb{R}^{n_x}$  to  $\mathbb{R}^{n_y}$ , and the observation noise  $\eta_t \sim \mathcal{N}(0, R_t)$ . It follows that the likelihood function can be written as,

$$\pi(y_t|x) = \mathcal{N}(G_t(x), R_t), \quad t = 1 \dots T. \quad (2.1)$$

It is important to note here that the restriction that the error model has to be additive Gaussian as is in Eq. (2.1) is due to the use of EnKF. While noting that relaxing such a restriction is possible, we emphasize here that additive Gaussian assumption noise is reasonable for a wide range of practical problems. We can now write the posterior distribution in a sequential form:

$$\pi_t(x) = \pi(x|y_1, \dots, y_t) \propto \pi_0(x) \prod_{i=1}^t \pi(y_i|x), \quad (2.2)$$

where  $\pi_0(x)$  is the prior distribution of  $x$ , and our goal is to draw samples from  $\pi_t$  for any  $0 < t \leq T$ .

The posterior in Eq. (2.2) is essential in a data tempering formulation, and as is pointed out in Fearnhead and Taylor (2013), Zhou et al. (2016), such problems pose challenges for usual MCMC methods, especially when the amount of data is large, as they cannot conveniently exploit the sequential structure of the problem. In what follows, we first discuss

two sequential methods for this type of problems: the EnKF and the SMCS algorithms, and we then propose a scheme to combine these two methods.

### 3 Sequential Monte Carlo sampler

We first give a brief introduction to the SMCS method for sampling the posterior distribution  $\pi_t(x)$ , following (Del Moral et al. 2006). The key idea of SMCS is to construct a joint distribution  $\pi(x_1, \dots, x_t)$ , the marginal of which is equal to the target distribution  $\pi_t(\cdot)$ . Note here that  $\pi(x_1, \dots, x_t)$  needs only to be known up to a normalization constant. One then applies the sequential importance sampling algorithm (Sanjeev Arulampalam et al. 2002; Doucet and Johansen 2009) to draw weighted samples from  $\pi(x_1, \dots, x_t)$ , which after being marginalized over  $x_1, \dots, x_{t-1}$ , yields samples from  $\pi_t(\cdot)$ .

Next we describe SMCS in a recursive formulation where, given an arbitrary conditional distribution  $L_{t-1}(x_{t-1}|x_t)$ , we can construct a joint distribution of  $x_{t-1}$  and  $x_t$  in the form of,

$$p_t(x_{t-1}, x_t) = \pi_t(x_t)L_{t-1}(x_{t-1}|x_t). \tag{3.1}$$

such that the marginal distribution of  $p_t(x_{t-1}, x_t)$  over  $x_{t-1}$  is  $\pi_t(x_t)$ . Now, given a marginal distribution  $q_{t-1}(x_{t-1})$  and a conditional distribution  $K_t(x_t|x_{t-1})$ , we can construct an importance sampling (IS) distribution for  $p_t(x_{t-1}, x_t)$  in the form of

$$q_t(x_{t-1}, x_t) = q_{t-1}(x_{t-1})K_t(x_t|x_{t-1}). \tag{3.2}$$

It is important to note here that a key requirement of the IS distribution  $q_t(x_{t-1}, x_t)$  is that we can directly draw samples from it. We let  $\{x_{t-1:t}^m\}_{m=1}^M$  be an ensemble drawn from  $q_t(x_{t-1}, x_t)$ , and note that the weighted ensemble  $\{(x_{t-1:t}^m, w_t^m)\}_{m=1}^M$  follows the distribution  $p_t(x_{t-1:t})$ , where the weights are computed according to

$$\begin{aligned} w_t(x_{t-1:t}) &= \frac{p_t(x_{t-1}, x_t)}{q_t(x_{t-1}, x_t)} = \frac{\pi_t(x_t)L_{t-1}(x_{t-1}|x_t)}{q_{t-1}(x_{t-1})K_t(x_t|x_{t-1})} \\ &= w_{t-1}(x_{t-1})\alpha_t(x_{t-1}, x_t), \end{aligned} \tag{3.3a}$$

where

$$\begin{aligned} w_{t-1}(x_{t-1}) &= \frac{\pi_{t-1}(x_{t-1})}{q_{t-1}(x_{t-1})}, \\ \alpha_t(x_{t-1}, x_t) &= \frac{\pi_t(x_t)L_{t-1}(x_{t-1}|x_t)}{\pi_{t-1}(x_{t-1})K_t(x_t|x_{t-1})}. \end{aligned} \tag{3.3b}$$

As can be seen here, once the two conditional distributions  $K_t$  and  $L_{t-1}$  (respectively, referred to as the forward and backward kernels in the rest of the paper) are chosen, we can draw

samples from Eq. (3.2) and compute the associated weights from Eq. (3.3), obtaining weighted samples from  $p_t(x_{t-1}, x_t)$  as well as its marginal  $\pi_t(x_t)$ . The SMCS essentially conducts this procedure in the following sequential manner:

1. let  $t = 0$ , draw an ensemble  $\{x_0^m\}_{m=1}^M$  from  $q_0(x_0)$ , and compute  $w_0^m = \pi_0(x_0^m)/q_0(x_0^m)$  for  $m = 1 \dots M$ ;
2. let  $t = t + 1$ ;
3. draw  $x_t^m$  from  $K(\cdot|x_{t-1}^m)$  for each  $m = 1 \dots M$ ;
4. compute  $w_t^m$  using Eq. (3.3);
5. return to step 2 if  $t < T$ .

Note here that, a resampling step is often used in SMCS algorithms to alleviate the ‘‘sample degeneracy’’ issue (Del Moral et al. 2006). The resampling techniques are well documented in the particle filter literature, e.g., Sanjeev Arulampalam et al. (2002), Doucet and Johansen (2009), and so are not discussed here.

As can be seen from the discussion above, to use SMCS one must choose the two kernels. In principle, optimal choices of these kernels are available. For example, it is known that once  $K_t(x_t|x_{t-1})$  is provided, one can derive that the optimal choice of  $L_{t-1}(x_{t-1}|x_t)$  is Del Moral et al. (2006):

$$\begin{aligned} L_{t-1}^{\text{opt}}(x_{t-1}|x_t) &= \frac{q_{t-1}(x_{t-1})K_t(x_t|x_{t-1})}{q_t(x_t)} \\ &= \frac{q_{t-1}(x_{t-1})K_t(x_t|x_{t-1})}{\int q_{t-1}(x_{t-1})K_t(x_t|x_{t-1})dx_{t-1}}, \end{aligned} \tag{3.4}$$

where the optimality is in the sense of yielding the minimal estimator variance. We also note that use of the optimal L-kernel allows the weights to be written as

$$w_t(x_{t-1:t}) = \frac{\pi_t(x_t)}{q_t(x_t)}. \tag{3.5}$$

Moreover, we can see here that if we can choose  $K_t$  such that  $q_t = \pi_t$ , then the weight function is always unity, which means that we now sample directly from the target distribution (the ideal case). While obtaining such an ideal  $K_t$  is usually not possible in practice, it nevertheless provides useful guideline regarding choice of the forward kernel  $K_t$ , i.e., it should be chosen such that the resulting  $q_t$  is close to  $\pi_t$ . For example, it is proposed in Del Moral et al. (2006) to use the MCMC moves as the forward kernel. A main limitation of the MCMC kernel is that it typically requires a number of MCMC moves to propose a ‘‘good’’ particle, and since each MCMC move involves an evaluation of the underlying mathematical model,  $G_t$ , the total computational cost can be high when  $G_t$  is computationally intensive.

In this work, we consider an alternative to the use of MCMC kernels. Specifically we propose to choose  $K_t$  of

the form

$$K_t(\cdot|x_{t-1}) = \mathcal{N}(\cdot|T_t(x_{t-1}), \Sigma_t^K), \tag{3.6}$$

i.e., a Gaussian distribution with mean  $T_t(x_{t-1})$  and variance  $\Sigma_t^K$ , where  $T_t(\cdot)$  is a  $\mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x}$  transformation. We shall compute  $T_t$  and  $\Sigma_t^K$  (or equivalently the forward kernel  $K_t$ ) using the EnKF method.

