IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 17, 2020, accepted January 1, 2021, date of publication January 11, 2021, date of current version September 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3050572

A Novel Real-Time Filtering Method to General Nonlinear Filtering Problem Without Memory

JI SHI[1], XIUQIONG CHEN[2], (Member, IEEE), AND STEPHEN SHING-TOUNG YAU[3,4], (Fellow, IEEE)
[1]Beijing Advanced Innovation Center for Imaging Theory and Technology, Academy for Multidisciplinary Studies, Capital Normal University, Beijing 100048, China
[2]Yau Mathematical Sciences Center, Tsinghua University, Beijing 100084, China
[3]Department of Mathematical Sciences, Tsinghua University, Beijing 100084, China
[4]Yanqi Lake Beijing Institute of Mathematical Sciences and Applications, Beijing 101408, China

Corresponding author: Stephen Shing-Toung Yau (yau@uic.edu)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 11961141005, in part by the Tsinghua University start-up Fund, and in part by the Tsinghua University Education Foundation Fund under Grant 042202008.

ABSTRACT In this paper, the filtering problem for the general time-invariant nonlinear state-observation system is considered. Our work is based on the Yau-Yau filtering framework developed by S.-T. Yau and the third author in 2008. The key problem of Yau-Yau filtering framework is how to compute the solution to forward Kolmogorov equation (FKE) off-line effectively. Motivated by the supervised learning in machine learning, we develop an efficient method to numerically solve the FKE off-line from the point of view of optimization. Specifically, for the off-line computation part, the computation of the solution to a FKE is reduced to computing a linear system of equations by making the temporal inverse transformation and the loss function optimization, and we store the results for the preparation of on-line computation. For the on-line computation part, the unnormalized density function is approximated by a complete polynomial basis, and then the estimation of the state is computed using the stored off-line data. Our method has the merits of easily implementing, real-time and memoryless. More importantly, it can be applicable for moderate-high dimensional cases. Numerical experiments have been carried out to verify the feasibility of our method. Our algorithm outperforms extended Kalman filter, unscented Kalman filter and particle filter both in accuracy and costing time.

INDEX TERMS Duncan-Mortensen-Zakai equation, nonlinear filtering, forward Kolmogorov equation, estimation theory, numerical algorithms.

I. INTRODUCTION

Ever since Kalman and Bucy proposed the famous Kalman filter which has been widely used in various fields of industry in the 1960s [1], [2], numerous researchers have devoted many efforts to the study of nonlinear filtering (NLF) theory and practical NLF algorithms. The central problem in NLF is to seek the optimal estimate of the state given the observations corrupted by the noises. It is known that the minimum mean square error estimate of the state is its conditional expectation based on the observation history [3].

To obtain the optimal estimate, one direction is to approximate the conditional expectation directly, such as the widely used extended Kalman filter (EKF) [3], unscented Kalman filter (UKF) [4], ensemble Kalman filter [5], etc. EKF and UKF assume the posterior distribution of the states is Gaussian or nearly Gaussian in essence which restricts their applications.

The other direction of NLF problem is to compute the conditional density function of the state. For instance, particle filter (PF) [6] use the empirical distribution of the particles to approximate the posterior density. PF is well known for its ability to be used in general NLF problem while it cannot be implemented in real-time for high-dimensional systems. In the late 1960s, Duncan, Mortensen and Zakai independently derived the famous Duncan-Mortensen-Zakai (DMZ) equation for nonlinear filtering problem, which is satisfied by the unnormalized conditional density function of the

The associate editor coordinating the review of this manuscript and approving it for publication was Qichun Zhang.

states [7]–[9]. The DMZ equation is a stochastic differential equation, and usually it does not has a closed form solution. For the purpose of real applications, we need to solve the DMZ equation in a real-time and memoryless manner. For finite dimensional filtering systems, the solution to the corresponding DMZ equation can be explicitly constructed by Lie algebra approach. However, only a few classes of systems possess finite dimensional filters [10]–[13]. Since the DMZ equation usually does not have a closed form solution, many efforts have been devoted to find a good approximation solution to the DMZ equation in some sense by mathematicians. One of these methods is the splitting up method first described by Bensoussan *et al.* [14], [15], and later studied in [16]–[19]. In 1990s, Lototsky, Mikulevicius and Rozovskii gave a recursive in time Wiener chaos representation of the optimal nonlinear filter [20]. However, all these methods require that the drift term and observation term, i.e. $f(x)$ and $h(x)$ in system (1) are bounded. Another method to solve the DMZ equation is the so-called direct method. Typically, direct method can only be applied to Yau filtering system, i.e. the drift term is a linear function plus a gradient function. This method was introduced in [21] and later generalized in [22]–[26]. In 2008, Yau and Yau developed a new algorithm called Yau-Yau algorithm to solve the "pathwise-robust" DMZ equation for time-invariant system [27], and it has been proved theoretically that the Yau-Yau algorithm can converge to the true solution, as long as the growth rate of the observation $|h|$ is greater than that of the drift $|f|$. Later, Luo and Yau generalized the Yau-Yau algorithm to time varying case [28].

Our work is based on the Yau-Yau algorithm for general time-invariant system. The key part of the Yau-Yau framework is that the computation of the forward Kolmogorov equation (FKE) can be moved off-line, which much accelerates the on-line running. Now the question is how to solve FKE efficiently. Motived by the supervised learning in machine learning, we transform this problem into an optimization problem, and finally into solving a linear system of equations through temporal inverse transformation, spectral approximation and numerical approximation of integral.

The proposed filtering method is easy-to-implement, fast, efficient and can be easily applied for moderate-high dimensional case. Besides, it also inherits the merits of real-time and memoryless. By "memoryless" we mean a filtering method uses each new observation to update a probability distribution for the state of the system without referring back to any earlier observations. The numerical simulations verify the effectiveness of the proposed method. Our method can track the notorious cubic sensor problem [29] while EKF totally fails and surpass PF in costing time.

The paper is organized as follows. In section 2, we recall the basic filtering problem and some preliminary results. In section 3, we derive the approximation solution of the FKE and give the off-line-on-line real-time filtering algorithm. In section 4, several numerical experiments were carried out to verify the feasibility and effectiveness of our method.

II. PRELIMINARY

In this paper we consider the following continuous filtering problem with implicit time-dependence in the drift term $f(x_t)$, observation term $h(x_t)$, diffusion term $g(x_t)$ and the variances of the noises:

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

where $x_t \in \mathbb{R}^n$ is the state of the system at time t, $y_t \in \mathbb{R}^m$ is the observation with $y_0 = 0$, $f(x_t) \in \mathbb{R}^n$, $h(x_t) \in \mathbb{R}^m$ are vector value drift term and observation term, respectively, $g(x_t) \in \mathbb{R}^{n\times p}$ is a matrix value diffusion term, and w_t, v_t are independent p-vector and m-vector Brownian motion processes with variance Q and S, respectively. We assume that $\{w_t\}_{t\geq 0}$, $\{v_t\}_{t\geq 0}$ and x_0 are independent.

