Noname manuscript No.
(will be inserted by the editor)

Ensemble Kalman filter based Sequential Monte Carlo Sampler for sequential Bayesian inference

Jiangqi Wu · Linjie Wen · Peter L. Green · Jinglai Li · Simon Maskell

Received: date / Accepted: date

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 can not 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). An advantage of the SMCS method is that, unlike the EnKF method that 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 (2000) 62F15 · 65C05

This work was supported by the NSFC under grant number 11301337 and by the EPSRC under grant number EP/R018537/1.

J. Wu · L. Wen
School of Mathematical Sciences, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai 200240, China.

P. L. Green
School of Engineering, University of Liverpool, Liverpool L69 7XL, UK.
E-mail: plgreen@liverpool.ac.uk

J. Li
School of Mathematics, University of Birmingham, Birmingham B15 2TT, UK.
E-mail: j.li.10@bham.ac.uk

Simon Maskell
Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 7XL, UK.
E-mail: S.Maskell@liverpool.ac.uk

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 [16]. In the current work we consider a special class of Bayesian inference problem where data have to be collected in a sequential manner. A typical example of this type of problem 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 [1] to biochemical networks [20]. 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 do not 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 that approximation based methods, such as the Variational Bayes [6] and the Expectation Propagation [28] will not be discussed here. Conventional sampling methods, such as Markov Chain Monte Carlo (MCMC) simulations [19], use all the data in a single batch, are unable to take advantage of the sequential structure of the type of problems. On the other hand, 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 filtering (EnKF) algorithm, which was initially developed to address the dynamical state estimation problems [13]. The EnKF method was extended to estimate parameters in many practical problems, e.g., [1, 2], and more recently it was generically formulated as a derivative-free optimization based parameter estimation method in [25]. The EnKF method for parameter estimation was further developed and analyzed in [4, 24, 32], etc. The basic idea of the EnKF method 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 the approximation may result in substantial approximation error, unless the actual posterior is highly close to Gaussian. Moreover the approximation error, unlike random sampling error, can not be reduced by increasing the sample size. On the other hand, the Sequential Monte Carlo sampler (SMCS) method [11], does not have such a limitation. The SMCS algorithm is a generalisation of the particle filter [5, 12] for dynamic state estimation, generating weighted samples from the posterior distribution. Since the SMCS algorithm was proposed in [11], considerable improvements and extensions of the method have been proposed, such as, [15, 7, 22, 14], and more information on the developments of the SMCS methods can be found in the recent reviews [10, 9] On the other hand, we need to note that there are other parameter estimation schemes also based on particle filtering, e.g., [18, 8], and the differences and connections between SMCS and these schemes are discussed in [11]. The SMCS method makes no assumption or approximation

of the posterior distribution, and can directly draw (weighted) samples from any posterior. As will be discussed later, a key issue in the implementation of SMCS is the choice of suitable forward and backward kernels, as the performance of SMCS depends critically on the choices of these kernels. As has been shown in [11], 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 [30,35], especially for problems in which the posteriors are 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. With several numerical examples we illustrate that the proposed method performs competitively relative to the EnKF approach. The numerical results also demonstrate that the EnKF-SMCS algorithm performs well for highly non-Gaussian posteriors.

The remaining work is organized as follows. In Section 2 we present the generic setup of the sequential inference problems that we consider in this work. In Sections 3 and 4 we respectively review the SMCS and the EnKF methods for solving sequential inference problems. In Section 5 we present the proposed EnkF-SMCS method and in Section 6 we provide several numerical examples to illustrate the performance of the proposed method. Finally Section 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 $y_1, ..., y_t, ..., y_T$ 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,

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

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,

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

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:

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

where $\pi_0(x)$ is the prior distribution of x, and our goal is to draw samples from π_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 [15,36], 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 [11]. The key idea of SMCS is to construct a joint distribution $\pi(x_1, ..., x_t)$, the marginal of which is equal to the target distribution $\pi_t(\cdot)$. Note here that $\pi(x_1, ..., x_t)$ needs only to be known up to a normalization constant. One then applies the sequential importance sampling algorithm [5,12] to draw weighted samples from $\pi(x_1, ..., x_t)$, which after being marginalized over $x_1, ..., x_{t-1}$, 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 $p(x_{t-1}, x_t)$ in the form of

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

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

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

where

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

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 $\{x^m_0\}^M_{m=1}$ from $q_0(x_0)$, and compute $w^m_0 = \pi_0(x^m_0)/q_0(x^m_0)$ for $m = 1...M$;

