Data Analytics and Topology Volume 1 (2025) 77–84

Yau-YauAL: A computer tool for solving nonlinear filtering problems

Yu Wang, Shuyuan Xu, Xueda Wei, Xinrui Luo, Stephen Shing-Toung Yau, Shing-Tung Yau, and Rongling Wu*

The Yau-Yau nonlinear filter has increasingly emerged as a powerful tool to study stochastic complex systems. To leverage it to a wider spectrum of application scenarios, we pack the Yau-Yau filtering ALgorithms (YauYauAL) into a package of computer software. Yau-YauAL was written in R, designed to simplify the implementation of the Yau-Yau filter for solving nonlinear filtering problems. Combining R's accessibility with C++ (via Rcpp) for computational efficiency, YauYauAL provides an intuitive Shiny-based interface that enables real-time parameter adjustment and result visualization. At its core, YauYauAL employs finite difference methods to numerically solve the Kolmogorov forward equation, ensuring a stable and accurate solution even for complex systems. YauYauAL's modular design and open-source framework further encourage customization and community-driven development. YauYauAL aims to bridge the gap between theoretical nonlinear filtering methods and practical applications, without requiring expertise in differential equation solving or programming, fostering its broader impact on various scientific fields, such as signal processing, finance, medicine, and biology among a long list.

Keywords and phrases: Nonlinear filtering, Yau-Yau algorithm, Kolmogorov equations, R package, Shiny.

1. INTRODUCTION

In the domain of modern control theory, filtering is an essential subject that has permeated numerous fields, such as signal processing [5, 29], weather prediction [9, 6], and aerospace engineering [13, 31]. The principal goal of filtering is to attain the ccurate estimation or forecast of a stochastic dynamical system's state, using a set of noisy observations [16, 1]. For real-world applications, it is crucial that these estimations or forecasts are computed iteratively and in real-time.

Nonlinear filtering, in particular, has a broad spectrum of applications in military, engineering, and commercial industries [26, 28, 27]. The nonlinear filtering problem considered here is to determine estimated states for a given observation history of the following signal-observation model [1, 16]:

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

where $\mathbf{x}(t) = (x_1(t), \ldots, x_D(t))^\top \in \mathbb{R}^D$ and $\mathbf{y}(t) = (y_1(t), \ldots, y_M(t))^\top \in \mathbb{R}^M$ are the state and the measurement/observation vectors at time t, respectively, $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_D(\mathbf{x}))^\top$ and $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_M(\mathbf{x}))^\top$ are given vector-valued functions, $\mathbf{v} \in \mathbb{R}^D$ and $\mathbf{w} \in \mathbb{R}^M$ are mutually independent standard Brownian processes. From the main results of Yau and Yau [35, 36, 42], the state vector $\mathbf{x}(t)$ can be estimated from the observation vectors $\{\mathbf{y}(s) \mid s \in [0, t]\}$ by solving the Kolmogorov equations.

Following the introduction of the famous Kalman filter by Kalman and Bucy in the 1960s [18, 19], which has been widely applied across various industries, many researchers have focused on studying nonlinear filtering (NLF) theory and developing practical NLF algorithms. A key challenge in NLF is how to find the best estimate of the state from noisy observations. The optimal estimation of the state can be expressed as a minimum mean square error estimate, which is its conditional expectation based on the observation history [16].

To achieve the best estimate, one method is to directly approximate the conditional expectation, as seen in the popular extended Kalman filter (EKF) [16], unscented Kalman filter (UKF) [17], and ensemble Kalman filter [12]. Both EKF and UKF assume that the posterior distribution of the states is essentially Gaussian or nearly Gaussian, which can limit their use.

Another approach to the NLF problem is to calculate the conditional density function of the state. For instance, the particle filter (PF) [10] uses the empirical distribution of particles to approximate the posterior density. PF is well-known for its versatility in various NLF problems, although it is not suitable for real-time implementation in high-dimensional systems. In the late 1960s, Duncan, Mortensen, and Zakai independently developed the renowned Duncan-Mortensen-Zakai (DMZ) equation for nonlinear filtering [8, 22, 43], which is satisfied by the unnormalized conditional density

arXiv: 2506.08976
*Corresponding author.

function of the states. The DMZ equation, a stochastic differential equation, generally does not have a closed-form solution. For practical use, it is necessary to solve the DMZ equation in real-time and without extensive memory. For finite-dimensional filtering systems, solutions to the DMZ equation can be explicitly built using Lie algebra methods. However, this is only possible for a few types of systems that have finite-dimensional filters [2, 40]. Since the DMZ equation usually lacks a closed-form solution, many mathematicians have sought good approximations. One such method is the splitting up technique, first described by Bensoussan et al. [3, 4] and later studied in [11, 25].

