Recurrent Neural Network Spectral Method and its Application in Stable Filtering Problems

Zeju Sun [a,1], Xiuqiong Chen [b,1,*], Stephen S.-T. Yau [c,a,*]

[a] *Department of Mathematical Sciences, Tsinghua University, Beijing, 100084, P. R. China*

[b] *School of Mathematics, Renmin University of China, Beijing, 100872, P. R. China*

[c] *Beijing Institute of Mathematical Sciences and Applications (BIMSA), Beijing, 101408, P. R. China*

Abstract

Solving high-dimensional filtering problems with high nonlinearity is essential for controlling complicated systems and for data assimilation. In this paper, we propose a novel recurrent neural network spectral method (RNNSM) to address this kind of problems, especially for the systems with additional stability properties. As a combination of modern deep learning strategy and classical spectral method, the proposed algorithm integrates the advantages of both. On the one hand, by exploiting the approximation capability of recurrent neural networks, RNNSM can overcome the obstacles that classical spectral methods face in high-dimensional problems, and obtain a heuristic approach to finding the optimal orthonormal basis in spectral methods; on the other hand, with the theoretical foundation of spectral methods, RNNSM provides a more reasonable mathematical interpretation of neural network-based filtering algorithms and bridges the gap between practical performance and theoretical convergence. Finally, the efficiency of RNNSM is also verified by numerical experiments.

Key words: Nonlinear observer and filter design, Artificial neural network, Recurrent neural network (RNN), Spectral method, Filtering stability

1 Introduction

Filtering is a subject on estimating the evolving state process in a partially observed system. With the development of science and technology, people encounter more and more complicated partially observed dynamical systems with high dimension and high nonlinearity. The development of efficient filtering algorithms for such systems has become a core issue in the field of control and signal processing.

For this kind of filtering systems, classical algorithms such as particle filter (PF) and Kalman filter based algorithms face significant challenges. As for Kalman filter based algorithms, the accuracy and convergence of these algorithms are doubtful for systems with high degree of nonlinearity [5,34]; as for PFs, one of the main difficulties is the severe particle degeneracy for high-dimensional systems [1, 21]. A great many particles are required for PFs to handle high-dimensional systems, which makes it hard to obtain an estimation to state processes in real time.

Nowadays, with the introduction of deep learning to control community, there have been a large number of neural network based filtering algorithms proposed in recent years [18, 23], which can outperform classical filtering algorithms in numerical simulations [32]. The connection between recurrent neural network (RNN) and finite dimensional filtering systems including Gaussian systems was established for the first time in [7] and the estimation error was rigorously analyzed. However, the mathematical theory of neural network based filtering algorithms for general nonlinear problems has not yet been sufficiently developed, in the sense that the performance

[*] Corresponding authors

Email addresses: szj20@mails.tsinghua.edu.cn (Zeju Sun), cxq0828@ruc.edu.cn (Xiuqiong Chen), yau@uic.edu (Stephen S.-T. Yau).

[1] These authors contributed equally to this work.

[2] This work is supported by National Natural Science Foundation of China (NSFC) under Grant 123B2020 for Zeju Sun, by NSFC under Grant 12201631 for Xiuqiong Chen, and by Tsinghua University Education Foundation fund under Grant 042202008 and Start-up Fund of Beijing Institute of Mathematical Sciences and Applications (BIMSA) for Stephen S.-T. Yau.

of neural network based algorithms has not been sufficiently interpreted through mathematical theory.

In fact, most of the mathematical theory of these algorithms is based on the universal approximation property of neural networks [16, 30], in which it is proved that any continuous function in a compact domain can be approximated with arbitrary accuracy by a shallow neural network. Meanwhile, the conditional probability density functions (pdf, for short), which we are concerned with in nonlinear filtering problems, are in general located in an infinite dimensional functional space [14, 26], and an infinite-width neural network is required to follow the evolution of the conditional pdfs according to the universal approximation theory.

For practical implementation of nonlinear filtering algorithms in finite dimensional spaces, and idea is to first project the conditional pdfs onto a finite dimensional manifold or subspace and then track the evolution of the projection of these functions [2, 6]. The choice of finite dimensional subspaces can be roughly divided into two categories: (i) manifolds of classical density families such as Gaussian and exponential distribution families; and (ii) subspaces spanned by orthonormal basis functions.

The first choice corresponds to filtering algorithms such as the extended Kalman filter (EKF) [17] and assumed density filters [19]. For this kind of filtering algorithms, important statistics of the approximated density functions, such as expectations, covariance matrices, etc., are easy to calculate, but the distance between the exact conditional distribution and its projection is difficult to estimate.

The second choice corresponds to filtering algorithms based on spectral methods [20,24,31,36]. Using the properties of orthonormal basis functions, a rigorous convergence analysis of these filtering algorithms can be provided. For high-dimensional filtering systems, however, the orthonormal basis in the algorithm is usually not so efficient, and how to find the optimal orthonormal basis functions such that the corresponding filtering algorithm enjoys a high convergence rate is still an open problem.

In this paper, we provide a heuristic approach to finding the optimal orthonormal basis for a given filtering system with additional stability property which is possessed by many practical filtering systems. Roughly speaking, the stability property states that the effect of the initial distribution on the filtering solution decays exponentially as time tends to infinity [3, 4]. Filter stability implies that the conditional probability density as a function of observations does not contain long-term memory, i.e., remote observations are quickly 'forgotten' in the expression of conditional probability distribution [27]. As an important branch of filtering theory, there are many studies on the stability of filtering systems, and readers can refer to the monograph [10] for detailed discussions. For stable filtering systems, we will show that our heuristic approach to finding the optimal orthonormal basis can be implemented using a recurrent neural network (RNN). In this way, we propose a new efficient nonlinear filtering algorithm, called Recurrent Neural Network Spectral Method (RNNSM).

The advantages of our proposed RNNSM over the existing filtering algorithms lie mainly in the following two aspects:

(1) With the help of neural network, RNNSM is much more computationally efficient than the existing filtering algorithms based on spectral methods, in the sense that the problem of finding an efficient orthonormal basis and the heavy computational burden of numerical integrals can be handled off-line by training the recurrent neural networks.
(2) Using the theory of spectral methods, we can provide a thorough convergence analysis for RNNSM. To the best of the authors' knowledge, this is the first rigorous convergence analysis of RNN-based nonlinear filtering algorithms for the *most general* stable nonlinear filtering systems (without assuming that the conditional probability distribution evolves in a finite dimensional subspace or manifold), on the whole timeline (without assuming a fixed terminal time $T < \infty$).

Moreover, the proposed RNNSM together with its convergence analysis serves as a novel mathematical interpretation of RNN and RNN-based filtering algorithms. It is validated both theoretically and numerically that RNNSM can provide a good approximation to the exact solution of these stable filtering systems in the whole timeline, even if the training data are generated only in a finite time interval. In this way, this paper also throws lights on the interpretability of RNN, in terms of its learning and generalization capability to handle temporal data.

The organization of this paper is summarized as follows. In Section 2, we will review some basic notations, concepts and results for nonlinear filtering problems. In Section 3, the algorithm, RNNSM, will be proposed to solve stable filtering systems. In Section 4, convergence results of RNNSM will be presented, together with the main ideas of the proofs. The details of the proofs can be found in Section 6, after the numerical results of RNNSM that are illustrated in Section 5. Section 7 is a conclusion.

Throughout this paper, some frequently used notations are summarized in Table 1 for readers' convenience. Also, we use the notations $\mathcal{L}^1(\mathbb{R}^d)$, $\mathcal{L}^1_+(\mathbb{R}^d)$, $\mathcal{L}^2(\mathbb{R}^d)$ and $\mathcal{L}^\infty(\mathbb{R}^d)$ to denote the space of integrable, positive integrable, square-integrable, and essentially bounded functions in $\mathbb{R}^d$, respectively. The norms on the above functional spaces $\mathcal{L}^p(\mathbb{R}^d)$ are denoted by $\|\cdot\|_p$, for

$p = 1, 2, \infty$, and we use $|\cdot|$ to denote the Euclidean norm (or modulus) of finite-dimensional vectors as usual.

Table 1
List of Notations

Meaning	Notations	Equations
State	x_k	(1)
Observation	y_k	(1)
Conditional distribution (or conditional pdf) of x_k given $\mathcal{Y}_k$	π_k	(2)
Unnormalized conditional pdf	u_k	(5)
Unnormalized conditional pdf with likelihood approximation	$\tilde{u}_k$	(15)
Finite-dimensional representation of unnormalized conditional pdf $\tilde{u}_k$	v_k	(18)
Finite-dimensional representation of normalized conditional pdf	ρ_k	(21)
Conditional pdf with different initial value	$\tilde{\pi}_k$	(46)
Propagation operator	$\mathcal{F}_1$	(4)
Correction operator	$\mathcal{F}_2$	(4)
Approximated correction operator	$\tilde{\mathcal{F}}_2$	(13)
Integral operator	$\mathcal{F}_3$	(10)
Normalization operator	$\mathcal{N}$	(7)

2 Preliminaries

In this paper, we consider the time-invariant filtering system in discrete time modeled by the following coupled stochastic difference equations:

$$\begin{cases} x_{k+1} = \varphi(x_k) + \Gamma(x_k)v_k \\ y_{k+1} = h(x_{k+1}) + w_{k+1} \end{cases} \quad 1 \leq k < \infty, \tag{1}$$

where $\{x_k : 1 \leq k < \infty\} \subset \mathbb{R}^d$ is the state process, with initial value $x_1 \sim \pi_1$; $\{y_k : 1 \leq k < \infty\} \subset \mathbb{R}^{d_1}$ is the observation process, with initial value $y_1 \equiv 0$; $\{w_k : 2 \leq k < \infty\} \subset \mathbb{R}^{d_1}$, $\{v_k : 1 \leq k < \infty\} \subset \mathbb{R}^{d_2}$ are mutually independent Gaussian random variables with zero mean and covariance matrices $R \in \mathbb{R}^{d_1 \times d_1}$, $Q \in \mathbb{R}^{d_2 \times d_2}$, respectively; $\varphi : \mathbb{R}^d \to \mathbb{R}^d$, $h : \mathbb{R}^d \to \mathbb{R}^{d_1}$ and $\Gamma : \mathbb{R}^d \to \mathbb{R}^{d \times d_2}$ are smooth vector- or matrix-valued functions. The system (1) is called time-invariant, because the functions φ, h, Γ and the matrices Q, R do not explicitly depend on the time step parameter k.

The main goal of filtering is to recursively calculate the conditional distribution of state process x_k given the history of noisy observations $\mathcal{Y}_k \triangleq \sigma\{y_j : 1 \leq j \leq k\}$, which means the σ-algebra generated by the random variables $\{y_j : 1 \leq j \leq k\}$, for each time step k, and we denote this conditional distribution by

$$\pi_k(dx) \triangleq P[x_k \in dx|\mathcal{Y}_k], \quad 1 \leq k < \infty. \tag{2}$$

If the conditional distribution π_k ($1 \leq k < \infty$) is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$, i.e., $\pi_k(dx) = \pi_k(x)dx$, according to the theory of discrete filters [17], the density function of π_k (which is also denoted by π_k when there is no confusion), evolves with the following two-step formula:

$$\begin{aligned} &\textbf{Prediction: } \pi_{k+1|k}(x_{k+1}) = \int_{\mathbb{R}^d} \pi_k(x_k)p(x_{k+1}|x_k)dx_k; \\ &\textbf{Correction: } \pi_{k+1}(x_{k+1}) \propto \pi_{k+1|k}(x_{k+1})q(y_{k+1}|x_{k+1}), \end{aligned} \tag{3}$$

with $p(x_{k+1}|x_k)$ the transition probability density of the state process in (1) and

$$q(y_{k+1}|x_{k+1}) = \exp\Big[-\frac{1}{2}(y_{k+1} - h(x_{k+1}))^\top \times R^{-1}(y_{k+1} - h(x_{k+1}))\Big],$$

where $\pi_{k+1|k}$ are often referred to as the distribution of x_{k+1}, i.e., $\pi_{k+1|k}(\cdot) \triangleq P[x_{k+1} \in \cdot|\mathcal{Y}_k]$; the symbol '$\propto$' means that the density functions $\pi_{k+1|k}$ and π_{k+1} are proportional to the right-hand expressions, that is, they are equal except for a normalization constant independent of x.

