Factorization-Based Online Variational Inference for State-Parameter Estimation of Partially Observable Nonlinear Dynamical Systems

Liliang Wang* and Alex A. Gorodetsky†
University of Michigan, Ann Arbor, Michigan, 48105

We propose an online variational inference framework for joint state-parameter estimation in nonlinear systems. This approach provides a probabilistic estimate of both parameters and states, and does so without relying on parameter-state augmented filtering nor on mean-field assumption of independence of the two. The proposed method leverages a factorized form of the target posterior distribution to enable an effective pairing of variational inference for the marginal posterior of parameters with conditional Gaussian filtering for the conditional posterior of the states. This factorization is retained at every time-step via formulation that combines variational inference and regression. The effectiveness of the framework is demonstrated through applications to three example systems in different application domains.

I. Introduction

In a variety of areas ranging from robotics and financial prediction to computational neuroscience, the task of estimating system parameters and states from time-dependent observations is critical. Effective inference methods for such tasks must demonstrate efficiency, robustness, and scalability to enable rapid responses, ensure safety, and address various challenges. Inference methods for dynamical systems are generally categorized into offline and online approaches. Offline inference methods store historical data and reprocess the entire dataset each time new data is received, leading to continuously increasing storage demands and computational costs over time. Consequently, offline methods may be unsuitable for many tasks. On the other hand, online inference methods avoid storing historical data. Instead, they maintain the state and parameter estimates from the previous time step and update them using newly received data. This approach ensures constant storage and computational requirements over time, making online inference well-suited for the underlying parameter-state estimation task. However, achieving accurate online state estimation in nonlinear dynamical systems is often challenging. Simultaneous learning of states and system parameters in an online manner further increases the complexity of the task. To ensure safety, it can be important to accurately quantify uncertainty in both states and parameters. This necessitates an online parameter-state estimation approach that computes the joint distribution of states and system parameters (referred to as the joint posterior distribution) from available data, rather than relying solely on point estimates, which can significantly underestimate uncertainty.

Among online inference methods that provide distributions as estimates, classic Bayesian filtering techniques, such as Gaussian filters [1] and particle filters (PF) [2], remain some of the most prominent and widely utilized approaches. Gaussian filters, which include the Extended Kalman Filter (ExKF), Unscented Kalman Filter (UKF), and Ensemble Kalman Filter (EnKF) [3], operate under the assumption that the posterior distribution of the states follows a Gaussian distribution. They recursively update the mean and covariance of this Gaussian approximate distribution over time by assimilating the received data points incrementally. Conversely, particle filters do not impose constraints on the distribution type of the posterior. Instead, they employ empirical distributions to approximate the posteriors, offering flexibility and robustness particularly in low-dimensional problems. However, particle filters suffer from the weight degeneracy for large-scale problems [4]. It's important to note that both Gaussian filter and PF were initially designed for inferring the distribution of states only. To enable simultaneous inference of states and system parameters, a common approach involves augmenting the state vector with system parameters, leading to the development of joint filters such as the joint Extended Kalman Filter (ExKF) [5] and joint Unscented Kalman Filter (UKF) [6–8]. While joint Gaussian filters are often computationally efficient, they may lack robustness, particularly in scenarios where the true posterior is highly non-Gaussian, a common occurrence when only a small amount of data are received. In PF, the simultaneous inference of system parameters and states exacerbates the issue of weight degeneracy as the parameters are static.

*Graduate Student, Department of Aerospace Engineering.
†Associate professor, Department of Aerospace Engineering.

While introducing artificial dynamics for the system parameters can partially alleviate this problem, it provides limited mitigation and often introduces artificial variance inflation, ultimately reducing estimation accuracy [9]. As a result, PF exhibits unsatisfactory performance even for problems that are not considered high-dimensional.

In addition to classic Bayesian filtering approaches, optimal transport (OT) techniques [10] have recently been explored for online state-parameter estimation. For example, [11] computes an approximate joint posterior distribution of states and parameters using an optimal transport map. However, OT methods face two significant limitations. First, few OT algorithms are able to compute transport maps in continuous spaces, which limits their practicality [12, 13]. Second, OT methods may be susceptible to the curse of dimensionality [14, 15]. As the dimensionality increases, the computational cost grows drastically, making OT methods less suitable for online estimation tasks.

Variational inference (VI) approximates the posterior distribution by minimizing the Kullback-Leibler (KL) divergence between the true and approximate posterior distributions. It is widely recognized for its superior scalability and ability to handle problems in continuous spaces. However, the majority of existing VI methods focus on offline inference, processing data in batches (e.g., [16–18]).

In recent years, some online variational methods have been developed to simultaneously estimate system parameters and states. A common approach among these methods is to combine particle filtering with variational inference. For instance, [19–21] incorporate particle filtering into amortized inference frameworks to achieve a tighter lower bound on the log-likelihood for sequential data, while [22] employs particle filtering to learn the filtering distribution of states. However, these particle-based online variational methods inherit the limitations of standard particle filters, such as weight degeneracy, which makes them unsuitable for large-scale problems.

In contrast, other online variational approaches use alternative distribution families to represent filtering distributions of states [23–25]. For instance, [25] employs the exponential family of distributions for state filtering, while [26] avoids predefined variational distribution families and utilizes a backward decomposition of the joint posterior and variational distribution. Despite these innovations, all of the aforementioned methods [19–26] rely on point estimates for system parameters, leading to significant underestimation of uncertainty.

Additionally, several methods have been proposed to leverage online variational techniques for computing the joint distribution of states and system parameters (e.g., [27, 28]). However, these approaches are restricted to linear systems and depend on the mean-field assumption of joint posterior, which presumes independence between the posterior distributions of states and system parameters. This strong assumption can lead to inaccurate approximations of joint posteriors in many practical scenarios.

In this paper, we propose a novel factorization-based online variational inference framework to estimate the joint distribution of system parameters and current states at each time step in an online fashion. The proposed method factorizes the joint posterior of states and parameters into the product of the marginal posterior distribution of the parameters and the conditional posterior of the current states given the parameters. The marginal posterior of the parameters is approximated using another distribution, while the conditional posterior of the current states is approximated by a Gaussian distribution with mean and covariance as functions of the parameters. Our method updates these approximations recursively over time using variational inference and Gaussian filtering. The primary contributions of this paper are summarized as follows:

1) We develop a posterior factorization-based online state-parameter estimation method that is able to provide a more accurate approximation of joint posterior than inference methods relying on mean-field assumptions and leverage existing filtering techniques. The method offers flexibility by allowing the approximate posterior distribution of the parameters to be chosen arbitrarily and by supporting the use of various Gaussian filtering methods to compute the conditional posterior of states given the parameters, to adapt to various problem settings.
2) Numerical experiments demonstrate that the proposed method is more robust than the joint UKF and exhibits superior scalability compared to the bootstrap particle filter with parameter-state augmentation. Furthermore, the results illustrate that the method effectively approximates highly non-Gaussian joint posterior distributions.

The remainder of this paper is organized as follows. Section II introduces the notation and formulates the online parameter-state estimation problem, along with an overview of variational inference and Gaussian filtering. Section III presents the proposed factorization-based online variational inference method. In Section IV, we evaluate the effectiveness of the proposed method through numerical experiments and compare its performance with several existing online inference methods for state-parameter estimation. Finally, the conclusions are summarized in Section V.

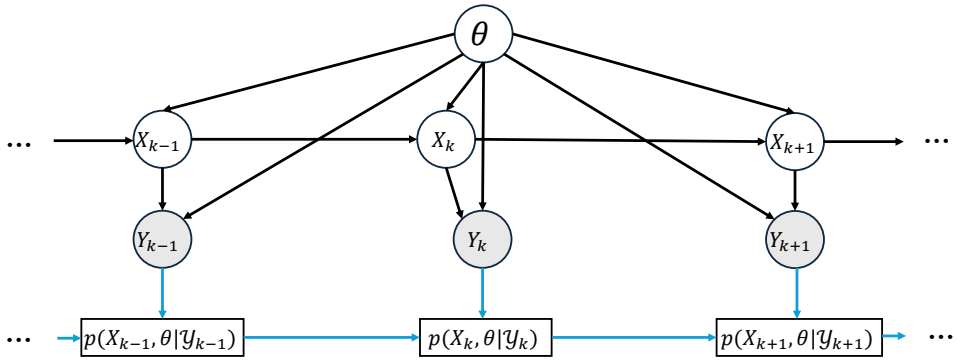


Fig. 1 Computation procedure: Blue arrows indicate the information flow used to compute the joint posteriors of states and parameters.

II. Background

In this section, we present the notation, outline the problem statement, and introduce the variational inference and Gaussian filtering method. In the following, let $\mathbb{R}$ denote the set of real numbers, $\mathbb{Z}_{\geq 0}$ denote the non-negative integers and $\mathbb{Z}_+$ denote the positive integers. Let $\mathcal{N}(\mu, R)$ denote the Gaussian distribution with mean μ and covariance R. The symbol $p_{\mathcal{N}}(x; \mu, R)$ denotes the probability density of a Gaussian distribution with mean μ and covariance R evaluated at x. Let $\nabla_x f(x)$ denote the gradient of function $f(x)$.

A. Problem statement

Consider a discrete-time system with a nonlinear dynamical model and a linear observation model

$$\begin{aligned} X_{k+1} &= \Phi(X_k, u_k; \theta) + W_k, \quad W_k \sim \mathcal{N}(0, \Sigma(\theta)), \\ Y_k &= H(\theta) X_k + V_k, \quad V_k \sim \mathcal{N}(0, \Gamma(\theta)), \end{aligned} \tag{1}$$

where $k \in \mathbb{Z}_{\geq 0}$ denotes a time index; and $X_k \in \mathbb{R}^n$, $Y_k \in \mathbb{R}^m$, $W_k \in \mathbb{R}^n$, and $V_k \in \mathbb{R}^m$ are random variables representing the state, observation, process noise, and measurement noise at time step k, respectively. The control inputs $u_k \in \mathbb{R}^d$, are assumed known. Furthermore, $\theta \in \mathbb{R}^r$ denotes unknown system parameters; Φ denotes the nonlinear dynamics mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$; and H denotes a linear (in state) observation function mapping from $\mathbb{R}^r$ to $\mathbb{R}^{m \times n}$.

Let y_k denote an observed value of Y_k at time step k and $\mathcal{Y}_k = \{y_1, y_2, \cdots, y_k\}$ denote the history of observations up to time k. Our objective is to compute the joint posterior $p(X_k, \theta \mid \mathcal{Y}_k)$ over both states X_k and parameters θ given observations $\mathcal{Y}_k$ obtained until time step k. Moreover, we seek an online (recursive) update formula that only performs one-pass over the data. Therefore the computation of the joint posterior should rely solely on the previous joint posterior $p(X_{k-1}, \theta \mid \mathcal{Y}_{k-1})$ from time $k-1$ and the newly received data point y_k. This computation procedure is illustrated in Figure 1.

