Proper orthogonal decomposition method to nonlinear filtering problems in medium-high dimension

Zhongjian Wang, Xue Luo, *Senior member, IEEE,* Stephen S.-T. Yau, *Fellow, IEEE*, Zhiwen Zhang

***Abstract*—In this paper, we investigate the proper orthogonal decomposition (POD) method to numerically solve the forward Kolmogorov equation (FKE). Our method aims to explore the low-dimensional structures in the solution space of the FKE and to develop efficient numerical methods. As an important application and our primary motivation to study the POD method to FKE, we solve the nonlinear filtering (NLF) problems with a real-time algorithm proposed in [34] combined with the POD method. This algorithm is referred as POD algorithm in this paper. This algorithm consists of off-line and on-line stages. In the off-line stage, we construct a small number of POD basis functions that capture the dynamics of the system and compute propagation of the POD basis functions under the FKE operator. In the on-line stage, we solve the NLF problem in a real-time manner using the POD basis functions. Its convergence analysis has also been discussed. Some numerical experiments of the NLF problems are performed to illustrate the feasibility of our algorithm and to verify the convergence rate. The POD algorithm in our paper provides significant computational savings over the particle filter.**

The research of Z. Wang is partially supported by the Hong Kong PhD Fellowship Scheme. The work of X. Luo is supported by the National Natural Science Foundation of China (NSFC) (grant no. 11501023). The work of S. S.-T. Yau is supported by NSFC (grant no. 11471184). The research of Z. Zhang is supported by the Hong Kong RGC grants (27300616, 17300817), NSFC (grant no. 11601457), Seed Funding Programme for Basic Research (HKU), and the Hung Hing Ying Physical Sciences Research Fund (HKU).

Z. Wang is with Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong SAR. (e-mail: aris1992@outlook.com)

X. Luo is the co-first author. She is with School of Mathematics and System Sciences, Beihang University, Beijing 100191, PR China. (e-mail: xluo@buaa.edu.cn).

S. S.-T. Yau is the corresponding author. He is with the department of mathematical science, Tsinghua University, Beijing 100084, PR China. (e-mail: yau@uic.edu).

Z. Zhang is the corresponding author. He is with Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong SAR. (e-mail: zhangzw@hku.hk)

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org

***Index Terms*—Nonlinear filtering problems; Duncan-Mortensen-Zakai equation; proper orthogonal decomposition; real-time algorithm.**

I. Introduction

Nonlinear filtering (NLF) problem is originated from the problem of tracking and signal processing. The fundamental problem in the NLF is to give an instantaneous and accurate estimation of the states based on the noisy observations [19]. In this paper, we proposed an efficient numerical method to solve the forward Kolmogorov equation (FKE) arising from the NLF problem [18] . Our method is based on the proper orthogonal decomposition (POD) method [32], [6], [33], which is an effective tool in exploring the intrinsic low-dimensional structures of high-dimensional solutions. We start from the following signal based model:

$$\begin{cases} dx_t = f(x_t,t)dt + g(x_t,t)dv_t \\ dy_t = h(x_t,t)dt + dw_t \end{cases}, \qquad (1)$$

where $x_t \in R^n$ is the *states* of the system at time t, the initial state x_0 satisfying some initial distribution, $y_t \in R^m$ is the *observations* at time t with $y_0 = 0$, and v_t and w_t are vector-valued Brownian motion processes with covariance matrices $E[dv_t dv_t^T] = Q(t)dt \in R^{n\times n}$ and $E[dw_t dw_t^T] = S(t)dt \in R^{m\times m}$, $S(t) > 0$ respectively. Furthermore, we assume that x_0, dw_t and dv_t are independent. The most popular method so far to solve (1) is the particle filter, see [2], [12], [3] and references therein. However, the main drawback of the particle filter is that it is hard to be implemented as a real-time solver due to its nature of the Monte Carlo simulation.

In 1960s, Duncan [10], Mortensen [27], and Zakai [36] independently derived the so-called Duncan-Mortensen-Zakai (DMZ) equation or Zakai equation in some literature, which asserts that the unnormalized

conditional density function of the states x_t, denoted by $\sigma(x,t)$, satisfies the following Ito stochastic partial differential equation (SPDE):

$$\begin{cases} d\sigma(x,t) = \mathcal{L}\sigma(x,t)dt + \sigma(x,t)h^T(x,t)S^{-1}dy_t, \\ \sigma(x,0) = \sigma_0(x), \end{cases} \tag{2}$$

where $\sigma_0(x)$ is the density of the initial states x_0, and

$$\mathcal{L}(\cdot) := \frac{1}{2}\sum_{i,j=1}^{n}\frac{\partial^2}{\partial x_i \partial x_j}\big((gQg^T)_{ij}\,\cdot\,\big) - \sum_{i=1}^{n}\frac{\partial (f_i\cdot)}{\partial x_i}. \tag{3}$$

The DMZ equation laid down the solid foundation to study the NLF problem using SPDE. However, we cannot solve the DMZ equation analytically in general. Many efforts have been made to develop efficient numerical methods. One of the commonly used method is the splitting-up method originated from the Trotter product formula, which was first introduced in [5] and has been extensively studied later, see [28], [17], [13]. In [23], the so-called S^3 algorithm was developed based on the Wiener chaos expansion, by separating the computations involving the observations from those dealing only with the system parameters. However, the limitation of these methods is the boundedness of the drifting term f and the observation term h in (1).

To overcome this restriction, the third author and his co-worker [34] developed a novel algorithm to solve the pathwise robust DMZ equation. Specifically, for each given realization of the observation process denoted by y_t, they made an invertible exponential transformation

$$\sigma(x,t) = \exp\big(h^T(x,t)S^{-1}(t)y_t\big)u(x,t), \tag{4}$$

and transformed the DMZ equation (2) into a deterministic partial differential equation (PDE) with stochastic coefficient

$$\begin{cases} \dfrac{\partial}{\partial t}u(x,t) + \dfrac{\partial}{\partial t}(h^TS^{-1})y_t u(x,t) = \\ \qquad\qquad \exp\big(-h^T(x,t)S^{-1}(t)y_t\big)\big(\mathcal{L} - \dfrac{1}{2}h^TS^{-1}h\big) \\ \qquad\qquad \cdot\big(\exp\big(-h^T(x,t)S^{-1}(t)y_t\big)u(x,t)\big) \\ u(x,0) = \sigma_0(x) \end{cases} \tag{5}$$

Equation (5) is the so-called pathwise robust DMZ equation. The boundedness of the drift term f (contained in the operator $\mathcal{L}(\cdot)$) and observation term h is replaced by some mild growth conditions in this case. Nevertheless, they still make the assumption that the drift term, the observation term, and the diffusion term are time invariant, which means that f, h, and g in (1) cannot explicitly depend on time. Later on, in [24] the second and the third author of this paper generalized Yau-Yau's algorithm to more general settings of the NLF problems, namely, the time-dependent ones.

Let us assume that the observation time sequences $0 = t_0 < t_1 < \cdots < t_{N_t} = T$ are given. In each time interval $t_{j-1} \le t < t_j$, one freezes the stochastic coefficient y_t to be $y_{t_{j-1}}$ in (5) and makes the exponential transformation

$$u_j(x,t) = \exp\big(h^T(x,t)S^{-1}(t)y_{t_{j-1}}\big)u(x,t).$$

It is easy to deduce that u_j satisfies the FKE

$$\frac{\partial}{\partial t}u_j(x,t) = \big(\mathcal{L} - \frac{1}{2}h^TS^{-1}h\big)u_j(x,t), \tag{6}$$

where the operator $\mathcal{L}$ is defined in (3). In [25], the second and the third author of this paper investigated the Hermite spectral method to numerically solve the 1-D FKE (6) and analyzed the convergence rate of the proposed method. In their algorithm, the main idea is to shift part of the heavy computations in the off-line stage, so that only computations requiring observations are performed in the on-line stage and synchronized with off-line data.

The bottle-neck of the algorithm in [24] is to solve high-dimensional FKE accurately and compute a huge amount of numerical integrations on-line, if the state in NLF problems is high-dimensional. The heavy computation makes the real-time performance impossible, especially in high-dimensional problems. For example, Yueh et al. [35] proposed a numerical scheme based on the quasi-implicit Euler method for solving the high-dimensional FKE, which took more than 131 minutes to solve a 6-dimensional problem in time interval $[0, 20]$ with observation time step $\Delta\tau = 0.01$ on a desktop computer, which is far from being real-time. We remark that many progresses have been made along this direction though, it is still very challenging to solve high-dimensional FKE accurately in an effective fashion.

This motivates us to investigate the possible low structure of the high-dimensional FKE arising from NLF problems, so that we can design more efficient numerical methods. In fact, many high-dimensional problems have certain low-dimensional structures, e.g., in the sense of approximation using L_2 norm, which suggests the existence of reduced-order models (ROMs) and better formulations for efficient numerical methods. Inspired

by the last author's recent work on developing problem-dependent basis functions to solve SPDEs [7], [8], [9], we propose to use the POD method to explore the low-dimensional structures of the solutions to FKE. This in turn will help us obtain an efficient numerical method to solve the NLF problems.

Our POD algorithm consists of off-line and on-line stages. In the off-line stage, we shall use some reference numerical solutions, obtained by some numerical methods, such as finite difference method (FDM) [35] or spectral method [25], to gather the snapshots. Then we construct a set of reduced basis functions from the snapshot solutions, which we refer them as POD basis functions in the sequel. These POD basis functions represent the most energetic structures of the FKE, which provide an efficient way to explore the low-dimensional structures of the FKE solutions. In the on-line stage, we seek the numerical solution of the FKE in the linear space spanned by our POD basis functions and update with the new observation.