In the 1990s, Lototsky, Mikulevicius, and Rozovskii provided a recursive, time-based Wiener chaos representation for the optimal nonlinear filter [20]. However, these methods generally require that the drift and observation terms, specifically $f(x)$ and $h(x)$ in system (1), are bounded. Another method to solve the DMZ equation is the direct method, which is typically applicable only to Yau filtering systems where the drift term is a linear function plus a gradient function. This method was introduced in [33, 34] and later generalized in [37, 7]. In 2008, Yau and Yau developed the Yau-Yau algorithm, a new method for solving the "pathwise-robust" DMZ equation in time-invariant systems [42]. It has been theoretically proven that the Yau-Yau algorithm can converge to the true solution, as long as the growth rate of the observation $|h|$ is greater than that of the drift $|f|$. Later, Luo and Yau extended the Yau-Yau algorithm to time-varying cases [21].

For readers interested in the mathematical principles and computational techniques enabling the software to address high-dimensional nonlinear filtering problems, we direct them to the work of Yueh et al. [23, 24]. These publications provide a detailed analysis of the underlying mechanisms that power the software's functionality.

In this article, leveraging the powerful capabilities of R and C++, we have crafted an advanced software package. Rooted in the YauYau algorithm, this tool elegantly addresses the complexities of nonlinear filtering problems with remarkable precision and efficiency. This package delivers sophisticated numerical techniques for filtering, while seamlessly integrating an engaging Shiny application designed for dynamic visualization and in-depth exploration. This software package is designed to be concise and user-friendly, enabling users to quickly get started and operate it with ease.

2. THE YAUYAUAL PACKAGE

2.1 Dependency

The `YauYauAL` package (v0.1.0) is built and tested on R version 4.4.2, which can work properly on a standard laptop computer with R version 4.4.2 or higher installed. Also, it is recommended to install RStudio for a better view and interaction with the objects stored in the R environment. The following dependencies are required for data processing: `Deriv` (v4.1.6), `RcppArmadillo` (v1.4.2.3-1), `RcppEigen` (v0.3.4.0.2), and `Matrix` (v1.7-1). In addition, `ggplot2` (v3.5.1), `reshape2` (v1.4.4), `gridExtra` (v2.3), `shiny` (v1.10.0), and `shinythemes` (v1.2.0) are required for visualization.

2.2 Software Installation Guide

Begin by installing the prerequisite R package `devtools`. Open your R console and execute the following command:

```
>install.packages("devtools")
```

After successful installation, proceed to install `YauYauAL` directly from the BIMSA-Stat GitHub repository using:

```
>devtools::install_github("BIMSA-Stat/
    YauYauAL")
```

For users preferring manual installation, first download the YauYauAL_0.1.0.tar.gz file from the GitHub repository, and then utilize R's built-in package manager to install the local archive.

Before the `YauYauAL` can be used in R, it is necessary to import the package using the following command:

```
>library(YauYauAL)
```

2.3 Demonstration

Users can quickly access the interactive graphical user interface of the YauYauAL computational tool using the following R command.

```
>YauYauAL::run_app()
```

3. STEP-BY-STEP METHOD DETAILS

3.1 Initialize Parameters

First, define the dimension of the nonlinear filtering problem and the drift function **f** for the state equation, as well as the observation function **h** in the observation equation:

```
>Dim <- 3
>f <- function(x) {return(c(
  cos(x[1]),
  cos(x[2]),
  cos(x[3]))
)}
>h <- function(x) {return(c(
  x[1]^3,
  x[2]^3,
  x[3]^3)
)}
```

Here, `Dim` specifies the dimension of the state vector, indicating that the system consists of three state variables. The drift function **f** applies the cosine function element-wise to the state vector **x**, capturing the nonlinear dynamics of the system. Meanwhile, the observation function **h** maps each state variable to its cubic power, establishing a highly nonlinear relationship between the states and the observations. This particular configuration is known as the **Cubic Sensor Problem**, a canonical example in nonlinear filtering that highlights the challenges of state estimation when both the system dynamics and measurement models are nonlinear. The cubic sensor problem is widely recognized for its ability to test the robustness and accuracy of nonlinear filtering algorithms, as traditional linear filtering methods, such as the Kalman Filter, are inadequate due to their reliance on linear assumptions.