It will be useful to consider a factorized representation of the joint posterior. To this end, $p(X_{k-1}, \theta \mid \mathcal{Y}_{k-1})$ factorizes as

$$p(X_{k-1}, \theta \mid \mathcal{Y}_{k-1}) = p(X_{k-1} \mid \theta, \mathcal{Y}_{k-1}) p(\theta \mid \mathcal{Y}_{k-1}). \tag{2}$$

The first term on the right-hand side is the classic filtering distribution computed during state estimation. The second term is the marginal posterior over the parameter. Updating these distributions to the next time-step relies on a straight-forward extension of the traditional prediction-update approach of data-assimilation [3]: the prediction step propagates the joint posterior at time $k-1$ forward in time without yet accounting for the data; and the update step incorporates the observation obtained at the current time step k via Bayes' rule. The details of the two steps are presented in the following two subsections.

Prediction The prediction step computes the joint predictive distribution $p(X_k, \theta \mid \mathcal{Y}_{k-1})$ by updating $p(X_{k-1} \mid \theta, \mathcal{Y}_{k-1})$ and $p(\theta \mid \mathcal{Y}_{k-1})$ via the dynamical model. Specifically, the joint predictive distribution $p(X_k, \theta \mid \mathcal{Y}_{k-1})$ can be factorized as

$$p(X_k, \theta \mid \mathcal{Y}_{k-1}) = p(\theta \mid \mathcal{Y}_{k-1}) p(X_k \mid \theta, \mathcal{Y}_{k-1}), \tag{3}$$

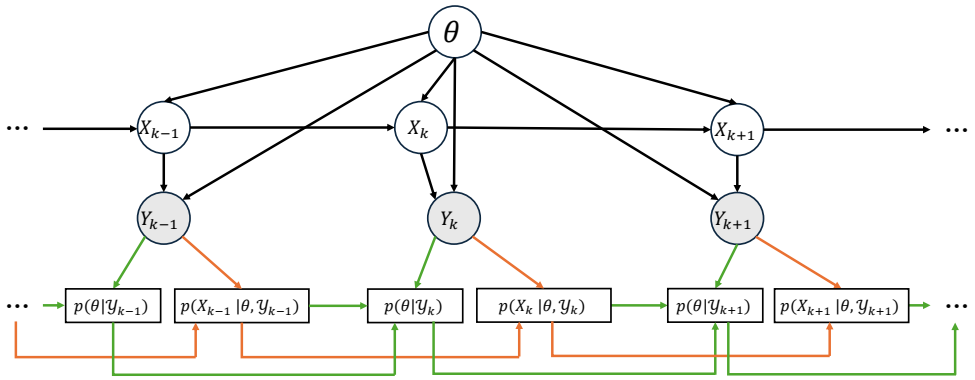


Fig. 2 Computation procedure for the factorized joint posterior: Green arrows represent the information flow for computing $p(\theta \mid \mathcal{Y}_k)$, while orange arrows indicate the flow for computing $p(X_k \mid \theta, \mathcal{Y}_k)$.

where $p(X_k \mid \theta, \mathcal{Y}_{k-1})$ is obtained through marginalization

$$p(X_k \mid \theta, \mathcal{Y}_{k-1}) = \int_{\mathbb{R}^n} p(X_k, X_{k-1} \mid \theta, \mathcal{Y}_{k-1}) dX_{k-1} = \int_{\mathbb{R}^n} p_{\mathcal{N}}(X_k; \Phi(X_{k-1}, u_{k-1}; \theta), \Sigma(\theta)) p(X_{k-1} \mid \theta, \mathcal{Y}_{k-1}) dX_{k-1}.$$

Update The predictive distribution (3) can then be updated with the new data point y_k through Bayes' rule

$$p(X_k, \theta \mid \mathcal{Y}_k) = \frac{p(y_k \mid X_k, \theta) p(X_k, \theta \mid \mathcal{Y}_{k-1})}{Z_k}, \tag{4}$$

where $Z_k = \int_{\mathbb{R}^n \times \mathbb{R}^r} p(y_k \mid X_k, \theta) p(X_k, \theta \mid \mathcal{Y}_k) dX_k d\theta$. However, we seek to exploit the factorized form (2), and instead seek solutions for $p(\theta \mid \mathcal{Y}_k)$ and $p(X_k \mid \theta, \mathcal{Y}_k)$, individually. Bayes' rule yields the following for the first term

$$p(\theta \mid \mathcal{Y}_k) = \frac{1}{Z_{\theta,k}} p(\theta \mid \mathcal{Y}_{k-1}) p(y_k \mid \theta, \mathcal{Y}_{k-1}),$$

where $Z_{\theta,k} = \int_{\mathbb{R}^r} p(\theta \mid \mathcal{Y}_{k-1}) p(y_k \mid \theta, \mathcal{Y}_{k-1}) d\theta$ is a normalizing constant, and the marginal likelihood $p(y_k \mid \theta, \mathcal{Y}_{k-1})$ can be written as

$$p(y_k \mid \theta, \mathcal{Y}_{k-1}) = \int_{\mathbb{R}^n} p(y_k, X_k \mid \theta, \mathcal{Y}_{k-1}) dX_k = \int_{\mathbb{R}^n} p_{\mathcal{N}}(y_k; H(\theta) X_k, \Gamma(\theta)) p(X_k \mid \theta, \mathcal{Y}_{k-1}) dX_k. \tag{5}$$

Similarly, the conditional posterior $p(X_k \mid \theta, \mathcal{Y}_k)$ is derived using Bayes' rule as

$$p(X_k \mid \theta, \mathcal{Y}_k) = \frac{1}{Z_{X,k}} p_{\mathcal{N}}(y_k; H(\theta) X_k, \Gamma(\theta)) p(X_k \mid \theta, \mathcal{Y}_{k-1}),$$

where $Z_{X,k} = \int_{\mathbb{R}^n} p_{\mathcal{N}}(y_k; H(\theta) X_k, \Gamma(\theta)) p(X_k \mid \theta, \mathcal{Y}_{k-1}) dX_k$ is a normalizing constant.

To conclude, given the factorized joint posterior from time step $k-1$, the joint posterior at time step k can be derived by iterating through the prediction and update steps. This computation procedure is summarized in Figure 2. However, computing this solution is typically intractable because expressions for $p(X_k \mid \theta, \mathcal{Y}_{k-1})$, $p(y_k \mid \theta, \mathcal{Y}_{k-1})$, $Z_{\theta,k}$, and $Z_{X,k}$ are generally not available in closed forms. As a result, approximating the posterior distributions becomes essential. In the following section, we introduce two existing methods that address the underlying problem based on distribution approximation.

B. Existing approaches

The prevalent existing approaches for online parameter-state estimation rely on parameter-state augmentation. In this section, we introduce two such methods: joint UKF and bootstrap particle filter (BPF) with parameter-state augmentation.

1. Parameter-State Augmentation

To estimate the parameters and states simultaneously from the observations, parameter-state augmentation endows the parameters with fictitious random walk dynamics

$$\theta_{k+1} = \theta_k + \epsilon_k, \tag{6}$$

where ϵ_k is a very small process noise. Next, an augmented state

$$\tilde{X}_k = \begin{bmatrix} X_k \\ \theta_k \end{bmatrix}$$

is introduced, so that the original system 1 with unknown parameters becomes a system with a fully known dynamical model

$$\begin{aligned} \tilde{X}_{k+1} &= \Phi(\tilde{X}_k, u_k) + \tilde{W}_k \\ Y_k &= h(\tilde{X}_k) + V_k, \end{aligned} \tag{7}$$

where

$$\tilde{W}_k = \begin{bmatrix} W_k \\ \epsilon_k \end{bmatrix},$$

and

$$h(\tilde{X}_k) = H(\theta_k) X_k.$$

This model can then be solved using various state-estimation techniques. Next we describe two approaches.

2. Gaussian filtering, joint UKF, and joint EnKF

Gaussian filtering [1] is one of the most commonly used classical inference methods. In this section, we introduce the general framework of Gaussian filtering. Gaussian filtering operates under the assumption that the joint distribution of the current state and observation, given all the previous observations, $p\left(X_k, Y_k \mid \mathcal{Y}_{k-1}\right)$ is Gaussian. Under this assumption, conditioning this distribution on new data yields a Gaussian posterior

$$p(X_k \mid \mathcal{Y}_k) = p_{\mathcal{N}}(X_k; m_k, C_k),$$

where

$$m_k = m_k^- + U_k S_k^{-1}(y_k - \mu_k), \tag{8}$$

$$C_k = C_k^- - U_k S_k^{-1} U_k^T, \tag{9}$$

where

$$\begin{aligned} m_k^- &:= \mathbb{E}[X_k \mid \mathcal{Y}_{k-1}], \\ C_k^- &:= \mathbb{V}ar[X_k \mid \mathcal{Y}_{k-1}], \\ \mu_k &:= \mathbb{E}[Y_k \mid \mathcal{Y}_{k-1}], \\ U_k &:= \mathbb{C}ov[X_k, Y_k \mid \mathcal{Y}_{k-1}], \\ S_k &:= \mathbb{C}ov[Y_k, Y_k \mid \mathcal{Y}_{k-1}]. \end{aligned}$$

Different Gaussian filtering approaches utilize different techniques to compute these expectations. The UKF method employs the unscented transformation (UT) for this computation [3]. UT is a method that estimates the means and covariances of random variables by approximating the corresponding integrals using quadrature points and associated weights. In contrast, the EnKF method utilizes the Monte Carlo technique to estimate these expectations [3].

In a manner analogous to the standard UKF, the joint UKF and joint EnKF applies the UKF method and EnKF method, respectively, to the augmented system 7. Both methods approximate the posterior distribution $p(\tilde{X}_k \mid \mathcal{Y}_k)$ with a Gaussian distribution and recursively computes its mean and covariance at each time step k.