Our POD algorithm has the advantage that with only a few POD basis functions, we can capture most dynamics of the system. Thus, it can provide significant savings over existing numerical methods in solving the FKE. We should point out the number of the POD basis functions depends on the decay speed of the eigenvalues of the correlation matrix (9) and is problem-dependent. Due to its energy-minimizing property in the sense that the POD basis functions provide the best approximation to the solution snapshots, our POD algorithm always provides computational savings over existing numerical methods. After the POD basis functions have been constructed, we only need to solve a much smaller-scaled FKE in the off-line stage and much fewer numerical integrations in the on-line stage. We shall demonstrate the performance of our algorithm through numerical experiments in section V.

The rest of the paper is organized as follows. In section II, we give the basic idea of the POD method and the well-posedness of the pathwise robust DMZ equation. In section III, we propose the POD method of solving the FKE. More details about our POD algorithm, including the off-line and on-line computing, will be discussed. Convergence and effectiveness analysis of the proposed method will be discussed in section IV. In section V, we present numerical results to demonstrate the accuracy and efficiency of our method. Conclusions are drawn in section VI.

II. Preliminaries

In this section, we shall introduce the POD first, which has been used to study the turbulence in fluid mechanics in the beginning. The method of snapshots are used in our POD method to construct the basis functions. Moreover, the existence and uniqueness of the solution to DMZ equation has been discussed.

A. Proper orthogonal decomposition

The POD, also known as Karhunen-Loève expansion in stochastic process and signal analysis [20], [22], or the principal component analysis in statistics [1], or singular value decomposition in linear algebra, or the method of empirical orthogonal functions in geophysical fluid dynamics [29], [14]. The POD method has firstly been introduced in solving the turbulence in fluid dynamics. It aims to generate optimally ordered orthonormal basis functions in the least squares sense for a given set of theoretical, experimental or computational data. ROMs or surrogate models are then obtained by truncating this optimal basis functions, which provide considerable computational savings over the original high-dimensional problems. We refer the interested readers to [32], [6], [33], [16], [37] and references therein for more details.

Let X be a Hilbert space equipped with the inner product $(\cdot,\cdot)_X$ and $u(\cdot,t)\in X$, $t\in[0,T]$ be the solution of a dynamic system. In practice, we approximate the space X with a linear finite dimensional space V with $dimV = d$, where d represents the degree of freedom of the solution space. We should point out that d can be extremely large for high-dimensional problem. Given a set of snapshot of solutions, a linear space V can be spanned, denoted as

$$V = \text{span}\{u(\cdot,t_1), u(\cdot,t_2), ..., u(\cdot,t_N)\}, \tag{7}$$

where $t_1,\cdots,t_N \in [0,T]$ are different time instances. The POD method aims to build a set of low-dimensional basis functions $\{\varphi_1(\cdot),\varphi_2(\cdot),...,\varphi_r(\cdot)\}$ with $r \ll \min(N,d)$ that optimally approximates the input solution snapshots. The optimality means that given any integer r and linear independent basis $\{\varphi_k(x)\}_{k=1}^r$, the POD basis functions minimize the following error

$$\frac{1}{N}\sum_{i=1}^{N}\Big|\Big|u(\cdot,t_i)-\sum_{k=1}^{r}(u(\cdot,t_i),\varphi_k(\cdot))_X\varphi_k(\cdot)\Big|\Big|_X^2, \tag{8}$$

subject to the constrains that $(\varphi_m(\cdot), \varphi_n(\cdot))_X = \delta_{mn}$, $1 \le m, n \le r$, where $\delta_{mn} = 1$ if $m = n$, otherwise $\delta_{mn} = 0$.

Using the *method of snapshot* proposed by Sirovich [32], we know that the optimization problem (8) can be reduced to an eigenvalue problem

$$Kv = \lambda v, \tag{9}$$

where $K \in R^{N\times N}$ is the correlation matrix with (i,j)-element $K_{ij} = \frac{1}{N}(u(\cdot,t_i), u(\cdot,t_j))_X$. We sort the eigenvalues in a decreasing order as $\lambda_1 \ge \lambda_2 \ge ... \ge \lambda_N > 0$ and the corresponding eigenvectors are denoted by v_k, $k = 1, ..., N$. It can be shown that the POD basis functions are constructed by

$$\varphi_k(\cdot) = \frac{1}{\sqrt{\lambda_k}} \sum_{j=1}^{N} (v_k)_j u(\cdot, t_j), \quad 1 \le k \le N, \tag{10}$$

where $(v_k)_j$ is the j-th component of the eigenvector v_k. The basis functions $\{\varphi_k\}_{k=1}^r$ minimizes the error (8). This fact as well as the error formula were proved in [15].

Proposition II.1 ([15]). *Let $\lambda_1 \ge \lambda_2 \ge ... \ge \lambda_N > 0$ denote the positive eigenvalues of K in (9). Then $\{\varphi_k\}_{k=1}^r$ constructed according to (10) is the set of POD basis functions of rank $r \le N$, and we have the following error formula:*

$$\frac{1}{N}\sum_{i=1}^{N} \left|\left| u(\cdot,t_i) - \sum_{k=1}^{r} (u(\cdot,t_i), \varphi_k(\cdot))_X \varphi_k(\cdot) \right|\right|_X^2 = \sum_{k=r+1}^{d} \lambda_k. \tag{11}$$

In practice, we shall make use of the decay property of eigenvalues in λ_k and choose the first r dominant eigenvalues such that the ratio $\rho = \frac{\sum_{k=1}^{r}\lambda_k}{\sum_{k=1}^{N}\lambda_k}$ is big enough to achieve an expected accuracy, for instance $\rho = 99\%$. One would prefer the eigenvalues decays as fast as possible so that the fewer POD basis functions can ensure the higher accuracy.

In the sequel, we shall use the method of snapshot to extract dominant POD basis functions from the solution snapshots and generate a low-dimensional subspace to approximate solutions of FKE in our NLF problems. More details will be provided in the Section III-B.

B. The pathwise robust DMZ equation

As we briefly mentioned in the introduction section that the solution of the DMZ equation (2) is the key to solve the NLF problems completely. However, it is impractical to solve in an efficient way. With a given observation path, one can derive the pathwise robust DMZ equation (5) easily with an exponential transform (4). The existence and uniqueness of (5) has been investigated by many researchers. The well-posedness is guaranteed when the drift term $f \in C^1$ and the observation term $h \in C^2$ are bounded in [30]. Later on, similar results were obtained under weaker conditions. For instance, the well-posedness results on the pathwise-robust DMZ equation with a class of unbounded coefficients were obtained in [4], [11], but the results were for one-dimensional case. In [34], the third author of this paper and his collaborator established the well-posedness result under the condition that f and g have at most linear growth. The second and third author of this paper used more delicate analysis to give a time-dependent analogous well-posedness result to the pathwise-robust DMZ equation under some mild growth conditions on f and h in [24].

Although compared to the DMZ equation (2), the pathwise robust DMZ equation (5) should be easier to solve, since the stochastic term has been transformed into the coefficients, it is still difficult to obtain an analytic solution in general. So many efforts have been devoted to develop efficient and robust numerical methods to solve the FKE equation (6), see [5], [28], [17], [13], [23] and references therein.

III. Our POD algorithm to solve the NLF problems

Our POD algorithm consists of off-line and on-line computing stages. The off-line computing means that it can be performed without any on-line observation or experimental data, while the on-line computing needs the observation data that is only available during the experiment.

The main idea of our algorithm is to pre-construct a set of POD basis functions in the off-line stage by the method of snapshots (10). The FKE (6) with initial conditions to be the POD basis functions are solved by FDM. These data are stored for on-line synchronization. We remark that other numerical methods, such as finite element method and spectral method will also work, but we will not discuss them further in detail. The choice of numerical method is not crucial as all these computations are implemented in the off-line stage. Once we get the

solution snapshots, we compute the POD basis functions using the method of snapshot (10).

A. Off-line computing

Let us assume that the observation time sequences $0 = t_0 < t_1 < \cdots < t_{N_t} = T$ are given. But the observation data $\{y_{t_j}\}$ at each observation time t_j, $j = 0, ..., N_t$ are unknown until the on-line experiment runs.

Firstly, we generate a set of Monte Carlo realizations of the random observations $\{y_{t_j}(\omega_i)\}$ with $0 \le j \le N_t, 1 \le i \le N_{mc}$, and use FDM to solve the FKE (6) along each sample path of the random observation. This procedure provides us sufficient amount of snapshots $\mathcal{U} = \{u(t_j, \cdot, \omega_i)\}$, with cardinality $\sharp(\mathcal{U}) = (N_t + 1)N_{mc}$. These solution snapshots are assumed to capture the information of the solution space (or manifold) of the FKE (6) well. We remark that Monte Carlo realizations $\{y_{t_j}(\omega_i)\}$ are served as training purpose. In practice, we can use historically collected data to compute the solution snapshots.

Then, we apply the method of snapshot to construct the POD basis functions from the solution snapshots $\mathcal{U}$, where $\{u(\cdot, t_i)\}_{i=1}^N$ in (10) are replaced by $\mathcal{U}$ here. In our algorithm, we compute the ratio of the partial sum of the eigenvalues and total sum of the eigenvalues

$$\rho = \frac{\sum_{i=1}^{N_m} \lambda_i}{\sum_{i=1}^{\sharp(\mathcal{U})} \lambda_i}, \tag{12}$$

where λ_i's are the eigenvalues (in the descending order) of the square correlation matrix K in (9) of size $(N_t + 1)N_{mc}$. The number of the POD basis functions N_m is decided according to some prescribed error tolerance. Namely, we choose the smallest N_m such that the ratio ρ exceeds the prescribed error threshold, say $\rho > 99\%$. In our numerical experiments, we observed that in the asymptotic regime, the accumulated ratio (12) obtained using our POD basis functions can achieve exponential decay properly, i.e.,

$$\rho \sim 1 - \mathrm{e}^{-cN_m}. \tag{13}$$

This can significantly reduce the number of the POD basis and the on-line computational cost. We shall show some numerical results in section V-B to demonstrate this exponential decay behavior.

