Communications in Information and Systems
Volume 26, Number 1, 119–149, 2026

A tutorial on Yau-Yau nonlinear filtering algorithm: from model-driven to data-driven

Zeju Sun, Jiayi Kang, Xiuqiong Chen, and Stephen S.-T. Yau*

Filtering is a subject of providing sequential estimations of a given stochastic dynamical system based on noisy observations. At the beginning of this century, a two-stage algorithm framework was proposed and analyzed for general nonlinear filtering problems, which is now referred to as the Yau-Yau algorithm. The two-stage structure of this framework theoretically guarantees the potential of solving nonlinear filtering problems in a real-time manner, which is crucial for practical applications. With the introduction of spectral method and neural networks, numerous model-driven and data-driven implementations of Yau-Yau algorithms have been proposed in the last decade. In this paper, we will present a thorough review of the development of nonlinear filters under the Yau-Yau algorithm framework, from model-driven approaches to data-driven approaches, which serves as a guidance for practitioners. Current status and promising future directions in the research of Yau-Yau nonlinear filtering algorithm framework are also summarized and discussed in this paper.

MSC 2020 subject classifications: 60G35, 93E11, 68T07, 65M70.
Keywords and phrases: Nonlinear filtering problem, Yau-Yau algorithm, DMZ equation, recurrent neural network, spectral method.

1. Introduction

Filtering is a subject about the sequential estimation of the state of a stochastic dynamical system based on noisy observations [27, 6]. The origin of modern filtering problems could date back to the middle of 20th century, when people started to use state-space models (SSM) to systematically describe and analyze the problem of sequential state estimations [57]. After nearly a century of development in theory and algorithms, nowadays, filtering has wide applications in various areas such as aerospace industrial [46], target tracking [1, 52], autonomous driving [15], weather forecasting [9], mathematical finance [13]

*Corresponding Author.

and so on. In this context, the study of filtering theory and algorithms has long been an important topic in applied mathematics and control theory. The progresses and breakthroughs in filtering theory and algorithms also make a great difference to the technological innovations in those related application areas.

In 1960s, the notable Kalman filter was proposed [31, 32], which theoretically solves the state estimation problem for linear-Gaussian systems (that is, the evolution of the state is governed by a linear stochastic dynamical system, the observation function is linear, and the noise in the system is modeled as Gaussian-type). The establishment of Kalman filter signifies the beginning of systematic research and application of modern filtering theory and algorithms within both the academic and industrial realms. Nevertheless, in real application scenarios, the dynamical systems in filtering problems are often nonlinear and the noise in the system may not be ideally modeled as Gaussian-type. In this case, the solution of these general nonlinear filtering problems cannot be expressed in an explicit form as Kalman filter. The study of nonlinear filtering theory and algorithms is still a major issue in modern control community.

In the sense of mean square errors (MSE), the optimal estimations of state process based on historical observations are the conditional expectations [6]. If the filtering problem is modeled in discrete-time setting, then the evolution of the conditional expectations can be formally expressed by the Bayesian formulae [27]. For practical implementations, although the evolution equations cannot be computed directly in general, sub-optimal estimations of the state processes can be obtained by approximating the theoretical expression with linearization or Monte-Carlo techniques, which results in the two most famous kinds of nonlinear filtering algorithms: Kalman-based algorithms [8, 44, 53, 3, 19] and particle filters [4, 59, 58, 50, 49, 33, 2].

For general nonlinear filtering problems in continuous-time setting, in the late 1960s, Duncan [18], Mortensen [42] and Zakai [65] independently derived the stochastic partial differential equation satisfied by the unnormalized conditional probability density function, which is now referred to as the Duncan-Mortensen-Zakai (DMZ) equation. After numerically solving the DMZ equation, the optimal estimations of the state process, i.e., the conditional expectations, can be obtained by computing the normalized integral. Therefore, the introduction of DMZ equation paved the way of the applications of numerical methods to partial differential equations (PDE) in filtering algorithms. Later on, a great many PDE-based nonlinear filtering algorithms have been proposed and studied in details [20, 7, 5, 22, 14].

Because in many application scenarios such as aerospace industrial, it is highly demanding that filtering problems can be solved in a real-time manner, the efficiency of solving the DMZ equation is essential for PDE-based

filtering algorithms. The idea of operator decomposition methods is dividing the complicated differential operators in PDEs into several relatively simple parts and dealing with each part respectively. The introduction of operator decomposition methods to nonlinear filtering algorithms can date back to the end of 20th century [20, 38, 14]. Later on, a two-stage nonlinear filtering algorithm was proposed by Yau and Yau [61, 62], which is now referred to as the Yau-Yau nonlinear filtering algorithm. Yau-Yau nonlinear filtering algorithm successfully applies the idea of operator decomposition, and the computationally expensive part of the DMZ equation is observation-free, and can be studied and solved offline. In this way, the DMZ equation, together with the nonlinear filtering problem, can theoretically be solved in a real-time manner. Theoretically, the Yau-Yau algorithm provides a pathway to extend PDE theory to filtering theory. For example, it enables the extension of the convex maximum principle for parabolic PDEs to the concept of the convex filter [34].

The offline-part of Yau-Yau algorithm consists of solving a Kolmogorov equation which does not depend on the observations. Although the Kolmogorov equation can be solved offline, the process of solving and transferring the solutions to the online-part is still challenging especially for general high-dimensional nonlinear filtering problems. With the development of artificial intelligence, the capability of neural networks and deep learning techniques to dealing with high-dimensional problems is attracting more and more attentions in the control community. Recently, many deep learning based nonlinear filtering algorithms have been proposed [11, 51, 55, 21, 28, 54, 63]. These algorithms outperform classical filters in numerical experiments especially for problems with high dimension and high nonlinearity. Nevertheless, the mathematical theory of deep-learning based nonlinear filtering algorithms has not yet been systematically established, which also restricts the spread of these kind of algorithms in application areas. In this context, deep learning based algorithms under the framework of Yau-Yau nonlinear filtering algorithms have been proposed and studied [10]. With the help of the solid mathematical foundation of Yau-Yau algorithm, such kind of algorithms has the potential of becoming the first mathematically interpretable deep learning based nonlinear filtering algorithms.

As a tutorial and review paper of Yau-Yau nonlinear filtering algorithms, this paper will summarize the theory and implementations of the filtering algorithms under the Yau-Yau framework from the aspect of model-driven to data-driven. The study of the model-driven Yau-Yau algorithm includes the convergence analysis of its two-stage framework, as well as numerical methods of solving the Kolmogorov equation, while the data-driven Yau-Yau algorithm

Table 1: The contents and major references of this tutorial

Yau-Yau Nonlinear Filtering Algorithm		
Contents	Topics	Major References
Theory	Time-invariant system	[61, 62]
	Time-varying system	[39]
Model-driven algorithms	Finite difference	[64]
	Wavelet Galerkin	[37]
	Orthogonal basis	[40, 17, 56]
	Tensor training	[36]
	PINN	[47, 48]
Data-driven algorithms	Uniform framework	[10]

consists of the deep-learning implementation as well as its capability of dealing with high-dimensional problems.

The organization of this paper is as follows. In Section 2, the preliminaries of general continuous-time nonlinear filtering problems will be presented, which includes the filtering system, the DMZ equation, and the details of the two-stage Yau-Yau algorithm. Section 3 will focus on the model-driven part of the Yau-Yau algorithm and will discuss the numerical methods of solving the Kolmogorov equations in the Yau-Yau algorithm. Section 4 will summarize the data-driven aspect of the Yau-Yau algorithm, and focus on the deep-learning implementation of the algorithm for high-dimensional problems. Finally, Section 5 is the conclusion and future prospects of the study of Yau-Yau algorithm and nonlinear filtering.

Based on the organization of this paper, the contents and major references of this tutorial are listed in Table 1.

2. Preliminaries

In this section, we will start with the state-space model of general continuous-time filtering problems, give a sketch of the derivation of the DMZ equation satisfied by the unnormalized conditional probability density functions, and then introduce the specific form of Yau-Yau nonlinear filtering algorithm framework.

2.1. Continuous-time filtering problem

In general, the state-space model of a continuous-time filtering problem can be described by the following coupled stochastic differential equations:

$$\begin{cases} dX_t = f(t, X_t)dt + g(t, X_t)dV_t, \quad X_0 = \xi, \\ dY_t = h(t, X_t)dt + dW_t, \quad Y_0 = 0, \end{cases} \quad t \in [0, T], \tag{1}$$

defined on a filtrated probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{0\le t\le T}, P)$, where $X = \{X_t : 0 \le t \le T\} \subset \mathbb{R}^d$ and $Y = \{Y_t : 0 \le t \le T\} \subset \mathbb{R}^d$ are $\mathcal{F}_t$-adapted stochastic processes, representing the state and observation of the system, respectively; $V = \{V_t : 0 \le t \le T\} \subset \mathbb{R}^d$ and $W = \{W_t : 0 \le t \le T\} \subset \mathbb{R}^d$ are mutually independent, $\mathcal{F}_t$-adapted standard d-dimensional Brownian motions; ξ is an $\mathcal{F}_0$-measurable, $\mathbb{R}^d$-valued random variable which is independent of V and W, with probability density function $\sigma_0(x) \in \mathcal{L}^2(\mathbb{R}^d)$; $f : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$, $g : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ and $h : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^d$ are smooth enough vector- or matrix-valued functions. If the functions f, g, h do not explicitly depend on the time variable t, then system (1) is called a time-invariant system; otherwise, it is called a time-varying system.