3. Bootstrap particle filter (BPF) with parameter-state augmentation

In this section, we introduce bootstrap particle filter (BPF) with parameter-state augmentation [9, 29]. This method approximates the posterior with an empirical distribution, as given by

$$p(\tilde{X}_k \mid \mathcal{Y}_k) \approx \sum_{i=1}^{N} w_k^{(i)} \delta(\tilde{X}_k - \tilde{x}_k^{(i)}),$$

where N is the number of particles. Here $\tilde{x}_k^{(i)}$ and $w_k^{(i)}$ are the value and the weight for the i-th particle at time step k, respectively. The notation δ represents the Dirac delta function. Let $\tilde{X}_k^{(i)} = \left(\tilde{x}_0^{(i)}, \tilde{x}_1^{(i)}, \cdots, \tilde{x}_k^{(i)}\right)$ denotes the trajectory of the i-th particle till time step k. Then the weight for this particle can be updated according to

$$w_{k+1}^{(i)} = v_k^{(i)} \bar{w}_k^{(i)},$$

where

$$\begin{aligned}
\bar{w}_k^{(i)} &= \frac{w_k^{(i)}}{\sum_{i=1}^{N} w_k^{(i)}}, \\
v_k^{(i)} &= \frac{p(Y_{k+1} = y_{k+1} \mid \tilde{X}_{k+1} = \tilde{x}_{k+1}^{(i)}) p(\tilde{X}_{k+1} = \tilde{x}_{k+1}^{(i)} \mid \tilde{X}_k = \tilde{x}_k^{(i)})}{\pi(\tilde{x}_{k+1}^{(i)} \mid \tilde{X}_k^{(i)})},
\end{aligned}$$

where π is the proposal distribution. BPF uses the dynamics as the proposal, i.e., we have

$$\pi(\tilde{x}_{k+1}^{(i)} \mid \tilde{X}_k^{(i)}) = \mathcal{N}(\tilde{x}_{k+1}^{(i)}; \Phi(\tilde{x}_k^{(i)}), \tilde{\Sigma}),$$

where $\tilde{\Sigma}$ is the covariance of the process noise $\tilde{W}_k$ in system 7. PF typically suffers from weight degeneracy [4]. While this problem cannot be entirely eliminated, resampling procedures are commonly employed to mitigate its effects [2]. For clarity and to avoid redundancy, we will refer to the Bootstrap particle filter with parameter-state augmentation as BPF throughout the remainder of this paper.

Our proposed approach also approximates the posterior. However, it does so by leveraging an optimization-based approach for variational inference.

C. Variational inference

This section provides the general background of variational inference following [30, 31]. Variational inference seeks to approximate a target probability distribution via one in some smaller space of distributions. It does so through forming an optimization problem based on some error measure on distributions. Below, we provide the background in the context of approximating the solution to Bayesian inference problems by minimizing the Kullback-Leibler (KL) divergence between the approximate distribution and the target posterior.

Consider a general setting with uncertain parameter z, data d, and prior information I. Here z is a random variable the distribution of which we seek to obtain given all available information. Data d is often in the form of observations. Prior information I includes the form of the model, knowledge of the system physics, the prior belief and space of z. Bayes' rule provides the posterior as

$$\pi(z) = \frac{p(d \mid z, I) p(z \mid I)}{p(d \mid I)},$$

where $\pi(z) := p(z \mid d, I)$ is the posterior density. Variational inference approximates $\pi(z)$ with a distribution $q(z)$, referred to as variational distribution, by minimizing the KL divergence between q and π:

$$KL\left(q \| \pi\right) := \mathbb{E}_{q(z)}\left[\log \frac{q(z)}{\pi(z)}\right].$$

Typically the KL divergence is reformulated to enable a more tractable optimization problem. To this end, the KL divergence $KL\left(q \| \pi\right)$ can be written as

$$\begin{aligned}
KL\left(q \| \pi\right) &= -\mathbb{E}_{q(z)}\left[\log \pi(z)\right] + \mathbb{E}_{q(z)}\left[\log q(z)\right] \\
&= -\mathbb{E}_{q(z)}\left[-\log p(d \mid I) + \log p(d \mid z, I) + \log p(z \mid I)\right] + \mathbb{E}_{q(z)}\left[\log q(z)\right] \\
&= \log p(d \mid I) - \left(\mathbb{E}_{q(z)}\left[\log(p(d \mid z, I) p(z \mid I))\right] - \mathbb{E}_{q(z)}\left[\log q(z)\right]\right).
\end{aligned} \tag{10}$$

Define the evidence lower bound (ELBO) as

$$\mathcal{L}(q) := \mathbb{E}_{q(z)}\left[\log p(d, z \mid I)\right] - \mathbb{E}_{q(z)}\left[\log q(z)\right].$$

Since the term $\log p(d \mid I)$ in Equation (10) is independent of q, the variational distribution q can be obtained by maximizing the ELBO.

III. Methodology

In this section, we propose a factorization-based online variational inference approach (FBOVI) to address the problem described in Section II.A. The approach leverages the general framework of variational inference for the marginal posterior over the parameters as we well as Gaussian filtering for the conditional posterior of states given parameters.

The key idea of our approach is to approximate the distributions $p(X_k \mid \theta, \mathcal{Y}_k)$ and $p(\theta \mid \mathcal{Y}_k)$ at each time step k using more general variational inference formats for the marginal posterior of the parameter than those typically performed (e.g., than Gaussian filtering). We retain the approximation of the conditional posterior $p(X_k \mid \theta, \mathcal{Y}_k)$ as a Gaussian. More specifically, we have

$$p(X_k \mid \theta, \mathcal{Y}_k) = p_{\mathcal{N}}\left(X_k; m_k(\theta), C_k(\theta)\right),$$

where m_k and C_k are functions of θ. Therefore, at each time step k, our scheme computes q_k, m_k, and C_k via a two-step procedure, described as follows.

Step 1: Compute q_k. At time step k, in the online setting, the distributions $p(X_{k-1} \mid \theta, \mathcal{Y}_{k-1})$ and $p(\theta \mid \mathcal{Y}_{k-1})$ are replaced with the previously obtained approximations $p_{\mathcal{N}}(X_{k-1}; m_{k-1}(\theta), C_{k-1}(\theta))$ and $q_{k-1}(\theta)$, respectively. Then by Bayes' rule, the posterior distribution $p(\theta \mid \mathcal{Y}_k)$ satisfies

$$p(\theta \mid \mathcal{Y}_k) \propto p(y_k \mid \theta, \mathcal{Y}_{k-1}) q_{k-1}(\theta). \tag{11}$$

Recall from (5), that the marginal likelihood requires computing

$$p(y_k \mid \theta, \mathcal{Y}_{k-1}) = \int_{\mathbb{R}^n} p_{\mathcal{N}}(y_k; H(\theta)X_k, \Gamma(\theta)) p(X_k \mid \theta, \mathcal{Y}_{k-1}) dX_k.$$

Here, we use the UKF to approximate the predictive distribution $p(X_k \mid \theta, \mathcal{Y}_{k-1})$ as

$$p(X_k \mid \theta, \mathcal{Y}_{k-1}) \approx p_{\mathcal{N}}\left(X_k; m_k^-(\theta), C_k^-(\theta)\right).$$

The marginal likelihood can then be obtained analytically as

$$p(y_k \mid \theta, \mathcal{Y}_{k-1}) = p_{\mathcal{N}}\left(y_k; H(\theta)m_k^-(\theta), H(\theta)C_k^-(\theta)H(\theta)^T + \Gamma(\theta)\right).$$

Substituting this result into Equation (11), the posterior distribution becomes

$$p(\theta \mid \mathcal{Y}_k) \propto p_{\mathcal{N}}\left(y_k; H(\theta)m_k^-(\theta), H(\theta)C_k^-(\theta)H(\theta)^T + \Gamma(\theta)\right) q_{k-1}(\theta).$$

Next we need to obtain some approximate representation $q_k(\theta)$ of this term, and we do so using variational inference by maximizing the ELBO

$$\mathcal{L}_k = \mathbb{E}_{q_k(\theta)}[\log\left(p_{\mathcal{N}}\left(y_k; H(\theta)m_k^-(\theta), H(\theta)C_k^-(\theta)H(\theta)^T + \Gamma(\theta)\right) q_{k-1}(\theta)\right)] - \mathbb{E}_{q_k(\theta)}[\log q_k(\theta)].$$

This simplifies to

$$\mathcal{L}_k = \mathbb{E}_{q_k(\theta)}[\Psi(\theta)] - KL(q_k \| q_{k-1}), \tag{12}$$

where

$$\Psi(\theta) = -\frac{1}{2}\left(y_k - H(\theta)m_k^-(\theta)\right)^T \left(H(\theta)C_k^-(\theta)H(\theta)^T + \Gamma(\theta)\right)^{-1} \left(y_k - H(\theta)m_k^-(\theta)\right) - \frac{1}{2}\log|H(\theta)C_k^-(\theta)H(\theta)^T + \Gamma(\theta)| - \frac{m}{2}\log(2\pi).$$

The problem of maximizing the ELBO in general settings typically either use stochastic gradient descent (SGD) [32] to account for the expectation or leverages sample average approximation [33]. In this paper we employ SGD to solve this optimization problem. Let ϕ_k represent the hyperparameters of the distribution q_k. The ELBO $\mathcal{L}_k$ is therefore a function of ϕ_k. At each optimization step t, let the current estimate of ϕ_k be denoted by $\phi_{k,t}$. The gradient $\nabla_{\phi_k}\mathcal{L}_k(\phi_k)|_{\phi_k=\phi_{k,t}}$ is estimated as

$$\nabla_{\phi_k}\mathcal{L}_k(\phi_k)|_{\phi_k=\phi_{k,t}} \approx \nabla_{\phi_k}\left(\frac{1}{N}\sum_{i=1}^{N}\Psi(\theta_t^{(i)}) + \log q_{k-1}(\theta_t^{(i)}) - \log q_k(\theta_t^{(i)})\right)\Bigg|_{\phi_k=\phi_{k,t}}, \tag{13}$$

where $\theta^{(i)}$ are samples generated from $q_k(\theta)$ with hyperparameters $\phi_{k,t}$.

With the distribution $q_k(\theta)$ determined, we proceed to approximate $p(X_k \mid \theta, \mathcal{Y}_k)$.

Step 2: Computation of m_k and C_k. Following step 1, we compute the functions $m_k(\theta)$ and $C_k(\theta)$ at step 2. Theoretically, the exact functional representations of m_k and C_k can be obtained at each time step k based on m_{k-1} and C_{k-1} using the UKF to solve Equations (8) and (9). In our case, however, we are representing the conditional posterior and this mean and covariance are *functions of* θ. These particular representations of the mean and covariance are inconvenient because it would requiring storing all intermediate terms such as $\mu_k(\theta), S_k(\theta)$, etc. over all time steps. Therefore, we seek a closed-form approximation of these terms as a function of θ.

