Recurrent Neural Networks are Universal Approximators with Stochastic Inputs

Xiuqiong Chen, *Member, IEEE*, Yangtianze Tao, Wenjie Xu and Stephen S.-T. Yau, *Fellow, IEEE*

Dedication to Roger Brockett on the occasion of his 84th Birthday.

***Abstract*—In this paper, we investigate the approximation ability of Recurrent Neural Networks (RNNs) with stochastic inputs in state space model form. More explicitly, we prove that open dynamical systems with stochastic inputs can be well approximated by a special class of RNNs under some natural assumptions, and the asymptotic approximation error has also been delicately analyzed as time goes to infinity. In addition, as an important application of this result, we construct a RNN based filter and prove that it can well approximate finite dimensional filters which include Kalman filter and Beneš filter as special cases. The efficiency of RNN based filter has also been verified by two numerical experiments compared with optimal Kalman filter.**

***Index Terms*—Recurrent neural networks, Dynamical systems with stochastic inputs, Kalman filter, Finite dimensional filter.**

I. Introduction

Recurrent neural networks (RNNs) are able to learn features and long term dependencies from time-series data [1], [2]. In the foundational paper [1], Rumelhart et al. used back-propagation to train a neural network with one or two hidden layers, and Elman popularized simple RNNs (Elman network) in [2]. RNNs have various applications in many fields, such as language modeling [3], [4], speech recognition [5], [6], image processing [7], [8], machine translation [9] and so on. Major and recent advancements of RNNs including the challenging problems are reviewed on [10]. Despite the great successes of RNN in applications, the theoretical parts still need to be further investigated. In 2007, Schäfer and Zimmermann proved that open dynamical systems can be approximated by RNN in state space model form with an arbitrary accuracy [11], based on the universal approximation of feedforward neural networks which was proved in [12], [13], [14] using different methods.

In this work, we shall prove that open dynamical systems with stochastic inputs can be approximated by a class of RNNs in state space model form with an arbitrary accuracy. There are three significant differences between [11] and our work. The first is that we consider the more general *stochastic* dynamical systems rather than deterministic. The second is that [11] considers finite time horizon while we investigate the performance of RNN when time goes to infinity. The third one, which is also the most challenging one, is that all inputs of the maps in [11] are assumed to be in a compact subset, whereas the inputs are stochastic and can be unbounded on Euclidean space in our work.

As a significant application, finite dimensional filters (FDFs) can be formulated as a special class of dynamical systems with stochastic inputs. The nonlinear filtering problem involves estimating a stochastic process $\{x_k\}_{k\geq 0}$ (the state process) that cannot be observed directly, from the observations of a related process $\{y_k\}_{k\geq 0}$ (the observation process). This problem arises in many areas including target tracking, mathematical finance and communication. The goal of nonlinear filtering is to seek the conditional expectation $\mathbb{E}[x_k|y_j, 1\leq j\leq k]$, which can be completely determined by the conditional density $p(x_k|y_j, 1\leq j\leq k)$ based on the observation history $\{y_1,\cdots,y_k\}$. More introductions of nonlinear filtering can be found in the classic textbook [15].

The famous Kalman filter (KF) and Kalman-Bucy filter were proposed in 1960s [16], [17]. However, they need Gaussian and linear assumptions w.r.t. the system. Therefore, there spring up many works aiming to solve the nonlinear filtering problems, such as the extended Kalman filter (EKF) [15], unscented Kalman filter [18], ensemble Kalman filter [19], and particle filter (PF) [20]. Nonetheless, these filtering algorithms are suboptimal for general nonlinear systems and we are interested in a special class of systems which possess FDFs, i.e., we can obtain the conditional density $p(x_k|y_j, 1\leq j\leq k)$ by recursively computing a statistic of finite dimension using the observations. Historically, Kalman and Bucy first established the FDFs for linear filtering system with Gaussian initial distributions [16], [17]. Since then, there has been an intense interest in finding new classes of FDFs, such as Beneš filter [21], [22]. From 1990s, Yau and his collaborators have completely classified all finite dimensional estimation algebra of maximal rank [23], [24], [25], [26], [27], and constructed explicitly the so-called Yau filter which includes Kalman-Bucy filter and Beneš filter as special cases [25].

In this work, we formulate the FDF as the dynamical system with stochastic inputs. Therefore, one natural idea is to approximate FDFs by RNN, i.e., we can solve filtering problems by RNN as shown in Fig. 1. Actually, there already have existed some works about nonlinear filtering algorithms using neural networks. Lo proposed a neural filter using recurrent

Xiuqiong Chen is with School of Mathematics, Renmin University of China, Beijing, 100872, P. R. China, and the Yau Mathematical Sciences Center, Tsinghua University, Beijing, 100084, P. R. China, e-mail: cxq0828@ruc.edu.cn.

Yangtianze Tao is with the Department of Mathematical Sciences, Tsinghua University, Beijing, 100084, China, email: tytz19@mails.tsinghua.edu.cn.

Wenjie Xu is with the École polytechnique fédérale de Lausanne (EPFL), Lausanne, 1015, Switzerland, e-mail:wenjie.xu@epfl.ch.

Stephen S.-T. Yau is with the Department of Mathematical Sciences, Tsinghua University, Beijing, 100084, China, and Yanqi Lake Beijing Institute of Mathematical Sciences and Applications, Huairou district, Beijing, 101400, P. R. China, email: yau@uic.edu.

Stephen S.-T. Yau is corresponding author. Tel: +86-10-62787874.

Manuscript received ; revised .

multilayer perceptron and analyzed the estimation error when time is finite and all observations are in a compact set [28]. Parlos et al. presented some practical algorithms for adaptive state filtering using the framework of EKF and RNN [29]. [30] proposed a neural PF whose capability was demonstrated by numerical experiments. However, to our knowledge, the connections between FDF and RNN have not been investigated, as well as the accumulated error of neural filter when time goes to infinity. Following the work [31], we start to investigate the mathematical theory behind the neural filtering algorithms. In our RNN based filter, the inputs are observations and the outputs are the optimal estimates of the states, i.e., both inputs and outputs are random.

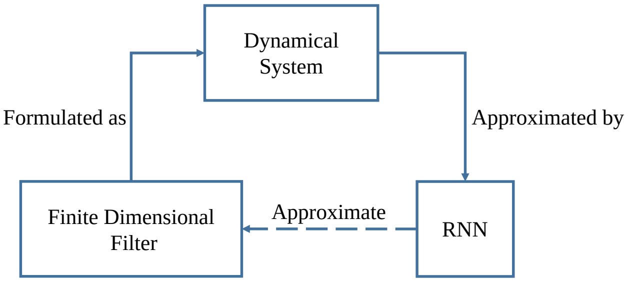


Fig. 1: The framework of this paper.

The motivation of this work is twofold. Firstly, as mentioned in the classic book [32], “Much as almost any function can be considered a feedforward neural network, essentially any function involving recurrence can be considered a recurrent neural network.” Therefore it is of great significance to investigate the approximation ability of RNN. We need to mention that the term “RNN” used here is not a specific type of RNN, and it is a more general framework as shown in Eq. (24). Secondly, FDF is a direct application of the universal approximation ability of RNN. Besides, FDF is the extension of KF and Beneš filter, on the other hand, the suboptimal filters can be regarded as the approximations of optimal filters by FDFs, such as EKF. Hence it is very meaningful to study the FDFs.

The main contributions of this work are listed as follows:

- We prove that an open dynamical system with stochastic possibly unbounded inputs can be well approximated by a specific class of RNN functions. More explicitly, we carefully analyze the approximation errors between the open dynamical system and RNN as time goes to infinity.
- We use a new viewpoint to formulate the FDFs, which include the classic KF as a special case. Using the concept of sufficient statistics, we express the FDF as a special class of dynamical systems with stochastic inputs, in which the system functions are unknown in most cases.
- Based on the previous two points, we construct a novel RNN based filter. Furthermore, for systems with FDFs, the L_1-error between the optimal estimate and the estimate by RNN filter can be arbitrarily small as time goes to infinity.

It needs to be pointed out that our work is for time-invariant filtering systems without delays and Markovian jump. The recently proposed asynchronous filtering scheme can be used to deal with Markovian jump systems subjected to time-varying delays and infinite distributed delays, and the filtering error system is exponentially stable in mean square and satisfies a given performance index simultaneously [33]. These two filters are used to deal with different filtering problems and both possess some kinds of stabilities.

This paper is organized as follows. In section II, we list some preliminary results about filtering problems, KF, FDFs and uniform integrability. Section III is devoted to present our first main result, i.e., any open dynamical systems can be well approximated by RNNs under some natural conditions. In section IV, we approximate the dynamical system of FDF by RNN and propose a novel RNN filter. The convergence of this new filter is also carefully analyzed. In section V, we show some numerical results which exhibit the efficiency of our algorithm. The conclusions are drawn in the last section.

II. Preliminaries

In this section, we list some preliminary knowledges. Some frequently used notations are introduced firstly. Then we give the general framework of filtering problems and introduce two special classes of filters subsequently, i.e., KF and FDF. At last, we introduce the uniform integrability which is the key part in the proofs of the main results.

A. Notations

For readers’ convenience, the notations used in this paper are summarized here.

We use $\mathcal{N}(m,P)$ to denote the Gaussian distribution with mean m and covariance P. The indicator function of a subset $A\subset\Omega$ is a function

$$\mathbb{1}_A:\ \Omega\to\{0,1\},$$

which is defined as

$$\mathbb{1}_A(x):=\left\{\begin{array}{ll}1 & \text{if } x\in A,\\ 0 & \text{if } x\notin A.\end{array}\right.$$

Since all norms on $\mathbb{R}^n$ are equivalent [34], without loss of generality, let $|\cdot|$ denote the 2-norm on $\mathbb{R}^n$. That is, for $\forall\ x=(x_1,\cdots,x_n)^{\mathrm{T}}\in\mathbb{R}^n$, $|x|:=\left(\sum_{i=1}^n|x_i|^2\right)^{1/2}$. Note that all the random variables are defined on the probability space $(\Omega,\mathscr{F},\mathbb{P})$. We set

$$L^p(\Omega;\mathbb{R}^n):=\{X(\omega):\Omega\to\mathbb{R}^n;X(\omega)\text{ is measurable and }\\ \mathbb{E}[|X(\omega)|^p]<\infty\},$$

and the norm on the space $L^p(\Omega;\mathbb{R}^n)$ is defined as $\|X\|_p:=\mathbb{E}^{1/p}\left[|X|^p\right]$, where $p=1,\ 2$.

We define the truncation operator $\mathcal{T}_K$ with level $K>0$ as

$$\mathcal{T}_K(x_i)=\left\{\begin{array}{ll}x_i & \text{if}\quad |x_i|\le K\\ K\cdot\text{sign}(x_i) & \text{otherwise}\ ,\end{array}\right. \tag{1}$$

and

$$\mathcal{T}_K(x):=(\mathcal{T}_K(x_1),\cdots,\mathcal{T}_K(x_n))^{\mathrm{T}} \tag{2}$$

for $x = (x_1, \cdots, x_n)^{\mathrm{T}} \in \mathbb{R}^n$. It can be easily checked that $\mathcal{T}_K x = x$ when $|x| \le K$, and $|\mathcal{T}_K x| \le |x|$ for all $x \in \mathbb{R}^n$. In addition,

$$\left\|\mathcal{T}_K X - \mathcal{T}_K \bar{X}\right\|_1 \le \left\|X - \bar{X}\right\|_1, \forall\ X, \bar{X} \in L^1(\Omega; \mathbb{R}^n), \quad (3)$$

and this property is proved in Appendix A.

