國立臺灣大學理學院數學研究所

碩士論文

Department of Mathematics

College of Science

National Taiwan University

Master's Thesis

卡爾曼濾波器方法及其應用

Kalman filter method and its applications

蕭羽晨

Yu-Chen Xiao

指導教授: 王振男 教授

Advisor: Jenn-Nan Wang , Professor

中華民國 113 年 5 月

May, 2024

doi:10.6342/NTU202400950

doi:10.6342/NTU202400950

Acknowledgements

在碩士的這兩年中，我要感謝嘉麗，我家的貓，雖然沒什麼幫助。

我也要感謝 548 大大大們，我們這間研究室應該是整棟天數最歡樂吵雜的存在，剛好聚集了各式各樣的專長人物。不是在沙發，就是在球場的計算大師林伯駿；永遠不辦 LinePay 的碼農陳俊憲；以及具備這兩交集的萬事通鄭書承。

田徑隊的各位是研究生活中的最大紓壓，雖然你們看不到。

我在台北的阿姨姨丈，也謝謝你們，使我在任何時候都可以回去度假。

當然還有我的老闆，大大的支持我這兩年的田徑運動生涯，金牌就歸給教授了!

最後，感謝高雄的三一豐收教會，你們就是我最大的動力來源。

願榮耀歸與 神!

doi:10.6342/NTU202400950

doi:10.6342/NTU202400950

摘要

在動態系統中，參數估計是一直在研究的問題。假設我們知曉一些資訊，可以將參數估計得更準確嗎? 卡爾曼濾波器是一種解答。其背後的想法涉及了偏微分，統計方法，還有貝式法則下的後驗均值等。

本文中針對各種情況介紹了五種濾波器: 若模型是線性，卡爾曼濾波器是最優解；對低階非線性模型使用的擴展卡爾曼濾波器和無跡卡爾曼濾波器；針對高度非線性模型的粒子濾波器；解決極高維的集群卡爾曼濾波器。這五種濾波器的詳細推導都寫在各章中。

最後，在第七章，我們透過對牛頓系統，高度非線性系統，以及電阻抗斷層掃描反問題的數值模擬展現這些濾波器的成效。

關鍵字：卡爾曼濾波器、擴展卡爾曼濾波器、無跡卡爾曼濾波器、粒子濾波器、集群卡爾曼濾波器、模型預測、貝式估計、參數估計

doi:10.6342/NTU202400950

doi:10.6342/NTU202400950

Abstract

In dynamic systems, parameter estimation is a persistent research challenge. Can we achieve more accurate parameter estimation if we have some prior information? The Kalman filter provides an answer to this question. Its underlying principles involve partial differential, statistical techniques, and Bayesian inference, including posterior mean estimation.

This paper introduces five types of filters for various scenarios: the Kalman filter is optimal for linear models; the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are suitable for models with low-order nonlinearity; the Particle Filter (PF) is applicable to highly nonlinear models; and the Ensemble Kalman Filter (EnKF) is effective for addressing extremely high-dimensional systems. The derivation of these five filters is presented in detail in respective chapters.

In the final chapter, Chapter 7, the effectiveness of these filters is demonstrated through numerical simulations of Newtonian systems, highly nonlinear systems, and Electrical Impedance Tomography (EIT) inverse problem.

Keywords: Kalman filter, Extended Kalman Filter, Unscented Kalman Filter, Particle Filter, Ensemble Kalman Filter, model prediction, Bayesian estimation, parameter estimation

doi:10.6342/NTU202400950

doi:10.6342/NTU202400950

Contents

doi:10.6342/NTU202400950

doi:10.6342/NTU202400950

Chapter 1 Introduction

The Kalman filter, originally proposed by Rudolf Kalman, is a mathematical technique utilized for state estimation and prediction in dynamic systems. It offers the optimal estimation of system states through the fusion of sensor measurements and dynamic models. Widely applied in various fields such as navigation, control, signal processing, and machine learning

The Kalman filter operates based on Bayesian filtering principles, estimating system states by integrating prior and measurement information through a weighted average. Its recursive, optimal, and linear nature renders it highly efficient for practical applications. The fundamental concept of the Kalman filter involves modeling the system state as a dynamic process, evolving according to a state equation influenced by Gaussian-distributed random noise.

The Kalman filter comprises two primary stages: prediction and update. During the prediction phase, the estimation of the state for the subsequent time step is forecasted based on the system's state equation and the prior state estimation. In the update phase, the optimal estimation of the system state is computed by comparing the measured value with the prior state estimate. These two steps are iterated recursively to continuously estimate the system state.

The Kalman filter is known for its efficiency, accuracy, and robustness, providing excellent state estimation and prediction performance even in the presence of large noise disturbances. However, the Kalman filter also has limitations, such as the assumption of linear models and Gaussian noise. Hence, numerous enhanced and extended techniques rooted in the Kalman filter framework have been devised in practical scenarios. These include the extended Kalman filter, unscented Kalman filter, and particle filter, tailored to address intricate nonlinear system dynamics and measurement models.

The structure of this paper is as follows: Chapter 2 provides a derivation of the Kalman

doi:10.6342/NTU202400950

filter under the assumption of linear models; Chapters 3 and 4 introduce the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF), which are capable of approximating low-order nonlinear models; Chapter 5 discusses filters that excel under highly nonlinear conditions, beginning with the Particle Filter (PF) and progressing to importance sampling and sequential importance resampling. Finally, the Rao-Blackwellized Particle Filter is mentioned. The essence of Chapter 5 revolves around Bayesian principles and Monte Carlo methods. Chapter 6 introduces the Ensemble Kalman Filter (EnKF), suitable for high-dimensional systems. Chapter 7 consists of numerical simulations of Newtonian systems and highly nonlinear systems.

Of course, there is no free lunch. The Kalman filter is fast but can only be used in linear models; the Extended Kalman Filter may make errors in locally highly nonlinear models and requires the computation of Jacobian matrices; the Unscented Kalman Filter may fail during numerical diagonalization and is influenced by the dimensionality; the Particle Filter can handle various situations, but implementing Monte Carlo methods requires significant computational costs; and the Ensemble Kalman Filter's choice of the prior can have a decisive impact on subsequent predictions.Therefore, considering the cost as well as the characteristics of the model, readers must carefully choose which filter to use.

doi:10.6342/NTU202400950

Chapter 2 Kalman Filter

2.1 Derivation of Kalman filter

Consider a linear dynamic equation with linear measurement equation:

$$x_k = A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1},$$
$$y_k = C_k x_k + v_k,$$

where A, B, C are known matrix, x_k is state, y_k is measurement, u is input, and w_k, v_k are noise, which we assume they are uncorrelated and

$$w_k \sim N(0, Q_k),$$
$$v_k \sim N(0, R_k).$$

Our purpose is to estimate better x under the interference of noise, the most straightforward way is to combine all observations y and calculate the expected value, i.e.

$$x_k^+ = E[x_k | y_1, ..., y_k],$$

where **"+"** mean that the estimate is a posteriori.

Similar, if we do not yet have information for time k, y_k, then we have estimate:

$$x_k^- = E[x_k | y_1, ..., y_{k-1}],$$

where **"-"** mean that the estimate is a priori.

But as our number of observations grows, this is not a practical way to do it, we need resort to a recursive way.

Next question is, if we have x_{k-1}^+ and P_{k-1}^+, how do we get x_k^- and P_k^- in recursive? And most importantly, how to update x_k^+ and P_k^+ when y_k is added? Kalman filter is one of

doi:10.6342/NTU202400950

the ways , a method with the optimization matrix K_k, a.k.a. Kalman Gain. The detailed derivation is as follows.

(1): x_k^-

Since no observations are added, one of the most straightforward way is to take the expected value. By our linear system, take expectations we have:

$$\begin{aligned} x_k^- &= E[x_k] = E[A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1}] \\ &= A_{k-1}E[x_{k-1}] + B_{k-1}u_{k-1}, \end{aligned}$$

and the best estimate at time $k-1$ is x_{k-1}^+, therefore the relation of x_k^- and x_{k-1}^+ is

$$x_k^- = A_{k-1}x_{k-1}^+ + B_{k-1}u_{k-1}.$$

(2): P_k^-

Naturally, $P_k^- = E[(x_k - x_k^-)(x_k - x_k^-)^t]$, and simplify it:

$$\begin{aligned} P_k^- &= E[(x_k - x_k^-)(x_k - x_k^-)^t] \\ &= E[(A_{k-1}x_{k-1} + B_{k-1}u_{k-1} + w_{k-1} - (A_{k-1}E[x_{k-1}] + B_{k-1}u_{k-1}))(same)^t] \\ &= E[(A_{k-1}(x_{k-1} - E[x_{k-1}]) + w_{k-1})(same)^t] \\ &= A_{k-1}E[(x_{k-1} - E[x_{k-1}])(x_{k-1} - E[x_{k-1}])^t]A_{k-1}^t + A_{k-1}E[(x_{k-1} - E[x_{k-1}])w_{k-1}^t] \\ &+ E[w_{k-1}(x_{k-1} - E[x_{k-1}])^t]A_{k-1}^t + E[w_{k-1}w_{k-1}^t] \\ &= A_{k-1}P_{k-1}A_{k-1}^t + Q_{k-1}, \end{aligned}$$

by x_{k-1} is independent with w_{k-1}. Substituting P_{k-1}^+ into P_{k-1} back we have

$$P_k^- = A_{k-1}P_{k-1}^+A_{k-1}^t + Q_{k-1}.$$

(3): K_k

Suppose we have x_{k-1} and new measured value y_k with a linear recursive system:

$$\begin{aligned} y_k &= C_k x + v_k, \\ x_k &= x_{k-1} + K_k(y_k - C_k x_{k-1}), \end{aligned}$$

where $y_k - C_k x_{k-1}$ is called the correction term, or innovations, it can be seen as the discrepancy between the ideal observed value and the actual observed value.

doi:10.6342/NTU202400950

First we observe the expected value of the systematic error $x - x_k$, denote as e_k:

$$\begin{aligned} E[e_k] &= E[x - x_k] \\ &= E[x - x_{k-1} - K_k(y_k - C_k x_{k-1})] \\ &= E[x - x_{k-1} - K_k(C_k x + v_k - C_k x_{k-1})] \\ &= E[e_{k-1}] - K_k C_k E[e_{k-1}] - K_k C_k E[v_k] \\ &= (I - K_k C_k) E[e_{k-1}] \end{aligned}$$

Observe the above formula, if the initial estimate is set to the expected value of x ($x_0 = E[x]$), then the recursive formula tells us that for each k, x_k is an unbiased estimator, whatever K_k is.

So how do we choose optimization matrix K_k? One of the most intuitive way is to minimize the variance:

$$E[e_k^t e_k] = E[tr(e_k e_k^t)] = tr P_k,$$

and

$$\begin{aligned} P_k &= E[((I - K_k C_k)e_{k-1} - K_k v_k)((I - K_k C_k)e_{k-1} - K_k v_k)^t] \\ &= (I - K_k C_k) E[e_{k-1} e_{k-1}^t](I - K_k C_k)^t - (I - K_k C_k) E[e_{k-1} v_k^t] K_k^t \\ &- K_k E[v_k e_{k-1}^t](I - K_k C_k)^t + K_k E[v_k v_k^t] K_k^t \\ &= (I - K_k C_k) P_{k-1} (I - K_k C_k)^t + K_k R_k K_k^t \end{aligned}$$

by e_{k-1} is independent with v_k.