Specifically, let us now refer to Equation (8) and (9) as giving the exact posterior mean m_k^* and covariance C_k^*, respectively. These exact representations are computed using the UKF given the current $m_{k-1}(\theta)$ and $C_{k-1}(\theta)$. The task then is to generate $m_k(\theta)$ and $C_k(\theta)$ as approximations to these exact mean and covariance. We do this by formulating a regression problem. We generate data from the exact equations by sampling θ from the marginal posterior, and then we fit a surrogate to this data. In this paper, we deploy neural networks to represent m_k and C_k, with their weights denoted by η_k. These networks are trained via stochastic gradient descent. At each gradient descent step t, $\tilde{N}$ samples of θ, $\{\tilde{\theta}_t^{(i)}\}_{i=1}^{\tilde{N}}$, are drawn from $q_k(\theta)$, and a mean squared error is minimized. This error is defined as:

$$MSE(\eta_k) := \frac{1}{\tilde{N}}\sum_{i=1}^{\tilde{N}}\left(\|m_k(\tilde{\theta}_t^{(i)}) - m_k^*(\tilde{\theta}_t^{(i)})\|_2^2 + \|c_k(\tilde{\theta}_t^{(i)}) - c_k^*(\tilde{\theta}_t^{(i)})\|_2^2\right), \quad \tilde{\theta}_t^{(i)} \sim q_k(\theta), \tag{14}$$

where $c_k(\tilde{\theta}_t^{(i)})$ and $c_k^*(\tilde{\theta}_t^{(i)})$ represent the vectors obtained by flattening matrices $C_k(\tilde{\theta}_t^{(i)})$ and $C_k^*(\tilde{\theta}_t^{(i)})$, respectively, and the values of $m_k^*(\tilde{\theta}_t^{(i)})$ and $C_k^*(\tilde{\theta}_t^{(i)})$ are computed given y_k and $\tilde{\theta}_t^{(i)}$ using UKF by assuming

$$p(X_{k-1} \mid \tilde{\theta}_t^{(i)}, \mathcal{Y}_{k-1}) = p_{\mathcal{N}}(X_{k-1}; m_{k-1}(\tilde{\theta}_t^{(i)}), C_{k-1}(\tilde{\theta}_t^{(i)})).$$

The framework of the proposed method is illustrated in Figure 3 and the detailed procedure for a single time step k is outlined in Algorithm 1. In practice, the selection of the number of samples for the optimization procedures in Steps 1 and 2, denoted by N and $\tilde{N}$, plays a critical role in determining the computational efficiency of the proposed method. A trade-off exists between the computational cost per optimization step and the total number of steps required for convergence. Smaller values of N and $\tilde{N}$ reduce the computational time needed for each optimization step but result in higher variance in the gradient estimation, which can increase the total number of steps required for convergence. Therefore, N and $\tilde{N}$ should be carefully chosen to balance the variance of gradient estimation and the number of optimization steps, in order to minimize the overall computational time.

Similarly, the choice of neural networks used to represent m_k and C_k should align with the desired performance and computational constraints. More expressive neural networks generally yield more accurate approximations, enhancing the method's overall performance. However, employing more expressive networks typically increases the time required for supervised learning in Step 2 and imposes higher storage requirements.

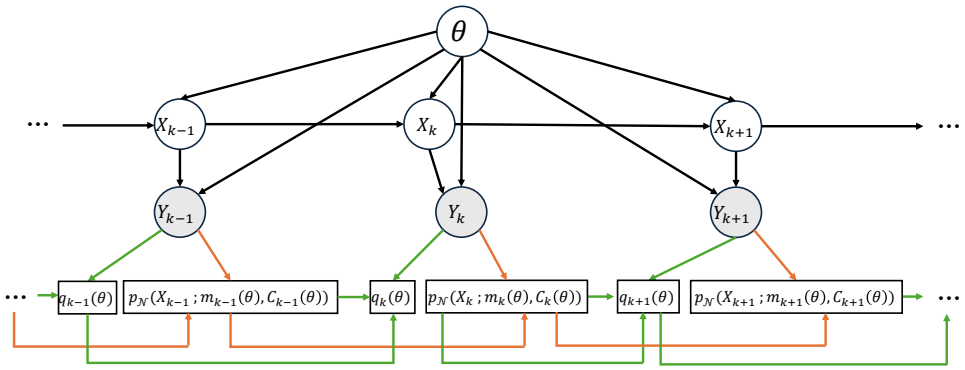


Fig. 3 Scheme of FBOVI. The green arrows represent the information flow for computing q_k. The orange arrows represent the information flow for computing m_k and C_k.

Algorithm 1 Factorization-Based Online Variational Inference Algorithm (Single Time Step)

Input:
Variational hyperparameters obtained at the previous time step ϕ_{k-1}, η_{k-1};
Measurement y_k;
Maximum number of optimization steps T_{max} and $\tilde{T}_{max}$ for optimization over ϕ_k and η_k respectively
Output:
Variational hyperparameters ϕ_k, η_k

Step 1 begins:
Optimization step $t \leftarrow 0$
Initialize ϕ_k with ϕ_{k-1}: $\phi_{k,0} \leftarrow \phi_{k-1}$
repeat
  Draw N samples of θ: $\{\theta^{(i)}\}_{i=1}^N$ with $\theta^{(i)} \sim q_k(\theta)$
  Estimate the gradient $\nabla_{\phi_k} \mathcal{L}_k(\phi_k)|_{\phi_k=\phi_{k,t}}$ using $\{\theta^{(i)}\}_{i=1}^N$ according to Equation (13)
  Set the learning rate γ_t
  Update the variational parameters: $\phi_{k,t+1} \leftarrow \phi_{k,t} + \gamma_t \nabla_{\phi_k} \mathcal{L}_k(\phi_k)|_{\phi_k=\phi_{k,t}}$
  $t \leftarrow t + 1$
until Convergence of $\phi_{k,t}$ or $t = T_{max}$
$\phi_k \leftarrow \phi_{k,t}$
Step 1 ends
Step 2 begins:
Optimization step $t \leftarrow 0$
Initialize η_k with η_{k-1}: $\eta_{k,0} \leftarrow \eta_{k-1}$
repeat
  Draw $\tilde{N}$ samples of θ: $\{\tilde{\theta}^{(i)}\}_{i=1}^N$ with $\tilde{\theta}_t^{(i)} \sim q_k(\theta)$
  Estimate $MSE(\eta_k)$ using $\{\tilde{\theta}^{(i)}\}_{i=1}^N$ according to Equation (14)
  Set the learning rate $\tilde{\gamma}_t$
  Update the hyperparameters for m_k and C_k: $\eta_{k,t+1} \leftarrow \eta_{k,t} + \tilde{\gamma}_t \nabla_{\eta_k} MSE(\eta_k)|_{\eta_k=\eta_{k,t}}$
  $t \leftarrow t + 1$
until Convergence of $\eta_{k,t}$ or $t = \tilde{T}_{max}$
$\eta_k \leftarrow \eta_{k,t}$
Step 2 ends

IV. Numerical Experiments and Evaluations

In this section, we implement FBOVI in several numerical experiments and compare it with several existing online state-parameter estimation algorithms to demonstrate its performance.

A. Evaluation criteria

In this section, we introduce the criteria used for evaluating the performance of the online parameter-state estimation algorithms. We first provide the criteria for assessing the accuracy of parameter learning and state estimation. Subsequently, we describe the criteria for evaluating the prediction performance.

1. Estimation accuracy evaluation

In this section, we introduce a criterion for evaluating the accuracy of parameter learning and state estimation: root mean squared error (RMSE). RMSE is computed using the mean of the marginal posterior distribution obtained by an algorithm over different data realizations. Firstly, to evaluate the accuracy of the estimates at each time step, we use single-step-element RMSE. The single-step-element RMSE for estimating the i-th element of the parameter θ and learning the i-th element of the state X_k at time step k are defined as:

$$RMSE_{\hat{\theta}_k^{(i)}} \triangleq \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\hat{\theta}_{j,k}^{(i)} - \bar{\theta}^{(i)}\right)^2}, \tag{15}$$

$$RMSE_{\hat{x}_k^{(i)}} \triangleq \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\hat{x}_{j,k}^{(i)} - x_{j,k}^{(i)}\right)^2}, \tag{16}$$

where N is the total number of data realizations. Here, $\hat{\theta}_{j,k}^{(i)}$ and $\hat{x}_{j,k}^{(i)}$ represent the mean of the marginal approximate posterior distribution for the i-th element of θ and the i-th element of state X at time step k, respectively, based on the j-th data realization. The symbol $\bar{\theta}^{(i)}$ denotes the true value of the i-th element of θ, while $x_{j,k}^{(i)}$ represents the true value of the i-th element of the state at time step k for the j-th data realization.

Secondly, to provide an overall evaluation for the accuracy of estimates, we use all-time-single-element RMSE and all-time-element RMSE. All-time-single-element RMSE calculates the average estimate error across all time steps for a single element of the parameter or state, while all-time-element RMSE calculates the average estimate error across all time steps and all elements of the parameter or state. The all-time-single-element RMSE for θ and state are defined as follows:

$$RMSE_{\theta_i} \triangleq \sqrt{\frac{1}{N}\frac{1}{K}\sum_{j=1}^{N}\sum_{k=1}^{K}\left(\hat{\theta}_{j,k}^{(i)} - \bar{\theta}^{(i)}\right)^2}, \tag{17}$$

$$RMSE_{x_i} \triangleq \sqrt{\frac{1}{N}\frac{1}{K}\sum_{j=1}^{N}\sum_{k=1}^{K}\left(\hat{x}_{j,k}^{(i)} - x_{j,k}^{(i)}\right)^2}, \tag{18}$$

where K is the total number of time steps.

Equation 19 and Equation 20 provide the definitions of the all-time-element RMSE for estimating the parameter θ and learning the state X, respectively:

$$RMSE_{\hat{\theta}} \triangleq \sqrt{\frac{1}{N}\frac{1}{K}\frac{1}{r}\sum_{j=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{r}\left(\hat{\theta}_{j,k}^{(i)} - \bar{\theta}^{(i)}\right)^2}, \tag{19}$$

$$RMSE_{\hat{x}} \triangleq \sqrt{\frac{1}{N}\frac{1}{K}\frac{1}{n}\sum_{j=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\left(\hat{x}_{j,k}^{(i)} - x_{j,k}^{(i)}\right)^2}. \tag{20}$$

2. Prediction ability evaluation

In this section, we introduce two criteria for evaluating the prediction performance of online parameter-state estimation algorithms: single-step prediction RMSE and all-time prediction RMSE. Both metrics quantify the error

between the predicted state at the next time step, based on the approximate joint posterior obtained at the current time step, and the true state value at the next time step. These RMSE values are computed using the approximate joint posterior distribution of parameters and states across different data realizations.

The single-step prediction RMSE reflects the average prediction error for the next-step state at each time step. At time step k, it is defined as

$$RMSE_{pred,k} \triangleq \sqrt{\frac{1}{N}\sum_{j=1}^{N} \mathbb{E}_{(X_k,\theta)\sim q_{j,k}(X_k,\theta),W_k\sim\mathcal{N}(0,\Sigma(\theta))}\left[\|(\Phi(X_k,u_k;\theta)+W_k)-x_{j,k+1}\|_2^2\right]}, \tag{21}$$

where $q_{j,k}(X_k,\theta)$ denotes the approximate joint posterior distribution at time step k for the j-th data realization and $x_{j,k+1}$ represents the true state value at time step $k+1$ for the j-th data realization.