Let us denote by $\{\varphi_k(x)\}$, $k = 1, ..., N_m$, the POD basis functions obtained from the solution snapshots. In off-line computing, we also compute the propagation of the POD basis functions, see [25]. Specifically, for each POD basis $\varphi_k(x)$, we solve the initial value problem

$$\begin{cases} \frac{\partial \phi}{\partial t}(x,t) = (\mathcal{L} - \frac{1}{2}h^T S^{-1} h)\phi(x,t)dt, & \text{on } [t_{j-1}, t_j], \\ \phi(x, t_{j-1}) = \varphi_k(x), & k = 1, ..., N_m, \end{cases} \tag{14}$$

by FDM using finer time step. In the sequel, we shall use the notation $I^{[t_{j-1},t_j]}$ to denote the integrator or propagator defined by solving (14), namely, $\phi(x, t_j) = I^{[t_{j-1},t_j]}\varphi_k(x)$. Moreover, if (14) is time-invariant and the observation intervals are uniform, i.e., $t_{j+1} = t_j + \Delta t$, $\forall j$, we only need to calculate the propagator (14) once. For the sake of concise notation, we shall use $I^{\Delta t}$, instead of $I^{[t_{j-1},t_j]}$, $\forall j$ in this case.

The merit of our method is to pre-compute the solutions of (14) at each time interval and obtain $\{I^{\triangle t}\varphi_k(x)\}$, $k = 1, \cdots, N_m$. These data should be stored in preparation for the on-line synchronization. In the general time-dependent case, which means the operator $(\mathcal{L} - \frac{1}{2}h^T S^{-1} h)$ depends on time t, $I^{[t_{j-1},t_j]}\varphi_k(x)$ are different in each time interval $[t_{j-1}, t_j]$ and all of them should be pre-computed and stored. Though we need more storage costs, it is feasible in engineering application and most importantly it will provide significant savings in the on-line computation.

B. On-line computing

In this subsection, we shall demonstrate that using the POD basis functions and their pre-computed time integration $I^{[t_{j-1},t_j]}\varphi_k(x)$, we can achieve fast computing in the on-line stage.

Let $u(x, 0)$ denote the distribution of the initial state x_0. In each time interval $[t_{j-1}, t_j]$, $j = 1, ..., N_t$, at time t_{j-1}, we project the initial condition $u(x, t_{j-1}) \in L^2(D)$ onto the POD basis functions $\{\varphi_k(x)\}$ and obtain $u(x, t_{j-1}) \approx \sum_{k=1}^{N_m} \hat{u}_k(t_{j-1})\varphi_k(x)$, where $\hat{u}_k(t_{j-1}) = (u(\cdot, t_{j-1}), \varphi_k)_{L^2(D)}$ are the projection coefficients, and $D \subset \mathbb{R}^n$ is the physical domain of the states. Then, using our pre-computed propagator, we get the solution at time t_j, i.e.,

$$u^-(x, t_j) \approx \sum_{k=1}^{N_m} \hat{u}_k(t_{j-1}) I^{[t_{j-1},t_j]}\varphi_k(x), \tag{15}$$

where $u^-(x, t_j)$ denotes the a priori solution before updating with the observation y_{t_j}. When the observation

y_{t_j} is available, we update $u^-(x,t_j)$ by

$$u(x,t_j)=\exp\left[h^T(x,t_j)S^{-1}(t_j)(y_{t_j}-y_{t_{j-1}})\right]u^-(x,t_j). \tag{16}$$

After we get the solution $u(x,t_j)$ at time t_j, we again project the solution $u(x,t_j)$ onto the POD basis functions $\{\varphi_k(x)\}$ and repeat the procedure in (15) and (16) to continue our algorithm.

C. The POD algorithm

In this subsection, we summarize the off-line and on-line stages in our algorithm in Algorithm 1 and 2, respectively. The performance of our numerical method, especially the real-time manner, will be demonstrated in section V.

Algorithm 1 Off-line computing

for $i=1\to N_{mc}$ **do**
  Generate particle paths $\{x_{t_j}(\omega_i)\}$ and observations $\{y_{t_j}(\omega_i)\}$.
  Compute the solution of the pathwise robust DMZ equation (5), denoted as $u(t,x,\omega_i)$.
  Store the snapshots of u as $\mathcal{U}=\{u(t_j,\cdot,\omega_i)\}_{i,j}$, $j=1,\cdots,N_t$.
end for
Run the method of snapshots: compute the SVD of the correlation matrix K, where the eigen-pairs are denoted as (λ_k,v_k), $k=1,\cdots,(N_t+1)N_{mc}$.
Set a tolerance tol_ρ, and $\rho=0$.
while $\rho<tol_\rho$ **do**
  Increase N_m and calculate $\rho=\frac{\sum_{k=1}^{N_m}\lambda_k}{\sum_{k=1}^{\sharp(\mathcal{U})}\lambda_k}$.
end while
Store the first N_m eigen-pairs $\{\lambda_k,v_k\}_{k=1}^{N_m}$.
for $k=1\to N_m$ **do**
  Construct the POD basis functions $\{\varphi_k\}_{k=1}^{N_m}$.
  Solve the initial value problem (14) by FDM, and get $I^{[t_{j-1},t_j]}\varphi_k(x)$, $j=1,\cdots,N_t$.
  Store $I^{[t_{j-1},t_j]}\varphi_k$.
end for

IV. Convergence Analysis

A. The connection with splitting-up method in [5]

Our POD algorithm can be viewed as an improved version of the on- and off-line algorithm developed in [24]. The difference is that the basis functions here are constructed after training by the snapshot solutions.

Algorithm 2 On-line computing

Set up the initial distribution from x_0.
for $i=1\to N_t$ **do**
  Project $u(\cdot,t_{i-1})$ onto the POD basis functions, and obtain the a priori solution at t_i:

$$u^-(x,t_i)=\sum_{j=1}^{N_m}(u(\cdot,t_{i-1}),\varphi_j)_{L^2(D)}I^{[t_{i-1},t_i]}\varphi_j(x).$$

  Assimilate the new observation data y_{t_i} into the a priori solution $u^-(x,t_i)$:

$$u(x,t_i)=\exp[h^T(x,t_i)S^{-1}(t_i)(y_{t_i}-y_{t_{i-1}})]u^-(x,t_i).$$

  Calculate related statistics by using $u(x,t_i)$ as the unnormalized density function at time t_i.
end for

The convergence analysis in [24] is based on a given realization of observation. In this subsection, we shall point out the connection between the on- and off-line algorithm in [24] and the splitting-up method in [5], so that the convergence in $L_F^2(0,T;H^1(\mathbb{R}^d))$ is applicable in our POD method.

Let us assume that the observation time sequences are uniform, namely $t_{j+1}-t_j=\Delta t$, $j=0,...,N_t-1$. The observation data at time t_j is denoted by y_{t_j} and $\Delta y_j=y_{t_j}-y_{t_{j-1}}$. Let us recall the splitting-up method briefly. To be consistent with the settings in [5], we assume in this subsection that $S=I$, the identity matrix. The DMZ equation (2) has been decomposed into two processes U and U^- in the time intervals $[t_{i-1},t_i)$, $i=1,\cdots,N_t$, which satisfy

$$dU(t)=\left(\mathcal{L}U-\frac{\mu}{2}U\right)dt \tag{17}$$

$$U(t_{i-1})=\begin{cases}U^-(t_{i-1}), & \text{if } i=2,3,\cdots,N_t\\ \pi_0, & \text{if } i=1\end{cases}$$

$$\begin{aligned}dU^-(t)+\frac{\mu}{2}U^-dt=&U^-h^Tdw_t\\ =&U^-h^Tdy_t-U^-h^Thdt\\ U^-(t_{i-1})=&U(t_{i-1}),\end{aligned} \tag{18}$$

where $\mathcal{L}$ is the operator in (3) and π_0 is the unnormalized conditional density function of the initial state x_0. Notice the following two important facts:

1) U satisfies FKE (6) with $\mu=h^Th$ in (17).
2) U^- can be solved explicitly, i.e.

$$U^-(t)=U^-(t_{i-1})e^{\int_{t_{i-1}}^t h^Tdy_s+\frac{1}{2}\int_{t_{i-1}}^t(h^Th-\mu)ds}.$$

If $\mu = h^T h$, then

$$\begin{aligned} U^-(t) =& U^-(t_{i-1}) e^{\int_{t_{i-1}}^{t} h^T(s) dy_s} \\ \approx& U^-(t_{i-1}) e^{h^T(t_{i-1}) \Delta y_i}. \end{aligned}$$

U and U^- are used to denote the solutions before numerical discretization, while u and u^- are those after numerical discretization.

B. Convergence analysis

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. Let us make the following generic assumptions on the drift and observation terms as those in [5].

[As-1] The drift term and the diffusion term are bounded, i.e.

$$\begin{aligned} f \in& L^\infty(\mathbb{R}^n \times (0,\infty), \mathbb{R}^n), \\ g \in& L^\infty(\mathbb{R}^n \times (0,\infty); L(\mathbb{R}^n, \mathbb{R}^n)), \end{aligned}$$

with f and g be Lipschitz in x, uniformly in t.

[As-2] The observation term is also bounded, i.e. $h \in L^\infty(\mathbb{R}^n \times (0,\infty); \mathbb{R}^m)$.

[As-3] The operator gQg^T is uniformly elliptic, i.e. for all $\xi \in \mathbb{R}^n$, there exists a constant $\alpha > 0$ such that

$$\xi^T (gQg^T) \xi \geq \alpha |\xi|^2.$$

Remark IV.1. Although [As-1] and [As-2] seem to be very restrictive, as [5] claimed in the end, "this limitation is purely technical". For further discussions on the growth of f and h, we refer the interested readers to [24] and references therein.