2. let $t = t + 1$;
3. draw x_t^m from $K(\cdot|x_{t-1}^m)$ for each $m = 1...M$;
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 algorithm to alleviate the "sample degeneracy" issue [11]. The resampling techniques are well documented in the PF literature, e.g., [5,12], 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 [11]:

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

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 π_t. For example, it is proposed in [11] 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 Σ_t, where $T_t(\cdot)$ is a $\mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ transformation. We shall compute T_t and Σ_t (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 [25], 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 to 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:

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

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 $m = 1...M$, 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 μ_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

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

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 $x_t = x_{t-1}$, we can derive from Eq. (4.6) that,

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

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

where $Q_t^x = Q_t[1 : n_x, 1 : n_y]$, δ is a small constant and Σ_{t-1}^q is the covariance of q_{t-1} (the evaluation of Σ_{t-1}^q is provided in Eq. (5.5b)). 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, η_t', in Eq. (5.1a) is to ensure that Σ_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, δ is set to be 10^{-4}. According to the discussion in Section 4, we have, if q_{t-1} is a good approximation to π_{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 here that since T_n is a nonlinear transform, in general, we can not derive the analytical expression for q_t and as a result, we can not 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 $\{x_{t-1}^m\}_{m=1}^M$:

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

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

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 can not be computed analytically. Here to obtain $\hat{L}_{t-1}$ in Eq. (5.4) explicitly, we take

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

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

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

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

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

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 $m = 1...M$ 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 ξ_{t-1} and Σ_{t-1}^q from the ensemble $\{(x_{t-1}^m, w_{t-1}^m)\}_{m=1}^M$ using Eq. (5.5b);
- let $\tilde{u}_t^m = [x_{t-1}^m, G(x_{t-1})^m]^T$ for $m = 1...M$;
- 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 $m = 1...M$ 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, 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 therefore, the total computational cost can be prohibitive if the total number of steps, T, is large. We propose a method to tackle the issue, which is based on the following two observations. First, in sequential inference problems, one is only interested in the posterior distribution at the final step where all data are incorporated; second, in

many practical problems 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 $\pi_t(\mathbf{x}_t)/\pi_{t-1}(\mathbf{x}_{t-1})$ which involves computing the forward models from $G_1(\mathbf{x}_t)$ all the way to $G_t(\mathbf{x}_t)$, 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 $\pi_t(\mathbf{x}_t)/\pi_{t-1}(\mathbf{x}_{t-1})$ as,

$$\frac{\pi_t(\mathbf{x}_t)}{\pi_{t-1}(\mathbf{x}_{t-1})} = \frac{\pi_{t-1}(\mathbf{x}_t)}{\pi_{t-1}(\mathbf{x}_{t-1})}\pi(y_t|\mathbf{x}_t),$$

  and naturally we can approximate π_{t-1} with q_{t-1}, yielding,

$$\frac{\pi_t(\mathbf{x}_t)}{\pi_{t-1}(\mathbf{x}_{t-1})} \approx \frac{q_{t-1}(\mathbf{x}_t)}{q_{t-1}(\mathbf{x}_{t-1})}\pi(y_t|\mathbf{x}_t).$$

  In principle q_{t-1} can be evaluated via Eq. (5.3), however, this is still computionally expensive. Thus we make another approximation, replacing q_t with $\hat{q}_t$, where $\hat{\pi}_{t-1}$ is the Gaussian approximation of q_t given by Eqs. (5.5), and as a result, we obtain

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

  which is used to compute the approximate weights.
- Second we need to prescribe the conditions for triggering the computation of the actual weights. Following [21], we use the Effective Sample Size (ESS) [12] (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., $t = T$.
- Finally we shall discuss how to compute the actual weight w_t. It should be noted here that the recursive formulas (3.3) can not be used here since the actual value of w_{t-1} is not available. However, letting 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. Finally note that in both EnKF-SMCS algorithms, a resampling step is needed.