The all-time prediction RMSE provides the average next-step prediction error across all time steps and data realizations, which is defined as

$$RMSE_{pred} \triangleq \sqrt{\frac{1}{N}\frac{1}{K}\sum_{j=1}^{N}\sum_{k=1}^{K} \mathbb{E}_{(X_k,\theta)\sim q_{j,k}(X_k,\theta),W_k\sim\mathcal{N}(0,\Sigma(\theta))}\left[\|(\Phi(X_k,u_k;\theta)+W_k)-x_{j,k+1}\|_2^2\right]}. \tag{22}$$

In all experiments detailed in this paper, the expectations required for computing RMSE are obtained using Monte Carlo method with a fixed sample size of 1×10^4.

B. Experimental settings

The FOBVI framework is implemented in Pytorch[*]. The ADAM optimizer is applied for maximizing the online ELBO $\mathcal{L}_k$ and minimizing the MSE of function fitting defined in Equation (14), to determine the hyperparameters ϕ_k and η_k, respectively. For the examples presented in this section, the conditional mean m_k and covariance C_k are represented using neural networks. Additionally, Gaussian distributions are employed as the variational distributions for the system parameters.

Each process noise in the artificial dynamics assigned by joint UKF, joint EnKF and BPF is distributed according to a Gaussian distribution with zero mean. To demonstrate the effectiveness of the proposed method, various combinations of hyperparameter values were tested for both the joint UKF and the BPF. For the joint UKF, the hyperparameters include those for the Unscented Transform (UT) and the process noise covariance assigned to the artificial dynamics of the system parameters. The process noise covariance is represented as a diagonal matrix with diagonal elements $[\sigma_1^2,\sigma_2^2,\cdots,\sigma_r^2]$. Different combinations of UT hyperparameter values and the values of σ_i^2 (for $i=1,2,\cdots,r$) are tested. For the BPF, the primary hyperparameters are elements of the process noise covariance for the artificial dynamics assigned to the system parameters. This process noise covariance is also represented as a diagonal matrix with the diagonal $[\tilde{\sigma}_1^2,\tilde{\sigma}_2^2,\cdots,\tilde{\sigma}_r^2]$. Different combinations of the values of $\tilde{\sigma}_i^2$ for $i=1,2,\cdots,r$ are explored. The results presented in the following subsections for both the joint UKF and BPF correspond to the configurations yielding the lowest mean of all-time-element RMSE for parameter and state. The experiments are not run in real time. The new data point is received after the method completes processing the previous data point. In the first two experiments, the approximate posterior obtained using BPF with 1×10^7 particles is treated as the ground truth posterior for the first 50 time steps. It is important to note that BPF (1×10^7) is used solely to provide a reference for the true posterior and is nearly impractical for online tasks in real-world applications due to its computational demands.

C. Pendulum System

In this section, we learn a nonlinear dynamical model for a two-dimensional system. The true system considered here is a nonlinear single pendulum system described by

$$\frac{d}{dt}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}x_2\\-\frac{g}{l}\sin(x_1)\end{bmatrix},$$

[*] https://pytorch.org

where $g = 9.8m/s^2$ represents the gravity acceleration and $l = 1.2m$ is the length of the pendulum. The corresponding discrete-time single pendulum system, with a time interval $\Delta t = 0.1s$, is given by

$$\begin{bmatrix} x_1^k \\ x_2^k \end{bmatrix} = \begin{bmatrix} x_1^{k-1} + \Delta t x_2^{k-1} \\ x_2^{k-1} - \frac{g}{l}\Delta t \sin(x_1^{k-1}) \end{bmatrix}.$$

Noisy measurements are taken at every time step according to

$$Y_k = x_1^k + r^k, \quad r^k \sim \mathcal{N}(0, R), \tag{23}$$

where $R = 0.01$. To evaluate performance, the simulation is repeated 100 times independently, generating 100 different data realizations.

Assuming the measurement model (23) is known, we aim to learn a nonlinear dynamical model parameterized as

$$\begin{bmatrix} x_1^k \\ x_2^k \end{bmatrix} = \begin{bmatrix} \theta_1 x_1^{k-1} + \Delta t x_2^{k-1} \\ \theta_1 x_2^{k-1} - \theta_2 \sin(x_1^{k-1}) \end{bmatrix} + q^k, \quad q^k \sim \mathcal{N}(0, 0.01 I_{2\times 2}),$$

where $\Delta t = 0.1$ is known. The system parameters $\theta = (\theta_1, \theta_2)$ are unknown and the measurements are available at every time step. The initial condition of the system is set to $x_1^0 = 0.5, x_2^0 = 0.5$. The prior distribution for the state is distant from the initial condition and is specified as $\mathcal{N}(\begin{bmatrix} 3 \\ 4.5 \end{bmatrix}, 4I_{2\times 2})$. The prior for the parameter θ is given by $\mathcal{N}(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, I_{2\times 2})$.

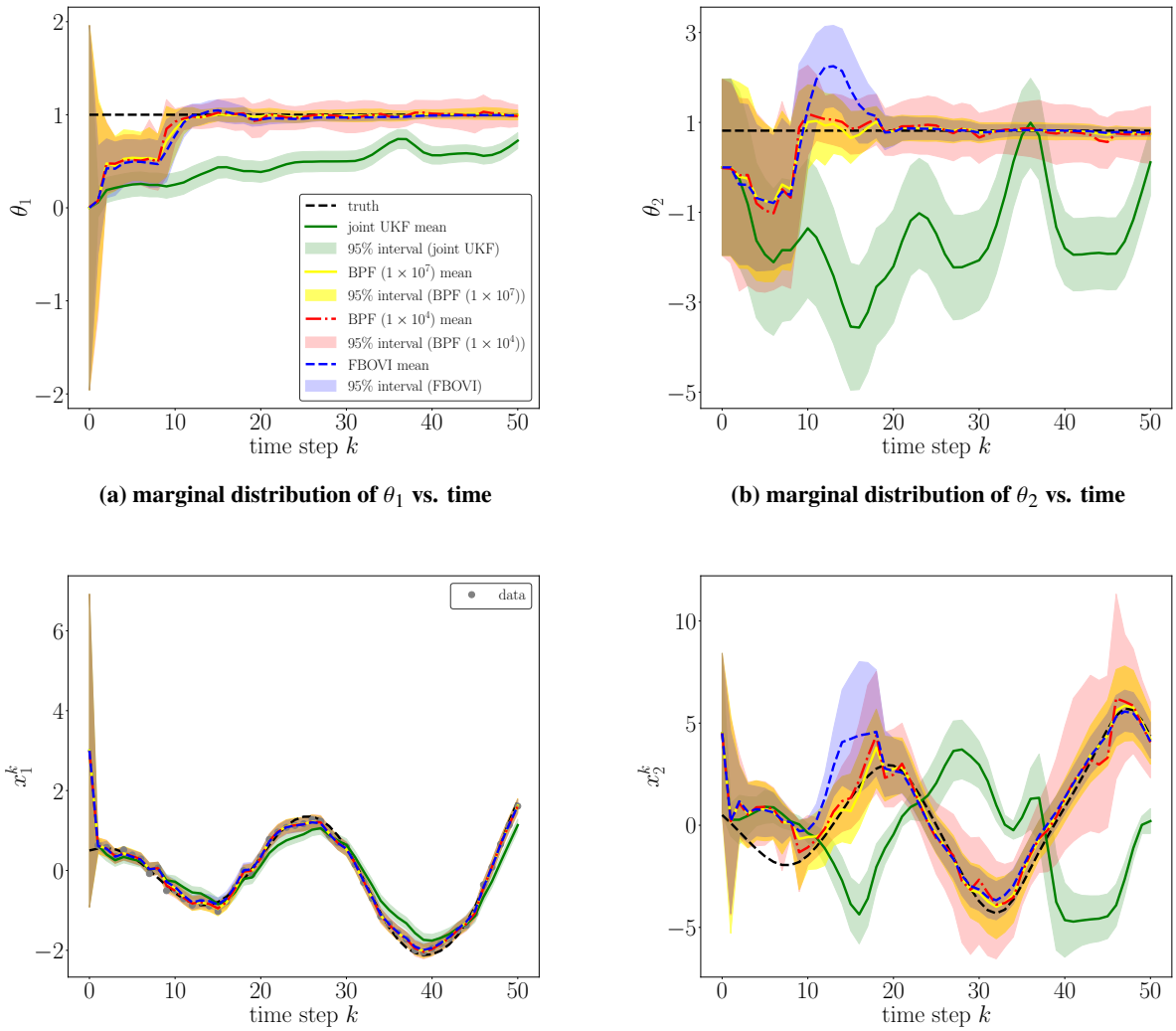
The marginal distributions of the states and system parameters obtained by the joint UKF, BPF with 1×10^4 particles, BPF with 1×10^7 particles, and the proposed method at each time step for a single data realization are presented in Figure 4. The results show that the marginal distribution means of the system parameters and states estimated by FOBVI converge to their true values after approximately 20 time steps. Addtionally, the approximate marginal posterior distribution obtained FBOVI closely aligns with the true posterior most of the time while BPF (1×10^4) overestimates uncertainty after time step 20. In contrast, the joint UKF demonstrates a considerable deviation from the true values, reflecting its limited capability to provide accurate online state-parameter estimates when handling simultaneous estimation of parameters and states. This discrepancy suggests that the joint posterior distribution of parameters and states may exhibit highly non-Gaussian characteristics, which the Gaussian assumption inherent in UKF struggles to capture.

Next, Figure 5 shows the estimation error of parameter learning and state estimation by employing the mean of the posterior as the point estimate over 100 data realizations: $RMSE_{\hat{\theta}_k^{(i)}}$ and $RMSE_{\hat{x}_k^{(i)}}$ for $i = 1, 2$ and $k = 1, 2, 3, \cdots, 50$. These metrics are defined in Equation (15) and Equation (16) respectively. FBOVI achieves the lowest RMSE for both parameter and state estimation after time step 20 compared to the other two algorithms. However, it is noted that FBOVI exhibits a higher error of estimating θ compared to PF before time step 17. Furthermore, Figure 5d indicates that FBOVI underperforms both BPF (1×10^4) and joint UKF in the early stage in terms of $RMSE_{\hat{x}_k^{(2)}}$. This temporary superior performance of BPF during this period can be attributed to the fact that the accumulated approximation error of FBOVI is at the peak during this time period and BPF's flexibility in representing distributions using 1×10^4 particles. Despite the limited approximation accuracy of these particles, they offer a more adaptable approximation compared to Gaussian distributions employed by FBOVI to represent the marginal distribution of the parameters. The observation that joint UKF outperforms FBOVI for $RMSE_{\hat{x}_k^{(2)}}$ during this period is unexpected. This might be partially explained by the discrepancy between the mean of the true posterior and the true values of the states during the early stages.

