The Yau-Yau Method for Nonlinear Filtering Problems

Shing-Tung Yau

Department of Mathematics
Harvard University

2021

Joint work with Mei-Heng Yueh, Wen-Wei Lin, and Stephen Shing-Toung Yau

1. Introduction
   - Nonlinear Filtering Problems
   - Duncan–Mortensen–Zakai Equation
   - The Kolmogorov Equation

2. Numerical Method
   - Algorithm for the Filtering Model
   - Finite Difference Schemes
   - Numerical Experiments

Nonlinear Filtering Problems

- In 1961, Kalman and Bucy first established the finite-dimensional filter for the linear filtering model with Gaussian initial distribution, which is highly influential in modern industry.
- Since then filtering theory has proved useful in science and engineering, for example, the navigational and guidance systems, radar tracking, sonar ranging, and satellite and airplane orbit determination.
- Despite its usefulness, however, the Kalman–Bucy filter is not perfect.

Nonlinear Filtering Problems

- Its main weakness is that it is restricted to the linear dynamical system with Gaussian initial distribution.
- Therefore there has been tremendous interest in solving the nonlinear filtering problem which involves the estimation of a stochastic process

$$x = \{x_t\} \qquad \text{(called the signal or state process)}$$

that cannot be observed directly.

Nonlinear Filtering Problems

- Information containing x is obtained from observations of a related process

$$y = \{y_t\} \qquad \text{(the observation process).}$$

- The goal of nonlinear filtering is to determine

$$\text{the conditional density } \rho(t, x) \text{ of } x_t$$

given the observation history of $\{y_s : 0 \leq s \leq t\}$.

Nonlinear Filtering Problems

- In the late 1960s, Duncan, Mortensen, and Zakai independently derived the Duncan–Mortensen–Zakai (DMZ) equation for the nonlinear filtering theory, which the conditional probability density $\rho(t, x)$ has to satisfy.
- The central problem of nonlinear filtering theory is to solve the DMZ equation in real-time and in a memoryless way.

Nonlinear Filtering Problems

- In 2000, we proposed a novel algorithm to do just that[1].
- Under the assumptions that the drift terms $f_i(x)$, $1 \leq i \leq n$, and their first and second derivatives, and the observation terms $h_i(x)$, $1 \leq i \leq m$, and their first derivatives, have linear growth, we showed that the solution obtained from our algorithms converges to the true solution of the DMZ equation.

[1] S.-T. Yau and S. S.-T. Yau. Real time solution of nonlinear filtering problem without memory I. Mathematical Research Letters, 7:671–693, 2000.

Nonlinear Filtering Problems

- Although the above approach is quite successful, so far it cannot handle the famous cubic sensor in engineering in which
$$f(x) = 0 \quad \text{and} \quad h(x) = x^3.$$
- It is well known that there is no finite-dimensional filter for the cubic sensor.

Nonlinear Filtering Problems

- We later showed that under some mild conditions (which essentially say that the growth of $|h|$ is greater than the growth of $|f|$), the DMZ equation admits a unique nonnegative solution

$$u \in W_0^{1,1}((0,T) \times \mathbb{R}^n)$$

  which can be approximated by solutions u_R of the DMZ equation on the ball B_R with $u_R|_{\partial B_R} = 0$.
- The rate of convergence can be efficiently estimated in the L^1 norm.

Nonlinear Filtering Problems

- The solution u_R can in turn be approximated efficiently by an algorithm depending only on solving the time-independent Kolmogorov equation on B_R.
- Our algorithm can solve practically all engineering problems, including the cubic sensor problem in real-time and in a memoryless fashion.

Nonlinear Filtering Problems

- Specifically we show that the solution obtained from our algorithms converges to the solution of the DMZ equation in the L^1 sense.
- Equally important, we have a precise error estimate of this convergence, which is important in numerical computation.

Signal–Observation Model

The (nonlinear) filtering problem considered here is

$$\begin{cases} \mathrm{d}x(t) = f(x(t))\,\mathrm{d}t + \mathrm{d}v(t), & x(0) = x_0, \\ \mathrm{d}y(t) = h(x(t))\,\mathrm{d}t + \mathrm{d}w(t), & y(0) = 0 \end{cases} \tag{1}$$

where

- $x(t) \in \mathbb{R}^N$: state.
- $y(t) \in \mathbb{R}^M$: measurement.
- $f : \mathbb{R}^N \to \mathbb{R}^N$: (nonlinear) drift term.
- $h : \mathbb{R}^N \to \mathbb{R}^M$: (nonlinear) observation term.
- $v(t) \in \mathbb{R}^N$, $w(t) \in \mathbb{R}^M$: independent Brownian motions.