### 4 Ensemble Kalman filter

In this section, we give a brief overview of the EnKF parameter estimation method proposed in Iglesias et al. (2013), which essentially aims to compute a Gaussian approximation of  $\pi_t(x_t)$  in each time step  $t$ . To formulate the problem in an EnKF framework, we first construct an *artificial dynamical system* denoted by  $F_t$ ; at any time  $t$ , we have the states  $u_t = [x_t, z_t]^T$  where  $z_t = G_t(x_t)$ , and the dynamical model,

$$u_t = F_t(u_{t-1}), \quad x_t = x_{t-1}, \quad z_t = G_t(x_t). \tag{4.1}$$

The data is associated with the states through  $y_t = z_t + \eta_t$ , or equivalently

$$y_t = Hu_t + \eta_t = [0_{n_y \times n_x}, I_{n_y \times n_y}]u_t + \eta_t,$$

where  $I_{n_y \times n_y}$  is a  $n_y \times n_y$  identity matrix and  $0_{n_y \times n_x}$  is a  $n_y \times n_x$  zero matrix. We emphasize here that once we have the posterior distribution  $\pi(u_t|y_{1:t})$ , we can obtain the posterior  $\pi_t(x_t) = \pi(x_t|y_{1:t})$  by marginalizing  $\pi(u_t|y_{1:t})$  over  $z_t$ .

Now let us see how the EnKF proceeds to compute a Gaussian approximation of the posterior distribution  $\pi(u_t|y_{1:t})$ . At time  $t$ , suppose that the prior  $\pi(u_t|y_{1:t-1})$  can be approximated by a Gaussian distribution with mean  $\tilde{\mu}_t$  and covariance  $\tilde{C}_t$ . It follows that the posterior distribution  $\pi(u_t|y_{1:t})$  is also Gaussian and its mean and covariance can be obtained analytically:

$$\mu_t = \tilde{\mu}_t + Q_t(y_t - H\tilde{\mu}_t), \quad C_t = (I - Q_tH)\tilde{C}_t, \tag{4.2}$$

where  $I$  is the identity matrix and

$$Q_t = \tilde{C}_t H^T (H\tilde{C}_t H^T + R_t)^{-1} \tag{4.3}$$

is the so-called Kalman gain matrix.

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. Suppose that at time  $t - 1$ , we have an ensemble of particles  $\{u_{t-1}^m\}_{m=1}^M$  drawn according to the posterior distribution  $\pi(u_{t-1}|y_{1:t-1})$ , we can propagate the particles via the dynamical model (4.1):

$$\tilde{u}_t^m = F_t(u_{t-1}^m), \tag{4.4}$$

for  $m = 1 \dots M$ , obtaining an assemble  $\{\tilde{u}_t^m\}_{m=1}^M$  following the prior  $\pi(u_t|y_{1:t-1})$ . We can compute a Gaussian approximation,  $\mathcal{N}(u_t|\tilde{\mu}_t, \tilde{C}_t)$ , of  $\pi(u_t|y_{1:t-1})$ , where the mean and the covariance of  $\pi(u_t|y_{1:t-1})$  are estimated from the samples:

$$\tilde{\mu}_t = \frac{1}{M} \sum_{m=1}^M \tilde{u}_t^m, \quad \tilde{C}_t = \frac{1}{M-1} \sum_{m=1}^M (\tilde{u}_t^m - \tilde{\mu}_t)(\tilde{u}_t^m - \tilde{\mu}_t)^T. \tag{4.5}$$

Once  $\tilde{\mu}_t$  and  $\tilde{C}_t$  are obtained, we then can compute  $\mu_t$  and  $C_t$  directly from Eq. (4.2), and by design, the posterior distribution  $\pi(u_t|y_{1:t})$  is approximated by  $\mathcal{N}(\mu_t, C_t)$ . Moreover it can be verified that the samples

$$u_t^m = \tilde{u}_t^m + Q_t(y_t - (H\tilde{u}_t^m - \eta_t^m)), \quad \eta_t^m \sim \mathcal{N}(0, R_t), \quad m = 1 \dots M, \tag{4.6}$$

with  $Q_t$  computed by Eq. (4.3), follow the distribution  $\mathcal{N}(\mu_t, C_t)$ . That is,  $\{u_t^m\}_{m=1}^M$  are the approximate ensemble of  $\pi(u_t|y_{1:t})$ , and consequently the associated  $\{x_t^m\}_{m=1}^M$  approximately follows distribution  $\pi_t(x_t) = \pi(x_t|y_{1:t})$ .

### 5 EnKF-SMCS

Now we shall discuss how to use the EnKF scheme to construct the forward kernel  $K_t$  for SMCS. First recall that  $u_t = [x_t, z_t]^T$ ,  $H = [0_{n_y \times n_x}, I_{n_y \times n_y}]$  and the propagation model  $x_t = x_{t-1}$ , and we can derive from Eq. (4.6) that,

$$x_t = x_{t-1} + Q_t^x(y_t - G_t(x_{t-1})) + Q_t^x \eta_t + \eta_t', \tag{5.1a}$$

$$\eta_t \sim \mathcal{N}(0, R_t), \quad \eta_t' \sim \mathcal{N}(0, \delta^2 \Sigma_{t-1}^q), \tag{5.1b}$$

where  $\delta$  is a small constant,  $\Sigma_{t-1}^q$  is the covariance of  $q_{t-1}$  (the evaluation of  $\Sigma_{t-1}^q$  is provided in Eq. (5.5b)), and  $Q_t^x$  is a submatrix of  $Q_t$  formed taking the first  $n_x$  rows and the first  $n_y$  columns of  $Q_t$ , denoted as  $Q_t^x = Q_t[1 : n_x, 1 : n_y]$ . Eq. (5.1a) can also be written as a conditional distribution:

$$K_t(\cdot|x_{t-1}) = \mathcal{N}(\cdot|T_t(x_{t-1}), \Sigma_t^K), \tag{5.2a}$$

where

$$T_t(x_{t-1}) = x_{t-1} + Q_t^x(y_t - G_t(x_{t-1})) \quad \text{and} \quad \Sigma_t^K = Q_t^x R_t (Q_t^x)^T + \delta^2 \Sigma_{t-1}^q. \tag{5.2b}$$

Note that the purpose of introducing the small noise term,  $\eta_t'$ , in Eq. (5.1a) is to ensure that  $\Sigma_t^K$  is strictly positive definite and so  $K_t$  is a valid Gaussian conditional distribution. In all

the numerical implementations performed in this work,  $\delta$  is set to be  $10^{-4}$ . According to the discussion in Sect. 4, we have, if  $q_{t-1}$  is a good approximation to  $\pi_{t-1}$

$$q_t(x_t) = \int K_t(x_t|x_{t-1})q_{t-1}(x_{t-1})dx_{t-1} \approx \pi_t(x_t). \tag{5.3}$$

That is, Eq. (5.2) provides a good forward Kernel for the SMC sampler. It should be noted that since  $T_t$  is a nonlinear transform, in general we cannot derive the analytical expression for  $q_t$ , and as a result, we cannot use the optimal backward kernel given in Eq. (3.4). Nonetheless, we can use a sub-optimal backward kernel:

$$\hat{L}_{t-1}(x_{t-1}|x_t) = \frac{\hat{q}_{t-1}(x_{t-1})\hat{K}_t(x_t|x_{t-1})}{\int \hat{q}_{t-1}(x_{t-1})\hat{K}_t(x_t|x_{t-1})dx_{t-1}}, \tag{5.4}$$

where  $\hat{q}_{t-1}$  is the Gaussian approximation of  $q_{t-1}$  and  $\hat{K}_t$  is an approximation of  $K_t$ . Next we need to determine  $\hat{q}_{t-1}$  and  $\hat{K}_t$ . Here  $\hat{q}_{t-1}$  can be estimated from the ensemble  $\{x_{t-1}^m\}_{m=1}^M$ :

$$\hat{q}_{t-1}(\cdot) = \mathcal{N}(\cdot|\xi_{t-1}, \Sigma_{t-1}^q), \tag{5.5a}$$

$$\xi_{t-1} = \frac{1}{M} \sum_{m=1}^M x_{t-1}^m,$$

$$\Sigma_{t-1}^q = \frac{1}{M-1} \sum_{m=1}^M (x_{t-1}^m - \xi_{t-1})(x_{t-1}^m - \xi_{t-1})^T. \tag{5.5b}$$

