Complete Solution to the Most General Nonlinear Filtering Problems with the Capability of Overcoming the Curse of Dimensionality

Stephen S.-T. Yau

Department of Mathematical Sciences,
Tsinghua University, Beijing, China.
&
Yanqi Lake Beijing Institute of Mathematical Sciences
and Applications, Beijing, China.

Pujiang Innovation Forum of Advances of Basic Science, 2023
Joint work with Xiuqiong Chen and Zeju Sun

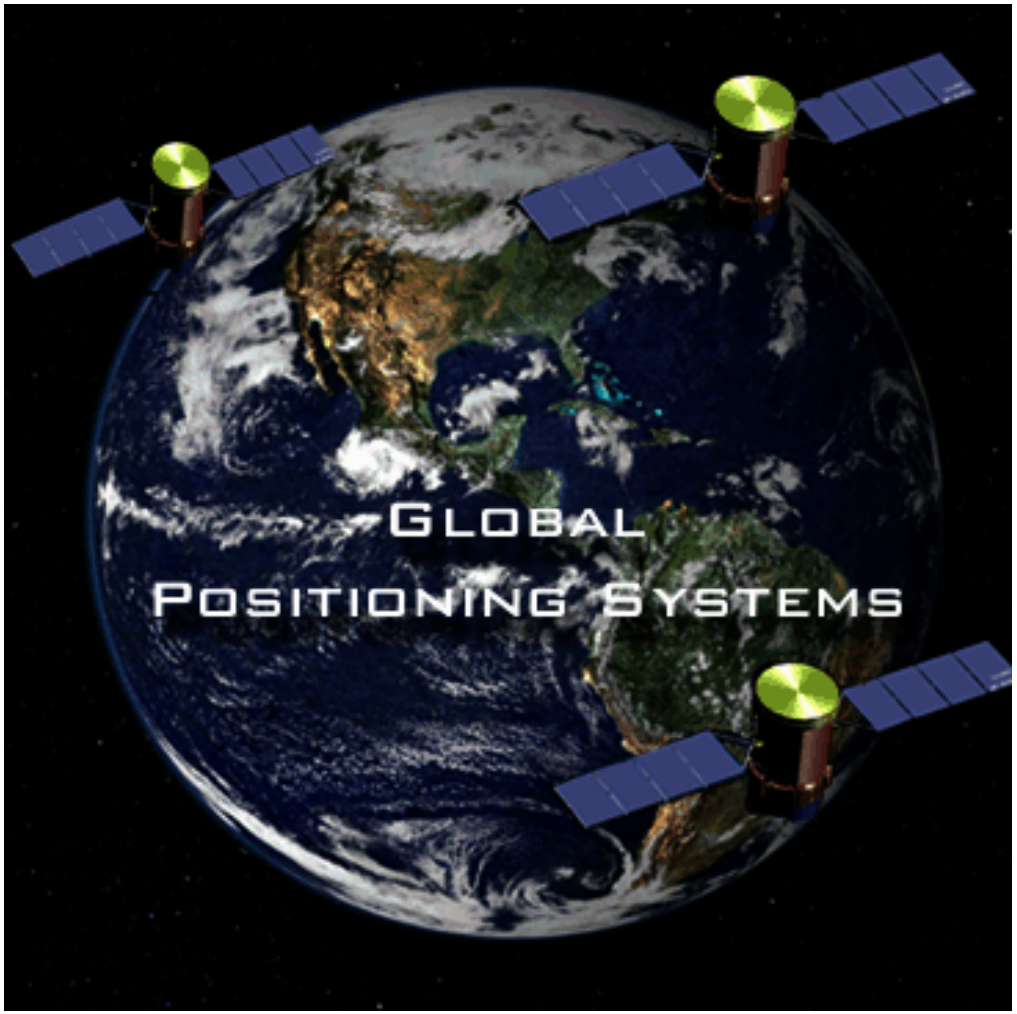
Abstract

The famous filtering problem of estimating the state of a stochastic dynamical system from noisy observations is of central importance in engineering, and high-dimensional nonlinear filtering is still a challenging problem. This problem is reduced to solving the Duncan-Mortensen-Zakai (DMZ) equation which is satisfied by the unnormalized conditional density of the state given the observation history. For general nonlinear filtering problems, we leverage on the representation ability of recurrent neural network and provide a computationally efficient and optimal framework for nonlinear filter design based on Yau-Yau algorithm and recurrent neural network. Theoretically, it can be proved that the size of the neural network required in this algorithm only increases in polynomial (rather than exponentially) with respect to the dimension, which implies that the Yau-Yau algorithm based on recurrent neural network has the capability to overcome the curse of dimensionality. This solves a century old nonlinear filtering problem.

Table Contents

Goal of filterings

Goal: to form the "best estimate" for the true value of some system, given only some potentially noisy observations of that system.



Kalman filter and its applications

R. E. Kalman, 1960: Kalman filter – Optimal linear filter

Application:

- in the navigation of Apollo 13 – by providing the estimates of its trajectory to guide it to the Moon and back;
- in the navigation systems of U.S. Navy nuclear ballistic missile submarine;
- in the guidance and navigation systems of cruise missiles, such as the U.S. Navy's Tomahawk missile and the U.S. Air Force's Air Launched Cruise Missile;
- in the guidance and navigation systems of the NASA Space Shuttle and the International Space Station.

Award: Because of Kalman filter, R. E. Kalman is awarded Charles Stark Draper Prize – one of three prizes that constitute the "Nobel Prizes of Engineering".

Drawbacks of Kalman filter and its derivatives

R. E. Kalman and R. S. Bucy, 1961:
Kalman-Bucy filter – continuous time version of the Kalman filter

"They try all sorts of fixes, but basically the problem is such that the linear theory does not apply"

– R. S. Bucy, *SIAM News* **26**, Aug 1993

Failures of Kalman filter may due to

- Nonlinearity: the outputs are **not** a linear function of the inputs;
- Non-Gaussian of the initial states.

Even its **derivatives**, such as Extended Kalman filter, Unscented Kalman filter, Ensemble Kalman filter, etc can **NOT** avoid these two dead spots completely.

Nonlinear filterings (NLF)

Office of Naval Research (around 1995)

Given the noisy observation of the real states, can we give the "accurate" estimates of the states instantaneously, provided as much computational resources as one needs?

It has been an OPEN question for more than 50 years. It is finally SOLVED theoretically in this talk.