Signal–Observation Model

$x(t)$: signal / state

$y(t)$: observation / measurement

Goal

Estimate the state $x(t)$ by a given history of observations

$$\{y(\tau) \mid \tau \in [0, t]\}, \text{ for } t \in (0, T].$$

Duncan–Mortensen–Zakai (DMZ) Equation (1960s)

Let $\sigma(t,x)$ be an unnormalized conditional probability density of $x(t)$ given $\{y(\tau)|\tau \in [0,t]\}$. The DMZ equation is given by

$$\mathrm{d}\sigma(t,x) = L_0\sigma(t,x)\,\mathrm{d}t + \sum_{i=1}^{n} L_i\sigma(t,x)\,\mathrm{d}y_i(t), \qquad \sigma(0,x) = \sigma_0,$$

where

$$L_0 = \frac{1}{2}\sum_{i=1}^{n}\frac{\partial^2}{\partial x_i^2} - \sum_{i=1}^{n} f_i\frac{\partial}{\partial x_i} - \sum_{i=1}^{n}\frac{\partial f_i}{\partial x_i} - \frac{1}{2}\sum_{i=1}^{m} h_i^2,$$

and for $i = 1, \ldots, m$, L_i is the zero degree differential operator of multiplication by h_i.

The Robust DMZ Equation (Davis, 1980s)

Define the new unnormalized density as

$$u(t,x) = \exp\left(-\sum_{i=1}^{m} h_i(x)y_i(t)\right)\sigma(t,x).$$

The robust DMZ equation[2] is given by

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) = L_0 u(t,x) + \displaystyle\sum_{i=1}^{m} y_i(t)[L_0, L_i]u(t,x) \\ \qquad\qquad + \dfrac{1}{2}\displaystyle\sum_{i,j=1}^{m} y_i(t)y_j(t)[[L_0, L_i], L_j]u(t,x), \\ u(0,x) = \sigma_0, \end{cases}$$

where $[\cdot,\cdot]$ denotes the Lie bracket.

[2] M. H. A. Davis, On a multiplicative functional transformation arising in nonlinear filtering theory, Z. Wahrsch. Verw. Gebiete, 54:125–139, 1980.

The Robust DMZ Equation (Yau and Yau, 2000s)

The robust DMZ equation[1] is reformulated as

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) = \dfrac{1}{2}\triangle u(t,x) + (-f(x) + \nabla K(t,x)) \cdot \nabla u(t,x) \\ \qquad + \left(-\nabla \cdot f(x) - \dfrac{1}{2}|h(x)|^2 + \dfrac{1}{2}\triangle K(t,x) \right. \\ \qquad \left. - f(x) \cdot \nabla K(t,x) + \dfrac{1}{2}|\nabla K(t,x)|^2 \right) u(t,x), \\ u(0,x) = \sigma_0(x), \end{cases} \tag{2}$$

where $K = \sum_{i=1}^{m} y_i(t) h_i(x)$, $f = (f_1, \ldots, f_n)^\top$, $h = (h_1, \ldots, h_m)^\top$.

[1] S.-T. Yau and S. S.-T. Yau. Real time solution of nonlinear filtering problem without memory I. Mathematical Research Letters, 7:671–693, 2000.

The Robust DMZ Equation

Theorem (Yau and Yau[3])

Consider the filtering model (1). *For any* $T > 0$, *let* u *be a solution of the robust DMZ equation* (2) *in* $[0, T] \times \mathbb{R}^N$. *Assume*

$$-\frac{1}{2}|h|^2 - \frac{1}{2}\triangle K - f \cdot \nabla K + \frac{1}{2}|\nabla K|^2 + |f - \nabla K| \leq c_1. \tag{3}$$

Let $R \geq 1$ *and* u_R *be the solution of the following DMZ equation on the ball* B_R:

$$\sup_{0 \leq t \leq T} \int_{|x| \geq R} u(t, x) \leq e^{-\sqrt{1+R^2}} e^{(c_1 + \frac{N+1}{2})T} \int_{\mathbb{R}^N} e^{\sqrt{1+|x|^2}} u(0, x).$$

Remark

The above theorem says that we can choose a ball large enough to capture almost all the density.

[3] S.-T. Yau and S. S.-T. Yau. Real time solution of nonlinear filtering problem without memory II. SIAM. J. Control Optim., 47(1):163–195, 2008.