Theorem IV.1 (Theorem 3.1, [5]). *Assume* [As-1]-[As-3] *hold, then we have*

1) $U, U^- \to \sigma$ *as* $\Delta t \to 0$ *in* $L^2_F(0,T; H^1(\mathbb{R}^n))$ *and* $L^2_F(0,T; L^2(\mathbb{R}^n))$, *respectively;*
2) $U(t), U^-(t) \to \sigma(t)$ *as* $\Delta t \to 0$ *in* $L^2(\Omega, \mathcal{A}, \mathbb{P}; L^2(\mathbb{R}^n))$, $\forall t \in [0,T)$*;*
3) $U(T), U^-(T) \to \sigma(T)$ *as* $\Delta t \to 0$ *in* $L^2(\Omega, \mathcal{A}, \mathbb{P}; L^2(\mathbb{R}^n))$*;*

where σ *is the solution to the DMZ equation* (2)*, the norms of* $L^2_F(0,T;V)$ *and* $L^2(\Omega, \mathcal{A}, \mathbb{P}; V)$ *are defined as*

$$\begin{aligned} ||\sigma||^2_{L^2_F(0,T;V)} =& \mathbb{E}\left[\int_0^T ||\sigma||^2_V dt \right], \qquad (19) \\ ||\sigma||^2_{L^2(\Omega,\mathcal{A},\mathbb{P};V)}(t) =& \mathbb{E}||\sigma||^2_V(t), \end{aligned}$$

where V *is some function space in concern. Here,* $V = L^2(\mathbb{R}^n)$ *or* $V = H^1(\mathbb{R}^n)$.

Compared with the splitting-up algorithm, further approximation in our POD method has been made in solving for U, where N_m POD basis functions $\{\varphi_k\}_{k=1}^{N_m}$ have been constructed and used to present U. Let us denote the approximate solution by U^{N_m}.

We expect that for fixed Δt, $U^{N_m} \to U$ in $H^1(\mathbb{R}^n)$, as $N_m \to \infty$. In the following analysis, we restrict ourselves with unbounded physical domain $D = \mathbb{R}^n$, yet the similar argument can be applied to a bounded domain $D \subset \mathbb{R}^n$, see Remark IV.3. Suppose instead of the N_m POD basis functions, we prescribe a set of N_m orthonormal basis in $H^1(\mathbb{R}^n)$, for example the generalized Hermite functions [26]:

$$\mathcal{H}_k^{\alpha,\beta}(x) = \left(\frac{\alpha}{2^k k! \sqrt{\pi}} \right)^{\frac{1}{2}} H_k(\alpha(x-\beta)) e^{-\frac{1}{2}\alpha^2 (x-\beta)^2},$$

where $H_n(x)$ are the univariate physical Hermite polynomials, α, β are two parameters. We define the n-dimensional generalized Hermite functions as

$$\mathcal{H}_{\mathbf{k}}^{\alpha,\beta}(\mathbf{x}) := \prod_{j=1}^{n} \mathcal{H}_{k_j}^{\alpha_j,\beta_j}(x_j),$$

where $\mathbf{x} \in \mathbb{R}^n$. It is clear to see that $\{\mathcal{H}_{\mathbf{k}}^{\alpha,\beta}\}_{\mathbf{k} \in \mathbb{N}_0^n}$ forms the orthonormal basis of $L^2(\mathbb{R}^n)$ and also those of $H^r(\mathbb{R}^n)$, for any $r \in \mathbb{N}$, where $\mathbb{N}_0$ is the natural numbers including zero. Suppose the prescribed orthonormal basis are $\left\{ \mathcal{H}_{\mathbf{k}}^{\alpha,\beta}(\mathbf{x}) \right\}_{\mathbf{k} \in \Omega_{N_m}}$, where $\Omega_{N_m} := \left\{ \mathbf{k} : \ |\mathbf{k}|_\infty \leq N_m^{\frac{1}{n}} \right\}$ with $|\mathbf{k}|_\infty = \max_{i \in \{1,\cdots,n\}} k_i$.

Theorem IV.2 (Theorem 2.1, [26]). *Given* $U \in W^m_{\alpha,\beta}(\mathbb{R}^n)$*, we have for any* $0 \leq l \leq r$*,*

$$\left|\left| P_{N_m}^{\alpha,\beta} U - U \right|\right|_{W^l_{\alpha,\beta}(\mathbb{R}^n)} \lesssim N_m^{\frac{l-r}{2n}} |U|_{W^r_{\alpha,\beta}(\mathbb{R}^n)}, \qquad (20)$$

where the projection operator

$$P_{N_m}^{\alpha,\beta} : \ W^l_{\alpha,\beta}(\mathbb{R}^n) \to \text{span} \left\{ \mathcal{H}_{\mathbf{k}}^{\alpha,\beta}, \mathbf{k} \in \Omega_{N_m} \right\}$$

and the norm and seminorm of $W^r_{\alpha,\beta}(\mathbb{R}^n)$ *are defined as*

$$\begin{aligned} ||U||^2_{W^r_{\alpha,\beta}(\mathbb{R}^n)} :=& \sum_{0 \leq |\mathbf{k}|_1 \leq r} ||\mathcal{D}^{\mathbf{r}}_{\mathbf{x}} U||^2, \\ |U|^2_{W^r_{\alpha,\beta}(\mathbb{R}^n)} :=& \sum_{j=1}^{n} ||\mathcal{D}^r_{x_j} U||^2, \end{aligned}$$

with $\mathcal{D}^{\mathbf{r}}_{\mathbf{x}} := \prod_{i=1}^n \mathcal{D}^{r_i}_{x_i}$, $\mathcal{D}^{r_i}_{x_i} = \partial_{x_i} + \alpha_i^2 (x_i - \beta_i)$.

Remark IV.2. The space $W^0_{\alpha,\beta}(\mathbb{R}^n) = L^2(\mathbb{R}^n)$ and $W^r_{\alpha,\beta}(\mathbb{R}^n) \subset H^r(\mathbb{R}^n)$ by Corollary 3.2, [26].

It is clear to see that if the function itself is extremely smooth, then the projection error decreases faster than any degree of polynomials of N_m. That is, it may present exponential convergence with respect to N_m. Remember that the basis functions here are prescribed without any information of the solution U. One would expect intuitively the elaborately constructed N_m POD basis functions (after training) should carry sufficient information of the solution and yield smaller projection error in arbitrary norm, i.e.

$$\begin{aligned}\left|\left|P_\Phi^{\alpha,\beta}U-U\right|\right|_{W^l_{\alpha,\beta}(\mathbb{R}^n)} &\lesssim \left|\left|P_{N_m}^{\alpha,\beta}U-U\right|\right|_{W^l_{\alpha,\beta}(\mathbb{R}^n)}\\ &\overset{(20)}{\lesssim} N_m^{\frac{l-r}{2n}}|U|_{W^r_{\alpha,\beta}(\mathbb{R}^n)},\end{aligned}$$

where

$$P_\Phi^{\alpha,\beta}:\ W^l_{\alpha,\beta}(\mathbb{R}^n)\to \text{span}\{\varphi_k(x), k=1,\cdots,N_m\}$$

with $\{\varphi_k(x)\}$ be the POD basis functions. Therefore, if for all $w\in\Omega$, for any $t\in[0,T]$, $U\in W^r_{\alpha,\beta}(\mathbb{R}^n)$, then we have

$$\begin{aligned}&\left|\left|P_\Phi^{\alpha,\beta}U-U\right|\right|^2_{W^l_{\alpha,\beta}(\mathbb{R}^n)}(\omega,t)\\ &\quad\lesssim N_m^{\frac{l-r}{2n}}|U(\omega)|_{W^r_{\alpha,\beta}(\mathbb{R}^n)}(\omega,t),\end{aligned}$$

where Ω denotes the event space of the randomness from the observation. Notice that U satisfies the parabolic PDE which exactly in the form in section 3 [26]. Let us denote the numerical solution obtained in span$\left\{\mathcal{H}^{\alpha,\beta}_{\mathbf{k}}:\ \mathbf{k}\in\Omega_{N_m}\right\}$ as U^{N_m}. With the similar argument as in [26], we assert that for all $w\in\Omega$, $t\in[0,T]$, we have

$$\begin{aligned}&\left|\left|U^{N_m}-U\right|\right|^2_{H^1(\mathbb{R}^n)}(\omega,t)\\ &\quad\le \left|\left|U^{N_m}-U\right|\right|^2_{W^1_{\alpha,\beta}(\mathbb{R}^n)}(\omega,t)\\ &\quad\lesssim \left|\left|P_\Phi^{\alpha,\beta}U-U\right|\right|^2_{W^1_{\alpha,\beta}(\mathbb{R}^n)}(\omega,t)\\ &\qquad+\left|\left|P_\Phi^{\alpha,\beta}U-U^{N_m}\right|\right|^2_{W^1_{\alpha,\beta}(\mathbb{R}^n)}(\omega,t)\\ &\quad\lesssim N_m^{\frac{1-r}{n}}\left\{|U|^2_{W^r_{\alpha,\beta}(\mathbb{R}^n)}(\omega,t)\right.\\ &\qquad\left.+\int_0^t|U|^2_{W^r_{\alpha,\beta}(\mathbb{R}^n)}(\omega,s)ds\right\},\end{aligned}\tag{21}$$

where the first inequality follows by Remark IV.2.

Proposition IV.3. *If for all $w\in\Omega$, $t\in[0,T]$, $U\in L^2_F(0,T;W^r_{\alpha,\beta}(\mathbb{R}^n))$, then we get*

$$\begin{aligned}&\left|\left|U^{N_m}-U\right|\right|^2_{L^2_F(0,T;H^1(\mathbb{R}^n))}\\ &\quad\lesssim N_m^{\frac{1-r}{n}}(1+T)\left|\left|U\right|\right|^2_{L^2_F(0,T;W^r_{\alpha,\beta}(\mathbb{R}^n))}.\end{aligned}\tag{22}$$

Proof. Integrating with respect to time from $t=0$ to T and taking the expectation on both sides of (21), the inequality (22) follows immediately. □

Combining Proposition IV.3 with Theorem IV.1, we get

Theorem IV.4. *Assume* [As-1]-[As-3] *hold, then we have*

1) *$U^{N_m}\to\sigma$ in $L^2_F(0,T;H^1(\mathbb{R}^n))$, as $N_m\to\infty$ and $\Delta t\to 0$ subsequently;*
2) *$U^{N_m}(t)\to U(t)$ in $L^2(\Omega,\mathcal{A},\mathbb{P};H^1(\mathbb{R}^n))$, for all $t\in[0,T]$.*

Remark IV.3. The similar result in Proposition IV.3 can be obtained for bounded domain $D\subset\mathbb{R}^n$. We refer the interested readers to [31].