The goal of filtering is to estimate the current state x_t given all observations till instant t under some criteria. According to [3], we know that the minimum variance estimate of x_t is given by $E[x_t|\mathcal{F}_t]$, where $\mathcal{F}_t := \sigma\{y_s : 0 \leq s \leq t\}$. Therefore, the filtering problem can be completely solved once we know the conditional density function $p(x, t)$ of the state x_t based on the observation history $\mathcal{F}_t$. It is well known that the unnormalized conditional density function $\sigma(x, t)$ of x_t satisfies the following DMZ equation [7]–[9]:

$$\begin{cases} d\sigma(x, t) = \mathcal{L}_0\sigma(x, t)dt + h^T(x)S^{-1}\sigma(x, t)dy_t \\ \sigma(x, 0) = \sigma_0(x), \end{cases} \tag{2}$$

where the operator $\mathcal{L}_0$ is

$$\mathcal{L}_0(*) := \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j}(G_{ij}(x)*) - \sum_{i=1}^{n} \frac{\partial (f_i *)}{\partial x_i}, \tag{3}$$

$G(x) := g(x)^T Q g(x)$, and $\sigma_0(x)$ is the density of the initial states x_0.

In real applications, we are interested in considering robust state estimator from observed sample paths with some properties of robustness. Davis considered this problem and proposed some robust algorithms [30]. In our case, by making the following gauge transformation

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

the DMZ equation can be transformed into the following deterministic partial differential equation (PDE) with stochastic coefficients

$$\begin{cases} \dfrac{\partial u}{\partial t}(x, t) = \dfrac{1}{2}\displaystyle\sum_{i,j=1}^{n} G_{ij}(x)\frac{\partial^2 u}{\partial x_i \partial x_j}(x, t) + F(x)\nabla u(x, t) \\ \qquad\qquad + J(x)u(x, t) \\ u(x, 0) = \sigma_0(x), \end{cases} \tag{5}$$

where

$$F(x) = \left[\sum_{j=1}^{n} \frac{\partial G_{ij}}{\partial x_j} + \sum_{j=1}^{n} G_{ij} \frac{\partial K}{\partial x_j} - f_i\right]_{i=1}^{n},$$

$$\begin{aligned} J(x) = &\frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial^2 G_{ij}}{\partial x_i \partial x_j} + \sum_{i,j=1}^{n} \frac{\partial G_{ij}}{\partial x_i}\frac{\partial K}{\partial x_j} \\ &+ \frac{1}{2}\sum_{i,j=1}^{n} G_{ij} \times \left[\frac{\partial^2 K}{\partial x_i \partial x_j} + \frac{\partial K}{\partial x_i}\frac{\partial K}{\partial x_j}\right] \\ &- \sum_{i=1}^{n} \frac{\partial f_i}{\partial x_i} - \sum_{i=1}^{n} f_i \frac{\partial K}{\partial x_i} - \frac{1}{2} h^T S^{-1} h \\ &- \frac{\partial}{\partial t}\left(h^T S^{-1}\right)^T y_t, \end{aligned} \tag{6}$$

in which

$$K(x,t) = h^T(x) S^{-1} y_t. \tag{7}$$

And (5) is equivalent to the following equation

$$\begin{cases} \frac{\partial u}{\partial t}(x,t) = \exp(-h^T S^{-1} y_t)\left(\mathcal{L}_0 - \frac{1}{2} h^T S^{-1} h\right) \\ \qquad \cdot \left[\exp(h^T S^{-1} y_t) u(x,t)\right] \\ u(x,0) = \sigma_0(x). \end{cases} \tag{8}$$

The equation (5) or (8) is called "pathwise-robust" DMZ equation. The existence and uniqueness of the "pathwise-robust" DMZ equation have been investigated by many mathematicians. When $f(x)$ and $h(x)$ satisfy some mild growth conditions, Yau and Yau obtained the well-posedness of (5) in [27]. Generally, the equation (5) does not have closed form solution. So many mathematicians devote to seeking the efficient algorithm to construct a good approximation solution. Yau and Yau first gave an off-line-on-line filtering framework [27] and later Luo and Yau extended the result to general time-varying case [31]. We will follow the Yau-Yau filtering framework in this paper.

Let us assume the observation time sequence is $P_N = \{0 = \tau_0 < \tau_1 < \cdots < \tau_N = T\}$. Let u_i be the solution of the robust DMZ eqnarray (5) with $y_t = y_{\tau_{i-1}}$ on the interval $\tau_{i-1} \le t < \tau_i$, $i = 1, 2, \cdots, N$, i.e.,

$$\begin{cases} \frac{\partial u_i}{\partial t}(x,t) = \exp(-h^T S^{-1} y_{\tau_{i-1}})\left(\mathcal{L}_0 - \frac{1}{2} h^T S^{-1} h\right) \\ \qquad \cdot \left[\exp(h^T S^{-1} y_{\tau_{i-1}}) u_i(x,t)\right] \\ u_1(x,0) = \sigma_0(x), \\ u_i(x,\tau_{i-1}) = u_{i-1}(x,\tau_{i-1}), \text{ for } i = 2, 3, \cdots, N. \end{cases} \tag{9}$$

Define the norm of P_N by $|P_N| := \sup_{1 \le i \le N}(\tau_i - \tau_{i-1})$. It has been proved that in both point-wise sense and L^2-sense,

$$u(x,t) = \lim_{|P_N| \to 0} u_i(x,\tau). \tag{10}$$

Without loss of generality, we assume that we have equal interval observations, i.e., $\tau_i - \tau_{i-1} = T_0$, $\forall 1 \le i \le N$, where $T_0 > 0$ is very small. Since the observations $y_{\tau_i}, i = 1, \cdots, N$ are contained in the coefficients of (9), it is impractical to solve the PDE (9) in real-time. Fortunately, the following proposition helps to move the heavy computations of solving PDE off-line, which is the key part of Yau-Yau filtering framework.

Proposition 1 (Yau-Yau [27]): For each $\tau_{i-1} \le t < \tau_i$, $i = 1, 2, \cdots, N$, $u_i(x,t)$ satisfies (9) if and only if

$$\rho_i(x,t) = \exp\left(h^T S^{-1} y_{\tau_{i-1}}\right) u_i(x,t), \tag{11}$$

satisfies the FKE

$$\frac{\partial \rho_i}{\partial t}(x,t) = \left(\mathcal{L}_0 - \frac{1}{2} h^T S^{-1} h\right) \rho_i(x,t), \tag{12}$$

where $\mathcal{L}_0$ is defined in (3).

It is obvious that there is a one-to-one correspondence between $\rho_i(x,t)$ and $u_i(x,t)$, and the FKE (12) does not depend on the information of the observations $\{y_{\tau_i}\}$. Hence we can solve (12) in advance.

Considering that the states are always bounded, we can denote the bounded domain as $U = [-1,1]^n$ after appropriate scaling. We use $L^2(U)$ to denote the square integrable functions on U, equipped with the inner product $\langle u, v\rangle := \int_U uvdx$ and the induced norm $\|u\|^2 := \langle u, u\rangle$, for $\forall\, u, v \in L^2(U)$.

The Yau-Yau filtering framework contains on-line and off-line computation parts.