The all-time-element RMSE for parameters and states, $RMSE_{\hat{\theta}}$ and $RMSE_{\hat{x}}$ obtained by joint UKF, BPF (1×10^4) as well as FBOVI are shown in Table 1. BPF with 1×10^4 particles yields the lowest $RMSE_{\hat{\theta}}$ and $RMSE_{\hat{x}}$, due to its superior performance prior to time step 20.

Table 1 All-time-element RMSE for parameters and states:$RMSE_{\hat{\theta}}$ and $RMSE_{\hat{x}}$

	joint UKF	BPF (1×10^4)	FBOVI
$RMSE_{\hat{\theta}}$	1.8246	**0.3571**	0.5238
$RMSE_{\hat{x}}$	3.3189	**0.7439**	0.9220



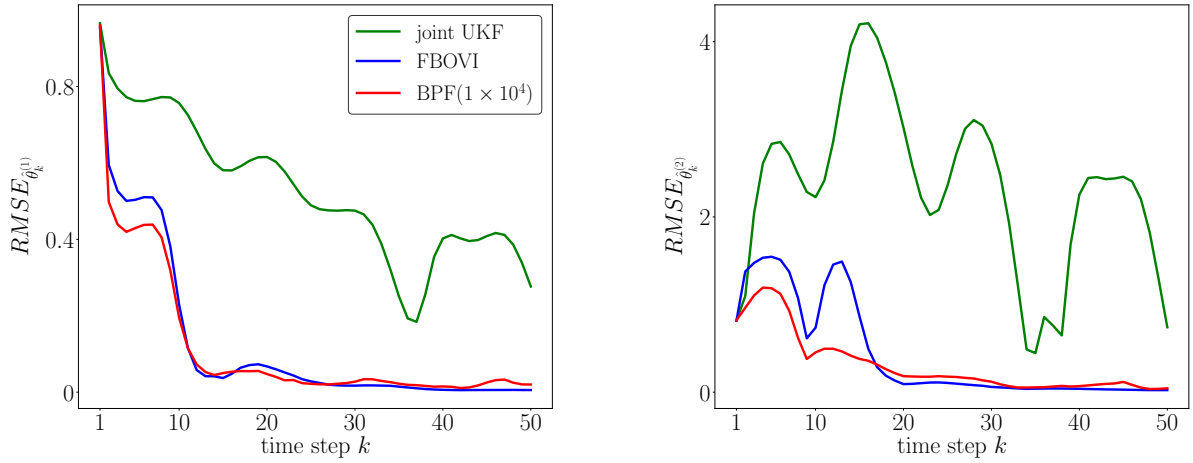
(a) marginal distribution of θ_1 vs. time

(b) marginal distribution of θ_2 vs. time

(c) marginal distribution of x_1 vs. time

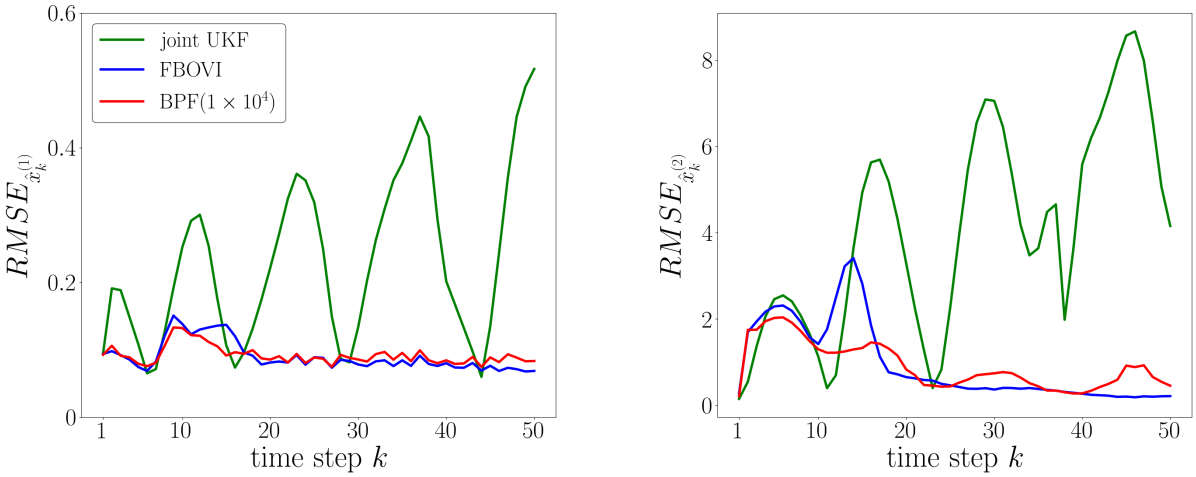
(d) marginal distribution of x_2 vs. time

Fig. 4 Temporal evolution of the marginal distributions of parameters and states for the nonlinear pendulum system. The top row illustrates the marginal distributions of the parameters θ_1 and θ_2 obtained using FOBVI, BPF, and joint UKF at each time step k. The bottom row depicts the marginal distributions of the states x_1 and x_2 obtained by the same methods. FBOVI and BPF learns the parameters θ rapidly.



(a) $RMSE_{\hat{\theta}_k^{(1)}}$ **vs. time** k

(b) $RMSE_{\hat{\theta}_k^{(2)}}$ **vs. time** k



(c) $RMSE_{\hat{x}_k^{(1)}}$ **vs. time** k

(d) $RMSE_{\hat{x}_k^{(2)}}$ **vs. time** k

Fig. 5 RMSE of parameter learning and state estimation over time in the pendulum system. The figures compare the outcomes of FBOVI, BPF, and joint UKF. The top two figures present the RMSE of parameter learning ($RMSE_{\hat{\theta}^{(i)}}$) for $i = 1, 2$, while the bottom row depicts the RMSE of state estimation ($RMSE_{\hat{\mathbf{x}}^{(i)}}$) for $i = 1, 2$. FBOVI outperforms BPF and joint UKF after time step 18**.**

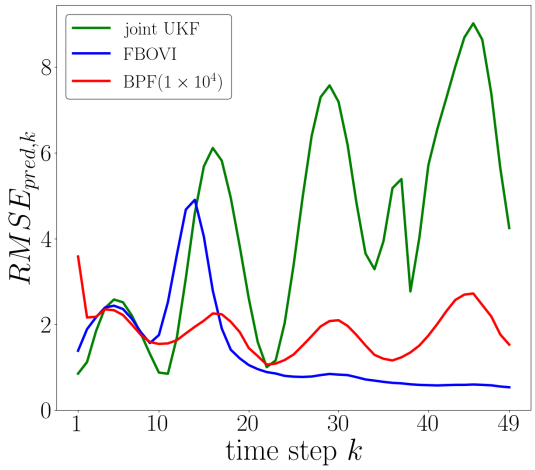


Fig. 6 Single-step prediction error for the pendulum system. Although FBOVI exhibits a higher prediction error during the early stage, its prediction error decreases significantly after step 15, ultimately achieving the lowest prediction error among the tested methods after time step 18.

The prediction RMSE of different algorithms at each time step are shown in Figure 6. FBOVI shows best prediction ability after time step 15 although it underperforms the other two methods before time step 14, which can be partially caused by its overestimation of the uncertainty from time step 10 to time step 15, as shown in Figure 4.

The all-time prediction RMSE for the joint UKF, BPF with 1×10^4 particles, and FBOVI are presented in Table 2. FBOVI achieves the lowest overall prediction error, indicating superior predictive performance. Since the prediction error is calculated using the entire approximate joint distribution of parameters and states, this result highlights FBOVI's ability to produce a more accurate approximation of the joint posterior compared to the other two methods.

Table 2 All-time prediction RMSE $RMSE_{pred}$

	joint UKF	BPF (1×10^4)	FBOVI
$RMSE_{pred}$	4.8977	1.9162	**1.7662**

D. Predator-prey model

Consider a predator-prey model described by the following Lotka-Volterra equations:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \alpha x_1 - \beta x_1 x_2 \\ -\gamma x_2 + \delta x_1 x_2 \end{bmatrix},$$

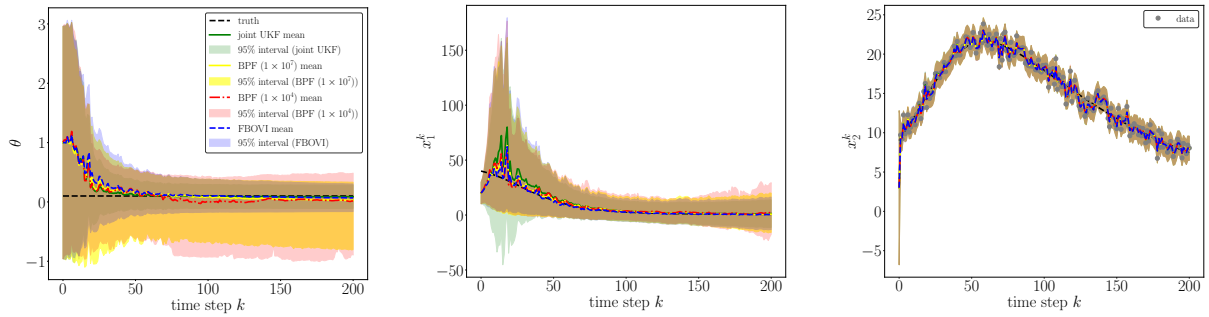
where x_1 and x_2 represent the population density of prey and predator, respectively. Here, $\alpha = 0.1$ and $\gamma = 0.1$ represent the prey birth rate and the predator death rate, while $\beta = 0.02$ and $\delta = 0.01$ represent the predation rate and the predator reproduction rate per unit prey consumed, respectively. Using zero-order hold approximation with a time interval $\Delta t = 0.1$, the corresponding discrete-time system is given by

$$\begin{bmatrix} x_1^{k+1} \\ x_2^{k+1} \end{bmatrix} = \begin{bmatrix} (1 + \Delta t\alpha)x_1^k - \Delta t\beta x_1^k x_2^k \\ (1 - \Delta t\gamma)x_2^k + \Delta t\delta x_1^k x_2^k \end{bmatrix}.$$

At each time step, a noisy measurement of the predator population density, x_2, is available. The measurement model is

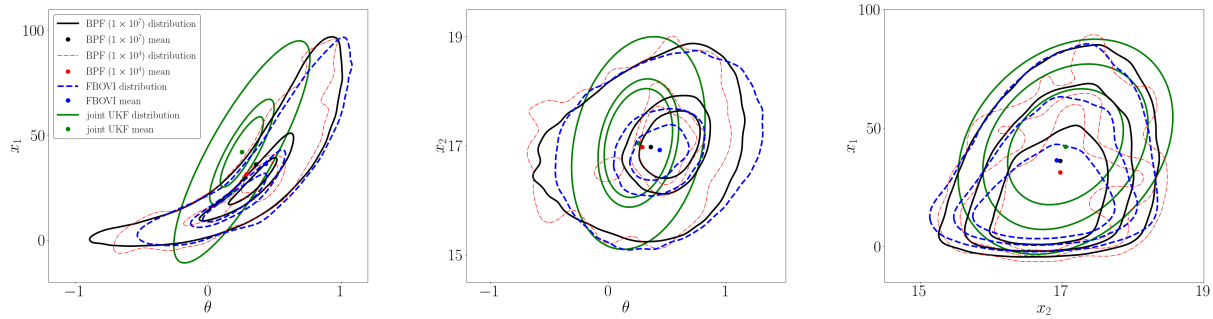
$$Y_k = x_2^k + r^k, \quad r^k \sim \mathcal{N}(0, 0.01).$$

To evaluate the performance of the proposed method, we employ it on two scenarios: one with prey birth rate α unknown and another with predator reproduction rate δ unknown.