To set up the parameters for our simulation, we proceed as follows:

```
>T <- 5
>Dt <- 0.001
>Dtau <- 5 * Dt                    # 0.005
>Nt <- as.integer(Dtau / Dt)       # 5
>Ntau <- as.integer(T / Dtau)      # 4000
>NtNtau <- as.integer(T / Dt)      # 20000
```

In this setup, the terminal time T is set to 20, which defines the total duration of the simulation. The time step $\Delta\tau$ is set to 0.005 for generating data of signals and observations, ensuring that the data is captured at a fine enough resolution to reflect the system's dynamics accurately. The smaller time step Δt is designated as 0.001 for solving the Kolmogorov equation, which is crucial for the accuracy of the state estimation in the nonlinear filtering process. The number of time steps N_t and N_τ are calculated based on T, Δt, and $\Delta\tau$ respectively, determining how often the system state is updated and observations are recorded throughout the simulation. These parameters are essential for configuring the simulation environment in the `YauYauAL` R package, allowing us to effectively address the nonlinear filtering problem. Figure 1 illustrates this process.

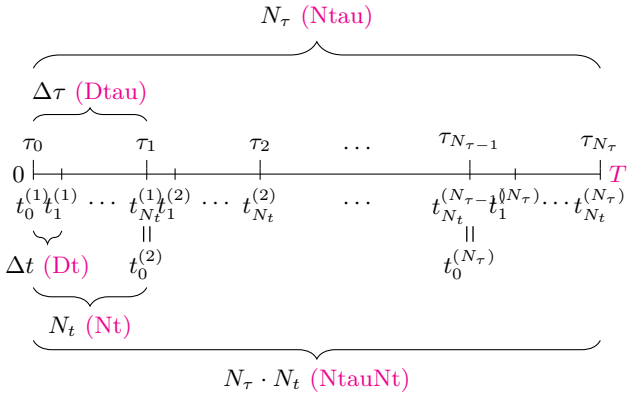

Figure 1. Schematic of Discretized Time Intervals

3.2 State and Observation Generator

The simulation of state and observation sequences is conducted using the following R script:

```
>seed_value <- 42
>result <- Simulate_State_Obser(
  Dt = 0.001,
  Ntau = 4000,
  NtNtau = 20000,
  f = f,
  h = h,
  Dim = 3,
  seed = 42)
>x <- result$x
>y <- result$y
```

In this simulation, a seed value of 42 is set for the random number generator to ensure the reproducibility of results. The function `Simulate_State_Obser` is employed with parameters $\Delta t = 0.001$, $N_\tau = 4000$, and $N_\tau \cdot N_t = 20000$ to define the temporal resolution and duration of the simulation. The nonlinear dynamics are modeled through the drift function f and the observation function h, with the state dimension set to 3. The sequences **x** and **y**, representing the state and observations, are extracted from the result object for subsequent analysis with nonlinear filtering algorithms.

3.3 Discretization

The discretization process involves creating a grid of points in the state space to facilitate the numerical solution of the Kolmogorov equation:

```
>Ds <- 0.5
>s <- seq(min(x), max(x)+Ds, by = Ds)
>Ns <- length(s)
>s <- ExpandGrid(Dim,s)
```

In the above code, we first define the grid spacing Δs. Then, we generate a sequence of grid points s that spans

from the minimum value of the state sequence x to slightly beyond its maximum value, incremented by Δs. The number of grid points N_s is determined by the length of this sequence. Finally, the `ExpandGrid` function is used to create a multi-dimensional grid covering the entire state space based on the state vector dimension `Dim`. This grid will serve as the basis for discretizing the state space in subsequent filtering computations.

Figure 2 illustrates the process of spatial discretization.

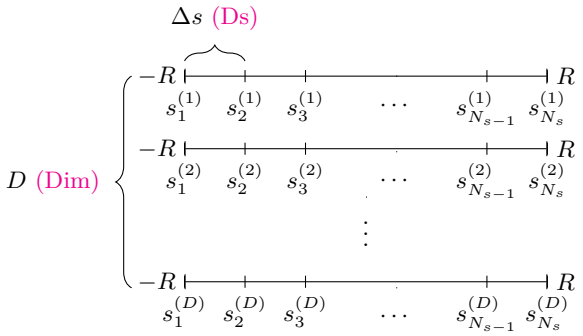

Figure 2. Schematic Representation of Spatial Discretization Across Dimensions

Next, we generate the necessary matrices and parameters for the discretization process:

```
>D <- generateD(Dim,Ns,Ds)
>Lambda <- computeLambda(Dim,Ns,Dt,Ds)
>df <- generate_derivative(f)
>B <- computeB(s,D,Dt,Ds,f,df,h)
```

In the above code, the function `generateD` creates the matrix D which represents the discrete derivative operator based on the state dimension `Dim`, the number of grid points N_s, and the grid spacing Δs. The function `computeLambda` calculates the matrix Λ using the state dimension `Dim`, the number of grid points N_s, the time step Δt, and the grid spacing Δs. The function `generate_derivative` generates the derivative function df based on the drift function f. Finally, the function `computeB` computes the matrix B using the grid points `s`, the matrix D, the time step Δt, the grid spacing Δs, the drift function f, its derivative df, and the observation function h. These matrices and parameters are essential for the subsequent steps in the nonlinear filtering process, particularly for solving the Kolmogorov equation numerically.