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 $m = 1...M$ 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 ξ_{t-1} and Σ_{t-1}^q from the ensemble $\{(x_{t-1}^m, w_{t-1}^m)\}_{m=1}^M$ using Eq. (5.5b);
  let $\tilde{u}_t^m = [x_{t-1}^m, G(x_{t-1})^m]^T$ for $m = 1...M$;
  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 $m = 1...M$ with K_t given by Eq. (5.2);
  calculate the approximate weights for $m = 1...M$:

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

  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 $ESS < \mathrm{ESS}_{\min} \vee t - t_0 > \Delta T_{\max} \vee t = T$ **then**
    compute $\hat{L}_{t-1}$ from Eq. (5.8);
    calculate the weights for $m = 1...M$:

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

    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 $ESS < \mathrm{ESS}_{\mathrm{resamp}}$ **then**
      resample;
    end
    let $t_0 = t$;
  end
end

6 Numerical examples

We shall provide three examples to demonstrate the performance of the proposed EnKF-SMCS algorithms. The first example is used to illustrate that the proposed method can perform well when the posterior is strongly non-Gaussian and the EnKF becomes highly inaccurate. The second and the third examples show that even for problems where the EnKF performs reasonably well, the EnKF-SMCS can further improve the performance. We emphasize here that in all these examples, we assume that 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 in this section 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,

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

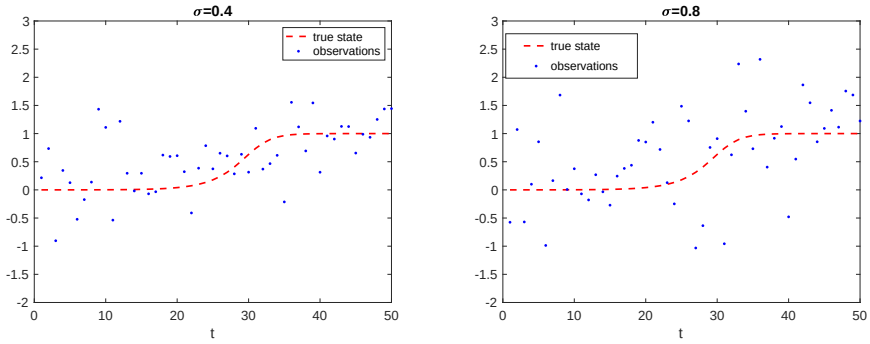

Fig. 1 The simulated data for $\sigma = 0.4$ (left) and $\sigma = 0.8$ (right). The lines show the simulated states in continuous time and the dots are the noisy observations.

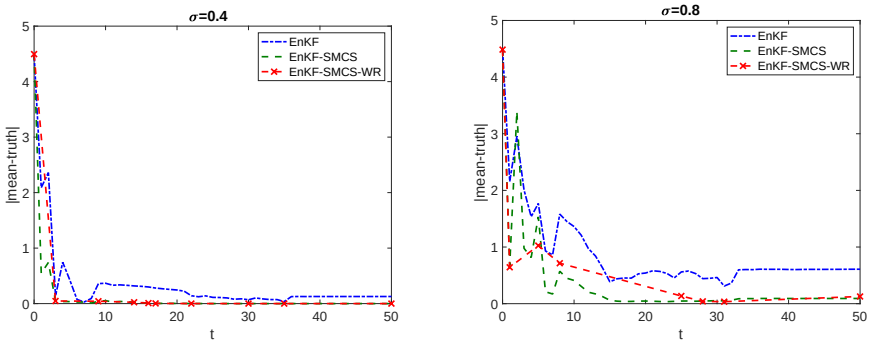

Fig. 2 The estimation bias error (the difference between the sample mean and the ground truth) plotted at each time step. The left plot is the bais for $\sigma = 0.4$ and the right figure is that for $\sigma = 0.8$.

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 strong non-Gaussian behavior [3]. 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 $\tau = t \cdot \Delta_t$ $\tau = \Delta t$ for $t = 1, ..., T$, 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 σ. In this example, we take $T = 50$, and $\Delta_t = 0.3$ and, moreover, we consider two different noise levels: $\sigma = 0.4$ and $\sigma_t = 0.8$. 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 Figs. 1. In the experiments the prior distribution for x is taken to be uniform: $U[-1, 10]$.

We sample the posterior distribution with three methods: the EnKF method in [25], EnKF-SMCS (Algorithm 1), and EnKF-SMCS-WR (Algorithm 2). In each method, we use 5×10^4 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