- Off-line computation: On a very small time interval $[0, T_0]$, we compute the solution to (12) with initial condition $\rho(x,0) = \phi_l(x)$, $l = 0, 1, \cdots$, where $\{\phi_l(x)\}_{l=0}^{\infty}$ is a complete basis in $L^2(U)$. Assume that the solution of FKE (12) with initial value $\phi_l(x)$ at time $t = T_0$ is $\Phi_l(x)$, $l = 0, 1, \cdots$. We need to store data $\{\Phi_l(x)\}_{l=0}^{\infty}$ in preparation of the on-line computation.
- On-line computation: In this part, for each time interval $\tau_{i-1} \le t < \tau_i$, $i = 1, \cdots, N$, we first project the initial condition $\rho_i(x,\tau_{i-1}) \in L^2(U)$ of (12) onto the complete basis $\{\phi_l(x)\}_{l=0}^{\infty}$ of $L^2(U)$, i.e.

$$\rho_i(x,\tau_{i-1}) = \sum_{l=0}^{\infty} c_{i,l}\phi_l(x). \tag{13}$$

Second, by the superposition principle of linear differential equation, the solution of (12) at τ_i is given by

$$\rho_i(x,\tau_i) = \sum_{l=0}^{\infty} c_{i,l}\Phi_l(x). \tag{14}$$

Last, update the initial condition $\rho_{i+1}(x,\tau_i)$ for next time interval $\tau_i \le t < \tau_{i+1}$:

$$\rho_{i+1}(x,\tau_i) = \exp(h^T S^{-1}(y_{\tau_i} - y_{\tau_{i-1}})) \cdot \rho_i(x,\tau_i). \tag{15}$$

We can obtain $u_i(x,t)$, $i = 1, \cdots, N$, from $\rho_i(x,t)$ and (11). Let us denote the approximation of $u(x,t)$ as $\hat{u}(x,t)$ which is

$$\hat{u}(x,t) = \sum_{i=1}^{N} 1_{[\tau_{i-1},\tau_i]}(t) u_i(x,t), \tag{16}$$

where $1_{[\tau_{i-1},\tau_i]}(\cdot)$ is the indicator function. It has been proved that $u(x,t)$ is well approximated by $\hat{u}(x,t)$ [31].

III. NUMERICAL APPROXIMATION OF THE SOLUTION TO FKE

In this section, we shall develop a new efficient algorithm to compute the FKE (12) by reducing the problem to solving a linear system of equations, since the key part of the Yau-Yau filtering framework is how to solve the (12) efficiently. We omit the subscript in (12) for convenience and consider how to solve the following FKE on a very small time interval $[0, T_0]$:

$$\begin{cases} \frac{\partial \rho}{\partial t}(x,t) = (\mathcal{L}_0 - \frac{1}{2}h^T S^{-1} h)\rho(x,t) \\ \rho(x,0) = \sigma_0(x), \end{cases} \tag{17}$$

where $\mathcal{L}_0$ is defined in (3) and $\sigma_0(x) \in C^\infty(U)$. Here $C^\infty(U)$ is the set of infinitely differentiable functions on U.

A. METHODOLOGY

In this part, we shall give a numerical approximation of $\rho(x, T_0)$, which is the solution of (17) at time T_0. We first do a temporal inverse transformation

$$v(x,t) := \rho(x, T_0 - t), \tag{18}$$

then the equation (17) can be transformed into the following eqnarray (19) with terminal condition:

$$\begin{cases} \frac{\partial v}{\partial t}(x,t) = \mathcal{L} v(x,t) \\ v(x,T_0) = \sigma_0(x), \end{cases} \tag{19}$$

where the operator $\mathcal{L}$ is defined as:

$$\mathcal{L} := -(\mathcal{L}_0 - \frac{1}{2}h^T S^{-1} h). \tag{20}$$

Observing that the operator $\mathcal{L}$ is time-invariant, we have

$$v(x,T_0) = \mathcal{A} v(x,0), \tag{21}$$

with a certain operator $\mathcal{A} : L^2(U) \to L^2(U)$, which is determined by T_0 and the operator $\mathcal{L}$ [31].

Now the problem is how to compute $v(x,0)$ since $\rho(x,T_0) = v(x,0)$.

Let $\mathcal{K}$ be the subspace generated by finite basis functions $\{\phi_l(x)\}_{l=0}^{N_1}$ of $L^2(U)$ with coefficients w_l, $l = 0, 1, 2, \cdots, N_1$. We project $v(x,0)$ onto $\mathcal{K}$ and obtain $\tilde{v}(x,0)$, i.e.,

$$\tilde{v}(x,0) = \sum_{l=0}^{N_1} w_l \phi_l(x), \tag{22}$$

and we want to obtain a good approximation of $v(x,0)$ on the subspace $\mathcal{K}$. Hence, the problem is transferred into how to determine w_l, $l = 0, \cdots, N_1$.

Let $P_{N'} := \{0 = s_0 < s_1 < \cdots < s_{N'} = T_0\}$ be an equal spacing partition of $[0, T_0]$, where $\Delta s = s_i - s_{i-1}$, $i = 1, \cdots, N'$. Since T_0 is very small, we can use Euler's method to discrete the equation (19) in temporal dimension, i.e., we compute the numerical approximation $\tilde{v}(x,t)$ of $v(x,t)$ as follows:

$$\begin{cases} \tilde{v}(x,s_1) - \tilde{v}(x,s_0) = \Delta s \mathcal{L} \tilde{v}(x,s_0), \\ \tilde{v}(x,s_2) - \tilde{v}(x,s_1) = \Delta s \mathcal{L} \tilde{v}(x,s_1), \\ \vdots \\ \tilde{v}(x,s_{N'}) - \tilde{v}(x,s_{N'-1}) = \Delta s \mathcal{L} \tilde{v}(x,s_{N'-1}). \end{cases} \tag{23}$$

Adding up all the terms on the left hand side and the right hand side of (23) respectively, we have

$$\begin{aligned} \tilde{v}(x,T_0) &= \tilde{v}(x,s_{N'}) \\ &= \Delta s \mathcal{L}(\tilde{v}(x,s_{N'-1}) + \cdots + \tilde{v}(x,s_1) + \tilde{v}(x,s_0)) \\ &\quad + \tilde{v}(x,s_0) \\ &= ((\Delta s \mathcal{L})^{N'} + N'(\Delta s \mathcal{L})^{N'-1} \\ &\quad + \cdots + N' \Delta s \mathcal{L} + 1)\tilde{v}(x,s_0) \\ &= (\Delta s \mathcal{L} + 1)^{N'} \tilde{v}(x,s_0). \end{aligned} \tag{24}$$

Let

$$\mathcal{A}_{N'} := (\Delta s \mathcal{L} + 1)^{N'}, \tag{25}$$

then $\mathcal{A}_{N'}$ can be regarded as the numerical approximation of $\mathcal{A}$, and

$$\tilde{v}(x,T_0) = \mathcal{A}_{N'} \tilde{v}(x,0) \tag{26}$$

is the numerical approximation of $v(x,T_0)$ by Euler's method with initial value $\tilde{v}(x,0)$. Substituting $\tilde{v}(x,0)$ in (26) with (22), we have

$$\tilde{v}(x,T_0) = \mathcal{A}_{N'} \tilde{v}(x,0) = \sum_{l=0}^{N_1} w_l Z_l(x), \tag{27}$$

where

$$Z_l := \mathcal{A}_{N'} \phi_l(x). \tag{28}$$

It is important to observe that for some complete bases $\{\phi_l(x)\}_{l=0}^{\infty}$, (28) can be calculated explicitly since $\mathcal{L}$ only involves derivations with respect to x, which much simplifies the off-line computations. For example we can choose generalized Legendre polynomials introduced in [32]. More details about this basis can be found in appendix V and it is obvious that the corresponding $\{Z_l(x)\}_{l=0}^{N_1}$ can be explicitly calculated.