3.4 State Estimation

The state estimation process involves computing the posterior distribution of the state variables given the observations. This is achieved using the following R script:

```
>Iu <- wrap_outiu_function(s, NtNtau,
    Ntau, Nt, Dim, y, h, Lambda, B, Ns,
     NormalizedExp, DST_Solver)
```

The function computes the posterior distribution of the state variables at each time step using the grid points, time steps, state dimension, observations, observation function, noise parameters, grid size, normalization function, and solver function. The Discrete Sine Transform to efficiently solve the Kolmogorov equation for accurate state estimation in nonlinear systems.

3.5 Interactive Implementation of the Algorithm with Shiny

You can launch the homepage of the interactive software interface using the following R command:

```
>YauYauAL::run_app()
```

Figure 3 shows the interface of our interactive software. When defining custom filters, the algorithm may encounter significant delays in execution. In such scenarios, Shiny might not update the results in real-time. We recommend that users run the scripts outlined in subsections 3.1–3.4 independently to obtain the results, followed by visualization using the plot function.

4. NUMERICAL EXPERIMENTS

4.1 almost linear sensor problem

Example 4.1. The almost linear sensor problem is as follows:

$$\begin{cases} \mathrm{d}x(t) = dv(t) \\ \mathrm{d}y(t) = x(t)\left(1 + 0.25\cos x(t)\right)\mathrm{d}t + \mathrm{d}w(t) \end{cases} \tag{2}$$

where $x(t)$, $y(t) \in \mathbb{R}$, $v(t)$, $w(t)$ are independent standard scalar Brownian motion processes. The parameters Dim $= 3$, $T = 50$, $\Delta t = 0.0001$, $\Delta\tau = 0.0005$, $\Delta s = 0.5$.

Figure 4 illustrates the results of the Yau-Yau Filter when dealing with the almost linear sensor problem.