The Robust DMZ Equation

Theorem (Yau and Yau[3])

Consider the filtering model (1). *For any* $T > 0$, *let* u *be a solution of the robust DMZ equation* (2) *in* $[0, T] \times \mathbb{R}^N$. *Assume* (3) *holds and*

1. $-\frac{1}{2}|h|^2 - \frac{1}{2}\triangle K - f \cdot \nabla K + \frac{1}{2}|\nabla K|^2 + 12 + 2N + 4|f - \nabla K| \leq c_2$.
2. $e^{-\sqrt{1+|x|^2}}(12 + 2N + 4|f - \nabla K|) \leq c_3$.

Let $R \geq 1$ *and* u_R *be the solution of the robust DMZ equation* (2) *on the ball* B_R *with the boundary condition* $u_R(t, x) = 0$ *for* $(t, x) \in [0, T] \times \partial B_R$. *Then*

$$\left|\int_{B_{\frac{R}{2}}} u(T, x) - u_R(T, x)\right| \leq \frac{2(e^{c_2 T} - 1)}{c_2} c_3 e^{-\frac{9}{16}R} e^{(c_1 + \frac{N+1}{2})T} \int_{\mathbb{R}^N} e^{\sqrt{1+|x|^2}} u(0, x).$$

Remark

The above theorem says that we can approximate u by u_R.

[3] S.-T. Yau and S. S.-T. Yau. Real time solution of nonlinear filtering problem without memory II. SIAM. J. Control Optim., 47(1):163–195, 2008.

The Kolmogorov Equation (Yau and Yau, 2008)

The density $u(\tau_k, x)$ can be computed by $\widetilde{u}_k(\tau_k, x)$, where $\widetilde{u}_k(t, x)|_{[\tau_{k-1},\tau_k]}$ satisfies the Kolmogorov equation[3]

$$
\begin{cases}
\dfrac{\partial \widetilde{u}_k}{\partial t}(t, x) = \dfrac{1}{2} \triangle \widetilde{u}_k - f(x) \cdot \nabla \widetilde{u}_k - \left(\nabla \cdot f + \dfrac{1}{2}|h|^2 \right) \widetilde{u}_k, \ t \in [\tau_{k-1}, \tau_k], \\
\widetilde{u}_k(\tau_{k-1}, x) = \exp\left\{ (y(\tau_{k-1}) - y(\tau_{k-2})) \cdot h(x) \right\} \widetilde{u}_{k-1}(\tau_{k-1}, x), \\
\widetilde{u}_1(0, x) = \sigma_0(x) \exp\left\{ y(0) \cdot h(x) \right\}, k = 2, \ldots, N_\tau.
\end{cases}
$$

Then,

$$
u(\tau_k, x) = \exp\left(-\sum_{j=1}^{m} y_j(\tau_{k-1}) h_j(x) \right) \widetilde{u}_k(\tau_k, x).
$$

[3] S.-T. Yau and S. S.-T. Yau. Real time solution of nonlinear filtering problem without memory II. SIAM. J. Control Optim., 47(1):163–195, 2008.

Algorithm of Yau-Yau Method for the Filtering Model

Recall the N-dimensional filtering model (1):

$$\begin{cases} \mathrm{d}x(t) = f(x(t))\,\mathrm{d}t + \mathrm{d}v(t), & x(0) = x_0, \\ \mathrm{d}y(t) = h(x(t))\,\mathrm{d}t + \mathrm{d}w(t), & y(0) = 0. \end{cases}$$

1. A bounded domain Ω is given by an N-cell, e.g., $[-10, 10]^N$.
2. An initial density $u(0, s)$ is given by, e.g., $\mathcal{N}(x_0, 0.2)$.
3. Set $\tau_0 = 0$. Once a new measurement $y(\tau_k)$ is observed, we update
$$u(\tau_k, s_j) \leftarrow \exp\left\{[y(\tau_k) - y(\tau_{k-1})]^\top h(s_j)\right\} u(\tau_k, s_j).$$
4. For $t \in [\tau_k, \tau_{k+1})$, we solve the Kolmogorov on $\{t\} \times \Omega$:
$$\begin{cases} \dfrac{\partial u}{\partial t}(t, s) = \dfrac{1}{2}\triangle u(t, s) - f(s) \cdot \nabla u(t, s) - \left(\nabla \cdot f(s) + \dfrac{1}{2}\|h(s)\|_2^2\right), \\ u(t, \partial\Omega) = 0. \end{cases}$$
5. The estimate of the state is computed by $x(t) \approx \int_\Omega s\,\mathrm{d}u(t, s)$.