Let us define the functionals $\mathcal{F}_1 : \mathcal{L}^2(\mathbb{R}^d) \to \mathcal{L}^2(\mathbb{R}^d)$, $\mathcal{F}_2 : \mathcal{L}^2(\mathbb{R}^d) \times \mathbb{R}^{d_1} \to \mathcal{L}^2(\mathbb{R}^d)$ with expressions:

$$\mathcal{F}_1(u)(x) \triangleq \int_{\mathbb{R}^d} p(x|z)u(z)dz; \ \mathcal{F}_2(u, y)(x) \triangleq q(y|x)u(x), \tag{4}$$

for all $u \in \mathcal{L}^2(\mathbb{R}^d)$, $y \in \mathbb{R}^{d_1}$, $x \in \mathbb{R}^d$. Then, the two-step formula (3) can be rewritten as follows:

$$\begin{aligned} &\textbf{Prediction: } \pi_{k+1|k} = \mathcal{F}_1(\pi_k); \\ &\textbf{Correction: } \pi_{k+1} \propto \mathcal{F}_2(\pi_{k+1|k}, y_{k+1}), \end{aligned} \tag{5}$$

If we ignore the normalization constant, we can define a series of unnormalized conditional pdfs, $\{u_k : k \geq 1\}$, by

$$u_{k+1} = \mathcal{F}_2(\mathcal{F}_1(u_k), y_{k+1}), \ k \geq 1, \tag{6}$$

with initial value $u_1 = \pi_1$, the initial distribution of state process. At each time step k, the conditional pdf π_k is then obtained through a normalization procedure:

$$\pi_k = \mathcal{N}(u_k) \triangleq \frac{u_k}{\int_{\mathbb{R}^d} u_k dx}, \quad k \geq 1, \tag{7}$$

where $\mathcal{N} : \mathcal{L}^1_+(\mathbb{R}^d) \to \mathcal{L}^1_+(\mathbb{R}^d)$ is defined as the normalization operator, with $\mathcal{L}^1_+(\mathbb{R}^d) \triangleq \{u \in \mathcal{L}^1(\mathbb{R}^d) : u \geq 0,\ a.e.\}$ the set of positive integrable functions on $\mathbb{R}^d$. The evolution of π_k is then given by

$$\pi_{k+1} = \mathcal{N} \circ \mathcal{F}_2(\mathcal{F}_1(\pi_k), y_{k+1}),\ k \geq 1. \tag{8}$$

In practice, people are often more concerned with some specific statistics of the conditional distribution π_k, such as the conditional mean and covariance matrix. In the meanwhile, the conditional distribution π_k can also be reconstructed or approximated based on its statistics with statistical methods such as moment matching [28].

With the evolution of π_k, these statistics can be calculated by

$$\hat{f}_k \triangleq \mathbb{E}[f(x_k)|\mathcal{Y}_k] = \mathcal{F}_3^f(\pi_k), \quad 1 \leq k < \infty, \tag{9}$$

where $\mathcal{F}_3^f : \mathcal{L}^2(\mathbb{R}^d) \cap \mathcal{L}^1_+(\mathbb{R}^d) \to \mathbb{R}^l$ is the integral operator defined as

$$\mathcal{F}_3^f(u) \triangleq \int_{\mathbb{R}^d} f(x)u(x)dx, \quad \forall\ u \in \mathcal{L}^2(\mathbb{R}^d) \cap \mathcal{L}^1_+(\mathbb{R}^d), \tag{10}$$

and $f : \mathbb{R}^d \to \mathbb{R}^l$ corresponds to the conditional statistics. When there is no ambiguity, we would like to omit the superscript in $\mathcal{F}_3^f$ to maintain the simplicity of notations. Without loss of generality, we focus on the real-valued f with $l = 1$ in this paper, and the results can be trivially extended to vector cases.

According to (8) and (9), the whole procedure of calculating the conditional statistics can be illustrated by Figure 1. As shown in Figure 1, the conditional pdfs π_k evolve according to an open dynamical system defined on $\mathcal{L}^2(\mathbb{R}^d) \cap \mathcal{L}^1_+(\mathbb{R}^d)$, which itself has a recursive structure in the infinite dimensional vector space. For practical implementation of this procedure, however, we need to find a finite-dimensional approximation to this infinite-dimensional open dynamical system. To this end, we propose the so-called Recurrent Neural Network Spectral Method (RNNSM) in this paper.

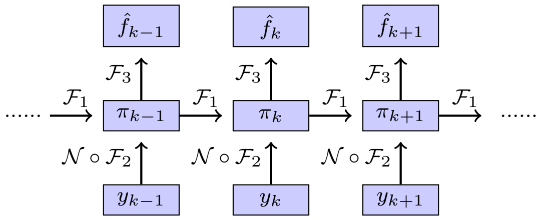


Fig. 1. Illustration of the calculation procedure of the conditional statistics $\hat{f}_k$.

3 Recurrent Neural Network Spectral Method

The recurrent neural network spectral method (RNNSM) proposed in this section gives a finite-dimensional approximation to the π_k shown in Figure 1, so that it can be performed in practice using a finite-width recurrent neural network.

This finite-dimensional approximation is done according to the following two main steps. First, the 'likelihood functions', $\exp\left[-\frac{1}{2}(y - h(x))^\top R^{-1}(y - h(x))\right]$, in the operator $\mathcal{F}_2$ are approximated by a linear combination of finitely many functions with respect to x, where the linear coefficients are functions of y. In this way, the variables x and y are separated. Next, the conditional pdfs π_k are represented (or approximated) by an element in a constructed finite-dimensional vector space, which fulfills the finite-dimensional approximation of the infinite-dimensional system shown in Figure 1. These two steps will be specifically described in the next two subsections, respectively.

3.1 Approximating the Likelihood Functions

Let us consider the functional $\mathcal{F}_2$ in Figure 1, where the unnormalized density functions are multiplied by a kind of 'likelihood functions', $q(y_k|x) = \exp[-\frac{1}{2}(y_k - h(x))^\top R^{-1}(y_k - h(x))]$. The first step of our proposed RNNSM is to project these 'likelihood function' onto a finite-dimensional subspace of $\mathcal{L}^2(\mathbb{R}^d)$.

For a given large real number $L > 0$, let us denote by $S_L = \{q(y|x) : y \in \mathscr{B}_L\}$ the set of all 'likelihood functions' with *regular* observations, where $\mathscr{B}_L = \{y \in \mathbb{R}^{d_1} : |y| \leq L\}$ is the ball with radius L. Since $\mathscr{B}_L$ is a compact subset of $\mathbb{R}^{d_1}$, according to the theory of Kolmogorov n-width [29], for any $\epsilon > 0$, there exists $N \in \mathbb{N}$ and an N-dimensional subspace $\mathcal{H}_N \subset \mathcal{L}^\infty(\mathbb{R}^d)$, such that

$$\max_{y \in \mathscr{B}_L} \left\| q(y|\cdot) - \sum_{i=1}^N \alpha_i(y)\phi_i(\cdot) \right\|_\infty < \epsilon, \tag{11}$$

where $\{\phi_i\}_{i=1}^N$ is an orthonormal basis of $\mathcal{H}_N$ and

$$\alpha_i(y) \triangleq \int_{\mathbb{R}^d} q(y|x)\phi_i(x)dx, \tag{12}$$

are the generalized Fourier coefficients, with $i = 1, \cdots, N$. Alternatives of the orthonormal basis $\{\phi_i\}_{i=1}^N$ include classical bases, such as Hermite functions [24], and basis functions derived from polynomial interpolations [25].

With this finite dimensional approximation to the likelihoods, we can define the functional $\tilde{\mathcal{F}}_2 : \mathcal{L}^2(\mathbb{R}^d) \times \mathbb{R}^{d_1} \to$

$\mathcal{L}^2(\mathbb{R}^d)$ by

$$\tilde{\mathcal{F}}_2(u,y)(x) \triangleq \left(\sum_{i=1}^{N} \alpha_i(y)\phi_i(x)\right) u(x), \qquad (13)$$

and we have

$$\|\mathcal{F}_2(u,y) - \tilde{\mathcal{F}}_2(u,y)\|_2 \leq \epsilon \|u\|_2 \qquad (14)$$

for all $u \in \mathcal{L}^2(\mathbb{R}^d)$ and $y \in \mathscr{B}_L$.

The approximation procedure (11) separates the space variable x in the likelihood functions from the observation variable y, which is essential for the finite-dimensional implementation of the computation graph shown in Figure 1, and we denote the approximated unnormalized density function by $\{\tilde{u}_k : k \geq 1\}$, which propagates according to

$$\tilde{u}_{k+1}(x) = \tilde{\mathcal{F}}_2\left(\mathcal{F}_1(\tilde{u}_k), y_{k+1}\right)(x), \ 1 \leq k < \infty. \qquad (15)$$

with initial value $\tilde{u}_1(x) = u_1(x) \equiv \pi_1(x), \ \forall \ x \in \mathbb{R}^d$.

3.2 Representing the Unnormalized Conditional Probability Density Functions

Before moving on to the procedure of RNNSM, let us first introduce the stability assumption of the filtering system.
Assumption 1. *The discrete filtering system* (1) *is stable, in the sense that*

$$\lim_{k\to\infty} \mathbb{E}\|\pi_k^{\mu} - \pi_k^{\nu}\|_2 = 0, \qquad (16)$$

where π_k^{μ} *and* π_k^{ν} *denote the conditional pdf, with initial distributions* $\mu, \nu \in \mathcal{L}^2(\mathbb{R}^d) \cap \mathcal{L}_+^1(\mathbb{R}^d)$*, respectively.*

Assumption 1 is a typical result in the studies of filtering stability such as [3, 35], which indicates that the initial value of the filtering system is 'forgotten' as the time step $k \to \infty$. This assumption also implies that the conditional distribution will not oscillate much when the time step $k \to \infty$, and will lie close to the finite-dimensional vector space spanned by the conditional density functions at the first few time steps. This will be rigorously proved later in Section 4.

Therefore, the finite-dimensional space on which we would like to project the conditional pdfs and practically implement the calculation in Figure 1 can be constructed with the information of the first few time steps.

For a given $K > 0$, because $\mathcal{F}_1$ and $\tilde{\mathcal{F}}_2(\cdot, y)$ are linear functionals in $\mathcal{L}^2(\mathbb{R}^d) \cap \mathcal{L}_+^1(\mathbb{R}^d)$, the functions $\{\tilde{u}_k : 1 \leq k \leq K\}$ in fact evolve in a finite-dimensional vector space spanned by $\mathcal{I}_{K,N} = \left\{ \left(\prod_{j=1}^{k-1}(\phi_{i_j}\mathcal{F}_1)\right) u_1 : 1 \leq k \leq K, \ 1 \leq i_1, \cdots, i_k \leq N \right\}$, where the products $\prod_{j=1}^{k-1}$ mean the composition of functionals (with the convenience of representing identical operator if $k = 1$) and with a slight abuse of notations, we use ϕ_i to denote the functional of multiplying $\phi_i(x)$.

For example, according to (13) and (15), in the first several steps,

$$\begin{aligned} &\tilde{u}_1(x) = u_1(x), \ \tilde{u}_2(x) = \sum_{i=1}^{N} \alpha_i(y_2)(\phi_i\mathcal{F}_1)u_1, \\ &\tilde{u}_3(x) = \sum_{i,j=1}^{N} \alpha_i(y_3)\alpha_j(y_2)\Big[(\phi_i\mathcal{F}_1)(\phi_j\mathcal{F}_1)\Big]u_1, \\ &\tilde{u}_4(x) = \sum_{i,j,k=1}^{N} \alpha_i(y_4)\alpha_j(y_3)\alpha_k(y_2) \\ &\qquad\qquad \times \Big[(\phi_i\mathcal{F}_1)(\phi_j\mathcal{F}_1)(\phi_k\mathcal{F}_1)\Big]u_1, \end{aligned}$$

which are all elements of the subspace spanned by functions of the form $\left(\prod_{j=1}^{k-1}(\phi_{i_j}\mathcal{F}_1)\right) u_1$, with $k = 2, 3, 4$. In this way, all the functions $\{\tilde{u}_k : 1 \leq k \leq K\}$ are contained in the subspace spanned by $\mathcal{I}_{K,N}$. Hereafter, we will use V_{N_1} to denote the N_1-dimensional vector space spanned by $\mathcal{I}_{K,N}$, i.e., $V_{N_1} = \text{Span}\{\mathcal{I}_{K,N}\}$, and let $\{\psi_i\}_{i=1}^{N_1}$ be an orthonormal basis of V_{N_1}.