(a) marginal distribution of θ **(b) marginal distribution of x_1^k** **(c) marginal distribution of x_2^k**

Fig. 7 Approximate marginal posterior distributions obtained using joint UKF, BPF, and FBOVI. The shaded areas represent the 95% credible intervals of the corresponding distributions. The means of the marginal distributions estimated by FBOVI converge rapidly to the true values.



(a) joint distribution of x_1 and θ **(b) joint distribution of x_2 and θ** **(c) joint distribution of x_1 and x_2**

Fig. 8 Approximate joint distributions of the parameter and states at time step 25 in the scenario with unknown prey birth rate. The lines represent the contours of the probability densities of the approximate joint posterior distributions. FBOVI effectively approximates the joint posterior overall.

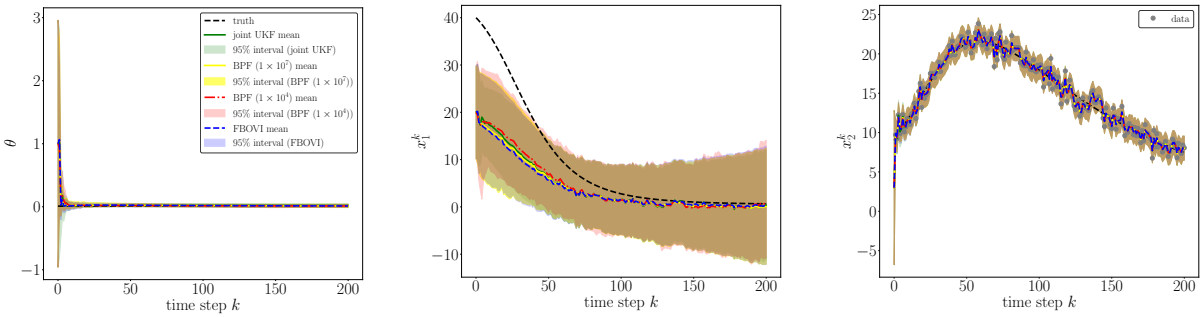
1. Unknown prey birth rate

In this section, the prey birth rate α is assumed to be unknown, and β, γ and δ are known. We represent α by θ and seek to infer the joint distribution of θ and the states (x_1^k, x_2^k) at each time step k. The dynamical model employed is:

$$\begin{bmatrix} x_1^{k+1} \\ x_2^{k+1} \end{bmatrix} = \begin{bmatrix} (1+\Delta t\theta)x_1^k - \Delta t\beta x_1^k x_2^k \\ (1-\Delta t\gamma)x_2^k + \Delta t\delta x_1^k x_2^k \end{bmatrix} + q^k, \quad q^k \sim \mathcal{N}(0, 0.01 I_{2\times 2}).$$

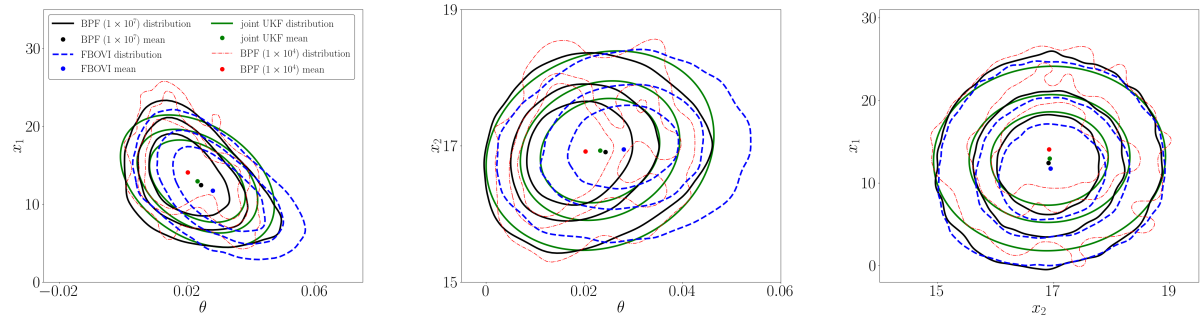
The marginal distributions of the states and parameter θ obtained using joint UKF, BPF, and FBOVI are shown in Figure 7. The results indicate that the means of the marginal distributions obtained by FBOVI and joint UKF converge to the true values after 50 time steps.

The 95% credible intervals of FBOVI and BPF (1×10^4) closely match that of the ground truth for the unobserved state x_1. To demonstrate the performance of joint posterior approximation of the proposed method, we present the approximate joint posterior of the states and parameter obtained by the algorithms at time step 25 in Figure 8. The joint posterior derived from FBOVI largely aligns with the ground truth, although it slightly underestimates the uncertainty for θ. In contrast, the joint UKF deviates significantly due to the highly non-Gaussian nature of the posterior, which cannot be captured by its Gaussian approximation. Notably, Figure 8 also reveals that BPF (1×10^4) suffers from severe weight degeneracy, even at an early time step of 25. This degeneracy prevents it from accurately approximating the posterior, despite its inherent flexibility.



(a) marginal distribution of θ **(b) marginal distribution of x_1^k** **(c) marginal distribution of x_2^k**

Fig. 9 Approximate marginal distributions of the parameter and states obtained using joint UKF, BPF, and FBOVI for the predator-prey model with an unknown predator reproduction rate. All methods successfully estimate the parameter θ within a short period. The 95% credible intervals produced by all methods are closely aligned.



(a) joint distribution of x_1 and θ **(b) joint distribution of x_2 and θ** **(c) joint distribution of x_1 and x_2**

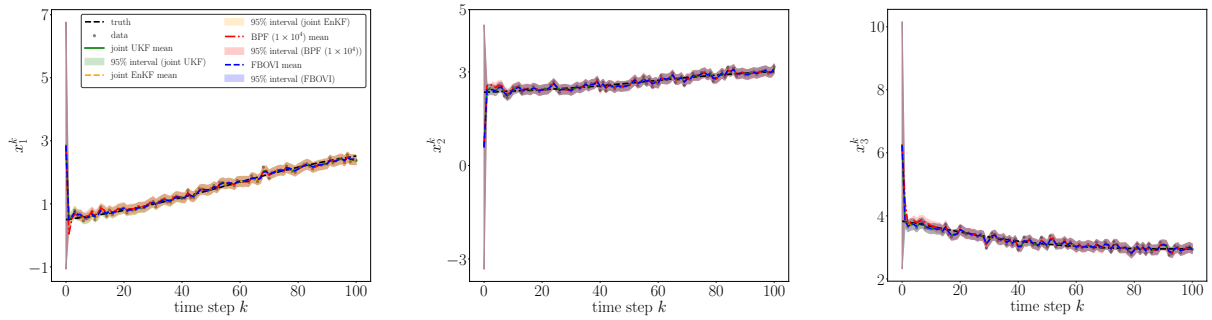
Fig. 10 Approximate joint distributions of the parameter and states at time step 25 in the scenario with unknown predator reproduction rate. The lines represent the contours of the probability densities of the approximate joint posterior distributions.

2. Unknown predator reproduction rate

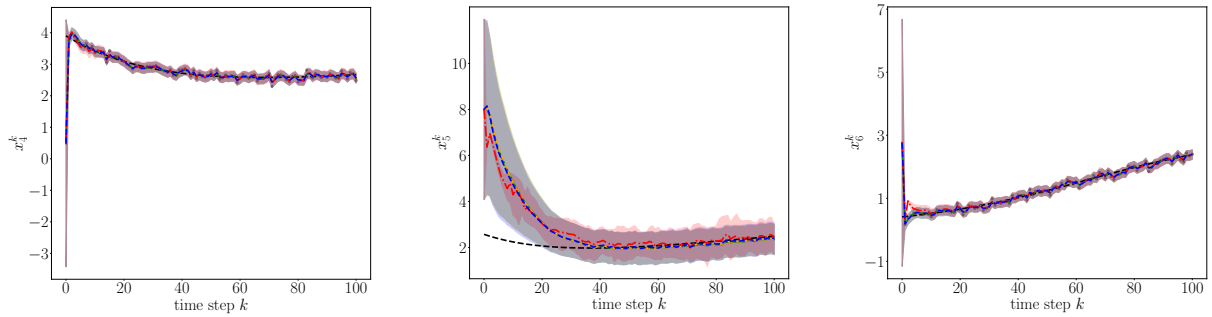
In this section, we assume the predator reproduction rate, δ, is unknown and α, β and γ are known. The dynamical system is parameterized by

$$\begin{bmatrix} x_1^{k+1} \\ x_2^{k+1} \end{bmatrix} = \begin{bmatrix} (1+\Delta t\alpha)x_1^k - \Delta t\beta x_1^k x_2^k \\ (1-\Delta t\gamma)x_2^k + \Delta t\theta x_1^k x_2^k \end{bmatrix} + q^k, \quad q^k \sim \mathcal{N}(0, 0.01 I_{2\times 2}).$$

The marginal distributions of parameters and states obtained by joint UKF, BPF and FBOVI are depicted in Figure 9. All methods learns the parameter θ rapidly. Compared with joint UKF and BPF, the mean of FBOVI for the unobserved state x_1 is closer to that of the ground truth. The approximate joint posteriors are provided in Figure 10. In this case, the true joint posterior is closer to a Gaussian distribution, allowing the joint UKF to achieve a good approximation. FBOVI continues to provide accurate approximation of the joint posterior. However, BPF (1×10^4) remains significantly affected by severe weight degeneracy, as highlighted in Figure 10.