B. Discrete filtering problems

The discrete time-invariant filtering system considered here is as follows:

$$\begin{cases} x_k = f(x_{k-1}) + g(x_{k-1})w_{k-1}, \\ y_k = h(x_k) + v_k, \end{cases} \quad (4)$$

where $x_k \in \mathbb{R}^n$ is the state at the discrete time instant k, $f : \mathbb{R}^n \to \mathbb{R}^n$ is the drift function, $g : \mathbb{R}^n \to \mathbb{R}^{n\times r}$ is the diffusion function, $y_k \in \mathbb{R}^m$ is the observation or the measurement of the system, $h : \mathbb{R}^n \to \mathbb{R}^m$ is the observation function, $\{w_k \in \mathbb{R}^r,\ k = 0, 1, \cdots\}$ and $\{v_k \in \mathbb{R}^m,\ k = 1, \cdots\}$ are Gaussian white noise processes with $w_k \sim \mathcal{N}(0, Q)$ and $v_k \sim \mathcal{N}(0, R)$. Here we need to assume that $\{w_k, k = 0, 1, \cdots\}$, $\{v_k, k = 1, \cdots\}$ and the initial state x_0 are independent of each other. We use Y_k to denote the history of the observations up to time instant k, i.e.,

$$Y_k := \{y_1, \cdots, y_k\}. \quad (5)$$

The aim of the filtering problem is to obtain the *optimal* estimate of state x_k based on the observation history Y_k. Here, *optimal* means that the estimate can minimize the mean square error and the rigorous definition is as follows:

Definition 1 (Minimum mean square error estimate ([15])). *Let $\hat{x}$ be an estimate of random variable x. Then the minimum mean square error estimate of x is*

$$\arg\min_{\hat{x}} \mathbb{E}[(x - \hat{x})^{\mathrm{T}}(x - \hat{x})].$$

The minimum mean square error estimate of state x_k based on Y_k is given by the following theorem.

Theorem 1 (Theorem 5.3 in [15]). *Let the estimate of x_k be a functional on Y_k. Then the minimum mean square error estimate of state x_k is its conditional mean $\mathbb{E}[x_k|Y_k]$.*

Apparently, if we can obtain the conditional density of x_k based on Y_k, i.e., $p(x_k|Y_k)$, then we can simply compute $\mathbb{E}[x_k|Y_k]$. However, for the general nonlinear filtering system Eq. (4), we cannot obtain $p(x_k|Y_k)$ by solving finite ordinary differential equations, except for some special cases, such as linear Gaussian systems, which can be solved by KF.

C. Kalman filter

In this part, we shall introduce the KF for linear Gaussian systems, i.e., f, g and h in Eq. (4) are linear functions, and the distribution of the initial state x_0 is Gaussian. More explicitly, we consider the following special case of system Eq. (4):

$$\begin{cases} x_k = Fx_{k-1} + Gw_{k-1}, \\ y_k = Hx_k + v_k, \end{cases} \quad (6)$$

where the initial state x_0 is Gaussian, and $\{w_k, k = 0, 1, \cdots\}$ and $\{v_k, k = 1, \cdots\}$ are two independent white Gaussian sequences that are also independent of the initial state x_0 jointly.

It is well known that the conditional density function $p(x_k|Y_k)$ is Gaussian for system Eq. (6) and it can be determined by the conditional mean and covariance. Let us denote the conditional means as

$$m_{k|k-1} := \mathbb{E}[x_k|Y_{k-1}],\ m_{k|k} := \mathbb{E}[x_k|Y_k], \quad (7)$$

the conditional covariances as

$$\begin{aligned} P_{k|k-1} &:= \mathbb{E}\left[(x_k - m_{k|k-1})(x_k - m_{k|k-1})^{\mathrm{T}}|Y_{k-1}\right], \\ P_{k|k} &:= \mathbb{E}\left[(x_k - m_{k|k})(x_k - m_{k|k})^{\mathrm{T}}|Y_k\right], \end{aligned} \quad (8)$$

and their evolution equations are given by KF in two iterative steps. Assume the distribution of the initial state x_0 is $p(x_0) = \mathcal{N}(m_{0|0}, P_{0|0})$. For $k = 1, 2, 3, \cdots$,

1) Prediction: given $m_{k-1|k-1}$ and $P_{k-1|k-1}$, we obtain $m_{k|k-1}$ and $P_{k|k-1}$ by

$$\begin{cases} m_{k|k-1} = Fm_{k-1|k-1}, \\ P_{k|k-1} = GQG^{\mathrm{T}} + FP_{k-1|k-1}F^{\mathrm{T}}. \end{cases} \quad (9)$$

2) Updating: when the latest observation y_k arrives, $m_{k|k}$ and $P_{k|k}$ are obtained by

$$\begin{cases} m_{k|k} = m_{k|k-1} + P_{k|k-1}H^{\mathrm{T}}(HP_{k|k-1}H^{\mathrm{T}} + R)^{-1}, \\ \qquad\qquad \cdot (y_k - Hm_{k|k-1}), \\ P_{k|k} = P_{k|k-1} - P_{k|k-1}H^{\mathrm{T}}(HP_{k|k-1}H^{\mathrm{T}} + R)^{-1} \\ \qquad\qquad \cdot HP_{k|k-1}. \end{cases} \quad (10)$$

It is apparently that the Gaussian conditional density $p(x_k|Y_k)$ is totally determined by the conditional mean $m_{k|k}$ and covariance $P_{k|k}$, so we put them together into a vector $s_{k|k}$, which is defined as follows:

$$s_{k|k} := [m_{k|k}^{\mathrm{T}}, \mathrm{vec}^{\mathrm{T}}(P_{k|k})]^{\mathrm{T}},\ \forall\ k \ge 0, \quad (11)$$

where $\mathrm{vec}(\circ_{n_1\times n_2})$ is the $n_1 n_2 \times 1$ column vector obtained by stacking the columns of the matrix $\circ$ on top of one another. Then we can easily have

$$s_{k|k} = \varphi(s_{k-1|k-1}, y_k), \quad (12)$$

where φ is determined by Eq. (9) and Eq. (10).

Obviously, $p(x_k|Y_k)$ can be completely determined by $s_{k|k}$, we call $s_{k|k}$ *sufficient statistic* and its definition is given as follows:

Definition 2 (Sufficient Statistic ([21])). *If the conditional distribution $p(x_k|Y_k)$ can be completely determined by a vector-valued function $s_{k|k} \in \mathbb{R}^{n_s}$ of the observation sequence Y_k, where $n_s \in \mathbb{N}$, then we say $s_{k|k}$ is a* sufficient statistic *for $p(x_k|Y_k)$.*

Hence, there exists a function $\gamma : \mathbb{R}^{n_s} \to \mathbb{R}^n$, such that

$$\mathbb{E}[x_k|Y_k] = \gamma(s_{k|k}), \quad (13)$$

since the optimal estimate $\mathbb{E}[x_k|Y_k]$ is determined by $p(x_k|Y_k)$, which is completely determined by the sufficient statistic $s_{k|k}$.

Similarly, when the noises $\{w_k\}$ and $\{v_k\}$ in Eq. (6) are correlated, we can also have Eq. (12)-Eq. (13), and more details can be found in Appendix C.

D. Finite dimensional filter

Naturally, we can generalize Eq. (12) and Eq. (13) to any filtering systems with finite statistics, i.e., the filtering systems with *FDFs*, which include KF and Beneš filter [21] as special cases. For instance, in KF, the conditional density function is determined by the conditional mean and covariance for linear Gaussian systems. In [21], it is proved that for a class of special filtering systems, the unnormalized conditional density can be written explicitly in terms of just 10 sufficient statistics satisfying a matrix-vector equation.

Similarly, we use vector $S_{k|k}$ to denote the finite dimensional sufficient statistics of the posterior distribution $p(x_k|Y_k)$. The evolution function of the statistics are denoted as Φ, and the map from $S_{k|k}$ to conditional mean $\mathbb{E}[x_k|Y_k]$ is denoted as Γ, i.e.,

$$S_{k|k} = \Phi(S_{k-1|k-1}, y_k), \tag{14}$$
$$\mathbb{E}[x_k|Y_k] = \Gamma(S_{k|k}). \tag{15}$$

As we know, in most cases, it is not easy to write down the explicit forms of the map functions Φ and Γ. However, by taking advantage of neural networks, we can approximate these functions just using the input and output data, which motivates us to use neural networks to solve the FDF problems.

E. Uniform integrability

Before we start the analysis of RNN, we need to introduce an important concept, i.e., uniform integrability.

Definition 3. *([35]) A collection of random variables $\{X_i \in \mathbb{R},\ i \in I\}$ in $L^1(\Omega;\mathbb{R})$ is said to be uniformly integrable if*

$$\lim_{M\to+\infty}\left(\sup_{i\in I}\mathbb{E}\left[|X_i|\,\mathbb{1}_{|X_i|>M}\right]\right) = 0.$$

Similarly, this definition can be extended to random vectors.

Definition 4. *A collection of random vectors $\{X_i \in \mathbb{R}^n,\ i \in I\}$ in $L^1(\Omega;\mathbb{R}^n)$ is said to be uniformly integrable if*

$$\lim_{M\to+\infty}\left(\sup_{i\in I}\mathbb{E}\left[|X_i|\,\mathbb{1}_{|X_i|>M}\right]\right) = 0. \tag{16}$$

A common way to check the uniform integrability is listed in the following lemma.

Lemma 1. *([35]) Let $\{X_i \in \mathbb{R}^n,\ i \in I\}$ be a collection of random vectors. If*

$$\sup_{i\in I}\mathbb{E}\left[|X_i|^p\right] < \infty, \quad \textit{for some } p > 1, \tag{17}$$

then $\{X_i\ \ i \in I\}$ is uniformly integrable.

Following Lemma 1, we can obtain the following two useful results which will be used in the subsequent sections.

Lemma 2. *Assume a collection of random vectors $\{X_i \in \mathbb{R}^n,\ i \in I\}$ is uniformly integrable. Then for any $\varepsilon > 0$, there exists a positive $K > 0$, such that*

$$\sup_{i\in I}\|X_i - \mathcal{T}_K X_i\|_1 < \varepsilon, \tag{18}$$

where the truncation operator $\mathcal{T}_K$ is defined in Eq. (2).

Proof. Since $\{X_i : i \in I\}$ is uniformly integrable, i.e.,

$$\lim_{M\to+\infty}\left(\sup_{i\in I}\mathbb{E}[|X_i|\,\mathbb{1}_{|X_i|>M}]\right) = 0, \tag{19}$$

there exists $K > 0$, such that

$$\sup_{i\in I}\mathbb{E}[|X_i|\,\mathbb{1}_{|X_i|>K}] < \frac{\varepsilon}{2}. \tag{20}$$

Then we have

$$\begin{aligned}
&\sup_{i\in I}\|X_i - \mathcal{T}_K X_i\|_1 \\
\le&\sup_{i\in I}\mathbb{E}\left[|X_i - \mathcal{T}_K X_i|\,\mathbb{1}_{|X_i|\le K}\right] \\
&+\sup_{i\in I}\mathbb{E}\left[|X_i - \mathcal{T}_K X_i|\,\mathbb{1}_{|X_i|>K}\right] \\
=&0+\sup_{i\in I}\mathbb{E}\left[|X_i - \mathcal{T}_K X_i|\,\mathbb{1}_{|X_i|>K}\right] \\
\le&\sup_{i\in I}\left(\mathbb{E}\left[|X_i|\,\mathbb{1}_{|X_i|>K}\right]+\mathbb{E}\left[|\mathcal{T}_K X_i|\,\mathbb{1}_{|X_i|>K}\right]\right) \\
\le&2\mathbb{E}\left[|X_i|\,\mathbb{1}_{|X_i|>K}\right] \\
<&\varepsilon.
\end{aligned}$$

□

Remark 1. *According to Lemma 2, it is known that we can find a sufficiently large cube, such that most of the densities of the uniformly integrable random vectors fall in this bounded set. In other words, if $\{X_i \in \mathbb{R}^n,\ i \in I\}$ is uniformly integrable, then we can choose a sufficient large $K > 0$, such that, uniformly over $X_i \in \{X_i \in \mathbb{R}^n,\ i \in I\}$, the random vector $\mathcal{T}_K X_i$ is a good approximation of X_i in terms of the L_1-norm. Crucially, every $\mathcal{T}_K X_i$ is a bounded random vector, which is the desired property allowing us to approximate functions in RNN with infinite time steps.*

Combing Lemma 1 and Lemma 2, we can easily obtain the following lemma.

Lemma 3. *Assume that a collection of random vectors $X_i \in \mathbb{R}^n,\ i \in I$, satisfy $\sup_{i\in I}\|X_i\|_2 < \infty$. Then for any $\varepsilon > 0$, there exists a positive $K > 0$, such that*

$$\sup_{i\in I}\|X_i - \mathcal{T}_K X_i\|_1 < \varepsilon, \tag{21}$$

where the truncation operator $\mathcal{T}_K$ is defined in Eq. (2).