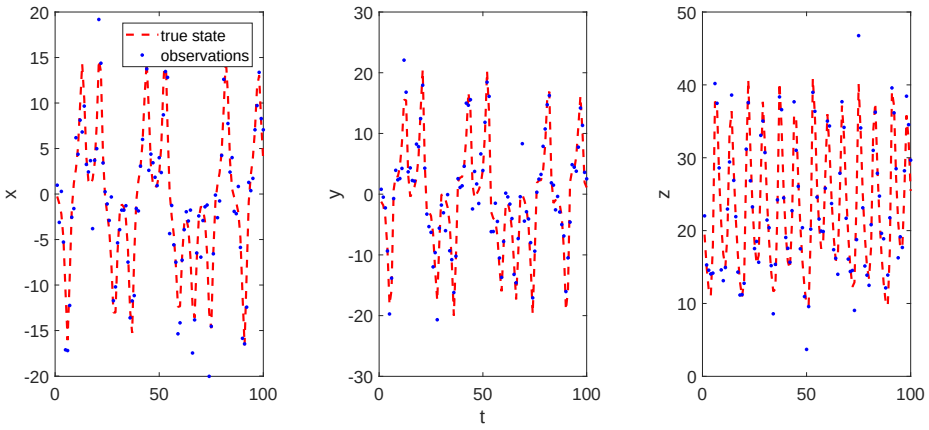

Fig. 3 The simulated data for the Lorenz 63 example. The lines show the simulated states in continuous time and the dots are the noisy observations.

computed at each time step. The results are shown in Figs. 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 ($\sigma = 0.8$). It is important to note here that, in the EnKF-SMCS-WR method, only the time steps where the actual weights are computed are shown (marked by asterisks). In other words, only the asterisk signs show the correct estimation errors and the line is merely for visual guidance. First, one can see from the figures that all the methods perform 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 two SMCS methods, suggesting that EnKF performs poorly for this highly non-Gaussian example. On the other hand, we observe that the two SMCS algorithms produce largely the same results in both cases (recall that only the asterisks show the results), while EnKF-SMCS-WR only calculates the actual sample weights at 10 time steps in the small noise case and 8 in the large noise case, as is compared to 50 in the EnKF-SMCS, which suggests that the proposed EnKF-SMCS-WR algorithm can significantly reduce the computational cost associated with the weight computation.

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 [1,27]. Specifically the model consists of three variables x, y and z, evolving according to the differential equations

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

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

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

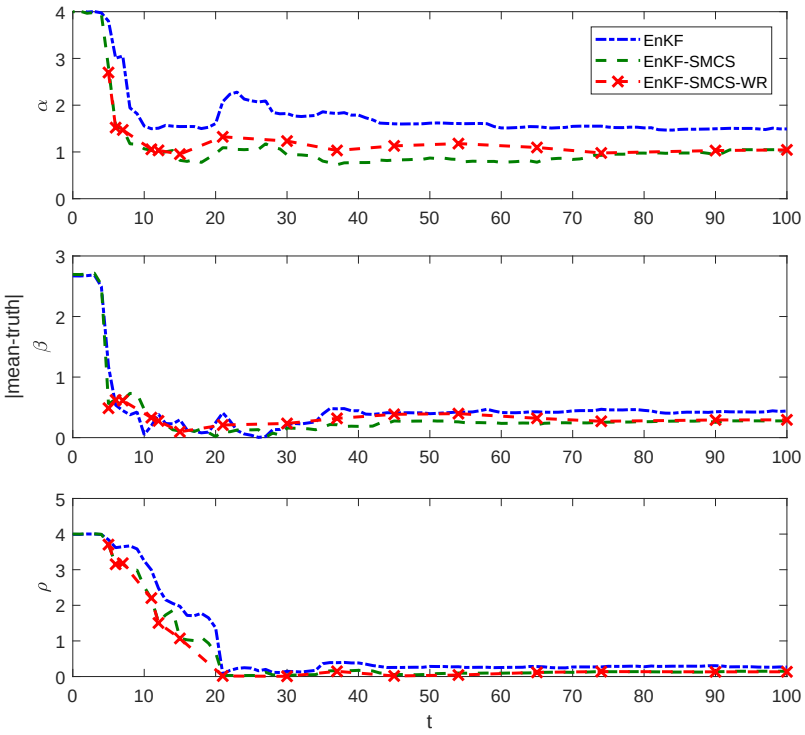

Fig. 4 The estimation bias error of each parameter when x is observed.

where α, ρ and β 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 $t = 1, ..., 100$, and we want to estimate the three parameters (α, β, ρ) 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 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, where 10^3 samples are drawn with each method. We plot the estimation bias errors for the case that x is observed in Fig. 4 and those for that with y being observed in Fig. 5. As before, the time steps where the actual weights are computed are marked by asterisks. One can see that, in both cases, the errors in the EnKF is larger than those in the two SMCS methods, especially for parameter α. Once gain, the two SMCS methods yield largely the same bias errors while EnKF-SMCS-WR employs much less computations of the actual weight: 15