Combining the above and finding the minimum with respect to K_k, i.e.

$$\frac{\partial tr P_k}{\partial K_k} = \frac{\partial tr((I - K_k C_k) P_{k-1} (I - K_k C_k)^t + K_k R_k K_k^t)}{\partial K_k} = 0.$$

Recall some rule of partial of matrix:

$$\begin{aligned} &\frac{\partial tr(KBK^t)}{\partial K} = 2KB, \quad \text{if B is symmetry.} \\ &\frac{\partial tr(KA)}{\partial K} = A^t, \\ &\frac{\partial tr(AK^t)}{\partial K} = A. \end{aligned}$$

doi:10.6342/NTU202400950

$$
\begin{aligned}
\Rightarrow 0 &= \frac{\partial tr P_k}{\partial K_k} \\
&= \frac{\partial tr((I - K_k C_k) P_{k-1} (I - K_k C_k)^t + K_k R_k K_k^t)}{\partial K_k} \\
&= \frac{\partial tr(-P_{k-1} C_k^t K_k^t - K_k C_k P_{k-1} + K_k C_k P_{k-1} C_k^t K_k^t)}{\partial K_k} + 2K_k R_k \\
&= -P_{k-1} C_k^t - P_{k-1}^t C_k^t + 2K_k C_k P_{k-1} C_k^t + 2K_k R_k \\
\Rightarrow K_k &= P_{k-1} C_k^t (C_k P_{k-1} C_k^t + R_k)^{-1}.
\end{aligned}
$$

And, after simplification, we can get the equivalent form for P_k :

$$P_k = (I - K_k C_k) P_{k-1}.$$

This form is more convenient for theory deduction or computer calculations, but it may not be symmetrical in numerical.

2.2 Algorithm of Kalman filter

Given a linear dynamic equation and linear measurement equation:

$$
\begin{aligned}
x_k &= A_{k-1} x_{k-1} + B_{k-1} u_{k-1} + w_{k-1}, \\
y_k &= C_k x_k + v_k,
\end{aligned}
$$

where A, B, C are known matrix, u is input, and w_k, v_k are noise, which we assume they are uncorrelated and

$$
\begin{aligned}
w_k &\overset{iid}{\sim} N(0, Q_k), \\
v_k &\overset{iid}{\sim} N(0, R_k).
\end{aligned}
$$

doi:10.6342/NTU202400950

Initial state $x_0^+ = E[x_0]$ and initial covariance $P_0^+ = E[(x_0 - x_0^+)(x_0 - x_0^+)^t]$.
for $k \geq 1$,

$$
\begin{aligned}
& & x_k^- &= A_{k-1}x_{k-1}^+ + B_{k-1}u_{k-1}, \\
& & P_k^- &= A_{k-1}P_{k-1}^+A_{k-1}^t + Q_{k-1}. \\
& update: & K_k &= P_k^-C_k^t(C_kP_k^-C_k^t + R_k)^{-1}, \\
& & x_k^+ &= x_k^- + K_k(y_k - C_kx_k^-) = (I - K_kC_k)x_k^- + K_ky_k, \\
& & P_k^+ &= (I - K_kC_k)P_k^-.
\end{aligned}
$$

2.3 Discussing of the Kalman filter

Some things to pay attention to:

1. We assume that the noise is Gaussian, but in fact our proof only uses zero-mean, uncorrelated, and white. In that case, KF is the optimal linear filter even the noise is non-Gaussian.
2. We assume that the system is linear. **But what if not**?

There are many ways to solve nonlinear systems, and we will introduce extended Kalman filter first.

doi:10.6342/NTU202400950

doi:10.6342/NTU202400950

Chapter 3 Extended Kalman Filter

3.1 Derivation of extended Kalman filter

For linear equations, we have learned that the standard Kalman filter is an optimal filter. However, when dealing with nonlinear equations, alternative methods are required. One approach is to linearize the equations using Taylor expansion and then apply the Kalman filter. This modified filter is commonly referred to as the Extended Kalman Filter.

Given a dynamic equation and measurement equation:

$$x_{k+1} = f(x_k, u_k) + w_k,$$
$$y_{k+1} = h(x_{k+1}) + v_{k+1}.$$

Where f, h are known nonlinear differentiable equation, u is input, and w_k, v_k are noise, which we assume they are uncorrelated and

$$w_k \overset{iid}{\sim} N(0, Q_k),$$
$$v_k \overset{iid}{\sim} N(0, R_k).$$

Suppose we have our best estimate at time k, denote as x_k^+. Now, we linearize function f by Taylor series at (x_k^+, u_k):

$$f(x, u) = f(x_k^+, u_k) + \nabla f|_{x_k^+, u_k} \cdot (x - x_k^+, u - u_k) + \text{H.O.T.}$$

Where H.O.T. mean second and higher order terms. And in general $u - u_k = 0$ because the input is perfect known. Note that if we are uncertain about the input, we can separate u into two parts, one is deterministic and one is unknown, and rewrite the system.

doi:10.6342/NTU202400950

Now we have the approximation:

$$f(x,u) \approx f(x_k^+, u_k) + \nabla_x f|_{x_k^+, u_k}(x - x_k^+),$$

Where $\nabla_x f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$.

Compute the priori estimate x_{k+1}^- :

$$\begin{aligned} x_{k+1}^- &= E[x_{k+1}|Y_k] \\ &= E[f(x_k, u_k) + w_k|Y_k] \\ &= E[f(x_k^+, u_k) + \nabla_x f|_{x_k^+, u_k}(x - x_k^+) + w_k|Y_k] \\ &= f(x_k^+, u_k) + \nabla_x f|_{x_k^+, u_k}(E[x|Y_k] - x_k^+) \\ &= f(x_k^+, u_k), \end{aligned}$$

cause $E[x|Y_k] = x_k^+$.

And covariance matrix for priori estimate, denote as P_{k+1}^-:

$$\begin{aligned} P_{k+1}^- &= E[(x_{k+1} - x_{k+1}^-)(x_{k+1} - x_{k+1}^-)^t] \\ &= E[(x_{k+1} - f(x_k^+, u_k))(x_{k+1} - f(x_k^+, u_k))^t] \\ &= E[(f(x_k^+, u_k) + \nabla_x f|_{x_k^+, u_k}(x - x_k^+) + w_k - f(x_k^+, u_k))(...)^t] \\ &= E[(\nabla_x f|_{x_k^+, u_k}(x - x_k^+) + w_k)(\nabla_x f|_{x_k^+, u_k}(x - x_k^+) + w_k)^t] \\ &= (\nabla_x f|_{x_k^+, u_k}) P_k^+ (\nabla_x f|_{x_k^+, u_k})^t + Q_k. \end{aligned}$$

Now we add the measurement and get a better estimate, a posteriori $\mathbf{x_{k+1}^+}$. Same idea follow from innovation term of the Kalman filter, that is:

$$x_{k+1}^+ = C_{k+1} + K_{k+1} y_{k+1}.$$

doi:10.6342/NTU202400950

By unbiasedness condition, we further calculate that:

$$\begin{aligned}
0 &= E[x_{k+1} - x_{k+1}^+ | Y_{k+1}] \\
&= E[x_{k+1} - C_{k+1} - K_{k+1} y_{k+1} | Y_{k+1}] \\
&= E[x_{k+1} - C_{k+1} - K_{k+1} h(x_{k+1}) - K_{k+1} v_{k+1} | Y_{k+1}]. \\
\Rightarrow C_{k+1} &= E[x_{k+1} - K_{k+1} h(x_{k+1}) - K_{k+1} v_{k+1} | Y_{k+1}] \\
&= E[f(x_k, u_k) + w_k - K_{k+1} h(x_{k+1}) | Y_{k+1}] \\
&= E[f(x_k^+, u_k) + \nabla_x f|_{x_k^+, u_k}(x_k - x_k^+) - K_{k+1} h(x_{k+1}) | Y_{k+1}] \\
&= x_{k+1}^- - K_{k+1} E[h(x_{k+1}) | Y_{k+1}].
\end{aligned}$$

Before proceeding with the computation, we expansion h at x_{k+1}^-:

$$h(x) \approx h(x_{k+1}^-) + \nabla h|_{x_{k+1}^-}(x - x_{k+1}^-),$$

and bring C_{k+1} back into the equation we have:

$$\begin{aligned}
\Rightarrow x_{k+1}^+ &= x_{k+1}^- + K_{k+1} E[y_{k+1} - h(x_{k+1}^-) - \nabla h|_{x_{k+1}^-}(x_{k+1} - x_{k+1}^-) | Y_{k+1}] \\
&= x_{k+1}^- + K_{k+1} E[y_{k+1} - h(x_{k+1}^-) | Y_{k+1}] \\
&= x_{k+1}^- + K_{k+1}(y_{k+1} - h(x_{k+1}^-)).
\end{aligned}$$

We're done calculating a posteriori. Next, we look at the covariance matrix for a posteriori estimate, i.e. $P_{k+1}^+ = E[(x_{k+1} - x_{k+1}^+)(x_{k+1} - x_{k+1}^+)^t]$.
For convenience, we denote $e_{k+1}^+ = x_{k+1} - x_{k+1}^+$.
Let's go further and calculate e_{k+1}^+:

$$\begin{aligned}
e_{k+1}^+ &= x_{k+1} - x_{k+1}^+ \\
&= f(x_k, u_k) + w_k - x_{k+1}^- - K_{k+1}(y_{k+1} - h(x_{k+1}^-)) \\
&= f(x_k, u_k) - f(x_k^+, u_k) + w_k - K_{k+1}(y_{k+1} - h(x_{k+1}^-)) \\
&= \nabla_f|_{(x_k^+, u_k)} e_k + w_k - K_{k+1}(\nabla h|_{x_{k+1}^-}(x_{k+1} - x_{k+1}^-) + v_{k+1}) \\
&= \nabla_f|_{(x_k^+, u_k)} e_k + w_k - K_{k+1}(\nabla h|_{x_{k+1}^-}(f(x_k, u_k) + w_k - f(x_k^+, u_k)) + v_k) \\
&= \nabla_f|_{(x_k^+, u_k)} e_k + w_k - K_{k+1}(\nabla h|_{x_{k+1}^-}(\nabla_f|_{(x_k^+, u_k)} e_k + w_k) + v_{k+1}) \\
&= (I - K_{k+1} \nabla h|_{x_{k+1}^-})(\nabla_f|_{(x_k^+, u_k)} e_k + w_k) - K_{k+1} v_{k+1}.
\end{aligned}$$

doi:10.6342/NTU202400950

Go back and calculate the covariance matrix P_{k+1}^+:

$$\begin{aligned} P_{k+1}^+ &= E[(x_{k+1} - x_{k+1}^+)(x_{k+1} - x_{k+1}^+)^t] \\ &= E[((I - K_{k+1}\nabla h|_{x_{k+1}^-})(\nabla_f|_{(x_k^+, u_k)})e_k^+ + w_k) - K_{k+1}v_{k+1})(same)^t] \\ &= (I - K_{k+1}\nabla h|_{x_{k+1}^-})P_{k+1}^-(I - K_{k+1}\nabla h|_{x_{k+1}^-})^t + K_{k+1}R_{k+1}K_{k+1}^t, \end{aligned}$$

by $P_{k+1}^- = (\nabla_x f|_{x_k^+, u_k})P_k^+(\nabla_x f|_{x_k^+, u_k})^t + Q_k$ we have shown before.
And we optimal of P_{k+1}^+ in trace sense, i.e.