Now we consider how to compute coefficients w_l, $l = 0, 1, \cdots, N_1$ in (22). Firstly we define the approximation loss function as follows:

$$\begin{aligned} J(W) :&= \|\tilde{v}(x,T_0) - v(x,T_0)\|^2 \\ &= \left\| \sum_{l=0}^{N_1} w_l Z_l(x) - \sigma_0(x) \right\|^2 \\ &= \left\| \sum_{l=0}^{N_1} w_l Z_l(x) \right\|^2 - 2\sum_{l=0}^{N_1} w_l \langle \sigma_0(x), Z_l(x) \rangle \\ &\quad + \|\sigma_0(x)\|^2, \end{aligned} \tag{29}$$

where $W = [w_0, w_1, \cdots, w_{N_1}]^T$.

The optimal coefficients W^* are determined by minimizing the loss function $J(W)$, i.e.

$$W^* = \underset{w_0, w_1, \cdots, w_{N_1}}{arg\min} \ J(W). \tag{30}$$

It is easy to see that $J(W)$ is a convex function with respect to W. Therefore, the critical point of $J(W)$ is its minimum point. The derivative of $J(W)$ with respect to $w_j, j = 0, 1, 2, \cdots, N_1$ is

$$\frac{\partial J(W)}{\partial w_j} = 2\|Z_j(x)\|^2 w_j + 2\sum_{i\neq j}^{N_1} w_i \langle Z_i(x), Z_j(x)\rangle - 2\langle \sigma_0(x), Z_j(x)\rangle. \tag{31}$$

The critical point of $J(W)$ is determined by the following equation:

$$\frac{\partial J(W^*)}{\partial w_0} = \frac{\partial J(W^*)}{\partial w_1} = \cdots = \frac{\partial J(W^*)}{\partial w_{N_1}} = 0. \tag{32}$$

From (31) and (32), we have

$$\begin{pmatrix} \langle Z_0, Z_0\rangle & \langle Z_0, Z_1\rangle & \cdots & \langle Z_0, Z_{N_1}\rangle \\ \langle Z_1, Z_0\rangle & \langle Z_1, Z_1\rangle & \cdots & \langle Z_1, Z_{N_1}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle Z_{N_1}, Z_0\rangle & \langle Z_{N_1}, Z_1\rangle & \cdots & \langle Z_{N_1}, Z_{N_1}\rangle \end{pmatrix} W^* = \begin{pmatrix} \langle \sigma_0(x), Z_0(x)\rangle \\ \langle \sigma_0(x), Z_1(x)\rangle \\ \vdots \\ \langle \sigma_0(x), Z_{N_1}(x)\rangle \end{pmatrix}. \tag{33}$$

Therefore, the coefficients of the approximation to the solution $\tilde{v}(0, x)$ on the subspace $\mathcal{K}$ are determined by the above linear system of equations (33), i.e.,

$$\tilde{v}^*(x, 0) = \sum_{l=0}^{N_1} w_l^* \phi_l(x) \tag{34}$$

can be regarded as a good numerical approximation of $v(x, 0)$.

According to (18), we know that

$$\tilde{\rho}^*(x, T_0) = \tilde{v}^*(x, 0) = \sum_{l=0}^{N_1} w_l^* \phi_l(x) \tag{35}$$

is a good approximation of $\rho(x, T_0)$. Now we summarize the procedures of computing $\tilde{\rho}^*(x, T_0)$ in the following framework:

Framework: Suppose $\{\phi_l(x)\}_{l=0}^{\infty}$ is a complete basis of $L^2(U)$. Under the assumption that T_0 is very small, a good approximation of the solution to the FKE (17) at time T_0 on the subspace $\mathcal{K} = span\{\phi_0(x), \cdots, \phi_{N_1}(x)\}$ is given by

$$\tilde{\rho}^*(x, T_0) = \sum_{l=0}^{N_1} w_l^* \phi_l(x), \tag{36}$$

where the coefficients $W^* := [w_0^*, w_1^*, \cdots, w_{N_1}^*]^T$ is determined by the following linear system of equations

$$AW^* = D, \tag{37}$$

in which the matrix A and the vector D are defined as follows:

$$A = \begin{pmatrix} \langle Z_0, Z_0\rangle & \langle Z_0, Z_1\rangle & \cdots & \langle Z_0, Z_{N_1}\rangle \\ \langle Z_1, Z_0\rangle & \langle Z_1, Z_1\rangle & \cdots & \langle Z_1, Z_{N_1}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle Z_{N_1}, Z_0\rangle & \langle Z_{N_1}, Z_1\rangle & \cdots & \langle Z_{N_1}, Z_{N_1}\rangle \end{pmatrix},$$

$$D = \begin{pmatrix} \langle \sigma_0(x), Z_0(x)\rangle \\ \langle \sigma_0(x), Z_1(x)\rangle \\ \vdots \\ \langle \sigma_0(x), Z_{N_1}(x)\rangle \end{pmatrix}, \tag{38}$$

and $\{Z_l\}_{l=0}^{N_1}$ are defined in (28). With specific chosen complete basis such as generalized Legendre polynomials, $\{Z_l(x)\}_{l=0}^{N_1}$ in equation (38) can be explicitly calculated, and then the matrix A and vector D can be explicitly calculated.

B. ALGORITHM

In this part, we shall give a detailed description of the Yau-Yau filtering method based on the proposed numerical approximation of $\rho(x, T_0)$. We firstly introduce the off-line part, which is listed in Algorithm 1, and we store up the data $\{\tilde{\rho}_k^*(x, T_0)\}_{k=0}^{N_2}$.

Algorithm 1 Off-Line Computation

1: **Initialization:** Given the complete basis $\{\phi_l(x)\}$, the number of the off-line approximation basis N_1, the number of the on-line approximation basis N_2, the time T_0 and the number of the partition N'.
2: Calculate $Z_l, l = 0, 1, 2, \cdots, N_1$ and the matrix A in (38).
3: **for** $k = 0 : N_2$ **do**
4: Calculate the vector D with $\sigma_0(x) = \phi_k(x)$.
5: Solve equation (37) with the matrix A and vector D, and the solution is $W_k^* = [w_{k,0}, w_{k,1}, \cdots, w_{k,N_1}]^T$.
6: Calculate the approximation solution $\tilde{\rho}_k^*(x, T_0) = \sum_{l=0}^{N_1} w_{k,l}^* \phi_l(x)$ and store up $\tilde{\rho}_k^*(x, T_0)$ for the preparation of the on-line computation.
7: **end for**

With the ready off-line data $\{\tilde{\rho}_k^*(x, T_0)\}_{k=0}^{N_2}$, now we are ready to describe the on-line computation algorithm, which is listed in Algorithm 2.

Combining Algorithm 1 and Algorithm 2, we obtain the complete procedure of our DMZ method based on Yau-Yau filtering framework.

IV. NUMERICAL SIMULATIONS

In this section, we investigate the approximation property and costing time of our DMZ method by two NLF examples. The first example is the cubic sensor problem which has been discussed in detail in [29]. It has been proved that for nonconstant ψ, there cannot exist a recursive finite-dimensional filter for $\hat{\psi}$ driven by the observations, where $\hat{\psi} = E[\psi(x_t)|\mathcal{F}_t]$ is the optimal estimate of $\psi(x_t)$. And it is a classic example in testing the efficiency of the filtering algorithm [28], [33].