Attempts

Attempts without much success:

- V. E. Beneš, 1981: derives an exact filter for a special class of nonlinear problems, so-called Beneš filter;
  – Does **not** include all linear problems.
- Around 1980, S. Mitter and R. Brockett proposed to use Lie algebra method to solve NLF. Finite dimensional Lie algebra will give finite dimensional filter. In a series of papers, Yau with his various collaborators classify all finite dimensional nonlinear filters of maximal rank.

Popular NLFs

Widely used NLFs nowadays:

Existing filters	*Shortcomings*
Assumed-density filter (extended Kalman filter)	Nonlinearity
	Gaussian assumption of initial state
Sequential Monte Carlo methods (particle filter)	Can't be implemented in real time

Signal based model

We consider the following signal based model:

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

where

- x_t: states, n-vector;
- f: drift term, n-vector;
- G: diffusion term, $n \times r$ matrix;
- y_t: observation path, m-vector;
- h: observation term, m-vector;
- v_t: r-vector Brownian motion with $E[dv_t dv_t^T] = Qdt$;
- w_t: m-vector Brownian motion with $E[dw_t dw_t^T] = Sdt$ and $S > 0$.

Assume $y_0 = 0$ and x_0, $\{v_t, t \geq 0\}$, $\{w_t, t \geq 0\}$ are independent.

Duncan-Mortensen-Zakai (DMZ) equation

1960s, Duncan, Mortensen and Zakai:

$\sigma(x,t)$: unnormalized density function of x_t conditioned on the observation history $Y_t = \{y_s : 0 \leq s \leq t\}$.

It satisfies the DMZ equation:

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

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

$$\mathcal{L}(*) \equiv \frac{1}{2}\sum_{i,j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j}\left[\left(GQG^T\right)_{ij} *\right] - \sum_{i=1}^{n} \frac{\partial (f_i *)}{\partial x_i}. \tag{3}$$

"Pathwise-robust" DMZ equation

Construct robust state estimators from any observed sample paths:

For each "given" observation y_t, let (Rozovsky, 1972)

$$\sigma(x,t) = \exp\left[h^T(x)S^{-1}y_t\right]\rho(x,t),$$

it yields the "pathwise-robust" DMZ equation:

$$\begin{cases} \dfrac{\partial \rho}{\partial t}(x,t) + \dfrac{\partial}{\partial t}(h^T S^{-1})^T y_t \rho(x,t) \\ \qquad\qquad = \exp\left(-h^T S^{-1} y_t\right)\left[\mathcal{L} - \dfrac{1}{2}h^T S^{-1} h\right]\left[\exp\left(h^T S^{-1} y_t\right)\rho(x,t)\right] \\ \rho(x,0) = \sigma_0(x). \end{cases} \tag{4}$$

"Pathwise-robust" DMZ equation

we shall assume that the observations arrive at discrete time $\tau_k = i\Delta t$, $i = 0, 1, \cdots, N_T$ and $\Delta t = T/N_T$. Let ρ_i be the solution of the "pathwise-robust" DMZ equation with y_t freezed at $t = \tau_{i-1}$, for $\tau_{i-1} \leq t \leq \tau_i$, $i = 1, 2, \cdots, N_T$, i.e.,

$$\begin{cases} \frac{\partial \rho_i}{\partial t}(x,t) + \frac{\partial}{\partial t}\left(h^T S^{-1}\right)^T y_{\tau_{i-1}} \rho_i(x,t) \\ \quad = \exp\left(-h^T S^{-1} y_{\tau_{i-1}}\right)\left[\mathcal{L} - \frac{1}{2} h^T S^{-1} h\right]\left[\exp\left(h^T S^{-1} y_{\tau_{i-1}}\right)\rho_i(x,t)\right] \\ \rho_1(x,0) = \sigma_0(x) \\ \rho_i\left(x, \tau_{i-1}\right) = \rho_{i-1}\left(x, \tau_{i-1}\right), \text{for } i = 2, 3, \cdots, N_T. \end{cases} \tag{5}$$

It is proved in that [1], in both pointwise and L^2 sense,

$$\lim_{\Delta t \to 0} \sum_{i=1}^{N_T} 1_{[\tau_{i-1}, \tau_i]}(t) \rho_i(x,t) \to \rho(x,t). \tag{6}$$

Therefore $\rho_i(x,t)$ is a good approximation of $\rho(x,t)$.

[1] Yau and Yau, *SIAM J. Control Optim.*, 2008

Kalman-Bucy filter

When system (1) is linear, i.e.,

$$\begin{cases} dx_t = Fx_t dt + Gdv_t, \\ dy_t = Hx_t dt + dw_t, \end{cases} \tag{7}$$

where the initial state x_0 is Gaussian, it can be checked that the conditional density function $p(x_t|Y_t)$ is Gaussian which is totally determined by its conditional mean and covariance matrix.
Let $m_t = E[x_t|Y_t], P_t = E[(x_t - m_t)(x_t - m_t)^T|Y_t]$ and their evolution equations are given by Kalman-Bucy filter:

$$\begin{cases} dm_t = Fm_t dt + P_t M^T S^{-1}(dy_t - Hm_t dt), \\ dP_t/dt = FP_t + P_t F^T + GQG^T - P_t H^T S^{-1} HP_t. \end{cases} \tag{8}$$

The number of the sufficient statistics which determine $p(x_t|Y_t)$ is quadratic w.r.t. the state dimension n.

Introduction to estimation algebra

1970s: Brockett and Clark, Brockett, and Mitter proposed estimation algebras method
1983 (International Congress of Mathematics): Brockett proposed the problem of classifying finite dimensional estimation algebras (FDEA).

Advantages:

- It takes into account of geometrical aspects of the situation.
- As long as the estimation algebra is finite dimensional, the finite dimensional recursive filter can be constructed explicitly.
- Lie algebraic methods are highly useful for classifying equivalence of finite dimensional filters.
- The number of sufficient statistics in the Lie algebra method linearly depends on state space dimension.

Basic concept

If noises in state and observation equations are independent standard Brownian motions, then we define operator:

$$L_0 := \frac{1}{2}\sum_{i=1}^{n}\frac{\partial^2}{\partial x_i^2} - \sum_{i=1}^{n} f_i\frac{\partial}{\partial x_i} - \sum_{i=1}^{n}\frac{\partial f_i}{\partial x_i} - \frac{1}{2}\sum_{i=1}^{m} h_i^2. \qquad (9)$$

For $i = 1, \cdots, m$, L_i is defined as zero degree differential operator of multiplication by h_i. Let $\eta := \sum_{i=1}^{n}\frac{\partial f_i}{\partial x_i} + \sum_{i=1}^{n} f_i^2 + \sum_{i=1}^{m} h_i^2$ and $D_i := \frac{\partial}{\partial x_i} - f_i,\ 1 \le i \le n$.