$$\frac{\partial tr P_{k+1}^+}{\partial K_{k+1}} = 0.$$

$$\begin{aligned} \Rightarrow 0 &= \frac{\partial tr P_{k+1}^+}{\partial K_{k+1}} \\ &= \frac{\partial tr((I - K_{k+1}\nabla h|_{x_{k+1}^-})P_{k+1}^-(I - K_{k+1}\nabla h|_{x_{k+1}^-})^t + K_{k+1}R_{k+1}K_{k+1}^t)}{\partial K_{k+1}} \\ &= \frac{\partial tr(-(P_{k+1}^-\nabla h|_{x_{k+1}^-}^t)K_{k+1}^t - K_{k+1}\nabla h|_{x_{k+1}^-}P_{k+1}^- + K_{k+1}(\nabla h|_{x_{k+1}^-}P_{k+1}^-\nabla h|_{x_{k+1}^-}^t)K_{k+1}^t)}{\partial K_{k+1}} \\ &\quad + 2K_{k+1}R_{k+1} \\ &= -P_{k+1}^-\nabla h|_{x_{k+1}^-}^t - (\nabla h|_{x_{k+1}^-}P_{k+1}^-)^t + 2K_{k+1}(\nabla h|_{x_{k+1}^-}P_{k+1}^-\nabla h|_{x_{k+1}^-}^t) + 2K_{k+1}R_{k+1} \\ \Rightarrow K_{k+1} &= P_{k+1}^-\nabla h|_{x_{k+1}^-}^t(\nabla h|_{x_{k+1}^-}P_{k+1}^-\nabla h|_{x_{k+1}^-}^t + R_{k+1})^{-1}. \end{aligned}$$

Now return K_{k+1} to P_{k+1}^+ we have:

$$P_{k+1}^+ = (I - K_{k+1}\nabla h|_{x_{k+1}^-})P_{k+1}^-.$$

It is worth noting that if $h(\cdot)$ is linear, then we have the same form as standard Kalman Filter.

doi:10.6342/NTU202400950

3.2 Algorithm of extended Kalman filter

Consider the following nonlinear dynamic equation with nonlinear measurement equation:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k, \\ y_{k+1} &= h(x_{k+1}) + v_{k+1}, \\ w_k &\overset{iid}{\sim} N(0, Q_k), \\ v_k &\overset{iid}{\sim} N(0, R_k), \end{aligned}$$

Where w_k and v_k are uncorrelated.
Initial state $x_0^+ = E[x_0]$ and initial covariance $P_0^+ = E[(x_0 - x_0^+)(x_0 - x_0^+)^t]$.

for $k \geq 1$,

$$\begin{aligned} & x_k^- = f(x_{k-1}^+), \\ & P_k^- = \nabla_x f|_{x_{k-1}^+, u_{k-1}} P_{k-1}^+ \nabla_x f|_{x_{k-1}^+, u_{k-1}}^t + Q_{k-1}. \\ update: \quad & K_k = P_k^- \nabla h|_{x_k^-}^t (\nabla h|_{x_k^-} P_k^- \nabla h|_{x_k^-}^t + R_k)^{-1}, \\ & x_k^+ = x_k^- + K_k(y_k - h(x_k^-)), \\ & P_k^+ = (I - K_k \nabla h|_{x_k^-}) P_k^-. \end{aligned}$$

3.3 Discussing of the extended Kalman filter

Let's discuss the drawbacks of EKF:

1. The Jacobian may be difficulty to calculate.
2. Only work when model is differentiable.
3. Poor performance if the model is **highly nonlinear**.

Most common issues is 3. In order to improve the issue of high linearity, we will discuss another method, **Unscented Kalman Filter**.

doi:10.6342/NTU202400950

doi:10.6342/NTU202400950

Chapter 4 Unscented Kalman Filter

4.1 Derivation of Unscented Kalman Filter

Because of the shortcomings we mentioned at the end of EKF, the result could be unsatisfactory. The main difference between UKF and EKF is that compared with EKF transform the entire pdf, UKF only bf propagate certain points. But these points are specially designed, they are called Sigma points, and the mean and covariance of these points are the same as the priori mean and priori covariance.

Assume the mean and covariance are $\bar{x}$ and P for random variable $x \in R^n$, respectively. Then a classical way to generate sigma points as follow:
Use Cholesky decomposition, write $nP = BB^t$, where $B = [b_1, ..., b_n] \in R^{n \times n}$.
Create the sigma point:

$$x^i = \bar{x} + b_i, \quad \text{for } i = 1, ..., n,$$
$$x^{i+n} = \bar{x} - b_i, \quad \text{for } i = 1, ..., n.$$

Note that the number of sigma point always be $2n$. More,

$$\frac{\sum_{i=1}^{2n} x^i}{2n} = \bar{x},$$
$$\frac{\sum_{i=1}^{2n}(x^i - \bar{x})(x^i - \bar{x})^t}{2n} = \frac{\sum_{i=1}^{2n} b_i b_i^t}{2n} = \frac{\sum_{i=1}^{n} b_i b_i^t}{n} = \frac{BB^t}{n} = P.$$

Generated the artificial measurement $\{y^i\}_{i=1}^{2n}$ by transform each sigma points with $h(\cdot)$:

$$y^i = h(x_i), \quad i = 1, ..., 2n,$$

doi:10.6342/NTU202400950

and define the sigma mean $\bar{y}_s = \frac{1}{2n}\sum_{i=1}^{2n} y^i$. In order to compare with the real mean $\bar{y}$ and covariance P_y, our analysis method is to use Taylor expansion, and it will match up to the third-order term, which mean that UKF performance well than EKF.

Let's define some symbols first:

Write $\Delta x = x - \bar{x} = [\Delta x_1, ..., \Delta x_n]^t$. Expansion $h(\cdot)$ at $\bar{x}$:

$$h(x) = h(\bar{x}) + (\Delta x_1 \frac{\partial}{\partial x_1} + ... + \Delta x_n \frac{\partial}{\partial x_n})h|_{\bar{x}} + \frac{1}{2!}(\Delta x_1 \frac{\partial}{\partial x_1} + ... + \Delta x_n \frac{\partial}{\partial x_n})^2 h|_{\bar{x}} + ...$$

For convenience, denote $D^k_{\Delta x} = (\sum_{i=1}^{n} \Delta x_i \frac{\partial}{\partial x_i})^k$.

$$\Rightarrow h(x) = h(\bar{x}) + D^1_{\Delta x}h|_{\bar{x}} + \frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}} + ...$$

And the true mean $\bar{y} = E[h(x)] = E[h(\bar{x}) + D^1_{\Delta x}h|_{\bar{x}} + \frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}} + ...]$.

First we calculate the expectation with odd k:

for k=1:

$$E[D^1_{\Delta x}h|_{\bar{x}}] = E[\nabla h|_{\bar{x}} \cdot (x - \bar{x})] = \nabla h|_{\bar{x}} E[x - \bar{x}] = 0$$

for k=3:

$$\begin{aligned} E[D^3_{\Delta x}h|_{\bar{x}}] &= E[(\Delta x_1 \frac{\partial}{\partial x_1} + ... + \Delta x_n \frac{\partial}{\partial x_n})^3 h|_{\bar{x}}] \\ &= E[\sum_{i,j,k=1}^{n} \Delta x_i \Delta x_j \Delta x_k] \frac{\partial}{\partial x_i \partial x_j \partial x_k} h|_{\bar{x}} \\ &= 0 \quad \text{by each } \Delta x_i \text{ has symmetry pdf.} \end{aligned}$$

Same result for other odd k.

Thus, our $\bar{y}$ can be reduced to $\bar{y} = h(\bar{x}) + E[\frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}} + \frac{1}{4!}D^4_{\Delta x}h|_{\bar{x}}...]$.

In fact, the sigma mean preserve this property. We can verified with same method:

$$\bar{y}_s = \frac{1}{2n}\sum_{i=1}^{2n} h(x^i) = \frac{1}{2n}\sum_{i=1}^{2n}[h(\bar{x}) + D^1_{\Delta x^i}h|_{\bar{x}} + \frac{1}{2!}D^2_{\Delta x^i}h|_{\bar{x}} + ...],$$

doi:10.6342/NTU202400950

where $\Delta x^i = x^i - \bar{x}$.

And, similarly, for odd k we also have $\sum_{i=1}^{2n} D^k_{\Delta x^i} h|_{\bar{x}} = 0$ by a simply calculate:

$$\begin{aligned}
\sum_{i=1}^{2n} D^k_{\Delta x^i} h|_{\bar{x}} &= \sum_{i=1}^{2n} (\sum_{j=1}^{n} \Delta x^i_j \frac{\partial}{\partial x_j})^k h|_{\bar{x}} \quad \text{where } \Delta x^i_j = e_j \cdot \Delta x^i \in R \\
&= \sum_{i=1}^{n} (\sum_{j=1}^{n} \Delta x^i_j \frac{\partial}{\partial x_j})^k h|_{\bar{x}} + \sum_{i=n+1}^{2n} (\sum_{j=1}^{n} \Delta x^i_j \frac{\partial}{\partial x_j})^k h|_{\bar{x}} \\
&= 0,
\end{aligned}$$

by $\Delta x^i_j = -\Delta x^{n+i}_j$.

Hence the sigma mean $\bar{y}_s$ can be reduced to

$$\bar{y}_s = h(\bar{x}) + \frac{1}{2n} \sum_{i=1}^{2n} [\frac{1}{2!} D^2_{\Delta x^i} h|_{\bar{x}} + \frac{1}{4!} D^4_{\Delta x^i} h|_{\bar{x}} + ...].$$

So far, we show that UKF mean $\bar{y}_s$ has same first-order with true mean $\bar{y}$.This means it performs at least as well as EKF. But we can, and we will further prove that, in fact, UKF can match the true mean $\bar{y}$ to the Third-order. Since the odd terms of both are zero, what we actually want to prove is that the second order terms are the same:

1.(for true mean)

$$\begin{aligned}
E[D^2_{\Delta x} h|_{\bar{x}}] &= E[(\sum_{j=1}^{n} \Delta x_j \frac{\partial}{\partial x_j})^2 h|_{\bar{x}}] \\
&= E[\sum_{i,j=1}^{n} (e_i \cdot (x - \bar{x}))(e_j \cdot (x - \bar{x})) \frac{\partial}{\partial x_i \partial x_j} h|_{\bar{x}}] \\
&= \sum_{i,j=1}^{n} E[(e_i (x - \bar{x}))((x - \bar{x})^t e_j^t)] \frac{\partial}{\partial x_i \partial x_j} h|_{\bar{x}} \\
&= \sum_{i,j=1}^{n} e_i E[(x - \bar{x})(x - \bar{x})^t] e_j^t \frac{\partial}{\partial x_i \partial x_j} h|_{\bar{x}} \\
&= \sum_{i,j=1}^{n} P_{ij} \frac{\partial}{\partial x_i \partial x_j} h|_{\bar{x}}.
\end{aligned}$$

doi:10.6342/NTU202400950

2.(for sigma mean)

$$\begin{aligned}
\frac{1}{2n}\sum_{i=1}^{2n} D^2_{\Delta x^i}h|_{\bar{x}} &= \frac{1}{2n}\sum_{i=1}^{2n}(\sum_{j=1}^{n}\Delta x^i_j\frac{\partial}{\partial x_j})^2 h|_{\bar{x}} \\
&= \frac{1}{2n} * 2\sum_{i=1}^{n}(\sum_{j=1}^{n}\Delta x^i_j\frac{\partial}{\partial x_j})^2 h|_{\bar{x}} \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{j,k=1}^{n}(e_j\cdot(x^i-\bar{x}))(e_k\cdot(x^i-\bar{x}))\frac{\partial}{\partial x_i\partial x_j}h|_{\bar{x}} \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{j,k=1}^{n}(e_j\cdot b_i)(e_k\cdot b_i)\frac{\partial}{\partial x_i\partial x_j}h|_{\bar{x}} \\
&= \sum_{j,k=1}^{n} e_j[\frac{1}{n}\sum_{i=1}^{n} b_i b_i^t]e_k^t\frac{\partial}{\partial x_i\partial x_j}h|_{\bar{x}} \\
&= \sum_{j,k=1}^{n} e_j P e_k^t\frac{\partial}{\partial x_i\partial x_j}h|_{\bar{x}} \\
&= \sum_{j,k=1}^{n} P_{j,k}\frac{\partial}{\partial x_i\partial x_j}h|_{\bar{x}}.
\end{aligned}$$

Clearly, the second-order terms of the two are the same.