Algorithm 2 On-Line Computation

Initialization: Given the number of the partition N, and the off-line data $\{\tilde{\rho}_k^*(x, T_0)\}_{k=0}^{N_2}$.

for $i = 1, \cdots, N$ **do**

Project $\rho_i(x, \tau_{i-1})$ into $span\{\phi_0(x), \cdots, \phi_{N_2}(x)\}$, calculate the coefficients of projection $\{c_{i,k}\}_{k=0}^{N_2}$, i.e.,

$$\rho_i(x, \tau_{i-1}) \approx \sum_{l=0}^{N_2} c_{i,l}\phi_l(x). \tag{39}$$

Calculate $\rho_i(x, \tau_i)$ by $\{c_{i,k}\}_{k=0}^{N_2}$ and the off-line data $\{\tilde{\rho}_k^*(x, T_0)\}_{k=0}^{N_2}$, i.e.,

$$\rho_i(x, \tau_i) \approx \sum_{k=1}^{N_2} c_{i,k}\tilde{\rho}_k(x, T_0). \tag{40}$$

Calculate $u_i(x, \tau_i)$ by $\rho_i(x, \tau_i)$ and (11).

Calculate the estimate of the state at time τ_i by $u_i(x, \tau_i)$, i.e.

$$\hat{x}_i = \frac{\int_{\mathbb{R}^n} x \cdot u_i(x, \tau_i)dx}{\int_{\mathbb{R}^n} u_i(x, \tau_i)dx}. \tag{41}$$

Update the initial value of $\rho_{i+1}(t, x)$ in the next time interval $[\tau_i, \tau_{i+1})$ by (15), i.e.

$$\rho_{i+1}(x, \tau_i) = \exp(h^T S^{-1}(y_{\tau_i} - y_{\tau_{i-1}})) \cdot \rho_i(x, \tau_i). \tag{42}$$

end for

The second one is used to show that our DMZ method can also work well for two dimensional systems.

In the simulations, our DMZ method is compared with EKF, UKF and PF. As we will see, our algorithm can track the state well while EKF and UKF totally fail and surpass PF both in accuracy and costing time. For the purpose of comparing the performance of different methods, we introduce the mean of the squared estimation error (MSE) and average absolute error $\mathrm{e_{mean}}(\tau_k)$ at instant τ_k based on 100 realizations, and they are defined as follows:

$$\begin{aligned} \mathrm{MSE} &:= \frac{1}{100}\sum_{i=1}^{100}\frac{1}{N+1}\sum_{k=0}^{N}(x_{\tau_k}^{(i)} - \hat{x}_{\tau_k}^{(i)})^2, \\ \mathrm{e_{mean}}(\tau_k) &:= \frac{1}{100}\sum_{i=1}^{100}\left|x_{\tau_k}^{(i)} - \hat{x}_{\tau_k}^{(i)}\right|, \end{aligned} \tag{43}$$

where $x_{\tau_k}^{(i)}$ is the real state at instant τ_k in the i-th experiment and $\hat{x}_{\tau_k}^{(i)}$ is the estimation of $x_{\tau_k}^{(i)}$, with $0 \le \tau_k \le T$, $0 \le k \le N$. The total simulation time is $T = 5s$, and we use Euler's method in time discretization with the time step $\Delta t = 0.01s$.

Example 1: The cubic sensor problem is as follows:

$$\begin{cases} dx_t = dv_t \\ dy_t = x_t^3 dt + dw_t, \end{cases} \tag{44}$$

where $x_t, y_t \in \mathbb{R}$, v_t, w_t are independent standard scalar Brownian motion processes. Assume that the initial distribution of the state is $p_0(x) = \exp(-\frac{(x-1)^2}{2v_0})$ with $v_0 = 0.1$.

In this case the corresponding 1-d FKE for (44) is

$$\frac{\partial \rho}{\partial t}(x, t) = \frac{1}{2}\frac{\partial^2 \rho}{\partial x^2}(x, t) - \frac{1}{2}x^6\rho(x, t). \tag{45}$$

The initial value of real state is $x_0 = 1$. The initial mean and covariance for EKF, UKF and PF are $\hat{x}_0 = 1$ and $\hat{P}_0 = 0.1$, respectively. And the parameters used in UKF are chosen as $\alpha = 0.05$, $\beta = 2$ and $\kappa = 0$ [34]. As for our method, $N' = 1$ and we use $N_1 = 10, N_2 = 15$ generalized Legendre basis functions [32]. The performances of different methods in one experiment are shown in FIGURE 1 and the average absolute errors of our DMZ method and PF with four different number of particles based on 100 experiments are shown in FIGURE 2. Here, N_{PF} denotes the number of PF. It can be clearly seen that our method can track the real state better than the EKF, UKF and PF in one experiment. Besides, in 100 experiments, on average, our method performs better than the PF with four different numbers of particles. To obtain the average performance of different algorithms, we repeat the experiment for 100 times and the MSE, $\mathrm{e_{mean}}$ and the costing times are shown in TABLE 1. EKF and UKF will blow up in some experiments and their MSEs are meaningless. It can be concluded that our method outperforms the EKF, UKF and PF both in MSE and costing time.

TABLE 1. **The average performance of different methods based on 100 simulations in Example 1.**

Method	MSE	Costing time (s)
DMZ method	0.4317	0.0036
EKF	—-[1]	0.0001
UKF	—-[1]	0.0093
PF ($N_{\mathrm{PF}} = 50$)	1.3067	0.0710
PF ($N_{\mathrm{PF}} = 100$)	1.1342	0.1392
PF ($N_{\mathrm{PF}} = 150$)	0.9657	0.2104
PF ($N_{\mathrm{PF}} = 200$)	0.9115	0.2828

[1] The MSE is meaningless since it fails tracking the real state.

Example 2: The 2-d numerical example considered here is as follows:

$$\begin{cases} d\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} -0.4 & 0.1 \\ 0 & -0.6 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} dt + \begin{bmatrix} dv_1(t) \\ dv_2(t) \end{bmatrix}, \\ d\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} x_1^3(t) \\ x_2^3(t) \end{bmatrix} dt + \begin{bmatrix} dw_1(t) \\ dw_2(t) \end{bmatrix}, \end{cases} \tag{46}$$

where $x = [x_1, x_2]^T$ is the two dimensional state, $y = [y_1, y_2]^T$ is the two dimensional observation, and $v = [v_1, v_2]^T$, $w = [w_1, w_2]^T$ are two dimensional independent standard Brownian motions. Assume the initial distribution is $p_0(x) = \exp(-\frac{(x_1-1)^2+(x_2-1)^2}{2v_0})$ with $v_0 = 0.1$.