The main procedure of RNNSM is to find elements in V_{N_1}, denoted by $\{v_k : k \geq 1\}$ and $\{\rho_k : k \geq 1\}$, which approximate $\{u_k : k \geq 1\}$ (or $\{\tilde{u}_k : k \geq 1\}$) and $\{\pi_k : k \geq 1\}$ well, respectively, for sufficiently large $N_1 \in \mathbb{N}$. Notice that according to the construction of V_{N_1}, it is only guaranteed that $\{\tilde{u}_k : 1 \leq k \leq K\} \subset V_{N_1}$ for a *fixed* K. For $k \geq K+1$, $\tilde{u}_k$ may not lie in V_{N_1} in general. Only v_k and ρ_k introduced here and specifically defined later, are finite-dimensional representations of u_k and π_k, for all $k \geq 1$. We summarize the relationship between the above family of functions in Figure 2 for the readers' convenience.

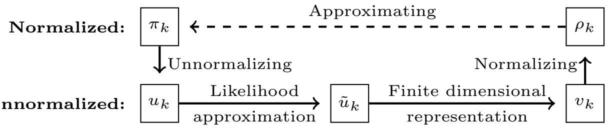


Fig. 2. Summation of the relationship among the families of functions π_k, u_k, $\tilde{u}_k$, v_k and ρ_k, for $k \geq 1$.

To find an element in V_{N_1} to approximate the exact solution of the filtering problem, let $\{\psi_i\}_{i=1}^{N_1}$ be an orthonormal basis of V_{N_1}, then there exist constants $\{a_i(1)\}_{i=1}^{N_1} \subset$

$\mathbb{R}$ such that

$$\pi_1(x) = u_1(x) = v_1(x) \triangleq \sum_{i=1}^{N_1} a_i(1)\psi_i(x). \tag{17}$$

since u_1 is an element in V_{N_1} according to the definition.

Next, if the element v_k $(k \geq 1)$, which we use to approximate u_k (or $\tilde{u}_k$) at time step k, can be expressed as

$$(\tilde{u}_k(x) \approx)\, v_k(x) \triangleq \sum_{i=1}^{N_1} \tilde{a}_i(k)\psi_i(x), \tag{18}$$

where $\tilde{a}(k) = [\tilde{a}_1(k), \cdots, \tilde{a}_{N_1}(k)]^\top \in \mathbb{R}^{N_1}$ a given N_1-dimensional vector. Then at time step $k+1$, the element v_{k+1} we use to approximate $\tilde{u}_{k+1}(x) = \tilde{\mathcal{F}}_2\left(\mathcal{F}_1(\tilde{u}_k), y_{k+1}\right)(x)$ is given by

$$(\tilde{u}_{k+1}(x) \approx)\, v_{k+1}(x) \triangleq \sum_{i=1}^{N_1} \tilde{a}_i(k+1)\psi_i(x), \tag{19}$$

where

$$\begin{aligned} \tilde{a}_i(k+1) &= \langle \tilde{\mathcal{F}}_2(\mathcal{F}_1(v_k), y_{k+1}), \psi_i \rangle \\ &= \left\langle \sum_{l=1}^{N_1} \tilde{a}_l(k) \left(\sum_{j=1}^{N} \alpha_j(y_{k+1})\phi_j \right) \mathcal{F}_1\psi_l, \psi_i \right\rangle \\ &\triangleq \lambda_i(\tilde{a}(k), y_{k+1}), \end{aligned} \tag{20}$$

with $\lambda \triangleq [\lambda_1, \cdots, \lambda_{N_1}]^\top : \mathbb{R}^{N_1} \times \mathbb{R}^{d_1} \to \mathbb{R}^{N_1}$ a continuous function, and $\langle \cdot, \cdot \rangle$ denoting the inner product in $\mathcal{L}^2(\mathbb{R}^d)$.

Remark 1. *Notice that, according to the construction of $\mathcal{I}_{K,N}$, the approximation signs '$\approx$' in (18) and (19) are in fact equality for $1 \leq k \leq K-1$, because each $\tilde{u}_k$ ($1 \leq k \leq K$) is itself an element in the finite-dimensional vector space spanned by $\mathcal{I}_{K,N}$. For $k \geq K+1$, we will prove rigorously in Section 4 that $\tilde{u}_k$ can also be well approximated by v_k because of the stability assumption.*

As a normalized version of v_k, the element ρ_k we use to approximate π_k, at each time step $k \geq 1$, is given by

$$\rho_k(x) = \sum_{i=1}^{N_1} a_i(k)\psi_i(x) \tag{21}$$

with

$$\begin{aligned} a_i(k+1) &= \frac{\tilde{a}_i(k+1)}{\int_{\mathbb{R}^d} v_{k+1}(x)dx} \\ &= \frac{\lambda_i(\tilde{a}_k, y_{k+1})}{\sum_{i=1}^{N_1} \lambda_i(\tilde{a}_k, y_{k+1}) \int_{\mathbb{R}^d} \psi_i(x)dx} \triangleq \eta_i(\tilde{a}_k, y_{k+1}), \end{aligned} \tag{22}$$

for each $i = 1, \cdots, N_1$.

As a homogeneous function of the first variable, the value of $\eta_i(\tilde{a}(k), y_{k+1})$ does not change if we make a scaling to $\tilde{a}(k)$. Especially, we have

$$\eta_i(\tilde{a}(k), y_{k+1}) = \eta_i(a(k), y_{k+1}), \quad \forall\, 1 \leq i \leq N_1,\ k \geq 1. \tag{23}$$

In fact, let us denote

$$c_k = \int_{\mathbb{R}^d} v_k(x)dx = \sum_{i=1}^{N_1} \tilde{a}_i(k) \int_{\mathbb{R}^d} \psi_i(x)dx,\ k \geq 1, \tag{24}$$

then

$$\begin{aligned} \eta_i(a(k), y_{k+1}) &= \frac{\lambda_i(\frac{\tilde{a}(k)}{c_k}, y_{k+1})}{\sum_{i=1}^{N_1} \lambda_i(\frac{\tilde{a}(k)}{c_k}, y_{k+1}) \int_{\mathbb{R}^d} \psi_i(x)dx} \\ &= \frac{\frac{1}{c_k}\lambda_i(\tilde{a}(k), y_{k+1})}{\frac{1}{c_k}\sum_{i=1}^{N_1} \lambda_i(\tilde{a}(k), y_{k+1}) \int_{\mathbb{R}^d} \psi_i(x)dx} = \eta_i(\tilde{a}(k), y_{k+1}), \end{aligned}$$

where the second equality holds because of the linearity of λ_i, with respect to $\tilde{a}(k)$, according to the definition (20).

Therefore, the propagation of conditional pdfs is represented by $\{\rho_k : k \geq 1\}$ with the orthonormal basis $\{\psi_i\}_{i=1}^{N_1}$ and the parameters $a(k) \in \mathbb{R}^{N_1}$ $(k \geq 1)$ evolving according to the open dynamics

$$a(k+1) = \eta(a(k), y_{k+1}), \quad k \geq 1. \tag{25}$$

3.3 Implementation of Recurrent Neural Network Spectral Method

Finally, with the finite-dimensional representation of conditional pdfs given by the parameters $a(k)$ and the orthonormal basis $\{\psi_i\}_{i=1}^{N_1}$, the integral operator $\mathcal{F}_3$ can be expressed as

$$\begin{aligned} \mathcal{F}_3(\rho_k) &= \int_{\mathbb{R}^d} f(x)\rho_k(x)dx \\ &= \sum_{i=1}^{N_1} a_i(k) \int_{\mathbb{R}^d} f(x)\psi_i(x)dx \triangleq \beta(a(k)) \end{aligned} \tag{26}$$

which is a linear combination of $a_i(k)$ with coefficients given by $\int_{\mathbb{R}^d} f(x)\psi_i(x)dx$.

Up to now, by (25) and (26), the calculation procedure illustrated in Figure 1 is conducted (or approximated) in finite dimensional, with the help of N_1-dimensional vectors $a(k)$ and this finite dimensional dynamics of $a(k)$ is illustrated by Figure 3.

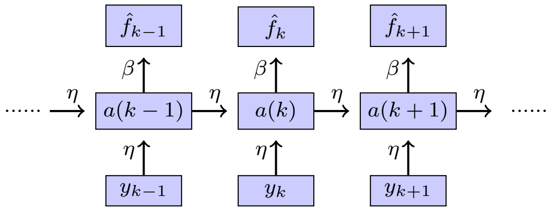


Fig. 3. Illustration of the finite-dimensional approximation to the calculation procedure in Figure 1.

Remark 2. *The open dynamical system defined by* (25) *and* (26)*, and illustrated by Figure 3, can be expressed by the following compact form:*

$$\begin{aligned} a(k+1) &= \frac{\Psi(y_{k+1})a(k)}{H^\top \Psi(y_{k+1})a(k)}, \\ \beta(a(k+1)) &= F^\top a(k+1). \end{aligned} \tag{27}$$

with $F, H \in \mathbb{R}^{N_1}$ constant vectors and $\Psi(y_{k+1}) \in \mathbb{R}^{N_1 \times N_1}$ matrix-valued function of y_{k+1}. This compact form (27) *is also shared with classical spectral method based nonlinear filtering algorithms such as those proposed in [12, 24].*

Although the above finite dimensional realization of the calculation procedure of Figure 1 depends on the choice of orthonormal basis $\{\psi_i\}_{i=1}^{N_1}$, the computation in Figure 3 only involves the propagation of the N_1-dimensional parameter $a(k)$, while the basis functions $\{\psi_i\}_{i=1}^{N_1}$ only affect the specific expression of the functions η and β. Hence, instead of the closed form of the basis functions $\{\psi_i\}_{i=1}^{N_1}$, we are more concerned with the number N_1, which corresponds to the computational efficiency of this algorithm.

Heuristically, in comparison with classical spectral methods, the orthonormal basis we use in this algorithm is constructed directly from the unnormalized conditional pdfs, and is supposed to be more efficient since it exploits the specific structure of the filtering system. More importantly, although the exact expression of each ψ_i may be hard to obtain, the functions η and β in Figure 3 can be easily learned and approximated by a recurrent neural network with a simulation dataset in short time steps $1 \leq k \leq K$ and this is why we refer to this proposed algorithm as Recurrent Neural Network Spectral Method (RNNSM).

In fact, because the functions $\eta : \mathbb{R}^{N_1} \times \mathbb{R}^{d_1} \to \mathbb{R}^{N_1}$, $\beta : \mathbb{R}^{N_1} \to \mathbb{R}$ are both continuous, it is theoretically proved that the open dynamical system,

$$\begin{cases} a(k+1) = \eta(a(k), y_{k+1}), \\ \hat{f}_k = \beta(a(k)), \end{cases} \tag{28}$$

as shown in Figure 3, can be approximated with arbitrary precision by a recurrent neural network [7, 30]. Thus, the learning and generalization capability of recurrent neural network allows us to focus on the propagation of parameters $a(k)$ and dispense the complexity of calculating the exact expression of orthonormal basis functions. The entire framework of RNNSM is summarized in Table 2.

Table 2
Framework of Recurrent Neural Network Spectral Method (RNNSM)

1. Initialization:

(1) Input the functions φ, Γ, h, and the covariance matrices Q, R in filtering system (1). Input a test function f.

(2) Fix two positive integers $K, N \in \mathbb{N}$.