Now recall that the issue with the optimal backward kernel  $L_{t-1}^{opt}$  is that the transform  $T_t$  inside the forward kernel  $K_t$  is nonlinear, and as a result,  $q_t$  cannot be computed analytically. Here to obtain  $\hat{L}_{t-1}$  in Eq. (5.4) explicitly, we take

$$\begin{aligned} \hat{K}_t(\cdot|x_{t-1}) &= \mathcal{N}(\cdot|x_{t-1} + Q_t^x(y_t - \bar{y}_t), \Sigma_t^K), \quad \text{with} \\ \bar{y}_t &= \mathbb{E}_{x_{t-1}|y_{1:t-1}}[G_t(x_{t-1})], \end{aligned} \tag{5.6}$$

and in practice  $\bar{y}_t$  is evaluated from the particles, i.e.,

$$\bar{y}_t \approx \frac{1}{M} \sum_{m=1}^M G_t(x_{t-1}^m). \tag{5.7}$$

It follows that the backward kernel  $\hat{L}_{t-1}$ , given by Eq. (5.4), is also Gaussian and is given by

$$\hat{L}_{t-1}(\cdot|x_t) = \mathcal{N}(\cdot|T_{t-1}^L(x_t), \Sigma_{t-1}^L), \tag{5.8a}$$

where

$$\begin{aligned} T_{t-1}^L(x_t) &= (I - \Sigma_t^K(\Sigma_t^K + \Sigma_{t-1}^q)^{-1})(x_t - Q_t^x(y_t - \bar{y}_t)) \\ &\quad + (I - \Sigma_{t-1}^q(\Sigma_{t-1}^q + \Sigma_t^K)^{-1})\xi_{t-1}, \end{aligned} \tag{5.8b}$$

and

$$\Sigma_t^L = \Sigma_{t-1}^q - \Sigma_{t-1}^q(\Sigma_{t-1}^q + \Sigma_t^K)^{-1}\Sigma_{t-1}^q. \tag{5.8c}$$

It follows that the resulting incremental weight function is

$$\alpha_t(x_{t-1}, x_t) = \frac{\pi_t(x_t)\hat{L}_{t-1}(x_{t-1}|x_t)}{\pi_{t-1}(x_{t-1})K_t(x_t|x_{t-1})}. \tag{5.9}$$

Now using the ingredients presented above, we summarize the EnKF-SMCS scheme in Algorithm 1.

---

**Algorithm 1** The EnKF-SMCS algorithm

---

Initialization: draw sample  $\{x_0^m\}_{m=1}^M$  from distribution  $q_0(x_0)$ ; compute the weights  $w_0^m = \pi_0(x_0^m)/q_0(x_0^m)$  for  $m = 1 \dots M$  and renormalize  $\{w_0^m\}_{m=1}^M$  so that  $\sum_{m=1}^M w_0^m = 1$ ;

**for**  $t = 1$  **to**  $T$  **do**

estimate  $\xi_{t-1}$  and  $\Sigma_{t-1}^q$  from the ensemble  $\{(x_{t-1}^m, w_{t-1}^m)\}_{m=1}^M$  using Eq. (5.5b);  
 let  $\tilde{u}_t^m = [x_{t-1}^m, G_t(x_{t-1}^m)]^T$  for  $m = 1 \dots M$ ;  
 evaluate  $\tilde{\mu}_t$  and  $\tilde{\Sigma}_t$  with Eq. (4.5), and compute  $Q_t$  with Eq. (4.3);  
 draw  $x_t^m \sim K_t(x_t|x_{t-1}^m)$  for  $m = 1 \dots M$  with  $K_t$  given by Eq. (5.2);  
 compute  $\hat{L}_{t-1}$  from Eq. (5.8);  
 update the weights:

$$w_t^m = w_{t-1}^m \frac{\pi_t(x_t^m)\hat{L}_{t-1}(x_{t-1}^m|x_t^m)}{\pi_{t-1}(x_{t-1}^m)K_t(x_t^m|x_{t-1}^m)}$$

and renormalize  $\{w_t^m\}_{m=1}^M$  so that  $\sum_{m=1}^M w_t^m = 1$ ;  
 resample if needed.

**end**

---

It is important to note that a key challenge is yet to be addressed in Algorithm 1, namely the cost for computing the particle weight. First recall that the main computational cost arises from the evaluation of the forward model  $G_t$ , and therefore, the total computational cost can be approximately measured by the number of evaluations of  $G_t$ . We can see from Eq. (5.9) that when updating the particle weight, we need to compute  $\pi_t(x_t)$ , which involves the evaluation of the forward model from  $G_1$  to  $G_t$ . This operation is required at each time step, and as a result the number of the model evaluations is at the order of  $O(T^2)$  for each particle. Therefore, the total computational cost can be prohibitive if  $T$  is large. We propose a method to tackle the issue, which is based on the following two observations. First, here we mainly consider the sequential inference problems where one is primarily interested in the posterior distribution at the final step where all data are incorporated. Second, in many practical problems, after some number of observations, the posteriors may not vary substantially in several consecutive steps. It therefore may not be necessary to *exactly* compute the posterior distribution at each time step, and as a result, we only need to sample the posterior distribution in a relatively small

number of selected steps. Based on this idea, we propose the following scheme in each time step to reduce the computational cost: We first compute an approximate weight for each particle and then assess that if some prescribed conditions (based on the approximate weights) are satisfied. If such conditions are satisfied, we evaluate the actual weights of the particles. To implement this scheme, we have to address the following issues:

- First we need a method to compute the approximate weight, which should be much easier to compute than the exact weight. Recall that in Eq. (5.9) one has to evaluate  $\pi_t(x_t)/\pi_{t-1}(x_{t-1})$  which involves computing the forward models from  $G_1(x_t)$  all the way to  $G_t(-x_t)$ , and so the computational cost is high. To reduce the computational cost, we propose the following approximate method to evaluate Eq. (5.9). Namely we first write  $\pi_t(x_t)/\pi_{t-1}(x_{t-1})$  as,

$$\frac{\pi_t(x_t)}{\pi_{t-1}(x_{t-1})} = \frac{\pi_{t-1}(x_t)}{\pi_{t-1}(x_{t-1})} \pi(y_t|x_t),$$

and naturally we can approximate  $\pi_{t-1}$  with  $q_{t-1}$ , yielding,

$$\frac{\pi_t(x_t)}{\pi_{t-1}(x_{t-1})} \approx \frac{q_{t-1}(x_t)}{q_{t-1}(x_{t-1})} \pi(y_t|x_t).$$

Though  $q_{t-1}$  is formally given by Eq. (5.3), it is not computationally tractable. Thus we make another approximation, replacing  $q_{t-1}$  with  $\hat{q}_{t-1}$ , where  $\hat{q}_{t-1}$  is the Gaussian approximation of  $q_{t-1}$  given by Eqs. (5.5), and as a result, we obtain

$$\alpha_t(x_{t-1}, x_t) \approx \frac{\hat{q}_{t-1}(x_t)\pi(y_t|x_t)\hat{L}_{t-1}(x_{t-1}|x_t)}{\hat{q}_{t-1}(x_{t-1})K_t(x_t|x_{t-1})}, \tag{5.10}$$

which is used to compute the approximate weights.

- Second we need to prescribe the conditions for triggering the computation of the actual weights. Following Green and Maskell (2017), we use the effective sample size (ESS) (Doucet and Johansen 2009) (based on the approximate weights) as the main indicator for computing the actual weights. Namely if the ESS calculated with the approximate weights is smaller than a threshold value, the actual weights are computed. Moreover we also have two additional conditions that can also trigger the computation of the actual weights: 1) if the actual weights have not been computed for a given number of steps; 2) if the inference reaches the final step, i.e.,  $t = T$ . We refer to such a step as a weight refinement.

- Finally we shall discuss how to compute the actual weight  $w_t$ . It should be noted here that the recursive formulas (3.3) cannot be used here since the actual value of  $w_{t-1}$  is not available. However, let  $t_0$  be the preceding step where the actual weights are computed, and it can be shown that

$$w_t = w_{t_0} \frac{\pi_t(x_t)}{\pi_{t_0}(x_{t_0})} \prod_{i=t_0}^{t-1} \frac{\hat{L}_i(x_i|x_{i+1})}{K_{i+1}(x_{i+1}|x_i)}, \tag{5.11}$$

which is used to calculate the actual weights of the particles.

We refer to this modified scheme as EnKF-SMCS with weight refinement (EnKF-SMCS-WR), the complete procedure of which is described in Algorithm 2. Note here that in both EnKF-SMCS algorithms, a resampling step is needed. Finally we can see that, in EnKF-SMCS-WR, the number of forward model evaluations can potentially be significantly reduced, and the actual number of evaluations depends on how frequently the weight refinement is triggered.