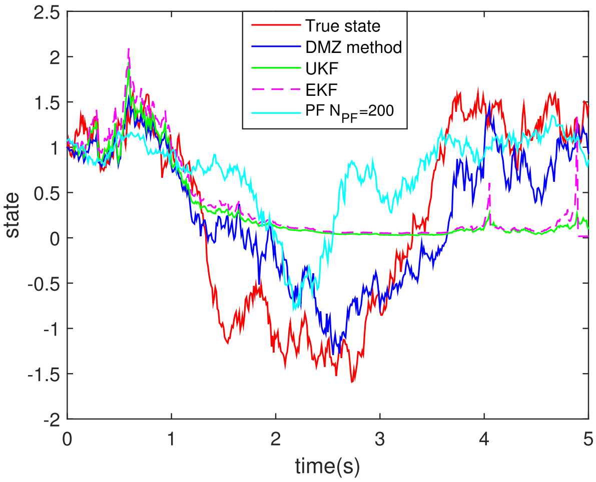

FIGURE 1. **One estimation results of different methods in Example 1.**

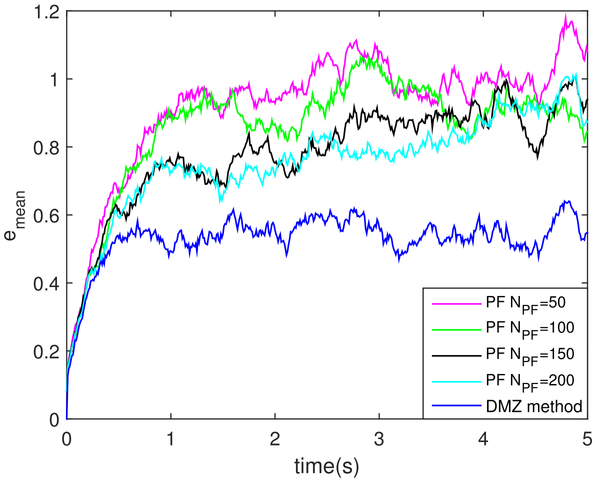

FIGURE 2. **Average estimation error of x by DMZ method and PF with different number of particles in Example 1.**

TABLE 2. **The average performance of different methods based on 20 simulations in Example 2.**

Method	MSE of x_1	MSE of x_2	Costing time (s)
DMZ method	0.4081	0.3646	0.0601
EKF	—-[1]	—-[1]	0.0151
UKF	—-[1]	—-[1]	0.0304
PF ($N_{\mathrm{PF}} = 50$)	1.5826	1.3508	0.6696
PF ($N_{\mathrm{PF}} = 100$)	1.6919	1.0715	1.3333
PF ($N_{\mathrm{PF}} = 150$)	1.3538	1.0411	1.9991
PF ($N_{\mathrm{PF}} = 200$)	1.1881	1.0133	2.6697

[1] The MSE is meaningless since it fails tracking the real state.

The corresponding 2-d FKE for (46) is

$$\begin{aligned}\frac{\partial \rho}{\partial t}(x,t) &= \frac{1}{2}\frac{\partial^2 \rho}{\partial x_1^2}(x,t) + \frac{1}{2}\frac{\partial^2 \rho}{\partial x_2^2}(x,t) \\ &\quad - (-0.4x_1 + 0.1x_2)\frac{\partial \rho}{\partial x_1}(x,t) + (0.6x_2)\frac{\partial \rho}{\partial x_2}(x,t) \\ &\quad + \left(1 - \frac{1}{2}(x_1^6 + x_2^6)\right)\rho(x,t). \qquad (47)\end{aligned}$$

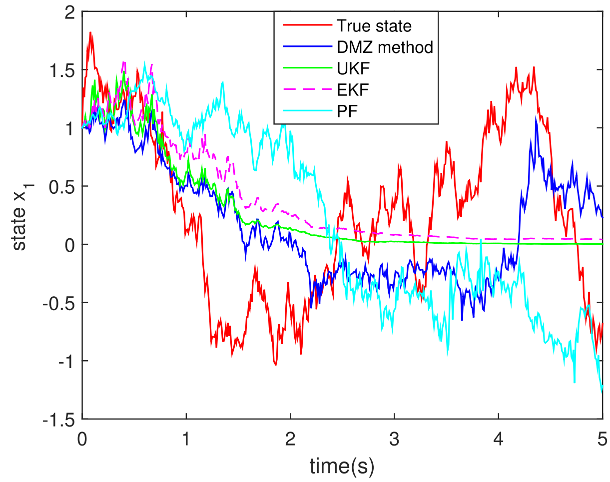

FIGURE 3. **One estimation results of state x_1 by different methods in Example 2.**

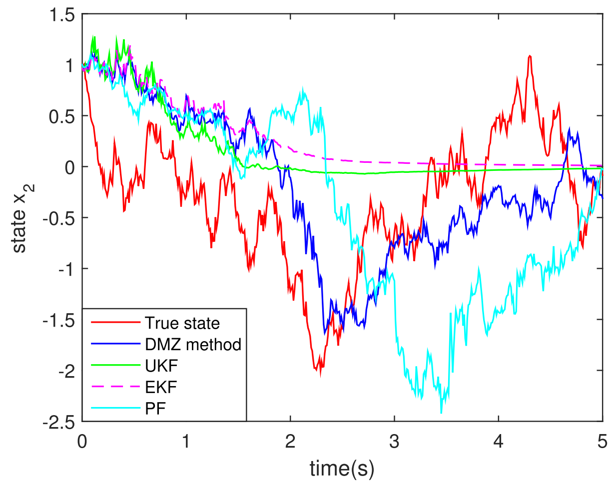

FIGURE 4. **One estimation results of state x_2 by different methods in Example 2.**

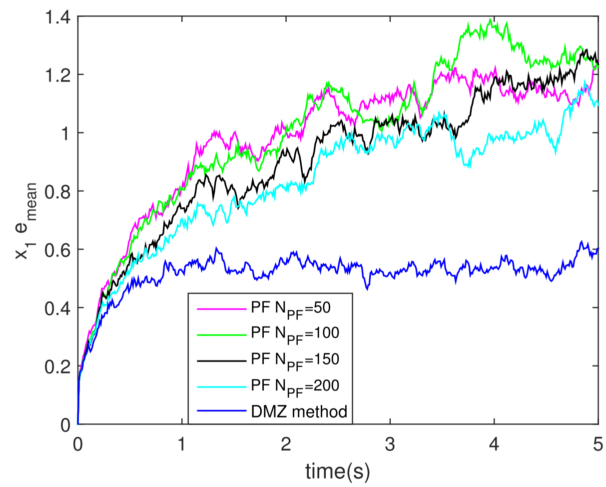

FIGURE 5. **Average estimation error of x_1 by DMZ method and PF with different number of particles in Example 2.**

The initial value of real state is $x_0 = [1, 1]^T$, and $N' = 1$. For our method, we use $N_1 = N_2 = 225$ generalized Legendre

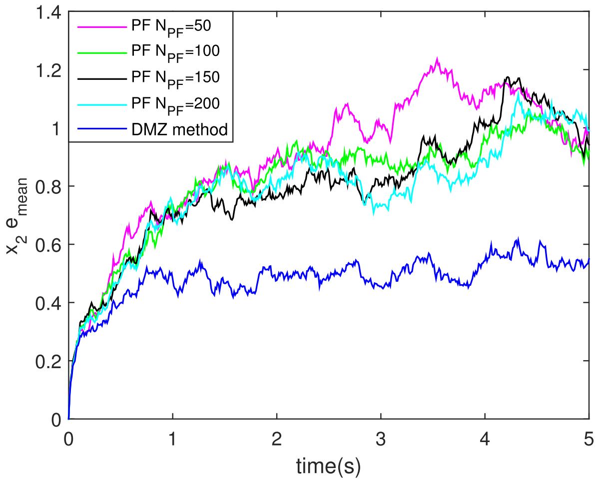

FIGURE 6. Average estimation error of x_2 by DMZ method and PF with different number of particles in Example 2.

basis functions [32] to solve the filtering problem on-line and off-line, respectively. The parameters used in UKF are chosen as $\alpha = 0.01$, $\beta = 2$ and $\kappa = 0$ [34]. The initial mean and covariance for EKF, UKF and PF are $\hat{x}_0 = [1, 1]^T$ and $\hat{P}_0 = 0.1 I_2$ respectively, where I_2 is the 2×2 identity matrix.