2. Likelihood approximation:

(1) Find an N-dimensional subspace $\mathcal{H}_N \subset \mathcal{L}^\infty(\mathbb{R}^d)$ and an orthonormal basis $\{\phi_i\}_{i=1}^N$ of $\mathcal{H}_N$.

(2) Determine an element $\sum_{i=1}^N \alpha_i(y)\phi_i$ to approximate the likelihood functions.

3. Finite dimensional representation:

(1) Construct a finite dimensional subspace V_{N_1} spanned by $\mathcal{I}_{K,N} = \left\{ \left(\prod_{j=1}^{k-1}(\phi_{i_j}\mathcal{F}_1)\right) u_1 : 1 \leq k \leq K,\ 1 \leq i_1, \cdots, i_k \leq N \right\}$, and choose an orthonormal basis $\{\psi_i\}_{i=1}^{N_1}$ of V_{N_1}.

(2) Find an element $\rho_k(x) = \sum_{i=1}^{N_1} a_i(k)\psi_i(x)$ to approximate the conditional density function π_k, with $a(k)$ evolves according to (25).

4. Recurrent neural network approximation:

Train a recurrent neural network to approximate the open dynamical system (28) and output $\hat{f}_k$ for each $k \geq 1$ as an estimation of $f(x_k)$ at time step k.

4 Convergence Analysis

In this section, we will present the convergence results of our proposed RNNSM. A proof of the main steps will be given here with the most important ideas and insights, while the details of the proofs can be found in Section 6, after the numerical results.

For the purpose of a rigorous convergence analysis, in addition to the stability assumption (Assumption 1), we also need the following regularity assumptions on the filtering system (1).

Assumption 2. *The functional $\mathcal{F}_1 : \mathcal{L}^2(\mathbb{R}^d) \cap \mathcal{L}^1_+(\mathbb{R}^d) \to \mathcal{L}^2(\mathbb{R}^d) \cap \mathcal{L}^1_+(\mathbb{R}^d)$ is bounded, i.e., there exists a constant $C > 1$ such that*

$$\|\mathcal{F}_1(u)\|_2 \leq C\|u\|_2,\ \|\mathcal{F}_1(u)\|_1 \leq C\|u\|_1, \tag{29}$$

holds for all $u \in \mathcal{L}^2(\mathbb{R}^d) \cap \mathcal{L}^1_+(\mathbb{R}^d)$.

Assumption 3. *For each $k \geq 1$, $L > 0$, and regular observation trajectories $\{y_k : k \geq 1\} \subset \mathscr{B}_L$, the unnormalized conditional pdf u_k decays faster than $|x|^{-d}$ as $|x| \to \infty$, i.e., there exist constants $\delta > 0$ and $M_0 > 0$, such that*

$$u_k(x) \leq C_0 |x|^{-(d+\delta)}, \quad \forall\, |x| > M_0,\ k \geq 1. \tag{30}$$

Assumption 4. *For each $L > 0$, the functional $\mathcal{F}_2(\mathcal{F}_1(\cdot), y) : \mathcal{L}^1(\mathbb{R}^d) \to \mathcal{L}^1(\mathbb{R}^d)$ is coercive for all $y \in \mathscr{B}_L$, in the sense that there exists a constant $0 < C_1 < 1$ (which may depend on L) such that*

$$\|\mathcal{F}_2(\mathcal{F}_1(u), y)\|_1 \geq C_1 \|u\|_1, \tag{31}$$

holds for all $u \in \mathcal{L}^1_+(\mathbb{R}^d), y \in \mathscr{B}_L$.

Assumption 3 and 4 are made on specific bounded observation trajectories. Because the convergence results in this paper are obtained in the sense of mathematical expectations, we also need the following assumption on the entire set of trajectories, which indicates that most of the observations we can obtain in practice will be *regular.*

Assumption 5. *The observation process $\{y_k : k \geq 1\}$ of the filtering system (1) will remain in a large ball $\mathscr{B}_L$ with high probability, that is,*

$$\lim_{L \to \infty} P\left[\sup_{k \geq 1} |y_k| \geq L\right] = 0. \tag{32}$$

Also, the conditional pdf π_k is uniformly square integrable, in the sense that there exists a constant $M_1 > 0$, such that

$$\sup_{k \geq 1} \mathbb{E}\left[\|\pi_k\|_2^2\right] \leq M_1 < \infty. \tag{33}$$

Our main theorem in this paper is stated as follows:

Theorem 1. *For a given nonlinear filtering system (1) and a given test function $f \in \mathcal{L}^2(\mathbb{R}^d)$, under Assumptions 1 to 5, the solution of the filtering system (1) can be approximated by the proposed RNNSM with arbitrary accuracy. That is, for any $\epsilon > 0$, there exists an N_1-dimensional system (28) such that*

$$\mathbb{E}\left|\mathbb{E}[f(x_k)|\mathcal{Y}_k] - \beta(a(k))\right| < \epsilon, \quad \forall\, k \geq 1. \tag{34}$$

where $a(k) = [a_1(k), \cdots, a_{N_1}(k)]^\top \in \mathbb{R}^{N_1}$ is the N_1-dimensional parameter which propagates according to (28), and β is the output function illustrated in Figure 3.

The proof of Theorem 1 can be divided into two parts. First, we use the theory of spectral methods to show that Theorem 1 holds in a finite time interval $1 \leq k \leq K_1$, with a given $K_1 \in \mathbb{N}$, and then use the stability assumption of the filtering system to prove that Theorem 1 holds on the whole timeline. The two parts of the proof will be demonstrated in the following two subsections, respectively.

4.1 Convergence analysis in a finite time interval

For a given terminal time $K_1 \in \mathbb{N}$, let us first deal with the case on the time interval $1 \leq k \leq K_1$. As in the approximation procedure shown in Figure 2, we first estimate the differences between the unnormalized functions u_k, $\tilde{u}_k$ and v_k, and then deal with the difference between the normalized functions π_k and ρ_k.

Theorem 2. *Let $u = \{u_k : k \geq 1\}$ be the unnormalized conditional pdf evolving according to*

$$u_{k+1} = \mathcal{F}_2(\mathcal{F}_1(u_k), y_{k+1}), \quad k \geq 1; \tag{35}$$

and let $\tilde{u} = \{\tilde{u}_k : k \geq 1\}$ be the approximated unnormalized conditional pdf evolving according to (15).

Let $V_{N_1} = Span\{\mathcal{I}_{K_1,N}\}$ be the finite-dimensional vector space in RNNSM, and we need to introduce the following ***distance assumption*** *at this stage:*

Distance Assumption. *Assume that for regular observation trajectory $\{y_k : 1 \leq k \leq K_1\} \subset \mathscr{B}_L$ and for each k, ($1 \leq k \leq K_1$), the distance between the normalized function $\mathcal{N}(\tilde{u}_k)$ and the finite dimensional space V_{N_1} is small, i.e., $\forall\ \epsilon > 0$, there exist $N \in \mathbb{N}$ and a finite-dimensional space $V_{N_1} = Span\{\mathcal{I}_{K_1,N}\}$, such that,*

$$dist(\mathcal{N}(\tilde{u}_k), V_{N_1}) < \epsilon,\ \forall\ 1 \leq k \leq K_1. \tag{36}$$

Let $\{\psi_i\}_{i=1}^{N_1}$ be the orthonormal basis of V_{N_1}, and $v = \{v_k : 1 \leq k \leq K_1\}$ be the set of functions defined according to (18) with parameters $\tilde{a}(k)$ evolving as (20).

For sufficiently small $\epsilon > 0$, if we can choose $N \in \mathbb{N}$ and the finite-dimensional $V_{N_1} = Span\{\mathcal{I}_{K_1,N}\}$ according to Distance Assumption, such that (36) holds, then,

(i) Under Assumption 2, for each regular observation trajectory $\{y_k : 1 \leq k \leq K_1\} \subset \mathscr{B}_L$, the following estimation holds:

$$\|u_k - v_k\|_2 \leq 4\epsilon K_1 C^{K_1 - 1}\left(\|u_1\|_2 + \|u_1\|_1\right), \tag{37}$$

with the constant C defined in (29).

(ii) Let us denote

$$\epsilon_0 \triangleq 4\epsilon K_1 C^{K_1-1}\left(\|u_1\|_2 + \|u_1\|_1\right). \tag{38}$$

Under Assumptions 2 and 3, for each regular observation trajectory $\{y_k : 1 \leq k \leq K_1\} \subset \mathscr{B}_L$, and for ϵ small enough such that $M \triangleq \epsilon_0^{-\frac{2}{d+2\delta}} > M_0$, with M_0 defined in (30), we have

$$\|u_k - v_k 1_{\{|x| \leq M\}}\|_1 < \left(C(d)^{\frac{1}{2}} + \frac{1}{\delta} C_0 C(d)\right) \epsilon_0^{\frac{2\delta}{d+2\delta}}, \tag{39}$$

for all $1 \leq k \leq K_1$, where $C(d) > 0$ is a constant that only depends on d and C_0 is the constant defined in (30).

The detailed proof of Theorem 2 is presented in Section 6.1. Here, we would like to remark that the distance assumption (36) in Theorem 2 naturally holds if we only consider the finite time interval $1 \leq k \leq K_1$ with initial value $\tilde{u}_1 = u_1$, because $\tilde{u}_k$ itself is an element in V_{N_1}. For the case $k > K_1$, however, **Distance Assumption** only holds under the stability property of the system, which is shown later in Lemma 1 in Section 4.2.

With Theorem 2, we can derive the following two corollaries, which state the convergence of RNNSM in a finite time interval both for regular observation trajectories and in the sense of taking expectations. The proof of these two corollaries are given in Section 6.2 and Section 6.3, respectively.

Corollary 1. *Under Assumptions 2 to 4 and the additional distance assumption in Theorem 2, for each regular observation trajectory $\{y_k : 1 \leq k \leq K_1\} \subset \mathscr{B}_L$, we have for $1 \leq k \leq K_1$,*

$$\|\pi_k - \rho_k 1_{\{|x| \leq M\}}\|_2 = O\left(\epsilon_0^{\frac{2\delta}{d+2\delta}}\right), \tag{40}$$

as $\epsilon_0 \to 0$, where ϵ_0 is defined in (38).

Remark 3. *The truncated functions $v_k 1_{\{|x| \leq M\}}$ (or $\rho_k 1_{\{|x| \leq M\}}$) still lie in an N_1-dimensional vector space which is spanned by $\{\psi_i 1_{\{|x| \leq M\}} : 1 \leq i \leq N_1\}$. For the rest of our discussion, we will not distinguish v_k (and ρ_k) with the truncated functions $v_k 1_{\{|x| \leq M\}}$ (and $\rho_k 1_{\{|x| \leq M\}}$), or simply make the assumption that v_k (and ρ_k) vanish when $|x| \geq M$, because in our proposed RNNSM, it is the coefficients $a_i(k)$ that are more important, rather than the exact form of the basis functions.*

Corollary 2. *Under Assumptions 2 to 5, and the additional **Distance Assumption** in Theorem 2, for $1 \leq k \leq K_1$, ρ_k approximates π_k well in the sense that for each $\epsilon > 0$, there exists an N_1-dimensional vector space V_{N_1}, such that the density function $\rho_k \in V_{N_1}$ constructed in RNNSM satisfies*

$$\mathbb{E}\left[\|\pi_k - \rho_k 1_{A_k}\|_2\right] < \epsilon, \ \forall\ 1 \leq k \leq K_1, \tag{41}$$

where $A_k \triangleq \{\sup_{1 \leq j \leq k} |y_j| \leq L(\epsilon)\}$ is the event containing observations with regular trajectories up to time step k, with $L(\epsilon)$ a constant depending on ϵ.