Definition 1

The estimation algebra E of a filtering system (1) is defined to be the Lie algebra generated by $\{L_0, L_1, \cdots, L_m\}$, i.e., $E = \langle L_0, h_1, \cdots, h_m\rangle_{L.A.}$. Furthermore, if $f = \nabla\phi$ for some $\phi \in C^\infty(\mathbb{R}^n)$, E is called exact.

Basic concept

Definition 2

Let $L(E) \subset E$ be the vector space consisting of all the homogeneous degree 1 polynomials in E. Then the linear rank of estimation algebra E is defined by $r := \dim L(E)$. Especially, if $r = n$, we call E has maximal rank.

Based on the structure of linear rank, classifications of estimation algebra have always been a research hotspot. Especially, from 1990 to 1997, in the series work of Yau and coworkers[2][3][4], complete classification of maximal rank estimation algebra has been finished[5],.

[2] Chen and Yau, *Math. Control Signals Systems*, 1996
[3] Chiou and Yau, *SIAM J. Control Optim.*, 1994
[4] Yau, *J. Math. Systems Estim. Control*, 1994
[5] Yau, *Internat. J. Control*, 2003

Classification of maximal rank estimation algebra

More precisely, Yau and his coworkers have proved the following theorem[6].

Theorem 3 (Complete classification)

Suppose that the state space of the filtering system is of dimension n. If E is the finite dimensional estimation algebra with maximal rank, then $f = (\alpha_1, \cdots, \alpha_n) + \nabla\phi$, where ϕ is a smooth function and $\alpha_i, 1 \leq i \leq n$, are affine functions and E is a real vector space of dimension $2n+2$ with basis given by $1, x_1, \cdots, x_n, D_1, \cdots, D_n, L_0$.

As an immediate result, Mitter conjecture holds for maximal rank FDEA, which states any function in E is an affine function.

[6] Yau, *Internat. J. Control*, 2003

Finite dimensional filter

In fact, finite dimensional filters can be constructed from finite dimensional estimation algebra. Followed by complete classification, the robust DMZ equation admits a solution for all $t \geq 0$ of the form:

$$\rho(t,x) = e^{T(t)} e^{r_n(t)x_n} \dots e^{r_1(t)x_1} e^{s_n(t)D_n} \dots e^{s_1(t)D_1} e^{tL_0} \sigma_0, \quad (10)$$

where $T(t), r_i(t), s_j(t)$ satisfy a system of ordinary differential equations (ODEs). The solvability of this set of ODEs for $t \geq 0$ is implied by solvability of estimation algebra. The following corollary is immediately obtained:

Corollary 4 (Sufficient statistics)

Suppose E is maximal rank finite dimensional estimation algebra on state dimension n. Then the number of sufficient statistics in order to compute the conditional density by Lie algebraic method is $2n$.

Progress of classification of non-maximal rank case

At the beginning of 20th century, classification of non-maximal rank estimation algebras becomes a very important and difficult open problem.

- 2006: Classification of estimation algebra with state dimension 2 (Wu and Yau)[7];
- 2018: Linear structure of Ω and Mitter conjecture of state dimension 3, rank 2 case (Shi and Yau)[8];
- 2020: Existence of novel finite dimensional filters (Jiao and Yau)[9].
- 2022: Classification with state dimension n, linear rank $n-1$ and constant Ω (Yu, Jiao and Yau).[10]

[7] Wu and Yau, *SIAM J. Control Optim.*, 2006
[8] Shi and Yau, *Internat. J. Control*, 2020
[9] Jiao and Yau, *SIAM J. Control Optim.*, 2020
[10] Yu, Jiao and Yau, *IEEE Trans. Automat. Contr.*, 2023

Introduction

- In 2008 [1], Yau and Yau show that the DMZ equation (4) admits a unique nonnegative weak solution ρ which can be approximated by a solution ρ_R of the DMZ equation on the ball B_R with $\rho_R|_{\partial B_R} = 0$. The error of this approximation is bounded by a function of R which tends to zero as R goes to infinity. The solution ρ_R can in turn be approximated efficiently by an algorithm depending only on solving the observation-independent Kolmogorov equation on B_R.

- In 2013 [11], Luo and Yau extend the algorithm developed previously by Yau and Yau to the most general setting of nonlinear filterings, where the explicit time-dependence is in the drift term, observation term, and the variance of the noises could be a matrix of functions of both time and the states.

  There are some works investigating Hermite spectral method [12], proper orthogonal decomposition method [13] and Legendre spectral method [14], to numerically solve the forward Kolmogorov equation which help to solve the DMZ equation.

[11] Luo and Yau, *IEEE Trans. Automat. Contr.*, 2013a
[12] Luo and Yau, *IEEE Trans. Automat. Contr.*, 2013b
[13] Wang, Luo, Yau and Zhang, *IEEE Trans. Automat. Contr.*, 2020
[14] Dong, Luo and Yau, *IEEE Trans. Automat. Contr.*, 2013

Kolmogorov forward equation

Proposition 1

[1] *For each* $\tau_{i-1} \leq t < \tau_i$, $i = 1, \cdots, N_T$, $\rho_i(x,t)$ *satisfies* (5) *if and only if*

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

satisfies the Kolmogorov forward equation (KFE)

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

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

Yau-Yau algorithm

- **Step 1 (Off-line)** $u_i(x, \tau_{i-1}) \rightarrow u_i(x, \tau_i)$**:** Solve the KFE (12) at the time interval $[\tau_{i-1}, \tau_i]$ with initial value $u_i(x, \tau_{i-1})$. Let us denote by $\mathcal{U}$ the semi-group associated with the KFE (12), then the solution of (12) can be expressed as

$$u_i(x, \tau_i) = \mathcal{U}(\Delta t)\, u_i(x, \tau_{i-1}) \tag{13}$$

  which can be computed off-line and stored.