Finite Difference Schemes

Let $\{t_n\}_{n=1}^{N_t}$ and $\{s_j\}_{j=1}^{N_s}$ be a uniform partition of $[0, T]$ and $[-R, R]$ with step size Δt and Δs, respectively. Then

$$\begin{aligned}
\frac{\partial u}{\partial s^2}(t_n, s_j) = & \frac{\alpha}{(\Delta s)^2}\left(u(t_{n+1}, s_{j+1}) - 2u(t_{n+1}, s_j) + u(t_{n+1}, s_{j-1})\right) \\
& + \frac{1-\alpha}{(\Delta s)^2}\left(u(t_n, s_{j+1}) - 2u(t_n, s_j) + u(t_n, s_{j-1})\right) + O((\Delta s)^2), \\
\frac{\partial u}{\partial s}(t_n, s_j) = & \frac{\beta}{2\Delta s}\left(u(t_{n+1}, s_{j+1}) - u(t_{n+1}, s_{j-1})\right) \\
& + \frac{1-\beta}{2\Delta s}\left(u(t_n, s_{j+1}) - u(t_n, s_{j-1})\right) + O(\Delta s).
\end{aligned}$$

1. Explicit Euler Method: $\alpha = 0$, $\beta = 0$.
2. Implicit Euler Method: $\alpha = 1$, $\beta = 1$.
3. Quasi-Implicit Euler Method (QIEM): $\alpha = 1$, $\beta = 0$.
4. Crank-Nicolson Method: $\alpha = \frac{1}{2}$, $\beta = \frac{1}{2}$.

Discrete Operators

Let $\{t_n\}_{n=1}^{N_t}$ and $\{s_j\}_{j=1}^{N_s}$ be a uniform partition of $[0, T]$ and $[-R, R]$ with step size Δt and Δs, respectively. Then $\{s_j\}_{j=1}^{(N_s)^N} := \left(\{s_j\}_{j=1}^{N_s}\right)^N$ is a partition of $[-R, R]^N$. The discretization of $\triangle$ and $f \cdot \nabla u$ are defined by

$$L_{N_s}^{(N)} = \sum_{d=1}^{N} \left(\mathrm{I}_{(N_s)^{N-d}} \otimes \mathrm{L}_{N_s} \otimes \mathrm{I}_{(N_s)^{d-1}}\right), \tag{4}$$

and

$$K_{N_s}^{(N)} = \sum_{d=1}^{N} \left\{\mathrm{diag}\left(f_d(s_j)\right)_{j=1}^{(N_s)^N} \left(\mathrm{I}_{(N_s)^{N-d}} \otimes K_{N_s} \otimes \mathrm{I}_{(N_s)^{d-1}}\right)\right\}, \tag{5}$$

respectively, where I_{N_s} denotes the identity matrix of size $N_s \times N_s$ and

$$\mathrm{L}_{N_s} = \frac{1}{(\Delta s)^2} \begin{bmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}_{N_s \times N_s} \text{and } \mathrm{K}_{N_s} = \frac{1}{2(\Delta s)} \begin{bmatrix} 0 & 1 & & \\ -1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & -1 & 0 \end{bmatrix}_{N_s \times N_s}.$$

QIEM for Kolmogorov Equation (Yueh, Lin, Yau, 2014)

The QIEM for Kolmogorov equation is nonnegativity preserving[4]. Denote the vector $[u(t_n, s_j)]_{j=1}^{(N_s)^N}$ as u^n. The QIEM for Kolmogorov equation is formulated as

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{1}{2} L_{N_s}^{(N)} u^{n+1} + \left(K_{N_s}^{(N)} + Q_{N_s}^{(N)} \right) u^n.$$

It can be solved by the linear system

$$\left[I_{(N_s)^N} - \frac{\Delta t}{2} L_{N_s}^{(N)} \right] u^{n+1} = \left[I_{(N_s)^N} + (\Delta t) \left(K_{N_s}^{(N)} + Q_{N_s}^{(N)} \right) \right] u^n, \tag{6}$$

where $L_{N_s}^{(N)}$ and $K_{N_s}^{(N)}$ are defined in (4) and (5), respectively, and

$$Q_{N_s}^{(N)} = \operatorname{diag} \left(-\nabla \cdot f(s_j) + \frac{1}{2} \|h(s_j)\|^2 \right)_{j=1}^{(N_s)^N}.$$

[4] M.-H. Yueh, W.-W. Lin and S.-T. Yau, An efficient algorithm of Yau-Yau method for solving nonlinear filtering problems, Commun. Inf. Syst. 14(2): 111–134, 2014.