Now that we're done calculating the mean, what about the covariance? Pleasurable, the covariance will also fit to the Third-order:

1.(for true covariance)

$$\begin{aligned}
P_y &= E[(y-\bar{y})(y-\bar{y})^t] \\
&= E[((h(\bar{x})+D^1_{\Delta x}h|_{\bar{x}}+\frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}}+...)-(h(\bar{x})+E[\frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}}+\frac{1}{4!}D^4_{\Delta x}h|_{\bar{x}}...]))(same)^t] \\
&= E[((D^1_{\Delta x}h|_{\bar{x}}+\frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}}+...)-(E[\frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}}+\frac{1}{4!}D^4_{\Delta x}h|_{\bar{x}}...]))(same)^t] \\
&= E[D^1_{\Delta x}h|_{\bar{x}}(D^1_{\Delta x}h|_{\bar{x}})^t]+E[D^1_{\Delta x}h|_{\bar{x}}(\frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}}-E[\frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}}])^t] \\
&+ E[(\frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}}-E[\frac{1}{2!}D^2_{\Delta x}h|_{\bar{x}}])(D^1_{\Delta x}h|_{\bar{x}})^t]+\text{H.O.T},
\end{aligned}$$

where H.O.T represents the fourth and higher terms.

doi:10.6342/NTU202400950

And, since x has a symmetric pdf, so

$$E[D^1_{\Delta x}h|_{\bar{x}}] = E[D^3_{\Delta x}h|_{\bar{x}}] = 0,$$

Hence

$$P_y = E[D^1_{\Delta x}h|_{\bar{x}}(D^1_{\Delta x}h|_{\bar{x}})^t] + \text{H.O.T.}$$

We proceed to further calculate $E[D^1_{\Delta x}h|_{\bar{x}}(D^1_{\Delta x}h|_{\bar{x}})^t]$:

$$\begin{aligned} E[D^1_{\Delta x}h|_{\bar{x}}(D^1_{\Delta x}h|_{\bar{x}})^t] &= E[\sum_{i,j=1}^{n}(e_i \cdot (x-\bar{x})\frac{\partial}{\partial x_i}h|_{\bar{x}})(e_j \cdot (x-\bar{x})\frac{\partial}{\partial x_j}h|_{\bar{x}})^t] \\ &= \sum_{i,j=1}^{n}\frac{\partial}{\partial x_i}h|_{\bar{x}}(e_i E[(x-\bar{x})(x-\bar{x})^t]e_j^t)\frac{\partial}{\partial x_j}h|_{\bar{x}}^t \\ &= \nabla h|_{\bar{x}} P \nabla h|_{\bar{x}}^t. \end{aligned}$$

Combining the above we have

$$P_y = \nabla h|_{\bar{x}} P \nabla h|_{\bar{x}}^t + \text{H.O.T.}$$

2.(for UKF covariance)

$$\begin{aligned} P_s &= \frac{1}{2n}\sum_{i=1}^{2n}(y^i - \bar{y}_s)(y^i - \bar{y}_s)^t \\ &= \frac{1}{2n}\sum_{i=1}^{2n}((h(\bar{x}) + D^1_{\Delta x^i}h|_{\bar{x}} + \frac{1}{2!}D^2_{\Delta x^i}h|_{\bar{x}} + ...) - (h(\bar{x}) + \frac{1}{2n}\sum_{j=1}^{2n}\frac{1}{2!}D^2_{\Delta x^j}h|_{\bar{x}} + ...))(same)^t \\ &= \frac{1}{2n}\sum_{i=1}^{2n}((D^1_{\Delta x^i}h|_{\bar{x}} + \frac{1}{2!}D^2_{\Delta x^i}h|_{\bar{x}} + ...) - \frac{1}{2n}\sum_{j=1}^{2n}(\frac{1}{2!}D^2_{\Delta x^j}h|_{\bar{x}} + ...))(same)^t \\ &= \frac{1}{2n}\sum_{i=1}^{2n}[D^1_{\Delta x^i}h|_{\bar{x}}(D^1_{\Delta x^i}h|_{\bar{x}})^t + D^1_{\Delta x^i}h|_{\bar{x}}(\frac{1}{2!}D^2_{\Delta x^i}h|_{\bar{x}} - \frac{1}{2n}\sum_{j=1}^{2n}\frac{1}{2!}D^2_{\Delta x^j}h|_{\bar{x}})^t \\ &+ (\frac{1}{2!}D^2_{\Delta x^i}h|_{\bar{x}} - \frac{1}{2n}\sum_{j=1}^{2n}\frac{1}{2!}D^2_{\Delta x^j}h|_{\bar{x}})(D^1_{\Delta x^i}h|_{\bar{x}})^t] + \text{H.O.T}, \end{aligned}$$

where H.O.T represents the fourth and higher terms.

doi:10.6342/NTU202400950

And,

$$\frac{1}{2n}\sum_{i=1}^{2n} D^1_{\Delta x^i}h|_{\bar{x}}(\frac{1}{2!}D^2_{\Delta x^i}h|_{\bar{x}} - \frac{1}{2n}\sum_{j=1}^{2n}\frac{1}{2!}D^2_{\Delta x^j}h|_{\bar{x}})^t = 0,$$

by $\Delta x^i = -\Delta x^{i+n}$.

We can see that UKF is a Third-order filter.

4.2 Algorithm of Unscented Kalman Filter

Consider the following nonlinear dynamic equation with nonlinear measurement equation:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_k \in R^n, \\ y_{k+1} &= h(x_{k+1}) + v_{k+1}, \\ w_k &\overset{iid}{\sim} N(0, Q_k), \\ v_k &\overset{iid}{\sim} N(0, R_k), \end{aligned}$$

Where w_k and v_k are uncorrelated.

Set a priori X with symmetry pdf, write initial state $x_0^+ = E[X]$ and initial covariance $P_0^+ = E[(X - x_0^+)(X - x_0^+)^t]$.

for $k \geq 1$,

write $nP_{k-1}^+ = BB^t$, where $B = [b_1, ..., b_n] \in R^{n\times n}$,

$$\begin{aligned} x_{k-1}^i &= x_{k-1}^+ + b_i, \quad \text{for } i = 1, ..., n, \\ x_{k-1}^{i+n} &= x_{k-1}^+ - b_i, \quad \text{for } i = 1, ..., n. \end{aligned}$$

doi:10.6342/NTU202400950

First propagate in $f(\cdot)$:

$$
\begin{aligned}
x_k^i &= f(x_{k-1}^i, u_k), \\
x_k^- &= \frac{1}{2n}\sum_{i=1}^{2n} x_k^i, \\
P_k^- &= \frac{1}{2n}\sum_{i=1}^{2n}(x_k^i - x_k^-)(x_k^i - x_k^-)^t + Q_{k-1}.
\end{aligned}
$$

Write $nP_k^- = BB^t$, where $B = [b_1, ..., b_n] \in R^{n\times n}$,

$$
\begin{aligned}
x_{k-1}^i &= x_{k-1}^+ + b_i, \qquad \text{for } i = 1, ..., n, \\
x_{k-1}^{i+n} &= x_{k-1}^+ - b_i, \qquad \text{for } i = 1, ..., n.
\end{aligned}
$$

Second propagate in $h(\cdot)$:

$$
\begin{aligned}
y_k^i &= h(x_k^i), \\
\bar{y}_k &= \frac{1}{2n}\sum_{i=1}^{2n} y_k^i, \\
P_y &= \frac{1}{2n}\sum_{i=1}^{2n}(y_k^i - \bar{y}_k)(y_k^i - \bar{y}_k)^t + R_k.
\end{aligned}
$$

Update:

$$
\begin{aligned}
P_{xy} &= \frac{1}{2n}\sum_{i=1}^{2n}(x_k^i - x_k^-)(y_k^i - \bar{y}_k)^t, \\
K_k &= P_{xy}P_y^{-1}, \\
x_k^+ &= x_k^- + K_k(y_k - \bar{y}_k), \\
P_k^+ &= P_k^- - K_k P_y K_k^t.
\end{aligned}
$$

Note that the first propagate is not necessary, users make decision whether to use it based on performance and computational cost.

doi:10.6342/NTU202400950

4.3 Discussing of Unscented Kalman Filter

Let's make a summary of UKF:

1. Performs better than EKF, is easy to implement, and doesn't require jacobian.
2. When n (dimension of x) is large, the Cholesky decomposition will cost a lot of computation.
3. We use the assumption that the pdf of x is symmetric, **but what if not**?

We usually assume that the pdf of x is Gaussian, but this may not be able to handle real situations such as bimodal, Fat-tailed, etc.
In order to be usable under any probability distribution, we will discuss another method, **Particle Filter**.

doi:10.6342/NTU202400950

Chapter 5 Particle Filter

5.1 Derivation of Particle Filter

The Particle Filter(PF) is a probability-based estimator, which are method by Monte Carol(MC) approximations to the solutions of the Bayesian estimation equations. The difference between UKF and PF is that PF can approximate any probability distribution using any number of particles. Of course, there is no free lunch in the world. To approximate any probability distribution, a large number of particles is required, and a large number of particles means a significant computational cost.

First, a quick derivation of the Bayesian estimation equation for recursive way to compute the conditional pdf $p(x_k|Y_k)$ ($Y_k = \{y_1, ..., y_k\}$, observed data set up to time k) as follow:

$$\begin{aligned} p(x_k|Y_k) &= \frac{p(Y_k|x_k)}{p(Y_k)}p(x_k) = \frac{p(y_k, Y_{k-1}|x_k)}{p(y_k|Y_{k-1})}\frac{p(x_k|Y_{k-1})p(Y_{k-1})}{p(Y_{k-1}|x_k)} \\ &= \frac{p(x_k, y_k, Y_{k-1})}{p(x_k)p(y_k, Y_{k-1})}\frac{p(x_k, Y_{k-1})p(Y_{k-1})}{p(Y_{k-1})p(Y_{k-1}|x_k)}\frac{p(x_k, y_k)}{p(x_k, y_k)} \\ &= \frac{p(Y_{k-1}|x_k, y_k)p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})p(Y_{k-1}|x_k)} \\ &= \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{p(y_k|Y_{k-1})} \end{aligned}$$

note that y_k is a function of x_k, so $p(Y_{k-1}|x_k, y_k) = p(Y_{k-1}|x_k)$. More,

$$\begin{aligned} p(y_k|Y_{k-1}) &= \int p(y_k, x_k|Y_{k-1})dx_k \\ &= \int p(y_k|x_k, Y_{k-1})p(x_k|Y_{k-1})dx_k \\ &= \int p(y_k|x_k)p(x_k|Y_{k-1})dx_k, \end{aligned}$$

doi:10.6342/NTU202400950

and

$$\begin{aligned} p(x_k|Y_{k-1}) &= \int p(x_k, x_{k-1}|Y_{k-1})dx_{k-1} \\ &= \int p(x_k|x_{k-1}, Y_{k-1})p(x_{k-1}|Y_{k-1})dx_{k-1} \\ &= \int p(x_k|x_{k-1})p(x_{k-1}|Y_{k-1})dx_{k-1} \end{aligned}$$

again, note that x_k is a function of x_{k-1}. Combining the above we get

$$p(x_k|Y_k) = \frac{p(y_k|x_k)p(x_k|Y_{k-1})}{\int p(y_k|x_k)p(x_k|Y_{k-1})dx_k}.$$

Unfortunately, analytical solutions are usually not found due to non-linearity. In particular, when $f(\cdot)$ and $h(\cdot)$ are linear and noise are independent Gaussian, then the analytical solution is KF. But in this case we will not use PF, because it is computationally expensive compared to KF.