- **Step 2 (On-line) $u_i(x,\tau_i) \to u_{i+1}(x,\tau_i)$:** When the new observation y_{τ_i} arrives at time τ_i, we need to update the initial value of $u_{i+1}(x,t)$ in the time interval $[\tau_i, \tau_{i+1})$. For $t \in [0,\tau_1)$, the initial value is $u_1(x,0) = \sigma_0(x)$. At time $t = \tau_i$, when the latest observation y_{τ_i} is available,

$$\begin{aligned} u_{i+1}(x,\tau_i) &\overset{(11)}{=} \exp\left[h^T(x) S^{-1} y_{\tau_i}\right] \rho_{i+1}(x,\tau_i) \\ &\overset{(5)(11)}{=} \exp\left[h^T(x) S^{-1}\left(y_{\tau_i} - y_{\tau_{i-1}}\right)\right] \\ &\quad \cdot u_i(x,\tau_i), \end{aligned} \tag{14}$$

since $y_0 = 0$. And $u_i(x,\tau_i)$ is pre-computed in the step 1. Now we obtain the initial value of $u_{i+1}(x,\tau_i)$ in the time interval $[\tau_i, \tau_{i+1})$, for each $i = 0, 1, \cdots, N_T$.

Uniform framework of Yau-Yau algorithm

With chosen complete basis $\{\phi_l(x)\}_{l=1}^{\infty}$, $u_{i+1}(x,\tau_i)$ is uniquely determined by its coordinate $(a_{i+1,l})_{l=1}^{\infty}$, and can be approximated by

$$u_{i+1}(x,\tau_i) \approx \sum_{l=1}^{N} a_{i+1,l}\phi_l(x), \tag{15}$$

with N large enough. For test function $\varphi : \mathbb{R}^n \to \mathbb{R}$, $E[\varphi(x_{\tau_i})|\mathcal{Y}_{\tau_i}]$ can be approximately represented by a function of $a_{i+1} = (a_{i+1,1}, \cdots, a_{i+1,N})$:

$$E[\varphi(x_{\tau_i})|\mathcal{Y}_{\tau_i}] = \frac{a_{i+1}^{\top}\beta_{\varphi}}{a_{i+1}^{\top}\beta_1}, \tag{16}$$

where $\beta_{\varphi} = (\beta_{\varphi,1}, \cdots, \beta_{\varphi,N})^{\top}$ and $\beta_1 = (\beta_{1,1}, \cdots, \beta_{1,N})^{\top}$ are constant vectors with

$$\beta_{\varphi,l} = \int_{\mathbb{R}^n} \varphi(x)\phi_l(x)dx, \ \beta_{1,l} = \int_{\mathbb{R}^n} \phi_l dx, l = 1, \cdots, N.$$

The evolution of $u_{i+1}(x, \tau_i)$ can also be described by the evolution of the coordinates a_{i+1} and can be computed recursively. In fact, let us denote by $\mathcal{U}$ the semi-group associated with the FKE (12), then

$$\begin{aligned} & u_{i+1}(x, \tau_i) \\ & = \exp\left[h^\top S^{-1}(y_{\tau_i} - y_{\tau_{i-1}})\right] u_i(x, \tau_i) \\ & = \exp\left[h^\top S^{-1}(y_{\tau_i} - y_{\tau_{i-1}})\right] \mathcal{U}(\Delta t) u_i(x, \tau_{i-1}). \end{aligned} \tag{17}$$

If we project all the computations onto the vector space spanned by $\{\phi_l\}_{l=1}^N$, then

$$u_{i+1}(x,\tau_i) \approx \sum_{j=1}^{N} a_{i+1,j}\phi_j(x),$$

$$\mathcal{U}(\Delta t)\phi_l(x) \approx \sum_{j=1}^{N} d_{l,j}(\Delta t)\phi_j(x),$$

$$\exp\left[h^\top(x)S^{-1}(y_{\tau_i}-y_{\tau_{i-1}})\right]\phi_l(x) \approx \sum_{j=1}^{N} r_{l,j}(y_{\tau_i}-y_{\tau_{i-1}})\phi_j(x).$$

where $d_{l,j}(\Delta t)$ and $r_{l,j}(y_{\tau_i}-y_{\tau_{i-1}})$ are coefficients determined by Δt and $y_{\tau_i}-y_{\tau_{i-1}}$, respectively.
According to (17), the evolution of the coordinates a_{i+1} is given by

$$a_{i+1,l} = \sum_{j=1}^{N}\sum_{k=1}^{N} a_{i,j}d_{j,k}(\Delta t)r_{k,l}(y_{\tau_i}-y_{\tau_{i-1}}), l = 1,\cdots,N. \quad (18)$$

Algorithm 1 The Uniform Framework of Yau-Yau Algorithm

Off-line Computation

Compute $d_{l,j}(\Delta t)$ for each $1 \leq l, j \leq N$, and $\beta_{\varphi}^{N}, \beta_{1}^{N}$.

Initialization

Compute $\left(a_1^N\right)^T$ by $u_1(x, \tau_0) = \sigma_0(x)$.

On-line Computation

for $i = 1$ to N_T **do**

Compute $r_{l,j}\left(y_{\tau_i} - y_{\tau_{i-1}}\right)$, for each $1 \leq l, j \leq N$.

Compute

$$a_{i+1,l} = \sum_{j=1}^{N} \sum_{k=1}^{N} a_{i,j} d_{j,k}(\Delta t) r_{k,l}(y_{\tau_i} - y_{\tau_{i-1}}), l = 1, \cdots, N.$$

Compute $\hat{\varphi}(x_{\tau_i}) = E[\varphi(x_{\tau_i})|\mathcal{Y}_{\tau_i}] = \dfrac{a_{i+1}^{\top} \beta_{\varphi}}{a_{i+1}^{\top} \beta_1}$.

end for

How to choose basis?

Choose the optimal basis $\{\phi_l(x)\}$

$\Updownarrow$

Choose the optimal coordinate $\{a_{i,l}(x)\}$

In the existing numerical implementations of Yau-Yau algorithm, they first determine basis $\{\phi_l(x)\}$, and then compute coordinate $\{a_{i,l}(x)\}$. If we can choose optimal basis, then we can certainly obtain better results. And this idea can be achieved by using deep learning. Instead of directly pursue optimal basis functions directly, we aim to obtain optimal coordinate $\{a_{i,l}(x)\}$, which is totally determined by the system and basis functions.

Feedforward neural networks

Let $\Sigma^{r,N}(\kappa)$ be the class of functions

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

where $\kappa : \mathbb{R} \to [0, 1]$ is the activation function and A_j is affine function. Apparently, $\bar{\zeta}$ represents the standard three-layered feedforward network with r input-neurons, q hidden-neurons and N output-neuron, which is shown in Fig. 1. 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[15].