---

**Algorithm 2** The EnKF-SMCS-WR algorithm

---

Initialization: draw sample  $\{x_0^m\}_{m=1}^M$  from distribution  $q_0(x_0)$ ; compute the weights  $w_0^m = \pi_0(x_0^m)/q_0(x_0^m)$  for  $m = 1 \dots M$  and renormalize  $\{w_0^m\}_{m=1}^M$  so that  $\sum_{m=1}^M w_0^m = 1$ ;  
let  $t_0 = 0$ ;

**for**  $t = 1$  **to**  $T$  **do**

estimate  $\xi_{t-1}$  and  $\zeta_{t-1}^q$  from the ensemble  $\{(x_{t-1}^m, w_{t-1}^m)\}_{m=1}^M$  using Eq. (5.5b);  
let  $\tilde{u}_t^m = [x_{t-1}^m, G_t(x_{t-1}^m)]^T$  for  $m = 1 \dots M$ ;  
evaluate  $\tilde{\mu}_t$  and  $\tilde{C}_t$  with Eq. (4.5), and compute  $Q_t$  with Eq. (4.3);  
draw  $x_t^m \sim K_t(x_t|x_{t-1}^m)$  for  $m = 1 \dots M$  with  $K_t$  given by Eq. (5.2);  
calculate the approximate weights for  $m = 1 \dots M$ :

$$w_t^m = w_{t-1}^m \alpha_t^m, \quad \alpha_t^m = \frac{\hat{q}_{t-1}(x_t^m)\pi(y_t|x_t^m)\hat{L}_{t-1}(x_{t-1}^m|x_t^m)}{\hat{q}_{t-1}(x_{t-1}^m)K_t(x_t^m|x_{t-1}^m)},$$

and renormalize  $\{w_t^m\}_{m=1}^M$  so that  $\sum_{m=1}^M w_t^m = 1$ ;  
calculate the ESS of the approximate weights  $\{w_t^m\}_{m=1}^M$ ;

**if**  $ESS < ESS_{\min} \vee t - t_0 > \Delta T_{\max} \vee t = T$  **then**

compute  $\hat{L}_{t-1}$  from Eq. (5.8);  
calculate the weights for  $m = 1 \dots M$ :

$$w_t^m = w_{t_0}^m \frac{\pi_t(x_t^m)}{\pi_{t_0}(x_{t_0}^m)} \prod_{i=t_0}^{t-1} \frac{\hat{L}_i(x_i^m|x_{i+1}^m)}{K_{i+1}(x_{i+1}^m|x_i^m)},$$

and renormalize  $\{w_t^m\}_{m=1}^M$  so that  $\sum_{m=1}^M w_t^m = 1$ ;  
calculate the ESS of the weights  $\{w_t^m\}_{m=1}^M$ ;

**if**  $ESS < ESS_{\text{resamp}}$  **then**

| resample;

**end**

let  $t_0 = t$ ;

**end**

**end**

---

## 6 Numerical examples

We provide three examples in this section to demonstrate the performance of the proposed method. We emphasize that in these examples, the forward model  $G_t$  is computationally intensive and thus the main computational cost arises from the simulation of  $G_t$ . As a result, the main computational cost of all methods is measured by the number of forward model evaluations, which in all the methods used is equal to the product of the number of steps and that of the particles.

### 6.1 The Bernoulli model

Our first example is the Bernoulli equation,

$$\frac{dv}{d\tau} - v = -v^3, \quad v(0) = x, \tag{6.1a}$$

which has an analytical solution,

$$v(\tau) = G(x, \tau) = x(x^2 + (1 - x^2)e^{-2\tau})^{-1/2}. \tag{6.1b}$$

This model is an often used benchmark problem for data assimilation methods as it exhibits certain non-Gaussian behavior (Apte et al. 2007). Here we pose it as a sequential inference problem. Namely, suppose that we can observe the solution of the equation,  $v(\tau)$ , at different times  $\tau = t \cdot \Delta_t$  for  $t = 1, \dots, T$ , and we aim to estimate the initial condition  $x$  from the sequentially observed data. The observation noise is assumed to follow a zero-mean Gaussian distribution with standard deviation  $\sigma$ . In this example, we take  $T = 50$ , and  $\Delta_t = 0.3$  and we consider two different noise levels:  $\sigma = 0.4$  and  $\sigma = 0.8$ . In the numerical experiments, we set the ground truth to be  $x = 10^{-4}$  and the data is simulated from the model (6.1) for  $\sigma = 0.4$  and  $\sigma = 0.8$ , which are shown in Fig. 1. In the experiments the prior distribution for  $x$  is taken to be uniform:  $U[-1, 10]$ .

We sample the posterior distribution with four methods: the EnKF method in Iglesias et al. (2013), EnKF-SMCS (Algorithm 1), EnKF-SMCS-WR (Algorithm 2) and MH-SMCS. Note that MH-SMCS is the SMCS with a Metropolis–Hastings forward proposal, a commonly used implementation of SMCS. (We provide a detailed description of the algorithm in Appendix A.) In each method, we use 200 particles, and the bias error, i.e., the difference between the sample mean, which is a commonly used estimator, and the ground truth is then computed at each time step. The procedure is repeated 100 times and the averaged results are shown in Fig. 2 where the left figure show the results for the small noise case ( $\sigma = 0.4$ ) and the right figure shows those for the large noise case ( $\sigma = 0.8$ ).

First, one can see from the figures that all the methods perform better in the small noise case, which is sensible as

intuitively the inference should be more accurate when the observation noise is small. More importantly, we can also see that in both cases, the EnKF results in significantly higher errors than the three SMCS methods, suggesting that EnKF performs poorly for this example. On the other hand, we observe that the three SMCS algorithms produce largely the same results in both cases, while EnKF-SMCS-WR only calculates the actual sample weights at 9 time steps on average in the small noise case and 6 in the large noise case, as is compared to 50 in EnKF-SMCS. Such a difference suggests that the EnKF-SMCS-WR algorithm can significantly reduce the computational cost associated with the weight computation. The two EnKF-SMCS algorithms and MH-SMCS yield similar results, but we need to emphasize that MH-SMCS is substantially more expensive than EnKF-SMCS-WR, as its procedure is similar to that of MCMC (see Appendix A for details).

### 6.2 Lorenz 63 model

Our second example is the Lorenz 63 model, a popular example used in several works on parameter estimation, such as Annan and Hargreaves (2004), Mehrkanoon et al. (2012). Specifically the model consists of three variables  $x$ ,  $y$  and  $z$ , evolving according to the differential equations

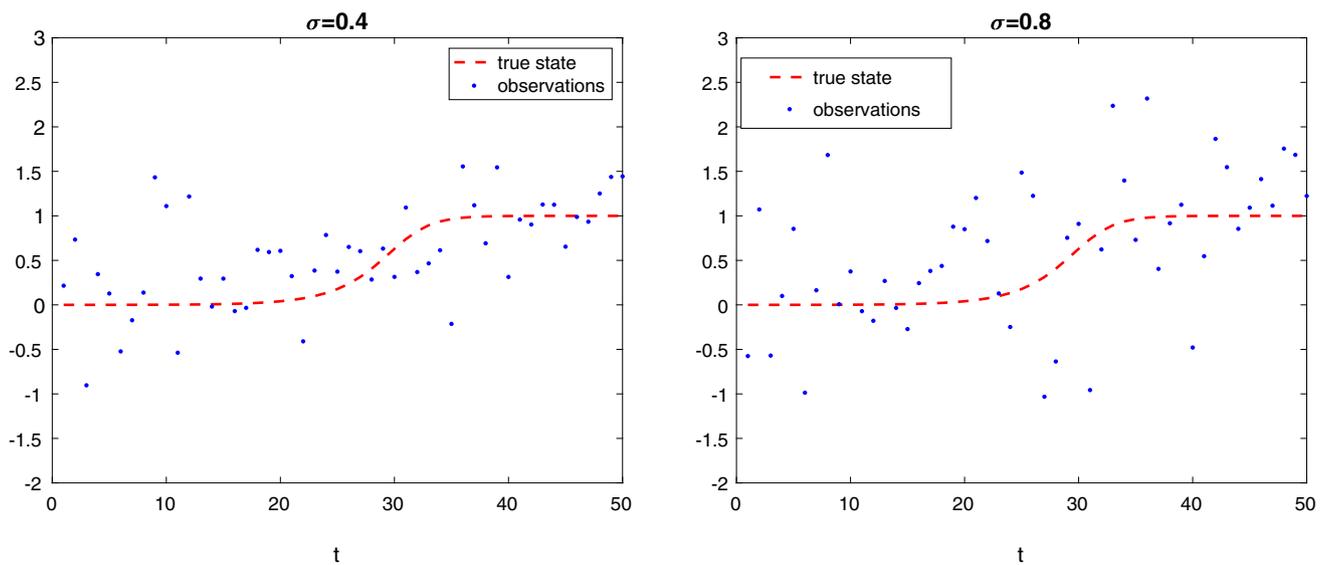
$$\frac{dx}{d\tau} = \alpha(y - x), \tag{6.2a}$$