5.2 Algorithm of Particle filter

Consider nonlinear dynamic equation with nonlinear measurement equation:

$$\begin{aligned} x_{k+1} &= f_k(x_k, w_k) \\ y_k &= h_k(x_k, v_k) \end{aligned}$$

where w_k and v_k are independent noise with known pdf.

Given a priori $p(x_0)$, and write $p(x_0|Y_0) = p(x_0)$.

Sampling N initial particles $\{x_{0,i}^+\}_{i=1}^N$ from $p(x_0)$.

For $k \geq 1$,

Generate N noise particles $\{w_{k-1}^i\}_{i=1}^N$ from w_{k-1} and obtain a prior particles $x_{k,i}^-$ by

$$x_{k,i}^- = f_{k-1}(x_{k-1,i}^+, w_{k-1}^i).$$

Computer likelihoods q_i for each $x_{k,i}^-$ conditioned on y_k:

$$q_i = p(y_k|x_{k,i}^-)$$

doi:10.6342/NTU202400950

and normalize.

Generate a posterior $x_{k,i}^{+}$ (aka Bootstrap filter):

$$x_{k,i}^{+} = x_{k,j}^{-} \text{ with probability } q_j.$$

Now we have a set of particles and those represent the pdf of $p(x_k|y_k)$, this allows us to calculate the sample mean and sample covariance.

However, in reality, such algorithms can only be applied to a few of distributions, because there are not many distributions that can be sampled from.

5.3 Important sampling

Usually, sampling from $p(x|Y_k)$ is infeasible. Therefore, we sampling from other distribution, $\pi(\cdot)$(aka important distribution, proposal distribution). It needs to satisfy the Markov property, and we can easily sampling from it.

Recall the definition of expectation:

$$E[g(x)|Y_k] = \int g(x)p(x|Y_k)dx.$$

Rewriting the expectation:

$$\int g(x)p(x|Y_k)dx = \int [g(x)\frac{p(x|Y_k)}{\pi(x|Y_k)}]\pi(x|Y_k)dx$$

and sampling N particles from $\pi(x|Y_k)$, say $\{x^i\}_{i=1}^N$, by LLN we can approximation the expectation

$$E[g(x)|Y_k] \approx \frac{1}{N}\sum_{i=1}^{N} g(x^i)\frac{p(x^i|Y_k)}{\pi(x^i|Y_k)}.$$

Next question is, how do we avoid directly calculating $p(x^i|Y_k)$?

Answer is Bayes's rule:

$$p(x^i|Y_k) = \frac{p(Y_k|x^i)p(x^i)}{\int p(Y_k|x)p(x)dx}.$$

doi:10.6342/NTU202400950

Usually, the term $p(Y_k|x^i)$ and $p(x^i)$ are computable, but the normalization constant not. The good news is, actually we don't need to calculate. Rewrite it again:

$$\begin{aligned}
E[g(x)|Y_k] &= \int g(x)p(x|Y_k)dx \\
&= \frac{\int g(x)p(Y_k|x)p(x)dx}{\int p(Y_k|x)p(x)dx} \\
&= \left(\int g(x)\frac{p(Y_k|x)p(x)}{\pi(x|Y_k)}\pi(x|Y_k)dx\right) \Big/ \int \frac{p(Y_k|x)p(x)}{\pi(x|Y_k)}\pi(x|Y_k)dx \\
&\approx \left(\sum_{i=1}^{N} g(x^i)\frac{p(Y_k|x^i)p(x^i)}{\pi(x^i|Y_k)}\right) \Big/ \sum_{j=1}^{N} \frac{p(Y_k|x^j)p(x^i)}{\pi(x^j|Y_k)} \\
&= \sum_{i=1}^{N} w^i g(x^i),
\end{aligned}$$

where

$$w^i = \frac{p(Y_k|x^i)p(x^i)}{\pi(x^i|Y_k)} \Big/ \sum_{j=1}^{N} \frac{p(Y_k|x^j)p(x^i)}{\pi(x^j|Y_k)},$$

and we have $p(x|Y_k) \approx \sum_{i=1}^{N} w^i \delta(x - x^i)$, where δ is Dirac delta function.

The IS is one step algorithm. When calculating each step, it is necessary to reconsider the previous observations once again. This means that as k increases, the computational cost will become unaffordable. To solve this problem, we're going to build sequential version, called as sequential important sampling(SIS). We assume the model $f(\cdot)$ has Markov properties.

Observed that

$$\begin{aligned}
p(x_{0:k}|Y_k) &\propto p(y_k|x_{0:k}, Y_{k-1})p(x_{0:k}|Y_{k-1}) \\
&= p(y_k|x_k)p(x_k, x_{0:k-1}|Y_{k-1}) \\
&= p(y_k|x_k)p(x_k|x_{0:k-1}, Y_{k-1})p(x_{0:k-1}|Y_{k-1}) \\
&= p(y_k|x_k)p(x_k|x_{k-1})p(x_{0:k-1}|Y_{k-1})
\end{aligned}$$

doi:10.6342/NTU202400950

via y_k is a function of x_k, and Markov properties. Sampling from $\pi(x_{0:k}|Y_k)$, say $\{x^i_{0:k}\}$, from the above formula we can compute w^i_k:

$$\begin{aligned} w^i_k &\propto \frac{p(x^i_{0:k}|Y_k)}{\pi(x^i_{0:k}|Y_k)} \propto \frac{p(y_k|x^i_k)p(x^i_k|x^i_{k-1})p(x^i_{0:k-1}|Y_{k-1})}{\pi(x^i_{0:k}|Y_k)} \\ &= \frac{p(y_k|x^i_k)p(x^i_k|x^i_{k-1})}{\pi(x^i_k|x^i_{0:k-1},Y_k)} \frac{p(x^i_{0:k-1}|Y_{k-1})}{\pi(x^i_{0:k-1}|Y_{k-1})} \\ &\propto \frac{p(y_k|x^i_k)p(x^i_k|x^i_{k-1})}{\pi(x^i_k|x^i_{0:k-1},Y_k)} w^i_{k-1} \\ &= \frac{p(y_k|x^i_k)p(x^i_k|x^i_{k-1})}{\pi(x^i_k|x^i_{k-1},Y_k)} w^i_{k-1} \end{aligned}$$

by $\pi(x_{0:k}|Y_k) = \pi(x_k|x_{0:k-1},Y_k)\pi(x_{0:k-1}|Y_k) = \pi(x_k|x_{0:k-1},Y_k)\pi(x_{0:k-1}|Y_{k-1})$ and Markov properties of $\pi(\cdot)$. Finally, normalize these w^i_k.

Because those weights are obtained iteratively, it's easy to have one particle with weight of 1 and the other particles with weight of 0. This is called the degeneracy problem. We can use resampling method to solve this problem.

The idea of resampling is as follows:

Treated $\{x^i_k\}_{i=1}^N$ as a discrete RV such that $p(x = x^i_k) = w^i_k$ for each i. Sampling N particles from such this distribution, and set the weight of each new particle to $1/N$.

In particle, We decide when to use resampling in two ways:

1. Use it every n-th step.
2. We defined effective number of particles

$$N_{eff} \approx 1/\sum_{i=1}^{N}(w^i_k)^2$$

and set a threshold N_{thr}, and then when $N_{eff} < N_{thr}$ do sampling.

5.4 Algorithm of sequential important resampling

Consider nonlinear dynamic equation with nonlinear measurement equation:

$$x_{k+1} = f_k(x_k, w_k)$$
$$y_k = h_k(x_k, v_k)$$

doi:10.6342/NTU202400950

where w_k and v_k are independent noise with known pdf.

Given a priori $p(x_0)$, and write $p(x_0|Y_0) = p(x_0)$.

Sampling N initial particles $\{x_0^i\}_{i=0}^N$ from the prior $p(x_0)$, and set the weight $w_0^i = 1/N$ for all i.

Choose the important distribution $\pi(\cdot)$ which satisfied the Markov property.

For $k \geq 1$

Sampling $\{x_k^i\}_{i=1}^N$ from $\pi(x_k|x_{k-1}^i, Y_k)$ for all i

Calculate the new weights by

$$w_k^i \propto \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{\pi(x_k^i|x_{k-1}^i, Y_k)} w_{k-1}^i$$

and normalize those weights.

If $N_{eff} < N_{thr}$, do resampling.

And we have $p(x_k|Y_k) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i)$, where δ is Dirac delta function.

5.5 Derivation of optimal important distribution

The key to determining whether the performance effectiveness is the priori and important distribution. Here we derived the optimal important distribution(in minimise variance sense) is

$$\pi(x_k|x_{0:k-1}^i, Y_k) = p(x_k|x_{k-1}^i, y_k).$$

Proof:

$$\begin{aligned} E[w_k^i] &= w_{k-1}^i \int \frac{p(y_k|x_k)p(x_k|x_{k-1}^i)}{\pi(x_k|x_{0:k-1}^i, Y_k)} \pi(x_k|x_{0:k-1}^i, Y_k) dx_k \\ &= w_{k-1}^i p(y_k|x_{k-1}^i) \\ E[(w_k^i)^2] &= (w_{k-1}^i)^2 \int \frac{(p(y_k|x_k)p(x_k|x_{k-1}^i))^2}{\pi(x_k|x_{0:k-1}^i, Y_k)} dx_k \end{aligned}$$

doi:10.6342/NTU202400950

hence the variance is

$$\begin{aligned}\text{Var}(w_k^i) &= (w_{k-1}^i)^2\Big[\int \frac{(p(y_k|x_k)p(x_k|x_{k-1}^i))^2}{\pi(x_k|x_{0:k-1}^i, Y_k)}dx_k - p(y_k|x_{k-1}^i)^2\Big] \\ &= (w_{k-1}^i)^2\Big[\int p(y_k|x_k)p(x_k|x_{k-1}^i)\frac{p(y_k|x_k)p(x_k|x_{k-1}^i)}{\pi(x_k|x_{0:k-1}^i, Y_k)}dx_k - p(y_k|x_{k-1}^i)^2\Big]\end{aligned}$$

Then,

$$\begin{aligned}\text{Var}(w_k^i) = 0 \ \text{ iff } \ &\frac{p(y_k|x_k)p(x_k|x_{k-1}^i)}{\pi(x_k|x_{0:k-1}^i, Y_k)} = p(y_k|x_{k-1}^i) \\ \text{iff } \ &\pi(x_k|x_{0:k-1}^i, Y_k) = \frac{p(y_k|x_k)p(x_k|x_{k-1}^i)}{p(y_k|x_{k-1}^i)} \\ = \frac{p(y_k|x_k, x_{k-1}^i)p(x_k, x_{k-1}^i)}{p(y_k, x_{k-1}^i)} &= \frac{p(y_k, x_k, x_{k-1}^i)}{p(y_k, x_{k-1}^i)} = p(x_k|x_{k-1}^i, y_k).\end{aligned}$$

This complete the proof.