The main goal of filtering is to give accurate estimations of the state process X_t at each time $t \in [0, T]$ based on the historical observations $\{Y_s : 0 \le s \le t\}$. In the sense of expected mean square errors (MSE), given a test function $\varphi(X_t)$ such that $\varphi(X_t)$ is integrable with respect to the probability measure P, the optimal estimation of X_t based on $\{Y_s : 0 \le s \le t\}$ is the conditional expectation $E[\varphi(X_t)|\mathcal{Y}_t]$, where $\mathcal{Y}_t \triangleq \sigma\{Y_s : 0 \le s \le t\}$ is the σ-algebra generated by the historical observations. In fact, according to the projection property of conditional expectations, we have

$$E[\varphi(X_t)|\mathcal{Y}_t] = \underset{U \text{ is } \mathcal{Y}_t\text{-measurable}}{\arg\min} E[(\varphi(X_t) - U)^2]. \tag{2}$$

If the conditional probability distribution $P(X_t \in \cdot|\mathcal{Y}_t)$ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$ with the density function $p(t, x)$, then the conditional expectations can be computed by

$$E[\varphi(X_t)|\mathcal{Y}_t] = \int_{\mathbb{R}^d} \varphi(x)p(t, x)dx, \tag{3}$$

and thus, a major task in filtering problem is to compute the evolution of the conditional probability density functions.

In linear Gaussian cases, where the functions $f(t, x) = F_t x$ and $h(t, x) = H_t x$ are linear with respect to x, $g(t, x) \equiv G_t$ is a constant matrix with respect to x, and the initial value ξ is a normal random variable, the conditional distribution $P(X_t \in \cdot|\mathcal{Y}_t)$ will remain Gaussian with $p(t, x) = \mathcal{N}(x; \hat{x}_t, P_t)$, the probability density function of a normal random variable with mean $\hat{x}_t$ and covariance matrix P_t. The evolution of $\hat{x}_t$ and covariance matrix P_t will satisfy the notable Kalman-Bucy filter [32]:

$$\begin{cases} d\hat{x}_t = F_t\hat{x}_t dt + K_t(dY_t - H_t\hat{x}_t dt) \\ \dfrac{dP_t}{dt} = F_tP_t + P_tF_t^\top + G_tG_t^\top - P_tH_t^\top H_tP_t \end{cases} \tag{4}$$

where $K_t \triangleq P_t H_t^\top$ is the Kalman gain matrix.

For general nonlinear systems, however, the conditional distribution do not remain Gaussian, and even do not possess a finite-dimensional sufficient statistics [24]. Therefore, the evolution of conditional probability density functions cannot be computed explicitly in general. In the next section, we will show that the evolution of conditional probability density function, without considering a normalizing constant, will satisfy a linear stochastic partial differential equation driven by the observation process, which is now referred to as the DMZ equation.

2.2. Unnormalized conditional probability density and the DMZ equation

In this subsection, we will introduce the evolution equation satisfied by the conditional probability density function without considering a normalizing constant. The evolution equation is now referred to as DMZ equation and the solution of DMZ equation is called the 'unnormalized conditional probability density function'. As for the derivation of the DMZ equation, we would like to follow the change-of-measure approach based on the Girsanov's transformation, which has been discussed in detail in the related monographs [6, 60].

Let us first consider the stochastic process $Z = \{Z_t : 0 \le t \le T\}$ which is defined by

$$Z_t = \exp\left(-\int_0^t h^\top(s, X_s) dW_s - \frac{1}{2}\int_0^t |h(s, X_s)|^2 ds\right), \ t \in [0, T]. \tag{5}$$

Under mild conditions, the process Z_t is an $\mathcal{F}_t$-adapted martingale. In this case, according to the Girsanov's theorem [35], we can introduce a series of reference probability measures $\{\tilde{P}_t : 0 \le t \le T\}$, with each $\tilde{P}_t$ defined on $\mathcal{F}_t$ and satisfying

$$Z_t = \left.\frac{d\tilde{P}_t}{dP}\right|_{\mathcal{F}_t}, \tag{6}$$

and under $\tilde{P}_T$, the observation process Y is a standard d-dimensional Brownian motion independent of X.

The inverse of Z_t, denoted by $\tilde{Z}_t$ can then be written as a functional of

Y as

$$\begin{aligned}\tilde{Z}_t = Z_t^{-1} &= \left.\frac{dP}{d\tilde{P}_t}\right|_{\mathcal{F}_t} \\ &= \exp\left(\int_0^t h^\top(s, X_s)dY_s - \frac{1}{2}\int_0^t |h(s, X_s)|^2 ds\right),\ t \in [0, T].\end{aligned} \tag{7}$$

and for each $\mathcal{F}_t$-measurable and integrable random variable $U \in \mathcal{F}_t$, its expectation with respect to P can then be computed as

$$E[U] = \tilde{E}\left[\tilde{Z}_t U\right], \tag{8}$$

where $\tilde{E}$ is the expectation with respect to $\tilde{P}_T$. Moreover, the conditional expectations $E[\varphi(X_t)|\mathcal{Y}_t]$ can also be written as the ratio of two expectations with respect to the reference probability measure $\tilde{P}$:

$$E[\varphi(X_t)|\mathcal{Y}_t] = \frac{\tilde{E}\left[\tilde{Z}_t\varphi(X_t)|\mathcal{Y}_t\right]}{\tilde{E}\left[\tilde{Z}_t|\mathcal{Y}_t\right]},\ t \in [0, T]. \tag{9}$$

The formula (9) is now called the Kallianpur-Striebel formula [29, 30], and can be regarded as a generalization of Bayesian formula in the context of continuous-time stochastic processes.

Because the denominator on the right-hand side of (9) is independent of the test function φ, the numerator $\tilde{E}[\tilde{Z}_t\varphi(X_t)|\mathcal{Y}_t]$ is often called the unnormalized conditional expectation of $\varphi(X_t)$ and the measure ρ_t defined by

$$\rho_t(\varphi) = \tilde{E}\left[\tilde{Z}_t\varphi(X_t)|\mathcal{Y}_t\right],\ \forall\varphi \in C_b(\mathbb{R}^d) \tag{10}$$

is then called the unnormalized conditional probability measure, where $C_b(\mathbb{R}^d)$ denotes the set of all bounded continuous functions in $\mathbb{R}^d$.

If the unnormalized conditional probability measure ρ_t is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$ and the density function (or Radon derivative) $\sigma(t, x)$, together with its first-order derivatives are square-integrable, then $\sigma(t, x)$ is the solution of the following stochastic partial differential equation driven by the observation process:

$$\begin{cases} d\sigma(t, x) = \mathcal{L}_t^*\sigma(t, x)dt + \sum_{j=1}^d h_j(t, x)\sigma(t, x)dY_t^j,\ t \in [0, T], \\ \sigma(0, x) = \sigma_0(x), \end{cases} \tag{11}$$

where

$$\mathcal{L}_t^*(\star) = \frac{1}{2}\sum_{i,j=1}^{d}\frac{\partial^2}{\partial x_i \partial x_j}(a^{ij}(t,\cdot)\star) - \sum_{i=1}^{d}\frac{\partial}{\partial x_i}(f_i(t,\cdot)\star). \tag{12}$$

is the second-order elliptic differential operator with

$$a(t,x) := (a^{ij}(t,x))_{1\le i,j\le d} = g(t,x)g(t,x)^\top.$$

Equation (11) is now referred to as the DMZ equation corresponding to the filtering system (1), and the solution of (11), $\sigma(t,x)$ is called the unnormalized conditional probability density function.

Although the DMZ equation (11) is a stochastic partial differential equation, if we introduce the exponential transformation [16]:

$$w(t,x) := \exp\left(-h^\top(t,x)Y_t\right)\sigma(t,x),\ t\in[0,T], \tag{13}$$

then the resulting function $w(t,x)$ will satisfies

$$\begin{cases}\dfrac{\partial w}{\partial t} = \dfrac{1}{2}\displaystyle\sum_{i,j=1}^{d} a^{ij}(x)\frac{\partial^2 w}{\partial x_i \partial x_j} + \sum_{i=1}^{d} F_i(t,x)\frac{\partial w}{\partial x_i} + J(t,x)w(t,x),\ t\in[0,T] \\ w(0,x) = \sigma_0(x)\end{cases} \tag{14}$$

with

$$F_i(t,x) = \sum_{j=1}^{d}\left(\frac{\partial a^{ij}}{\partial x_j} + a^{ij}\sum_{k=1}^{d}Y_t^k\frac{\partial h_k}{\partial x_j}\right) - f_i,\quad i=1,\cdots,d, \tag{15}$$

and

$$\begin{aligned} J(t,x) =& \frac{1}{2}\sum_{i,j=1}^{d}\frac{\partial^2 a^{ij}}{\partial x_i\partial x_j} + \sum_{i,j,k=1}^{d}Y_t^k\frac{\partial h_k}{\partial x_j}\frac{\partial a^{ij}}{\partial x_i} \\ &+ \frac{1}{2}\sum_{i,j=1}^{d}a^{ij}\left(\sum_{k=1}^{d}Y_t^k\frac{\partial^2 h_k}{\partial x_i\partial x_j} + \sum_{k=1}^{d}\sum_{l=1}^{d}Y_t^kY_t^l\frac{\partial h_k}{\partial x_i}\frac{\partial h_l}{\partial x_j}\right) \\ &- \sum_{i=1}^{d}\frac{\partial f_i}{\partial x_i} - \sum_{i,j=1}^{d}Y_t^j\frac{\partial h_j}{\partial x_i}f_i - \frac{1}{2}|h|^2 - Y_t^\top\frac{\partial h}{\partial t} \end{aligned} \tag{16}$$

In comparison with the original DMZ equation (11), the stochastic differential terms are eliminated in (14), which will result in a more robust equation when conducting the time-discretization. Therefore, the deterministic partial differential equation with stochastic coefficients (14) is called the robust DMZ equation corresponding to the filtering system (1).