[15] Hornik, Stinchcombe and White, *Neural Netw.* 1989

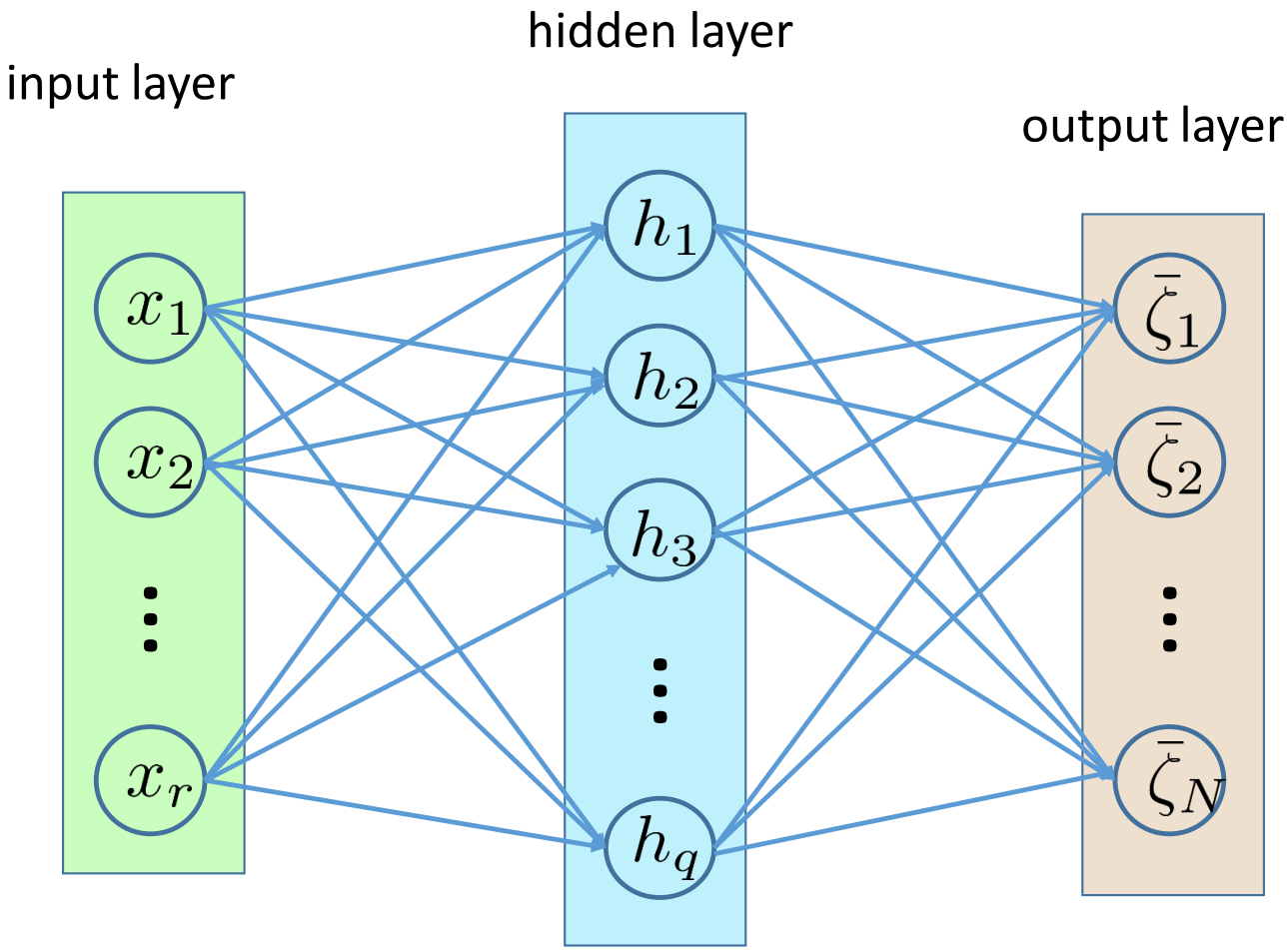


Figure 1: Three-layered feedforward network with r input-neurons, q hidden-neurons and N output-neuron.

Recurrent neural network (RNN)

Mathematically, recurrent neural network (RNN) is a class of functions defined as follows.

Definition 5

For any $r_1, r_2, r_3 \in \mathbb{N}$, the recurrent neural network $RNN^{r_1,r_2,r_3}(\kappa)$ is a class of functions with the following state space model form:

$$\begin{cases} \tilde{s}_{i+1} = \tilde{\eta}(\tilde{s}_i, \alpha_{i+1}), \\ \tilde{\beta}_i = \tilde{\xi}(\tilde{s}_i), \end{cases} \tag{19}$$

where $\alpha_i \in \mathbb{R}^{r_1}$ is the input, $\tilde{s}_i \in \mathbb{R}^{r_2}$ is the hidden state, $\tilde{\beta}_i \in \mathbb{R}^{r_3}$ is the output, and $\bar{\eta} : \mathbb{R}^{r_2} \times \mathbb{R}^{r_1} \to \mathbb{R}^{r_2}$, $\bar{\xi} : \mathbb{R}^{r_2} \to \mathbb{R}^{r_3}$ are feedforward neural networks with squashing function κ.

Universal Approximation Theorem for RNN

Theorem 6

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 inputs* $\alpha_i \in \mathbb{R}^{r_1}$*, the inner state* $s_i \in \mathbb{R}^{r_2}$*, and the output* $\beta_i \in \mathbb{R}^{r_3}$*,* $i = 1, 2, \cdots N_T$*. Assume that* (s_i, α_i) *are contained in a compact set* $\mathscr{K} \subset \mathbb{R}^{r_2} \times \mathbb{R}^{r_1}$*, for all* $i = 1, \cdots, N_T$*. Then, any open dynamical system of the form*

$$\begin{cases} s_{i+1} = \eta(s_i, \alpha_{i+1}), \\ \beta_i = \xi(s_i), \end{cases} \tag{20}$$

can be approximated by a function 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 $\tilde{\xi}$, which determine the RNN system (19) with the same input $\{\alpha_i, 1 \leq i \leq N_T\}$ of (20), such that

$$\begin{aligned} |s_i - \tilde{s}_i| &< \varepsilon, \\ |\beta_i - \tilde{\beta}_i| &< \varepsilon, \end{aligned} \tag{21}$$

for all $1 \leq i \leq N_T$, where $\tilde{s}_i$ and $\tilde{\beta}_i$ are the state and output of the RNN system (19), respectively.[a]

[a] Schäfer and Zimmermann *Int. J. Neural Syst.* 2007

RNN and Yau-Yau algorithm

The uniform framework of Yau-Yau algorithm is also illustrated in the left half of Fig. 2. Comparing the left half of Fig. 2 with the framework of RNN which is shown in the right half of Fig. 2, one natural idea is that we can use RNN to approximate the evolution of the coordinates a_i. Now rewrite (16) and (18) as the following dynamical system:

$$\begin{cases} a_{i+1} = \bar{\eta}\left(a_i, y_{\tau_i} - y_{\tau_{i-1}}\right), \text{ coordinate transition} \\ \hat{\varphi}(x_{\tau_i}) = \bar{\xi}\left(a_{i+1}\right), \text{ output equation} \end{cases} \tag{22}$$

The filtering algorithm based on RNN and this new framework (22) of the Yau-Yau algorithm is called RNN based Yau-Yau filter (RNNYYF).