$$\frac{dy}{d\tau} = x(\rho - z) - y, \tag{6.2b}$$

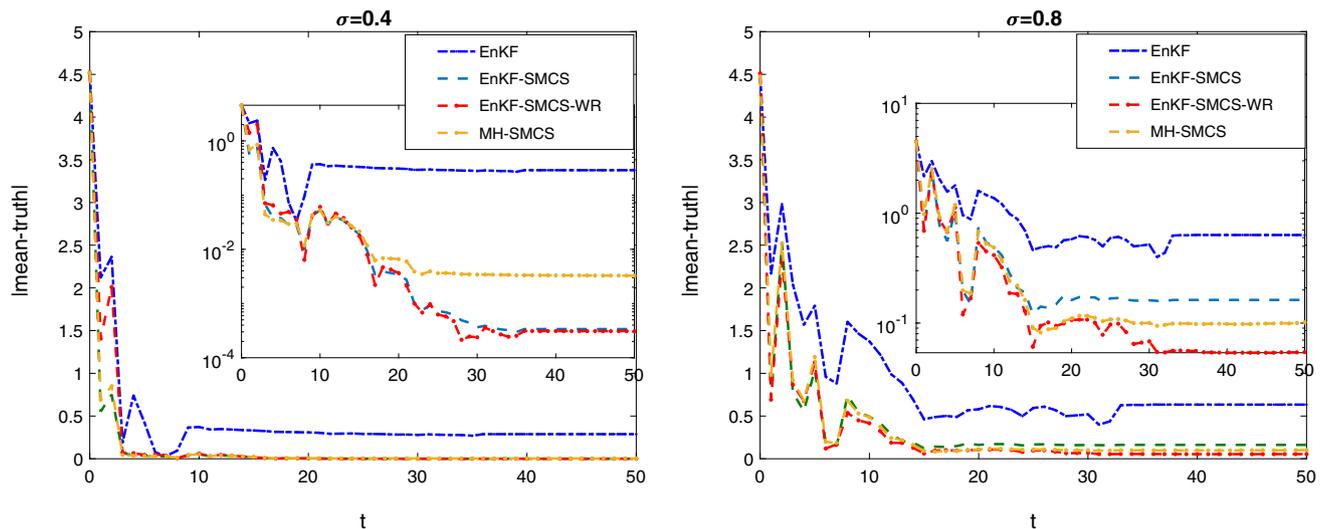
$$\frac{dz}{d\tau} = xy - \beta z, \tag{6.2c}$$

where  $\alpha$ ,  $\rho$  and  $\beta$  are three constant parameters. In this example we take the true values of the three parameters to be  $\alpha = 10$ ,  $\beta = 8/3$  and  $\rho = 28$ , which we assume that we have no knowledge of. Now suppose that observations of  $(x, y, z)$  are made at a sequence of discrete time points:  $\tau = t \cdot \Delta_t$  for  $\Delta_t = 0.1$  and  $t = 1, \dots, 50$ , and we want to estimate the three parameters  $(\alpha, \beta, \rho)$  from these observed data. The measurement noise here is taken to be zero-mean Gaussian with variance  $3^2$ , and the priors of the three parameters are also taken to be Gaussian with means  $[6, 0, 24]$ , and variances  $[1, 1, 1]$ . (The prior is chosen so that it covers the regime that can result in chaotic behavior). The data used in our numerical experiments are shown in Fig. 3.

In the numerical experiments, we conduct inference for two different cases: one is that variable  $x$  is observed and the other is that  $y$  is observed. In each case, we draw samples from the posterior distributions with EnKF, EnKF-SMCS and EnKF-SMCS-WR, MH-SMCS, where 500 samples are drawn with each method. All the numerical experiments are repeated 10 times. We plot the average errors for the case



**Fig. 1** Simulated data for  $\sigma = 0.4$  (left) and  $\sigma = 0.8$  (right). The lines show the simulated states in continuous time and the dots are the noisy observations

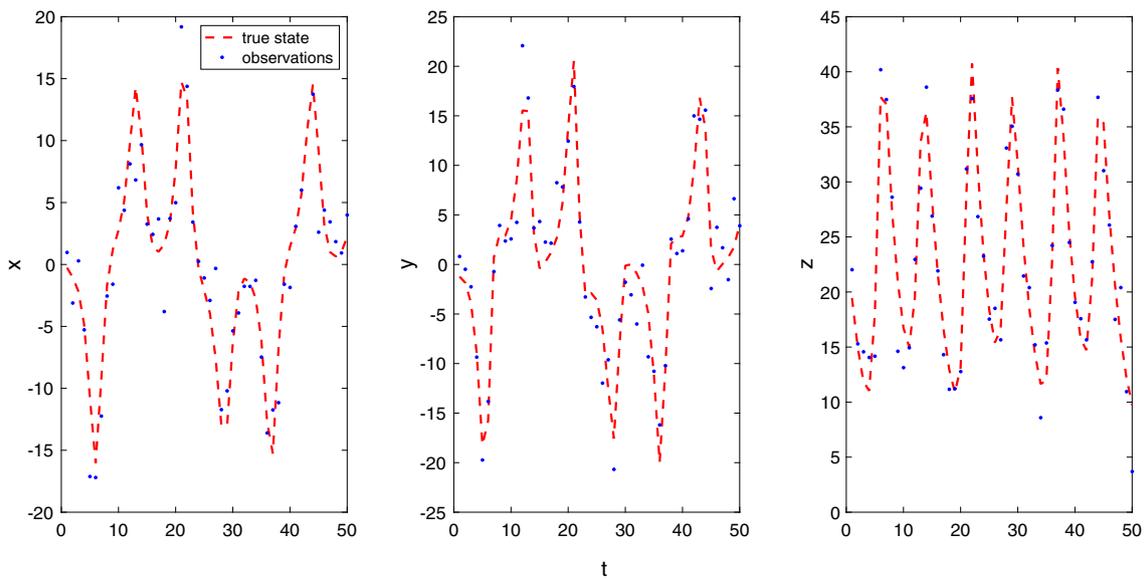


**Fig. 2** Average bias error (the difference between the sample mean and the ground truth) plotted at each time step where the insets are the same plots on a logarithmic scale. The left plot is the error for  $\sigma = 0.4$  and the right figure is that for  $\sigma = 0.8$

that  $x$  is observed in Fig. 4 and those for that with  $y$  being observed in Fig. 5. One can see that, in both cases, the errors in the EnKF is larger than those in the three SMCS methods, especially for parameter  $\alpha$ . Once again, the three SMCS methods yield similar errors while EnKF-SMCS-WR employs much less computations of the actual weight: on average 9 time steps in the first case and 8 in the second. The example shows that even for problems where the posterior distributions are rather close to Gaussian, the use of SMCS can further improve the estimation accuracy.