Proof. It is apparent that

$$\sup_{i\in I}\mathbb{E}\left[|X_i|^2\right] < \infty.$$

Then according to Lemma 1, we know that, $\{X_i,\ i \in I\}$ is uniformly integrable. Using Lemma 2, we obtain the desired result.

□

III. Universal approximation of RNN with stochastic inputs

In this section, we shall investigate the universal approximation ability of RNN with stochastic inputs. First of all, we need to introduce the feedforward networks whose approximation ability will be used in our analysis. Then we give a class of systems which can be approximated by RNN. Furthermore, the accumulated error is also delicately analyzed.

A. Feedforward network

Now we revisit some well known results of feedforward networks presented in [14]. To begin with, we need to define several classes of functions precisely.

Definition 5. *For any $r \in \mathbb{N} \equiv \{1, 2, \cdots\}$, $\mathbf{A}^r$ is the set of all affine functions from $\mathbb{R}^r$ to $\mathbb{R}$, i.e.,*

$$\mathbf{A}^r := \left\{A(x) = w^{\mathrm{T}}x + b : w, x \in \mathbb{R}^{r\times 1}, b \in \mathbb{R}\right\}. \tag{22}$$

In the feedforward network, x, w and b represent the input, weight and bias of the network, respectively. And $A(x)$ is the linear operator in feedforward networks.

Definition 6. *A function $\kappa : \mathbb{R} \to [0, 1]$ is a squashing function if it is non-decreasing, $\lim_{\lambda\to+\infty} \kappa(\lambda) = 1$, and $\lim_{\lambda\to-\infty} \kappa(\lambda) = 0$.*

Here, κ represents the activation function.

Definition 7. *$\Sigma^r(\kappa)$ be the class of functions*

$$\{\bar{\zeta} : \mathbb{R}^r \to \mathbb{R} : \bar{\zeta}(x) = \sum_{j=1}^{q} \beta_j \kappa(A_j(x)), x \in \mathbb{R}^r, \beta_j \in \mathbb{R},$$
$$A_j \in \mathbf{A}^r, q = 1, 2, \cdots\}.$$

Apparently, $\bar{\zeta}$ represents the standard three-layered feedforward network with r input-neurons, q hidden-neurons and one output-neuron. It is well-known that this class of feedforward network functions are capable to approximate any continuous function over a compact set to any desired degree of accuracy. Let $\mathcal{C}^r$ be the set of continuous functions from $\mathbb{R}^r$ to $\mathbb{R}$. We now state the universal approximation theorem of feedforward neural network.

Theorem 2 (Universal Approximation of Multilayer Feedforward Networks ([14]))**.** *For every squashing function κ, every $r \in \mathbb{N}$, $\Sigma^r(\kappa)$ is uniformly dense on compacta in $\mathcal{C}^r$, i.e., for every compact subset $S \subset \mathbb{R}^n$, $\Sigma^r(\kappa)$ is ρ_S-dense in $\mathcal{C}^r$, where for $f, g \in \mathcal{C}^r$, $\rho_S(f, g) := \sup_{x\in S} |f(x) - g(x)|$.*

This theorem tells us that, standard feedforward networks with only a single hidden layer can approximate any continuous function uniformly on any compact set.

Naturally, Theorem 2 can be extended to the approximation of vector-valued functions. Let $\mathcal{C}^{r,N}$ be the set of continuous functions from $\mathbb{R}^r$ to $\mathbb{R}^N$, and $\Sigma^{r,N}(\kappa)$ be the class of functions

$$\{\bar{\zeta} = (\bar{\zeta}_1, \cdots, \bar{\zeta}_N)^{\mathrm{T}} : \mathbb{R}^r \to \mathbb{R}^N : \bar{\zeta}_l(x) = \sum_{j=1}^{q} \beta_{l,j} \kappa(A_j(x)),$$
$$x \in \mathbb{R}^r, \beta_{l,j} \in \mathbb{R}, A_j \in \mathbf{A}^r, 1 \le l \le N, q = 1, 2, \cdots\}.$$

Then we have the following corollary.

Corollary 1. *([11]) Theorem 2 holds for the approximation of functions in $\mathcal{C}^{r,N}$ by the extended function class $\Sigma^{r,N}(\kappa)$. Thereby the metric $\rho_S^N(f, g) := \sup_{x\in S} \sum_{l=1}^{N} |f_l(x) - g_l(x)|$.*

B. Open dynamical systems and RNNs

While feedforward networks can be used to approximate continuous functions in compact set, RNN can be mapped to an open dynamical system with sequential external inputs [11], which is shown in Fig. 2.

An open dynamical system in discrete time can be represented by the following equations:

$$\begin{cases} s_{k+1} = \eta(s_k, \alpha_{k+1}), & \text{state transition} \\ \beta_k = \xi(s_k), & \text{output equation} \end{cases} \tag{23}$$

where α_k is the stochastic external input, s_k is the state and β_k is the observable output for $\forall\ k \ge 1$.

Now we aim to approximate the open dynamical system Eq. (23) with stochastic inputs by a class of RNNs. More explicitly, we investigate $RNN^{r_1,r_2,r_3}(\kappa)$, which is defined as follows:

Definition 8. *For any squashing function κ, and $r_1, r_2, r_3 \in \mathbb{N}$, $RNN^{r_1,r_2,r_3}(\kappa)$ is a class of functions with the following state space model form:*

$$\begin{cases} \tilde{s}_{k+1} = \tilde{\eta}(\tilde{s}_k, \alpha_{k+1}), \\ \tilde{\beta}_k = \tilde{\xi}(\tilde{s}_k), \end{cases} \tag{24}$$

where $\alpha_k \in \mathbb{R}^{r_1}$ is the input, $\tilde{s}_k \in \mathbb{R}^{r_2}$ is the hidden state, $\tilde{\beta}_k \in \mathbb{R}^{r_3}$ is the output, and

$$\tilde{\eta}(\tilde{s}, \alpha) = \bar{\eta}(\mathcal{T}_{K^s}\tilde{s}, \mathcal{T}_{K^\alpha}\alpha), \tag{25}$$
$$\tilde{\xi}(\tilde{s}) = \bar{\xi}(\mathcal{T}_{K^s}\tilde{s}), \tag{26}$$

in which $\bar{\eta} \in \Sigma^{r_1+r_2,r_2}(\kappa)$, $\bar{\xi} \in \Sigma^{r_2,r_3}(\kappa)$, K^s and K^α are two positive numbers which are the parameters of the RNN, and $\mathcal{T}$ is the truncation operator defined in Eq. (2).

It can be seen that, compared with the standard RNN, we use truncations in the inputs of $\bar{\eta}$ and $\bar{\xi}$ in Eq. (25)-Eq. (26). This is because that the inputs α and the hidden state $\tilde{s}$ can be unbounded instead of in some compact sets. We aim to approximate $\tilde{s}$ and α by bounded $\mathcal{T}_{K^s}\tilde{s}$ and $\mathcal{T}_{K^\alpha}\alpha$, respectively.

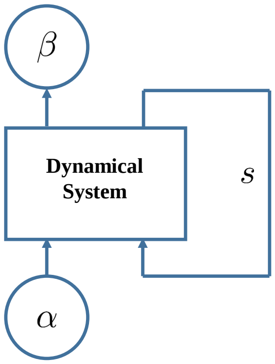


Fig. 2: Open dynamical system with external input α, state s and output β.

The framework of RNN is displayed in Fig. 3 [32], which is similar to the open dynamical system as shown in Fig. 2. In the following theorem, we shall prove that the open dynamical system Eq. (23) can be approximated by functions in $RNN^{r_1,r_2,r_3}(\kappa)$ with arbitrary accuracy.

Theorem 3. *(Universal Approximation Theorem for RNN with Stochastic Inputs) Let $\eta(\cdot):\mathbb{R}^{r_2}\times\mathbb{R}^{r_1}\to\mathbb{R}^{r_2}$ and $\xi(\cdot):\mathbb{R}^{r_2}\to\mathbb{R}^{r_3}$ be continuous, the external stochastic inputs $\alpha_k\in\mathbb{R}^{r_1}$, the inner state $s_k\in\mathbb{R}^{r_2}$, and the output $\beta_k\in\mathbb{R}^{r_3}$, $k=1,2,\cdots$. For any open dynamical system of the form*

$$\begin{cases} s_{k+1}=\eta(s_k,\alpha_{k+1}), \\ \beta_k=\xi(s_k), \end{cases} \tag{27}$$

if the following conditions hold:

- *$\{\alpha_k,k\geq 1\}$ and $\{s_k,k\geq 1\}$ are uniformly integrable;*
- *for $\forall\ s,\bar{s}\in L^1(\Omega;\mathbb{R}^{r_2})$ and $\forall\ \alpha,\bar{\alpha}\in L^1(\Omega;\mathbb{R}^{r_1})$, $\|\eta(s,\alpha)-\eta(\bar{s},\bar{\alpha})\|_1\leq C_{\eta 1}\|s-\bar{s}\|_1+C_{\eta 2}\|\alpha-\bar{\alpha}\|_1$, and the Lipschitz constant $C_{\eta 1}$ satisfies $|C_{\eta 1}|<1$;*
- *for $\forall\ \epsilon>0$, there exists $\delta>0$, such that for any $s,\bar{s}\in L^1(\Omega;\mathbb{R}^{r_2})$ satisfying $\|s-\bar{s}\|_1<\delta$, we have $\|\xi(s)-\xi(\bar{s})\|_1<\epsilon$,*

then Eq. (27) can be approximated by the functions in $RNN^{r_1,r_2,r_3}(\kappa)$ with an arbitrary accuracy, i.e., for $\forall\ \varepsilon>0$, there exist functions $\tilde{\eta}$ and $\tilde{\xi}$ of forms Eq. (25)-Eq. (26), which determine the RNN system Eq. (24) with the same input $\{\alpha_k,k\geq 1\}$ of Eq. (27), such that

$$\begin{aligned} &\overline{\lim_{k\to\infty}}\|s_k-\tilde{s}_k\|_1<\varepsilon, \\ &\overline{\lim_{k\to\infty}}\left\|\beta_k-\tilde{\beta}_k\right\|_1<\varepsilon, \end{aligned} \tag{28}$$

where $\tilde{s}_k$ and $\tilde{\beta}_k$ are the state and output of the RNN system Eq. (24), respectively.

Proof. The theorem is proved in three steps. We first construct appropriate approximated RNN functions using the universal approximation of multilayer feedforward networks. Then we try to obtain the iterative inequalities for errors. And at last, we compute the upper bounds of the accumulated errors.

Step 1: In this step, we will construct functions in $RNN^{r_1,r_2,r_3}(\kappa)$ to approximate system Eq. (27).

Since $\{\alpha_k,k\geq 1\}$ and $\{s_k,k\geq 1\}$ are uniformly integrable, for $\forall\ \varepsilon_1>0$, we can find $K_1>0$ and $K_2>0$, such that

$$\begin{aligned} &\sup_{k\geq 1}\|s_k-\mathcal{T}_{K_1}s_k\|_1<\varepsilon_1, \\ &\sup_{k\geq 1}\|\alpha_k-\mathcal{T}_{K_2}\alpha_k\|_1<\varepsilon_1, \end{aligned} \tag{29}$$

according to Lemma 2. Let

$$B_1:=\{x=(x_1,\cdots,x_{r_2})^{\mathrm{T}}\in\mathbb{R}^{r_2}:\ |x_i|\leq K_1, 1\leq i\leq r_2\},$$
$$B_2:=\{x=(x_1,\cdots,x_{r_1})^{\mathrm{T}}\in\mathbb{R}^{r_1}:\ |x_i|\leq K_2, 1\leq i\leq r_1\}.$$

Observing B_1 and B_2 are compact sets, and by Corollary 1, we know that, for $\forall\ \varepsilon_2>0$, there exists functions $\bar{\eta}\in\Sigma^{r_1+r_2,r_2}$ and $\bar{\xi}\in\Sigma^{r_2,r_3}$ represented by feedforward networks, such that

$$\begin{aligned} &\sup_{s\in B_1}\left|\xi(s)-\bar{\xi}(s)\right|<\varepsilon_2, \\ &\sup_{s\in B_1,\alpha\in B_2}|\eta(s,\alpha)-\bar{\eta}(s,\alpha)|<\varepsilon_2. \end{aligned} \tag{30}$$

Set

$$\begin{aligned} \tilde{\xi}(s)&:=\bar{\xi}(\mathcal{T}_{K_1}s), \\ \tilde{\eta}(s,\alpha)&:=\bar{\eta}(\mathcal{T}_{K_1}s,\mathcal{T}_{K_2}\alpha). \end{aligned} \tag{31}$$

Step 2: Define $e_k:=\|s_k-\tilde{s}_k\|_1$, where $\tilde{s}_k$ is the state of system Eq. (24) with $\tilde{\eta}$ and $\tilde{\xi}$ defined in Eq. (31). Now we derive the evolution equation of the error e_k.