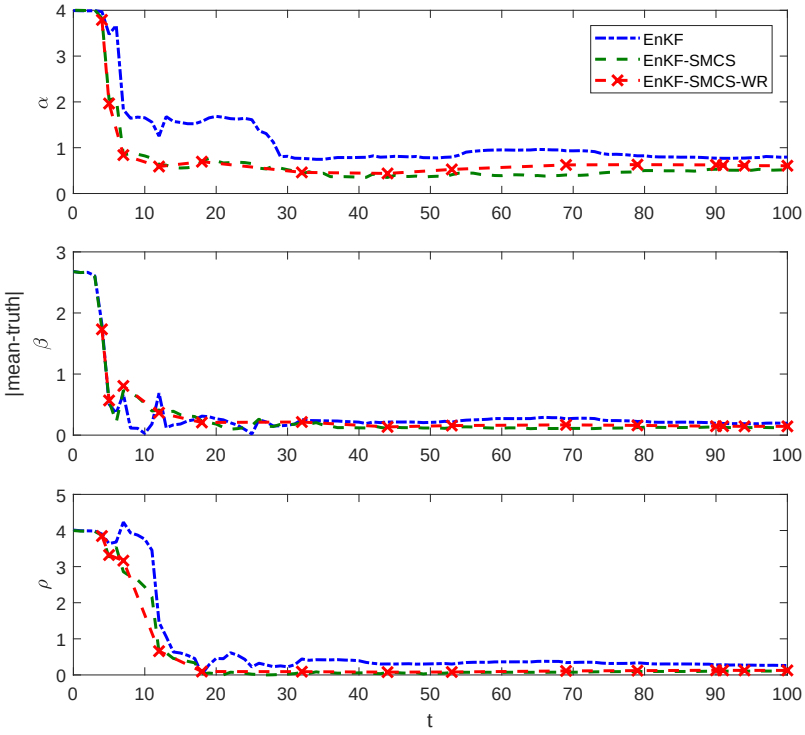

Fig. 5 The estimation bias error of each parameter when y is observed.

time steps in the first case and 14 in the second (indicated by asterisks in Fig. 5). The example shows that even for problems where the posterior distributions are rather close to Gaussian, the SMCS can further improve the estimation accuracy.

Table 1 The estimation bias of each parameter when x is observed

	t	54	65	74	90	100
α	EnKF	1.608	1.54	1.552	1.496	1.492
	EnKF-SMCS	0.8389	0.7805	0.9274	0.9383	1.046
	EnKF-SMCS-WR	1.179	1.095	0.978	1.03	1.044
β	EnKF	0.4264	0.4262	0.468	0.4262	0.4394
	EnKF-SMCS	0.2601	0.2123	0.2496	0.2535	0.2751
	EnKF-SMCS-WR	0.3963	0.3195	0.2699	0.2914	0.2936
ρ	EnKF	0.275	0.2808	0.2699	0.3043	0.265
	EnKF-SMCS	0.0981	0.1176	0.1373	0.1362	0.1492
	EnKF-SMCS-WR	0.0421	0.1167	0.1396	0.1313	0.1334

Table 2 The estimation bias of each parameter when y is observed

	t	79	90	91	94	100
α	EnKF	0.8221	0.7711	0.7686	0.7733	0.795
	EnKF-SMCS	0.4839	0.5297	0.5091	0.5101	0.5177
	EnKF-SMCS-WR	0.6305	0.6239	0.6154	0.6083	0.6063
β	EnKF	0.2232	0.1924	0.1909	0.1864	0.1956
	EnKF-SMCS	0.1228	0.1331	0.1259	0.1239	0.1243
	EnKF-SMCS-WR	0.1591	0.1462	0.142	0.1439	0.1436
ρ	EnKF	0.3362	0.2767	0.2836	0.2706	0.261
	EnKF-SMCS	0.0939	0.1041	0.1032	0.1055	0.1084
	EnKF-SMCS-WR	0.1188	0.1293	0.1266	0.1261	0.1255

6.3 A kinetic model of the ERK pathway

In the last example we consider the parameter estimation problems in kinetic models of biochemical networks. Estimating the kinetic parameters is an essential task in kinetic modeling of the biochemical reaction networks, including genetic regulatory networks and signal transduction pathways [31]. In particular we consider the kinetic model of the Extracellular signal Regulated Kinase (ERK) pathway suppressed by Raf-1 kinase inhibitor protein (RKIP) [26,34]. Here we shall omit further details of biological background of the problem and proceed directly to the mathematical formulation of the problem; readers who are interested in more application-related information may consult [26,34].