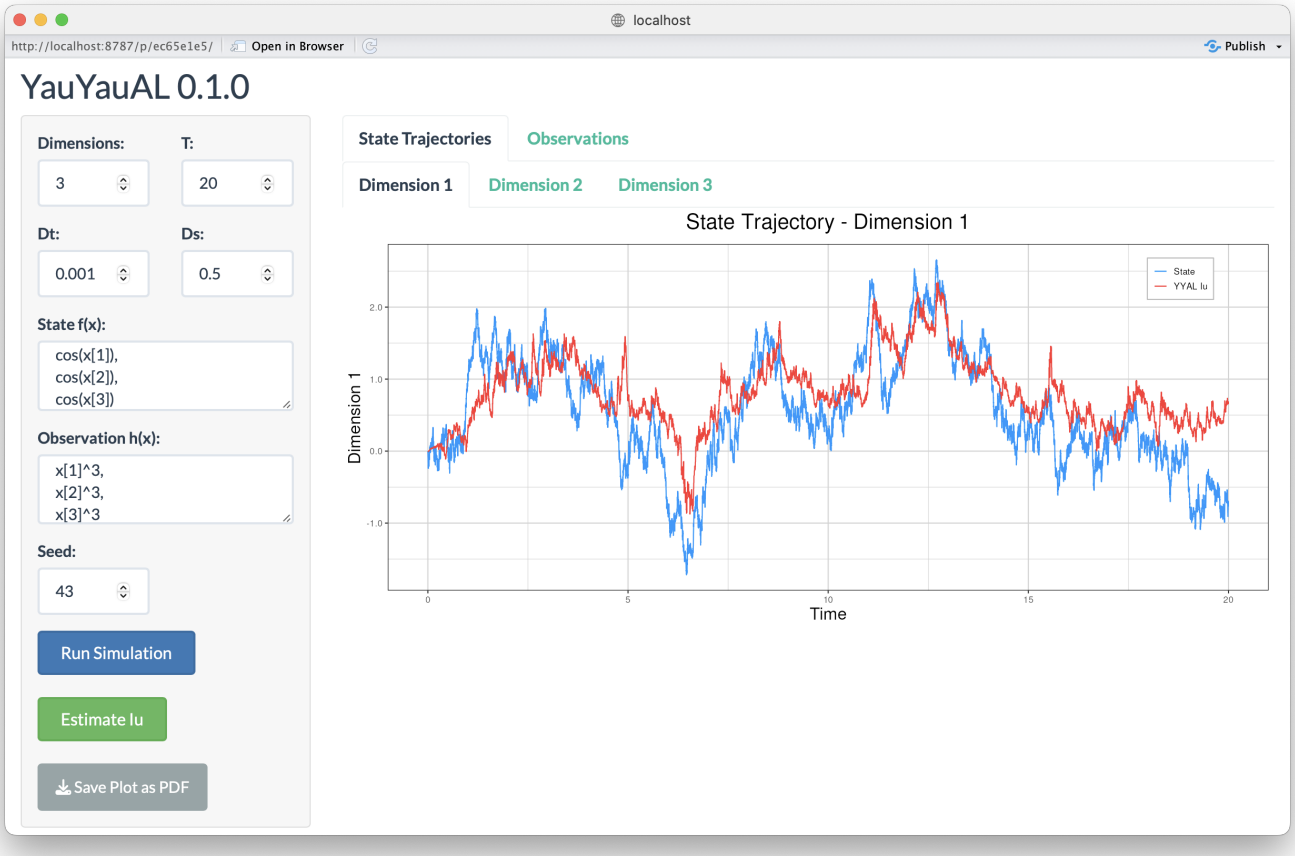

Figure 3. The interface of the interactive YauYauAL software based on Shiny.

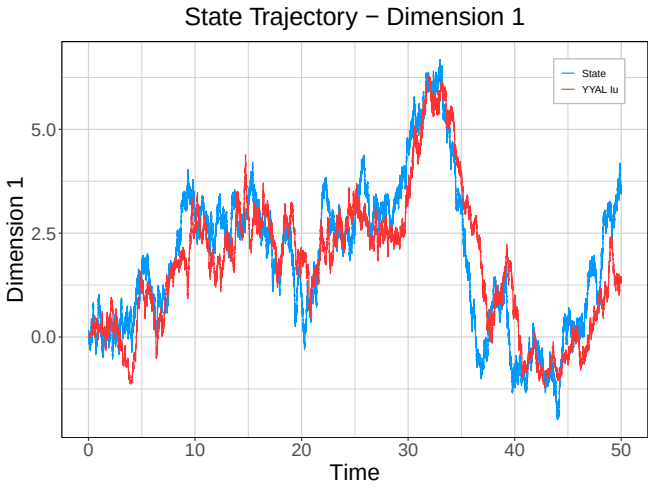

Figure 4. Estimations of the almost linear sensor for the model Eq.2, and $T = 50$, with the time step $\Delta t = 0.0001$. Blue: real state; Red: Yau-Yau filter.

4.2 cubic sensor problem

Example 4.2. The cubic sensor problem [41] is as follows:

$$\begin{cases} \mathrm{d}x(t) = \cos x(t)\mathrm{d}t + \mathrm{d}v(t) \\ \mathrm{d}y(t) = x^3\mathrm{d}t + \mathrm{d}w(t) \end{cases} \tag{3}$$

where $x(t)$, $y(t) \in \mathbb{R}$, $v(t)$, $w(t)$ are independent standard scalar Brownian motion processes. The parameters Dim $= 3$,$T = 20$, $\Delta t = 0.001$, $\Delta\tau = 0.005$, $\Delta s = 0.5$.

Figure 5 presents the outcomes of the Yau-Yau Filter applied to the 3D cubic sensor problem.

5. DISCUSS

The development of the Yau-YauAL software package represents a significant step forward in the practical application of the Yau-Yau nonlinear filter. By integrating the powerful capabilities of R with the computational efficiency of C++ through Rcpp, Yau-YauAL effectively addresses the challenges associated with implementing complex filtering

