2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Effective filtering approach for joint parameter-state estimation in SDEs via Rao-Blackwellization and modularization

Zhou Fang, Ankit Gupta, and Mustafa Khammash

Abstract— **For joint parameter-state estimation problems, classical particle filters often suffer from sample degeneracy or problems related to artificial noise, stemming from the particles representing unknown parameters. To address this challenge, this paper provides a novel filtering approach that avoids generating such particles by utilizing Rao-Blackwellization, which therefore provides more accurate estimates. Moreover, our method employs a modularization approach when integrating out the parameters, significantly reducing the computational complexity. All these designs ensure the superior performance of our method. Finally, a couple of numerical examples are presented to illustrate the superior performance of our method compared with several existing approaches.**

I. INTRODUCTION

Stochastic filtering is a popular topic in the control community, which aims to infer hidden system states and parameters from partial, noisy observations. A good solution to this problem can provide deep insights into the considered system and enable better feedback control. In nonlinear and non-Gaussian cases, the bootstrap particle filter (BPF) [1] is one popular approach for the filtering problem. This method employs the simulation-based Monte-Carlo technique to recursively approximate conditional distributions, and its reliability is guaranteed by its convergence to the exact filtering solution [2]. This method has been successfully used in many applications, including wireless communications [3] and biological studies [4], [5], [6], [7], etc.

Despite its successes, the BPF performs poorly in estimating static variables, e.g., system parameters. The problem is attributed to the resampling step, which repeatedly reduces the number of distinct particles representing unknown parameters. Since the BPF does not increase this number in other steps, these particles soon become identical after a few initial iterations. This sample degeneracy diminishes the effectiveness of the Monte-Carlo technique (which relies on a population of distinct particles) and can provide highly inaccurate estimates [8].

To mitigate sample degeneracy, researchers have developed some improved methods that introduce additional (artificial) noise to perturb the particles, thereby increasing the number of distinct particles. Such methods include the resample-move method [9], [10] and regularized particle filter (RPF) [11], [12], [13]. This idea also appears in the nested particle filter (NPF) [14], which operates in two layers: the first layer estimates the system parameters using an RPF, and the second layer infers the state variables using a BPF. Some improvements to this nested approach were reported in [15], [16]. Nevertheless, the use of artificial noise (in all the above methods) "throws away" the time-invariant property of parameters, as it assumes parameters to be time-varying [11]. When the particle size is finite, this can result in highly inaccurate estimates, especially if the intensity of the artificial noise is not properly chosen [13]. Consequently, introducing artificial noise might not be the optimal solution to tackle sample degeneracy.

Both sample degeneracy and the problem related to artificial noise stem from the particles representing static variables. From this perspective, a method that avoids generating such particles can effectively tackle these issues. This aligns with the Rao-Blackwellized particle filter (RB-PF) [17], [18], which reduces the filter's dependence on Monte Carlo techniques by integrating out some variables using a finite-dimensional filter. When the integrated-out variables include the unknown parameters, this method has the potential to effectively mitigate sample degeneracy and the problem related to artificial noise. The RB-PF has been theoretically shown to have superior performance compared to the BPF in terms of accuracy under quite general conditions [17], [19]. The RB-PF has been successfully applied to joint parameter-state estimation in biochemical reaction systems described by continuous-time Markov Chains (CTMCs) [20], [21].

Despite its success with CTMCs, the idea of Rao-Blackwellization has not been sufficiently explored for the joint state-parameter estimation in SDEs. This paper is devoted to filling this gap by providing a principled method based on Rao-Blackwellization and modularization. Our approach adopts the strategy of integrating out unknown parameters by applying a finite-dimensional filter to the Zakai equation. This strategy circumvents the need to generate particles representing parameters, thereby effectively mitigating the problems related to sample degeneracy and artificial noise. Utilizing conditional independence, our method employs a modularization approach to solve the Zakai equation for each parameter separately, significantly reducing the computational cost. These designs result in an effective method for joint parameter-state estimation in SDEs.

The rest of the paper is organized as follows. Section II first introduces the mathematical problem of joint parameter-state estimation in SDEs. In Section III, we introduce our novel approach to this filtering problem. Numerical examples are presented in Section IV to demonstrate our method. Finally, Section V concludes this paper.

This work was funded by the Swiss National Science Foundation under grant number 182653. Zhou Fang, Ankit Gupta, and Mustafa Khammash are with the Department of Biosystems Science and Engineering at ETH-Zrich. zhou.fang@bsse.ethz.ch, ankit.gupta@bsse.ethz.ch, mustafa.khammash@bsse.ethz.ch

II. Stochastic filtering for SDEs

We consider stochastic differential systems expressed as:

$$\mathrm{d}X_i(t) = f_i(\Theta_i, X(t))\mathrm{d}t + \sigma_i \mathrm{d}B_i(t), \quad i = 1, \dots, n \quad (1)$$

where $X(t)$ is the n-dimensional state vector, $X_i(t)$ is its i-th component, $\{f_1, \dots, f_n\}$ are measurable functions, $\{\Theta_1, \dots, \Theta_n\}$ are n unknown system parameters, $\{B_1(t), \dots, B_n(t)\}$ are independent standard Brownian motions, and $\{\sigma_1, \dots, \sigma_n\}$ represent the noise intensities which we assume to be known. In many real-world problems, such a system is measured at consecutive time points $\{t_1, t_2, \dots\}$ with corresponding measurements $Y(t_k)$ $(k = 1, 2, \dots)$. We assume that these measurements satisfy

$$Y(t_k) = h(X(t_k)) + \Sigma W(t_k), \qquad k = 1, 2, \dots,$$

where $h(\cdot)$ is a vector-valued function, $\{W(t_1), W(t_2), \dots\}$ are vectors of independent standard normal random variables, and Σ is the covariance matrix of the observation noise.