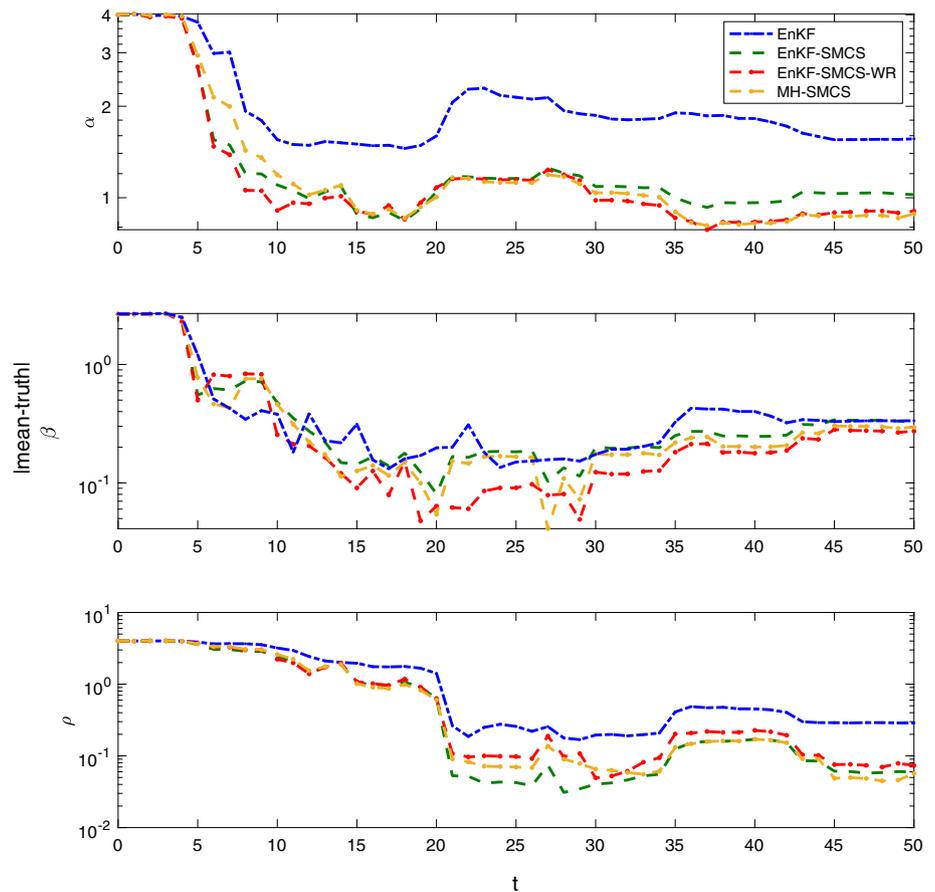
### 6.3 A kinetic model of the ERK pathway

In the last example, we consider the parameter estimation problems in the kinetic models. Estimating the kinetic parameters is an essential task in the modeling of the biochemical reaction networks, including genetic regulatory networks and signal transduction pathways (Quach et al. 2007). In particular we consider the kinetic model of the extracellular signal regulated kinase (ERK) pathway suppressed by Raf-1 kinase inhibitor protein (RKIP) (Kwang-Hyun et al. 2003; Sun et al. 2008). Here we shall omit further details of biological background of the problem and proceed directly to

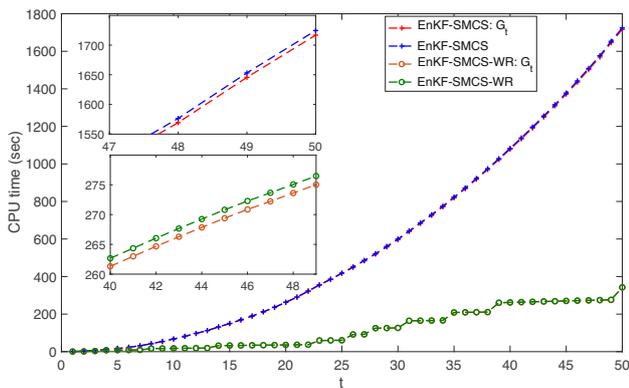
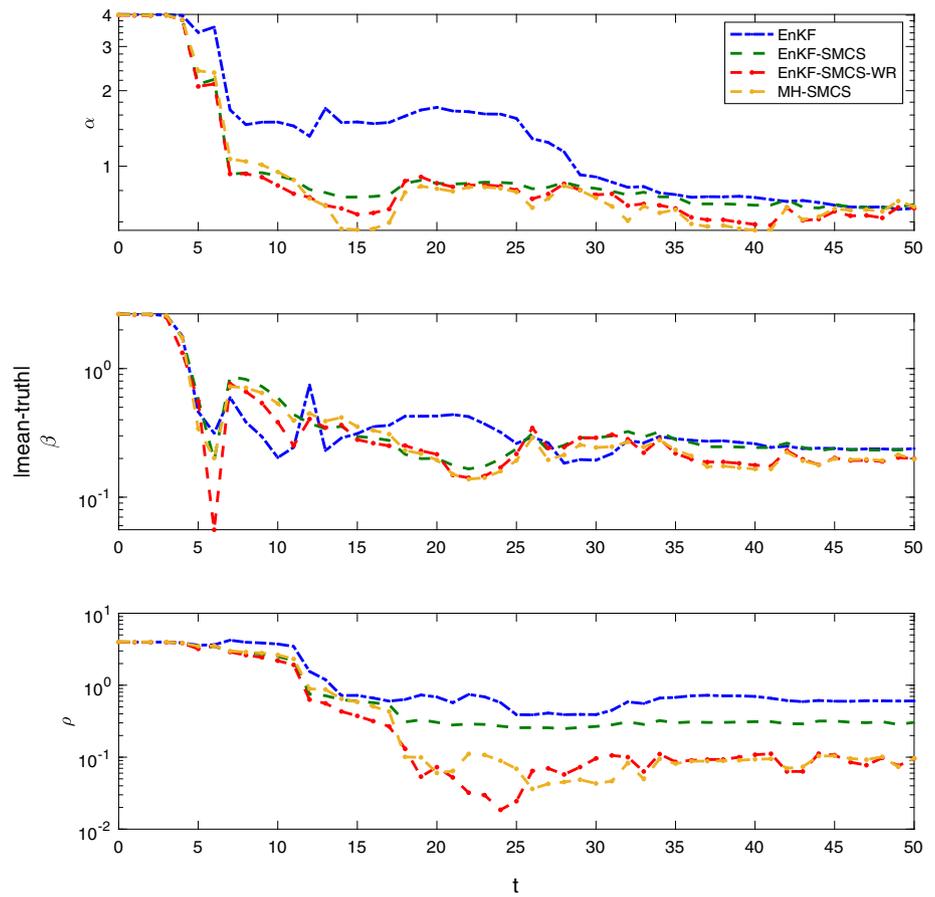


**Fig. 3** Simulated data for the Lorenz 63 example. The lines show the simulated states in continuous time and the dots are the noisy observations

**Fig. 4** Average estimation error of each parameter when  $x$  is observed



**Fig. 5** Average estimation error of each parameter when  $y$  is observed

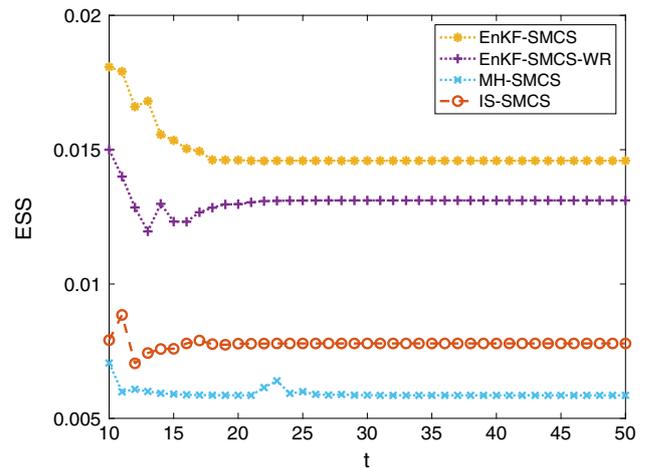


**Fig. 6** Total CPU time and the CPU time for the forward model evaluation (marked with  $G_t$ ) in both EnKF-SMCS and EnKF-SMCS-WR. Insets are the zoom-in plots

the mathematical formulation of the problem; readers who are interested in more application-related information may consult (Kwang-Hyun et al. 2003; Sun et al. 2008).