RNN and Yau-Yau algorithm

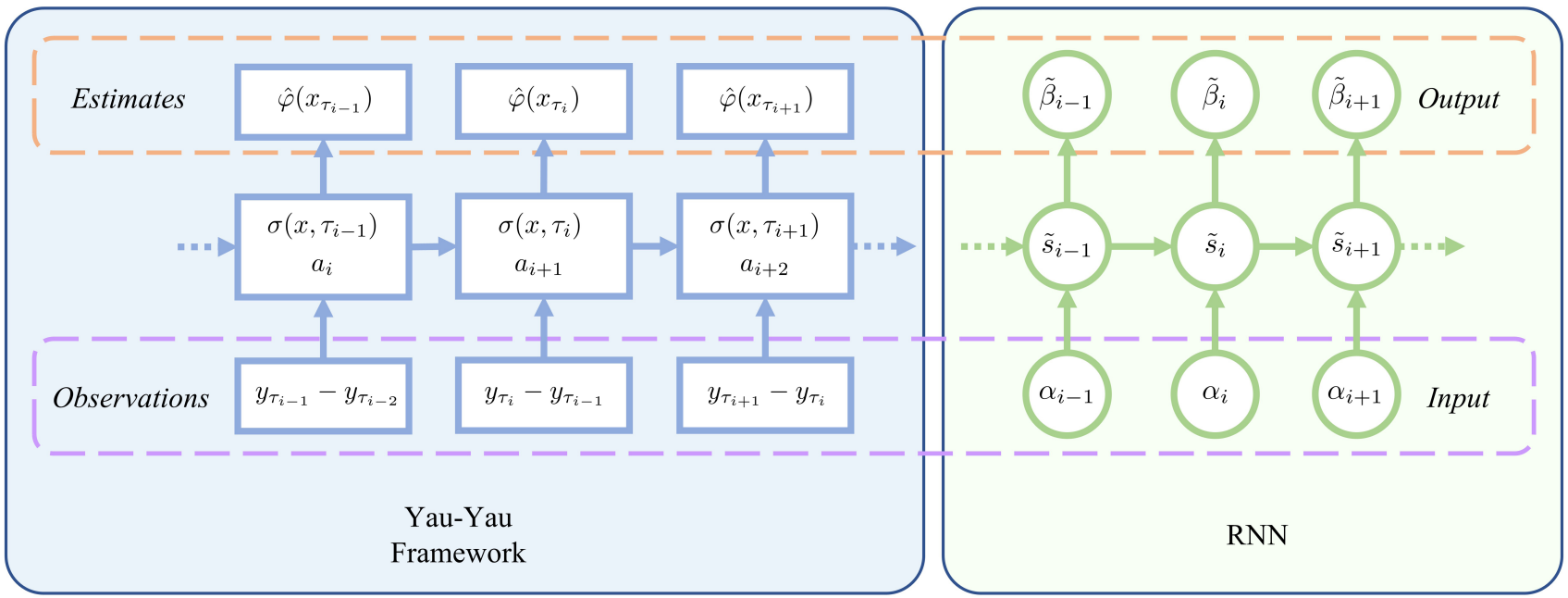


Figure 2: The connection between Yau-Yau framework and RNN

Change of measure

The reference probability measure $\tilde{P}$ is given by

$$\frac{d\tilde{P}}{dP}\bigg|_{\mathcal{F}_t} = \exp\bigg(- \int_0^t h(x_s)^\top S^{-1} dw_s - \frac{1}{2}\int_0^t h(x_s)^\top S^{-1} h(x_s) ds \bigg), \tag{23}$$

and it has been shown that the unnormalized probability density function $\sigma(x,t)$ is the density of the measure ρ_t, defined by

$$\rho_t(\varphi) = E\bigg[\varphi(X_t)\frac{d\tilde{P}}{dP}\bigg|\mathcal{Y}_t\bigg], \tag{24}$$

for all smooth functions φ defined on $\mathbb{R}^d$ with bounded derivatives.

Overcome the curse of dimensionality

Theorem 7

Consider the functions $u_{i+1}(x, \tau_i)$, $i = 0, 1, \cdots, N_T$ *defined in the Yau-Yau algorithm. There exists a set of* N *normalized square-integrable functions,* $\{\phi_j\}_{j=1}^N \subset L^2(B_R)$, *which are orthogonal to each other, such that for each* $i = 0, \cdots, N_T$, *we can find a function* $\tilde{v}_i(x)$ *in the* N*-dimensional vector space spanned by* $\{\phi_j\}_{j=1}^N$, *which satisfies*

$$\begin{aligned} &\tilde{E} \int_{B_R} |\tilde{v}_i(x) - u_{i+1}(x, \tau_i)| dx \leq C_T \sqrt{\Delta t}, \\ &\forall i = 0, 1, \cdots, N_T. \end{aligned} \tag{25}$$

where C_T is a constant which depends on T, R, and the coefficients in the filtering system, but does not depend directly on the dimension of the filtering system, n, m, or the time discretization step Δt. Here, the notation $\tilde{E}$ means that the expectation is taken with respect to the reference probability measure $\tilde{P}$ which is defined by (23).
Next, if we represent $\tilde{v}_i$ by

$$\tilde{v}_i(x) = \sum_{j=1}^{N} a_{i+1,j}\phi_j(x), \tag{26}$$

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

$$a_{i+1} = \eta(a_i, y_{\tau_i} - y_{\tau_{i-1}}), \quad i = 1, \cdots, N_T, \tag{27}$$

with a given initial value a_1, where $\eta : \mathbb{R}^N \times \mathbb{R}^m \to \mathbb{R}^N$ is a continuous function with respect to $a_i \in \mathbb{R}^N$ and $y_{\tau_i} - y_{\tau_{i-1}} \in \mathbb{R}^m$, and is time-invariant.
Moreover, the number N of the functions in the set $\{\phi_j\}_{j=1}^N$ can be selected to be of at most polynomial growth with respect to the dimension m, which shows the capability of this framework of the Yau-Yau algorithm to overcome the *curse of dimensionality*.