The considered system (1) requires the parameters (Θ_i) and noise (i.e., $B_i(t)$) immediately influencing different state variables be independent. This is not too restrictive, as many practical systems either fall into this framework or can be transformed into an equivalent system of this form. Specifically, when $X_i(t)$ (for $i = 1, \dots, n$) models the effect of each driving force (e.g., the cumulative firings of each chemical reaction or the work done by each physical force), this requirement naturally holds, as different driving forces are often governed by distinct parameters and noise sources.

To estimate hidden system state and parameters, stochastic filtering aims to compute the conditional distribution: $\pi_{t_k}(\mathrm{d}\theta, \mathrm{d}x) \triangleq \mathbb{P}\left(\Theta \in \mathrm{d}\theta, X(t_k) \in \mathrm{d}x \middle| Y(t_1), \dots Y(t_k)\right)$ for $k = 1, 2, \dots$. Let us denote the initial distribution as $\pi_{t_0}(\mathrm{d}\theta, \mathrm{d}x) \triangleq \mathbb{P}\left(\Theta \in \mathrm{d}\theta, X(0) \in \mathrm{d}x\right)$, which is assumed to be uniform, and define $\rho_{t_{k+1}}(\mathrm{d}\theta, \mathrm{d}x) \triangleq \mathbb{P}\left(\Theta \in \mathrm{d}\theta, X(t_{k+1}) \in \mathrm{d}x \middle| Y(t_1), \dots Y(t_k)\right)$. Then by Bayes' rule, the target distribution $\pi_{t_k}(\mathrm{d}\theta, \mathrm{d}x)$ satisfies [2]:

$$\rho_{t_{k+1}}(\mathrm{d}\theta, \mathrm{d}x) \quad (2)$$
$$= \int_{x'} \mathbb{P}\left(X(t_{k+1}) \in \mathrm{d}x | \Theta = \theta, X(t_k) = x'\right) \pi_{t_k}(\mathrm{d}\theta, \mathrm{d}x')$$
$$\pi_{t_{k+1}}(\mathrm{d}\theta, \mathrm{d}x) \propto L\left(Y(t_{k+1}) \middle| x\right) \rho_{t_{k+1}}(\mathrm{d}\theta, \mathrm{d}x) \quad (3)$$

for $k = 0, 1, \dots$, where $t_0 = 0$, and $L(y|x)$ is the likelihood function for the observation given the system state. We can interpret (2) as the prediction step, which forecasts $X(t_{k+1})$ and Θ using the observations up to time t_k; Eq. (3) can be interpreted as the correction step, which adjusts the predicted distribution based on the new measurement collected at t_{k+1}.

A. Existing filtering methods

Eq. (2) and (3) often cannot be solved explicitly. Thus, many Monte-Carlo based methods were proposed for them.

1) Bootstrap particle filter (BPF) [1]: The bootstrap particle filter (BPF) solves (2) and (3) by Monte-Carlo samples together with a resampling scheme (see Algorithm 1). Initially, the algorithm samples N particles from the initial distribution (Line 1, Algorithm 1). For each t_k, the algorithm simulates the particles from time t_k to t_{k+1} according to the dynamical equation (Line 3, Algorithm 1). Then, the empirical distribution of the particles $(\theta_1, \mathtt{x}_1(t_{k+1})), \dots, (\theta_N, \mathtt{x}_N(t_{k+1}))$ approximates the predicting distribution $\rho_{t_{k+1}}(\cdot, \cdot)$. Next, the BPF computes particle weights according to the measurement $Y(t_{k+1})$ (Line 4, Algorithm 1) and uses the empirical distribution of the weighted particles to approximate the conditional distribution $\pi_{t_{k+1}}(\cdot, \cdot)$ (Line 5, Algorithm 1). Finally, the algorithm resamples particles to remove non-important samples and reproduce important particles so that the computational complexity is reduced [8]. The convergence of the BPF to the exact filter (as $N \to \infty$) has been shown in [2].

Algorithm 1: Bootstrap particle filter ([1], [8])

1 Sample N particles $(\theta_1, \mathtt{x}_1(0)), \dots, (\theta_N, \mathtt{x}_N(0))$ from the initial distribution π_{t_0};
2 **for** *each time point* t_k $(k \in \mathbb{Z}_{\geq 0})$ **do**
3 Simulate each $(\theta_j, \mathtt{x}_j(\cdot))$ from t_k to t_{k+1} by (1);
4 Compute weights $\mathtt{w}_j = L\left(Y(t_{k+1}) \middle| \mathtt{x}_j(t_{k+1})\right)$;
5 Filter: $\bar{\pi}_{t_{k+1}}(\theta, x) = \frac{\sum_{j=1}^{N} \mathtt{w}_j \mathbb{1}(\theta_j = \theta, \mathtt{x}_j(t_{k+1}) = x)}{\sum_j w_j}$;
6 Resample $\{\mathtt{w}_j, (\theta_j, \mathtt{x}_j(\cdot))\}_j$ to obtain N equally weighted particles to replace the old ones;
7 **end**