Remark 4. *The meaning of $\rho_k 1_{A_k}$ indicates that our proposed filtering algorithm will only give responses to regular trajectories for practical use. In fact, since every observation sensor has its own bandwidth or threshold, it is supposed to report error when the bandwidth or threshold is broken through, and therefore, it is reasonable to consider the performance of filtering algorithms based on only regular observations. Moreover, just as we did in Remark 3, we will not distinguish ρ_k and $\rho_k 1_{A_k}$ when there in no confusion.*

4.2 Convergence analysis for the whole timeline

In this subsection, we will prove that Theorem 1 in fact holds throughout the whole timeline under the stability assumption of filtering systems. To this end, we will first show that under Assumption 1, the **Distance Assumption** in Theorem 2 holds, i.e., for all $k \geq 1$, the normalized function $\mathcal{N}(\tilde{u}_k)$ is indeed close to the finite dimensional space V_{N_1}, when N_1 is sufficiently large.

Lemma 1. *Under Assumptions 1 - 5, the normalized function $\mathcal{N}(\tilde{u}_k)$ can be approximated well by elements in V_{N_1}, the finite-dimensional vector space constructed in RNNSM, for sufficiently large N_1, in the sense that for each $\epsilon > 0$, there exists $K_0, N_0 \in \mathbb{N}$, such that*

$$\mathbb{E}\|\mathcal{N}(\tilde{u}_k) - w_k\|_2 < \epsilon, \ \forall\ k \geq 1. \tag{42}$$

where w_k is the projection of $\mathcal{N}(\tilde{u}_k)$ onto the finite dimensional space $V_{N_1} = Span\{\mathcal{I}_{K_0,N_0}\}$.

The technical and somewhat tedious proof of Lemma 1will be included in Section 6.4. We are now prepared to present a comprehensive proof of Theorem 1.

Proof of Theorem 1. For each $\epsilon > 0$, according to Assumption 1, there exists a constant $K_1 > 0$, such that

$$\mathbb{E}\|\pi_k^\mu - \pi_k^\nu\|_2 < \frac{\epsilon}{2}. \tag{43}$$

for all $k \geq K_1$ and initial values $\mu, \nu \in \mathcal{L}^2(\mathbb{R}^d)$.

In the meanwhile, according to Lemma 1 and the theory of Kolmogorov n-width (11), for arbitrary $\epsilon' > 0$, there exist $K_0, N \in \mathbb{N}$, such that

(i) There exists an N-dimensional subspace $\mathcal{H}_N \subset \mathcal{L}^2(\mathbb{R}^d)$ such that $\|\mathcal{F}_2(u, y) - \tilde{\mathcal{F}}_2(u, y)\|_2^2 \leq \epsilon' \|u\|_2^2$, for all $y \in \mathscr{B}_L$, $u \in \mathcal{L}^2(\mathbb{R}^d)$.

(ii) There exists a finite dimensional space $\tilde{V}$ given by $\tilde{V} = \text{Span}\{\mathcal{I}_{K_0,N}\}$, such that $\mathbb{E}\|\mathcal{N}(\tilde{u}_k) - w_k\|_2 < \epsilon'$, for all $k \geq 1$, where w_k is the projection of $\mathcal{N}(\tilde{u}_k)$ onto $\tilde{V}$.

Let us take $K = \max\{K_0, K_1\}$, and define $V_{N_1} = \text{Span}\{\mathcal{I}_{K,N}\}$. Notice that according to the definition of $\tilde{u}_k$ in (15), the value of $\tilde{u}_k$ only depends on the coefficient N in $\mathcal{I}_{K,N}$, and is independent of the coefficient K. Because $\tilde{V} \subset V_{N_1}$, we have

$$\text{dist}(\mathcal{N}(\tilde{u}_k), V_{N_1}) \leq \text{dist}(\mathcal{N}(\tilde{u}_k), \tilde{V}) < \epsilon', \tag{44}$$

Thus, according to Theorem 2 and Corollary 2, we can choose sufficiently small ϵ' in (44), such that the approximated conditional pdf $\rho_k \in V_{N_1}$ obtained through RNNSM (28) satisfies

$$\mathbb{E}\left[\|\pi_k - \rho_k\|_2\right] < \frac{\epsilon}{2}, \ \forall\, 1 \leq k \leq K. \tag{45}$$

which proves the convergence result for $1 \leq k \leq K_1$, since $K_1 \leq K$.

For $k \geq K_1 + 1$, consider the exact solution to the filtering system, $\tilde{\pi} = \{\tilde{\pi}_j : k - K_1 \leq j \leq k\}$, on the interval $[k - K_1, k]$, with initial value $\tilde{\pi}_{k-K_1} = \rho_{k-K_1}$. On the one hand, because of the filtering stability, $\tilde{\pi}_k$ is close to π_k; on the other hand, because the initial value $\tilde{\pi}_{k-K_1}$ is equal to ρ_{k-K_1}, the convergence result in the case of finite time interval also implies that $\tilde{\pi}_k$ is close to ρ_k. In fact, combining (43) and (45), we have

$$\mathbb{E}\left[\|\pi_k - \rho_k\|_2\right] \leq \mathbb{E}\|\rho_k - \tilde{\pi}_k\|_2 + \mathbb{E}\|\tilde{\pi}_k - \pi_k\|_2 < \epsilon. \tag{46}$$

At last, for each $f \in \mathcal{L}^2(\mathbb{R}^d)$, according to Cauchy-Schwartz inequality,

$$\mathbb{E}\left|\mathbb{E}[f(x_k)|\mathcal{Y}_k] - \beta(a(k))\right| \leq \mathbb{E}\|f\|_2\|\pi_k - \rho_k\|_2 < \epsilon\|f\|_2.$$

which is the desired result. □

Finally, we would like to give a rough estimation of the convergence rate of RNNSM. On the one hand, according to the Kolmogorov n-width theory [29], the convergence rate in (11) with respect to N is exponential. On the other hand, according to the filtering stability theory [3], for a quite general class of stable filtering systems, the convergence rate in (16) with respect to K is also exponential. In the meanwhile, even if we do not consider the potential linear dependence of functions in $\mathcal{I}_{K,N}$, the dimension N_1 of V_{N_1} not bigger than N^K, which is the cardinality of $\mathcal{I}_{K,N}$.

Therefore, in order to obtain an error bound $\epsilon > 0$, N and K only need to be of order $\log\left(\frac{1}{\epsilon}\right)$, as $\epsilon \to 0$, and the relationship between ϵ and the number of basis functions $N_1 \left(\leq N^K\right)$ can be roughly estimated as

$$\epsilon \leq N_1^{-\frac{C_2}{\log C_1 + \log\log\left(\frac{1}{\epsilon}\right)}}. \tag{47}$$

for some constants $C_1, C_2 > 0$, corresponding to the smoothness of likelihood functions and the stability of the filtering system, respectively. Although the denominator in the exponential on the right-hand side of (47) contains a term $\log\log\left(\frac{1}{\epsilon}\right)$, which will tend to infinity as $\epsilon \to 0$, yet for practice, it is more reasonable to regard this term as a constant, because $\log\log\left(\frac{1}{\epsilon}\right) < 4$, even if we take $\epsilon = 10^{-16}$.

Therefore, for filtering systems with *high* stability, RNNSM also enjoys a high convergence rate with respect to the number of basis functions N_1, which is often referred to as *spectral accuracy* [33].

5 Numerical Results

5.1 Neural network architecture

The RNN architecture of RNNSM is as follows [13]:

$$\begin{cases} \tilde{a}_{k+1} = \Phi\left(\tilde{a}_k, y_{k+1}; \theta_1\right), \\ \hat{f}(x_k) = \Gamma\left(\tilde{a}_k; \theta_2\right), \end{cases} \tag{48}$$

where $\theta = [\theta_1, \theta_2]$ represents all the trainable parameters. As an approximation to the open dynamics (28), Φ represents three-layered feedforward network with one input layer, one hidden layer with l neurons and one output layer, Γ is a linear function with input dimension l and output dimension is equal to the dimension of $f(x)$.

Remark 5. *Here in* (48)*, a generic RNN architecture is presented, and hereafter in this paper, we will apply this generic RNN to conduct our proposed RNNSM. In the meanwhile, other RNN architectures, such as long-short term memory (LSTM) [15] and gated recurrent unit (GRU) [8], mathematically share the same expression as* (48)*. Therefore, it is straightforward to replace the generic architecture by these variants of RNN, and may result in an improvement in the training speed and performance. For applications, finding the most suitable network architecture for our proposed RNNSM is also a promising direction, and is left for future work.*

Since the optimal estimate of the statistics $f(x_k)$ (and the exact solution of filtering problem) is $\mathbb{E}\left[f(x_k)\,|\mathcal{Y}_k\right]$, we aim to minimize

$$L_0(\theta) := \frac{1}{K_1 + 1}\mathbb{E}\left[\sum_{k=1}^{K_1}\left|\hat{f}(x_k) - \mathbb{E}\left[f(x_k)\,|\mathcal{Y}_k\right]\right|^2\right], \tag{49}$$

where $K_1 \in \mathbb{N}$ is the total time step in training. However, we cannot obtain the data of $\mathbb{E}[f(x_k)|\mathcal{Y}_k]$ using filtering system. One key observation is that $\mathbb{E}[|f(x_k) - \hat{f}(x_k)|^2] = \mathbb{E}[|f(x_k) - \mathbb{E}[f(x_k)|\mathcal{Y}_k]|^2] + \mathbb{E}[|\mathbb{E}[f(x_k)|\mathcal{Y}_k] - \hat{f}(x_k)|^2]$. Therefore $\text{argmin}_\theta L_0(\theta) = \text{argmin}_\theta L(\theta)$, where

$$L(\theta) \triangleq \frac{1}{K_1 + 1}\mathbb{E}\left[\sum_{k=1}^{K_1}\left|\hat{f}(x_k) - f(x_k)\right|^2\right]. \tag{50}$$

It means that the training data we used are $\{y_k, f(x_k)\}_{k=1}^{K_1}$ which can be easily generated from the system (1), rather than using the unavailable $\mathbb{E}\left[f(x_k)\,|\mathcal{Y}_k\right]$. In real computations, the expectation in $L(\theta)$ is approximated by the average of the results of a large number of trials, and hence the loss function is defined as follows:

$$L^{(K_2)}(\theta) \triangleq \frac{1}{K_2}\frac{1}{K_1+1}\sum_{j=1}^{K_2}\sum_{k=1}^{K_1}\left|\hat{f}(x_k)(\omega_j) - f(x_k)(\omega_j)\right|^2, \tag{51}$$

where $\hat{f}(x_k)(\omega_j)$ is the output of RNNSM with input $y_k(\omega_j)$, and K_1, K_2 are the total time steps and the numbers of Monte Carlo paths in training, respectively.

Training data is crucial for the RNNSM. If we know the system dynamics in (1), we can generate synthetic data $\left\{\{(y_k(\omega_j), f(x_k)(\omega_j))\}_{k=1}^{K_1}\right\}_{j=1}^{K_2}$ by simulating these dynamics. Classic filtering algorithms like PF and EKF also require knowledge of these system dynamics. Without this knowledge, EKF and PF cannot be applied. When real-world data is available, we collect observations from sensors or measurement devices in the system. Sometimes, relevant datasets are available from public repositories or previous studies. For example, in stock price prediction, historical price data can be used [9].

The detailed steps of the implementation of RNNSM and the training procedure of the RNN architecture (48) are listed in Algorithm 1.

Algorithm 1 Implementation of Recurrent Neural Network Spectral Method

```
Off-line algorithm: Training RNN
(1) Input the parameters including: batch size $M$; total epochs $I$; hidden layer neurons $l$; learning rate $\lambda$.
(2) Input synthetic training data or real-world data $\left\{\{(y_k(\omega_j), f(x_k)(\omega_j))\}_{k=1}^{K_1}\right\}_{j=1}^{K_2}$.
(3) Train RNN with the following procedure. Record the trained RNN with parameter $\theta$.
    for $i = 1, \dots I$ do
        1. Sample batch $\left\{\{(y_{\tau_i}(\omega_j), f(x_k)(\omega_j))\}_{k=1}^{K_1}\right\}_{j=1}^{M}$ from the training data;
        2. Compute loss $L^{(M)}(\theta)$ via (51);
        3. Update $\theta$ via $\theta \leftarrow \theta - \lambda\nabla_\theta L^{(M)}(\theta)$.
    end for

Online algorithm: Implementation of RNNSM
while time step $k \geq 1$ do
    (1) Input the observation $y_k$ to the trained RNN with parameter $\theta$.
    (2) Record the output $\hat{f}_k$ from the RNN as the solution of the filtering problem (i.e. the estimation to $f(x_k)$) at time step $k$.
    (3) Update $k \leftarrow k + 1$.
end while
```

For selecting the hyperparameters in RNNSM, we approximate the true loss function using sample paths in the training set via Monte Carlo methods, choosing over 1500 paths. The batch size, which affects training speed, ranges from 32 to 512. The number of hidden neurons l, learning rate, and total epochs significantly influence the results. The parameter l impacts the model's approximation capability: if too small, it fails to capture filtering information; if too large, it slows convergence. An appropriate l is proportional to the linear or squared dimensionality of the state d. A learning rate between 10^{-5} and 10^{-3} is suitable, and the total epochs should range from 1500 to 5000.