V. Numerical Results

In this section, we are interested in investigating the approximation properties of our POD method and the computational savings over existing methods. The experiments are performed in two-dimensional NLF problems. We shall clarify the settings of these two problems first.

Example 1: Almost linear problem

This problem is modeled by a SDE in the Ito form below:

$$\begin{cases}dx_1=dv_1,\\ dx_2=dv_2,\\ dy_1=x_1(1+0.2\cos(x_2))dt+dw_1,\\ dy_2=x_2(1+0.2\cos(x_1))dt+dw_2,\end{cases}\tag{23}$$

where $E[d\mathbf{w}_t d\mathbf{w}_t^T]=I_2dt$, $E[d\mathbf{v}_t d\mathbf{v}_t^T]=0.1I_2dt$, with $\mathbf{w}=[w_1,w_2]^T$, $\mathbf{v}=[v_1,v_2]^T$, I_2 be the identity matrix of size 2×2. The states are two independent standard Brownian motions. The initial state is $\mathbf{x}(0)=[x_1(0),x_2(0)]^T=[1,1.2]^T$. We shall denote the state in vector form $\mathbf{x}(t)=[x_1(t),x_2(t)]^T$. The total experimental time is $T=20$.

Example 2: Cubic sensor problem

The observations in cubic sensor problem have higher nonlinearity than those in (23), which may cause problem when using the conventional extended Kalman filter

(EKF). It is modeled in the form below:

$$\begin{cases} dx_1 = (-0.4x_1 + 0.1x_2)dt + dv_1, \\ dx_2 = -0.6x_2 dt + dv_2, \\ dy_1 = (x_1^3 + x_2)dt + dw_1, \\ dy_2 = (x_2^3 + x_1)dt + dw_2, \end{cases} \tag{24}$$

where $E[d\mathbf{w}_t d\mathbf{w}_t^T] = I_2 dt$ and $E[d\mathbf{v}_t d\mathbf{v}_t^T] = 0.1 I_2 dt$. The initial state is $\mathbf{x}(0) = [x_1(0), x_2(0)]^T = [0.1, 0.05]^T$. The total experimental time is $T = 10$.

A. Comparison with existing methods

In this subsection, we shall mainly compare the estimation performance and real-time manner of our POD algorithm with the reference solutions in two examples (23) and (24) respectively.

In both examples, the real state is generated by solving the SDE (23) or (24) for $\mathbf{x}$ in the time interval $[0, T]$ (T=20 or 10) with time step $dt = 0.01$ using the Euler-Maruyama method [21]. This provides us the values of the real state at discrete times $t_j = jdt$, $j = 1, ..., 2000$ (or 1000).

Example 1: Almost linear problem

We use FDM to solve the FKE (6) online to obtain the reference solution. The spacial domain is $[-5, 5] \times [-5, 5]$ and is partitioned with 1-D mesh size $\Delta x = 10/128$. The unnormalized conditional density function of the initial state is $\sigma_0(\mathbf{x}) = \exp(-2|\mathbf{x}|^2)$. The Courant-Friedrichs-Lewy (CFL) stability condition of FDM is satisfied by choosing the time step as $\frac{dt}{10}$.

To obtain sufficient amount of snapshots, as described in Algorithm 1, we partition the time interval $[0, 20]$ with observation time step $\Delta t = 0.2$, generate $N_{mc} = 500$ random observations $\{y_{t_j}(\omega_i)\}$ with $1 \le j \le 100, 1 \le i \le 500$, and use FDM to solve FKE (6) along each sample path ω_i of the random observation. The POD basis functions are constructed as in (10). In Figure 1, we plot the estimations of both two states obtained by our POD algorithm with the number of the POD basis functions $N_m = 70$ and the reference solution in one realization. It seems that both methods give acceptable experimental results. Yet our algorithm gives significant computational savings over the reference method, that is, the CPU time of the reference method is 48.38s, while that of our algorithm is only 4.23s, which is almost $\frac{1}{12}$ of the former one. This is because the POD basis functions enable us to use the on- and off-line algorithm [24] and solve the NLF problems in a real-time manner. On the contrary, the reference solution by the algorithm in [24] using FDM can only be performed online. No off-line computation is available, like the one in [25]. Thus, the time saving is obvious. It is expected that the computational savings of our POD method can be more significant in solving higher dimensional NLF problems, which is our ongoing research.

We repeat the experiments for $N_{path} = 300$ times and record the mean square errors (MSEs) averaged over 300 sample paths. The MSE between $\mathbf{x}_1$ and $\mathbf{x}_2$ is defined as

$$MSE(\mathbf{x}_1, \mathbf{x}_2) := \frac{1}{300} \sum_{i=1}^{300} \frac{1}{N_t} \sum_{j=1}^{N_t} |\mathbf{x}_1^i(t_j) - \mathbf{x}_2^i(t_j)|,$$

where $|\cdot|$ is the Euclidean distance, for i-th sample path, $\mathbf{x}_j^i$, $j = 1, 2$ are the true state or the numerical estimation obtained by different methods, such as our algorithm and reference method. We find that the MSE between our POD algorithm and the reference method is 0.0126, while that between the reference method and the real state is 0.5991. On the contrary, the MSE between EKF and the real state is 0.9932. It shows that the further compression by POD method has comparable accuracy with the reference solution, which is only with the difference less than 2%.

Example 2: Cubic sensor problem

In this example, the reference solution is also obtained by using FDM to solve FKE (6). The spacial domain is $[-3, 3] \times [-3, 3]$ and is partitioned with 1-dimensional mesh size $\Delta x = 6/128$. The unnormalized conditional density function of the initial state is $\sigma_0(\mathbf{x}) = \exp(-|\mathbf{x}|^4/4)$. The time step is chosen to be $\frac{dt}{40}$ so that the CFL stability condition is satisfied. $N_m = 100$ POD basis functions are constructed according to (10) after the similar training in Example 1.

In Figure 2, we display the similar results as those in Figure 1. The widely used EKF has been included in this comparison. It is clear to see that EKF yields worse estimation than the other two at least for this realization. In fact, the high nonlinearity in observations will normally lead to bad performance in EKF. As to the real-time manner, the CPU time of the reference method is 99.83s, while that of our algorithm is only 2.21s.

Remark V.1. The CPU time of the reference solution in Example 2 is significantly longer than that in Example 1, since all the computations are carried online and the time

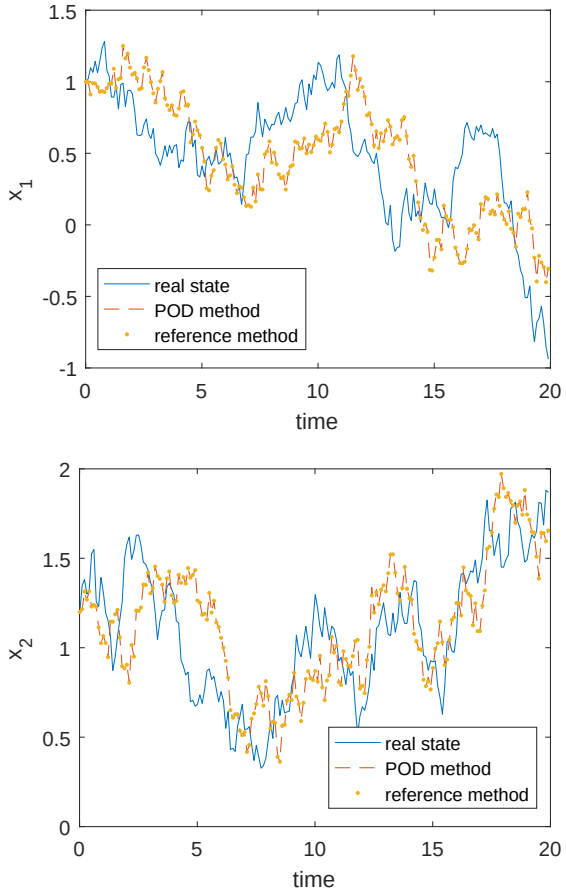


Fig. 1: The estimations of the almost linear problem (23) obtained by our POD method (in red dashed line) and the reference solution (in orange dot line) versus time have been plotted. The blue line is the true state generated by one realization.

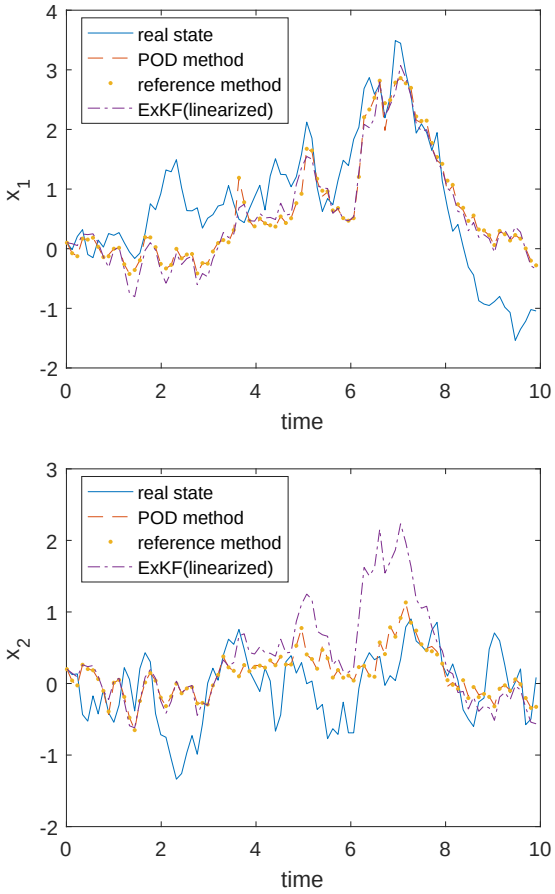


Fig. 2: The estimations of the cubic sensor problem (24) obtained by our POD method (in red dashed line), the reference solution (in orange dot line), and the EKF (in purple dashed-dot line) versus time have been plotted. The blue line is the true state generated by one realization.

discretization in this example is 4 times finer than that in Example 1. We believe that the finer time discretization is due to the higher nonlinearity.