2) Regularized particle filter (RPF): When the particle size is finite, the performance of the BPF in estimating system parameters is poor due to sample degeneracy. Specifically, the resampling step reduces the number of distinct particles $\theta_1, \dots, \theta_N$ used for estimating parameters. After several iterations, the BPF often ends up with particles sharing the same parameter part θ_j. This shared θ_j is not necessarily equal to the true parameter values. Often, random effects can cause this shared θ_j to deviate significantly from the true parameter values, thereby greatly affecting the accuracy of the method. A more effective alternative to the BPF is the regularized particle filter (RPF) [11], [22], [13], [12], which introduces some artificial noise to θ_j (in each iteration in the BPF) to ensure greater diversity among the particles $\theta_1, \dots, \theta_N$. The RPF has demonstrated excellent performance in numerous applications (as shown in the aforementioned references), and it also converges to the exact filtering result as $N \to \infty$ under some mild conditions, [13], [23], [24], [25].

3) Nested particle filter (NPF): The literature [14] introduced the nest particle filter (NPF) for joint parameter-state estimation. The first layer of the NPF employs a regularized particle filter (RPF) to infer the parameters, i.e., targeting $\pi_{t_k}(\mathrm{d}\theta) \triangleq \mathbb{P}\left(\Theta \in \mathrm{d}\theta \middle| Y(t_1), \dots Y(t_k)\right)$. The second layer uses the BPF to estimate the state variables given a fixed θ, i.e., targeting $\pi_{t_k}(\mathrm{d}x|\theta) \triangleq \mathbb{P}\left(X(t_k) \in \mathrm{d}x \middle| Y(t_1), \dots Y(t_k), \Theta = \theta\right)$. The two layers are wisely integrated, enabling the NPF to operate in a recursive manner. Finally, the NPF gives an approximated solution to the filtering problem by combining the results in both layers

according to $\pi_{t_k}(\mathrm{d}\theta, \mathrm{d}x) = \pi_{t_k}(\mathrm{d}x|\theta)\pi_{t_k}(\mathrm{d}\theta)$. Its validity in the limit of large particles has been shown in [14].

4) Further remarks: The performance of RPFs and NPFs largely depends on the choice of the artificial noise intensity, which is not easy to determine in advance. Weak artificial noise cannot effectively circumvent sample degeneracy; strong artificial noise could result in a big bias [11]. Our previous study [13] demonstrates that in some examples, the RPF requires a training step in advance to find the optimal noise intensity, which can be extremely time-consuming. In summary, employing artificial noise to perturb particles may not be optimal for joint parameter-state estimation.

III. Filtering approach based on Rao-Blackwellization and modularization

We propose a new filtering method for joint parameter-state estimation based on Rao-Blackwellization and modularization. The key idea is to marginalize out the parameters Θ using an efficient finite-dimensional filter. This strategy circumvents the need for generating particles representing Θ, thereby mitigating the related issues. Moreover, when integrating out Θ, our method employs a divide-and-conquer approach to reduce the computational complexity.

A. Derivation of our Rao-Blackwellized particle filter

First, we give a new formula to re-express the filter π_{t_k}. Let $X_{0:t}$ be the whole trajectory of $X(\cdot)$ from time zero to t, $Y_{t_1:t_k}$ the measurements up to t_k, and $\pi_{\Theta|X}(t, \mathrm{d}\theta) = \mathbb{P}\left(\Theta \in \mathrm{d}\theta \middle| X_{0:t}\right)$. Then, by the tower property, we have

$$\begin{aligned}\pi_{t_k}(\mathrm{d}\theta, \mathrm{d}x) &= \mathbb{E}\left[\mathbb{P}\left(\Theta \in \mathrm{d}\theta \middle| X_{0:t_k}, Y_{t_1:t_k}\right) \mathbb{1}\left(X(t_k) \in \mathrm{d}x\right) \middle| Y_{t_1:t_k}\right] \\ &= \mathbb{E}\left[\pi_{\Theta|X}(t_k, \mathrm{d}\theta)\, \mathbb{1}\left(X(t_k) \in \mathrm{d}x\right) \middle| Y_{t_1:t_k}\right] \end{aligned} \tag{4}$$

where the second line holds because $L(y|x)$ does not depend on the parameters in our problem. Similarly, $\rho_{t_{k+1}}$ satisfies

$$\rho_{t_{k+1}}(\mathrm{d}\theta, \mathrm{d}x) = \mathbb{E}\left[\pi_{\Theta|X}(t_{k+1}, \mathrm{d}\theta)\, \mathbb{1}\left(X(t_{k+1}) \in \mathrm{d}x\right) \middle| Y_{t_1:t_k}\right]. \tag{5}$$

Eq. (4) and (5) suggest that the filtering problem can be numerically solved by generating samples for the process $\left(\pi_{\Theta|X}(t, \cdot), X(t)\right)$. This scheme avoids generating particles that represent parameters Θ.