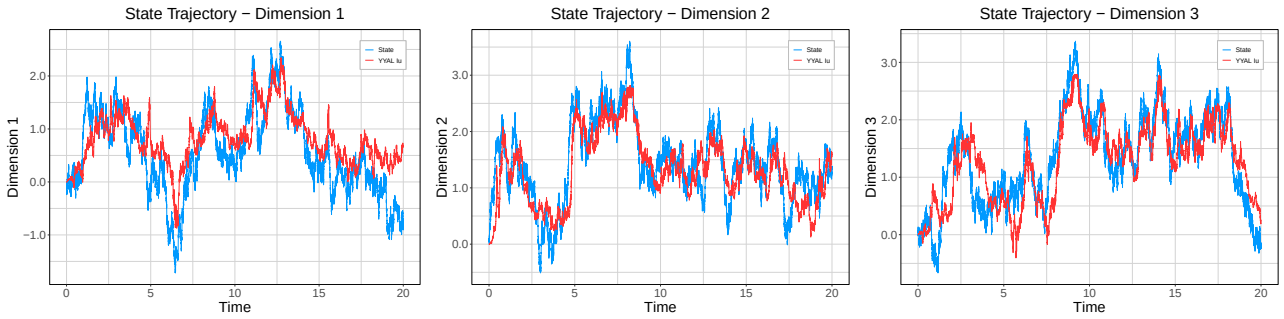

Figure 5. Estimations of the 3-D Cubic sensor for the model Eq.3, and $T = 20$, with the time step $\Delta t = 0.001$. Blue: real state; Red: Yau-Yau filter.

algorithms. The use of finite difference methods to solve the Kolmogorov forward equation ensures that the tool can handle a wide range of complex systems while maintaining stability and accuracy in its solutions.

The inclusion of an intuitive Shiny-based interface is particularly noteworthy. This feature democratizes access to advanced filtering techniques by allowing users to adjust parameters and visualize results in real-time, without the need for extensive programming knowledge. This is especially beneficial for researchers in fields such as finance, signal processing, and biology, where the application of nonlinear filtering can yield valuable insights but may be hindered by the complexity of the underlying mathematics.

The modular design and open-source nature of Yau-YauAL further enhance its utility. By allowing for customization and community-driven development, the software can be adapted to meet the specific needs of various users and evolve with advancements in the field. This collaborative approach not only fosters innovation but also ensures that the tool remains relevant and effective in a rapidly changing scientific landscape.

However, it is important to acknowledge potential limitations. While Yau-YauAL has been designed to be user-friendly, the complexity of some nonlinear filtering problems may still require a certain level of expertise to fully exploit the tool's capabilities. Additionally, the performance of the finite difference methods employed may be influenced by the specific characteristics of the system being analyzed, and further research may be needed to optimize these methods for different applications.

Future work could focus on expanding the range of numerical methods available within Yau-YauAL to provide users with even more options for solving nonlinear filtering problems. Additionally, incorporating machine learning techniques could enhance the tool's ability to handle large datasets and improve the accuracy of its solutions. Collaboration with researchers across disciplines will be crucial in identifying new applications and refining the software to meet emerging challenges.

In conclusion, Yau-YauAL has the potential to significantly impact the field of nonlinear filtering by making advanced techniques more accessible and user-friendly. Its innovative design and open-source framework position it as a valuable resource for both current and future interdisciplinary research endeavors.

CODE AND DATA AVAILABILITY

The code and data related to this study are publicly available on GitHub. The repository can be accessed via the following link: https://github.com/BIMSA-Stat/YauYauAL. This repository contains the complete implementation of the YauYauAL software package, including the source code, documentation, and example datasets used in this research. Users are encouraged to explore the repository for further details and to utilize the provided resources for their own research and development purposes.

ACKNOWLEDGEMENTS

We thank the colleagues at the Beijing Key Laboratory of Topological Statistics and Applications for Complex Systems at the Beijing Institute of Mathematical Sciences and Applications for their contributions to this work. The authors thank the anonymous referees for their careful reading and valuable suggestions. This work was supported in part by the Natural Science Foundation of Inner Mongolia (No. 2024MS01017) and the First Class Discipline Project, Inner Mongolia Autonomous Region, China (No. YLXKZX-NSD-007)

Received 10 May 2025

REFERENCES

[1] A. Bain, and D. Crisan. Fundamentals of Stochastic Filtering. Springer Science Business Media, 1st ed. (2009).

[2] V. E. Benes. Exact finite-dimensional filters for certain diffusions with nonlinear drift. *Stochastics* **5** (1981), 65–92.

[3] A. Bensoussan, R. Glowinski, and A. Rascanu. Approximation of the Zakai equation by the splitting up method. *SIAM J. Control Optim.* **28**(1990), 1420–1431.