In most cases, the optimal important distribution cannot be found, or is difficult to sampling, but for certain models we can still do it, Consider the following models:

$$\begin{aligned}x_k &= f(x_{k-1}) + v_k, \ \ v_k \sim N(0_v, \Sigma_v) \\ y_k &= Cx_k + w_k, \ \ w_k \sim N(0_w, \Sigma_w)\end{aligned}$$

where $f(\cdot)$ is nonlinear function, C is a matrix, v_k and w_k are independent Gaussian noise with known covariance matrix. Then the optimal important distribution $\pi(\cdot)$ is:

$$\pi(x_k|x_{k-1}, y_k) \sim N(m_k, \Sigma)$$

where

$$\begin{aligned}\mu_k &= \Sigma(\Sigma_v^{-1}f(x_{k-1}) + C^t\Sigma_w^{-1}y_k) \\ \Sigma &= (\Sigma_v^{-1} + C^t\Sigma_w^{-1}C)^{-1}\end{aligned}$$

doi:10.6342/NTU202400950

Proof:

$$
\begin{aligned}
p(x_k|x_{k-1}, y_k) &= p(v_k = x_k - f(x_{k-1}), w_k = y_k - Cx_k) \\
&= p(v_k = x_k - f(x_{k-1}))p(w_k = y_k - Cx_k) \\
&= C_0 * exp[\frac{1}{2}(x_k - f(x_{k-1}))^t \Sigma_v^{-1}(x_k - f(x_{k-1})) + \frac{1}{2}(y_k - Cx_k)^t \Sigma_w^{-1}(y_k - Cx_k)] \\
&= C_1 * exp[\frac{1}{2}(x_k^t \Sigma_v^{-1} x_k - 2x_k^t \Sigma_v^{-1} f(x_{k-1})) + \frac{1}{2}((Cx_k)^t] \Sigma_w^{-1} x_k - 2(Cx_k)^t \Sigma_w^{-1} y_k) \\
&= C_1 * exp[\frac{1}{2}(x_k^t(\Sigma_v^{-1} + C^t \Sigma_v^{-1} C)x_k - 2x_k^t(\Sigma_v^{-1} f(x_{k-1}) + C^t \Sigma_w^{-1} y_k))] \\
&= C_1 * exp[\frac{1}{2}(x_k - \mu_k)^t \Sigma^{-1}(x_k - \mu_k)]
\end{aligned}
$$

where C_0, C_1 are constant. This complete the proof.

More, when the observation $h(\cdot)$ is nonlinear, we can do this by locally linearisation such as EKF, or something.

5.6 Rao-Blackwellized Particle Filter

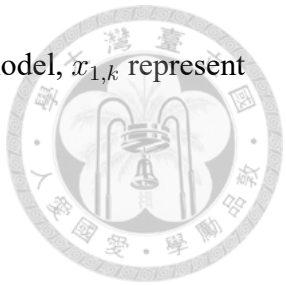
At the end of this chapter, we propose a very practical model, called as Rao-Blackwellized Particle Filter(RBPF). The basic idea of RBPF is to improve the efficiency of particle filters by decomposing the posterior distribution into an analysis part and a MC part. This decomposition can be achieved through Bayes' theorem. In RBPF, the task of the particle filter is to sample particles from the MC part and calculate particle weights by analysis part, then estimate the posterior distribution by combing both. The main advantage of RBPF is that by decomposing the posterior distribution, it can effectively reduce the number of particles required by the particle filter, thereby improving computational efficiency and reduce variance.

Consider the following model:

$$
\begin{aligned}
p(x_{1,k}|x_{k-1}) &\text{ any given form} \\
x_{2,k} &= A_k(x_{1,k-1})x_{2,k-1} + B_k(x_{1,k-1})v_k, \;\; v_k \sim N(0, \Sigma_v) \\
y_k &= C_k(x_{1,k})x_{2,k} + D_k(x_{1,k})w_k, \;\; w_k \sim N(0, \Sigma_w)
\end{aligned}
$$

doi:10.6342/NTU202400950

where $x_{1,k}$ is Markov process, v_k, w_k are independent noise. In this model, $x_{1,k}$ represent the MC part, and $x_{2,k}$ represent the analytically part step.



The algorithm of RBPF as follow:

Given $m_0^i = E[x_{2,0}]$ and p_0^i the covariance matrix of $x_{2,0}$

Sampling $x_{1,0}^i \sim \pi(x_{1,0})$ for $i = 1, ..., N$ For $k \geq 1$,

$$m_k^{-i} = A_k(x_{1,k-1}^i)m_{k-1}^i$$
$$P_k^{-i} = A_k(x_{1,k-1}^i)P_{k-1}^i(A_k(x_{1,k-1}^i))^t + B_k(x_{1,k}^i)\Sigma_v(B_k(x_{1,k}^i))^t$$

Sampling$\{x_{1,k}^i\}_{i=1}^N$ from important distribution $\pi(x_{1,k}|x_{1,0:k}^i, Y_k)$. Update weights by

$$w_k^i \propto w_{k-1}^i \frac{p(y_k|x_{1,0:k}^i, Y_{k-1})p(x_{1,k}^i|x_{1,k-1}^i)}{\pi(x_{1,k}^i|x_{1,0:k-1}^i, Y_k)}$$

for each i, and normalize those weights. More,

$$p(y_k|x_{1,0:k}^i, Y_{k-1}) \sim N(y_k|M_k, \Sigma_k),$$

where

$$M_k = C_k(x_{1,k}^i)m_k^{-i}$$
$$\Sigma_k = D_k(x_{1,k}^i)\Sigma_w(D_k(x_{1,k}^i))^t + C_k(x_{1,k}^i)P_k^{-i}(C_k(x_{1,k}^i))^t.$$

Update $x_{2,k}^i$ via Kalman Filter:

$$K_k^i = P_k^{-i}(C_k(x_{1,k}^i))^t\Sigma_k^{-1}$$
$$m_k^i = m_k^{-i} + K_k^i(y_k - M_k)$$
$$P_k^i = (I - K_k^i C_k(x_{1,k}^i))P_k^{-i}$$
$$= P_k^{-i} - K_k^i\Sigma_k(K_k^i)^t$$

The last equation is symmetry form for computer, which is more convenient to compute.

Then we have

$$p(x_{1,k}, x_{2,k}|Y_k) \approx \sum_{i=1}^N w_k^i N(x_{1,k}^i|m_k^i, P_k^i)\delta(x - x_{1,k}^i)$$

doi:10.6342/NTU202400950

where δ is Dirac function. And hence

$$E[g(x_{1,k}, x_{2,k})|Y_k] \approx \frac{1}{N}\sum_{i=1}^{N} g(x_{1,k}^i, x_{2,k}^i) w_k^i N(x_{1,k}^i | m_k^i, P_k^i).$$

5.7 Discussing of Particle filter

Let's make a summary of PF:

1. The advantages of Particle Filter are that it can handle non-linear and non-Gaussian distribution problems.
2. The disadvantage is that it requires a large amount of computation.
3. If the important distribution is chosen poorly, there may be an issue of sample impoverishment.

doi:10.6342/NTU202400950

Chapter 6 Ensemble Kalman Filter

6.1 Derivation of Ensemble Kalman filter

The last filter we introduce is Ensemble Kalman Filter(EnKF). The advantages of the Ensemble Kalman Filter (EnKF) lie in its ability to perform well with relatively low costs when dealing with high-dimensional systems, which is challenging for the aforementioned four filters. Enkf can be viewed as an approximate version of Kalman Filter, compared to KF which propagates the entire pdf, Enkf only updates the ensemble of vector, this is reason why we called it Ensemble Kalman Filter. The difference between EnKF and PF is that EnKF does not pass particles through weight iterations, thereby avoiding the issue of particle degeneracy. Therefore, there is no need for further resampling.
Recall the Kalman filter:

$$x_k = A_k x_{k-1} + v_k \;\; v_k \sim N(0, Q_k)$$
$$y_k = B_k x_k + w_k \;\; w_k \sim N(0, R_k)$$

where A_k and B_k are matrix, v_k and w_k are independent. Given

$$x_{k-1}|Y_{k-1} \sim N(\mu_{k-1}, \Sigma_{k-1}),$$

we have

$$x_k|Y_{k-1} \sim N(\widehat{\mu}_k, \widehat{\Sigma}_k), \text{ where}$$
$$\widehat{\mu}_k = A_k \mu_{k-1}$$
$$\widehat{\Sigma}_k = A_k \Sigma_k A_k^t + Q_k$$

doi:10.6342/NTU202400950

and update by Kalman gain K_k:

$$K_k = \widehat{\Sigma}_k B_k^t (B_k \widehat{\Sigma}_k B_k^t + R_k)^{-1}$$
$$\mu_k = \widehat{\mu}_k + K_k(y_k - B_k \widehat{\mu}_k)$$
$$\Sigma_k = (I - K_k B_k)\widehat{\Sigma}_k.$$

In EnKF, compared to calculating the entire covariance and Kalman gain, we approximate those by particles $\{x_k^i\}_{i=1}^N$ from $N(\widehat{\mu}_k, \widehat{\Sigma}_k)$, and update each particle by Kalman filter, with artificial noise $y_k^i = y_k + w_k^i$, where w_k^i is sampling from w_k. The whole idea is to achieve statistical improvements by adding small perturbations around the observed date, to be able to capture more directions.

6.2 Algorithm of Ensemble Kalman filter

Given an initial priori $p(x_0)$.
Sampling from $p(x_0)$, say $\{x_0^i\}_{i=1}^N$

For $k \geq 1$,
Generate artificial noise v_k^i from v_k, and do prediction step

$$\widehat{x}_k^i = A_k x_{k-1}^i + v_k^i.$$

Write C_k be sample covariance of $\{\widehat{x}_k^i\}$, and approximate Kalman gain by

$$K_k = C_k B_k^t (B_k C_k B_k^t + R_k)^{-1}.$$

And update by Kalman gain with artificial observations $y_k^i = y_k + w_k^i$
(w_k^i is generated from w_k):

$$x_k^i = \widehat{x}_k^i + K_k(y_k^i - B_k \widehat{x}_k^i).$$

doi:10.6342/NTU202400950

6.3 Initial Ensemble

One of the few drawbacks of the EnKF is that its effectiveness is highly dependent on the selection of the initial ensemble.

One way to chosen initial ensemble is via Karhunen-Loève basis, an idea from PCA. More precisely, we use KL basis when the ensemble size is small.

6.4 Covariance inflation

In addition to the initial ensemble, another key factor that determines the effectiveness is C_k. Usually, Compared with the huge dimension, the size of the ensemble will be much smaller, lead to poor approximation of Kalman gain. Here are two possible solutions:

1. multiply C_k by a number greater than one
2. multiply C_k by a sparse positive definite matrix G_k in Hadamard product

Such this adjustment is called covariance inflation.