Next, we introduce a divide-and-conquer approach to compute $\pi_{\Theta|X}(t, \cdot)$. When the terms in the dynamical equation (1) are regular enough, the density of $\pi_{\Theta|X}(t, \cdot)$ is the unique normalized solution of the Zakai equation [26]

$$\begin{cases} \mathrm{d}\rho_{\Theta|X}(t, \theta) = \rho_{\Theta|X}(t, \theta) \sum_{i=1}^{n} \frac{f_i(\theta_i, X(t))}{\sigma_i^2} \mathrm{d}X_i(t) \\ \rho_{\Theta|X}(0, \theta)\mathrm{d}\theta = \pi_{\Theta|X}(0, \mathrm{d}\theta) \end{cases}.$$

Basically, this Zakai equation is high dimensional, and it suffers the curse of dimensionality when solved directly using grid-based methods. Fortunately, our system (1) has a nice structure where the i-th parameter only immediately affects the i-th state. This enables a modularization method for computing the Zakai equation. Specifically, this Zakai equation has an explicit solution [27]

$$\begin{aligned}\rho_{\Theta|X}(t, \theta) = \ & \rho_{\Theta|X}(0, \theta) \times \\ & \prod_{i=1}^{n} \exp\left\{\int_0^t \frac{f_i(\theta_i, X(s))}{\sigma_i^2} \left[\mathrm{d}X_i(s) - \frac{f_i(\theta_i, X(s))}{2}\mathrm{d}s\right]\right\}.\end{aligned}$$

Recall that we assume a uniform prior distribution for Θ and $X(0)$. This means that $\rho_{\Theta|X}(0, \theta) = \prod_{i=1}^{n} \rho_{\Theta_i}(\theta_i)$ where $\rho_{\Theta_i}(\cdot)$ is the marginal distribution for Θ_i. Thus, the solution of the Zakai equation can be expressed by

$$\begin{aligned}&\rho_{\Theta|X}(t, \theta) = \\ &\prod_{i=1}^{n} \underbrace{\rho_{\Theta_i}(\theta_i) \exp\left\{\int_0^t \frac{f_i(\theta_i, X(s))}{\sigma_i^2} \left[\mathrm{d}X_i(s) - \frac{f_i(\theta_i, X(s))}{2}\mathrm{d}s\right]\right\}}_{=:\ \rho_{\Theta_i|X}(t, \theta)}\end{aligned} \tag{6}$$

indicating that the parameters are conditionally independent given $X_{0:t}$. Here, we denote $\rho_{\Theta_i|X}(t, \cdot)$ as the unnormalized marginal conditional distribution for Θ_i. With this conditional independence, we can compute $\pi_{\Theta|X}(t, \cdot)$ (or equivalently, $\rho_{\Theta|X}(t, \cdot)$) by applying the Euler-Maruyama method to each marginal distribution $\rho_{\Theta_i|X}(t, \cdot)$ rather than the joint distribution. This strategy reduces the computational complexity from $O\left(\mathbb{G}^n\right)$ to $O\left(n\mathbb{G}\right)$, with G the number of grid points for each parameter. Consequently, this divide-and-conquer method, as suggested by (6), is efficient even in high-dimensional cases.

B. Algorithm of our Rao-Blackwellized particle filter

Following the above idea, we provide a Rao-Blackwellized particle filter (RB-PF) in Algorithm 2. Essentially, the method generates samples for the processes $\left(X(\cdot), \rho_{\Theta_1|X}, \ldots, \rho_{\Theta_n|X}\right)$ and approximates $\pi_{t_k}(\cdot)$ by (4).

First, the algorithm generates particle $\mathtt{x}_1(0), \ldots, \mathtt{x}_N(0)$ from the initial distribution (Line 1, Algorithm 2). Then, for each $\mathtt{x}_j(0)$, the algorithm creates $\bar{\rho}_j^i(0, \cdot)$ to represent the conditional density of Θ_i given $X(0) = \mathtt{x}_j(0)$ (Line 2, Algorithm 2). Due to memory constraints, a digital computer cannot store all the values of this conditional density function. Consequently, for each Θ_i, the algorithm only stores the values of the density function corresponding to some representative points selected within $\mathbb{R}$.

Then, at each time point t_k, Algorithm 2 solves the prediction and correct equations (2) and (3) by simulating the particles according to (1) and (6). The algorithm simulates every $\mathtt{x}_j(\cdot)$ from time t_k to t_{k+1} according to the dynamics (1) with parameters sampled from the conditional distribution $\bar{\rho}_j(t_k, \theta) \triangleq \prod_{i=1}^{n} \bar{\rho}_j^i(t_k, \theta_i)$ (Line 4, Algorithm 2). For each $\mathtt{x}_j(\cdot)$, the algorithm then computes the marginal conditional densities $\bar{\rho}_j^i(t_{k+1}, \cdot)$ according to (6) (Line 5, Algorithm 2). By (5), the predicting distribution $\rho_{t_{k+1}}$ can be approximated by $\bar{\rho}_{t_{k+1}}(\theta, x) = \frac{\sum_{j=1}^{N}\left[\mathbb{1}(\mathtt{x}_j(t_{k+1}) = x) \prod_{i=1}^{n} \bar{\rho}_j^i(t_{k+1}, \theta_i)\right]}{N}$. To approximate the filter $\pi_{t_{k+1}}(\theta, x)$, our algorithm computes weights for all the particles (Line 6, Algorithm 2) and provides a filter $\bar{\pi}_{t_{k+1}}(\theta, x)$ according to (3) and (4) (Line 7, Algorithm 2). Finally, our algorithm resamples the particles to accelerate the speed.