Comparing Eq. (24) and Eq. (27), we have

$$\begin{aligned} e_{k+1}&=\|s_{k+1}-\tilde{s}_{k+1}\|_1 \\ &=\|\eta(s_k,\alpha_{k+1})-\tilde{\eta}(\tilde{s}_k,\alpha_{k+1})\|_1 \\ &=\|\eta(s_k,\alpha_{k+1})-\bar{\eta}(\mathcal{T}_{K_1}\tilde{s}_k,\mathcal{T}_{K_2}\alpha_{k+1})\|_1 \\ &\leq\|\eta(s_k,\alpha_{k+1})-\eta(\mathcal{T}_{K_1}s_k,\mathcal{T}_{K_2}\alpha_{k+1})\|_1 \\ &\quad+\|\eta(\mathcal{T}_{K_1}s_k,\mathcal{T}_{K_2}\alpha_{k+1})-\eta(\mathcal{T}_{K_1}\tilde{s}_k,\mathcal{T}_{K_2}\alpha_{k+1})\|_1 \\ &\quad+\|\eta(\mathcal{T}_{K_1}\tilde{s}_k,\mathcal{T}_{K_2}\alpha_{k+1})-\bar{\eta}(\mathcal{T}_{K_1}\tilde{s}_k,\mathcal{T}_{K_2}\alpha_{k+1})\|_1 \\ &\triangleq\Pi_1+\Pi_2+\Pi_3. \end{aligned} \tag{32}$$

Now we analyze these three terms separately. As for Π_1, we have

$$\begin{aligned} \Pi_1&=\|\eta(s_k,\alpha_{k+1})-\eta(\mathcal{T}_{K_1}s_k,\mathcal{T}_{K_2}\alpha_{k+1})\|_1 \\ &\leq C_{\eta 1}\|s_k-\mathcal{T}_{K_1}s_k\|_1+C_{\eta 2}\|\alpha_{k+1}-\mathcal{T}_{K_2}\alpha_{k+1}\|_1 \\ &<(C_{\eta 1}+C_{\eta 2})\,\varepsilon_1, \end{aligned} \tag{33}$$

where the first inequality is due to the second condition and the second inequality comes from Eq. (29). In terms of Π_2, using the Lipschitz property of η and Eq. (3), we have

$$\begin{aligned} \Pi_2&=\|\eta(\mathcal{T}_{K_1}s_k,\mathcal{T}_{K_2}\alpha_{k+1})-\eta(\mathcal{T}_{K_1}\tilde{s}_k,\mathcal{T}_{K_2}\alpha_{k+1})\|_1 \\ &\leq C_{\eta 1}\|\mathcal{T}_{K_1}s_k-\mathcal{T}_{K_1}\tilde{s}_k\|_1 \\ &\leq C_{\eta 1}e_k. \end{aligned} \tag{34}$$

As for Π_3, according to the second inequality in Eq. (30), we know that

$$\Pi_3=\|\eta(\mathcal{T}_{K_1}\tilde{s}_k,\mathcal{T}_{K_2}\alpha_{k+1})-\bar{\eta}(\mathcal{T}_{K_1}\tilde{s}_k,\mathcal{T}_{K_2}\alpha_{k+1})\|_1<\varepsilon_2, \tag{35}$$

since $\mathcal{T}_{K_1}\tilde{s}_k\in B_1$ and $\mathcal{T}_{K_2}\alpha_{k+1}\in B_2$. Put Eq. (33)-Eq. (35) into Eq. (32), we can obtain

$$e_{k+1}<C_{\eta 1}e_k+(C_{\eta 1}+C_{\eta 2})\,\varepsilon_1+\varepsilon_2. \tag{36}$$

Step 3: Now we analyze the accumulated errors. Using Eq. (36) repeatedly, it follows that

$$\begin{aligned} e_{k+1}&<C_{\eta 1}e_k+(C_{\eta 1}+C_{\eta 2})\,\varepsilon_1+\varepsilon_2 \\ &<C_{\eta 1}^2e_{k-1}+(C_{\eta 1}+1)\left((C_{\eta 1}+C_{\eta 2})\,\varepsilon_1+\varepsilon_2\right) \\ &\ \ \vdots \\ &<C_{\eta 1}^ke_1+\left((C_{\eta 1}+C_{\eta 2})\,\varepsilon_1+\varepsilon_2\right)\sum_{i=0}^{k-1}C_{\eta 1}^i \\ &=C_{\eta 1}^ke_1+\frac{C_{\eta 1}^k-1}{C_{\eta 1}-1}\left((C_{\eta 1}+C_{\eta 2})\,\varepsilon_1+\varepsilon_2\right). \end{aligned} \tag{37}$$

Thus we have

$$\overline{\lim_{k\to\infty}}\,e_k\leq\frac{1}{1-C_{\eta 1}}\left((C_{\eta 1}+C_{\eta 2})\,\varepsilon_1+\varepsilon_2\right) \tag{38}$$

once the condition $|C_{\eta 1}|<1$ holds.

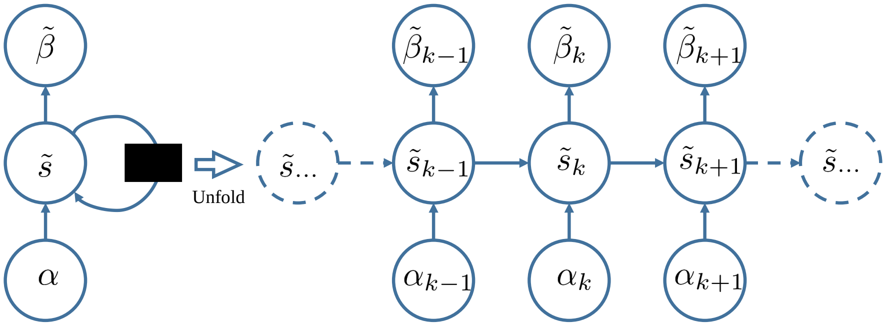


Fig. 3: Recurrent neural networks with input α, hidden state $\tilde{s}$ and output $\tilde{\beta}$.

Based on the third condition, we know that for $\forall\, \varepsilon > 0$, there exists $\delta > 0$, such that for any $s, \bar{s} \in L^2(\Omega; \mathbb{R}^{r_2})$ satisfying $\|s - \bar{s}\|_1 < \delta$, we have $\|\xi(s) - \xi(\bar{s})\|_1 < \varepsilon/6$. Apparently, we can choose small enough ε_1 and ε_2, so that

$$\begin{cases} \overline{\lim_{k\to\infty}} e_k \leq \dfrac{1}{1 - C_{\eta 1}} \left((C_{\eta 1} + C_{\eta 2})\, \varepsilon_1 + \varepsilon_2 \right) < \min\left\{ \varepsilon, \dfrac{\delta}{2} \right\}, \\ \|s_k - \mathcal{T}_{K_1} s_k\|_1 < \varepsilon_1 < \delta, \\ \sup\limits_{s \in B_1} \left| \xi(s) - \bar{\xi}(s) \right| < \varepsilon_2 < \varepsilon/6 \end{cases} \tag{39}$$

based on Eq. (29) and the first inequality in Eq. (30). It follows that there exists $N_0 > 0$, such that

$$e_k = \|s_k - \tilde{s}_k\|_1 < \delta,\ \forall\ k \geq N_0.$$

Therefore, for any $k \geq N_0$, we have

$$\begin{aligned} &\left\| \beta_k - \tilde{\beta}_k \right\|_1 \\ =& \left\| \xi(s_k) - \tilde{\xi}(\tilde{s}_k) \right\|_1 \\ =& \left\| \xi(s_k) - \bar{\xi}(\mathcal{T}_{K_1} \tilde{s}_k) \right\|_1 \\ \leq& \|\xi(s_k) - \xi(\mathcal{T}_{K_1} s_k)\|_1 + \|\xi(\mathcal{T}_{K_1} s_k) - \xi(\mathcal{T}_{K_1} \tilde{s}_k)\|_1 \\ &+ \left\| \xi(\mathcal{T}_{K_1} \tilde{s}_k) - \bar{\xi}(\mathcal{T}_{K_1} \tilde{s}_k) \right\|_1 \\ <& \varepsilon/6 + \varepsilon/6 + \varepsilon/6 = \varepsilon/2, \end{aligned} \tag{40}$$

since $\|s_k - \mathcal{T}_{K_1} s_k\|_1 < \delta$ and $\|\mathcal{T}_{K_1} s_k - \mathcal{T}_{K_1} \tilde{s}_k\|_1 < \delta$. Then

$$\overline{\lim_{k\to\infty}} \left\| \beta_k - \tilde{\beta}_k \right\|_1 < \varepsilon. \tag{41}$$

It is obvious that we obtain the desired results from the first inequality of Eq. (39) and Eq. (41). □

Remark 2. *As for the three conditions in Theorem 3, we have the following discussions.*

- *In terms of the first condition, if* $\sup_{k\geq 1} \mathbb{E}\left[|s_k|^{p_1}\right] < \infty$ *and* $\sup_{k\geq 1} \mathbb{E}\left[|\alpha_k|^{p_2}\right] < \infty$*, for some* $p_1, p_2 > 1$*, then by Lemma 1, we know that this condition is satisfied. We put this condition since we need to find a big enough high dimensional cube, which can capture most of the densities of all the input and state random vectors. Then we can approximate the functions on the bounded domain using the approximation ability of feedforward networks, and neglect the unbounded parts. This is why we can approximate functions on the whole space.*
- *The second condition implies that the system Eq. (27) is stable [36], which is natural and useful in practice. This condition is used to ensure that the accumulated error will not blow up.*
- *The third condition means that* ξ *is continuous in the given metric space. So that we can estimate the approximation error of the outputs from the approximation error of the hidden state.*

Now we give an example which satisfies the three conditions in Theorem 3.

Example 1. *Consider the following linear scalar system:*

$$\begin{cases} s_{k+1} = c_0 s_k + c_1 \alpha_{k+1} \\ \beta_k = c_2 s_k, \end{cases} \tag{42}$$

where $|c_0| < 1$*,* $c_1 \neq 0$*, and* $\{\alpha_k, k \geq 0\}$ *is a white Gaussian random sequence which is independent of* s_1*. By iterations, we can easily get*

$$s_k = c_0^{k-1} s_1 + c_1 \sum_{i=0}^{k-2} c_0^i \alpha_{k-i},\ \forall\ k \geq 2, \tag{43}$$

then we have

$$\begin{aligned} \mathbb{E}[|\alpha_k|^2] =& 1, \\ \mathbb{E}[|s_k|^2] =& c_0^{2(k-1)} \mathbb{E}[|s_1|^2] + c_1^2 \sum_{i=0}^{k-2} c_0^{2i} \\ =& c_0^{2(k-1)} \mathbb{E}[|s_1|^2] + c_1^2 \frac{1 - c_0^{2(k-1)}}{1 - c_0^2}. \end{aligned} \tag{44}$$

Apparently, $\sup_{k\geq 1} \mathbb{E}\left[|s_k|^2\right] < \infty$ *and* $\sup_{k\geq 1} \mathbb{E}\left[|\alpha_k|^2\right] < \infty$*. It can be easily checked that the three conditions in Theorem 3 are satisfied.*

IV. RNN BASED FDFS

In this section, we shall investigate the connections between RNNs and FDFs. KF, as a special case of FDFs, is discussed explicitly.