Error of RNN based Yau-Yau filter

Corollary 8

Consider the dynamical system (22) derived from the uniform framework of Yau-Yau algorithm. Let $M_1 > 0$ be a given constant and

$$\Omega_{1,M_1} = \{\omega : \sup_{0\le t\le T} |y_t| < M_1\} \tag{28}$$

Then, for any $\epsilon > 0$, there exists a recurrent neural network system given by

$$\begin{cases} \tilde{a}_{i+1} = \tilde{\eta}(\tilde{a}_i, y_{\tau_i} - y_{\tau_{i-1}}), \\ \tilde{\varphi}_i = \tilde{\xi}(\tilde{a}_{i+1}), \end{cases} \quad i = 1, \cdots, N_T, \tag{29}$$

such that

$$\tilde{E}\left[1_{\Omega_{1,M_1}} \sup_{1\le i\le N_T} |\hat{\varphi}_i - \tilde{\varphi}_i|\right] < \epsilon. \tag{30}$$

That is to say, the open dynamical system (22) can be approximated by an RNN system with arbitrary accuracy.

Algorithm

Let θ denote all parameters we need to train in the RNNYYF which contains two parts:

$$\begin{cases} \tilde{a}_{i+1} = \Phi\left(\tilde{a}_i, y_{\tau_i} - y_{\tau_{i-1}}; \theta_1\right), \\ \hat{\varphi}(x_{\tau_i}) = \Gamma\left(\tilde{a}_{i+1}; \theta_2\right), \end{cases} \tag{31}$$

where $\theta = [\theta_1, \theta_2]$ represents all the trainable parameters in RNNYYF, Φ represents single-layered feedforward network with l hidden layer neurons with hyperparameter l to be determined, Γ is a linear function with input dimension l and output dimension is 1.

Naturally, we aim to minimize

$$L_0(\theta) := \frac{1}{K_1+1} E\left[\sum_{i=0}^{K_1} |\hat{\varphi}(x_{\tau_i}) - E\left[\varphi(x_{\tau_i})\,|\mathcal{Y}_{\tau_i}\right]|^2\right], \tag{32}$$

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

$$\begin{aligned} &E\left[|\varphi(x_{\tau_i}) - \hat{\varphi}(x_{\tau_i})|^2\right] \\ =&E\left[|\varphi(x_{\tau_i}) - \mathbb{E}\left[\varphi(x_{\tau_i})\,|\mathcal{Y}_{\tau_i}\right]|^2\right] + E\left[|\mathbb{E}\left[\varphi(x_{\tau_i})\,|\mathcal{Y}_{\tau_i}\right] - \hat{\varphi}(x_{\tau_i})|^2\right], \end{aligned}$$

it follows that

$$\text{argmin}_\theta L_0(\theta) = \text{argmin}_\theta L(\theta), \tag{33}$$

where

$$L(\theta) := \frac{1}{K_1+1} E\left[\sum_{i=0}^{K_1} |\hat{\varphi}(x_{\tau_i}) - \varphi(x_{\tau_i})|^2\right]. \tag{34}$$

Therefore, instead of data $\{y_{\tau_i}, E[\varphi(x_{\tau_i})\,|\mathcal{Y}_{\tau_i}]\}_{i\geq 0}$ where $E[\varphi(x_{\tau_i})\,|\mathcal{Y}_{\tau_i}]$ cannot be obtained in most cases, we only need data $\{y_{\tau_i}, \varphi(x_{\tau_i})\}_{i\geq 0}$ which can be easily generated from the system (1). We need to remark that this step is crucial since it allows us to get accessible data.

Simulation

we introduce the mean of the squared error (MSE) and the mean of the relative error (MRE) based on 100 realizations, which are defined as follows:

$$\begin{aligned} \text{MSE} &:= \frac{1}{100}\sum_{l=1}^{100}\frac{1}{K_2+1}\sum_{i=0}^{K_2}\left|x_{\tau_i}^{(l)}-\hat{x}_{\tau_i}^{(l)}\right|^2, \\ \text{MRE} &:= \frac{\sum_{l=1}^{100}\sum_{i=0}^{K_2}\left|x_{\tau_i}^{(l)}-\hat{x}_{\tau_i}^{(l)}\right|}{\sum_{l=1}^{100}\sum_{i=0}^{K_2}\left|x_{\tau_i}^{(l)}\right|}, \end{aligned} \tag{35}$$

where $x_{\tau_i}^{(l)}$ is the real state at time instant τ_i in the l-th experiment and $\hat{x}_{\tau_i}^{(l)}$ is the estimation of $x_{\tau_i}^{(l)}$, with $0 \leq i \leq K_2$, where $K_2 \in \mathbb{N}$ is the total time step.

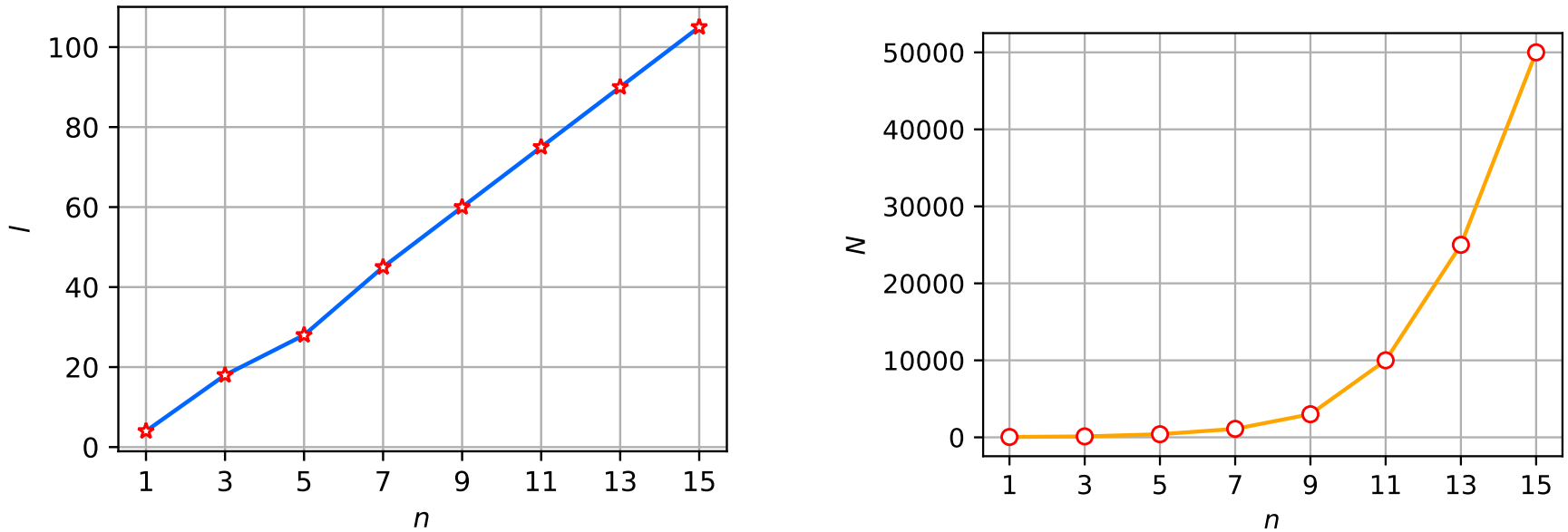
We consider the following cubic sensor filtering system:

$$\begin{cases} dx_t = (A_n x_t + \cos(x_t))dt + dv_t, \\ dy_t = x_t^3 dt + dw_t, \end{cases} \tag{36}$$

where $x_0 \sim \mathcal{N}(0, I_n)$ with identity matrix $I_n \in \mathbb{R}^n$, $n = 10$, v_t and w_t are Brownian motion processes with $E\left[dv_t dv_t^T\right] = I_n dt$ and $E\left[dw_t dw_t^T\right] = 0.01 I_n dt$, and $A_n = [a_{ij}]$ is a matrix with elements as follows:

$$a_{ij} = \begin{cases} 0.1, \text{ if } i + 1 = j, \\ -0.5, \text{ if } i = j, \\ 0, \text{ otherwise.} \end{cases}$$

For the sake of investigating the curse of dimensionality, we choose the state dimension $n = 1, 3, 5, 7, 9, 11, 12, 13, 15$, and choose proper number of particles in PF and hidden neurons in RNNYYF such that the MREs are the same in different dimensional cases. The results are displayed in Fig. 3. It can be seen that the number of the particles in PF grows exponentially with the dimension which implies that PF suffers from the curse of dimensionality. On the contrary, the number of the hidden neurons grows linearly with the dimension and this certainly verifies our theoretical result in Theorem 7.



(a) RNNYYF

(b) PF

Figure 3: The relationship between the number of hidden layer nodes or particles and the dimension of the state. Here, n is the dimension of the state and vertical axis represents the number of hidden layer nodes in RNNYYF or particles in PF. In all systems with different state dimension, the average relative errors of RNNYYF and PF are the same and MRE $= 0.22$.

When the dimension of the state is $n = 15$, the performance of EKF, PF with $N = 50000$ particles and RNNYYF with $l = 105$ hidden neurons in one experiment for (36) are shown in Figure 4, and the average performance of different methods based on 100 simulations are shown in Table 1. It can be seen that all these algorithms can track the real state while RNNYYF outperforms PF in costing time and outperforms EKF both in accuracy and costing time.

Table 1: The average performance of different methods based on 100 simulations for system (36).

Method	MSE	Running time (s)
RNNYYF	**1.03**	**0.0004**
EKF	1.19	0.0483
PF	**1.03**	118.3698

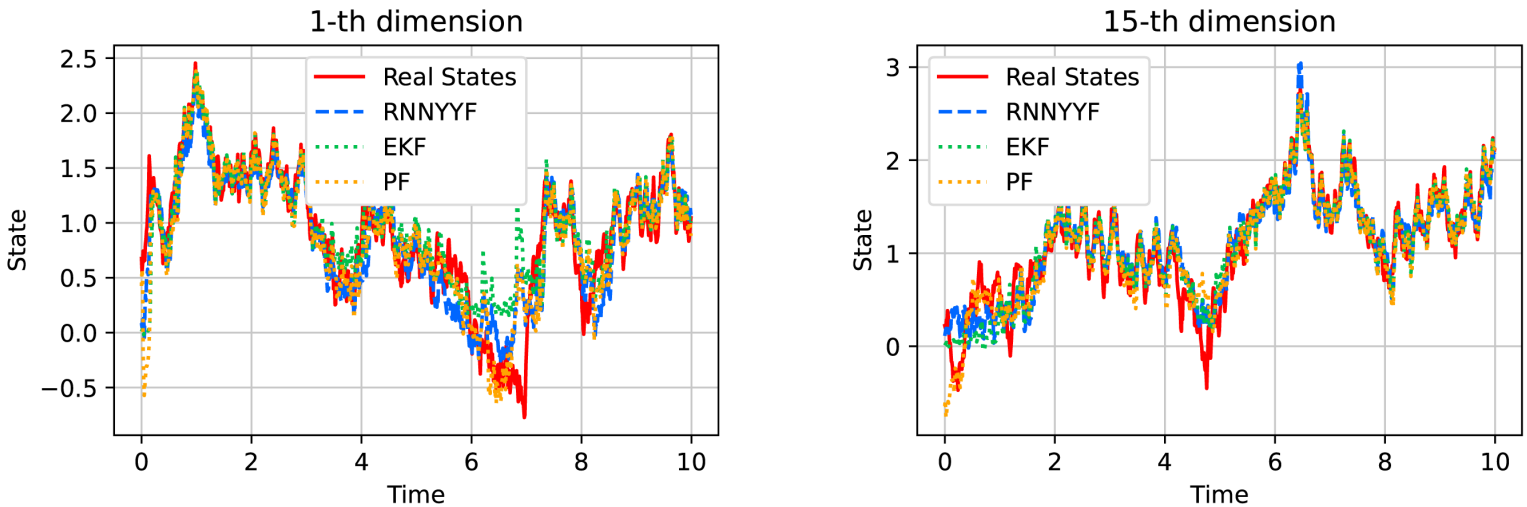


Figure 4: The estimation results of EKF, PF and RNNYYF in one experiment.

Summary

In this report, we investigate the curse of dimensionality problem in most existing filtering methods. It can be found that finite dimensional filters including Kalman-Bucy filter do not suffer from this problem. For the general filtering problems, we construct a novel filtering algorithm based on RNN and Yau-Yau filtering framework, and prove that the size of the neural network required in this algorithm only increases in polynomial (rather than exponentially) with respect to the dimension, which implies the proposed filtering algorithm has the capability to overcome the curse of dimensionality.

THANKS!