Algorithm 2: Rao-Blackwellized particle filter

Sample N particles $\mathtt{x}_1(0), \ldots, \mathtt{x}_N(0)$ from π_{t_0} ;

For every Θ_i, select a finite set $\mathbb{O}_i \subset \mathbb{R}$. Then, for each $\mathtt{x}_j(0)$ and each i, denote marginal distribution $\bar{\rho}_j^i(0, \theta_i) \propto p_{\Theta_i}(\theta_i)$, $\forall \theta_i \in \mathbb{O}_i$;

for *each time point* t_k $(k \in \mathbb{Z}_{\geq 0})$ **do**

  Simulate each $\mathtt{x}_j(\cdot)$ from t_k to t_{k+1} by (1) with parameters θ_j sampled from the distribution $\bar{\rho}_j(t_k, \theta) \triangleq \prod_{i=1}^n \bar{\rho}_j^i(t_k, \theta_i)$;

  For each $\mathtt{x}_j(\cdot)$ and $i \in \{1, \ldots, n\}$, compute

$$\bar{\rho}_j^i(t_{k+1}, \theta_i) = \bar{\rho}_j^i(t_k, \theta_i) \mathrm{e}^{\int_{t_k}^{t_{k+1}} \frac{f_i(\theta_i, \mathtt{x}_j(s))}{\sigma_i^2} \left[\mathrm{d}(\mathtt{x}_j)_i(s) - \frac{f_i(\theta_i, \mathtt{x}_j(s))}{2} \mathrm{d}s \right]}$$

  for every $\theta_i \in \mathbb{O}$, and then normalize it;

  Compute weights $\mathtt{w}_j = L\left(Y(t_{k+1}) \middle| \mathtt{x}_j(t_{k+1})\right)$;

  Approximated filter:

$$\bar{\pi}_{t_{k+1}}(\theta, x) = \frac{\sum_{j=1}^N \left[\mathtt{w}_j \mathbb{1}(\mathtt{x}_j(t_{k+1}) = x) \prod_{i=1}^n \bar{\rho}_j^i(t_{k+1}, \theta_i) \right]}{\sum_j w_j}$$

  Reample $\left\{ \mathtt{w}_j, \left(\mathtt{x}_j, \bar{\rho}_j^1, \ldots, \bar{\rho}_j^n \right) \right\}_j$ to obtain N new particles to replace the old ones;

end

C. Some discussions about our method

Our RB-PF also has a nested structure. Concretely, the particles $\mathtt{x}_1(\cdot), \ldots, \mathtt{x}_N(\cdots)$ can be seen as the first layer estimating the system state, and $\left\{ \left(\bar{\rho}_j^1, \ldots, \rho_j^n \right) \right\}_{j=1,\ldots,N}$ form the second layer estimating the parameters (given the state trajectories). Compared with the NPF, our filter reversed its order of layers for estimating the parameters and states. More importantly, the second layer in our algorithm uses a finite-dimensional filter rather than a BPF or RPF for parameter estimation. This strategy circumvents the need to generate particles representing Θ, thereby mitigating issues related to sample degeneracy and artificial noise.

Our algorithm improves classical particle filtering methods (e.g., the BPF, RPF, and NPF) at the cost of requiring more computational resources for the same particle size. Essentially, our RB-PF needs to additionally compute and store the distributions $\bar{\rho}_j^i(t, \cdot)$, which necessitates more computational time and computer memory. We provided a modularization-based approach for these conditional distributions, thereby reducing the additional computational costs to some extent. Still, the extra computational effort is not negligible. Nonetheless, this additional computation is worthwhile. It can effectively mitigate the problems associated with RPF and NPF, and, consequently, our method can outperform existing methods within the same computational time.

IV. Numerical Examples

We show some examples to demonstrate our approach. All the algorithms were performed on the Euler Computing Cluster at ETH Zurich, using a node with 12-core CPUs.

A. Stochastic Lorenz-63 model

Here, we consider a stochastic Lorenz-63 model [28]. The model consists of three states $(X_1(t), X_2(t), X_3(t))$ and three parameters $(\Theta_1, \Theta_2, \Theta_3)$. Its dynamics is described by

$$\begin{aligned}
\mathrm{d}X_1(t) &= -\Theta_1 \left[X_1(t) - X_2(t) \right] \mathrm{d}t + \mathrm{d}B_1(t) \\
\mathrm{d}X_2(t) &= \left[\Theta_2 X_1(t) - X_2(t) - X_1(t) X_3(t) \right] \mathrm{d}t + \mathrm{d}B_2(t) \\
\mathrm{d}X_3(t) &= \left[X_1(t) X_2(t) - \Theta_3 X_3(t) \right] \mathrm{d}t + \mathrm{d}B_3(t).
\end{aligned}$$

Clearly, this system conforms to the model (1). Moreover, we assume that the process are measured at time points $\{0.05, 0.10, \ldots, 10\}$, and the measurements $Y(t_k)$ satisfies

$$Y(t_k) = \begin{bmatrix} X_1(t_k) \\ X_3(t_k) \end{bmatrix} + \begin{bmatrix} W_1(t_k) \\ W_2(t_k) \end{bmatrix} \quad \text{for } t_k = 0.05, \ldots.$$