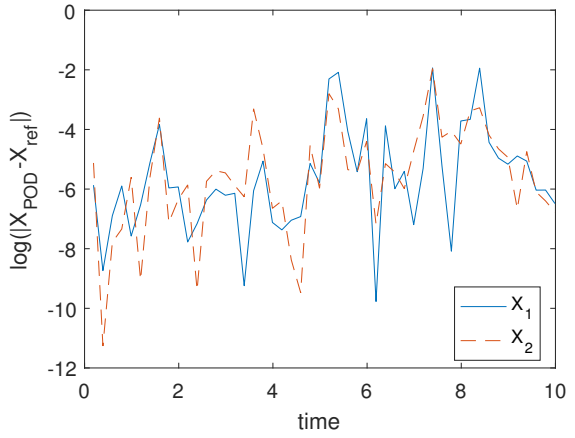
In Figure 1 and 2, the difference between our POD algorithm and the reference solution is too small to be distinguished by eyes. We shall quantify this in cubic sensor problem for one realization in Figure 3. To be more precise, let u_{ref} and u_{POD} be the numerical solutions of (6) using FDM and POD method respectively. We define the relative L^2 error as

$$err(t) := \frac{||u_{\text{ref}} - u_{\text{POD}}||_{L^2}(t)}{||u_{\text{ref}}||_{L^2}(t)}. \tag{25}$$

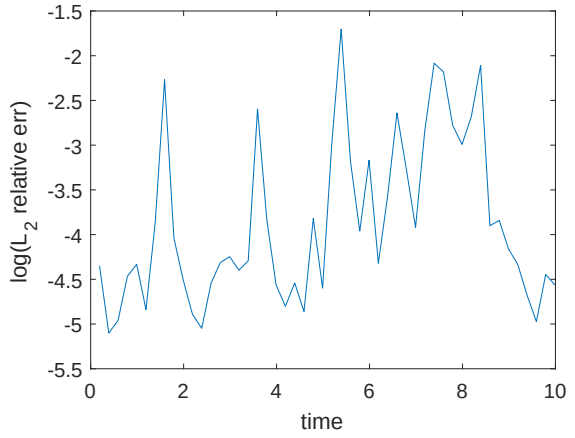
Let X_{ref} and X_{POD} be the expectation of the states with respect to different probability measures, i.e. $\mathbb{E}_{\text{ref}}(\mathbf{x})$ and $\mathbb{E}_{\text{POD}}(\mathbf{x})$, respectively. It is well known that X_{ref} and X_{POD} are minimal MSE estimations of the true state. One can find in Figure 3 that actually the relative L^2 error (25) of the unnormalized density functions is indeed small, which is roughly of order $\mathcal{O}(10^{-4}) \sim \mathcal{O}(10^{-3})$ in average, while the error between the minimal MSE estimations is of order $\mathcal{O}(10^{-6})$ averagely in time for both states x_1 and x_2.

B. More discussions on our POD algorithm

In our POD algorithm, there are still some parameters to be tuned in, for example, the number of the POD basis functions, the choice of the training solutions, etc. In this subsection, we shall do the numerical experiments mainly on Example 2, since both examples show similar



(a) Absolute value of the difference between two minimal MSE estimations based on two methods versus time is plotted. The blue line is for the state x_1, while the red dashed line is for x_2.



(b) Relative L^2 error $err(t)$ versus time is plotted.

Fig. 3: The comparisons between POD method and the reference solution in cubic sensor problem (24) are displayed for one realization.

behaviors and Example 2's low dimensional structure seems to be more difficult to be captured.

1) The decay property of the relative errors versus the number of the POD basis functions: It has been shown in Proposition II.1 that the relative L^2 error of the training solutions can be presented by the quantity $1-\rho$, where ρ is defined in (12):

$$\frac{\frac{1}{N}\sum_{i=1}^{N}\left|\left|u(\cdot,t_i)-\sum_{j=1}^{N_m}(u(\cdot,t_i),\varphi_j(\cdot))_{L^2}\varphi_j(\cdot)\right|\right|_{L^2}^2}{\frac{1}{N}\sum_{i=1}^{N}||u(\cdot,t_i)||_{L^2}^2} = \frac{\sum_{j=N_m+1}^{d}\lambda_j}{\sum_{j=1}^{N}\lambda_j} = 1-\rho.$$

where N and N_m are the total number of snapshots and that of the POD basis. In Figure 4, we plot the quantity $1-\rho$ versus the number of POD basis. We use regression to fit the data and find the decay speed of the quantity $1-\rho$ is proportional to $\exp(-C_1 N_m)$, with $C_1 = 0.0422$.

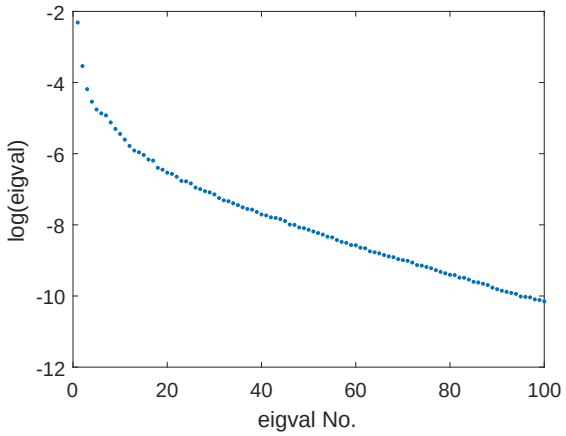


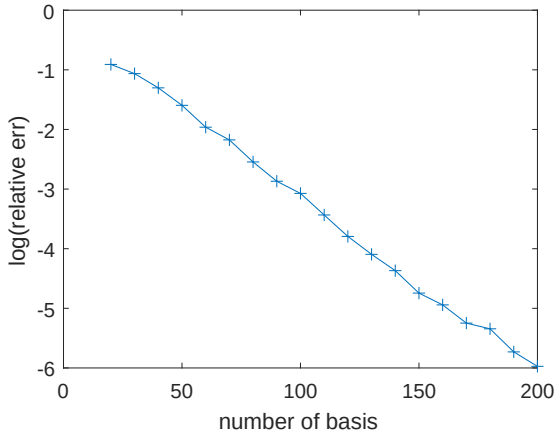
Fig. 4: The decay property of eigenvalues in the POD method.

In section IV, we show theoretically in Proposition IV.3 that the relative error

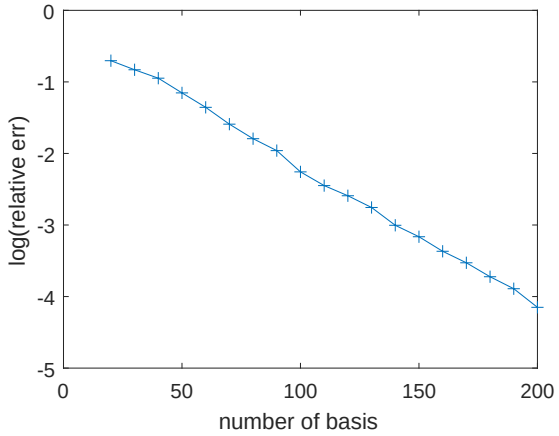
$$\frac{\left|\left|U^{N_m}-U\right|\right|^2_{L^2_F(0,T;H^1(\mathbb{R}^n))}}{||U||^2_{L^2_F(0,T;W^r_{\alpha,\beta}(\mathbb{R}^n))}} \tag{26}$$

is controlled by $N_m^{\frac{1-r}{n}}(1+T)$. In other words, if the reference solution is smooth enough, the relative error (26) can present exponential decay as $N_m \to \infty$. Here, we generate $N_{path} = 300$ sample paths $\mathbf{x}_t$ and observation paths $\mathbf{y}_t$. We record the relative L^2 error defined in (25) of the numerical solution obtained using fixed number of POD basis functions ranging from 1 to 200 and for each sample paths. For fixed number of POD basis functions, one averages the relative errors over all these 300 sample paths.

In Figure 5, we illustrate in both examples how the number of POD basis functions affects the averaged relative L^2 errors over $N_{path} = 300$ sample paths between two methods. We use regression to fit the data and find the decay rate of the relative error is proportional to $\exp(-C_2 N_m)$ with $C_2 = 0.0293$ and 0.0195 for Example 1 and 2 respectively, where N_m is the number of POD basis. The relatively slow decay in the cubic sensor problem may imply that it has more complicated structure than the almost linear problem. It is interesting



(a) Almost linear problem



(b) Cubic sensor problem

Fig. 5: The relative L^2 errors averaged over $N_{path} = 300$ sample paths versus the number of POD basis function is plotted.

to notice that C_1 in Figure 4 is almost twice of C_2 for the cubic sensor problem. This implies that on the average sense the POD basis constructed from random paths can capture most energy of the solution from other random paths. This also explains why in section V-A different N_m are chosen to guarantee $\mathcal{O}(10^{-2})$ relative error in two examples.