A. Algorithm

Observing that, in FDFs, we have the following evolution functions of the sufficient statistics and the estimation:

$$\begin{cases} S_{k|k} = \Phi(S_{k-1|k-1}, y_k), \\ \mathbb{E}[x_k|Y_k] = \Gamma(S_{k|k}), \end{cases} \tag{45}$$

by Eq. (14), which includes KF with Eq. (12) and Eq. (13) as a special case. The framework of dynamical system Eq. (45) is shown in Figure 4. Comparing Fig. 3 and Fig. 4, it is obvious that Eq. (45) is an open dynamical system with the stochastic inputs $\{y_k, k \geq 0\}$, and the stochastic outputs $\{\mathbb{E}[x_k|Y_k]\}$ which are the desired optimal estimates of the states.

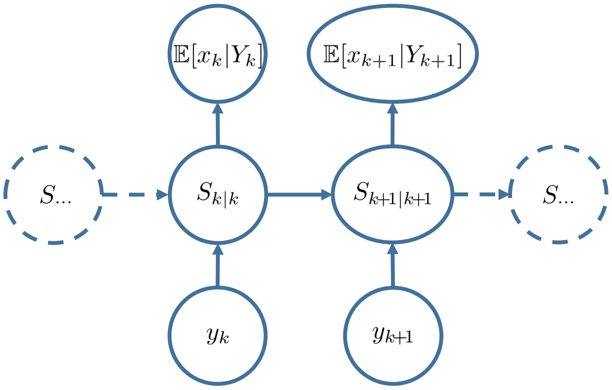


Fig. 4: The framework of FDFs with observation y, sufficient statistics S and optimal estimate $\mathbb{E}[x_k|Y_k]$.

Naturally, using the universal approximation of RNN with stochastic inputs as shown in Theorem 3, we can approximate open dynamical system Eq. (45) by RNN functions as detailed in the previous section. Following Theorem 3, it is known that we can approximate Φ and Γ by functions $\tilde{\Phi}$ and $\tilde{\Gamma}$ represented by feedforward networks, respectively, i.e.,

$$\tilde{\Phi}(s, y) = \bar{\Phi}(\mathcal{T}_{K_1}s, \mathcal{T}_{K_2}y), \tag{46}$$

$$\tilde{\Gamma}(s) = \bar{\Gamma}(\mathcal{T}_{K_1}s), \tag{47}$$

where $\bar{\Phi} \in \Sigma^{n_s+m,n_s}(\kappa)$, $\bar{\Gamma} \in \Sigma^{n_s,n}(\kappa)$, K_1 and K_2 are two positive numbers which are the parameters of the RNN, and $\mathcal{T}$ is the truncation operator defined in Eq. (2). Then we can obtain a RNN system which is as follows:

$$\begin{cases} \tilde{S}_{k|k} = \tilde{\Phi}(\tilde{S}_{k-1|k-1}, y_k), \\ \hat{x}_{k|k} = \tilde{\Gamma}(\tilde{S}_{k|k}), \end{cases} \tag{48}$$

where $\tilde{S}_{k|k}$ and $\hat{x}_{k|k}$ are defined as the state and output of the RNN system Eq. (48), respectively. We need to remark here that $\hat{x}_{k|k}$ is a function of Y_k.

Using the data $\{y_k, \mathbb{E}[x_k|Y_k]\}_{k\geq 0}$, we can train RNN system Eq. (48) such that $\mathbb{E}[x_k|Y_k]$ can be well approximated by the output $\hat{x}_{k|k}$, which can be regarded as the estimate of the state x_k of Eq. (4) based on observation history Y_k. And we call this filtering method as RNN filter (RNNF).

B. Error analysis

1) RNN based FDFs: Following the results in Theorem 3, we can easily obtain the following theorem, which says that the error between the optimal estimate $\mathbb{E}[x_k|Y_k]$ and the estimate $\hat{x}_{k|k}$ by RNNF can be arbitrarily small as time k approaches to infinity.

Theorem 4. *Consider a discrete filtering system Eq. (4) with optimal FDF. Let $S_{k|k}, k \geq 0$ be the theoretical statistics evolving according to Eq. (45) and $\tilde{S}_{k|k}, k \geq 0$ be the statistics generated by our RNNF which evolve according to Eq. (48). We need the following assumptions:*

- *the sufficient statistics $\left\{S_{k|k}\right\}_{k\geq 0}$ and the observations $\{y_k\}_{k\geq 0}$ are uniformly integrable;*
- *function Φ is Lipschitz, i.e., for any $S, \bar{S} \in \mathbb{R}^{n_S}$ and $y, \bar{y} \in \mathbb{R}^m$,*
$$\left\|\Phi(S, y) - \Phi(\bar{S}, \bar{y})\right\|_1 \leq C_{\Phi 1}\left\|S - \bar{S}\right\|_1 + C_{\Phi 2}\left\|y - \bar{y}\right\|_1, \tag{49}$$
where n_S is the dimension of $S_{k|k}$, $C_{\Phi 1}$ and $C_{\Phi 2}$ are Lipschitz constants, and $C_{\Phi 1}$ satisfies $|C_{\Phi 1}| < 1$;
- *for $\forall\ \epsilon > 0$, there exists $\delta > 0$, such that for any $s, \bar{s} \in L^1(\Omega; \mathbb{R}^{n_S})$ satisfying $\|s - \bar{s}\|_1 < \delta$, we have $\|\Gamma(s) - \Gamma(\bar{s})\|_1 < \epsilon$.*

then for any $\varepsilon > 0$, there exists a RNNF Eq. (48), i.e., there exist $\tilde{\Phi}$ and $\tilde{\Gamma}$ of the forms Eq. (46)-Eq. (47), respectively, such that

$$\varlimsup_{k\to\infty}\left\|S_{k|k} - \tilde{S}_{k|k}\right\|_1 < \varepsilon, \tag{50}$$

and

$$\varlimsup_{k\to\infty}\left\|\hat{x}_{k|k} - \mathbb{E}[x_k|Y_k]\right\|_1 < \varepsilon. \tag{51}$$

Proof. The proof is similar to that of Theorem 3. □

This theorem highlights that RNNF can approximate any optimal FDFs.

2) RNNF for linear Gaussian system Eq. (6): Obviously, Theorem 4 works for linear Gaussian system Eq. (6), which is a special case of Eq. (4). Furthermore, the first assumption in Theorem 4 can be replaced by some more explicit assumptions w.r.t. the system.

Naturally, for linear Gaussian system Eq. (6), we can assume that the observations and the sufficient statistics in KF are uniformly integrable, which is the same as the first condition in Theorem 4, i.e., we need the following assumption.

Assumption 1. *The sufficient statistics $\left\{s_{k|k}\right\}_{k\geq 0}$ and the observations $\{y_k\}_{k\geq 0}$ are uniformly integrable.*

In stead of Assumption 1, we can also make two more explicit but strong assumptions, which can make sure that the sufficient statistics and the observations are uniformly bounded under $\|\cdot\|_2$ norm. Then according to Lemma 1, we know that Assumption 1 can be satisfied. The two new assumptions are as follows.

Assumption 2. *We assume that the linear dynamical system of the state in Eq. (6) is stable in mean square sense ([37]), i.e.,*

$$\varlimsup_{k\to\infty}\|x_k\|_2 \leq C_2, \tag{52}$$

where $C_2 > 0$ is a finite constant.

Assumption 3. *The dynamical system Eq. (6) is uniformly completely observable and uniformly completely controllable.* [1]

Based on Assumption 2 and Assumption 3, we give the key Lemma 4 which says that the sufficient statistics $\{s_{k|k}\}_{k\geq 0}$ and the observations $\{y_k\}_{k\geq 0}$ are uniformly bounded.

Lemma 4. *In the discrete linear system Eq. (6), if Assumption 2 and Assumption 3 are satisfied, then $\left\{s_{k|k}\right\}_{k\geq 0}$ and $\{y_k\}_{k\geq 0}$ are uniformly bounded, i.e., there exists a constant $C_3 > 0$, such that*

$$\sup_{k\geq 0} \left\|s_{k|k}\right\|_2 \leq C_3, \tag{53}$$

and

$$\sup_{k\geq 0} \|y_k\|_2 \leq C_3. \tag{54}$$

The proof of this lemma is technical and we put it into Appendix B.

Here we list an example satisfying Assumption 2 and Assumption 3.

Example 2. *The system is as follows:*

$$\begin{cases} x_k = (1-\alpha)x_{k-1} + \sqrt{\alpha}w_{k-1} \\ y_k = \alpha x_k + \sqrt{\alpha}v_k, \end{cases} \tag{55}$$

where $0 < \alpha < 1$ is a small positive parameter.

- *As for Assumption 2, from the state equation of Eq. (55) we have*

$$\begin{aligned} x_k =& (1-\alpha)x_{k-1} + \sqrt{\alpha}w_{k-1} \\ =& (1-\alpha)^2 x_{k-2} + (1-\alpha)\sqrt{\alpha}w_{k-2} + \sqrt{\alpha}w_{k-1} \\ & \vdots \\ =& (1-\alpha)^k x_0 + \sum_{i=0}^{k-1}(1-\alpha)^{k-1-i}\sqrt{\alpha}w_i, \end{aligned} \tag{56}$$

  then we can obtain

$$\begin{aligned} \mathbb{E}[|x_k|^2] =& (1-\alpha)^{2k}\mathbb{E}[|x_0|^2] + \sum_{i=0}^{k-1}(1-\alpha)^{2k-2-2i}\alpha Q \\ =& (1-\alpha)^{2k}\mathbb{E}[|x_0|^2] \\ & + \frac{(1-\alpha)^{2k-2} - (1-\alpha)^2}{(1-\alpha)^2 - 1}\alpha Q. \end{aligned} \tag{57}$$

  It can be easily checked that Assumption 2 is satisfied.
- *As for Assumption 3, it can be easily checked that system shown in Eq. (55) satisfies Assumption 3 using the definitions of uniformly completely observable and uniformly completely controllable in section 7.5 of [15].*

We then derive our result for linear Gaussian system Eq. (6).

Theorem 5. *Assume $s_{k|k}, k \geq 0$ are the theoretical statistics evolving according to Eq. (12) and Eq. (13), and $\tilde{S}_{k|k}, k \geq 0$ are the real statistics computed by our RNN-based filter evolving according to Eq. (48). We make the following assumptions:*

- *the Assumption 2 and Assumption 3 are satisfied, or Assumption 1 is satisfied;*
- *functions φ and γ are Lipschitz, i.e., for any $s, \bar{s} \in L^1(\Omega;\mathbb{R}^{n_s})$, and $y, \bar{y} \in L^1(\Omega;\mathbb{R}^m)$,*

$$\|\varphi(s,y) - \varphi(\bar{s},\bar{y})\|_1 \leq C_{\varphi 1}\|s-\bar{s}\|_1 + C_{\varphi 2}\|y-\bar{y}\|_1,$$

  where n_s is the dimension of $s_{k|k}$, $C_{\varphi 1}$ and $C_{\varphi 1}$ are Lipschitz constants, and $C_{\varphi 1}$ satisfies $|C_{\varphi 1}| < 1$;
- *for $\forall\ \epsilon > 0$, there exists $\delta > 0$, such that for any $s, \bar{s} \in L^1(\Omega;\mathbb{R}^{n_s})$ satisfying $\|s-\bar{s}\|_1 < \delta$, we have $\|\gamma(s) - \gamma(\bar{s})\|_1 < \epsilon$.*

Then for any $\varepsilon > 0$, there exists an RNNF Eq. (48), i.e., there exist $\tilde{\Phi}$ and $\tilde{\Gamma}$ of the forms Eq. (46)-Eq. (47), respectively, such that

$$\overline{\lim_{k\to\infty}} \left\|s_{k|k} - \tilde{S}_{k|k}\right\|_1 < \varepsilon, \tag{58}$$

and

$$\overline{\lim_{k\to\infty}} \left\|\mathbb{E}[x_k|Y_k] - \hat{x}_{k|k}\right\|_1 < \varepsilon. \tag{59}$$

Proof. Under Assumption 2 and Assumption 3, it is known from Lemma 4 that $\left\{\left\|s_{k|k}\right\|_2\right\}_{k\geq 0}$, $\left\{\left\|s_{k+1|k}\right\|_2\right\}_{k\geq 0}$ and $\left\{\|y_k\|_2\right\}_{k\geq 0}$ are uniformly bounded. Then according to Lemma 3, we know that, for any $\varepsilon_1 > 0$, there exist positive numbers K_1 and K_2, such that

$$\begin{aligned} &\sup_{k\geq 0}\left\|s_{k|k} - \mathcal{T}_{K_1}(s_{k|k})\right\|_1 < \varepsilon_1, \\ &\sup_{k\geq 0}\left\|s_{k+1|k} - \mathcal{T}_{K_1}(s_{k+1|k})\right\|_1 < \varepsilon_1, \\ &\sup_{k\geq 0}\left\|y_k - \mathcal{T}_{K_2}(y_k)\right\|_1 < \varepsilon_1. \end{aligned}$$

Then we can obtain the desired results following the same procedures as in Theorem 3. □

V. Experiments

In the experiments, our proposed RNNF is compared with KF which provides the optimal estimate. For the purpose of comparing the performance of different methods, we introduce the mean of the squared error (MSE) based on 100 realizations, which is defined as follows:

$$\text{MSE} := \frac{1}{100}\sum_{i=1}^{100}\frac{1}{K_2+1}\sum_{k=0}^{K_2}\left|x_k^{(i)} - \hat{x}_k^{(i)}\right|^2, \tag{60}$$

where $x_k^{(i)}$ is the real state at discrete time instant k in the i-th experiment and $\hat{x}_k^{(i)}$ is the estimation of $x_k^{(i)}$, with $0 \leq k \leq K_2$, where $K_2 \in \mathbb{N}$ is the total time step.