In order to compare the performance of different methods, we introduce the Mean Squared Error (MSE) and the Mean Error at time k (ME$_k$) based on 100 realizations, which are defined as follows:

$$\begin{aligned} \mathrm{MSE} &\triangleq \frac{1}{100}\sum_{l=1}^{100}\frac{1}{K+1}\sum_{k=0}^{K}\left|x_k^{(l)} - \hat{x}_k^{(l)}\right|^2, \\ \mathrm{ME}_k &\triangleq \frac{1}{100}\sum_{l=1}^{100}\left|x_k^{(l)} - \hat{x}_k^{(l)}\right|, \end{aligned} \tag{52}$$

where $x_k^{(l)}$ is the real state at time instant k in the l-th experiment and $\hat{x}_k^{(l)}$ is the estimation of $x_k^{(l)}$, with $0 \leq k \leq K$, where $K \in \mathbb{N}$ is the total time step.

In the following numerical illustrations, we evaluate the efficacy of RNNSM compared to EKF and PF. All experiments were conducted on NVIDIA RTX2060 GPUs and a computational platform with 16 Intel Core i7-10700 CPUs at 2.90 GHz. RNNSM is implemented using PyTorch, while EKF and PF use NumPy. [3]

5.2 *Example 1*

The numerical example we consider in this paper is the following discrete filtering system:

$$\begin{cases} x_{k+1} = (I_d + \alpha A_d)x_k + \alpha\cos(x_k) + v_k, \\ y_k = h(x_k) + w_k, \end{cases} \tag{53}$$

where $h(x) = \alpha[x_1^3, \cdots, x_d^3]^\top$, $k = 1, \cdots, K$ with $K = 1000$, $\alpha = 0.01$, the initial state x_1 follows $\mathcal{N}(0, I_d)$ with identity matrix $I_d \in \mathbb{R}^{d\times d}$, $d = 10$, $\{v_k\}_{k=1}^K$ and $\{w_k\}_{k=1}^K$ are mutually independent Gaussian random vectors with zero means and covariance matrices $\mathbb{E}\left[v_k v_k^\top\right] = 0.01 I_d$

[3] The codes of the following numerical examples are available on the server at Beijing Institute of Mathematical Sciences and Applications (BIMSA). Readers with interests in the implementations of RNNSM may contact the corresponding authors to get access to the codes for academical use.

and $\mathbb{E}\left[w_k w_k^\top\right] = 0.01 I_d$, and $A_d = [a_{ij}]$ is a matrix with elements as follows: $a_{ij} = \begin{cases} 0.5, \ \text{if } i+1=j, \\ -1, \ \text{if } i=j, \\ 0, \ \text{otherwise.} \end{cases}$

Here, we compare the RNNSM with the classic EKF and PF. The initial mean and covariance of EKF are 0 and I_d, respectively. The parameters used in RNNSM is listed in TABLE 3.

Table 3
The parameters of RNNSM used in the three numerical examples.

Parameters	Ex. 1	Ex. 2	Ex. 3
paths in training set	1500	1500	1500
paths in test set	100	100	100
activation function	ReLU	ReLU	ReLU
optimizer	Adam	Adam	Adam
total epochs I	2000	2000	2000
batch size M	256	64	64
hidden layer neurons l	70	450	20
learning rate	0.0005	0.0001	0.0005

We first set the training step of RNNSM to $K_1 = K = 1000$ and N_{PF}, the number of particles of PF to the set $\{100, 500, 1000, 1500, 2000, 2500, 3000\}$. The performance of the three algorithms based on 100 experiments is shown in TABLE 4. It can be seen that RNNSM is the fastest, its running time being only 2.8% of that of the simple EKF. This is because the training session of RNNSM is done offline and we only need to run a simple RNN with $l = 70$ hidden layer neurons.The MSE of RNNSM is almost similar to that of PF with 3000 particles, while the running time of RNNSM is only **2 ‱** of that of PF. Considering the trade-off between MSE and computational time, RNNSM gives the best result.

Table 4
The average performance of different methods based on 100 simulations for system (53) in example 1.

Method	Number of particles	MSE	Running time (s)
EKF	\	1.5868	0.0423
RNNSM	\	**1.5083**	**0.0012**
PF	1000	1.5483	1.9689
	2000	1.5185	4.4508
	3000	**1.5070**	6.2048

The ME_k of different algorithms are displayed in Fig. 4. As time k goes on, the errors ME_k tend to be stable. RNNSM and PF with 3000 particles have similar performance and outperform the other algorithms. The stability of ME_k partially verifies that the filtering system (53) possesses the stable property stated in Assumption 1. Because as an empirical realization of statistics, for large $k_1, k_2 \gg 1$, ME_{k_1} and ME_{k_2} can be regarded as the mean error at the same time K_0, with different initial distributions $\pi_{k_1-K_0}$ and $\pi_{k_2-K_0}$. The stability of ME_k thus, to some extent, validate the capability of filtering system (53) to 'forget' initial values.

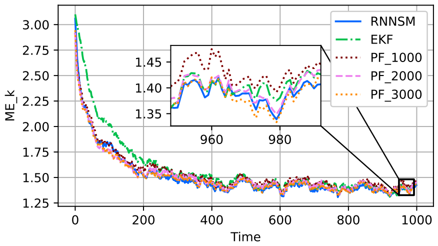


Fig. 4. The ME_k of RNNSM, EKF and PF. PF_N represents the estimate by PF with N particles.

Next, we train RNNSM with training step $K_1 \in \{50, 200, 400, 600, 800, 1000\}$ and test it with $K = 1000$ steps. The result is shown in Fig. 5. For this filtering system with stability property, the 1000-time-step MSE of RNNSM starts to approach the theoretical bound obtained by particle filters with training data of only 400 time steps. This numerical result verifies that we can train the RNNSM with temporal data generated through a short period of time while implementing it for a much longer period of time, which coincides with the construction procedure of the finite-dimensional vector space V_{N_1} in Section 3 and our theoretical result obtained in Section 4.

5.3 *Example 2*

We consider system (53) with dimension $d = 50$. The other settings are the same as that of (53). The parameters used in RNNSM are listed in TABLE 3. Let particle number $N_{\text{PF}} \in \{10000, 50000, 90000\}$. The detailed MSE based on 100 experiments can be found in Table 5. It can be found that RNNSM shows apparent advantages over EKF and PF from both accuracy and computational time in this high-dimensional example.

5.4 *Example 3*

We consider the following discrete system:

$$\begin{cases} x_{k+1} = \varphi(x_k) + v_k \\ y_{k+1} = h(x_{k+1}) + w_{k+1} \end{cases}, \tag{54}$$

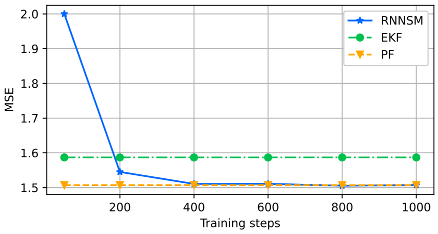


Fig. 5. The MSE of RNNSM with different training steps.

Table 5
The average performance of different methods based on 100 simulations for system (53) in example 2.

Method	Number of particles	MSE	Running time (s)
EKF	\	3.7149	0.2795
RNNSM	\	**3.3201**	**0.0009**
PF	10000	3.8508	96.6163
	50000	3.6870	440.1928
	90000	3.6466	1093.7228

where $v_k \sim N(0, I_2)$, $w_k \sim N(0, 0.01 I_2)$,

$$\varphi(x) = \begin{bmatrix} \dfrac{-x_1^3}{1+5x_1^2} - 0.2x_2 \\ -\dfrac{x_2}{4+x_1^2+x_2^2} \end{bmatrix}, \; h(x) = \begin{bmatrix} -x_1x_2 \\ x_2^3 \end{bmatrix},$$

the initial state follows $\mathcal{N}(m_1, I_2)$, with $m_1 = [1,1]^\top$, and $1 \le k \le K = 100$. The initial mean and covariance used in EKF and PF are $[0,0]^\top$ and I_2, respectively. The initial hidden state of RNNSM is a zero vector and parameters used in RNNSM are listed in TABLE 3.

The estimation results of the EKF, PF, and RNNSM are displayed in Fig. 6a-6b. It is observed that RNNSM can track the real state well, EKF completely fails and PF suffers from particle degeneracy and cannot output efficient numerical results in some experiments. For example, in Fig. 6b, the estimation of PF disappears around instant 70 in this experiment. This conclusion is also supported by Fig. 6c, which presents the ME_k based on 100 experiments. Detailed MSE and computation times can be found in TABLE 6.

Table 6
The average performance of different methods based on 100 simulations for system (54).

Method	Number of particles	MSE	Running time (s)
EKF	\	\	0.0033
RNNSM	\	**0.4250**	**0.000049**
PF	500	\	0.0590
	800	\	0.0720

5.5 Discussion

Theoretically, PF demonstrates a promising characteristic whereby, under certain assumptions, the mean square error between the optimal estimate of a test function and its PF estimation converges to zero at a rate of $1/N_{\mathrm{PF}}$ [11]. Consequently, the PF estimation can be considered approximately optimal when N_{PF} is sufficiently large.

While both PF and RNNSM can achieve arbitrarily small estimation errors when functioning optimally, the distinction lies in their underlying mechanisms. PF relies on finite stochastic particles to approximate the posterior density function of the state, whereas RNNSM employs finite basis functions, akin to the finite neurons in its hidden state, to perform the same approximation. Notably, RNNSM leverages supervised learning principles, optimizing the finite neurons in its hidden state to refine its estimation.

In practical terms, simulations reveal a substantial time advantage for RNNSM over PF, particularly when aiming for comparable accuracy. This efficiency stems from RNNSM's offline training step, which contrasts with the real-time nature of PF's operation. Thus, while both methods may converge to similarly accurate estimates, RNNSM offers a more computationally efficient approach, making it a compelling choice in real-world applications.

6 Detailed Proofs of the Convergence Results

6.1 Proof of Theorem 2

Proof of Theorem 2. (i) For the $\mathcal{L}^2$-norm estimation (37), the difference between u_k and v_k can be estimated by $\|u_k - v_k\|_2 \le \|u_k - \tilde{u}_k\|_2 + \|\tilde{u}_k - v_k\|_2$.

For $1 \le k \le K_1 - 1$, the difference between $\tilde{u}_k$ and u_k satisfies

$$\begin{aligned} &\|u_{k+1} - \tilde{u}_{k+1}\|_2 \\ &\quad \le \|\mathcal{F}_2(\mathcal{F}_1(u_k), y_{k+1}) - \mathcal{F}_2(\mathcal{F}_1(\tilde{u}_k), y_{k+1})\|_2 \\ &\qquad + \|\mathcal{F}_2(\mathcal{F}_1(\tilde{u}_k), y_{k+1}) - \tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_k), y_{k+1})\|_2 \end{aligned}$$

Because $\exp(-\frac{1}{2}(y - h(x))^\top R^{-1}(y - h(x))) \le 1$, $\|\mathcal{F}_2(\mathcal{F}_1(u_k), y_{k+1}) - \mathcal{F}_2(\mathcal{F}_1(\tilde{u}_k), y_{k+1})\|_2 \le C\|u_k - \tilde{u}_k\|_2$, where the second inequality holds by Assumption 2.