Starting from the robust DMZ equation (14), Yau and Yau [61, 62] proposed a two-stage numerical method of solving the original DMZ equation (11) as well as the filtering problem, which is now referred to as the **Yau-Yau nonlinear filtering algorithm** (or 'Yau-Yau algorithm' for short). The idea and framework of Yau-Yau algorithm will be presented in the next subsection.

2.3. Yau-Yau nonlinear filtering algorithm

The idea of Yau-Yau nonlinear filtering algorithm is to decompose the task of solving the DMZ equation into two parts. This algorithm decomposition is designed such that the computationally expensive part of numerically solving partial differential equations is independent of the real-time observations, and therefore, can be done offline in advance. In this way, the real-time computation cost is sharply declined while the accuracy of the solution is largely maintained.

Let $0 = \tau_0 < \tau_1 < \cdots < \tau_K = T$ be a uniform partition of the time interval $[0, T]$, with $\tau_k - \tau_{k-1} = \delta = \frac{T}{K}$, $k = 1, \cdots, K$. At each time interval $[\tau_{k-1}, \tau_k]$, in order to solve the robust DMZ equation (14) numerically, we will restrict the equation to a large ball $B_R = \{x \in \mathbb{R}^d : |x| \leq R\}$ and freeze the observation-dependent part of the coefficients $F(t, x)$ and $J(t, x)$ to the left endpoint $t = \tau_{k-1}$. The initial-value problem of the robust DMZ equation (14) is then converted into a initial-boundary value problem on the ball B_R:

(17)
$$\begin{cases} \dfrac{\partial u_k}{\partial t} = \dfrac{1}{2}\displaystyle\sum_{i,j=1}^{d} a^{ij}(x)\frac{\partial^2 u_k}{\partial x_i \partial x_j} + \sum_{i=1}^{d} \tilde{F}_i(t, \tau_{k-1}, x)\frac{\partial u_k}{\partial x_i} + \tilde{J}(t, \tau_{k-1}, x)u_k(t, x), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (t, x) \in (\tau_{k-1}, \tau_k] \times B_R, \\ u_k(\tau_{k-1}, x) = u_{k-1}(\tau_{k-1}, x), \ x \in B_R, \\ u_k(t, x) = 0, \ (t, x) \in [\tau_{k-1}, \tau_k] \times \partial B_R, \end{cases}$$

where $\tilde{F}(t, \tau_{k-1}, x)$ and $\tilde{J}(t, \tau_{k-1}, x)$ are the same functions as $F(t, x)$ and $J(t, x)$, except that each Y_t is substituted by $Y_{\tau_{k-1}}$ in the definitions (15) and (16), respectively.

The following theorem states that under mild conditions, the solution of the initial-boundary value problem (17) can approximate the solution of the robust DMZ equation (14) well, as long as the time-discretization step δ is small enough. The time-invariant version of the theorem was first presented and proved in [62]. Later on, Luo and Yau [39] obtained the result for the time-variant case.

Theorem 2.1 ([62, 39]). *With mild assumptions on the coefficients of the filtering system* (1) *and the observation trajectory* $\{Y_t : 0 \le t \le T\}$*, as is stated in [62, 39], the solution* u_k *of the initial-boundary value problem* (17) *will converge to the solution* $w(t,x)$ *of the robust DMZ equation* (14) *on the time interval* $[\tau_{k-1}, \tau_k]$*, as the time-discretization step* $\delta \to 0$ *and the radius* $R \to \infty$*. Especially, we have the following estimation:*

(18)
$$\int_{\mathbb{R}^d} |w(t,x) - u_k(t,x)|dx \le c_1 e^{-c_2 R} + c_3\sqrt{\delta},\ \forall\ t \in [\tau_{k-1}, \tau_k],\ k = 1, \cdots, K.$$

where $c_1, c_3 > 0$ *are generalized constant which depends on the coefficients of the filtering system, the terminal time* T*, and the regularity of the observation trajectory* $\{Y_t : 0 \le t \le T\}$*;* $c_2 > 0$ *is a constant. Here, we define* $u_k(t,x) \equiv 0$ *when* $|x| > R$*.*

Since the solution of (17) is a good approximation of (14), instead of solving the robust DMZ equation (14) directly, in the framework of Yau-Yau algorithm, we will deal with (17) for large $R > 0$ and small $\delta > 0$. The benefits of this approximation will be revealed if we consider another exponential transformation for the solution $u_k(t,x)$ of (17) on $[\tau_{k-1}, \tau_k]$:

$$\tilde{u}_k(t,x) = \exp\left(h^\top(t,x) Y_{\tau_{k-1}}\right) u_k(t,x),\ t \in [\tau_{k-1}, \tau_k], \tag{19}$$

because the equation satisfied by $\tilde{u}_k$:

$$\begin{aligned} \frac{\partial \tilde{u}_k}{\partial t} &= \frac{1}{2}\sum_{i,j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}(a^{ij}\tilde{u}_k(t,x)) \\ &\quad - \sum_{i=1}^{d} \frac{\partial}{\partial x_i}(f_i \tilde{u}_k(t,x)) - \frac{1}{2}|h|^2 \tilde{u}_k(t,x),\ (t,x) \in (\tau_{k-1}, \tau_k] \times B_R, \end{aligned} \tag{20}$$

is a Kolmogorov-type partial differential equation independent of the observations, it can be dealt with offline in advance.

According to the exponential transformations (15) and the initial value of (17), the initial value of (20) satisfied by $\tilde{u}_k$ can be recursively computed by

$$\begin{aligned}\tilde{u}_k(\tau_{k-1},x) &= \exp\left(h^\top(\tau_{k-1},x)Y_{\tau_{k-1}}\right) u_k(\tau_{k-1},x)\\ &= \exp\left(h^\top(\tau_{k-1},x)Y_{\tau_{k-1}}\right) u_{k-1}(\tau_{k-1},x)\\ &= \exp\left(h^\top(\tau_{k-1},x)(Y_{\tau_{k-1}}-Y_{\tau_{k-2}})\right) \tilde{u}_{k-1}(\tau_{k-1},x),\ k\geq 2.\end{aligned} \tag{21}$$

In summary, the robust DMZ equation (14) can be numerically solved by (i) dealing with the Kolmogorov equation (20) offline and (ii) computing the exponential transformation (21) when real-time observations are obtained.

If we further take the first exponential transformation (13) into consideration, then the solution of the original DMZ equation, $\sigma(\tau_{k-1},x)$, can be approximated by $\tilde{u}_k(\tau_{k-1},x)$ at each time $t=\tau_{k-1}$, $k=1,\cdots,K$. The two-stage algorithm of computing $\tilde{u}_k(\tau_{k-1},x)$ is now referred to as the Yau-Yau nonlinear filtering algorithm.

The Yau-Yau nonlinear filtering algorithm provides a framework of solving the DMZ equation as well as the general nonlinear filtering problem. Nevertheless, in order to implement Yau-Yau algorithm in practice, we still need to consider the specific methods of dealing with the Kolmogorov equation (20) and transferring the result to online computations. In the next two sections, we will review the existing studies on the implementation of Yau-Yau algorithm from the perspectives of model-driven and data-driven, respectively.

3. Model-driven Yau-Yau algorithm

The model-driven implementations of Yau-Yau algorithm mainly consists of numerically solving the Kolmogorov equation (20) with different initial values. In fact, many classical numerical methods can be used to solve (20) and will result in different implementations of Yau-Yau algorithm. For example, the finite-difference method is applied in [64] and the Wavelet-Galerkin approach is studied in [37].

As is remarked in [62], the aim of the offline step of Yau-Yau algorithm can be achieved by solving Kolmogorov equation (20) with initial values being an orthogonal basis $\{\phi_i\}$ of the Hilbert space $\mathcal{L}^2(\mathbb{R}^d)$, an then the online computation is to express an observation-dependent function as a linear combination of $\{\phi_i\}$ and conduct the exponential transformations. For practical implementations, this idea corresponds to the spectral methods of solving partial differential equations, in which the equation is first projected on a finite-dimensional subspace spanned by a properly-chosen orthogonal basis

and the solution is constructed and computed in this finite-dimensional subspace. In the rest of this section, we will mainly focus on this spectral-method based implementation of Yau-Yau algorithm.

3.1. Orthogonal basis and the spectral method

The idea of spectral method is to approximate functions with its projection on a finite-dimensional subspace, and the first step of implementing spectral method is to construct this subspace with a set of mutually orthogonal functions.