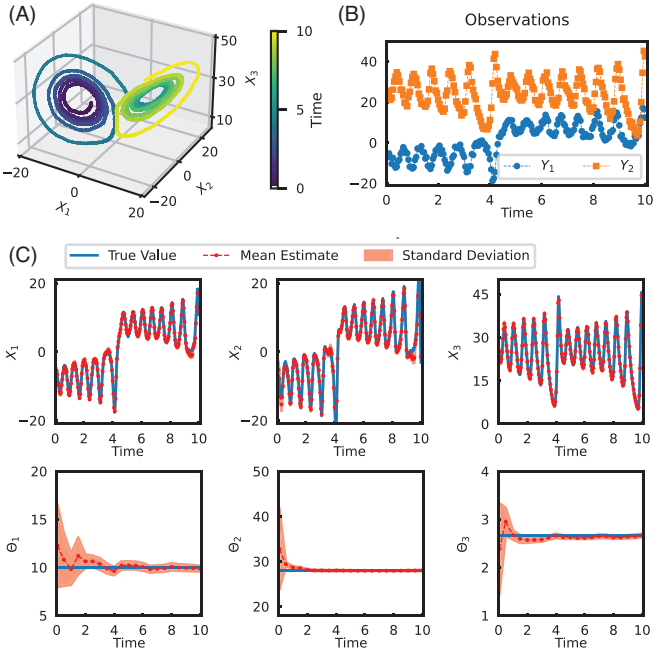

Fig. 1. Results of our RB-PF applied to the stochastic Lorenz-63 system. (A) Simulated trajectory of the system. (B) Observation process. (C) Estimation results.

First, we examined the performance of our RB-PF in solving the filtering problem. We assumed that the initial state and parameters were independent and satisfied uniform distributions: $X_1(0) \sim \mathcal{U}(-9, -3)$, $X_2(0) \sim \mathcal{U}(-9, -3)$, $X_3(0) \sim \mathcal{U}(20, 28)$, $\Theta_1 \sim \mathcal{U}(5, 20)$, $\Theta_2 \sim \mathcal{U}(18, 50)$, and $\Theta_3 \sim \mathcal{U}(1, 8)$. Then, we simulated a trajectory of the system with the initial state $(-6, -5, 24.5)^\top$ and parameters $(10, 28, 8/3)$. Finally, we applied our RB-PF (with a particle size of $N = 20,000$) to solve this filtering problem. The whole computational time was approximately 55 minutes. The numerical result in Figure 1 shows that our approach provides sharp estimates for the system states and parameters. The mean estimates of the state variables almost overlap with the true state trajectories, with the standard deviations too small to be visible (Figure 1.C). The mean estimates for the system parameters also fast converge to the true values (Figure 1.C). From time four onward, these estimates closely

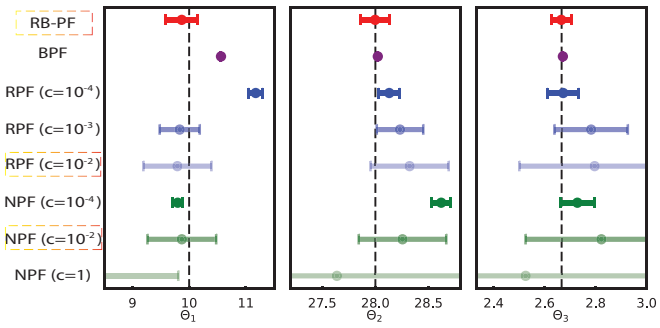

Fig. 2. Comparison of different filters in parameter estimation for the stochastic Lorenz-63 system. The black dashed lines represent the true parameter values. The colored lines present the estimates from different filters, with the dots indicating the mean estimates and the error bars representing the standard deviation. The names of the particle filters are listed on the left, where c denotes the artificial noise intensity.

match the true parameter value, with very small standard deviations. These results demonstrate the accuracy of our method in this filtering problem.

Next, we compared our RB-PF with other approaches (the BPF, RPF, and NPF). We applied these filters to the same observation trajectory and carefully selected their particle sizes so that their computational time was similar to that of the RB-PF ($\approx$ 55 minutes). The sample sizes of the BPF and RPF were set to be 40,000. For the NPF, the particle sizes in the first and second layers were both set to be 200. The artificial noise in the RPF and NPF was generated from a normal distribution with mean $(0,0,0)^\top$ and covariance matrix cI_d. Here, c is a tunable hyper-parameter, and I_d is the identity matrix.

The results in Figure 2 show that the RB-PF is the most accurate in estimating system parameters. First, the RB-PF is one of the three filters where all the confidence intervals (mean $\pm$ one standard deviation) include their respective true parameter values. Moreover, among these three filters, the RB-PF has the smallest standard deviations.

Figure 2 reveals the great challenge of choosing a proper artificial noise intensity (c) for the RPF and NPF. When c is small, these filters still suffer sample degeneracy, resulting in very similar particles whose confidence interval does not necessarily cover the true parameter value (see the RPFs and NPFs with $c < 10^{-2}$). Particularly, when $c = 0$, the RPF degenerates to the BPF, resulting in particles with identical parameter parts. As c increases, the artificial noise can lead to a big error, causing a large standard deviation in parameter estimation (see the RPFs and NPFs with $c \geq 10^{-2}$).

Thanks to the divide-and-conquer strategy (as suggested by (6)), the additional computational resources required in RB-PF for the Zakai equation are acceptable. In this example, the computational time required by the additional computation is comparable to that spent on the remaining part of the algorithm. Consequently, for the same computational time, the RB-PF can still employ half of the sample size compared to the other competing methods, resulting in its superior performance in parameter estimation.