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

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

where τ 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. Specifically, in this ERK pathway model we have

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

which forms a system of 11 ordinary differential equations. Moreover the rates of reactions $V(x)$ are [26,34]:

$$\begin{aligned}
v_1 &= k_1 x_1 x_2 - k_2 x_3, \\
v_2 &= k_3 x_3 x_9 - k_4 x_4, \\
v_3 &= k_5 x_4, \\
v_4 &= k_6 x_5 x_7 - k_7 x_8, \\
v_5 &= k_8 x_8, \\
v_6 &= k_9 x_6 x_{10} - k_{10} x_{11}, \\
v_7 &= k_{11} x_{11},
\end{aligned}$$

where $k_1, ..., k_{11}$ are the kinetic parameters, and the stoichiometric matrix S is given by [26,34]:

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

In this problem, we can make observations of some of the concentrations $x_1, ...x_{11}$ at different times, from which we estimate the 11 kinetic parameters $k_1, ..., k_{11}$. In our numerical experiments the specific setup is the following. In many practical problems, not all the species' concentrations can be conveniently observed [26,34]. 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 3. The initial values of the concentrations are also given in Table 3. We use simulated data in this example where the true values of the eleven parameters are shown in Table 4. The prior of the eleven parameters are also taken to be Gaussian with means and standard deviations both shown in Table 4.

Next we compare the simulation results of the EnKF-SMCS-WR and EnKF. With each method we generate 10000 samples, and for the EnKF-SMCS-WR method, the actual weights are computed at steps 2, 4, 8, 13, 17, 25, 27, 28, 30, 36, 39, 41, 50. We then select 4 representative steps (note that at all these steps the actual weights are computed): $t = 30$, 36, 41, and 50, and at each time step we compute the sample mean, standard deviation, and bias error, where the results are shown in Table 5. We restate here that the sample mean is usually used as an estimator of the parameters and the bias error can be used to measure the performance of the estimator. For the purpose of sequential inference, we should devote the majority of our attention to the estimator accuracy at the final step. Therefore to compare the performance of the two methods, we mark the smaller bias error at $t = 50$ as bold in Table 5, where one can see that the EnKF-SMCS-WR method yields smaller bias error in all but two dimensions (k_4 and k_5). It should be noted here that there is another dimensions k_2 where the difference is rather small and may not be statistically significant. That said, we can see that EnKF-SMCS-WR has a better performance overall, as it is either more accurate or close to the EnKF in terms of the bias error, in all dimensions but k_4. We should also mention that we have also conducted the simulation with EnKF-SMCS, whose results are very close to those of EnKF-SMCS-WR and are omitted here.

Table 3 The 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	5×10^{-5}	2×10^{-5}	0.035	0.0005	5×10^{-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 4 The 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	

7 Conclusions

In this work we propose a sampling method to compute the posterior distribution that arrises 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 to 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 becomes 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 [17] and, consequently, the EnKF-SMCS algorithm can be applied in these scanrios. 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 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. [23,33]). Finally, as the method is based on an EnKF scheme, it requires that the observation noise is additive Gaussian. We believe that such a requirement can be relaxed by borrowing the ideas of some meth-

Table 5 Comparison of the results of the kinetic model.