Discrete Sine Transform (DST)

The spectral decomposition of the matrix $\left[I_{(N_s)^N} - \frac{\Delta t}{2} L_{N_s}^{(N)}\right]$ in the linear system (6) is

$$\left[I_{(N_s)^N} - \frac{\Delta t}{2} L_{N_s}^{(N)}\right] = W_{N_s}^{(N)} \Lambda_{N_s} \left(W_{N_s}^{(N)}\right)^*,$$

where $W_{N_s}^{(N)} = \bigotimes_{d=1}^{N} W_{N_s}$ and $\Lambda_{N_s}^{(N)} = \sum_{d=1}^{N} \left(\mathrm{I}_{(N_s)^{N-d}} \otimes \Lambda_{N_s} \otimes \mathrm{I}_{(N_s)^{d-1}}\right)$ in which

$$W_{N_s} = \left[\sqrt{\frac{2}{N_s+1}} \sin\left(\frac{ij\pi}{N_s+1}\right)\right]_{i,j=1}^{N_s}$$

and

$$\Lambda_{N_s} = \mathrm{diag}\left(1 - \frac{2\Delta t}{(\Delta s)^2} \sin^2\left(\frac{i\pi}{2(N_s+1)}\right)\right)_{i=1}^{N_s}.$$

The complexity of the product of $W_{N_s}^{(N)}$ is $O((N_s)^N \log N_s)$.

DST for Solving the Linear System

Recall the spectral decomposition of the matrix

$$\left[I_{(N_s)^N} - \frac{\Delta t}{2} L_{N_s}^{(N)}\right] = W_{N_s}^{(N)} \Lambda_{N_s} \left(W_{N_s}^{(N)}\right)^* .$$

The matrix inverse

$$\left[I_{(N_s)^N} - \frac{\Delta t}{2} L_{N_s}^{(N)}\right]^{-1} = W_{N_s}^{(N)} \Lambda_{N_s}^{-1} \left(W_{N_s}^{(N)}\right)^* .$$

1. The matrix-vector product of $W_{N_s}^{(N)}$ costs $O((N_s)^N \log N_s)$.
2. The product of diagonal matrix $\Lambda_{N_s}^{-1}$ costs $O((N_s)^N)$.

The complexity for solving the linear system $\left[I_{(N_s)^N} - \frac{\Delta t}{2} L_{N_s}^{(N)}\right] u = b$ is

$$O((N_s)^N \log N_s).$$

High-Order Scheme of Discrete Laplacian

To reduce the number of grid point N_s, we apply the high-order difference scheme for the Laplacian operator $\triangle$, defined by

$$\overline{L}_{N_s}^{(N)} u(s) = \frac{1}{(\Delta s)^2} \sum_{\substack{j=0 \\ |\mathbf{k}|=\sqrt{j}\Delta s \\ \mathbf{k}\in\mathcal{Q}_\mathbf{s}}}^{N} \alpha_j u(s+\mathbf{k}), \tag{7}$$

where s, $k \in \mathbb{R}^N$, $\mathcal{Q}_s$ is the cube with the center at s and the side length $2\Delta s$, the coefficients $\alpha_j = 2^{2-N-j}$, $j = 1, \ldots, N$, and

$$\alpha_0 = -\sum_{j=1}^{D} \left(\alpha_j \# \left\{ \mathbf{k} \in \mathcal{Q}_\mathbf{s} \,\middle|\, |\mathbf{k}| = \sqrt{j}\Delta s \right\} \right).$$

Construction of High-Order Discrete Laplacian Matrix

The matrix $\overline{L}_{N_s}^{(N)} := \mathsf{S}_{N_s}^{(1,\ldots,N)}$ can be recursively defined by

$$\mathsf{S}_{N_s}^{(j,\ldots,k)} = \mathsf{I}_{N_s} \otimes \mathsf{S}_{N_s}^{(j,\ldots,k-1)} + \mathsf{J}_{N_s} \otimes \mathsf{S}_{N_s}^{(i+1,\ldots,k)}, \tag{8}$$

for $k = j+1, \ldots, N$, in which the matrices J_{N_s} and $\mathsf{S}_{N_s}^{(j)}$ are defined by

$$\mathsf{J}_{N_s} = \begin{bmatrix} 0 & 1 & & \\ 1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 0 \end{bmatrix}_{N_s \times N_s} \text{ and } \mathsf{S}_{N_s}^{(j)} = \begin{bmatrix} \alpha_{j-1} & \alpha_j & & \\ \alpha_j & \alpha_{j-1} & \ddots & \\ & \ddots & \ddots & \alpha_j \\ & & \alpha_j & \alpha_{j-1} \end{bmatrix}_{N_s \times N_s},$$

respectively, $j = 1, \ldots, N$.