In Figure 6, we plot the relative L^2 error evolution of one realization in the cubic sensor problem versus different number of POD basis functions. One find that at each time discretization the error decays monotonically as the number of the POD basis functions increases. More significant observation is that just increasing the number of POD basis cannot improve resolutions, if

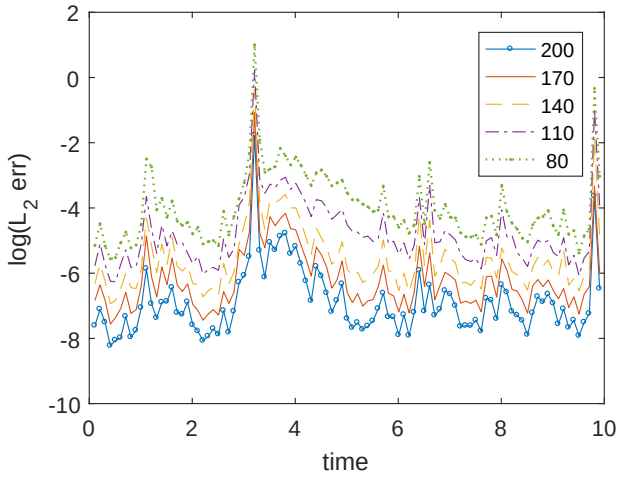


Fig. 6: The relative L^2 errors of one realization versus time are plotted with the number of POD basis functions being $N_m = 80, 110, 140, 170$ and 200.

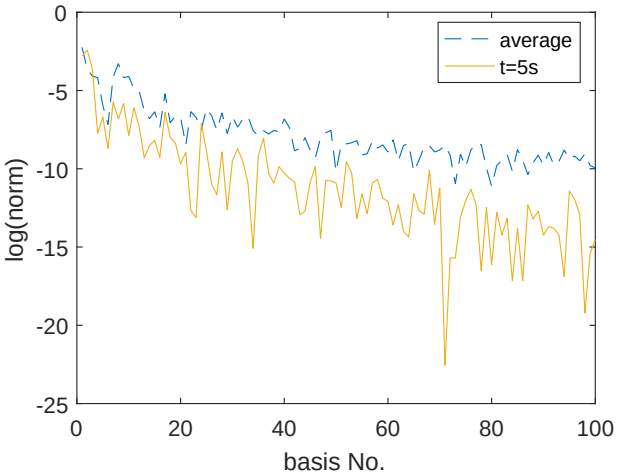


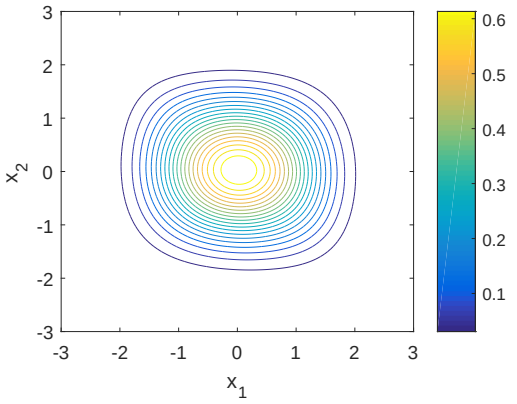
Fig. 7: The absolute value of the expansion coefficients of the solution to the POD basis functions in cubic sensor problem (24) versus the number of the POD basis functions are plotted for one realization. Orange line: that at time $t = 5$; blue dashed line: that averaged over t_j, $j = 1, \cdots, 10000$ in $[0, 10]$.

the POD basis functions have not contained enough information after the training process. This phenomena can be seen from Figure 6 at time instance around $t = 3.7$ and 9.8, where the peaking of the errors are not relieved even after doubling the number of POD basis functions.

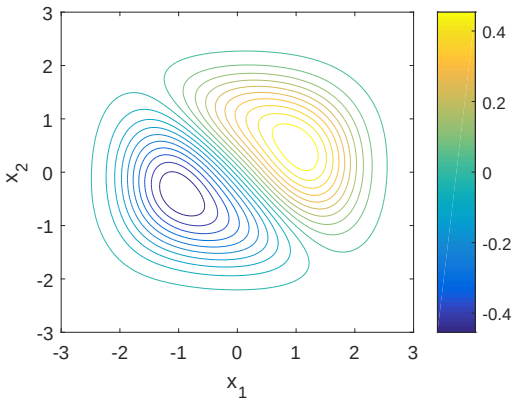
As one knows, for one particular realization of the state, the most key ingredient of our POD algorithm to present the solution well is to see whether the first few POD basis functions can capture the most energy of that solution. To investigate this property, we study the expansion coefficients of the solution on the POD basis functions. In Figure 7, we plot the absolute value of the expansion coefficients of the solution at time $t = 5$ and the average of those over time discretization t_j, $j = 1, 2, \cdots, 10000$. It is clear to see that the absolute values of the expansion coefficients decay exceptionally fast, which implies that the POD basis functions approximates the solution well, and with only a small number of POD basis functions it can efficiently capture the dynamics of the system. Notice that the absolute values of the coefficients do not decay monotonically, since with the same set of POD basis functions the ability of approximating solutions at each time discretization and realizations is of great difference. Yet the same trend can be observed in the average of the absolute values of expansion coefficients with less oscillations.

2) The selection of the training solutions: The construction of the POD basis functions depends highly on the training set. How the training set affects the POD basis functions? Recall that we generate $N_{mc} = 500$ sample paths of the states and the observations, and the snapshots are $\mathcal{U} = \{u(t_j, \cdot, \omega_i)\}$, here $\omega_i \in \Omega$, $i = 1, \cdots, N_{mc}$, $t_j = j\Delta t$, $j = 1, \cdots, N_t (= \frac{T}{\Delta t})$ with $\Delta t = 0.2$. We also try to generate less sample paths such as $N_{mc} = 125$ or $N_{mc} = 250$, and find that the first few dominant POD basis functions are indistinguishable from various N_{mc}.

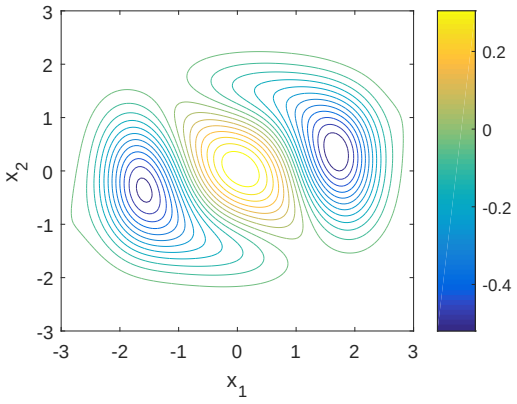
In Figure 9, we show the first six dominant POD basis functions obtained with $N_{mc} = 500$. The higher order of POD basis function is, the more local structures of the solutions have been captured. It would be interesting and challenging to generate the snapshots capable of capturing most of the variations of the solution space. This issue will be investigated in our future work, especially in higher-dimensional NLF problems.



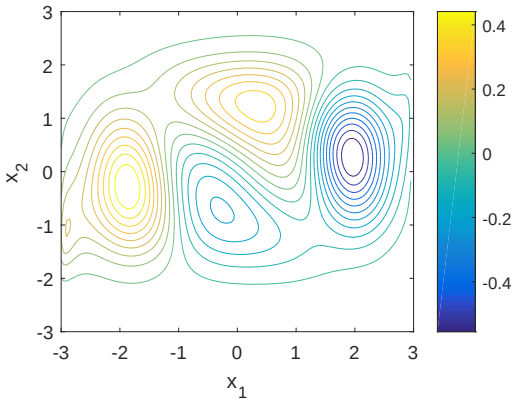
(a) First POD basis



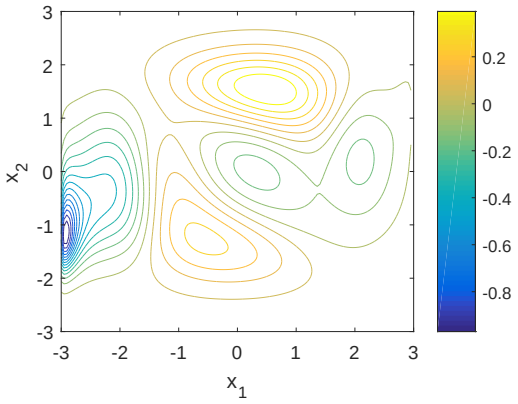
(b) Second POD basis



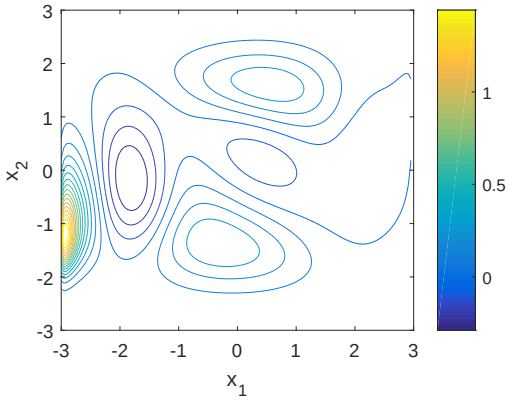
(c) Third POD basis



(d) Fourth POD basis



(e) Fifth POD basis



(f) Sixth POD basis

Fig. 9: The profiles of the first six POD basis functions in the cubic sensor problem (24).

VI. Conclusions

In this paper, we investigate the proper orthogonal decomposition (POD) method to numerically solve the forward Kolmogorov equation (FKE), which has important application in solving nonlinear filtering (NLF) problems. The POD method can be viewed as further compression in using on- and off-line algorithm [24]. The beforehand numerical experiments or history data is necessary as a training set. The low-dimensional structures in the solution space of the FKE has been trained and used to build the POD basis functions in advance. Combined with the on- and off-line algorithm, in the off-line stage, besides the construction of POD basis, we still need to compute the propagation of the POD basis functions according to the FKE equation. In the on-line stage, we only need to do numerical integrations, that is the expansion coefficients of the POD basis, and update with the new-coming observations. This algorithm enables us to solve the NLF problem in a real-time manner.