		EnKF – SMCS – WR				**EnKF**			
	t	30	36	41	50	30	36	41	50
k_1	Mean	0.4829	0.4969	0.5129	0.5178	0.4804	0.4946	0.5124	0.5165
	STD	0.0329	0.0302	0.0291	0.0268	0.0356	0.0329	0.0314	0.0295
	Bias	0.0413	0.0273	0.0113	**0.0064**	0.0438	0.0296	0.0118	0.0077
k_2	Mean	0.0988	0.0992	0.0997	0.0997	0.1019	0.1020	0.1011	0.1015
	STD	0.0308	0.0308	0.0312	0.0309	0.0297	0.0297	0.0297	0.0297
	Bias	0.0913	0.0917	0.0922	**0.0922**	0.0944	0.0945	0.0936	0.0940
k_3	Mean	0.6186	0.6187	0.6185	0.6185	0.6204	0.6203	0.6202	0.6201
	STD	0.0094	0.0094	0.0094	0.0097	0.0100	0.0100	0.0100	0.0100
	Bias	0.0078	0.0079	0.0077	**0.0077**	0.0096	0.0095	0.0094	0.0093
k_4	Mean	0.0053	-0.0049	-0.0081	-0.0059	0.0061	-0.0037	-0.0073	-0.0054
	STD	0.0147	0.0129	0.0118	0.0104	0.0146	0.0129	0.0119	0.0105
	Bias	0.0028	0.0074	0.0106	0.0084	0.0036	0.0062	0.0098	**0.0079**
k_5	Mean	0.0365	0.0367	0.0376	0.0373	0.0364	0.0365	0.0376	0.0373
	STD	0.0017	0.0016	0.0015	0.0014	0.0019	0.0018	0.0017	0.0016
	Bias	0.0006	0.0004	0.0005	0.0002	0.0007	0.0006	0.0005	0.0002
k_6	Mean	0.8013	0.8005	0.8009	0.8004	0.7996	0.7999	0.7999	0.7998
	STD	0.0213	0.0217	0.0221	0.0222	0.0201	0.0201	0.0200	0.0200
	Bias	0.0088	0.0096	0.0092	**0.0097**	0.0105	0.0102	0.0102	0.0103
k_7	Mean	0.0154	0.0131	0.0136	0.0106	0.0101	0.0057	0.0063	0.0031
	STD	0.0464	0.0447	0.0432	0.0417	0.0487	0.0477	0.0467	0.0446
	Bias	0.0559	0.0582	0.0577	**0.0607**	0.0612	0.0656	0.0650	0.0682
k_8	Mean	0.0787	0.0826	0.0827	0.0822	0.0819	0.0869	0.0864	0.0895
	STD	0.0342	0.0288	0.0259	0.0210	0.0349	0.0298	0.0265	0.0220
	Bias	0.0100	0.0139	0.0140	**0.0135**	0.0132	0.0182	0.0177	0.0208
k_9	Mean	0.9966	0.9915	0.9534	0.9586	1.0060	0.9947	0.9585	0.9638
	STD	0.0763	0.0737	0.0693	0.0646	0.0735	0.0701	0.0683	0.0654
	Bias	0.0366	0.0315	0.0066	**0.0014**	0.0460	0.0347	0.0015	0.0038
k_{10}	Mean	0.0002	0.0001	0.0005	0.0005	-0.0006	-0.0005	-0.0003	-0.0003
	STD	0.0052	0.0053	0.0053	0.0053	0.0050	0.0050	0.0050	0.0050
	Bias	0.0010	0.0011	0.0007	**0.0007**	0.0018	0.0017	0.0015	0.0015
k_{11}	Mean	0.8859	0.8819	0.8802	0.8751	0.8872	0.8803	0.8811	0.8762
	STD	0.0367	0.0363	0.0357	0.0360	0.0406	0.0404	0.0403	0.0400
	Bias	0.0139	0.0099	0.0082	**0.0041**	0.0152	0.0083	0.0091	0.0052

ods developed for the dynamic state estimation problems in the data assimilation community, e.g., [29]. We plan to investigate these issues in the future.

References

1. JD Annan and JC Hargreaves. Efficient parameter estimation for a highly chaotic system. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):520–526, 2004.
2. JD Annan, DJ Lunt, JC Hargreaves, and PJ Valdes. Parameter estimation in an atmospheric gcm using the ensemble kalman filter. In *Nonlinear Processes in Geophysics, European Geosciences Union (EGU),*, volume 12, pages 363–371, 2005.
3. Amit Apte, Martin Hairer, AM Stuart, and Jochen Voss. Sampling the posterior: An approach to non-gaussian data assimilation. *Physica D: Nonlinear Phenomena*, 230(1-2):50–64, 2007.
4. Andrea Arnold, Daniela Calvetti, and Erkki Somersalo. Parameter estimation for stiff deterministic dynamical systems via ensemble kalman filter. *Inverse Problems*, 30(10):105008, 2014.