For a given domain $D \subset \mathbb{R}^d$, a set of polynomials $\{\tilde{\phi}_i\}_{i=0}^{\infty}$ with degree $\deg \tilde{\phi}_i = i$ are called an orthogonal basis of the Hilbert space $L^2(D)$ with weight function $w(x)$, if

$$\int_D \tilde{\phi}_i(x)\tilde{\phi}_j(x)w(x)dx = 0, \ \forall\ i \neq j. \tag{22}$$

The corresponding orthogonal basis functions $\{\phi_i\}_{i=0}^{\infty}$ can then be obtained by defining $\phi_i(x) = \tilde{\phi}_i(x)\sqrt{w(x)}$, and we have

$$\langle \phi_i, \phi_j \rangle \triangleq \int_D \phi_i(x)\phi_j(x)dx = 0, \ \forall\ i \neq j. \tag{23}$$

Different choices of the domain D and weight function $w(x)$ will bring about different orthogonal bases. Orthogonal bases which are commonly used in practice include:

- Hermite functions, with $D = \mathbb{R}^d$ and $w(x) = e^{-x^2}$;
- Legendre functions, with $D = [-1, 1]^d$ and $w(x) = 1$.

With a given number N, according to the orthogonal property of $\{\phi_i\}_{i=0}^{\infty}$, the projection of an arbitrary square-integrable function $F \in L^2(D)$ onto the $N+1$-dimensional subspace spanned by $\{\phi_i\}_{i=0}^{N}$ can be written as

$$\mathcal{P}_N F(x) = \sum_{i=0}^{N} \frac{\langle F, \phi_i \rangle}{\langle \phi_i, \phi_i \rangle} \phi_i(x). \tag{24}$$

For the univariate case $d = 1$, if the function F is smooth enough, the projection $\mathcal{P}_N F$ will converge to F rapidly as $N \to \infty$. This property is called the 'spectral accuracy'. For example, for Hermite functions and Legendre functions, we have the following theorems:

Theorem 3.1 ([40]). *Let us define the differential operator* $\mathcal{D}_x = \frac{\partial}{\partial x} + x$, *and the Sobolev-type space*

$$\mathcal{W}^m(\mathbb{R}) = \{u \in L^2(\mathbb{R}) : \ \mathcal{D}^k u \in L^2(\mathbb{R}), \ 0 \le k \le m\}, \ \forall \ m \in \mathbb{N}. \tag{25}$$

Then, the distance between a function $u \in \mathcal{W}^m(\mathbb{R})$ *and its projection* $\mathcal{P}_N^{(H)} u$ *on the* $(N+1)$*-dimensional subspace spanned by the Hermite functions up to degree* N *can be estimated by*

$$\|\mathcal{P}_N^{(H)} u - u\|_{L^2(\mathbb{R})} \le C N^{-\frac{m}{2}} |u|_{\mathcal{W}^m(\mathbb{R})}, \tag{26}$$

where $|\cdot|_{\mathcal{W}^m(\mathbb{R})}$ *is the seminorm defined by*

$$|u|_{\mathcal{W}^m(\mathbb{R})} = \|\mathcal{D}_x^m u\|_{L^2(\mathbb{R})}. \tag{27}$$

Theorem 3.2 ([17]). *Let* $H^r([-1,1])$ *be the Sobolev space which is defined by*

$$H^r([-1,1]) = \left\{ u \in L^2([-1,1]) : \frac{d^k u}{dx^k} \in L^2([-1,1]), \ 0 \le k \le r \right\}, \tag{28}$$

then, the distance between a function $u \in H^r([-1,1])$ *and its projection* $\mathcal{P}_N^{(L)} u$ *onto the* $(N+1)$*-dimensional subspace spanned by the Legendre polynomials up to degree* N *can be estimated by*

$$\|u - \mathcal{P}_N^{(L)} u\|_{L^2([-1,1])} \le C N^{-r} \|u\|_{H^r([-1,1])}. \tag{29}$$

Although spectral method often performs well in univariate case, its approximation efficiency may sharply decrease for high-dimensional problems. Because in high-dimensional space with $d > 1$, the orthogonal polynomials are expressed as the tensor products of one-dimensional polynomials, the orthogonal basis constructed in this way may suffer from the 'curse of dimensionality' for $d \gg 1$, in the sense that the number of the basis functions are required to grow exponentially with respect to d in order to maintain the approximation accuracy.

Apart from orthogonal polynomials, there are other approaches to construct basis functions which can improve the approximation efficiency in high-dimensional problems. Proper decomposition method (POD) constructs orthogonal basis from the snapshots of some reference solutions of the Kolmogorov equation [56]. Such a basis can extract the low-dimensional structure of a high-dimensional equation and reduce the number of basis functions in the spectral method.

3.2. Numerical solutions of the Kolmogorov equation

With properly chosen orthogonal basis $\{\phi_i\}_{i=1}^{\infty} \subset L^2(D)$, the Kolmogorov equation (20) can be numerically solved through standard Galerkin approach. Without loss of generality, in the rest of this section, we also assume that the orthogonal basis is normalized, that is,

$$\|\phi_i\|^2 = \langle \phi_i, \phi_i \rangle = 1, \ \forall\ i \in \mathbb{N}. \tag{30}$$

According to the Kolmogorov equation (20), for each smooth test function with compact support $\varphi \in C_0^{\infty}(D)$, we have

$$\frac{\partial}{\partial t}\langle \tilde{u}_k, \varphi \rangle = \frac{1}{2}\sum_{i,j=1}^{d} \langle a^{ij}\tilde{u}_k, \frac{\partial^2 \varphi}{\partial x_i \partial x_j} + \sum_{i=1}^{d} \langle f_i \tilde{u}_k, \frac{\partial \varphi}{\partial x_i} \rangle - \frac{1}{2}\langle |h|^2 \tilde{u}_k, \varphi \rangle. \tag{31}$$

Let us denote by V_N the subspace spanned by the orthogonal basis $\{\phi_i\}_{i=1}^{N}$. The Galerkin approximation of the solution of the Kolmogorov equation (20) is a function $v(t,x)$ with $v(t,\cdot) \in V_N$ for every $t \in [0,T]$, which satisfies

$$\frac{\partial}{\partial t}\langle v, \phi_l \rangle = \frac{1}{2}\sum_{i,j=1}^{d} \langle a^{ij} v, \frac{\partial^2 \phi_l}{\partial x_i \partial x_j} + \sum_{i=1}^{d} \langle f_i v, \frac{\partial \phi_l}{\partial x_i} \rangle - \frac{1}{2}\langle |h|^2 v, \phi_l \rangle, \ \forall\ 1 \le l \le N. \tag{32}$$

If we write $v(t,x) = \sum_{i=1}^{N} \lambda_i(t)\phi_i(x)$, then (32) implies that

$$\begin{aligned} \frac{d}{dt}\lambda_m(t) =& \frac{1}{2}\sum_{i,j=1}^{d}\sum_{l=1}^{N} \lambda_l(t) \langle a^{ij}\phi_l, \frac{\partial^2 \phi_m}{\partial x_i \partial x_j} \rangle \\ &+ \sum_{i=1}^{d}\sum_{l=1}^{N} \lambda_l(t) \langle f_i \phi_l, \frac{\partial \phi_m}{\partial x_i} \rangle - \frac{1}{2}\sum_{l=1}^{N} \lambda_l(t) \langle |h|^2 \phi_l, \phi_m \rangle, \end{aligned} \tag{33}$$

for all $1 \le m \le N$.

As a set of linear ordinary differential equations of the coefficients

$$\lambda(t) = [\lambda_1(t), \cdots, \lambda_N(t)]^{\top},$$

equation (33) can be solved either explicitly or numerically. Up to now, we have derived the numerical solution $v(t,x)$ of the Kolmogorov equation (20) in the Yau-Yau algorithm based on spectral method and Galerkin approximation.

Although the Galerkin approximation $v(t,x)$ is not the exact projection of $\tilde{u}_k(t,x)$ onto the N-dimensional subspace V_N, convergence results can also be obtained for classical Legendre polynomials [17], Hermite functions [41] as well as the proper orthogonal decomposition approach [56]. We demonstrate the convergence result of spectral method based on Legendre polynomials here as an example.

Theorem 3.3 ([17]). *Let Ω be a bounded domain, and assume that on $\Omega \times [0,T]$, the coefficients of the Kolmogorov equation (20) possess sufficient regularity, then for $r > 1$, if the solution $\tilde{u}_k \in L^2([\tau_{k-1},\tau_k], H^r(\Omega))$, then*

$$\|\tilde{u}_k(t,\cdot) - v(t,\cdot)\|^2 \le cN^{1-r}\|\tilde{u}_k(\tau_k,\cdot)\|_{H^r(\Omega)},\ \forall\ t \in [\tau_{k-1},\tau_k], \tag{34}$$

where v is the Galerkin approximation of $\tilde{u}_k$ with Legendre polynomials and c is a constant which depends only on $\tau_k - \tau_{k-1}, d$ and the coefficients of the Kolmogorov equations.