Next, according to (14), for any $\epsilon > 0$, there exists $N \in \mathbb{N}$, such that the approximation functional $\tilde{\mathcal{F}}_2$, corresponding to N satisfies

$$\|\mathcal{F}_2(u, y) - \tilde{\mathcal{F}}_2(u, y)\|_2 \le \epsilon \|u\|_2, \tag{55}$$

for all $y \in \mathscr{B}_L$, $u \in \mathcal{L}^2(\mathbb{R}^d)$, and thus,

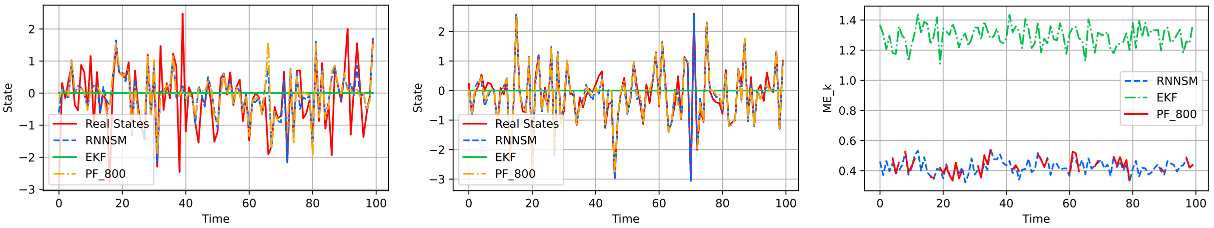
$$\|\mathcal{F}_2(\mathcal{F}_1(\tilde{u}_k), y_{k+1}) - \tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_k), y_{k+1})\|_2 \le C\epsilon \|\tilde{u}_k\|_2$$

holds for each regular observation trajectory $\{y_k : 1 \le k \le K_1\} \subset \mathscr{B}_L$. Hence,

$$\|u_{k+1} - \tilde{u}_{k+1}\|_2 \le C(\|u_k - \tilde{u}_k\|_2 + \epsilon \|\tilde{u}_k\|_2). \tag{56}$$

Inductively, we have $\|u_k - \tilde{u}_k\|_2 \le \epsilon \sum_{j=1}^{k-1} C^j \|\tilde{u}_{k-j}\|_2$.

Finally, let us estimate the $\mathcal{L}^2$-norm of the approximated solution $\tilde{u}_k$, for $1 \le k \le K_1$.



(a) First dimension (b) Second dimension (c) Mean Error at time k

Fig. 6. Estimation results of the EKF, PF and RNNSM for the example (54).

Note that, $\|\tilde{u}_{k+1}\|_2 = \|\tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_k), y_{k+1})\| \le C(1+\epsilon)\|\tilde{u}_k\|_2$, and $\|\tilde{u}_k\|_2 \le C^{k-1}(1+\epsilon)^{k-1}\|u_1\|_2$, $1 \le k \le K_1$. For all $1 \le k \le K_1$,

$$\|u_k - \tilde{u}_k\|_2 \le \epsilon C^{K_1-1}\sum_{j=0}^{K_1-2}(1+\epsilon)^j\|u_1\|_2.$$

Since $\sum_{j=0}^{K_1-2}(1+\epsilon)^j = \frac{1}{\epsilon}((1+\epsilon)^{K_1-1}-1)$, and for sufficiently small ϵ, for example $0 < \epsilon < \frac{1}{K_1-1}$, we have $(1+\epsilon)^{K_1-1} \le 1 + 3(K_1-1)\epsilon$, and thus,

$$\|u_k - \tilde{u}_k\|_2 \le 3\epsilon(K_1-1)C^{K_1-1}\|u_1\|_2, \tag{57}$$

for all $1 \le k \le K_1$.

Next, we need to estimate the norm $\|\tilde{u}_k - v_k\|_2$. Let us denote by w_k the projection of $\mathcal{N}(\tilde{u}_k)$ onto the finite dimensional space V_{N_1}, then for all $1 \le k \le K_1$,

$$\|\mathcal{N}(\tilde{u}_k) - w_k\|_2 = \mathrm{dist}(\mathcal{N}(\tilde{u}_k), V_{N_1}) < \epsilon. \tag{58}$$

Since $\tilde{u}_k = \|\tilde{u}_k\|_1\mathcal{N}(\tilde{u}_k)$, the function $\tilde{w}_k \triangleq \|\tilde{u}_k\|_1 w_k$ is also the projection of $\tilde{u}_k$ onto V_{N_1}. The idea is to track the evolution of the difference $v_k - \tilde{w}_k$.

Firstly, since $\tilde{w}_k$ is the projection of $\tilde{u}_k$ on V_{N_1}, we have for all $1 \le i \le N_1$, $\langle \tilde{w}_{k+1}, \psi_i\rangle = \langle \tilde{u}_{k+1}, \psi_i\rangle = \langle \tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_k), y_{k+1}), \psi_i\rangle$. Because $\tilde{w}_k, v_k \in V_{N_1}$,

$$\begin{aligned}\tilde{w}_{k+1} - v_{k+1} = &\sum_{i=1}^{N_1}\langle \tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_k - \tilde{w}_k), y_{k+1}), \psi_i\rangle\psi_i \\ &+ \sum_{i=1}^{N_1}\langle \tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{w}_k - v_k), y_{k+1}), \psi_i\rangle\psi_i.\end{aligned}$$

According to Bessel's inequality,

$$\begin{aligned}\|\tilde{w}_{k+1} - v_{k+1}\|_2 \le &\|\tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{u}_k - \tilde{w}_k), y_{k+1})\|_2 \\ &+ \|\tilde{\mathcal{F}}_2(\mathcal{F}_1(\tilde{w}_k - v_k), y_{k+1})\|_2.\end{aligned} \tag{59}$$

With Assumption 2 and the condition (55), $\|\tilde{w}_{k+1} - v_{k+1}\|_2 \le C(1+\epsilon)\left(\|\tilde{u}_k - \tilde{w}_k\|_2 + \|\tilde{w}_k - v_k\|_2\right)$. Inductively, we have $\|\tilde{w}_k - v_k\|_2 \le \sum_{j=1}^{k-1} C^{k-j}(1+\epsilon)^{k-j}\|\tilde{u}_j - \tilde{w}_j\|_2$. Thus, for $1 \le k \le K_1$,

$$\|\tilde{u}_k - v_k\|_2 \le \sum_{j=1}^{k} C^{k-j}(1+\epsilon)^{k-j}\|\tilde{u}_j - \tilde{w}_j\|_2. \tag{60}$$

According to (58), $\|\tilde{u}_j - \tilde{w}_j\|_2 = \|\tilde{u}_j\|_1\|\mathcal{N}(\tilde{u}_j) - w_j\|_2 < \epsilon\|\tilde{u}_j\|_1$, and for all $1 \le k \le K_1$,

$$\begin{aligned}\|\tilde{u}_k - v_k\|_2 &\le \epsilon K_1 C^{K_1-1}(1+\epsilon)^{K_1-1}\|u_1\|_1 \\ &\le 4\epsilon K_1 C^{K_1-1}\|u_1\|_1,\end{aligned} \tag{61}$$

Therefore, $\|u_k - v_k\|_2 \le \|u_k - \tilde{u}_k\|_2 + \|\tilde{u}_k - v_k\|_2 \le 4\epsilon K_1 C^{K_1-1}\left(\|u_1\|_2 + \|u_1\|_1\right)$.

(ii) For the $\mathcal{L}^1$-norm estimation (39), note that

$$\begin{aligned}&\|u_k - v_k 1_{\{|x|\le M\}}\|_1 \\ &\quad= \int_{|x|\le M}|u_k - v_k|dx + \int_{|x|\ge M}u_k dx.\end{aligned} \tag{62}$$

According to Cauchy-Schwartz inequality and (i), $\int_{|x|\le M}|u_k - v_k|dx \le C(d)^{\frac{1}{2}}\epsilon^{1-\frac{d}{d+2\delta}} = C(d)^{\frac{1}{2}}\epsilon_0^{\frac{2\delta}{d+2\delta}}$.

For the integration outside the ball $\{x \in \mathbb{R}^d : |x| \le M\}$, according to Assumption 3,

$$\int_{|x|\ge M}u_k dx \le \int_{|x|\ge M}\frac{C_0}{|x|^{d+\delta}}dx = C_0 C(d)\frac{1}{\delta}\epsilon_0^{\frac{2\delta}{d+2\delta}}.$$

Thus, we obtain the desired result. □

6.2 Proof of Corollary 1

Proof of Corollary 1. Firstly, with Assumption 3,

$$\begin{aligned}&\|u_k - v_k 1_{\{|x|\le M\}}\|_2 \\ &\quad\le \left(\int_{|x|\le M}|u_k - v_k|^2 dx + \int_{|x|\ge M}\left(\frac{C_0}{|x|^{d+\delta}}\right)^2 dx\right)^{\frac{1}{2}}.\end{aligned}$$

According to (37), which is the first result of Theorem 2, $\|u_k - v_k\|_2 < \epsilon_0$, and $M = \epsilon_0^{-\frac{2}{d+2\delta}}$, thus

$$\|u_k - v_k 1_{\{|x|\leq M\}}\|_2 < \left(1 + \frac{C_0^2 C(d)}{d+2\delta}\right)^{\frac{1}{2}} \epsilon_0. \qquad (63)$$

Moreover, since $\|u_k\|_1 \geq C_1^{k-1}\|u_1\|_1 \geq C_1^{K_1-1}\|u_1\|_1$ by (31), $\|u_k\|_2 \leq C^{k-1}\|u_1\|_2 \leq C^{K_1-1}\|u_1\|_2$ by (29) and according to Theorem 2,

$$\|u_k - v_k 1_{\{|x|\leq M\}}\|_1 < \left(C(d)^{\frac{1}{2}} + \frac{1}{\delta} C_0 C(d)\right)\epsilon_0^{\frac{2\delta}{d+2\delta}}, \quad (64)$$

for all $1 \leq k \leq K_1$, then for sufficiently small $\epsilon_0 > 0$, $\|v_k 1_{\{|x|\leq M\}}\|_1 \geq \frac{1}{2}C_1^{K_1-1}\|u_1\|_1$; $\|v_k 1_{\{|x|\leq M\}}\|_2 \leq 2C^{K_1-1}\|u_1\|_2$. Hence,

$$\begin{aligned}
&\|\pi_k - \rho_k 1_{\{|x|\leq M\}}\|_2 \\
&\leq \frac{\|v_k 1_{\{|x|\leq M\}}\|_2 \|u_k - v_k 1_{\{|x|\leq M\}}\|_1}{\|v_k 1_{\{|x|\leq M\}}\|_1 \|u_k\|_1} \\
&\quad + \frac{\|u_k - v_k 1_{\{|x|\leq M\}}\|_2}{\|u_k\|_1} \\
&\leq \frac{4C^{K_1-1}\|u_1\|_2}{C_1^{2K_1-2}\|u_1\|_1^2}\left(C(d)^{\frac{1}{2}} + \frac{1}{\delta}C_0 C(d)\right)\epsilon_0^{\frac{2\delta}{d+2\delta}} \\
&\quad + \frac{1}{C_1^{K_1-1}\|u_1\|_1}\left(1 + \frac{C_0^2 C(d)}{d+2\delta}\right)^{\frac{1}{2}}\epsilon_0.
\end{aligned}$$

which is of order $O\left(\epsilon_0^{\frac{2\delta}{d+2\delta}}\right)$ as $\epsilon_0 \to 0$. □

6.3 Proof of Corollary 2

Proof of Corollary 2. According to Assumption 5, for each $\epsilon > 0$, there exists $L(\epsilon) > 0$ such that $P\left[\sup_{k\geq 1}|y_k| > L(\epsilon)\right] \leq \frac{\epsilon^2}{4M_1}$, with M_1 defined in (33).