5. M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.
6. Matthew J Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, UCL (University College London), 2003.
7. Alexandros Beskos, Ajay Jasra, Kody Law, Raul Tempone, and Yan Zhou. Multilevel sequential monte carlo samplers. *Stochastic Processes and their Applications*, 127(5):1417–1440, 2017.
8. Nicolas Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.
9. Nicolas Chopin and Omiros Papaspiliopoulos. An introduction to sequential monte carlo, 2020.
10. Chenguang Dai, Jeremy Heng, Pierre E Jacob, and Nick Whiteley. An invitation to sequential monte carlo samplers. *arXiv preprint arXiv:2007.11936*, 2020.
11. Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.
12. Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.
13. Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.
14. Richard G Everitt, Richard Culliford, Felipe Medina-Aguayo, and Daniel J Wilson. Sequential monte carlo with transformations. *Statistics and computing*, 30(3):663–676, 2020.
15. Paul Fearnhead, Benjamin M Taylor, et al. An adaptive sequential monte carlo sampler. *Bayesian analysis*, 8(2):411–438, 2013.
16. Andrew Gelman, John B Carlin, Hal S Stern, David Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis (3rd edition)*. Chapman & Hall/CRC, 2013.
17. Charles J Geyer. Importance sampling, simulated tempering, and umbrella sampling. In *Handbook of Markov Chain Monte Carlo*, pages 321–338. Chapman and Hall/CRC, 2011.
18. Walter R Gilks and Carlo Berzuini. Following a moving target—monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001.
19. Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.
20. Andrew Golightly and Darren J Wilkinson. Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo. *Interface focus*, 1(6):807–820, 2011.
21. PL Green and Simon Maskell. Estimating the parameters of dynamical systems from big data using sequential monte carlo samplers. *Mechanical Systems and Signal Processing*, 93:379–396, 2017.
22. Jeremy Heng, Adrian N Bishop, George Deligiannidis, Arnaud Doucet, et al. Controlled sequential monte carlo. *Annals of Statistics*, 48(5):2904–2929, 2020.
23. Ibrahim Hoteit, Dinh-Tuan Pham, George Triantafyllou, and Gerasimos Korres. A new approximate solution of the optimal nonlinear filter for data assimilation in meteorology and oceanography. *Monthly Weather Review*, 136(1):317–334, 2008.
24. Marco A Iglesias. A regularizing iterative ensemble kalman method for pde-constrained inverse problems. *Inverse Problems*, 32(2):025002, 2016.
25. Marco A Iglesias, Kody JH Law, and Andrew M Stuart. Ensemble kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2013.
26. Cho Kwang-Hyun, Shin Sung-Young, Kim Hyun-Woo, Olaf Wolkenhauer, Brian McFerran, and Walter Kolch. Mathematical modeling of the influence of rkip on the erk signaling pathway. In *International Conference on Computational Methods in Systems Biology*, pages 127–141. Springer, 2003.
27. Siamak Mehrkanoon, Tillmann Falck, and Johan Suykens. Parameter estimation for time varying dynamical systems using least squares support vector machines. In *Proc. of the 16th IFAC Symposium on System Identification (SYSID 2012)*, volume 16, pages 1300–1305, 2012.
28. Thomas Peter Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

29. Oliver Pajonk, Bojana V Rosić, Alexander Litvinenko, and Hermann G Matthies. A deterministic filter for non-gaussian bayesian estimation—applications to dynamical system estimation with noisy measurements. *Physica D: Nonlinear Phenomena*, 241(7):775–788, 2012.
30. Nicolas Papadakis, Étienne Mémin, Anne Cuzol, and Nicolas Gengembre. Data assimilation with the weighted ensemble kalman filter. *Tellus A: Dynamic Meteorology and Oceanography*, 62(5):673–697, 2010.
31. Minh Quach, Nicolas Brunel, and Florence d'Alché Buc. Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. *Bioinformatics*, 23(23):3209–3216, 2007.
32. Claudia Schillings and Andrew M Stuart. Analysis of the ensemble kalman filter for inverse problems. *SIAM Journal on Numerical Analysis*, 55(3):1264–1290, 2017.
33. Andreas S Stordal, Hans A Karlsen, Geir Nævdal, Hans J Skaug, and Brice Vallès. Bridging the ensemble kalman filter and particle filters: the adaptive gaussian mixture filter. *Computational Geosciences*, 15(2):293–305, 2011.
34. Xiaodian Sun, Li Jin, and Momiao Xiong. Extended kalman filter for estimation of parameters in nonlinear state-space models of biochemical networks. *PloS one*, 3(11):e3758, 2008.
35. Linjie Wen, Jiangqi Wu, Linjun Lu, and Jinglai Li. A defensive marginal particle filtering method for data assimilation. *SIAM/ASA Journal on Uncertainty Quantification*, 8(3):1215–1235, 2020.
36. Yan Zhou, Adam M Johansen, and John AD Aston. Toward automatic model comparison: an adaptive sequential monte carlo approach. *Journal of Computational and Graphical Statistics*, 25(3):701–726, 2016.