[1] The definitions of uniformly completely observable and uniformly completely controllable can be found in section 7.5 of [15].

A. Neural network architecture and training algorithm

The RNN based filter Eq. (48), which is also denoted as RNNF$(y;\theta)$, consists of two parts:

$$\begin{aligned}\tilde{S}_{k|k} &= \tilde{\Phi}(\tilde{S}_{k-1|k-1}, y_k; \theta_1),\\ \hat{x}_{k|k} &= \tilde{\Gamma}(\tilde{S}_{k|k}; \theta_2),\end{aligned} \tag{61}$$

where $\theta^T = [\theta_1^T, \theta_2^T]$ is all trainable parameters in RNNF, $\tilde{\Phi}$ is represented by a single layer feedforward network with l neurons, l is a hyperparameters to be determined, $\tilde{\Gamma}$ is a linear function with input dimension l and output dimension n equal to the dimension of state x_k.

Naturally, we aim to minimize

$$L_0(\theta) := \frac{1}{K_1+1}\mathbb{E}\left[\sum_{k=0}^{K_1}\left|\hat{x}_{k|k} - \mathbb{E}[x_k|Y_k]\right|^2\right], \tag{62}$$

where $K_1 \in \mathbb{N}$ is the total time step in training. Observing that

$$\begin{aligned}
&\mathbb{E}\left[\left|x_k - \hat{x}_{k|k}\right|^2\right]\\
=&\mathbb{E}\left[\left|x_k - \mathbb{E}[x_k|Y_k] + \mathbb{E}[x_k|Y_k] - \hat{x}_{k|k}\right|^2\right]\\
=&\mathbb{E}\left[\left|x_k - \mathbb{E}[x_k|Y_k]\right|^2\right] + \mathbb{E}\left[\left|\mathbb{E}[x_k|Y_k] - \hat{x}_{k|k}\right|^2\right]\\
&+ 2\mathbb{E}\left[\left(x_k - \mathbb{E}[x_k|Y_k]\right)^T\left(\mathbb{E}[x_k|Y_k] - \hat{x}_{k|k}\right)\right]\\
=&\mathbb{E}\left[\left|x_k - \mathbb{E}[x_k|Y_k]\right|^2\right] + \mathbb{E}\left[\left|\mathbb{E}[x_k|Y_k] - \hat{x}_{k|k}\right|^2\right]\\
&+ 2\mathbb{E}\left[\mathbb{E}\left(\left(x_k - \mathbb{E}[x_k|Y_k]\right)^T\left(\mathbb{E}[x_k|Y_k] - \hat{x}_{k|k}\right)\Big| Y_k\right)\right]\\
=&\mathbb{E}\left[\left|x_k - \mathbb{E}[x_k|Y_k]\right|^2\right] + \mathbb{E}\left[\left|\mathbb{E}[x_k|Y_k] - \hat{x}_{k|k}\right|^2\right]\\
&+ 2\mathbb{E}\left[\mathbb{E}\left(x_k - \mathbb{E}[x_k|Y_k]\right| Y_k\right)^T\left(\mathbb{E}[x_k|Y_k] - \hat{x}_{k|k}\right)\right]\\
=&\mathbb{E}\left[\left|x_k - \mathbb{E}[x_k|Y_k]\right|^2\right] + \mathbb{E}\left[\left|\mathbb{E}[x_k|Y_k] - \hat{x}_{k|k}\right|^2\right],
\end{aligned}$$

where the third equality comes from the tower property of conditional expectation, and the forth equality is due to the fact that $\hat{x}_{k|k}$ is $\sigma(Y_k)$-measurable, it follows that

$$\arg\min_{\theta} L_0(\theta) = \arg\min_{\theta} L(\theta), \tag{63}$$

where

$$L(\theta) := \frac{1}{K_1+1}\mathbb{E}\left[\sum_{k=0}^{K_1}\left|\hat{x}_{k|k} - x_k\right|^2\right]. \tag{64}$$

Therefore, instead of data $\{y_k, \mathbb{E}[x_k|Y_k]\}_{k\geq 0}$ where $\mathbb{E}[x_k|Y_k]$ cannot be obtained in most cases, we only need data $\{y_k, x_k\}_{k\geq 0}$ which can be easily generated from the system Eq. (4). We need to remark that this step is crucial since it allows us to get accessible data.

In real computations, the expectation in $L(\theta)$ is approximated by the average of the results obtained from a large number of trials. Hence, we define the loss function as follows:

$$L^{(N)}(\theta) := \frac{1}{N}\frac{1}{K_1+1}\sum_{n=1}^{N}\left(\sum_{k=0}^{K_1}\left|x_k(\omega_n) - \hat{x}_{k|k}(\omega_n)\right|^2\right), \tag{65}$$

where $\hat{x}_{k|k}(\omega_n) = \text{RNNF}(y_k(\omega_n);\theta)$ is the output of RNNF with input $y_k(\omega_n)$, and N and K_1 are the numbers of Monte Carlo paths and total time steps in training, respectively.

The detailed procedures of RNNF are listed as follows:

Algorithm 1 RNNF training algorithm

```
Require:
    Train data: $\left\{\{(y_k(\omega_n), x_k(\omega_n))\}_{k=0}^{K_1}\right\}_{n=1}^{N}$;
    Batch size: $M$;
    Total epochs: $I$;
    Learning rate: $\lambda$;
Ensure:
    RNNF output: $\left\{\{\text{RNNF}(y_k(\omega_n));\theta\}_{k=0}^{K_1}\right\}_{n=1}^{N}$;
for $i = 1, \dots I$ do
    Sample batch $\left\{\{(y_k(\omega_n), x_k(\omega_n))\}_{k=0}^{K_1}\right\}_{n=1}^{N}$ from Train data;
    Compute loss $L(\theta)$ via Eq. (65);
    Update $\theta$ via $\theta \leftarrow \theta - \lambda\nabla_\theta L(\theta)$.
end for
```

We have given the theoretical support of the RNNF in the previous sections. Now from the computational point of view, we do not have to check the conditions posed in Theorem 4 and Theorem 5 which are required in the proof of the convergence of the proposed algorithm. During the implementation of the RNNF, our algorithm is divided into two parts: off-line computational step and on-line computational step. In the off-line step, as shown in Algorithm 1, we only need to generate data from the system model Eq. (4) to train the RNN of the form Eq. (24) with the MSE loss defined in Eq. (65). This is a rather standard procedure in the deep learning. In the on-line step, we just get the estimate $\hat{x}_{k|k}$ using the trained RNNF with y_k as the input. And therefore our algorithm can be implemented in the real-time manner [38].

For the purpose of showing RNN is a universal Approximators with stochastic inputs which means that a well-trained RNN within finite time can be a approximator in global time (infinite time theoretically), in the following experiment, we train the RNN with $K_1 = 1000$ and test it on the data generated from the system with $K_2 = 1000$ and $K_2 = 10000$. The former proves validity, and the latter proves universality.

B. Implementation details

In the following two numerical examples, we shall investigate the efficiency of our RNNF through the comparison with KF. All experiments are using Nvidia RTX2060s and run on 16 Intel(R) Core(TM) i7-10700 CPU @ 2.90GHZ. Besides, we use Pytorch in RNNF and numpy which is a python package for scientific computing in KF.

The parameters used in RNNF are summarized in TABLE I. The sample paths in training set are used to approximate the true loss function via Monte Carlo, and we find that the number of paths N can be chosen to be greater than 2000. The batch size mainly affects the speed of training, but has little effect on the results. We need to remark that the dimension of the hidden layer, learning rate and total epochs have a greater impact on the results. The dimension of the hidden layer l is directly related to the approximation ability of the model. If l is too small, the RNNF cannot capture the filtering information.

TABLE I: The Parameters of RNNF Used in Two Examples.

Example	System Eq. (66)	System Eq. (67)
dimension of input layer	10	10
dimension of hidden layer	100	128
dimension of output layer	10	10
paths in training set	2000	5000
paths in test set	100	100
activation function	tanh	tanh
optimizer	Adam	Adam
learning rate λ	0.0001	0.00005
total epochs I	3000	3000
batch size M	256	256

If it is too large, RNNF will converge slowly. As suggested by the design of RNNF, and also through our experiments, we can choose this parameter approximately as the square of the state dimension n. The choice of the initial learning rate is very critical. We find that $\lambda \in [10^{-5},\ 10^{-4}]$ is appropriate. And the number of total epochs can be chosen to be in $[2000, 5000]$.

C. Linear system with independent noises

The first example we consider here is a linear Gaussian system with independent noises which is as follows:

$$\begin{cases} x_k = (\alpha A_n + I_n)x_{k-1} + \sqrt{\alpha}w_{k-1}, \\ y_k = \alpha x_k + \sqrt{\alpha}v_k, \end{cases} \tag{66}$$

where $x_0 \sim \mathcal{N}(0, I_n)$ with identity matrix $I_n \in \mathbb{R}^n$, $n = 10$, $\alpha = 0.01$, w_k and v_k are standard white noises, I_n is a $n \times n$ identity matrix and $A_n = [a_{ij}]$ is a matrix with elements as follows:

$$a_{ij} = \begin{cases} 0.1, & \text{if } i + 1 = j, \\ -0.4, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

Firstly, we test the RNNF on the data generated from the system Eq. (66) with $K_2 = 1000 = K_1$. The estimation results of our RNNF and KF in one experiment are shown in Fig. 5. It can be easily seen that the estimation result of our RNNF is very close to that of optimal KF. Secondly, we test the RNNF on the data generated from the system Eq. (66) with $K_2 = 10000$ while we train RNNF with $K_1 = 1000 < K_2$. The L_1-error between the estimate by RNNF and the optimal estimate by KF is displayed in Fig. 6. It is obvious that our RNNF can still perform well on the larger time interval $0 \leq k \leq 10000$. This is natural since the filtering system Eq. (66) is time-invariant and therefore the parameters of RNNF in every step are the same. That is, $\tilde{\Phi}$ and $\tilde{\Gamma}$ in Eq. (48) corresponding to this example are independent of time k. Furthermore, it can be seen that the L_1-error between the estimate by RNNF and the optimal estimate by KF stabilizes around a small value as time step k increases, which also verifies our conclusion in Theorem 5.

The MSE defined in Eq. (60) and average running time based on 100 simulations are listed in TABLE II. We can know that RNNF has comparable performance compared with optimal KF.