According to Corollary 1, there exists an N_1-dimensional space V_{N_1} and $\rho_k \in V_{N_1}$ obtained by RNNSM, such that $\|\hat{\pi}_k - \rho_k\|_2 < \frac{\epsilon}{2}$, for all $1 \leq k \leq K_1$ and almost surely in the event A_k. Since $\mathbb{E}\left[\|\pi_k\|_2^2\right] \leq M_1$, we have

$$\begin{aligned}
&\mathbb{E}\left[\|\pi_k - \rho_k 1_{A_k}\|_2\right] < \frac{\epsilon}{2}P(A_k) + \mathbb{E}\left[\|\pi_k\|_2 1_{A_k^c}\right] \\
&\leq \frac{\epsilon}{2} + \sqrt{\mathbb{E}[\|\pi_k\|_2^2]P(A_k^c)} \leq \frac{\epsilon}{2} + \sqrt{M_1\frac{\epsilon^2}{4M_1}} = \epsilon,
\end{aligned}$$

□

6.4 Proof of Lemma 1

Proof of Lemma 1. The idea of the proof is to find an element $\bar{w}_k$ in V_{N_1} using Assumption 1 for each $k \geq 1$, such that the distance between $\bar{w}_k$ and $\mathcal{N}(\tilde{u}_k)$ is smaller than ϵ, then with the property of the projection, (42) holds for all $k \geq 1$.

Step 1: To this end, we first show that for each $\epsilon > 0$, there exists $N \in \mathbb{N}$, such that $\mathbb{E}\left[\|\mathcal{N}(\tilde{u}_k) - \pi_k\|_2\right] < \frac{\epsilon}{3}$, $\forall\ k \geq 1$. In fact, for an arbitrarily given $\epsilon > 0$, under Assumption 1, there exists a constant $K_0 > 0$, such that $\mathbb{E}\|\pi_k^\mu - \pi_k^\nu\|_2 < \frac{\epsilon}{6}$, for all initial distributions $\mu, \nu \in \mathcal{L}^2(\mathbb{R}^d)$, and $k \geq K_0 + 1$.

Notice that the additional Distance Assumption only make sense when we consider the approximation results for v_k and ρ_k, which are elements in the finite dimensional space V_{N_1}. Without the Distance Assumption, for this constant K_0 chosen above, we can still derive the following estimation of the distance between $\mathcal{N}(\tilde{u}_k)$ and π_k:

$$\mathbb{E}\left[\|\mathcal{N}(\tilde{u}_k) - \pi_k\|_2\right] < \frac{\epsilon}{6},\ \forall\ 1 \leq k \leq K_0, \qquad (65)$$

because the estimation (57) in the proof of Theorem 2 is independent of the Distance Assumption.

Now, for each $k \geq K_0 + 1$, let us consider the exact solution of the filtering problem, $\tilde{\pi} = \{\tilde{\pi}_j : k - K_0 \leq j \leq k\}$, on $[k - K_0, k]$, with initial value $\tilde{\pi}_{k-K_0} = \mathcal{N}(\tilde{u}_{k-K_0})$. According to the result (65), $\mathbb{E}\left[\|\mathcal{N}(\tilde{u}_k) - \tilde{\pi}_k\|_2\right] < \frac{\epsilon}{6}$.

In the meanwhile, since $\tilde{\pi}$ evolves according to the exact solution of the filter system, $\mathbb{E}\|\tilde{\pi}_k - \pi_k\|_2 < \frac{\epsilon}{6}$. Thus, for all $k \geq 1$, $\mathbb{E}\|\mathcal{N}(\tilde{u}_k) - \pi_k\|_2 \leq \mathbb{E}\|\mathcal{N}(\tilde{u}_k) - \tilde{\pi}_k\|_2 + \mathbb{E}\|\tilde{\pi}_k - \pi_k\|_2 < \frac{\epsilon}{3}$.

Step 2: Let $V_{N_1} = \mathrm{Span}\{\mathcal{I}_{K_0,N}\}$ be the finite dimensional space in the RNNSM, and N_1 is the dimension of V_{N_1}. In this step, we would like to find an element $\bar{w}_k \in V_{N_1}$ for each $k \geq 1$, such that $\mathbb{E}\|\mathcal{N}(\tilde{u}_k) - \bar{w}_k\|_2 < \epsilon$.

For $1 \leq k \leq K_0$, according to our construction of V_{N_1}, the function $\tilde{u}_k$ is inside the finite dimensional space V_{N_1} and we can take $\bar{w}_k = \mathcal{N}(\tilde{u}_k)$. For $k \geq K_0 + 1$, Step 1 shows that $\mathbb{E}\|\mathcal{N}(\tilde{u}_k) - \pi_k\|_2 < \frac{\epsilon}{3}$. In the meanwhile, $\mathbb{E}\|\pi_k - \pi_{K_0}\|_2 < \frac{\epsilon}{6}$, because we can regard π_k and π_{K_0} as the exact solution of the filtering problem at time K_0, with initial value π_{k-K_0+1} and π_1. Therefore,

$$\begin{aligned}
\mathbb{E}\|\mathcal{N}(\tilde{u}_k) - \mathcal{N}(\tilde{u}_{K_0})\|_2 &\leq \mathbb{E}\|\mathcal{N}(\tilde{u}_k) - \pi_k\|_2 + \\
\mathbb{E}\|\pi_k - \pi_{K_0}\|_2 &+ \mathbb{E}\|\pi_{K_0} - \mathcal{N}(\tilde{u}_{K_0})\|_2 < \epsilon.
\end{aligned}$$

Since $\tilde{u}_{K_0} \in V_{N_1}$ and $\mathcal{N}(\tilde{u}_{K_0})$ is only a scaling of $\tilde{u}_{K_0}$, we may take $\bar{w}_k = \mathcal{N}(\tilde{u}_{K_0})$ and we obtain the desired result. □

7 Conclusion

In this paper, we propose a novel RNNSM to numerically solving stable nonlinear filtering problems. Heuristically, the introduction of RNN allows us to construct a finite-dimensional space which inherits the structure of specific filtering systems and improve the approximation efficiency in comparison with classical spectral methods. The effectiveness of the proposed RNNSM is verified both theoretically and numerically. The thorough convergence analysis of RNNSM also serves as a mathematical interpretation of the deep learning procedure of RNN when dealing with temporal data.

As is shown in the construction of the finite-dimensional vector space V_{N_1}, the approximation efficiency of RNNSM depends on the efficiency of using vector space to approximate likelihood functions and the decay rate of the dependency of conditional distribution π_k on initial values.

The former one, according to the theory of Kolmogorov n-width, depends on the observation function $h(x)$. A possible direction for further studies may be the specific dependence of the subspace approximation efficiency (or the convergence rate of (11)) on the type of observation function h.

The latter one corresponds to the well-known 'curse of memory' when implementing recurrent neural network in practice [22]. Therefore, RNNSM is proposed only for filtering systems with additional stability properties or problems in finite time. In both cases, the target output of the recurrent neural network, i.e., the conditional expectations of the state process do not preserve long memory of the observations.

References

[1] E. Abedi, S. C. Surace, and J.-P. Pfister. A unification of weighted and unweighted particle filters. *SIAM Journal on Control and Optimization*, 60(2):597–619, 2022.

[2] J. Armstrong and D. Brigo. Stochastic PDE projection on manifolds: assumed-density and Galerkin filters. In *Geometric Science of Information, GSI 2015*, Lecture Notes in Computer Science, pages 713–722, 2015.

[3] R. Atar. Exponential stability for nonlinear filtering of diffusion processes in a noncompact domain. *Annals of Probability*, 26(4):1552–1574, 1998.

[4] R. Atar and O. Zeitouni. Exponential stability for nonlinear filtering. *Annales de L'Institut Henri Poincare-Probabilites et Statistiques*, 33(6):697–725, 1997.

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

[6] D. Brigo, B. Hanzon, and F. LeGland. A differential geometric approach to nonlinear filtering: the projection filter. *IEEE Transactions on Automatic Control*, 43(2):247–252, 1998.

[7] X. Chen, Y. Tao, W. Xu, and S. S.-T. Yau. Recurrent neural networks are universal approximators With stochastic inputs. *IEEE Transactions on Neural Networks and Learning Systems*, 34(10):7992–8006, 2023.

[8] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[9] Jimbo Henri Claver, Mbiazi Dave, and Shu Felix Che. Kalman filtering for stocks price prediction and control. *Journal of Computer Science*, 19(6):739–748, 2023.

[10] D. Crisan and B. Rozovskii. *The Oxford handbook of nonlinear filtering*. Oxford University Press, 1 edition, 2011.

[11] Dan Crisan and Arnaud Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on signal processing*, 50(3):736–746, 2002.

[12] W. Dong, X. Luo, and S. S.-T. Yau. Solving nonlinear filtering problems in real time by Legendre Galerkin spectral method. *IEEE Transactions on Automatic Control*, 66(4):1559–1572, 2021.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[14] M. Hazewinkel, S. I. Marcus, and H. J. Sussmann. Nonexistence of finite-dimensional filters for conditional statistics of the cubic sensor problem. *Systems & Control Letters*, 3(6):331–340, 1983.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] K Hornik, M Stinchcombe, and H White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[17] A. H. Jazwinski. *Stochastic processes and filtering theory*, volume 64 of *Mathematics in Science and Engineering*. Elsevier, 1970.

[18] L. Jiao, D. Wang, Y. Bai, P. Chen, and F. Liu. Deep learning in visual tracking: a review. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):5497–5516, 2023.

[19] H. J. Kushner and A. S. Budhiraja. A nonlinear filtering algorithm based on an approximation of the conditional distribution. *IEEE Transactions on Automatic Control*, 45(3):580–585, 2000.

[20] S. Li, Z. Wang, S. S. T. Yau, and Z. Zhang. Solving nonlinear filtering problems using a tensor train decomposition method. *IEEE Transactions on Automatic Control*, 68(7):4405–4412, 2023.

[21] Y. Li, S. Pa, and M. J. Coates. Invertible particle-flow-based sequential MCMC with extension to Gaussian mixture noise models. *IEEE Transactions on Signal Processing*, 67(9):2499–2512, 2019.

[22] Z. Li, J. Han, W. E, and Q. Li. On the curse of memory in recurrent neural networks: approximation and optimization analysis. *CoRR*, abs/2009.07799, 2020.

[23] J. T. H. Lo. Synthetic approach to optimal filtering. *IEEE Transactions on Neural Networks*, 5(5):803–811, 1994.

[24] X. Luo and S. S. T. Yau. Hermite spectral method to 1-d forward Kolmogorov equation and its application to nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 58(10):2495–2507, 2013.

[25] Y. Maday, A. T. Patera, and G. Turinici. A priori convergence theory for reduced-basis approximations of single-parameter

elliptic partial differential equations. *Journal of Scientific Computing*, 17(1-4):437–446, 2002.

[26] M. C. Maurel and D. Michel. Nonexistence results for a finite dimensional filter. *Stochastics*, 13(1-2):83–102, 1984.

[27] D. Ocone and E. Pardoux. Asymptotic stability of the optimal filter with respect to its initial condition. *SIAM Journal on Control and Optimization*, 34(1):226–243, 1996.

[28] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[29] A. Pinkus. *N-widths in approximation theory*. Springer-Verlag, 1 edition, 1985.

[30] A. M. Schaefer and H.-G. Zimmermann. Recurrent neural networks are universal approximators. *International Journal of Neural Systems*, 17(4):253–263, 2007.

[31] Z. Sun and S. S.-T. Yau. Solving nonlinear filtering problems with correlated noise based on Hermite-Galerkin spectral method. *Automatica*, 156, 2023.

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

[33] Lloyd N. Trefethen. *Spectral methods in MATLAB*, volume 10. Society for Industrial and Applied Mathematics, 2000.

[34] A. Trevisan and L. Palatella. On the Kalman filter error covariance collapse into the unstable subspace. *Nonlinear Processes in Geophysics*, 18(2):243–250, 2011.

[35] B. Van Bien and S. Rubenthaler. Stability of the optimal filter in continuous time: beyond the Benes filter. *Stochastic Analysis and Applications*, 38(5):797–855, 2020.

[36] Z. Wang, X. Luo, S. S.-T. Yau, and Z. Zhang. Proper orthogonal decomposition method to nonlinear filtering problems in medium-high dimension. *IEEE Transactions on Automatic Control*, 65(4):1613–1624, 2020.