3.3. Neural networks and the spectral method

Consider the initial value problem for the Kolmogorov equation (20) in a bounded domain $\Omega \subset \mathbb{R}^d$ over $[0,T]$:

$$\begin{cases} \dfrac{\partial \tilde{u}_k}{\partial t} = \dfrac{1}{2}\displaystyle\sum_{i,j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}(a^{ij}\tilde{u}_k(t,x)) \\ \qquad - \displaystyle\sum_{i=1}^{d} \frac{\partial}{\partial x_i}(f_i \tilde{u}_k(t,x)) - \frac{1}{2}|h|^2 \tilde{u}_k(t,x),\ \ (t,x) \in (\tau_{k-1},\tau_k] \times \Omega, \\ \tilde{u}_k(\tau_{k-1},x) = p(x). \end{cases} \tag{35}$$

where $p(x)$ is the initial density, typically determined by the previous time step in an online algorithm, and homogeneous Dirichlet boundary conditions are imposed for simplicity.

Given the linearity of (35), the solution can be viewed as the action of a semigroup operator $S(\tau_{k-1},t)$ on the initial condition: $\tilde{u}_k(t,x)=S(\tau_{k-1},t)p(x)$. To compute $\tilde{u}_k(t,x)$, we can further assume $p(x) \in L^2(\Omega)$ and decompose $p(x)$ using an orthogonal basis $\{\phi_l(x)\}_{l=1}^{\infty} \subset L^2(\Omega)$, satisfying:

$$p(x) = \sum_{k=1}^{\infty} \langle \phi_k, p\rangle \phi_k(x). \tag{36}$$

Selecting a finite subset $\mathcal{K}_M = \{l_1, \ldots, l_M\} \subseteq \mathbb{N}$, the initial condition is approximated as:

$$p(x) \approx \sum_{l \in \mathcal{K}_M} c_l \phi_l(x), \quad c_l = \langle p, \phi_l \rangle_{L^2(\Omega)}. \tag{37}$$

By linearity, the solution is:

$$\tilde{u}_k(t, x) = S(\tau_{k-1}, t)p(x) \approx \sum_{l \in \mathcal{K}_M} c_l S(\tau_{k-1}, t)\phi_l(x). \tag{38}$$

At $t = \tau_k$, we only need to compute:

$$\tilde{u}_k(\tau_k, x) \approx \sum_{l \in \mathcal{K}_M} c_l \hat{\phi}_l(\tau_k, x), \tag{39}$$

where $\hat{\phi}_l(\tau_k, x) = S(\tau_{k-1}, \tau_k)\phi_l(x)$ solves (35) with initial condition $\hat{\phi}_l(\tau_{k-1}, x) = \phi_l(x)$. Thus, the problem reduces to solving M Kolmogorov equations, each with a basis function as the initial condition.

Neural networks provide a parameterized model to approximate any function. Specifically, a fully connected neural network with L layers can be expressed as a composition of activation function κ and affine functions A_i, $i = 1, \cdots, L$, which can be expressed as

$$\mathcal{F} = A_L \circ \kappa \circ A_{L-1} \circ \cdots \circ \kappa \circ A_1, \tag{40}$$

where

$$A_i(x) = W_i x + b_i, \ i = 1, \cdots, L, \tag{41}$$

and $\{(W_i, b_i) : 1 \leq i \leq L\}$ are constant matrices and vectors with proper sizes; the activation function κ can be chosen to be the ReLU function with $\kappa(x) = \max\{x, 0\}$ [23]. The set $\theta = \{(W_i, b_i) : 1 \leq i \leq L\}$ is the parameters in the neural network $\mathcal{F}$ to be learned or optimized. With parameters θ being carefully trained, the fully connected neural network can be used to approximate a very general class of functions with arbitrary precision, and we have the following universal approximation theorem for neural networks:

Theorem 3.4 (Universal approximation theorem for fully-connected neural network [26])**.** *For $L \geq 2$, the set of all L-Layer fully-connected neural network $\mathcal{F}$ is uniformly dense in the space of continuous functions in a compact*

set $D \subset \mathbb{R}^d$, i.e., given a continuous function $u \in C(D)$, for every $\epsilon > 0$, there exists a fully-connected neural network $\mathcal{F}$ such that

$$\sup_{x \in D} |\mathcal{F}(x) - u(x)| < \epsilon. \tag{42}$$

Solving the Kolmogorov equation requires embedding physical constraints directly into the model. Physics-Informed Neural Networks (PINNs) provide an elegant framework for this task by transforming the PDEs, boundary conditions, and initial conditions into distinct loss functions [43]. A fully-connected neural network serves as the parameterized model, generating solutions from sampled spatio-temporal points and optimizing via standard techniques to satisfy these physical constraints.

For each $l \in \mathcal{K}_M$, we approximate the solution $S(t, \tau_{k-1})\phi_l(x)$ using a PINN, denoted $u_{\theta_l}(t, x)$, where θ_l represents the network parameters. This neural network takes spatio-temporal coordinates $(t, x) \in D = [\tau_{k-1}, \tau_k] \times \Omega$ as input and produces the approximated solution:

$$u_{\theta_l}(t, x) = f_{\text{NN}}(t, x; \theta_l), \tag{43}$$

implemented as a fully connected architecture with smooth activation functions such as hyperbolic tangent to ensure the differentiability required for evaluating PDE residuals.

The training data $X_{\text{train}} = X_f \cup X_b \cup X_{\text{ic}}$ is synthetically generated through strategic sampling of three types of points:

- Interior points X_f sampled uniformly throughout the domain to capture the PDE's behavior
- Boundary points X_b sampled on the domain's boundary to enforce Dirichlet conditions
- Initial points X_{ic} sampled across Ω at $t = \tau_{k-1}$ to encode the initial state $\phi_l(x)$

This approach eliminates the need for pre-existing numerical solutions, as the network can evaluate residuals and conditions at arbitrary locations during training.

The loss function consists of three components, each addressing a specific physical constraint:

- **PDE Residual Loss $\mathcal{L}_{PDE}$**: This term enforces adherence to the Kolmogorov equation within the domain $(\tau_{k-1}, \tau_k] \times \Omega$. The residual is

defined as:

(44)
$$F(u_{\theta_l}) = \frac{\partial u_{\theta_l}}{\partial t} - \frac{1}{2}\sum_{i,j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j}\left(a^{ij} u_{\theta_l}\right) + \sum_{i=1}^{d} \frac{\partial}{\partial x_i}\left(f_i u_{\theta_l}\right) + \frac{1}{2}|h|^2 u_{\theta_l},$$

with the corresponding loss calculated as the mean squared residual over interior points X_f:

$$\mathcal{L}_{PDE} = \frac{1}{|X_f|} \sum_{(t_i, x_i) \in X_f} \left|F(u_{\theta_l}(t_i, x_i))\right|^2. \tag{45}$$

- **Initial Condition Loss** $\mathcal{L}_{IC}$: This term ensures the network satisfies the initial condition $u_{\theta_l}(\tau_{k-1}, x) = \phi_l(x)$ at $t = \tau_{k-1}$:

$$\mathcal{L}_{IC} = \frac{1}{|X_{\text{ic}}|} \sum_{x_i \in X_{\text{ic}}} \left|u_{\theta_l}(\tau_{k-1}, x_i) - \phi_l(x_i)\right|^2, \tag{46}$$

  where X_{ic} is the set of sampled initial points.
- **Boundary Condition Loss** $\mathcal{L}_{BC}$: This term enforces the homogeneous Dirichlet condition $u_{\theta_l}(t, x) = 0$ on the boundary $\partial\Omega$:

$$\mathcal{L}_{BC} = \frac{1}{|X_b|} \sum_{(t_i, x_i) \in X_b} \left|u_{\theta_l}(t_i, x_i)\right|^2, \tag{47}$$

  where X_b contains points on $[\tau_{k-1}, \tau_k] \times \partial\Omega$.

The total loss function combines these components with appropriate weights:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{PDE} + \beta \mathcal{L}_{IC} + \gamma \mathcal{L}_{BC}, \tag{48}$$

where α, β, and γ are positive hyperparameters that balance the contribution of each term. The network parameters θ_l are optimized using Stochastic Gradient Descent (SGD) or Adam optimizer, typically with a decaying learning rate schedule to ensure stable convergence. By minimizing $\mathcal{L}_{\text{total}}$, the PINN effectively learns a solution that satisfies the Kolmogorov equation along with its boundary and initial conditions.

Then, the (39) can be approximated by M neural network u_{θ_l} as

$$\tilde{u}_k(\tau_k, x) \approx \sum_{l \in \mathcal{K}_M} c_l u_{\theta_l}(t, x), \tag{49}$$

where $c_l = \langle p, \phi_l \rangle_{L^2(\Omega)}$ which only depends on basis function ϕ_l and initial density $p(x)$.

Unlike traditional spectral methods, which project solutions onto a fixed subspace of orthogonal basis functions and solve resulting ordinary differential equations, PINNs parameterize solutions directly, offering greater flexibility for non-linear dynamics and complex initial conditions without predefined bases. An efficient filtering algorithm consisting of a forward Kolmogorov equation solver based on a PINN and a probability density approximator based on generalized Legendre polynomials is proposed in [48].