[4] A. Bensoussan, R. Glowinski, and A. Rascanu. Approximation of some stochastic differential equations by the splitting up method. *Appl. Math. Optim.* **25** (1992), 81–106.
[5] J. V. Candy. Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods, 1st ed. John Wiley & Sons, Hoboken, NJ. (2016).
[6] K. Chen, and J. Yu. Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach. *Applied Energy* **113** (2014), 690–705.
[7] X. Chen, J. Shi, and S. S.-T. Yau. Real-time solution of time-varying Yau filtering problems via direct method and Gaussian approximation. *IEEE Trans. Autom. Control* **64** (2019), 1648–1654.
[8] T. E. Duncan. Probability densities for diffusion processes with applications to nonlinear filtering theory and detection theory. Stanford California, Stanford, CA, USA, Tech. Rep. TR-7001-4. (1967).
[9] G. Galanis, P. Louka, P. Katsafados, I. Pytharoulis, and G. Kallos. Applications of Kalman filters based on non-linear functions to numerical weather predictions. *Annales Geophysicae* **24** (2006), 2451–2460.
[10] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proc. F, Radar Signal Process.* **140** (1993), 107–113.
[11] I. Gyongy, and N. Krylov. On the splitting-up method and stochastic partial differential equations. *Ann. Probab.* **31** (2003), 564–591.
[12] P. L. Houtekamer, and H. L. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Rev.* **126** (1998), 796–811.
[13] C. Ichard. Random Media and Processes Estimation Using Nonlinear Filtering Techniques: Application to Ensemble Weather Forecast and Aircraft Trajectories. PhD thesis, Université de Toulouse, Université Toulouse III–Paul Sabatier. (2015).
[14] K. Ito. Approximation of the Zakai equation for nonlinear filtering. *SIAM J. Control Optim.* **34** (1996), 620–634.
[15] K. Ito, and B. Rozovskii. Approximation of the Kushner equation for nonlinear filtering. *SIAM J. Control Optim.* **38** (2000), 893–915.
[16] A. H. Jazwinski. Stochastic Processes and Filtering Theory. Courier Corporation. (2007).
[17] S. J. Julier, and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proc. IEEE* **92** (2004), 401–422.
[18] R. E. Kalman. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **82** (1960), 35–45.
[19] R. E. Kalman, and R. S. Bucy. New results in linear filtering and prediction theory. *J. Basic Eng.* **83** (1961), 95–108.
[20] S. Lototsky, R. Mikulevicius, and B. L. Rozovskii. Nonlinear filtering revisited: A spectral approach. *SIAM J. Control Optim.* **35** (1997), 435–461.
[21] X. Luo, and S. S.-T. Yau. Hermite spectral method to 1-D forward Kolmogorov equation and its application to nonlinear filtering problems. *IEEE Trans. Autom. Control* **58** (2013), 2495–2507.
[22] R. E. Mortensen. Optimal control of continuous time stochastic systems. Ph.D. dissertation, Electron. Res. Lab., California Univ. Berkeley, Berkeley, CA, USA. (1966).
[23] M.-H. Yueh, W.-W. Lin, and S.-T. Yau. An efficient numerical method for solving high-dimensional nonlinear filtering problems. *Communications in Information and Systems* **14** (2014), 243–262.
[24] M.-H. Yueh, W.-W. Lin, and S.-T. Yau. An Efficient Algorithm of Yau-Yau Method for Solving Nonlinear Filtering Problems. *Communications in Information and Systems* **14** (2014), 111–134.
[25] N. Nagase). Remarks on nonlinear stochastic partial differential equations: An application of the splitting-up method. *SIAM J. Control Optim.* **33**(1995), 1716–1730.
[26] G. G. Rigatos. Modelling and Control for Intelligent Industrial Systems: Adaptive Algorithms in Robotics and Industrial Engineering. Springer-Verlag Berlin Heidelberg. (2011).
[27] G. G. Rigatos, and P. Siano. Sensorless control of electric motors with Kalman filters: applications to robotic and industrial systems. *International Journal of Advanced Robotic Systems* **8** (2011), 62–80.
[28] G. G. Rigatos. Nonlinear Estimation and Applications to Industrial Systems Control. Nova Science Publishers Inc. (2013).
[29] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson. The ensemble Kalman filter: a signal processing perspective. *EURASIP Journal on Advances in Signal Processing* **2017** (2017), 1–16.
[30] J. Shi, Z. Yang, and S. S. T. Yau. Direct method for Yau filtering system with nonlinear observations. *Int. J. Control* **91** (2018), 678–687.
[31] J. Sun, H. A. Blom, J. Ellerbroek, andJ. M. Hoekstra. Particle filter for aircraft mass estimation and uncertainty modeling. *Transportation Research Part C: Emerging Technologies* **105** (2019), 145–162.
[32] W. S. Wong. New classes of finite-dimensional nonlinear filters. *Syst. Control Lett.* **3** (1983),155–164.
[33] S. S.-T. Yau and S. T. Yau. New direct method for Kalman-Bucy filtering system with arbitrary initial condition. In *Proc. 33rd IEEE Conf. Decis. Control*, vol. 2, pp. 1221–1225.
[34] S. S.-T. Yau. Finite-dimensional filters with nonlinear drift. I: A class of filters including both Kalman-Bucy and Bene's filters. *J. Math. Syst. Estimation Control* **4** (1994), 181–203.
[35] S. T. Yau, and S. S.-T. Yau. Finite dimensional filters with nonlinear drift iii: Duncan-Mortensen-Zakai equation with arbitrary initial condition for kalman-bucy filtering system and benes filtering system. *IEEE Transactions on Aerospace and Electronic Systems* **33** (1997), 1277–1294.
[36] S. T. Yau, and S. S.-T. Yau. Real time solution of nonlinear filtering problem without memory I. *Mathematical Research Letters* **7** (2000), 671–693.
[37] S. S.-T. Yau, and Y.-T. Lai. Explicit solution of DMZ equation in nonlinear filtering via solution of ODEs. *IEEE Trans. Autom. Control* **48** (2003), 505–508.
[38] S. T. Yau, and S. S.-T. Yau. Nonlinear filtering and time varying Schrödinger equation. *IEEE Trans. Aerosp. Electron. Syst.* **40** (2024), 284–292.
[39] S. S.-T. Yau, C. Yan, and S.-T. Yau. Linear filtering with nonlinear observations. In *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, vol. 2, pp. 2112–2117.
[40] S. S.-T. Yau and G.-Q. Hu. Classification of finite-dimensional estimation algebras of maximal rank with arbitrary state-space dimension and Mitter conjecture. *Int. J. Control* **78** (2005), 689–705.
[41] C. Yan, and S. S.-T. Yau. A new suboptimal filter and numerical solutions for the cubic sensor problem. In *2006 IEEE International Conference on Networking, Sensing and Control*, pages 351–356. (2006).
[42] S.-T. Yau, and S. S.-T Yau. Real time solution of nonlinear filtering problem without memory II. *SIAM J. Control Optim.* **47** (2008), 163–195.
[43] M. Zakai. On the optimal filtering of diffusion processes. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* **11**(1969), 230–243.