The estimation results in one experiment can be seen from Fig. 3 - Fig. 4. It is obvious that EKF and UKF fail tracking the real state and our method can track the real state better than the PF. Similarly, we also plot the estimation error of our DMZ method and PF with four different number of particles in FIGURE 5- FIGURE 6. The average performances including MSE and the costing times in 100 experiments are shown in the Table 2 and from which we can arrive at the conclusion that our method has the best performance.

V. CONCLUSION

In this paper we consider the filtering problem for the general time-invariant NLF systems. In the Yau-Yau algorithm framework, the key problem is to solve the FKE off-line and efficiently. By temporal inverse transformation and loss function optimization, the computation of solving the FKE on a small interval is now reduced to solving a linear system of equations. The proposed method enjoys the merits of easy-to-implement, fast, efficient and can be used in moderate-high dimensional problem. The method also inherits the advantages of Yau-Yau filtering algorithm, i.e., our algorithm can be implemented in a real-time and memoryless manner. Two numerical experiments have been done to verify the effectiveness of our algorithm. Compared with EKF and UKF, our method is very stable and can track the states of the classic cubic sensor problem very well. Compared with PF, our method has higher accuracy and needs less costing time. However, in our method, we need to know the system model and the statistics of the noises. In this respect, H_∞ filtering has the advantage of no statistical assumptions required on the noises [35], [36], which can be an interesting direction in our future work.

APPENDIX
GENERALIZED LEGENDRE POLYNOMIALS

In this part, we shall introduce the generalized Legendre polynomials $\{\phi_k(x)\}$ constructed in [32].

Firstly we introduce the Legendre polynomials $\{\varphi_k(x),\ k = 0, 1, \cdots, \}$, and for any k:

$$\varphi_k(x) = \frac{1}{2^k} \sum_{l=0}^{[k/2]} (-1)^l \binom{k}{l} \binom{2k-2l}{k} x^{k-2l}$$

where $[k/2]$ denotes the integral part of $k/2$. Besides, the Legendre polynomials satisfy the recurrence relation:

$$\begin{aligned} \varphi_0(x) &= 1 \\ \varphi_1(x) &= x \\ \varphi_{k+1}(x) &= \frac{2k+1}{k+1} x \varphi_k(x) - \frac{k}{k+1} \varphi_{k-1}(x), \end{aligned}$$

for $k = 1, 2, \cdots,\ x \in [-1, 1]$. It is easy to see that Legendre polynomials are a system of complete and orthogonal polynomials in $L^2([-1, 1])$, and their inner product is as follows:

$$\langle \varphi_k(x), \varphi_j(x) \rangle = \begin{cases} \dfrac{2}{2k+1}, & \text{if } j = k \\ 0, & \text{otherwise.} \end{cases}$$

Let us denote

$$\begin{aligned} S_N &:= \text{span}\{\varphi_0, \varphi_1, \cdots, \varphi_N\}, \\ V_N &:= \{v \in S_N : v(-1) = v(1) = 0\}. \end{aligned}$$

Apparently $V_N \subset S_N$.

For high dimensional case, i.e., $U = [-1, 1]^n,\ n \geq 2$, we use S_N^n to represent the space spanned by multivariate Legendre polynomials of degree up to N of x. Let us define the tensor product of the Legendre polynomials:

$$\boldsymbol{\varphi_k}(x) := \varphi_{k_1}(x_1) \cdot \varphi_{k_2}(x_2) \cdots \varphi_{k_n}(x_n),$$

$\mathbf{k} = (k_1, \cdots, k_n) \in \{0, 1, \cdots, N\}^n$. And we denote the orthogonal projection operator from $L^2(U)$ upon S_N^n, i.e., $\mathcal{P}_N : L^2(U) \to S_N^n$, such that for all $v \in L^2(U)$,

$$\begin{aligned} \mathcal{P}_N v &= \sum_{\|k\|_\infty \leq N} w_{\boldsymbol{k}} \boldsymbol{\varphi_k}(x), \\ \langle v - \mathcal{P}_N v, \phi \rangle &= 0, \quad \forall \phi \in S_N^n. \end{aligned} \tag{48}$$

Similarly, we can define

$$V_N^n := \left\{ v \in S_N^n : v\big|_{\partial U} = 0 \right\}.$$

The subspace V_N^n can be spanned by the multivariate generalized Legendre polynomials introduced in [32]. The one dimensional generalized Legendre polynomials $\{\phi_k(x)\}$ is defined in the following lemma.

Lemma 1 [32]: Let us define the generalized Legendre polynomials $\{\phi_k(x)\}$ as

$$\begin{aligned} \phi_k(x) &:= c_k \left(\varphi_k(x) - \varphi_{k+2}(x) \right), \\ c_k &= \frac{1}{\sqrt{4k+6}}, \ k \in \mathbb{N}, \end{aligned}$$

$$b_{k,j} := \langle \phi_k(x), \phi_j(x) \rangle, \ k, j \in \mathbb{N}. \tag{49}$$

Then

$$b_{j,k} = b_{k,j} = \begin{cases} c_k c_j \left(\frac{2}{2k+1} + \frac{2}{2k+5} \right), & k = j, \\ -c_k c_j \frac{2}{2j+1}, & k = j-2, \\ 0 & \textit{otherwise}, \end{cases}$$

and

$$\begin{aligned} V_N^n &= \text{span} \Big\{ \phi_{\tilde{k}_1}(x) \cdot \phi_{\tilde{k}_2}(x) \cdots \phi_{\tilde{k}_n}(x) : \\ &\qquad 0 \le \tilde{k}_j \le N-2, \ j = 1, 2, \cdots, n \Big\} \\ &:= \text{span} \Big\{ \boldsymbol{\phi}_{\tilde{\mathbf{k}}}(\mathbf{x}) : \tilde{\mathbf{k}} \in \{0, 1, \cdots, N-2\}^n, \mathbf{x} \in \mathbb{R}^n \Big\}. \end{aligned}$$