Diagram for the Recursive Construction

$$\begin{array}{ccccccccc}
S^{(1)}_{N_s} & \xrightarrow{I_{N_s}\otimes} & S^{(1,2)}_{N_s} & \xrightarrow{I_{N_s}\otimes} & \cdots & \xrightarrow{I_{N_s}\otimes} & S^{(1,\dots,N-1)}_{N_s} & \xrightarrow{I_{N_s}\otimes} & S^{(1,\dots,N)}_{N_s} \\
 & J_{N_s}\otimes \nearrow & & J_{N_s}\otimes \nearrow & & J_{N_s}\otimes \nearrow & & J_{N_s}\otimes \nearrow & \\
S^{(2)}_{N_s} & \xrightarrow{I_{N_s}\otimes} & S^{(2,3)}_{N_s} & \xrightarrow{I_{N_s}\otimes} & \cdots & \xrightarrow{I_{N_s}\otimes} & S^{(2,\dots,N)}_{N_s} & & \\
 & J_{N_s}\otimes \nearrow & & J_{N_s}\otimes \nearrow & & J_{N_s}\otimes \nearrow & & & \\
\vdots & \vdots & \vdots & \vdots & \iddots & & & & \\
 & J_{N_s}\otimes \nearrow & & J_{N_s}\otimes \nearrow & & & & & \\
S^{(N-1)}_{N_s} & \xrightarrow{I_{N_s}\otimes} & S^{(N-1,N)}_{N_s} & & & & & & \\
 & J_{N_s}\otimes \nearrow & & & & & & & \\
S^{(N)}_{N_s} & & & & & & & &
\end{array}$$

Figure: Diagram for the recursive construction of $\overline{L}^{(N)}_{N_s} := S^{(1,\dots,N)}_{N_s}$.

Explicit Form of High-Order Laplacian

The diagram immediately leads to an explicit form of the matrix

$$\overline{L}_{N_s}^{(N)} = \sum_{d=1}^{N} \left[\sum_{\substack{\#\{i|\mathrm{X}_i=\mathrm{I}_{N_s}\}=N-d \\ \#\{i|\mathrm{X}_i=\mathrm{J}_{N_s}\}=d-1}} \left(\bigotimes_{\substack{i=1 \\ \mathrm{X}_i\in\{\mathrm{I}_{N_s},\mathrm{J}_{N_s}\}}}^{N-1} \mathrm{X}_i \right) \otimes \mathrm{S}_{N_s}^{(d)} \right],$$

in which I_{N_s} denotes the identity matrix of size $N_s \times N_s$,

$$\mathrm{J}_{N_s} = \begin{bmatrix} 0 & 1 & & \\ 1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 0 \end{bmatrix}_{N_s\times N_s} \quad \text{and} \quad \mathrm{S}_{N_s}^{(j)} = \begin{bmatrix} \alpha_{j-1} & \alpha_j & & \\ \alpha_j & \alpha_{j-1} & \ddots & \\ & \ddots & \ddots & \alpha_j \\ & & \alpha_j & \alpha_{j-1} \end{bmatrix}_{N_s\times N_s},$$

$j = 1, \ldots, N.$

DST of High-Order Laplacian

The spectral decomposition of $\overline{L}_{N_s}^{(N)}$ is

$$\overline{L}_{N_s}^{(N)} = W_{N_s}^{(N)} \overline{\Lambda}_{N_s}^{(N)} \left(W_{N_s}^{(N)} \right)^* ,$$

where $\mathbf{W}_{N_s}^{(N)} = \left(\bigotimes_{d=1}^{N} W_{N_s} \right)$ and $\overline{\Lambda}_{N_s}^{(N)}$ is the diagonal matrix of the form

$$\overline{\Lambda}_{N_s}^{(N)} = \sum_{d=1}^{N} \left[\sum_{\substack{\#\{i | \mathrm{X}_i = \mathrm{I}_{N_s}\} = N-d \\ \#\{i | \mathrm{X}_i = \Gamma_{N_s}\} = d-1}} \left(\bigotimes_{\substack{i=1 \\ \mathrm{X}_i \in \{\mathrm{I}_{N_s}, \Gamma_{N_s}\}}}^{N-1} \mathrm{X}_i \right) \otimes \Theta_{N_s}^{(d)} \right] ,$$

where $\Theta_{N_s}^{(d)} = \alpha_d \Lambda_{N_s} + (2\alpha_d + \alpha_{d-1}) \, \mathrm{I}_{N_s}$, $d = 1, \ldots, N$, and $\Gamma_{N_s} = \Lambda_{N_s} + 2 \, \mathrm{I}_{N_s}$,
in which

$$\Lambda_{N_s} = \mathrm{diag} \left(1 - \frac{2\Delta t}{(\Delta s)^2} \sin^2 \left(\frac{i\pi}{2\,(N_s + 1)} \right) \right)_{i=1}^{N_s} .$$