In this problem, the mathematical model that is derived based on enzyme kinetics is represented by a dynamical system:

$$\frac{dx}{d\tau} = SV(x), \tag{6.3}$$



**Fig. 7** ESS (without resampling) plotted as a function of  $t$

where  $\tau$  is the time,  $x$  is a vector of state variables which are concentrations of metabolites, enzyme and proteins or gene expression levels,  $S$  is a stoichiometric matrix that describes the biochemical transformation in a biochemical network and  $V(x)$  is the vector of reaction rates and is usually the vector of nonlinear function of the state and input variables. Specif-

ically, in this ERK pathway model we have

$$x = [x_1, x_2, \dots, x_{11}]^T, \quad V(x) = [v_1, v_2, \dots, v_7]^T,$$

which forms a system of 11 ordinary differential equations. Moreover the rates of reactions  $V(x)$  are Kwang-Hyun et al. (2003), Sun et al. (2008):

$$\begin{aligned} v_1 &= k_1x_1x_2 - k_2x_3, & v_2 &= k_3x_3x_9 - k_4x_4, \\ v_3 &= k_5x_4, & v_4 &= k_6x_5x_7 - k_7x_8, \\ v_5 &= k_8x_8, & v_6 &= k_9x_6x_{10} - k_{10}x_{11}, & v_7 &= k_{11}x_{11}, \end{aligned}$$

where  $k_1, \dots, k_{11}$  are the kinetic parameters, and the stoichiometric matrix  $S$  is given by Kwang-Hyun et al. (2003), Sun et al. (2008):

$$S = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

In this problem, we can make observations of some of the concentrations  $x_1, \dots, x_{11}$  at different times, from which we estimate the 11 kinetic parameters  $k_1, \dots, k_{11}$ . In our numerical experiments the specific setup is the following. In many practical problems, not all the species' concentrations can be conveniently observed (Kwang-Hyun et al. 2003; Sun et al. 2008). To mimic the situation, we assume that the observations can only be made on 4 of the states:  $\{x_1, x_4, x_7, x_{10}\}$ , and  $\{x_2, x_3, x_5, x_6, x_8, x_9, x_{11}\}$  are not observed. Second the observation is made 50 times with each time spacing is  $\Delta_t = 0.001$ , and the measurement noise is taken to be zero-mean Gaussian with standard deviation (STD) shown in Table 1. The initial values of the concentrations are also given in Table 1. We use simulated data in this example where the true values of the eleven parameters are shown in Table 2. The prior of the eleven parameters are also taken to be Gaussian with means and standard deviations both shown in Table 2.

In this example, we focus on the four SMCS algorithms: EnKF-SMC, EnKF-SMC-WR, MH-SMCS, and IS-SMCS (a special MH-SMCS implementation with an independent proposal; see Appendix A for details), where the main purpose is to compare the EnKF-based and the MH-based forward proposals. We test two sample sizes  $M = 5000$  and  $M = 10000$  for each method and all the tests are repeated 10 times.

First we want to examine the computational cost of EnKF-SMCS and EnKF-SMCS-WR. To do this, we plot the CPU time of the two algorithms as a function of  $t$ , where in each algorithm we show both the total time cost and that used for evaluating the forward model. First, we can see from the figure that, in both algorithms the main computational cost arises from the forward model evaluation; second, the EnKF-SMCS-WR can significantly reduce the computational cost by using less forward model evaluations. As discussed earlier, for the purpose of sequential inference, we should devote the majority of our attention to the estimator accuracy at the final step, and therefore, in Table 3 we show the estimation error for each parameter at the final step  $t = 50$ . Specifically, we provide in the table the mean-squared errors (MSE) of the estimation results. We can see from the table that the two EnKF-SMCS algorithms yield lower estimation errors than MH-SMCS in all the cases, and in particular the difference is substantially large for parameters  $k_1, k_4, k_5, k_6, k_8, k_9$  and  $k_{11}$ , with both sample sizes. The results of IS-SMCS are considerably better than those of MH-SMCS, suggesting that the posterior distributions in this problem may be reasonably close to Gaussian. That said, IS-SMCS results in clearly higher MSE for  $k_1$  and  $k_9$  (in the  $M = 10000$  case). On the other hand, the two EnKF-based SMCS algorithms yield similar performance in terms of the estimation error, but the EnKF-SMCS-WR method only conducts WR at 12 steps on average for both sample sizes, resulting in much higher computational efficiency than EnKF-SMCS.

## 7 Conclusions

In this work, we propose a sampling method to compute the posterior distribution that arises in sequential Bayesian inference problems. The method is based on SMCS, which seeks to generate weighted samples from the posterior in a sequential manner, and specifically, we propose to construct the forward kernel in SMCS using an EnKF framework and also derive a backward kernel associated with it. With numerical examples, we demonstrate that the EnKF-SMCS method can often yield more accurate estimations than the direct use of either SMCS or EnKF for a class of problems. We believe that the method can be useful in a large range of real-world parameter estimation problems where data become available sequentially in time.

Some extensions and improvements of the EnKF-SMCS algorithm are possible. First, in this work we focus on problems with a sequential structure, but we expect that the method can be applied to batch inference problems (where the data are available and used for inference altogether) as well. In fact, many batch inference problems can be artificially "sequentialized" by some data tempering treatments (Geyer 2011), and consequently, the EnKF-SMCS algorithm

**Table 1** Initial values and observation noise of the concentrations (states  $x_i$ )

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
Initial values	66	0.054	0.019	59	0.09	0.012
Noise STD	0.005	$5 \times 10^{-5}$	$2 \times 10^{-5}$	0.035	0.0005	$5 \times 10^{-6}$
	$x_7$	$x_8$	$x_9$	$x_{10}$	$x_{11}$	
Initial values	65	26	175	161	2.18	
Noise STD	0.05	0.02	0.03	0.003	0.002	

**Table 2** True values and priors of the kinetic parameters

	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$	$k_6$
Truth	0.5242	0.0075	0.6108	0.0025	0.0371	0.8101
Prior mean	0.5	0.1	0.62	0.04	-0.5	0.8
Prior STD	0.05	0.03	0.01	0.04	0.5	0.02
	$k_7$	$k_8$	$k_9$	$k_{10}$	$k_{11}$	
Truth	0.0713	0.0687	0.96	0.0012	0.872	
Prior mean	0	0.4	0.9	0	0.9	
Prior STD	0.05	0.3	0.1	0.005	0.05	

**Table 3** Comparison of the MSE results of the kinetic model at  $t = 50$

	$M = 5000$			
	EnKF-SMCS-WR	EnKF-SMCS	MH-SMCS	IS-SMCS
$k_1$	0.0078	0.0094	0.0287	0.0215
$k_2$	0.0877	0.0915	0.1011	0.0920
$k_3$	0.0093	0.0091	0.0119	0.0094
$k_4$	0.0080	0.0078	0.0201	0.0075
$k_5$	0.0003	0.0002	0.0011	0.0001
$k_6$	0.0091	0.0092	0.0202	0.0100
$k_7$	0.0668	0.0651	0.0936	0.0669
$k_8$	0.0199	0.0191	0.0421	0.0307
$k_9$	0.0108	0.0065	0.0888	0.0111
$k_{10}$	0.0012	0.0011	0.0016	0.0015
$k_{11}$	0.0068	0.0056	0.0299	0.0045
	$M = 10000$			
	EnKF-SMCS-WR	EnKF-SMCS	MH-SMCS	IS-SMCS
$k_1$	0.0092	0.0082	0.0429	0.0213
$k_2$	0.0906	0.0928	0.0942	0.0922
$k_3$	0.0095	0.0101	0.0113	0.0092
$k_4$	0.0080	0.0082	0.0128	0.0071
$k_5$	0.0001	0.0002	0.0027	0.0001
$k_6$	0.0101	0.0097	0.0177	0.0106
$k_7$	0.0695	0.0687	0.0946	0.0653
$k_8$	0.0207	0.0206	0.0373	0.0289
$k_9$	0.0023	0.0040	0.0532	0.0125
$k_{10}$	0.0014	0.0015	0.0028	0.0014
$k_{11}$	0.0023	0.0036	0.0652	0.0058

can be applied in these scenarios. In this respect, combining data tempering methods and the EnKF-SMCS method to address batch inference problems can be a highly interesting research problem. Second, as has been discussed previously, the proposed method relies on the assumption that the posterior distributions do not deviate strongly from being Gaussian. For problems with highly nonlinear models, the posterior distributions may depart far from Gaussian, and as a result, the kernels obtained with the EnKF method may not be effective for SMCS. In this case, the performance of the EnKF-SMCS method may be improved by approximating the posterior with a mixture distribution (e.g., Hoteit et al. 2008; Stordal et al. 2011). Finally, as the method is based on an EnKF scheme, it requires that the observation noise is additive Gaussian. In the EnKF literature, a number of methods have been developed to deal with non-Gaussian observations. One simple approach is to calculate the Kalman gain matrix using the sample covariance between  $u_t$  and  $y_t$  (Houtekamer and Mitchell 2001), and another example is the EnKF variant proposed in Lei and Bickel (2011). In principle, all these ideas can be used in our method to construct the EnKF forward kernel, and to this end an important question is how to effectively incorporate them in the EnKF-SMCS framework. We plan to investigate these issues in the future.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### A The MH-SMCS algorithms

In this section, we provide a brief description of the MH-SMCS algorithm, largely following (Del Moral et al. 2006). As is mentioned in Sect. 6.1, a commonly used forward proposal is the MH kernel, which generates a new sample  $x_t$  with the following procedure,

- Draw  $x^*$  from a condition distribution  $q(\cdot|x_{t-1})$ ;
- Calculate the acceptance probability  $a(x^*, x_{t-1}) = \min\{1, \frac{\pi_t(x^*)}{\pi_t(x_{t-1})} \frac{q(x_{t-1}|x^*)}{q(x^*|x_{t-1})}\}$ ;
- Draw  $v$  from  $U[0, 1]$ ;
- If  $v \leq a$ , let  $x_t = x^*$ ; otherwise let  $x_t = x_{t-1}$ .