4. Data-driven Yau-Yau algorithm

Recently, in the face of the urgent demands of solving high-dimensional and highly-nonlinear filtering problems from application areas, as well as the bottleneck of the model-driven implementations of Yau-Yau nonlinear filtering algorithm in dealing with high-dimensional problems, novel data-driven implementations of the Yau-Yau algorithm framework based on deep-learning have been proposed and studied in details [10]. Leveraging on the strong representation and approximation capability of neural networks, this new approach overwhelms classical filtering algorithms in many numerical experiments. In this section, we will start with the introduction of neural network and deep learning, and then summarize and highlight the theoretical and practical advantages of this data-driven Yau-Yau algorithm.

4.1. Neural network and the universal approximation theorem

From a mathematical perspective, the neural network is a set of hierarchically parameterized functions which can be used to approximate target functions in high-dimensional spaces. The deep learning process is to optimize the parameters in the neural network functions with stochastic gradient descent methods.

From the fully-connected neural networks, researchers have proposed and studied neural networks with various structures, in order to deal with different application scenarios. Among all the neural networks, we would like to briefly introduce the recurrent neural network (RNN) here because the recursive structure of RNN is suitable for filtering problems and has been applied in the data-driven Yau-Yau algorithm.

Definition 4.1. *Let $r_1, r_2, r_3 \in \mathbb{Z}_+$ be positive integers. A recurrent neural network RNN^{r_1,r_2,r_3} is the set of functions with the following open dynamical*

system form:

$$\begin{cases} s_{k+1} = \eta(s_k, \alpha_{k+1}, \theta_1); \\ \beta_k = \zeta(s_k, \theta_2); \end{cases} \quad k \geq 1. \tag{50}$$

where $\alpha_k \in \mathbb{R}^{r_1}$, $s_k \in \mathbb{R}^{r_2}$ *and* $\beta_k \in \mathbb{R}^{r_3}$ *for all* $k \geq 1$*; and* η, ζ *are fully-connected neural networks with parameter* θ_1 *and* θ_2*, respectively.*

Definition 4.1 consists of a quite general class of neural networks with recursive structure. Especially, the notable Long-Short Term Memory (LSTM) [25] and Gated Recurrent Units (GRU) [12] are included.

Recurrent neural network also inherits the universal approximation property of fully-connected neural networks, and can approximate a general class of dynamical systems with arbitrary precision. This result was first stated and proved in [45] for deterministic inputs, while Chen et al. [11] generalized the result to the stochastic version. In the next subsection, we will summarize the result that the implementation of the Yau-Yau nonlinear filtering algorithm can be expressed as an open dynamical system, which can then be approximated by an RNN with arbitrary precision.

4.2. The uniform framework of Yau-Yau algorithm

The idea of interpreting the Yau-Yau algorithm as an open dynamical system originates from the spectral method implementation (33), which is an ordinary differential equation of the linear-combination coefficients $\lambda(t)$. According to the continuous dependence of the solutions of linear ordinary differential equations on initial values, at each time interval $[\tau_{k-1}, \tau_k]$ in the framework of the Yau-Yau algorithm, the evolution of $\lambda(t)$ can be expressed as

$$\lambda(\tau_k) = \eta_1(\lambda(\tau_{k-1})), \tag{51}$$

where $\eta_1 : \mathbb{R}^N \to \mathbb{R}^N$ is a continuous function.

Assume that with properly chosen orthogonal basis $\{\phi_j\}_{j=1}^{\infty}$, $v(\tau_k, x) = \sum_{j=1}^{N} \lambda_j(\tau_k)\phi_j(x)$ is a numerical solution to the Kolmogorov equation (20), and therefore an approximation of the offline step solution $\tilde{u}_k(\tau_k, x)$. Then, according to (21), the online update procedure of the Yau-Yau algorithm can be written as:

$$\tilde{u}_{k+1}(\tau_k, x) \approx \exp\left(h^\top (Y_{\tau_k} - Y_{\tau_{k-1}})\right) v(\tau_k, x) \tag{52}$$

The computation of (52) can also be projected onto the N-dimensional subspace V_N spanned by $\{\phi_i\}_{i=1}^N$, and we obtain

(53)
$$\tilde{u}_{k+1}(\tau_k, x) \approx \sum_{j=1}^{N} \lambda_j(\tau_k) \langle \exp\left(h^\top (Y_{\tau_k} - Y_{\tau_{k-1}})\right), \phi_j \rangle \triangleq \eta_2(\lambda(\tau_k), Y_{\tau_k} - Y_{\tau_{k-1}}),$$

where $\eta_2 : \mathbb{R}^N \times \mathbb{R}^d \to \mathbb{R}^N$ is also a continuous function.

Therefore, if we use an element in V_N to approximate the function $\tilde{u}_{k+1}(\tau_k, \cdot)$ and express it as

$$\tilde{u}_{k+1}(\tau_k, x) \approx \sum_{j=1}^{N} \tilde{\lambda}_j(\tau_k) \phi_i(x), \tag{54}$$

for each $k \geq 0$, then the evolution of the coefficients $\tilde{\lambda}(\tau_k)$ is given by

$$\tilde{\lambda}(\tau_k) = \eta_2(\eta_1(\tilde{\lambda}(\tau_{k-1})), Y_{\tau_k} - Y_{\tau_{k-1}}) \triangleq \eta_3(\tilde{\lambda}(\tau_{k-1}), Y_{\tau_k} - Y_{\tau_{k-1}}), \tag{55}$$

where $\eta_3 : \mathbb{R}^N \times \mathbb{R}^d \to \mathbb{R}^N$ is a continuous function.

Finally, for a test function $\varphi : \mathbb{R}^d \to \mathbb{R}$, the conditional expectation $E[\varphi(X_{\tau_k})|\mathcal{Y}_{\tau_k}]$ can be computed by

(56)
$$E[\varphi(X_{\tau_k})|\mathcal{Y}_{\tau_k}] \approx \frac{\int_{\mathbb{R}^d} \varphi(x)\tilde{u}_{k+1}(\tau_k, x)dx}{\int_{\mathbb{R}^d} \tilde{u}_{k+1}(\tau_k, x)dx} \approx \frac{\sum_{j=1}^N \beta_{\varphi,j}\tilde{\lambda}_j(\tau_k)}{\sum_{j=1}^N \beta_{1,j}\tilde{\lambda}_j(\tau_k)} \triangleq \zeta_1(\tilde{\lambda}(\tau_k)),$$

where

$$\beta_{\varphi,j} = \int_{\mathbb{R}^d} \varphi(x)\phi_j(x)dx, \quad \beta_{1,j} = \int_{\mathbb{R}^d} \phi_j(x)dx, \tag{57}$$

and therefore, $\zeta_1 : \mathbb{R}^N \to \mathbb{R}$ is also a continuous function.

Let us combine (55) and (56), and we find that the entire implementation of Yau-Yau algorithm based on spectral method can be expressed by

$$\begin{cases} s_{k+1} = \eta_3(s_k, Y_{\tau_k} - Y_{\tau_{k-1}}) \\ \beta_k = \zeta_1(s_k) \end{cases}, \quad 1 \leq k \leq K, \tag{58}$$

which is an open dynamical system with respect to the coefficients $s_k \triangleq \tilde{\lambda}(\tau_k)$ with input given by the observation process $Y_{\tau_k} - Y_{\tau_{k-1}}$ and output β_k approximating the conditional expectation $E[\varphi(X_{\tau_k})|\mathcal{Y}_{\tau_k}]$.

The open dynamical system (58) can be approximated with arbitrary precision by a recurrent neural network (RNN), and therefore, can be learned and computed in a data-driven approach that we will introduce later. Remarkably, although the dynamical system (58) originates from the spectral method implementation of Yau-Yau algorithm, it is uniform for every different choice of orthogonal basis. Thus, (58) is called **the uniform framework of Yau-Yau nonlinear filtering algorithm**. In addition, because the orthogonal basis functions do not explicitly appear in the expression (58), the basis can be arbitrarily chosen such that the conditional probability density functions of the filtering problem can be most efficiently approximated and represented in theory, which means that the number of the basis functions (accounting for the dimension N of the state s_k and the magnitude of the RNN) required in the algorithm may be sharply decreased. In fact, we have the following theorem:

Theorem 4.1 ([10]). *Consider the time-invariant filtering system and approximated unnormalized conditional probability functions* $\{u_{k+1}(\tau_k, x)\}_{k=0}^{K}$ *defined in the Yau-Yau algorithm in a large ball* $B_R \subset \mathbb{R}^d$. *There exists a set of* N *normalized square-integrable functions,* $\{\phi_j\}_{j=1}^{N}$, *which are orthogonal to each other, such that for each* $k = 0, \cdots, K$, *we can find a function* $\tilde{v}_k(x)$ *in the* N*-dimensional vector space spanned by* $\{\phi_j\}_{j=1}^{N}$, *which satisfies*

$$\begin{aligned} &\tilde{E} \int_{B_R} |\tilde{v}_k(x) - u_{k+1}(\tau_k, x)| dx \leq C_T \sqrt{\delta}, \\ &\forall k = 0, 1, \cdots, K, \end{aligned} \tag{59}$$

where C_T *is a constant which depends on* T, R, *and the coefficients in the filtering system, but does not depend directly on the dimension of the filtering system,* d, *or the time discretization step* δ.