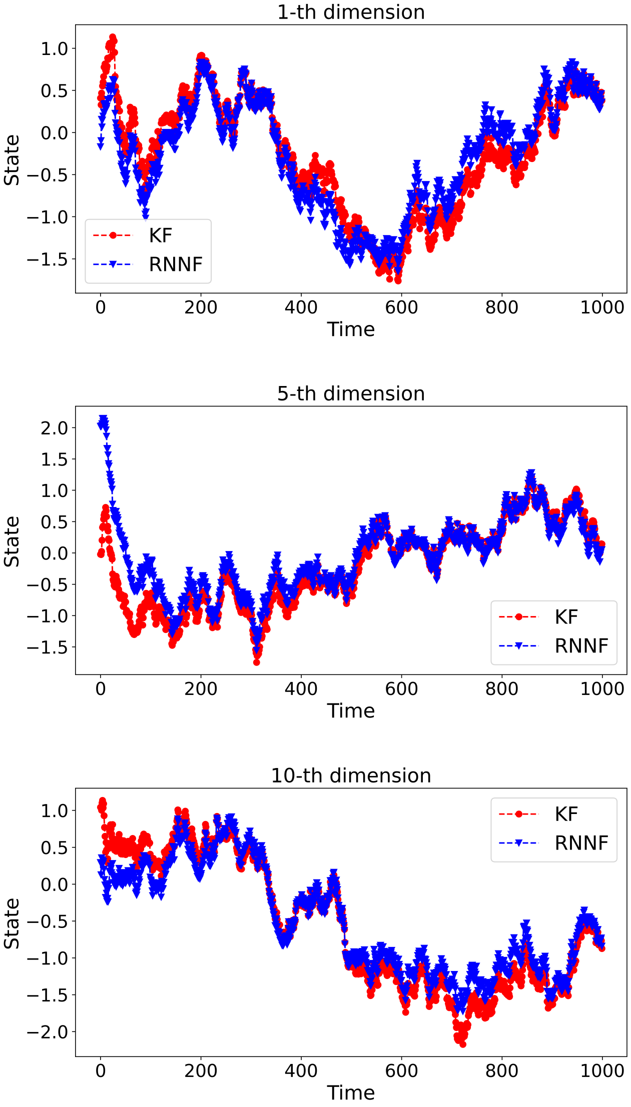


Fig. 5: The estimation results of KF and RNNF in one experiment for linear filtering system Eq. (66) with $0 \leq k \leq 1000$.

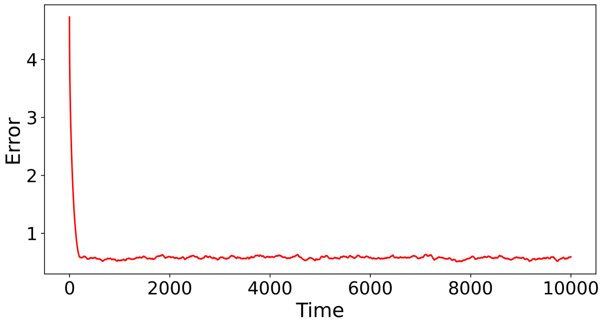


Fig. 6: The L_1-error between the estimates of KF and RNNF over discrete time k for linear filtering system Eq. (66) with $0 \leq k \leq 10000$.

TABLE II: The average performance of different methods based on 100 simulations with time step $K_2 = 1000, 10000$ for system Eq. (66).

Method	MSE	Running time (s)
RNN Filter ($K_2 = 1000$)	6.9782	0.0379
KF ($K_2 = 1000$)	7.6302	0.0396
RNN Filter ($K_2 = 10000$)	7.8948	0.4301
KF ($K_2 = 10000$)	7.4164	0.3883

TABLE III: The average performance of different methods based on 100 simulations with time step $K_2 = 1000, 10000$ for system Eq. (67).

Method	MSE	Running time (s)
RNNF ($K_2 = 1000$)	3.7720	0.0372
KF ($K_2 = 1000$)	4.6101	0.0387
RNNF ($K_2 = 10000$)	3.8097	0.4238
KF ($K_2 = 10000$)	3.8493	0.3805

D. Linear system with correlated noises

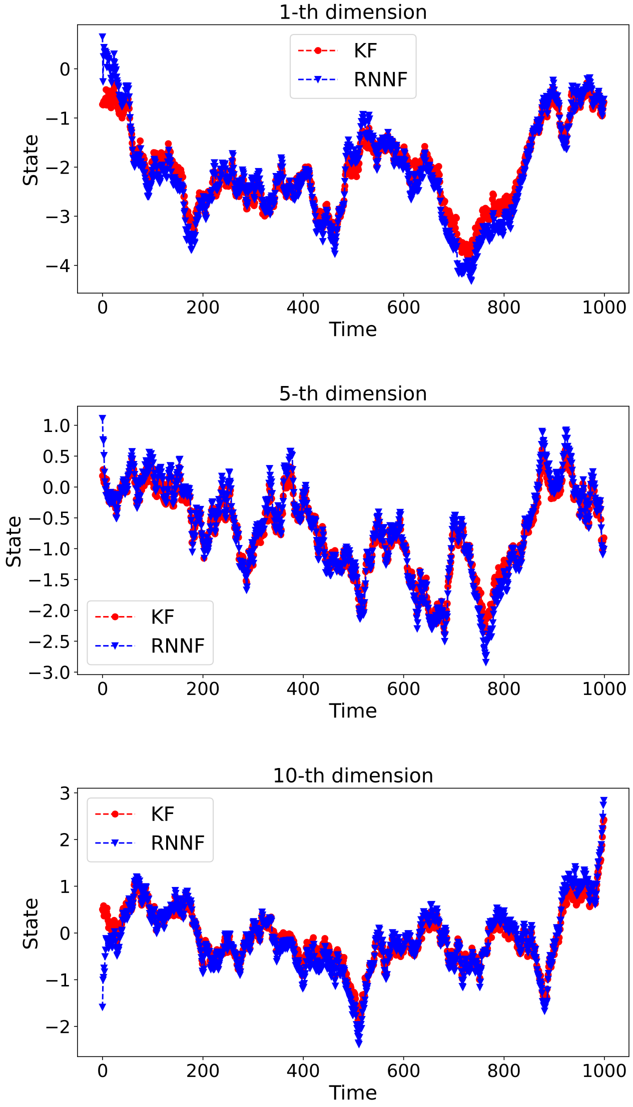
The second example is similar to the first one, and the only difference is that the noises in state equation and observation equation are correlated. More explicitly, we consider the following system:

$$\begin{cases} x_k = (\alpha A_n + I_n)x_{k-1} + \sqrt{\alpha}w_{k-1} + \sqrt{\alpha}v_{k-1}, \\ y_k = \alpha x_k + \sqrt{\alpha}v_k, \end{cases} \tag{67}$$

where $x_0 \sim \mathcal{N}(0, I_n)$.

As pointed out in Appendix C, we can obtain the optimal estimation of state in system Eq. (67) by KF, which means that system Eq. (67) possesses a FDF. The estimation results in one experiment with $K_2 = 1000$ are displayed in Fig. 7. The MSEs and running time with $K_2 = 1000$ and $K_2 = 10000$ are listed in TABLE III. And the L_1-error between the estimate by RNNF and the optimal estimate by KF is shown in Fig. 8 when $K_2 = 10000$.

These two examples illustrate that a well-trained RNNF can learn the true optimal estimate which is given by KF in the linear case. And we no longer need to distinguish whether the noises of the system are correlated or not. As long as the system has FDF, we can use RNNF to solve it. Actually, for systems that does not possess FDF, we can also use RNNF. In this case, RNNF can be regarded as the approximation of the optimal filter through FDF, and this idea is also employed in many suboptimal filters such as EKF. Compared with traditional suboptimal filters, we do not need to know the exact models of the filtering system and all we need are the data $\{y_k, x_k\}$ generated from the dynamical systems. Moreover, the RNNF can have better capability to capture the information of the optimal filter since it aims to minimize the mean square error loss Eq. (64).


1-th dimension

5-th dimension

10-th dimension

Fig. 7: The estimation results of KF and RNNF in one experiment for linear filtering system Eq. (67) with $0 \leq k \leq 1000$.

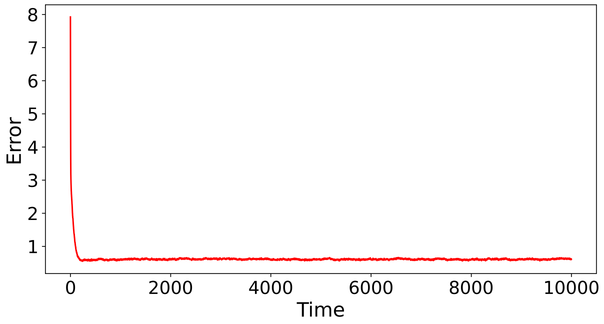


Fig. 8: The L_1-error between the estimates of KF and RNNF over discrete time k for linear filtering system Eq. (67) with $0 \leq k \leq 10000$.

VI. Conclusion

In this work, we investigate the approximation capability of RNN and prove that any open dynamical system with stochastic inputs can be well approximated by RNN. Based on this, we construct a novel filter by RNN, and the estimation error has also been analyzed for systems with finite dimensional filter.

In addition, our theoretical results have also been verified by numerical examples.

However, we use traditional RNNs in this work, and the numerical performances may be improved by using some other RNNs such as long short term memory (LSTM) [39] and gated recurrent units (GRU) [40]. Besides, we have not investigate general nonlinear filtering systems which do not possess FDFs in this work. We will focus on these problems in our future works.

Acknowledgment

This work is supported by National Natural Science Foundation of China (NSFC) grant (11961141005), Tsinghua University start-up fund, and Tsinghua University Education Foundation fund (042202008). Professor Stephen Shing-Toung Yau is grateful to the National Center for Theoretical Sciences (NCTS) for providing an excellent research environment while part of this research was done.

Appendix A
Proof of Eq. (3)

In order to obtain Eq. (3), we only need to prove that

$$|\mathcal{T}_K x - \mathcal{T}_K \bar{x}| \le |x - \bar{x}|, \ \forall \ x, \bar{x} \in \mathbb{R}. \tag{68}$$

Without loss of generality, we can assume $|x| \le |\bar{x}|$, and then the proof can be divided into three cases.

Case 1: $0 < K \le |x| \le |\bar{x}|$. We have

$$\begin{aligned} |\mathcal{T}_K x - \mathcal{T}_K \bar{x}| =& |K \cdot \operatorname{sign} x - K \cdot \operatorname{sign} \bar{x}| \\ =& \begin{cases} 0 & \text{if } x \cdot \bar{x} > 0 \\ 2K & \text{if } x \cdot \bar{x} < 0 \end{cases}, \\ \le& |x - \bar{x}|. \end{aligned}$$

Case 2: $0 \le |x| \le K \le |\bar{x}|$. In this case, we have

$$\begin{aligned} |\mathcal{T}_K x - \mathcal{T}_K \bar{x}| =& |x - K \cdot \operatorname{sign} \bar{x}| \\ =& \begin{cases} |x - K| & \text{if } \bar{x} > 0 \\ |x - (-K)| & \text{if } \bar{x} < 0 \end{cases}, \\ \le& |x - \bar{x}|. \end{aligned}$$

Case 3: $0 \le |x| \le |\bar{x}| \le K$. It is obvious that $|\mathcal{T}_K x - \mathcal{T}_K \bar{x}| = |x - \bar{x}|$.

Combining these three cases, we obtain the desired Eq. (68), and then Eq. (3) holds naturally.

Appendix B
Proof of Lemma 4

Before we give the proof, we need to list a lemma which gives the bound of the conditional covariance.

Lemma 5 (Lemma 7.1 in [15])**.** *If Assumption 3 is satisfied and $P_{0|0} \succcurlyeq 0$[2], then $P_{k|k}$ is uniformly bounded from above for all $k \ge N$,*

$$P_{k|k} \preccurlyeq \left(\frac{1 + \alpha\beta}{\alpha} \right) I, \ k \ge N, \tag{69}$$

where N is a positive integer, I is the $n \times n$ identity matrix and α, β are positive constants.

Based on Lemma 5, Assumption 2 and Assumption 3, we give the proof of Lemma 4.

Proof of Lemma 4:

The proof can be divided into two steps. In the first step, we shall analyze the bounds of the sufficient statistics, and in the second step, the bounds of the observations will be discussed.