6.5 Discussing of Ensemble Kalman filter

Let make a summary for EnKF:

1. For extremely high-dimension.
2. Updating in a linear way instead of reweighting (no degeneracy problem).
3. When ensemble size N goes to infinity, the Enkf converge to the exact Kalman filter.

doi:10.6342/NTU202400950

4. Even though we are assuming a linear model, it also works for nonlinear models
5. EnKF very sensitive to initial ensemble, i.e. the initial prior.

doi:10.6342/NTU202400950

Chapter 7 Numerical Simulation

7.1 Newtonian systems

We employ the Kalman filter to test the Newtonian system, with initial position $x = 0$, initial velocity $v = 0$, acceleration $a = 1$, discretization steps $dt = 1$, and measurement noise is $N(0, 30)$.

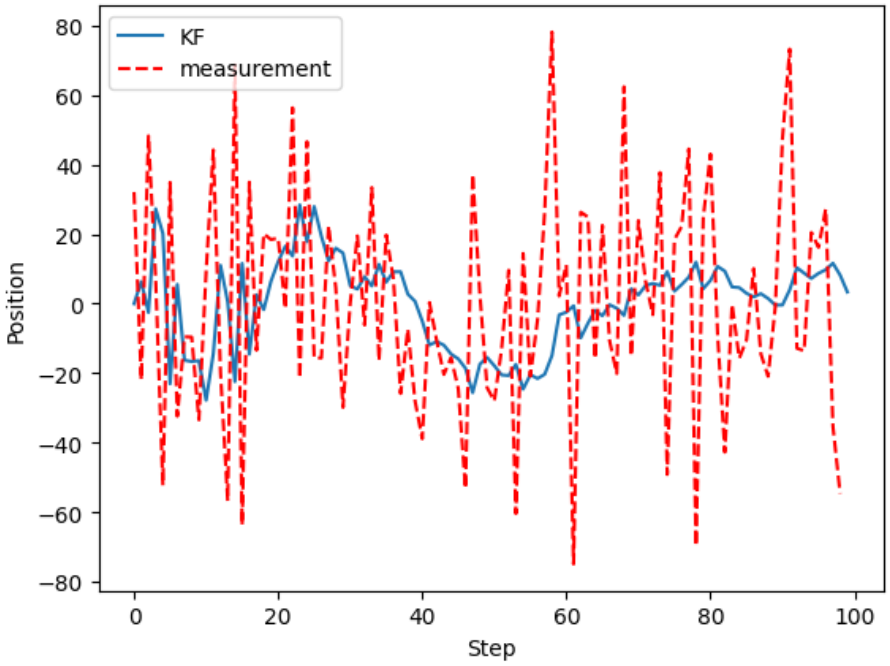


Figure 7.1: time state

In figure 7.1(a), x-axis represents time, y-axis represents position, we test for 100 steps. The red line represents the true observed values, while the blue line depicts the results after applying the Kalman filter. It can be observed that the observed values exhibit significant oscillations due to the addition of substantial noise. However, following the filtering process, the variance notably decreases, and the stability improves as more

doi:10.6342/NTU202400950

observations are incorporated.

Figure 7.2 present standard deviation. The first figure illustrates the initial 15 steps,

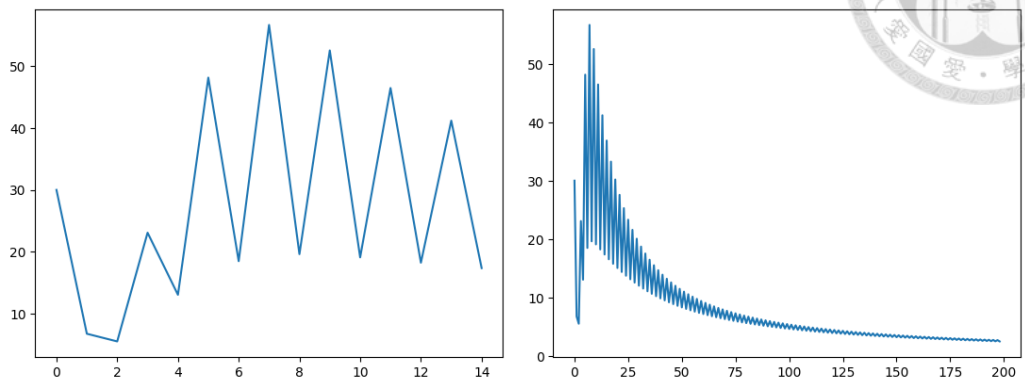


Figure 7.2: standard deviation

showing a series of spikes. This phenomenon arises because the variance is smaller when observed values are present compared to when they are absent. In other words, the variance of the prediction decreases after the update step. The second figure depicts the initial 200 steps, where it can be observed that the variance converges steadily to approximately 2.47.

7.2 Using Particle filters in highly nonlinear equations

Consider the following nonlinear-equations:

$$x_k = \frac{1}{2}x_{k-1} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2(k-1)) + w_k$$
$$y_k = \frac{1}{20}x_k^2 + v_k$$

where $\{w_k\} \sim N(0,3)$ and $\{v_k\} \sim N(0,5)$. Clearly, this is a highly nonlinear equation, so in our experiment, we chose to compare the performance of the Particle Filter (PF) with the Extended Kalman Filter (EKF).

Figure 7.3 compares PF, EKF, and the true state, and it can be observed that PF performs better in regions where there are significant changes in curvature.

Figure 7.4 tests the MSE after gradually increasing the noise, demonstrating the remarkably stable results of PF.,

Figure 7.5 shows that as the number of steps increases, the MSE of PF exhibits better

doi:10.6342/NTU202400950

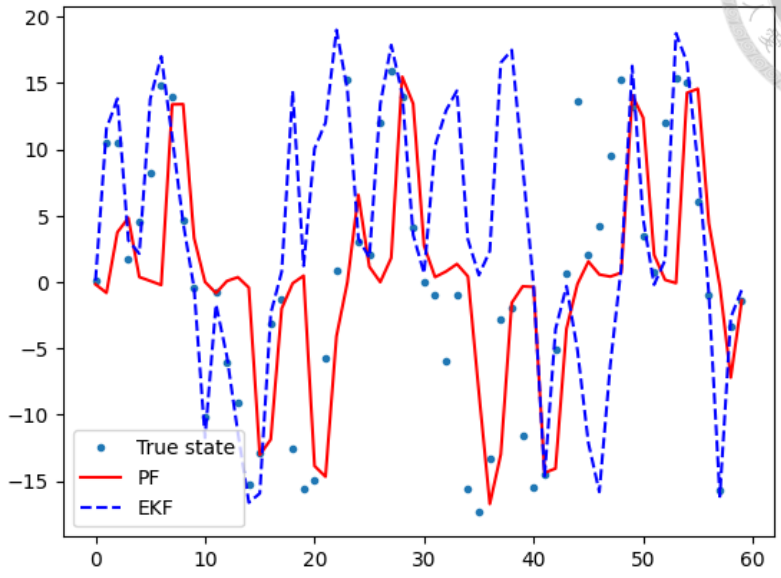


Figure 7.3: time state, PF with 200 particles

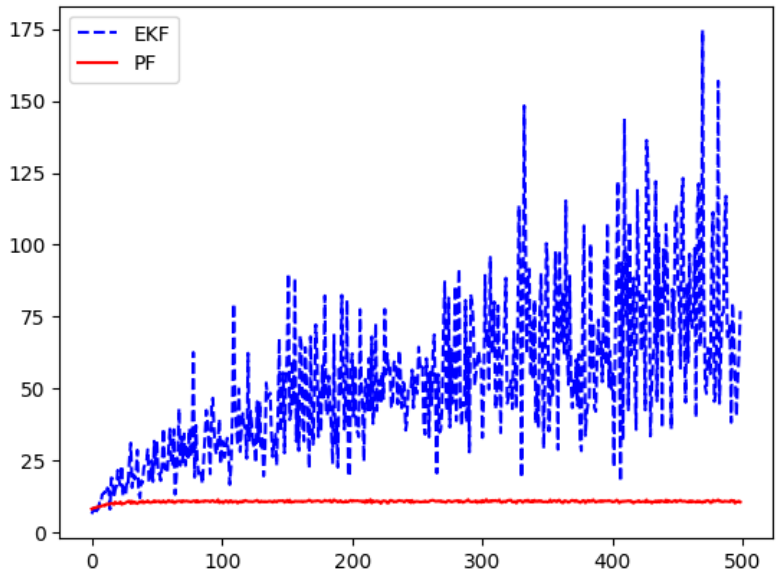
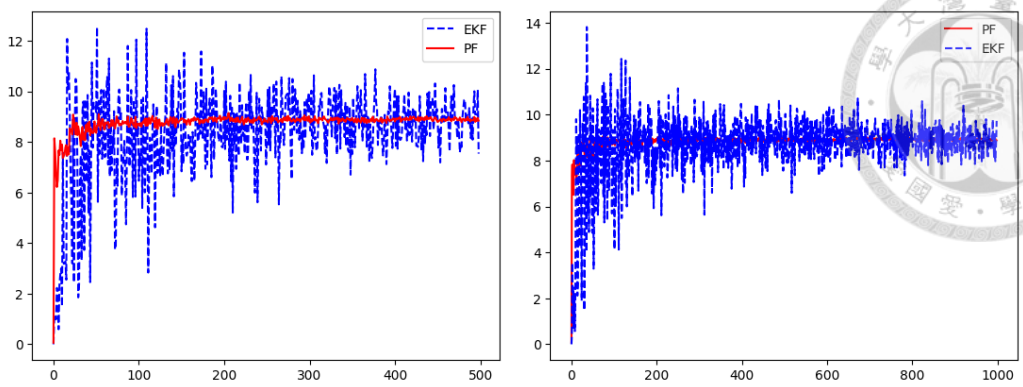


Figure 7.4: time-MSE, PF with 60 steps

doi:10.6342/NTU202400950



(a) 500 steps (b) 1000 steps

Figure 7.5: time-MSE, PF with 200 particles

robustness compared to EKF.

Figure 7.6 demonstrates a decreasing trend in MSE with an increase in the number of

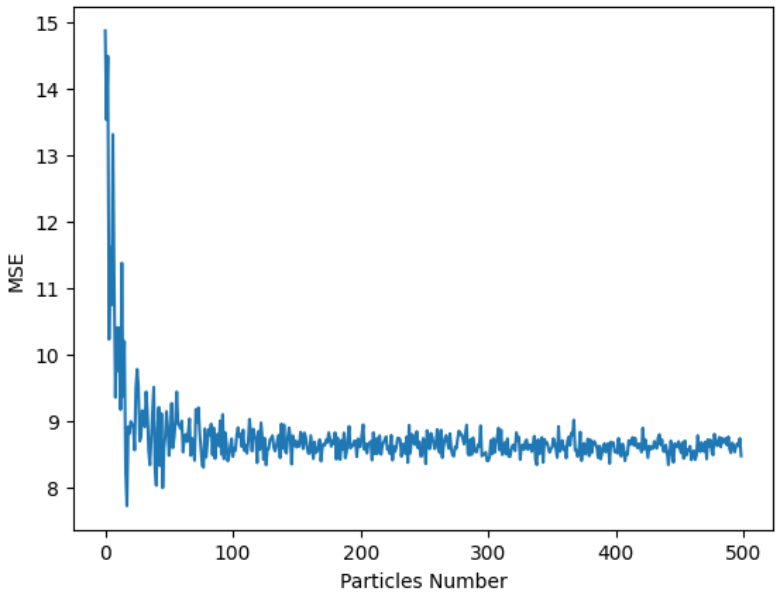


Figure 7.6: time-MSE, PF with 60 steps

particles, confirming the convergence property of Monte Carlo methods.

doi:10.6342/NTU202400950

7.3 Ensemble Kalman Filter in Electrode Impedance Tomography (EIT) Inverse Problem

The EIT problem is

$$\nabla \cdot (\sigma \nabla \mathrm{u}) = 0 \quad \text{in } \Omega, \tag{7.1}$$

$$\mathrm{u} = g \quad \text{on } \partial\Omega. \tag{7.2}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain with smooth boundary, $\sigma > 0$ is conductivitie, u is voltage, g prescribes boundary value.