Next, if we represent $\tilde{v}_k$ *by*

$$\tilde{v}_k(x) = \sum_{j=1}^{N} s_{k+1,j} \phi_j(x), \tag{60}$$

then the evolution of $s_{k+1} = (s_{k+1,1}, \cdots, s_{k+1,N})^\top \in \mathbb{R}^N$ *satisfies an open dynamical system*

$$s_{k+1} = \eta_3(s_k, Y_{\tau_k} - Y_{\tau_{k-1}}), \quad k = 1, \cdots, K, \tag{61}$$

with a given initial value s_1, *where* $\eta_3 : \mathbb{R}^N \times \mathbb{R}^d \to \mathbb{R}^N$ *is a continuous function with respect to* $s_k \in \mathbb{R}^N$ *and* $Y_{\tau_k} - Y_{\tau_{k-1}} \in \mathbb{R}^d$, *and is time-invariant.*

Moreover, the number N of functions in the set $\{\phi_j\}_{j=1}^N$ can be chosen to grow at most polynomially with respect to the dimension d, which shows the capability of this framework of the Yau-Yau algorithm to overcome the curse of dimensionality.

4.3. The implementation of Yau-Yau algorithm based on deep learning

In order to practically implement the uniform framework of Yau-Yau nonlinear filtering algorithm, we need to compute or approximate the dynamical system (58). Although the functions η_3 and ζ_1 can be explicitly calculated if the orthogonal basis functions are given, it is very inefficient to introduce the orthogonal basis again and conduct the computations in this model-driven approach for high-dimensional problems. The reason mainly lies in the following two aspects. On the one hand, the computations of the inner products in the model-driven expression (33) involve many numerical integrals in high dimensions, which is very hard to accurately obtain. On the other hand, the explicit expression of the basis functions which satisfies the properties in Theorem 4.1 may be very complicated while the number of classical orthogonal basis functions constructed by tensor products is required to grow exponentially with respect to the dimension of the problem.

Therefore, in this subsection, we will introduce a data-driven approach to implement the computation of (58) directly, without involving the orthogonal basis functions. Specifically, we will train an RNN to approximate (58) with data generated from Monte-Carlo realizations of the filtering system.

Let us denote by

$$\begin{cases} s_{k+1} = \eta(s_k, \alpha_{k+1}, \theta_1); \\ \beta_k = \zeta(s_k, \theta_2); \end{cases} \quad 1 \le k \le K, \tag{62}$$

the RNN we used to approximate the dynamical system (58). Then, the parameters $\theta = [\theta_1, \theta_2]$ should be trained by minimizing the mean-square loss function

$$\hat{\theta} = \arg\min_{\theta=[\theta_1,\theta_2]} \sum_{k=1}^{K} E\left[\left|\beta_k - E[\varphi(X_{\tau_k})|\mathcal{Y}_{\tau_k}]\right|^2\right]. \tag{63}$$

However, the target conditional expectations $E[\varphi(X_{\tau_k})|\mathcal{Y}_{\tau_k}]$ are hard to obtain. Fortunately, according to the projection properties of conditional expec-

tations,

$$
\begin{aligned}
&E\left[|\beta_k - \varphi(X_{\tau_k})|^2\right] \\
&\quad = E\left[|\beta_k - E[\varphi(X_{\tau_k})|\mathcal{Y}_{\tau_k}]|^2\right] + E\left[|\varphi(X_{\tau_k}) - E[\varphi(X_{\tau_k})|\mathcal{Y}_{\tau_k}]|^2\right],
\end{aligned} \tag{64}
$$

for each $k = 1, \cdots, K$. Therefore, the parameters θ can be equivalently trained by optimizing,

$$
\hat{\theta} = \arg\min_{\theta=[\theta_1,\theta_2]} \sum_{k=1}^{K} E\left[|\beta_k - \varphi(X_{\tau_k})|^2\right], \tag{65}
$$

in which the targets $\varphi(X_{\tau_k})$ can be easily obtained through Monte-Carlo realizations.

Let $\{(X_{\tau_k}^{(i)}, Y_{\tau_k}^{(i)}) : 1 \le i \le M,\ 1 \le k \le K\}$ be M trajectories of the Monte-Carlo realizations of the filtering system (1), then the optimal parameters $\hat{\theta}$ can be approximated by

$$
\hat{\theta} \approx \arg\min \frac{1}{M} \sum_{k=1}^{K} \sum_{i=1}^{M} \left|\beta_k - \varphi(X_{\tau_k}^{(i)})\right|^2, \tag{66}
$$

which can be implemented by standard stochastic gradient descent method through the back-propagation process of deep learning.

Up to now, we have finished the procedure of conducting the uniform framework of Yau-Yau algorithm with a well-trained RNN and fulfilled the data-driven implementation of Yau-Yau algorithm.

5. Conclusion and future prospects

In this paper, we review the development of nonlinear filtering problem and provide a comprehensive introduction of the theory, implementation and related topics of the Yau-Yau nonlinear filtering algorithm from the perspective of model-driven to data-driven. The convergence results of the Yau-Yau algorithm itself, the spectral method of solving the corresponding Kolmogorov equation as well as the uniform framework of Yau-Yau algorithm are highlighted in this review. The idea and implementation details of Yau-Yau algorithm together with its model-driven and data-driven approach are also handled and sorted out in logical order, so that the readers with different backgrounds will get a better grasp of the Yau-Yau algorithm and can make an attempt to exploit it in practical application scenarios.

There are at least two promising and important future research directions in the study of Yau-Yau nonlinear filtering algorithms and related topics. Firstly, apart from the standard filtering system of the form as in (1), there are various non-standard filtering systems which are encountered by practitioners from different fields. One promising research direction is to generalize the Yau-Yau algorithm such that it can deal with these non-standard filtering systems, and then to establish the corresponding convergence theory and implementation methods for these problems.

Secondly, the relationship between the model-driven and data-driven implementation of Yau-Yau algorithm still needs to be further discussed. Although the target dynamical system in the data-driven implementation originates from the model-driven approach, the guidance of model-driven algorithm to data-driven implementation deserves further analysis. For example, how to engage the theory of model-driven Yau-Yau algorithm into the design of network training process in the data-driven implementation, or further improve the network structure, is an important research direction.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) grant (123B2020, 12201631, 42450242), Tsinghua University Education Foundation fund.

References

[1] Aidala, V. J. (1979). Kalman filter behavior in bearings-only tracking applications. *IEEE Transactions on Aerospace and Electronic Systems* **15** 29–39.

[2] Al-Jarrah, M., Hosseini, B. and Taghvaei, A. (2023). Optimal transport particle filters. In *2023 62nd IEEE Conference on Decision and Control (CDC)* 6798–6805. IEEE.

[3] Arasaratnam, I. and Haykin, S. (2009). Cubature kalman filters. *IEEE Transactions on automatic control* **54** 1254–1269. MR2532614

[4] Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on signal processing* **50** 174–188.

[5] Atar, R., Viens, F. and Zeitouni, O. (1999). *Robustness of Zakai's equation via Feynman-Kac representations*. Springer. MR1702969

[6] Bain, A. and Crisan, D. (2009). *Fundamentals of stochastic filtering. Stochastic Modelling and Applied Probability* **60**. Springer Science+Bussiness Media. MR2454694

[7] Baras, J., Blankenship, G. and Hopkins, W. (1983). Existence, uniqueness, and asymptotic behavior of solutions to a class of Zakai equations with unbounded coefficients. *IEEE Transactions on Automatic Control* **28** 203–214. MR0711678

[8] Bensoussan, A. (1988). On some approximation techniques in non linear filtering. In *Stochastic differential systems, stochastic control theory and applications* 17–31. Springer. MR0934714

[9] Cassola, F. and Burlando, M. (2012). Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output. *Applied energy* **99** 154–166.

[10] Chen, X., Sun, Z., Tao, Y. and Yau, S. S.-T. (2024). A Uniform Framework of Yau-Yau Algorithm Based on Deep Learning with the Capability of Overcoming the Curse of Dimensionality. *IEEE Transactions on Automatic Control*. MR4844909

[11] Chen, X., Tao, Y., Xu, W. and Yau, S. S.-T. (2022). Recurrent neural networks are universal approximators with stochastic inputs. *IEEE Transactions on Neural Networks and Learning Systems* **34** 7992–8006. MR4653628

[12] Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

[13] Claver, J. H., Dave, M. and Che, S. F. (2023). Kalman Filtering for Stocks Price Prediction and Control. *Journal of Computer Science* **19** 739–748.

[14] Crisan, D., Lobbe, A. and Ortiz-Latorre, S. (2022). An application of the splitting-up method for the computation of a neural network representation for the solution for the filtering equations. *Stochastics and Partial Differential Equations: Analysis and Computations* **10** 1050–1081. MR4491509

[15] Cui, Q., Ding, R., Zhou, B. and Wu, X. (2018). Path-tracking of an autonomous vehicle via model predictive control and nonlinear filtering. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* **232** 1237–1252.

[16] Davis, M. H. (1980). On a multiplicative functional transformation arising in nonlinear filtering theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **54** 125–139. MR0597335

[17] Dong, W., Luo, X. and Yau, S. S. T. (2021). Solving nonlinear filtering problems in real time by Legendre Galerkin spectral method. *IEEE Transactions on Automatic Control* **66** 1559–1572. MR4240187

[18] Duncan, T. E. (1967). Probability densities for diffusion processes with applications to nonlinear filtering theory and detection theory PhD thesis, Stanford University, Stanford, California.