Nonnegativity of the Linear Systems

Theorem (Yueh, Lin, and Yau[5])

Both the matrices $\left[I_{N_s^N} - \frac{\Delta t}{2} L_{N_s}^{(N)}\right]^{-1}$ *and* $\left[I_{N_s^N} - \frac{\Delta t}{2} \overline{L}_{N_s}^{(N)}\right]^{-1}$ *are nonnegative operators.*

Remark

The nonnegativity of the solution to the linear systems

$$\left[I_{N_s^N} - \frac{\Delta t}{2} L_{N_s}^{(N)}\right] u = b \quad \text{and} \quad \left[I_{N_s^N} - \frac{\Delta t}{2} \overline{L}_{N_s}^{(N)}\right] u = b$$

are important, since the solution $u(t, s)$ represents the unnormalized probability density function of the state at time t.

[5] M.-H. Yueh, W.-W. Lin and S.-T. Yau, An efficient numerical method for solving high-dimensional nonlinear filtering problems, Commun. Inf. Syst. 14(4), pp. 243-262, 2014.

Numerical Experiments

To simulate the filtering model (1), we generate a set of signals and observations $\{x_k, y_k\}_{k=0}^{N_\tau}$ using the explicit Euler method

$$\begin{cases} x_{k+1} = x_k + f(x_k)\Delta\tau + v\sqrt{\Delta\tau}, \\ y_{k+1} = y_k + h(x_k)\Delta\tau + w\sqrt{\Delta\tau}, \end{cases}$$

where

- $x_0, y_0 = 0$: a given initial state / observation.
- $f : \mathbb{R}^N \to \mathbb{R}^N$: a given drift function.
- $h : \mathbb{R}^N \to \mathbb{R}^M$: a given observation function.
- $\Delta\tau = 0.005$: the time step size.
- v, w: independent Brownian motions with mean 0 and variance 1.

Numerical Experiments

Algorithm 1 Yau-Yau Method for Nonlinear Filtering

Input: Partitions of time $\{t_n\}_{n=1}^{N_t}$ and space $\{s_j\}_{j=1}^{(N_s)^N}$, functions f, h, and observations $\{y_k\}_{k=1}^{N_\tau}$.

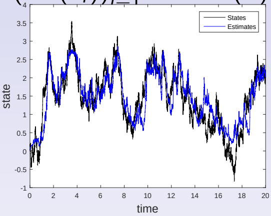
Output: The estimate of the state x_n, $n = 1 \ldots, N_t$.

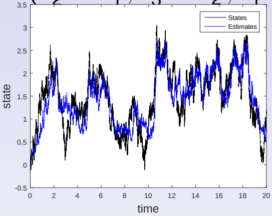
1: **for** $n = 1 \ldots, N_t$ **do**
2: **if** A new measurement $y(\tau_k)$ is observed, i.e. $t_n = \tau_k$ for some k. **then**
3: $u_j^n \leftarrow u_j^n \exp\left\{(y(\tau_k) - y(\tau_{k-1}))^\top h(s_j)\right\}$, $j = 1, \ldots, (N_s)^N$.
4: **end if**
5: $b = \left[I_{(N_s)^N} + (\Delta t)\left(K_{N_s}^{(N)} + Q_{N_s}^{(N)}\right)\right] u^{n-1}$.
6: $u^n = W_{N_s}^{(N)}\left(\Lambda_{N_s}^{-1}\left(\left(W_{N_s}^{(N)}\right)^* b\right)\right)$.
7: Normalize $u^n \leftarrow \frac{u^n}{\sum_j u_j^n}$.
8: The estimate of the state $x_n \approx \sum_{j=1}^{(N_s)^N} u_j^n s_j$.
9: **end for**

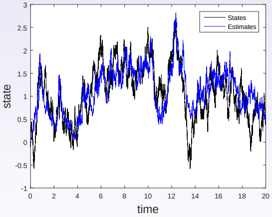
Numerical Result of a 3-D Filtering Problem

The following is a numerical result of the filtering model (1) in which

$$f(x) = (\cos(x_i))_{i=1}^3 \text{ and } h(x) = (x_2^3 - x_1^2, x_3^3 - x_2^2, x_1^3 - x_3^2).$$