B. Stochastic Lotka-Volterra dynamics

Next, we consider the Lotka-Volterra system [29], [30],

$$\mathrm{d}X_1(t) = X_1(t)\Big(\Theta_1 - X_2(t)\Big)\mathrm{d}t + 0.1\mathrm{d}B_1(t)$$
$$\mathrm{d}X_2(t) = X_2(t)\Big(\Theta_2 X_1(t) - 1\Big)\mathrm{d}t + 0.1\mathrm{d}B_2(t)$$

describing the interaction between a prey species (X_1) and a predator species (X_2). The parameters Θ_1 and $1/\Theta_2$ are the carrying capacities of the predator (X_2) and the prey (X_1), indicating the maximum sustainable populations of these species in this environment when randomness is absent. We assume the initial conditions $(X_1(0), X_2(0))$ and parameters (Θ_1, Θ_2) are independent and satisfy a uniform distribution $\mathcal{U}(0.1, 2)$. The measurements of the system have the form

$$Y(t_k) = X_2(t_k) + 0.1W(t_k) \quad \text{for } t_k = 0.5, 1, 1.5, \ldots, 40.$$

First, we examined the performance of our RB-PF in this example. We simulated a trajectory of the system (see Figure 3.A–B) and then applied our RB-PF with a particle size of $N = 10,000$. The numerical results shown in Figure 3 indicate that our method can accurately estimate the dynamics of both the prey and predator. Moreover, it can accurately estimate the carrying capacity of both species (i.e., Θ_1 and $1/\Theta_2$). Since the dynamics of the predator are measured in this example, the algorithm provides a more accurate estimate for the carrying capacity of the predator (i.e., Θ_1) than that of the prey (i.e., $1/\Theta_2$). In conclusion, our method is effective in solving this filtering problem.

We also compared our RB-PF with the BPF, RPF, and NPF. We applied these filters to the same observation trajectory and carefully selected their particle sizes so that their computational time was similar to that of the RB-PF (around 35 minutes). The sample sizes of the BPF and RPF were set to be 20,000. For the NPF, the particle sizes in the first and second layers were set to be 100 and 200. The comparison results in Figure 4 demonstrate that only three filters (including our RB-PF) can accurately estimate the parameters with all the confidence intervals (mean $\pm$

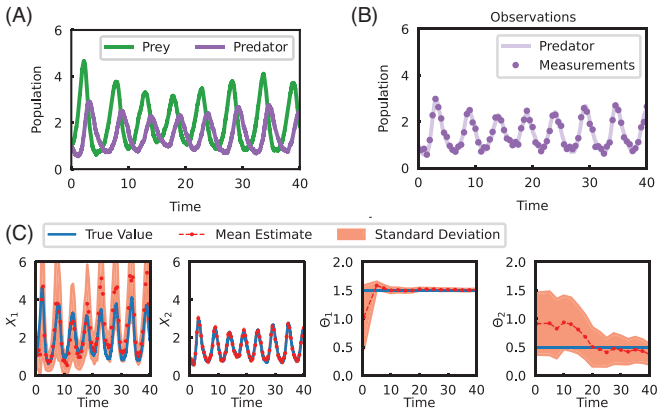

Fig. 3. Results of our RB-PF applied to the stochastic Lotka-Volterra system with the predator being measured. (A) Simulated trajectory of the system. (B) Measurements. (C) Estimation results. The standard deviation of X_2-estimates is too small to be visible.

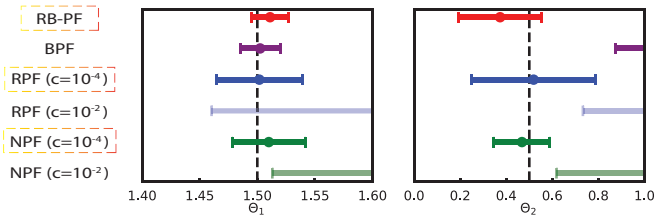

Fig. 4. Comparison of different filters in parameter estimation for the stochastic Lotka-Volterra system. All the elements, such as the dashed lines and colored error bars, retain the same meanings as described in Figure 2.

one standard deviation) encompassing their respective true parameter values. Although an RPF and an NPF are among these accurate filters, their effectiveness depends on a careful selection of the artificial noise intensity, which is challenging to determine in advance. In contrast, our method does not require fine-tuning of any such parameters. These results further indicate the superior performance of our approach.

V. Conclusion

This paper proposed a novel and effective filtering approach for joint parameter-state estimation in SDEs based on Rao-Blackwellization and modularization. Our method operates in two layers: the first layer estimates state variables using a BPF, and the second layer integrates out all the parameters using the Euler-Maruyama method. This strategy eliminates the need to generate particles representing parameters and, therefore, circumvents the problems related to sample degeneracy and artificial noise presented in the state-of-the-art methods. Moreover, our method employs a modularization approach in the second layer, which significantly reduces the required additional computational effort. These designs result in an effective filtering algorithm for joint parameter-state estimation in SDEs. Its superior performance was also demonstrated through several numerical examples.

A few topics deserve further investigation in future work. First, a theoretical analysis of this method is needed to investigate its superior performance and limitations. Second, the method can be further improved by integrating out some state variables in addition to the system parameters. Our previous work [21] can be beneficial for this extension, as it successfully employed this idea to CTMCs.