The forward problem is given σ and g to evaluate the solution $\mathrm{u}_{\sigma,g}$, existence, uniqueness, and stability of $\mathrm{u}_{\sigma,g}$ have already been proof.

But the problem is, in practical, we cannot measure the internal voltage or conductivitie , we can only measure boundary data

$$\mathcal{G}_\sigma(g) = \sigma \frac{\partial \mathrm{u}_{\sigma,g}}{\partial \nu}\bigg|_{\partial\Omega}, \tag{7.3}$$

where ν denotes the unit outer normal to $\partial\Omega$, and $\mathcal{G}_\sigma$ is the Dirichlet to Neumann map (DN map).

The inverse problem is to reconstruction σ by $\mathcal{G}_\sigma$ with many distinct g.

$$\text{Idea: } \left\{ \begin{array}{c} \mathcal{G}_\sigma(g_1) \\ \mathcal{G}_\sigma(g_2) \\ \cdots \end{array} \right\} \xrightarrow{\textbf{reconstruct}} \sigma$$

Here we consider $\Omega = [0,1]\times[0,1]$, and divide Ω into four parts, say $\{\Omega_i\}_{i=1}^4$, and consider $\sigma = \sum_{i=1}^4 \sigma_i 1_{\{\Omega_i\}}$, each $\sigma_i \in (10, 15)$, and $\sigma = 1$ on $\partial\Omega$.

Due to the difficulty in computing the Jacobian matrix, the Extended Kalman Filter (EKF) is impractical here, and the likelihood in the Particle Filter (PF) is also challenging to calculate. We opted for the Ensemble Kalman Filter (EnKF) because it converges well even for nonlinear inverse problems and does not require derivatives.

In the experiment, the settings for the Ensemble Kalman Filter (EnKF) are as follows: the number of ensembles is 50, and the iteration count is 300. Figure 7.7 shows the comparison between the ground truth on the left and the results of the EnKF on the right.

doi:10.6342/NTU202400950

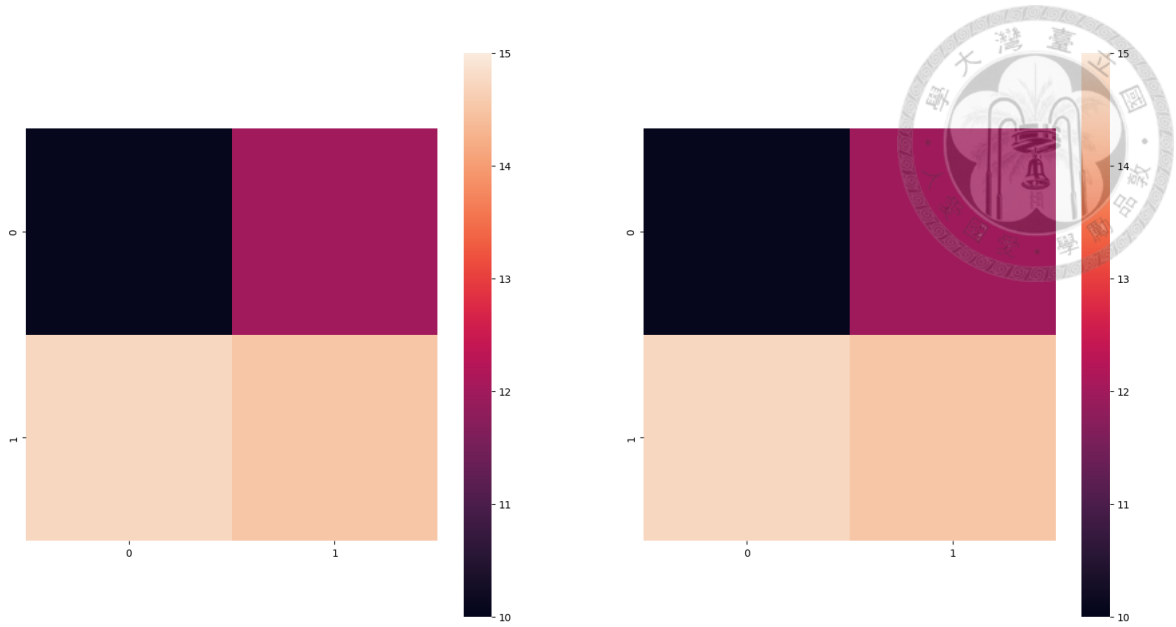


Figure 7.7: ground truth on the left and the results of the EnKF on the right

Figure 7.8 has the iteration count on the x-axis and the difference between the EnKF and the ground truth on the y-axis. It can be observed that they converge into four clusters, each corresponding to the values of four blocks, validating the statistical aspect.

Figure 7.9 demonstrates that the MSE decreases with an increase in the iteration count, confirming the robustness of the EnKF.

doi:10.6342/NTU202400950

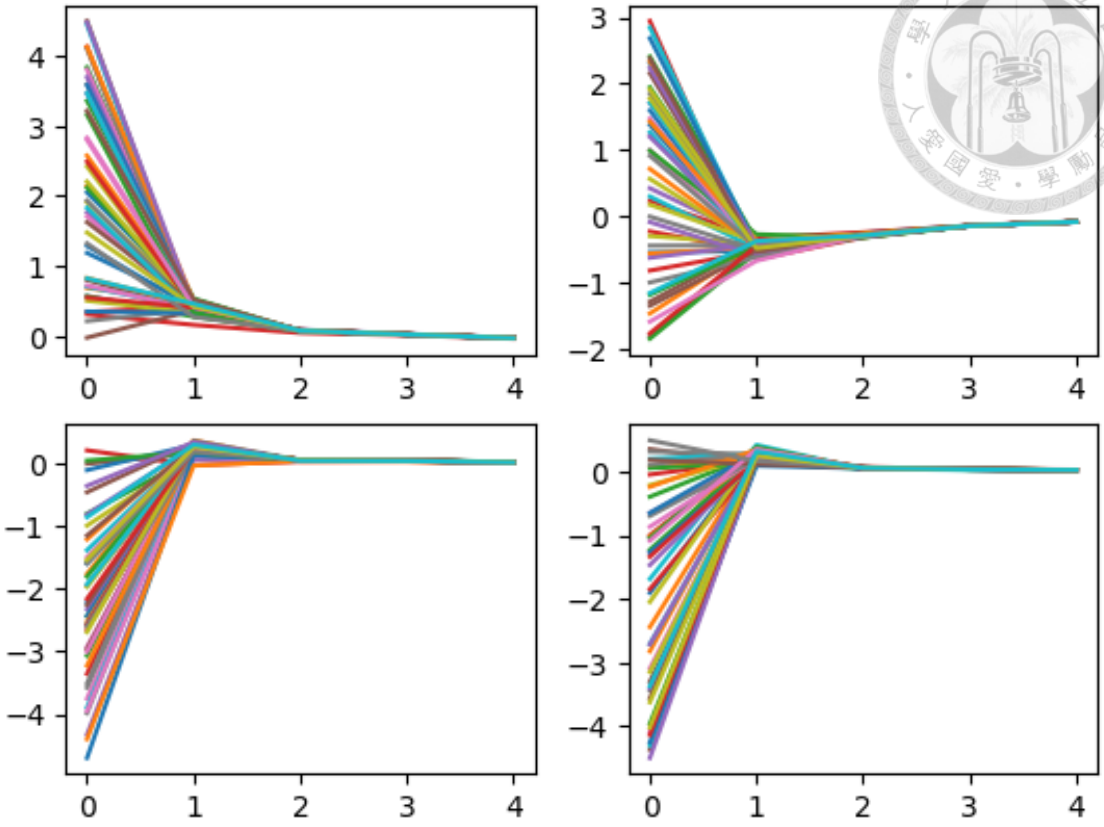


Figure 7.8: performance in each block

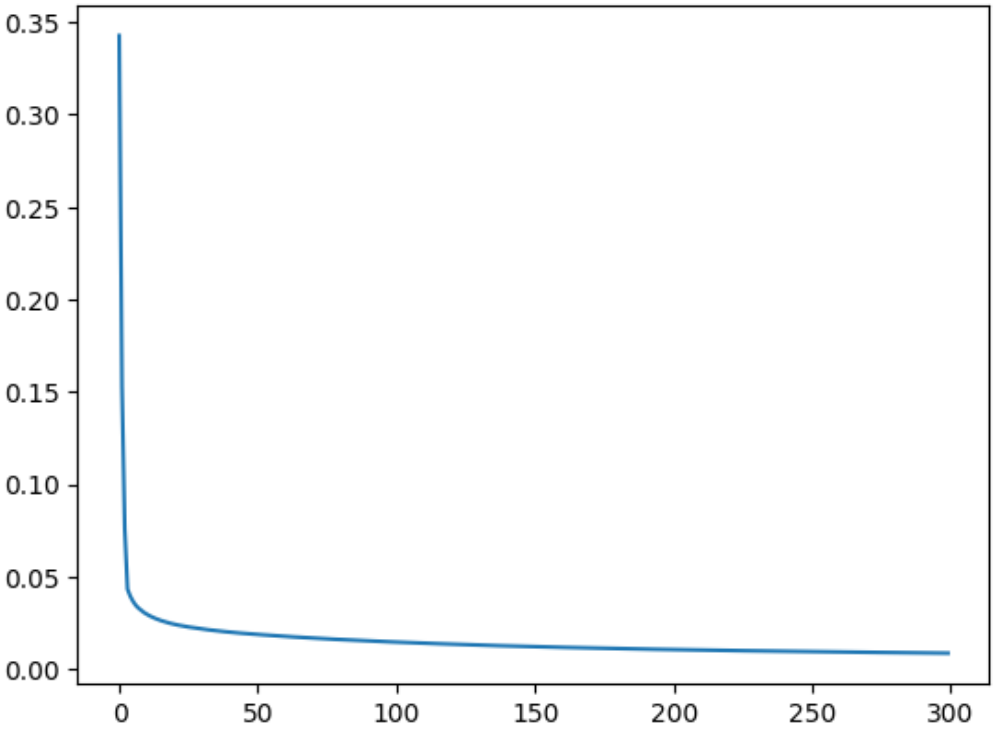


Figure 7.9: time-MSE EnKF with 50 ensembles

doi:10.6342/NTU202400950

doi:10.6342/NTU202400950

References

[1] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. Statistics and computing, 10:197–208, 2000.

[2] M. Katzfuss, J. R. Stroud, and C. K. Wikle. Understanding the ensemble kalman filter. The American Statistician, 70(4):350–357, 2016.

[3] S. Särkkä and L. Svensson. Bayesian filtering and smoothing, volume 17. Cambridge university press, 2023.

[4] D. Simon and T. L. Chia. Kalman filtering with state equality constraints. IEEE transactions on Aerospace and Electronic Systems, 38(1):128–136, 2002.

doi:10.6342/NTU202400950