Yu Wang
Beijing Key Laboratory of Topological Statistics and Applications for Complex Systems, Beijing Institute of Mathematical Sciences and Applications
Beijing 101408, P.R. China
Institute of Statistics and Big Data, Renmin University of China
Beijing 100872, P.R. China
E-mail address: yuwang@bimsa.cn

Shuyuan Xu
Beijing Key Laboratory of Topological Statistics and Applications for Complex Systems, Beijing Institute of Mathematical Sciences and Applications
Beijing 101408, P.R. China
Academy of Mathematics and Systems Science, Chinese Academy of Sciences
Beijing 100190, P.R. China
University of Chinese Academy of Sciences
Beijing 100049, P.R. China
E-mail address: xushuyuan@bimsa.cn

Xueda Wei
Digital Economy Laboratory, Beijing Institute of Mathematical Sciences and Applications
Beijing 101408, P.R. China
Beijing Key Laboratory of Topological Statistics and Applications for Complex Systems, Beijing Institute of Mathematical Sciences and Applications
Beijing 101408, P.R. China
Institute of Statistics and Big Data, Renmin University of China
Beijing 100872, P.R. China
E-mail address: weixueda@bimsa.cn

Xinrui Luo
College of Mathematics Science, Inner Mongolia Normal University
Hohhot 010022, P.R. China
Beijing Key Laboratory of Topological Statistics and Applications for Complex Systems, Beijing Institute of Mathematical Sciences and Applications
Beijing 101408, P.R. China
E-mail address: luoxinrui@bimsa.cn

Stephen Shing-Toung Yau
Beijing Key Laboratory of Topological Statistics and Applications for Complex Systems, Beijing Institute of Mathematical Sciences and Applications
Beijing 101408, P.R. China
Department of Mathematical Sciences, Tsinghua University
Beijing 100084, P.R. China
E-mail address: yau@uic.edu

Shing-Tung Yau
Beijing Key Laboratory of Topological Statistics and Applications for Complex Systems, Beijing Institute of Mathematical Sciences and Applications
Beijing 101408, P.R. China
Yau Mathematical Sciences Center, Tsinghua University
Beijing 100084, P.R. China
E-mail address: styau@tsinghua.edu.cn

Rongling Wu
Beijing Key Laboratory of Topological Statistics and Applications for Complex Systems, Beijing Institute of Mathematical Sciences and Applications
Beijing 101408, P.R. China
Yau Mathematical Sciences Center, Tsinghua University
Beijing 100084, P.R. China
E-mail address: ronglingwu@bimsa.cn