Under some generic assumptions as in [5], we provide the convergence analysis of our POD method theoretically. Two 2-dimensional NLF problems: almost linear problem and cubic sensor problem have been investigated in details. The theoretical convergence rate has been verified numerically. It is shown numerically that our POD algorithm yields as good approximations as the reference solution obtained by FDM. But our algorithm can be much more efficient, that is, more than 10 times faster in both examples. We expect even better performance of efficiency in higher-dimensional NLF problems, which is one of our future topics. Some further discussions on the POD algorithm, such as the choice of number of POD basis, the number of snapshots, etc, have been included. It seems that it is unnecessary to provide a huge amount of snapshots for training in our numerical experiments, but "how many and which" is an interesting question of practical importance.

References

[1] H. Abdi and L. Williams, "Principal component analysis," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433-459, 2010.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp.174-188, 2002.

[3] A. Bain and D. Crisan, *Fundamentals of stochastic filtering*, Berlin: Springer, 2009, vol. 60, Stochastic Modeling and Applied Probability.

[4] J. Baras, G. Blankenship, and W. Hopkins, "Existence, uniqueness, and asymptotic behavior of solutions to a class of Zakai equations with unbounded coefficients," *IEEE Trans. Automat. Control.*, vol. 28, no. 2, pp. 203-214, 1983.

[5] A. Bensoussan, R. Glowinski, and A. Rascanu, "Approximation of the Zakai equation by the splitting up method," *SIAM J. Control Optim.*, vol. 28, no. 6, pp. 1420-1431, 1990.

[6] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539-575, 1993.

[7] M. Cheng, T. Y. Hou, M. Yan, and Z. Zhang, "A data-driven stochastic method for elliptic PDEs with random coefficients," *SIAM/ASA J. Uncertain. Quantif.*, vol. 1, pp. 452-493, 2013.

[8] M. Cheng, T. Y. Hou, and Z. Zhang, "A dynamically bi-orthogonal method for stochastic partial differential equations I: derivation and algorithms," *J. Comput. Phys.*, vol. 242, pp. 843-868, 2013.

[9] M. Cheng, T. Y. Hou, and Z. Zhang, "A dynamically bi-orthogonal method for stochastic partial differential equations II: adaptivity and generalizations," *J. Comput. Phys.*, vol. 242, pp. 753-776, 2013.

[10] T. Duncan, "Probability densities for diffusion processes with applications to nonlinear filtering theory and detection theory," Technical report, Stanford Univ. CA, Stanford Electronics Labs, 1967.

[11] W. Fleming and S. Mitter, "Optimal control and nonlinear filtering for nondegenerate diffusion processes," *Stochastics*, vol. 8, no. 1, pp. 63-77, 1982.

[12] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 425-437, 2002.

[13] I. Gyöngy and N. Krylov, "On the splitting-up method and stochastic partial differential equations," *Ann. Probab.*, vol. 31, no. 2, pp. 564-591, 2003.

[14] A Hannachi, I. Jolliffe, and D. Stephenson, "Empirical orthogonal functions and related techniques in atmospheric science: A review," *International journal of climatology*, vol. 27, no. 9, pp. 1119-1152, 2007.

[15] P. Holmes, J. Lumley, and G. Berkooz, *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge University Press, 1998.

[16] T. Iliescu and Z. Wang, "Variational multiscale proper orthogonal decomposition: Convection-dominated convection-diffusion-reaction equations," *Math. Comp.*, vol. 82, no. 283, pp. 1357-1378, 2013.

[17] K. Itô, "Approximation of the Zakai equation for nonlinear filtering," *SIAM J. Control Optim.*, vol. 34, no. 2, pp. 620-634, 1996.

[18] K. Itô, *Diffusion processes*. Wiley Online Library, 1974.

[19] G. Kallianpur, *Stochastic filtering theory*, volume 13. Springer Science & Business Media, 2013.

[20] K. Karhunen, Uber lineare methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys.*, vol. 37, pp. 1-79, 1947.

[21] P. E. Kloeden and E. Platen, *Numerical solution of Stochastic Differential Equations*. Springer-Verlag Berlin, Heidelberg, 1992.

[22] M. Loève, *Probability theory. Vol. II, 4th ed. GTM. 46.* Springer-Verlag, 1978.

[23] S. Lototsky, R. Mikulevicius, and B. L. Rozovskii, "Nonlinear filtering revisited: a spectral approach," *SIAM J. Control Optim.*, vol. 35, no. 2, pp. 435-461, 1997.

[24] X. Luo and S. S. T. Yau, "Complete real time solution of the general nonlinear filtering problem without memory," *IEEE Trans. Automat. Control.*, vol. 58, no. 10, pp. 2563-2578, 2013.

[25] X. Luo and S. S. T. Yau, "Hermite spectral method to 1-D forward Kolmogorov equation and its application to nonlinear filtering problems," *IEEE Trans. Automat. Control.*, vol. 58, no. 10, pp. 2495-2507, 2013.

[26] X. Luo and S. S. T. Yau, "Hermite spectral method with hyperbolic cross approximations to high-dimensional parabolic PDEs," *SIAM J. Numer. Anal.*, vol. 51, no. 6, pp. 3186-3212, 2013.

[27] R. Mortensen, "Optimal control of continuous-time stochastic systems," Technical report, California Univ., Berkeley Electronics Research Lab,1966.

[28] N. Nagase, "Remarks on nonlinear stochastic partial differential equations: an application of the splitting-up method," *SIAM J. Control Optim.*, vol. 33, no. 6, pp. 1716-1730, 1995.

[29] G. North, T. Bell, R. Cahalan, and F. Moeng, "Sampling errors in the estimation of empirical orthogonal functions," *Monthly Weather Review*, vol. 110, no. 7, pp. 699-706, 1982.

[30] E. Pardoux, "Stochastic partial differential equations and filtering of diffusion processes," *Stochastics*, vol. 3, pp. 127-167, 1980.

[31] J. Shen and L.-L. Wang, "Sparse spectral approximations of high-dimensional problems based on hyperbolic cross," *SIAM J. Numer. Anal.*, vol. 48, no. 3, pp. 1087-1109, 2010.

[32] L. Sirovich, "Turbulence and the dynamics of coherent structures. I. Coherent structures," *Quart. Appl. Math.*, vol. 45, no. 3, pp. 561-571, 1987.

[33] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *AIAA J.*, vol. 40, no. 11, pp. 2323-2330, 2002.

[34] S.-T. Yau and S. S.-T. Yau, "Real time solution of the nonlinear filtering problem without memory II," *SIAM J. Control Optim.*, vol. 47, no. 1, pp. 163-195, 2008.

[35] M. Yueh, W. Lin, and S. T. Yau, "An efficient numerical method for solving high-dimensional nonlinear filtering problems," *Commun. Inf. Syst.*, vol. 14, no. 4, pp. 243-262, 2014.

[36] M. Zakai, "On the optimal filtering of diffusion processes," *Probab. Theory Related Fields*, vol. 11, no. 3, pp. 230-243, 1969.

[37] S. Zhu, L. Dedè, and A. Quarteroni, "Isogeometric analysis and proper orthogonal decomposition for parabolic problems," *Numer. Math.*, vol. 135, no. 2, pp. 333-370, 2017.

Z. Wang received his B.S. degree in mathematics from Tsinghua University, Beijing, P.R. China, in 2016. He is currently pursuing his Ph.D. degree in applied and computational mathematics from department of mathematics, the University of Hong Kong.

His research interests are numerical methods for stochastic differential equations and stochastic partial differential equations.

X. Luo (SM'15) received her first Ph.D. degree in mathematics from East China Normal University (ECNU), Shanghai, P.R. China in 2010 and her second Ph.D. degree in applied mathematics from University of Illinois at Chicago (UIC) in 2013. During her study as a Ph. D. candidate in ECNU, she visited the department of Mathematics, University of Connecticut in 2008-2009 and the department of mathematics, statistics and computer science, UIC in 2009-2010, as a visiting scholar respectively. After her graduation from UIC, she joined in Beihang University (BUAA), Beijing, P. R. China. She is currently an associated professor in School of Mathematics and System Sciences, BUAA. She was elevated as IEEE senior member in 2015.

Dr Luo's research interests include nonlinear filtering theory, numerical analysis of spectral methods, analysis of partial differential equations, sparse grid algorithm and fluid mechanics.

S. S.-T. Yau (F'03) received the Ph.D. degree in mathematics from the State University of New York at Stony Brook, NY, US, in 1976. He was a member of Institute of Advanced Study at Princeton 1976-1977 and 1981-1982, and a Benjamin Pierce Assistant Professor at Harvard University during 1977-1980. After that, he joined the department of mathematics, statistics and computer science (MSCS), University of Illinois at Chicago (UIC), and served for over 30 years. He was awarded Sloan Fellowship in 1980, Guggenheim Fellowship in 2000, IEEE Fellow Award in 2003 and AMS Fellow Award in 2013. In 2005, he was entitled the UIC distinguished professor. During 2005-2011, he became a joint-professor of department of electrical and computer engineering and MSCS, UIC. After his retirement in 2012, he joined Tsinghua University, Beijing, P. R. China, where he is a full-time professor in department of mathematical science.

Dr Yau's research interests include nonlinear filtering, bioinformatics, complex algebraic geometry, CR geometry and singularities theory.

Dr Yau is the Managing Editor and founder of *Journal of Algebraic Geometry* from 1991, and the Editors-in-Chief and founder of *Communications in Information and Systems* from 2000 till now. He was the General Chairman of IEEE International Conference on Control and Information, which was held in the Chinese University of Hong Kong in 1995.

Z. Zhang received his B.S. degree and Ph.D. degree in mathematics from Tsinghua University, Beijing, P.R. China, in 2006 and 2011, respectively. As a Ph.D. candidate at Tsinghua University, he visited the University of Wisconsin-Madison as a visiting student in 2008-2009. After his graduation, he was a postdoctoral scholar at California Institute of Technology from 2011 to 2015. He joined the University of Hong Kong as an assistant professor since 2015.

Dr. Zhang's research interests are scientific computation. Research topics include uncertainty quantification (UQ), numerical methods for partial differential equations (PDEs) arising from quantum chemistry, wave propagation, multiscale porous media, nonlinear filtering, data assimilation, and stochastic fluid dynamics.