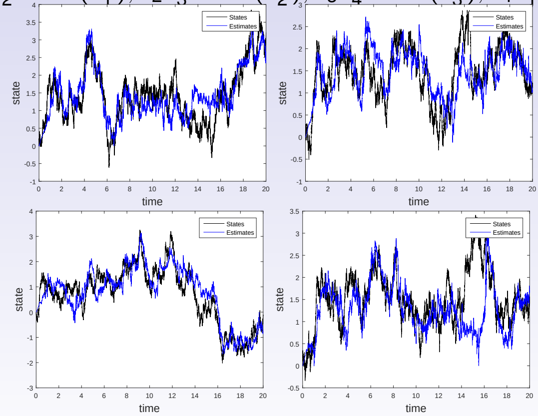




Black: Real State, Blue: Estimate

Numerical Result of a 4-D Filtering Problem

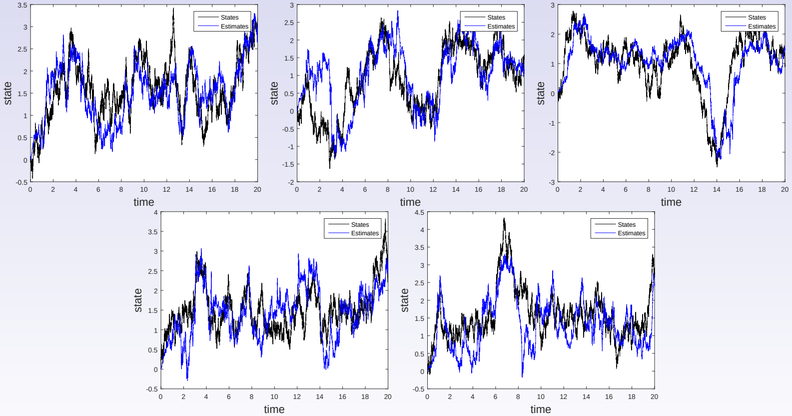
$$f(x) = (\cos(x_i))_{i=1}^{4},$$
$$h(x) = (x_1 x_2^2 - \sin(x_1^2), x_2 x_3^2 - \sin(x_2^2), x_3 x_4^2 - \sin(x_3^2), x_4 x_1^2 - \sin(x_4^2)).$$



Black: Real State, Blue: Estimate

Numerical Result of a 5-D Filtering Problem

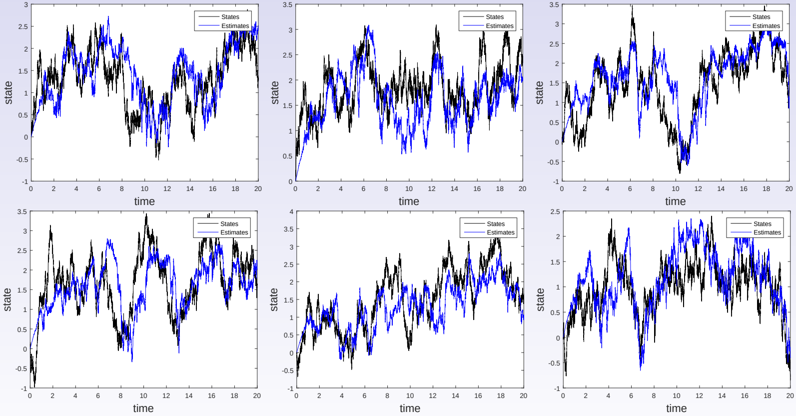
$$f(x) = (\cos(x_i))_{i=1}^{5},$$
$$h(x) = (x_5 x_2^2 - x_1, x_1 x_3^2 - x_2, x_2 x_4^2 - x_3, x_3 x_5^2 - x_4, x_4 x_1^2 - x_5).$$



Black: Real State, Blue: Estimate

Numerical Result of a 6-D Cubic Filtering Problem

$$f(x) = (\cos(x_i))_{i=1}^{6},$$
$$h(x) = (x_6x_1x_2, x_1x_2x_3, x_2x_3x_4, x_3x_4x_5, x_4x_5x_6, x_5x_6x_1).$$



Black: Real State, Blue: Estimate

Summary

We have briefly introduced the history of the Yau-Yau method for the N-dimensional nonlinear filtering problem and the numerical algorithm using the QIEM. The advantages of the QIEM are

- The linear system of the QIEM is probability density preserving, i.e. the solution of the linear system is always nonnegative.
- The solver for QIEM is efficient with complexity $O((N_s)^N \log N_s)$.
- The convergence of the QIEM is guaranteed.
- Numerical results show the accuracy of the algorithm.

Summary

We have briefly introduced the history of the Yau-Yau method for the N-dimensional nonlinear filtering problem and the numerical algorithm using the QIEM. The advantages of the QIEM are

- The linear system of the QIEM is probability density preserving, i.e. the solution of the linear system is always nonnegative.
- The solver for QIEM is efficient with complexity $O((N_s)^N \log N_s)$.
- The convergence of the QIEM is guaranteed.
- Numerical results show the accuracy of the algorithm.

Thank you for your attention.