ACKNOWLEDGMENT

Prof. Stephen Shing-Toung Yau is grateful to the National Center for Theoretical Sciences (NCTS) for providing an excellent research environment while part of this research was done. *(Xiuqiong Chen is the co-first author.)* This article is dedicated to Professor Tyrone Duncan on the occasion of his 80th Birthday.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problem," *J. Basic Eng.*, vol. 82, pp. 35–45, Mar. 1960.
[2] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng.*, vol. 83, no. 1, pp. 95–108, Mar. 1961.
[3] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. London, U.K.: Academic, 1970.
[4] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.
[5] P. L. Houtekamer and H. L. Mitchell, "Data assimilation using an ensemble Kalman filter technique," *Monthly Weather Rev.*, vol. 126, no. 3, pp. 796–811, 1998.
[6] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. F, Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, Apr. 1993.
[7] T. E. Duncan, "Probability densities for diffusion processes with applications to nonlinear filtering theory and detection theory," Stanford California, Stanford, CA, USA, Tech. Rep. TR-7001-4, May 1967.
[8] R. E. Mortensen, "Optimal control of continuous time stochastic systems," Ph.D. dissertation, Electron. Res. Lab., California Univ. Berkeley, Berkley, CA, USA, Aug. 1966.
[9] M. Zakai, "On the optimal filtering of diffusion processes," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 11, no. 3, pp. 230–243, 1969.
[10] V. E. Beneš, "Exact finite-dimensional filters for certain diffusions with nonlinear drift," *Stochastics*, vol. 5, nos. 1–2, pp. 65–92, Jun. 1981.
[11] W. S. Wong, "New classes of finite-dimensional nonlinear filters," *Syst. Control Lett.*, vol. 3, no. 3, pp. 155–164, Aug. 1983.
[12] S. S.-T. Yau, "Finite-dimensional filters with nonlinear drift. I: A class of filters including both Kalman-Bucy and Benes filters," *J. Math. Syst. Estimation Control*, vol. 4, no. 2, pp. 181–203, 1994.
[13] S. S.-T. Yau and G.-Q. Hu, "Classification of finite-dimensional estimation algebras of maximal rank with arbitrary state–space dimension and Mitter conjecture," *Int. J. Control*, vol. 78, no. 10, pp. 689–705, Jul. 2005.
[14] A. Bensoussan, R. Glowinski, and A. Rascanu, "Approximation of the Zakaï equation by the splitting up method," *SIAM J. Control Optim.*, vol. 28, no. 6, pp. 1420–1431, Nov. 1990.
[15] A. Bensoussan, R. Glowinski, and A. Raşcanu, "Approximation of some stochastic differential equations by the splitting up method," *Appl. Math. Optim.*, vol. 25, no. 1, pp. 81–106, Jan. 1992.
[16] 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.
[17] K. Ito, "Approximation of the Zakai equation for nonlinear filtering," *SIAM J. Control Optim.*, vol. 34, no. 2, pp. 620–634, Mar. 1996.
[18] K. Ito and B. Rozovskii, "Approximation of the Kushner equation for nonlinear filtering," *SIAM J. Control Optim.*, vol. 38, no. 3, pp. 893–915, Jan. 2000.
[19] 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, Nov. 1995.
[20] 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, Mar. 1997.
[21] S. S.-T. Yau and S. T. Yau, "New direct method for Kalman-bucy filtering system with arbitrary initial condition," in *Proc. 33rd IEEE Conf. Decis. Control*, vol. 2, Dec. 1994, pp. 1221–1225.
[22] S. S.-T. Yau and Y.-T. Lai, "Explicit solution of DMZ equation in nonlinear filtering via solution of ODEs," *IEEE Trans. Autom. Control*, vol. 48, no. 3, pp. 505–508, Mar. 2003.
[23] S. T. Yau and S. S.-T. Yau, "Nonlinear filtering and time varying Schrodinger equation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 40, pp. 284–292, 2004.
[24] S. S.-T. Yau, C. Yan, and S.-T. Yau, "Linear filtering with nonlinear observations," in *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, vol. 2, Dec. 2004, pp. 2112–2117.
[25] J. Shi, Z. Yang, and S. S. T. Yau, "Direct method for Yau filtering system with nonlinear observations," *Int. J. Control*, vol. 91, no. 3, pp. 678–687, Mar. 2018.
[26] X. Chen, J. Shi, and S. S.-T. Yau, "Real-time solution of time-varying Yau filtering problems via direct method and Gaussian approximation," *IEEE Trans. Autom. Control*, vol. 64, no. 4, pp. 1648–1654, Apr. 2019.
[27] 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, Jan. 2008.
[28] 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. Autom. Control*, vol. 58, no. 10, pp. 2495–2507, Oct. 2013.
[29] M. Hazewinkel, S. I. Marcus, and H. J. Sussmann, "Nonexistence of finite-dimensional filters for conditional statistics of the cubic sensor problem," *Syst. Control Lett.*, vol. 3, no. 6, pp. 331–340, Dec. 1983.
[30] M. H. A. Davis, "On a multiplicative functional transformation arising in nonlinear filtering theory," *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 54, no. 2, pp. 125–139, 1980.
[31] X. Luo and S. S.-T. Yau, "Complete real time solution of the general nonlinear filtering problem without memory," *IEEE Trans. Autom. Control*, vol. 58, no. 10, pp. 2563–2578, Oct. 2013.
[32] J. Shen, "Efficient spectral-Galerkin method I. Direct solvers of second- and fourth-order equations using Legendre polynomials," *SIAM J. Sci. Comput.*, vol. 15, no. 6, pp. 1489–1505, Nov. 1994.
[33] X. Luo, X. Chen, and S. S.-T. Yau, "Suboptimal linear estimation for continuous–discrete bilinear systems," *Syst. Control Lett.*, vol. 119, pp. 92–100, Sep. 2018.
[34] S. Sarkka, "On unscented Kalman filtering for state estimation of continuous-time nonlinear systems," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1631–1641, Sep. 2007.
[35] X. Chang and G. Yang, "Robust H-infinity filtering for uncertain discrete-time systems using parameter-dependent Lyapunov functions," *J. Control Theory Appl.*, vol. 11, no. 1, pp. 122–127, Feb. 2013.
[36] X.-H. Chang and G.-H. Yang, "Nonfragile H_∞ filter design for T–S fuzzy systems in standard form," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3448–3458, Jul. 2014.

JI SHI received the B.S. degree in information and computing science from North China Electric Power University, Beijing, China, in 2013, and the Ph.D. degree in applied mathematics from the Department of Mathematical Sciences, Tsinghua University, Beijing. He currently holds a post-doctoral position with the Academy of Multi-disciplinary Studies, Capital Normal University, Beijing.

His research interests include nonlinear filtering and active disturbance rejection control.

XIUQIONG CHEN (Member, IEEE) received the B.S. degree from the School of Mathematics and Systems Science, Beihang University, Beijing, China, in 2014, and the Ph.D. degree in applied mathematics from the Department of Mathematical Sciences, Tsinghua University, Beijing. She currently holds a postdoctoral position with the Yau Mathematical Sciences Center, Tsinghua University, Beijing.

Her research interests include nonlinear filtering and deep learning.

STEPHEN SHING-TOUNG YAU (Fellow, IEEE) received the Ph.D. degree in mathematics from The State University of New York, Stony Brook, NY, USA, in 1976.

From 1976 to 1977 and from 1981 to 1982, he was a member of the Institute of Advanced Study, Princeton. From 1977 to 1980, he was a Benjamin Pierce Assistant Professor with Harvard University. After that, he joined the Department of Mathematics, Statistics and Computer Science (MSCS), University of Illinois at Chicago (UIC), and served for more than 30 years. From 2005 to 2011, he was a joint Professor with the Department of Electrical and Computer Engineering and the MSCS, UIC. After his retirement in 2011, he joined Tsinghua University, Beijing, China, where he is a full-time Professor with the Department of Mathematical Sciences. His research interests include nonlinear filtering, bioinformatics, complex algebraic geometry, CR geometry, and singularities theory.

Dr. Yau was awarded the Sloan Fellowship in 1980, the Guggenheim Fellowship in 2000, and the AMS Fellow Award in 2013. He was the General Chairman of the IEEE International Conference on Control and Information, which was held in The Chinese University of Hong Kong, in 1995. Since 1991, he has been the Managing Editor and the Founder of the *Journal of Algebraic Geometry*. Since 2000, he has been the Editor-in-Chief and the Founder of *Communications in Information and Systems*. In 2005, he was entitled the UIC Distinguished Professor.