(a) marginal distribution of x_1^k **(b) marginal distribution of x_2^k** **(c) marginal distribution of x_3^k**



(d) marginal distribution of x_4^k **(e) marginal distribution of x_5^k** **(f) marginal distribution of x_6^k**

Fig. 11 Approximate marginal posterior distributions of states

E. Six-dimensional system

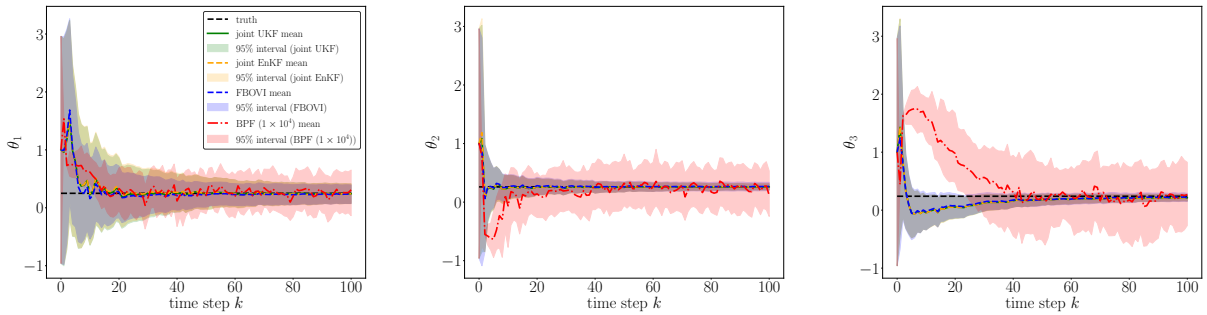
Consider a six-dimensional system described by:

$$\begin{bmatrix} x_1^{k+1} \\ x_2^{k+1} \\ x_3^{k+1} \\ x_4^{k+1} \\ x_5^{k+1} \\ x_6^{k+1} \end{bmatrix} = \begin{bmatrix} x_1^k + \Delta t \theta_1 \sin(x_1^k) \\ x_2^k + \Delta t \theta_2 (x_1^k + \sin(x_1^k)) + 0.25 \Delta t (x_3^k - 2x_2^k) \\ x_3^k + \Delta t \theta_2 (x_2^k + \sin(x_2^k)) + 0.25 \Delta t (x_4^k - 2x_3^k) \\ x_4^k + \Delta t \theta_3 (x_3^k + \sin(x_3^k)) + 0.25 \Delta t (x_5^k - 2x_4^k) \\ x_5^k + \Delta t \theta_3 (x_4^k + \sin(x_4^k)) + 0.25 \Delta t (x_6^k - 2x_5^k) \\ x_6^k + \Delta t \theta_1 \sin(x_6^k) \end{bmatrix},$$

where θ_1, θ_2 and θ_3 are unknown parameters. The noisy measurements of x_1, x_2, x_3, x_4, and x_6 are collected at each time step, with measurement noise distributed according to a Gaussian distribution with zero mean and covariance $0.01 I_{5\times5}$.

In this section, we evaluate the performance of the joint UKF, BPF (1×10^4), FBOVI, and joint EnKF. EnKF is particularly recognized for its effectiveness in high-dimensional problems. The approximate marginal distributions of states obtained by the four methods are shown in Figure 11. The 95% credible intervals of joint UKF, joint EnKF, and FBOVI are very closely aligned. Moreover, all four methods effectively track the unobserved state x_5 after 50 steps.

The marginal distribution of the parameters $\theta = (\theta_1, \theta_2, \theta_3)$ are presented in Figure 12. The means of the approximate posteriors obtained by joint UKF and FBOVI converge rapidly to the true values, whereas BPF struggles to learn the parameters. This is expected, as increasing dimensionality exacerbates weight degeneracy in BPF. To evaluate the performance of the four methods, the all-time-single-element $RMSE$ for the parameters and the unobserved state x_5 based on this single data realization is summarized in Table 3. The proposed method achieves the lowest error for estimating θ_2 and θ_3, though it slightly underperforms compared to joint UKF and joint EnKF in learning θ_1.



(a) marginal distribution of θ_1 **(b) marginal distribution of θ_2** **(c) marginal distribution of θ_3**

Fig. 12 Approximate marginal distributions of parameters θ

Table 3 All-time-single-element RMSE

	joint UKF	joint EnKF	BPF (1×10^4)	FBOVI
$RMSE_{\theta_1}$	**0.2013**	0.2031	0.2017	0.2135
$RMSE_{\theta_2}$	0.08542	0.0950	0.2264	**0.06421**
$RMSE_{\theta_3}$	0.1799	0.1811	0.5591	**0.1609**
$RMSE_{x_5}$	1.3922	1.3996	**1.1636**	1.3652

V. Conclusion

In this work, we propose a factorization-based online variational inference framework for parameter-state estimation in nonlinear dynamical systems. Unlike traditional methods, the proposed framework avoids restrictive assumptions about the structure of the joint posterior and supports the integration of various Gaussian filtering techniques. Numerical experiments across different application systems were conducted to validate its effectiveness. The results show that the proposed approach exhibits greater robustness compared to the joint UKF and superior scalability relative to the BPF with parameter-state augmentation.

Future work will focus on extending the proposed method to more general application problems. Furthermore, the algorithm will be modified to accommodate online estimation of system parameters and states under constraints, broadening its applicability to more complex and practical scenarios.

VI. Acknowledgements

This work was partially supported by an NSF CAREER Award, grant number CMMI-2238913.

References

[1] Ito, K., and Xiong, K., "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control*, Vol. 45, No. 5, 2020, pp. 910–927.

[2] Doucet, A., Freitas, N., and Neil, G., *Sequential Monte Carlo Methods in Practice*, Cambridge University Press, 2013.

[3] Säkkä, S., "Bayesian Filtering and Smoothing," 2013.

[4] Bengtsson, T., Bickel, P., and Li, B., "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," *Probability and Statistics: Essay in Honor of David A. Freedman*, 2008, pp. 316–334.

[5] Matthews, M. B., "A state-space appraoch to adaptive nonlinear filtering using recurrent neural networks," *Proceedings of IASTED Internat. Symp. Artificial Intelligence Application and Neural Networks*, 1990, pp. 197–200.

[6] Wielitzka, M., Dagen, M., and Ortmaier, T., "State estimation of vehicle's lateral dynamics using unscented Kalman filter," *IEEE 53rd Annual Conference on Decision and Control (CDC)*, 2014, pp. 5015–5020.

[7] Mahdianfar, H., Pavlov, A., and Aamo, O., "Joint unscented Kalman filter for state and parameter estimation in managed pressure drilling," *European Control Conference (ECC)*, 2013, pp. 1645–1650.

[8] Wielitzka, M., Dagen, M., and Ortmaier, T., "Joint Unscented Kalman Filter for State and Parameter Estimation in Vehicle Dynamics," *IEEE Conference on Control Applications (CCA)*, Sdyney, Australia, 2015.

[9] Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N., "On particle methods for parameter estimation in state-space models," *Statitical Science*, Vol. 30, No. 3, 2015, pp. 328–351.

[10] Moselhy, T., and Marzouk, Y., "Bayesian inference with optimal maps," *Journal of Computational Physics*, Vol. 231, No. 23, 2017, pp. 7815–7850.

[11] Grashorn, J., Broggi, M., Chamoin, L., and Beer, M., "Transport map coupling filter for state-parameter estimation," *Advances in Reliability, Safety and Security ESREL*, 2024.

[12] Nath, J. S., and Jawanpuria, P., "Statitical optimal transport posed as learning kernel mean embedding," *Advances in Neural Information Processing Systems*, 2020.

[13] Genevay, A., Cuturi, M., and Peyré, G., "Stochastic optimization for large-scale optimal transport," *Advances in Neural Information Processing Systems*, 2016.

[14] Meng, C., Ke, Y., Zhang, J., Zhong, M., Zhong, W., and Ma, P., "Large-scale optimal transport map estimation using projection pursuit," *Advances in Neural Information Processing Systems*, 2019, pp. 8116–8127.

[15] Niles-Weed, J., and Rigollet, P., "Estimation of Wasserstein distance in the spiked transport model," *Bernoulli*, Vol. 28, No. 4, 2022, pp. 2663–2688.

[16] Archer, E., Park, I. M., Buesing, L., Cunningham, J., and Paninski, L., "Black box variational inference for state space models," *International Conference on Learning Representations Workshop Track*, 2016.

[17] Courts, J., Hendriks, J., Wills, A., Schön, T. B., and Ninness, B., "Variational state and parameter estimation," *IFAC Symposium on System Identification*, 2021.

[18] Lambert, M., Chewi, S., Bach, F., Bonnabel, S., and Rigollet, P., "Variational inference via Wasserstein gradient flows," *Advances in Neural Information Processing Systems*, 2022.

[19] Le, T. A., Igl, M., Jin, T., Rainforth, T., and Wood, F., "Auto-encoding sequential monte carlo," *International Conference on Machine Learning*, 2018.

[20] Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Yee, T., "Filtering variational objectives," *Advances in Neural Information Processing Systems*, 2017.

[21] Naesseth, C., Linderman, S., Ranganath, R., and Blei, D., "Variational sequential monte carlo," *International Conference on Artifical Intelligence and Statistics*, 2018.

[22] Zhao, Y., Nassar, J., Jordan, I., Bugallo, M., and Park, I. M., "Streaming Variational monte carlo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, No. 1, 2023.

[23] Marino, J., Cvitkovic, M., and Yue, Y., "A general method for amortizing variational filtering," *Advances in Neural Information Processing Systems*, 2018.

[24] Zhao, Y., and Park, I. M., "Variational online learning of neural dynamics," *Frontiers in Computational Neuroscience*, Vol. 14, 2020.

[25] Dowling, M., Zhao, Y., and Park, I. M., "real-time variational method for learning neural trajectory and its dynamics," *11th International Conference on Learning Representations*, 2023.

[26] Campbell, A., Shi, Y., Rainforth, T., and Doucet, A., "Online variational filtering and parameter learning," *Advances in Neural Information Processing Systems*, 2021.

[27] Yoshimoto, J., Ishii, S., and Sato, M., "System identification based on online variational bayes method and its applications to reinforcement learning," *ICANN*, 2003, pp. 123–131.

[28] Neri, J., Badeau, R., and Depalle, P., "Probabilistic Filter and Smoother for Variational Inference of Bayesian Linear Dynamical System," *2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

[29] Kitagawa, G., "A self-organizing state-space model," *J. Amer. Statist. Assoc.*, Vol. 93, 1998, pp. 1203–1215.

[30] Blei, D. M., Kucukelbir, A., and McAnliffe, J. D., "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, 2017.

[31] Beal, M. J., "Variational algorithms for approximate Bayesian inference," *PhD thesis, University College London*, 2003.

[32] Kushner, H., and Yin, G., "Stochastic Approximation Algorithms and Applications," 1997.

[33] Kleywegt, A. J., Shapiro, A., and Homem-de mello, T., "The sample average approximation method for stochastic discrete optimization," *SIAM Journal on Optimization*, Vol. 12, No. 2, 2002, pp. 479–502.