[19] Evensen, G. (2003). The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics* **53** 343–367.

[20] Florchinger, P. and Gland, F. l. (1991). Time-discretization of the Zakai equation for diffusion processes observed in correlated noise. *Stochastics: An International Journal of Probability and Stochastic Processes* **35** 233–256. MR1113256

[21] Fu, Q., Lu, K. and Sun, C. (2023). Deep learning aided state estimation for guarded semi-Markov switching systems with soft constraints. *IEEE Transactions on Signal Processing* **71** 3100–3116. MR4649659

[22] Gobet, E., Pages, G., Pham, H. and Printems, J. (2006). Discretization and simulation of the Zakai equation. *SIAM Journal on Numerical Analysis* **44** 2505–2538. MR2272604

[23] Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep learning*. MIT press. MR3617773

[24] Hazewinkel, M., Marcus, S. I. and Sussmann, H. J. (1983). Nonexistence of finite-dimensional filters for conditional statistics of the cubic sensor problem. *Systems & Control Letters* **3** 331–340. MR0729114

[25] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation* **9** 1735–1780.

[26] Hornik, K., Stinchcombe, M. and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks* **2** 359–366.

[27] Jazwinski, A. H. (1970). *Stochastic processes and filtering theory. Mathematics in Science and Engineering* **64**. Elsevier. MR4945312

[28] Jiao, L., Wang, D., Bai, Y., Chen, P. and Liu, F. (2023). Deep learning in visual tracking: a review. *IEEE Transactions on Neural Networks and Learning Systems* **34** 5497–5516. MR4690732

[29] Kallianpur, G. and Striebel, C. (1968). Estimation of stochastic systems: Arbitrary system process with additive white noise observation errors. *The Annals of Mathematical Statistics* **39** 785–801. MR0224242

[30] Kallianpur, G. and Striebel, C. (1969). Stochastic differential equations occurring in the estimation of continuous parameter stochastic processes. *Theory of Probability & Its Applications* **14** 567–594. MR0264780

[31] Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME, D* **82** 35–44. MR3931993

[32] Kalman, R. and Bucy, R. (1961). New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering* **83** 95–108. MR0234760

[33] Kang, J., Chen, X., Tao, Y. and Yau, S. S.-T. (2022). Optimal transportation particle filter for linear filtering systems with correlated noises. *IEEE Transactions on Aerospace and Electronic Systems* **58** 5190–5203.

[34] Kang, J., Salmon, A. and Yau, S. S. T. (2023). Log-concave posterior densities arising in continuous filtering and a maximum a posteriori algorithm. *SIAM Journal on Control and Optimization* **64** 2407–2424. MR4623279

[35] Karatzas, I. (1998). *Brownian motion and stochastic calculus*, 2nd ed. New York: Springer.

[36] Li, S., Wang, Z., Yau, S. S. T. and Zhang, Z. (2023). Solving nonlinear filtering problems using a tensor train decomposition method. *IEEE Transactions on Automatic Control* **68** 4405–4412. MR4614325

[37] Liang, Z. and Yau, S. S.-T. (2004). Wavelet-Galerkin method for the Kolmogorov equation. *Mathematical and computer modelling* **40** 1093–1121. MR2113839

[38] Luo, X., Pan, T. and Dong, W. (2023). Convergence Analysis of Splitting-Up Algorithm of the Zakai's Equation with Correlated Noises. *Journal of Systems Science & Complexity* **36** 922–946. MR4591507

[39] Luo, X. and Yau, S. S.-T. (2013). Complete real time solution of the general nonlinear filtering problem without memory. *IEEE Transactions on Automatic Control* **58** 2563–2578. MR3106062

[40] Luo, X. and Yau, S. S.-T. (2013). Hermite spectral method with hyperbolic cross approximations to high-dimensional parabolic PDEs. *SIAM Journal on Numerical Analysis* **51** 3186–3212. MR3131484

[41] Luo, X. and Yau, S. S. T. (2013). Hermite spectral method to 1-d forward Kolmogorov equation and its application to nonlinear filtering problems. *IEEE Transactions on Automatic Control* **58** 2495–2507. MR3106057

[42] Mortensen, R. E. (1966). Optimal control of continuous time stochastic systems PhD thesis, University of California, Berkley, California.

[43] Raissi, M., Perdikaris, P. and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **378** 686–707. MR3881695

[44] Ribeiro, M. I. (2004). Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics* **43** 3736–3741.

[45] Schäfer, A. M. and Zimmermann, H. G. (2007). Recurrent neural networks are universal approximators. *International Journal of Neural Systems* **17** 253–263.

[46] Schmidt, S. F. (1981). The Kalman filter-Its recognition and development for aerospace applications. *Journal of Guidance and Control* **4** 4–7. MR2659223

[47] Shi, J., Jiao, X. and Yau, S. S.-T. (2024). A Novel Logarithmic Transformed Deep Galerkin Approach To Optimal Filtering Problem. In *2024 IEEE 63rd Conference on Decision and Control (CDC)* 2721–2726. IEEE.

[48] Shi, J., Jiao, X. and Yau, S. S. T. (2025). DGLG: A Novel Deep Generalized Legendre–Galerkin Approach to Optimal Filtering Problem. *IEEE Transactions on Automatic Control* **70** 2584–2590. MR4884980

[49] Taghvaei, A. and Mehta, P. G. (2016). An optimal transport formulation of the linear feedback particle filter. In *2016 American Control Conference (ACC)* 3614–3619. IEEE.

[50] Taghvaei, A. and Mehta, P. G. (2023). A survey of feedback particle filter and related controlled interacting particle systems (CIPS). *Annual Reviews in Control* **55** 356–378. MR4594829

[51] Tao, Y., Kang, J. and Yau, S. S. T. (2023). Neural projection filter: learning unknown dynamics driven by noisy observations. *IEEE Transactions on Neural Networks and Learning Systems*. MR4771552

[52] Vermaak, J., Godsill, S. J. and Perez, P. (2005). Monte Carlo filtering for multi target tracking and data association. *IEEE Transactions on Aerospace and Electronic systems* **41** 309–332.

[53] Wan, E. A. and Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)* 153–158. Ieee.

[54] Wang, K., Chen, J., Song, Z., Wang, Y. and Yang, C. (2022). Deep neural network-embedded stochastic nonlinear state-space models and their applications to process monitoring. *IEEE Transactions on Neural Networks and Learning Systems* **33** 7682–7694. MR4516648

[55] Wang, L. Y., Yin, G. and Zhang, Q. (2021). Deep filtering. *Communications in Information and Systems* **21** 651–667. MR4273517

[56] Wang, Z., Luo, X., Yau, S. S. T. and Zhang, Z. (2020). Proper orthogonal decomposition method to nonlinear filtering problems in medium-high dimension. *IEEE Transactions on Automatic Control* **65** 1613–1624. MR4085532

[57] Wiener, N. (2019). *Cybernetics or Control and Communication in the Animal and the Machine*. MIT press. MR0127995

[58] Yang, T., Laugesen, R. S., Mehta, P. G. and Meyn, S. P. (2016). Multivariable feedback particle filter. *Automatica* **71** 10–23. MR3521947

[59] Yang, T., Mehta, P. G. and Meyn, S. P. (2013). Feedback particle filter. *IEEE transactions on Automatic control* **58** 2465–2480. MR3106055

[60] Yau, S. S. T., Chen, X., Jiao, X., Kang, J., Sun, Z. and Tao, Y. (2024). *Principles of Nonlinear Filtering Theory*, 1 ed. *Algorithms and Computation in Mathematics*. Springer, Cham. MR4890473

[61] Yau, S.-T. and Yau, S. S.-T. (2000). Real time solution of nonlinear filtering problem without memory I. *Mathematical Research Letters* **7** 671–693. MR1809293

[62] Yau, S. T. and Yau, S. S. T. (2008). Real time solution of the nonlinear filtering problem without memory II. *SIAM Journal on Control and Optimization* **47** 163–195. MR2373467

[63] Yin, X., Zhang, Q., Wang, H. and Ding, Z. (2020). RBFNN-based minimum entropy filtering for a class of stochastic nonlinear systems. *IEEE Transactions on Automatic Control* **65** 376–381. MR4052887

[64] Yueh, M.-H., Lin, W.-W. and Yau, S.-T. (2014). An efficient numerical method for solving high-dimensional nonlinear filtering problems. *Communications in Information and Systems* **14** 243–262. MR3340942

[65] Zakai, M. (1969). On the optimal filtering of diffusion process. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* **11** 230–243.

Zeju Sun
Beijing Institute of Mathematical Sciences and Applications (BIMSA)
P. R. China
E-mail address: sunzeju@bimsa.cn

Jiayi Kang
Hetao Institute of Mathematics and Interdisciplinary Sciences (HIMIS)
P. R. China
Beijing Institute of Mathematical Sciences and Applications (BIMSA)
P. R. China
E-mail address: kangjiayi@bimsa.cn

Xiuqiong Chen
School of Mathematics, Renmin University of China
P. R. China
E-mail address: cxq0828@ruc.edu.cn

Stephen S.-T. Yau
Department of Mathematical sciences
Tsinghua University
P. R. China
E-mail address: yau@uic.edu

Received August 12, 2025