We then choose the backward kernel given by Eq. (32) in Del Moral et al. (2006), and the resulting incremental weight is

$$\alpha_t = \frac{\pi_t(x_{t-1})}{\pi_{t-1}(x_{t-1})}.$$

It is worth mentioning that, since the MH kernel includes an acceptance–rejection step that involves  $\pi_t(\cdot)$ , the number of forward model evaluations is also at the order of  $O(T^2)$  where  $T$  is the total number of time steps. Finally we note that, in the numerical experiments, we choose two proposal distribution within the MH kernel: (1) a random walk proposal,  $q(\cdot|x_{t-1}) = \mathcal{N}(x_{t-1}, \lambda^2 I)$ , and the variance  $\lambda^2$  is adjusted so that the average acceptance rate is about 50%; (2) an independence sampler proposal  $q(\cdot|x_{t-1}) = \hat{q}_{t-1}(\cdot)$  where  $\hat{q}_{t-1}(\cdot)$  is given by Eq. (5.5). We refer to the former as MH-SMCS and the latter as IS (independent sampler)-SMCS. Finally we note that, in principle, multiple-iteration MH proposal can be used in SMCS; however, since each MH iteration requires an evaluation of the forward model, which may result in formidable computational cost, here we only perform one MH iteration in both MH-SMCS and IS-SMCS.

### References

Annan, J.D., Lunt, D.J., Hargreaves, J.C., Valdes, P.J.: Parameter estimation in an atmospheric GCM using the ensemble Kalman filter. In: *Nonlinear Processes in Geophysics*, vol. 12, pp. 363–371. European Geosciences Union (EGU) (2005)

Annan, J.D., Hargreaves, J.C.: Efficient parameter estimation for a highly chaotic system. *Tellus A Dyn. Meteorol. Oceanogr.* **56**(5), 520–526 (2004)

Apte, A., Hairer, M., Stuart, A.M., Voss, J.: Sampling the posterior: an approach to non-gaussian data assimilation. *Physica D* **230**(1–2), 50–64 (2007)

Arnold, A., Calvetti, D., Somersalo, E.: Parameter estimation for stiff deterministic dynamical systems via ensemble Kalman filter. *Inverse Prob.* **30**(10), 105008 (2014)

Beal, M.J.: Variational algorithms for approximate Bayesian inference. PhD thesis, UCL (University College London) (2003)

Beskos, A., Jasra, A., Law, K., Tempone, R., Zhou, Y.: Multi-level sequential monte Carlo samplers. *Stochastic Process. Appl.* **127**(5), 1417–1440 (2017)

Chopin, N., Papaspiliopoulos, O.: *An introduction to sequential Monte Carlo* (2020)

Chopin, N.: A sequential particle filter method for static models. *Biometrika* **89**(3), 539–552 (2002)

Dai, C., Heng, J., Jacob, P.E., Whiteley, N.: An invitation to sequential Monte Carlo samplers. arXiv preprint [arXiv:2007.11936](https://arxiv.org/abs/2007.11936) (2020)

Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. *J. Roy. Stat. Soc. Ser. B (Statistical Methodology)* **68**(3), 411–436 (2006)

Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: fifteen years later. *Handb. Nonlinear Filter* **12**(656–704), 3 (2009)

Evensen, G.: *Data Assimilation: The Ensemble Kalman Filter*. Springer, New York (2009)

- Everitt, R.G., Culliford, R., Medina-Aguayo, F., Wilson, D.J.: Sequential Monte Carlo with transformations. *Stat. Comput.* **30**(3), 663–676 (2020)
- Fearnhead, P., Taylor, B.M., et al.: An adaptive sequential Monte Carlo sampler. *Bayesian Anal.* **8**(2), 411–438 (2013)
- Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D., Vehtari, A., Rubin, D.B.: *Bayesian Data Analysis*, 3rd edn. Chapman & Hall, London (2013)
- Geyer, C.J.: Importance sampling, simulated tempering, and umbrella sampling. In: *Handbook of Markov Chain Monte Carlo*, pp. 321–338. Chapman and Hall/CRC (2011)
- Gilks, W.R., Berzuini, C.: Following a moving target—Monte Carlo inference for dynamic bayesian models. *J. Roy. Stat. Soc. Ser. B (Statistical Methodology)* **63**(1), 127–146 (2001)
- Gilks, W.R., Richardson, S., Spiegelhalter, D.: *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, London (1995)
- Golightly, A., Wilkinson, D.J.: Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface Focus* **1**(6), 807–820 (2011)
- Green, P.L., Maskell, S.: Estimating the parameters of dynamical systems from big data using sequential Monte Carlo samplers. *Mech. Syst. Signal Process.* **93**, 379–396 (2017)
- Heng, J., Bishop, A.N., Deligiannidis, G., Doucet, A., et al.: Controlled sequential Monte Carlo. *Ann. Stat.* **48**(5), 2904–2929 (2020)
- Hoteit, I., Pham, D.-T., Triantafyllou, G., Korres, G.: A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. *Mon. Weather Rev.* **136**(1), 317–334 (2008)
- Houtekamer, P.L., Mitchell, H.L.: A sequential ensemble Kalman filter for atmospheric data assimilation. *Mon. Weather Rev.* **129**(1), 123–137 (2001)
- Iglesias, M.A.: A regularizing iterative ensemble Kalman method for PDE-constrained inverse problems. *Inverse Prob.* **32**(2), 025002 (2016)
- Iglesias, M.A., Law, K.J.H., Stuart, A.M.: Ensemble Kalman methods for inverse problems. *Inverse Prob.* **29**(4), 045001 (2013)
- Kwang-Hyun, C., Sung-Young, S., Hyun-Woo, K., Wolkenhauer, O., McFerran, B., Kolch, W.: Mathematical modeling of the influence of RKIP on the ERK signaling pathway. In: *International Conference on Computational Methods in Systems Biology*, pp. 127–141. Springer (2003)
- Lei, J., Bickel, P.: A moment matching ensemble filter for nonlinear non-Gaussian data assimilation. *Mon. Weather Rev.* **139**(12), 3964–3973 (2011)
- Mehrkanoon, S., Falck, T., Suykens, J.: Parameter estimation for time varying dynamical systems using least squares support vector machines. In: *Proceedings of the 16th IFAC Symposium on System Identification (SYSID 2012)*, vol. 16, pp. 1300–1305 (2012)
- Minka, T.P.: A family of algorithms for approximate Bayesian inference. PhD thesis, Massachusetts Institute of Technology (2001)
- Papadakis, N., Mémin, É., Cuzol, A., Gengembre, N.: Data assimilation with the weighted ensemble Kalman filter. *Tellus A Dyn. Meteorol. Oceanogr.* **62**(5), 673–697 (2010)
- Quach, M., Brunel, N., Buc, F.A.: Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics* **23**(23), 3209–3216 (2007)
- Sanjeev Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking. *IEEE Trans. Signal Process.* **50**(2), 174–188 (2002)
- Schillings, C., Stuart, A.M.: Analysis of the ensemble Kalman filter for inverse problems. *SIAM J. Numer. Anal.* **55**(3), 1264–1290 (2017)
- 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. *Comput. Geosci.* **15**(2), 293–305 (2011)
- Sun, X., Jin, L., Xiong, M.: Extended Kalman filter for estimation of parameters in nonlinear state-space models of biochemical networks. *PLoS ONE* **3**(11), e3758 (2008)
- Wen, L., Jiangqi, W., Linjun, L., Li, J.: A defensive marginal particle filtering method for data assimilation. *SIAM/ASA J. Uncertain. Quant.* **8**(3), 1215–1235 (2020)
- Zhou, Y., Johansen, A.M., Aston, J.A.D.: Toward automatic model comparison: an adaptive sequential Monte Carlo approach. *J. Comput. Gr. Stat.* **25**(3), 701–726 (2016)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.