Step 1: It can be known from the evolution functions Eq. (9) and Eq. (10) of the conditional covariance that

$$\begin{cases} P_{k|k-1} = GQG^T + FP_{k-1|k-1}F^T \\ \quad P_{k|k} = P_{k|k-1} - P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \\ \qquad\qquad \cdot HP_{k|k-1}, \end{cases} \tag{70}$$

and the initial value is $P_{0|0}$. It follows that the conditional covariance P evolves in a deterministic manner and without any randomness according to the evolution equation Eq. (70), therefore P is independent of the observations. Since

$$P_{k|k} = \mathbb{E} \left[\left(x_k - m_{k|k} \right) \left(x_k - m_{k|k} \right)^T \middle| Y_k \right],$$

and $P_{k|k}$ is independent of Y_k, we have

$$P_{k|k} = \mathbb{E} \left[\left(x_k - m_{k|k} \right) \left(x_k - m_{k|k} \right)^T \right]. \tag{71}$$

According to Lemma 5, we have

$$P_{k|k} \preccurlyeq \left(\frac{1 + \alpha\beta}{\alpha} \right) I, \ k \ge N, \tag{72}$$

where N is a positive integer and α, β are positive constants. It can be easily checked that, there exists a positive constant α_0, such that

$$P_{k|k} \preccurlyeq \alpha_0 I, \ \forall \ k \ge 0. \tag{73}$$

According to Eq. (71), we have

$$\begin{aligned} & \|x_k - m_{k|k}\|_2^2 \\ =& \mathbb{E} \left[\left(x_k - m_{k|k} \right)^T \left(x_k - m_{k|k} \right) \right] \\ =& \operatorname{tr} \left(P_{k|k} \right). \end{aligned} \tag{74}$$

Combining Eq. (73) and Eq. (74), we know that

$$\left\| x_k - m_{k|k} \right\|_2 \le \sqrt{n\alpha_0}, \ \forall \ k \ge 0. \tag{75}$$

Then according to Eq. (52) and Eq. (75), we have

$$\begin{aligned} \left\| m_{k|k} \right\|_2 \le& \left\| x_k - m_{k|k} \right\|_2 + \left\| x_k \right\|_2 \\ \le& \sqrt{n\alpha_0} + C_2, \ \forall \ k \ge 0. \end{aligned} \tag{76}$$

Now we have the conclusion that $m_{k|k}$ and $P_{k|k}$ are bounded for all $k \ge 0$. Following the similar procedure as above, we can know that $m_{k|k-1}$ and $P_{k|k-1}$ are bounded for all $k \ge 0$. According to Eq. (11), we know that all sufficient statistics $\left\{ \left\| s_{k|k} \right\|_2, \ k \ge 0 \right\}$ are uniformly bounded by some constant $C_{31} > 0$.

[2] Here, $X \succcurlyeq Y$ ($X \preccurlyeq Y$ resp.) if and only if $X - Y$ ($Y - X$ resp.) is positive semi-definite, where X and Y are symmetric matrices.

Step 2: We now prove $\|y_k\|_2$ is uniformly bounded. According to Eq. (6), we have

$$\begin{aligned}&\mathbb{E}\left[y_k^T y_k\right]\\ =&\mathrm{tr}\mathbb{E}\left[y_k y_k^T\right]\\ =&\mathrm{tr}\mathbb{E}\left[Hx_kx_k^TH^T+v_kv_k^T\right]\\ \leq& C_2^2\sum_{l=1}^{m}\sum_{i,j=1}^{n} H_{l,j}H_{l,i}+\mathrm{tr}R,\end{aligned}\tag{77}$$

where $H_{l,j}$ is the (l,j)-th entry of the matrix H, and the inequality follows from Assumption 2. Apparently, $\|y_k\|_2$ is uniformly bounded by some constant $C_{32}>0$ for all $k\geq 0$.

Let $C_3 := \max\{C_{31}, C_{32}\}$, then we obtain Eq. (53)-Eq. (54).

Appendix C
Kalman filter for linear system with correlated noises

Instead of linear system Eq. (6) with uncorrelated noises, we consider the following system:

$$\begin{cases}x_k = Fx_{k-1}+Gw_k,\\ y_k = Hx_k+v_k,\end{cases}\tag{78}$$

where the initial state $x_0 \sim \mathcal{N}(m_{0|0}, P_{0|0})$ is Gaussian, $\{w_k, k=0,1,\cdots\}$ and $\{v_k, k=1,\cdots\}$ are two *correlated* white Gaussian sequences which are independent of the initial state x_0. More explicitly, we have

$$\mathbb{E}\left[w_k v_l^T\right] = C\delta_{kl}.$$

Similar to the KF, the discrete Kalman-Bucy filter for system Eq. (78) is listed as follows, where the conditional mean m and covariance P are defined in Eq. (7) and Eq. (8), respectively. For $k\geq 1$:

1) Prediction: given $m_{k-1|k-1}$ and $P_{k-1|k-1}$, we obtain $m_{k|k-1}$ and $P_{k|k-1}$ by

$$\begin{cases}m_{k|k-1} = Fm_{k-1|k-1},\\ P_{k|k-1} = GQG^T+FP_{k-1|k-1}F^T.\end{cases}$$

2) Updating: $m_{k|k}$ and $P_{k|k}$ are updated according to

$$\begin{cases}m_{k|k} = m_{k|k-1}+K_k^{\mathrm{c}}(y_k - Hm_{k|k-1}),\\ P_{k|k} = P_{k|k-1}-K_k^{\mathrm{c}}\left[HP_{k|k-1}+C^TG\right],\end{cases}$$

where the Kalman gain

$$\begin{aligned}K_k^{\mathrm{c}} = \left[P_{k|k-1}H^T+GC\right]\cdot\Big[&HP_{k|k-1}H^T\\ &+HGC+C^TG^TH^T+R\Big]^{-1}.\end{aligned}$$

Obviously, system Eq. (78) has the finite dimensional filter, i.e., the conditional density function $p(x_k|Y_k)$ can be determined by finite statistics. Therefore, we have Theorem 4 for system Eq. (78).

References

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[2] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[3] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH*, 2010, pp. 1045–1048.

[4] T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, and S. Khudanpur, "Extensions of recurrent neural network language model," in *2011 IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP)*. IEEE, 2011, pp. 5528–5531.

[5] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*. PMLR, 2014, pp. 1764–1772.

[6] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[7] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with LSTM recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3547–3555.

[8] A. Graves, S. Fernández, M. Liwicki, H. Bunke, and J. Schmidhuber, "Unconstrained online handwriting recognition with recurrent neural networks," in *Advances in Neural Information Processing Systems 20, NIPS 2008*, 2008.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[10] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *arXiv preprint arXiv:1801.01078*, 2017.

[11] A. M. Schäfer and H.-G. Zimmermann, "Recurrent neural networks are universal approximators," *International Journal of Neural Systems*, vol. 17, no. 04, pp. 253–263, 2007.

[12] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[13] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural networks*, vol. 2, no. 3, pp. 183–192, 1989.

[14] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[15] A. H. Jazwinski, *Stochastic processes and filtering theory*. New York and London: Academic Press, 1970.

[16] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[17] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[18] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[19] P. L. Houtekamer and H. L. Mitchell, "Data assimilation using an ensemble Kalman filter technique," *Month. Weather. Rev.*, vol. 126, no. 3, pp. 796–811, 1998.

[20] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, 1993.

[21] V. E. Beneš, "Exact finite-dimensional filters for certain diffusions with nonlinear drift," *Stochastics: an International Journal of Probability & Stochastic Processes*, vol. 5, no. 1-2, pp. 65–92, 1981.

[22] ——, "New exact nonlinear filters with large Lie algebras," *Systems & Control Letters*, vol. 5, no. 4, pp. 217–221, 1985.

[23] W.-L. Chiou and S. S.-T. Yau, "Finite-dimensional filters with nonlinear drift II: Brockett's problem on classification of finite-dimensional estimation algebras," *SIAM Journal on Control and Optimization*, vol. 32, no. 1, pp. 297–310, 1994.

[24] R.-T. Dong, L.-F. Tam, W. S. Wong, and S. S.-T. Yau, "Structure and classification theorems of finite-dimensional exact estimation algebras," *SIAM Journal on Control and Optimization*, vol. 29, no. 4, pp. 866–877, 1991.

[25] S. S.-T. Yau, "Finite dimensional filters with nonlinear drift. I: A class of filters including both Kalman-Bucy and Benes filters," *Journal of Mathematical Systems, Estimation, and Control*, vol. 4, pp. 181–203, 1994.

[26] ——, "Complete classification of finite-dimensional estimation algebras of maximal rank," *International Journal of Control*, vol. 76, no. 7, pp. 657–677, 2003.

[27] S. S.-T. Yau and G. Hu, "Classification of finite-dimensional estimation algebras of maximal rank with arbitrary state–space dimension and Mitter conjecture," *International Journal of Control*, vol. 78, no. 10, pp. 689–705, 2005.

[28] J. T.-H. Lo, "Synthetic approach to optimal filtering," *IEEE Transactions on Neural Networks*, vol. 5, no. 5, pp. 803–811, 1994.

[29] A. G. Parlos, S. K. Menon, and A. Atiya, "An algorithmic approach to adaptive state filtering using recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1411–1432, 2001.

[30] A. Kutschireiter, S. C. Surace, H. Sprekeler, and J.-P. Pfister, "Nonlinear Bayesian filtering and learning: a neuronal dynamics for perception," *Scientific Reports*, vol. 7, no. 1, pp. 1–13, 2017.

[31] W. Xu, "On deep learning based nonlinear filtering, bachelor thesis, Tsinghua University," Beijing, China, June 2018.

[32] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[33] F. Li, X. Li, X. Zhang, and C. Yang, "Asynchronous filtering for delayed Markovian jump systems via homogeneous polynomial approach," *IEEE Transactions on Automatic Control*, vol. 65, no. 5, pp. 2163–2170, 2020.

[34] S. G. Johnson, "Notes on the equivalence of norms," *MIT Course 18.335*, pp. 1–2, 2012.

[35] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2019, vol. 49.

[36] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2018.

[37] J. Samuels, "On the mean square stability of random linear systems," *IRE Transactions on Circuit Theory*, vol. 6, no. 5, pp. 248–259, 1959.

[38] 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*, vol. 58, no. 10, pp. 2495–2507, 2013.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[40] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

Xiuqiong Chen (M' 20) received the B.S. degree in School of Mathematics and Systems Science, Beihang University, Beijing, China, in 2014, and the Ph.D. degree in applied mathematics from the Department of Mathematical Sciences, Tsinghua University, Beijing, China.

After her graduation, She was a Postdoctoral Scholar with Yau Mathematical Sciences Center, Tsinghua University, Beijing, China, from 2019 to 2021. She joined in Renmin University of China, Beijing, China, since 2021. She is currently an Assistant Professor with School of Mathematics, Renmin University of China. Her research interests include nonlinear filtering and deep learning.

Yangtianze Tao Yangtianze Tao received the B.S degree in College of Mathematics, Sichuan University, Sichuan, China in 2019. Now he is pursuing Ph.D. degree with Department of Mathematical Sciences, Tsinghua University, Beijing, China, under the supervision of Prof. Stephen Yau in the field of applied mathematics.

His research interests include deep learning, machine learning and nonlinear filtering.

Wenjie Xu Wenjie Xu is currently a doctoral student at EPFL in Switzerland. He received his master's degree in Information Engineering from the Chinese University of Hong Kong in 2020 and B.E. degree in Electronic Engineering from Tsinghua University in 2018.

His research interests include optimization, control, and machine learning, with applications to building control and intelligent transportation system.

Stephen S.-T. Yau (F' 03) received the Ph.D. degree in mathematics from the State University of New York at Stony Brook, NY, USA in 1976.

He was a Member of the Institute of Advanced Study at Princeton from 1976-1977 and 1981-1982, and a Benjamin Pierce Assistant Professor at Harvard University during 1977-1980. After that, he joined the Department of Mathematics, Statistics and Computer Science (MSCS), University of Illinois at Chicago (UIC), and served for over 30 years, During 2005-2011, he became a joint Professor with the Department of Electrical and Computer Engineering at the MSCS, UIC. After his retirement in 2012, he joined Tsinghua University, Beijing, China, where he is a full-time professor in the Department of Mathematical Sciences. His research interests include nonlinear filtering, bioinformatics, complex algebraic geometry, CR geometry and singularities theory.

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