References

[1] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE proceedings F (radar and signal processing)*, vol. 140, pp. 107–113, IET, 1993.

[2] D. Crisan, "Particle filtersa theoretical perspective," in *Sequential Monte Carlo methods in practice*, pp. 17–41, Springer, 2001.

[3] P. Djurić, J. Zhang, T. Ghirmai, Y. Huang, and J. H. Kotecha, "Applications of particle filtering to communications: A review," in *2002 11th European Signal Processing Conference*, pp. 1–4, IEEE, 2002.

[4] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf, "Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems," *Journal of the Royal Society Interface*, vol. 6, no. 31, pp. 187–202, 2009.

[5] M. Rathinam and M. Yu, "State and parameter estimation from exact partial state observation in stochastic reaction networks," *The Journal of Chemical Physics*, vol. 154, no. 3, 2021.

[6] Z. Fang, A. Gupta, and M. Khammash, "Stochastic filters based on hybrid approximations of multiscale stochastic reaction networks," in *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 4616–4621, IEEE, 2020.

[7] Z. Fang, A. Gupta, and M. Khammash, "Stochastic filtering for multiscale stochastic reaction networks based on hybrid approximations," *Journal of Computational Physics*, vol. 467, p. 111441, 2022.

[8] A. Doucet, A. M. Johansen, *et al.*, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, vol. 12, no. 656-704, p. 3, 2009.

[9] C. Berzuini and W. Gilks, "Resample-move filtering with cross-model jumps," *Sequential Monte Carlo Methods in Practice*, pp. 117–138, 2001.

[10] W. R. Gilks and C. Berzuini, "Following a moving targetMonte Carlo inference for dynamic bayesian models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 1, pp. 127–146, 2001.

[11] J. Liu and M. West, "Combined parameter and state estimation in simulation-based filtering," in *Sequential Monte Carlo methods in practice*, pp. 197–223, Springer, 2001.

[12] N. Oudjane and C. Musso, "Progressive correction for regularized particle filters," in *Proceedings of the Third International Conference on Information Fusion*, vol. 2, pp. THB2–10, IEEE, 2000.

[13] Z. Fang, A. Gupta, and M. Khammash, "Convergence of regularized particle filters for stochastic reaction networks," *SIAM Journal on Numerical Analysis*, vol. 61, no. 2, pp. 399–430, 2023.

[14] D. Crisan and J. Míguez, "Nested particle filters for online parameter estimation in discrete-time state-space markov models," *Bernoulli*, vol. 24, no. 4A, pp. 3039–3086, 2018.

[15] S. Pérez-Vieites, I. P. Mariño, and J. Míguez, "Probabilistic scheme for joint parameter estimation and state prediction in complex dynamical systems," *Physical Review E*, vol. 98, no. 6, p. 063305, 2018.

[16] S. Pérez-Vieites and J. Míguez, "Nested gaussian filters for recursive bayesian inference and nonlinear tracking in state space models," *Signal Processing*, vol. 189, p. 108295, 2021.

[17] K. Murphy and S. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Sequential Monte Carlo methods in practice*, pp. 499–515, Springer, 2001.

[18] C. Andrieu, N. Freitas, and A. Doucet, "Rao-blackwellised particle filtering via data augmentation," *Advances in Neural Information Processing Systems*, vol. 14, 2001.

[19] N. Chopin, "Central limit theorem for sequential Monte Carlo methods and its application to bayesian inference," *The Annals of Statistics*, vol. 32, no. 6, pp. 2385–2411, 2004.

[20] C. Zechner, M. Unger, S. Pelet, M. Peter, and H. Koeppl, "Scalable inference of heterogeneous reaction kinetics from pooled single-cell recordings," *Nature methods*, vol. 11, no. 2, pp. 197–202, 2014.

[21] Z. Fang, A. Gupta, S. Kumar, and M. Khammash, "Advanced methods for gene network identification and noise decomposition from single-cell data," *Nature Communications*, vol. 15, no. 1, p. 4911, 2024.

[22] M. West, "Approximating posterior distributions by mixtures," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 55, no. 2, pp. 409–422, 1993.

[23] D. Crisan and J. Míguez, "Particle-kernel estimation of the filter density in state-space models," *Bernoulli*, vol. 20, no. 4, pp. 1879–1929, 2014.

[24] P. Del Moral and L. Miclo, *Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non-linear filtering*. Springer, 2000.

[25] F. Le Gland and N. Oudjane, "Stability and uniform approximation of nonlinear filters using the hilbert metric and application to particle filters," *The Annals of Applied Probability*, vol. 14, no. 1, pp. 144–187, 2004.

[26] A. Bain and D. Crisan, *Fundamentals of stochastic filtering*, vol. 3. Springer, 2009.

[27] C. Doléans-Dade, "Quelques applications de la formule de changement de variables pour les semimartingales," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 16, pp. 181–194, 1970.

[28] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[29] A. J. Lotka, "Contribution to the theory of periodic reactions," *The Journal of Physical Chemistry*, vol. 14, no. 3, pp. 271–274, 2002.

[30] V. Volterra, "Variazioni e fluttuazioni del numero d'individui in specie animali conviventi.," *Memor. Accad. Lincei, Ser.*, vol. 6, pp. 31–113, 1926.