Applications of Filter Theory in Robotics and Finance

Ruihan Liu

Abstract

The genius of this paper is to review some important results in filter theory and make several numerical experiments about the application of filter theory in the finance and robotics. In detail, we separately discuss the linear and non-linear filter theory in the beginning, then apply particle filter, Monte-Carlo filter and extended Kalman filter in the filtering problems of pose and volatility estimation. In addition, we use EM (Expectation-Maximum) algorithm to analytically infer the unknown parameters in filtering problems and apply Monte-Carlo simulation to numerically infer theses parameters. Moreover, we introduce the RNNs (recurrent neural networks) and construct RNNs filter to deal with the same numerical experiments of pose and volatility estimation. Finally, we compare the performance of different filters and summarize some possible improvement methods about these filters.

Keywords: Kalman filter, Zakai equation, EM algorithm, Monte-Carlo simulation, RNNs filter, Pose estimation, Volatility estimation

Contents

1 Introduction

Filtering theory has many important results and applications in the filed of stochastic control, which mainly concerns about estimating the true value of one system based on incomplete observed information. The history of filtering theory was derived from Kolmogorov [21] and Wiener [37], they extended least square method, which introduced by Gauss, to work on estimation and prediction problems in stochastic process. Later, Wiener formulated initial filter theory based on his former results for its application in engineer field. However, Wiener's method can not be generalized to more complicated problems. In 1960s, Kalman revised Wiener's work and re-formulated filtering problem. In addition, he provided the solution of linear filters in [19], which is one of the most powerful and successful results in both mathematics and engineer fields in past several decades. Although the Kalman filter (KF) can even be applied in some non-linear cases, the main difficulties of more general non-linear filtering problem was still unsolved. Stratonovich [35] and Kushner [23] first developed the study of non-linear filtering problems in around 1960s. They formulated a stochastic partial differential equation for the estimation of signal process with a Markovian observation. Later Mortensen [29], Duncan [8], and Zakai [38] formulated the Duncan-Mortensen-Zakai (DMZ) equation for solving non-linear filtering based on measure reference approach.

On the other hand, the numerical solution of filtering problems is necessary for the real-life application. Although the analytical solution have been derived by Kalman [19] (linear case), Kushner [23] (non-linear case), it is still difficult to exactly solve a Ricatti equation (19) and DMZ equation (9). As a result, some classical numerical approximation methods, such as extended the Kalman filter (EKF) [18], Monte-Carlo simulation method [20], and the particle filter (PF) [5], have been widely applied in both linear and non-linear filtering problems. In addition, the development of deep-learning and neural networks also provide another numerically method for filter theory. Actually, the recurrent neural networks can even work on more complicate and general filtering problems than most of classical methods, we will show this advantage in §7.

For the application of filter theory in finance and engineer, we refer two special estimation problems in track volatility and robotic pose. Therefore, for the preliminary introduction of these problems, we introduce the Lie group theory to mathematically represent and compute the pose of one vehicle in $\mathbb{R}^3$. The merit of Lie group theory is to simply compute the derivatives of matrix function, which plays an important role in the kinematic differential equation of one moving vehicle. For the solution of filtering problem in pose estimation, we apply EKF and make a numerical experiment in §5.6. As for the application in finance, we formulate a parameter inference problem in §6.2, which is analytically solved by Expectation-Maximum method. But the numerical inference of parameters is derived by Monte-Carlo simulation in §6.3, since it is difficult for computers to calculate the analytical solution of EM method. Moreover, the filtering problem of volatility estimation is solved by particle filter in §6.4, which is a non-linear method and can deal with more general filtering problems than EKF.

Finally, we also construct the recurrent neural networks to deal with the same filtering problems of pose and volatility estimation in §7, and we give a theorem in §7.2 to prove that any discrete-filtering problems can be approximated by a RNNs within any accuracy. Furthermore, we train two RNNs filter in §7.3 to solve these two filtering problems and compare the results from EKF and PF with those from RNNs filter. In the final section, we summary the performance of differential filters and discuss the advantages and weakness separately, and also give some possible ideas about the improvement of these filters.

2 Basic Theory

The general model of filtering problem has the following assumption, see [36]. Denote a n-dimensional $X = (X_t)_{t\geq 0}$ as the signal process, which can not be directly observed. And $Y = (Y_t)_{t\geq 0}$ is another m-dimensional stochastic process, which can be observed and related to X. In addition, suppose X and Y are governed by the following equations.

$$\begin{cases} dX_t = a(t, X_t, u_t)dt + b(t, X_t, u_t)dW_t. \\ dY_t = h(t, X_t, u_t)dt + k(t)d\widetilde{W}_t. \end{cases} \tag{1}$$

where $W_t, \widetilde{W}_t$ are two independent Wiener processes and u_t is the control strategy. Our goal is to make best estimation of X based on information given by Y, hence, we define the observation filtration by $\mathcal{F}_t^Y := \sigma\{Y_s : s \leq t\}$. Next, define the cost function as

$$J(Z) = \mathbb{E}[||X_t - Z||^2].$$

where Z is any $\mathcal{F}_t^Y$-measurable random variable. Due to the property of conditional expectation, we obtain

$$\widehat{X}_t = \mathbb{E}[X_t|\mathcal{F}_t^Y] = \arg\min_Z \mathbb{E}[||X_t - Z||^2].$$

which is the projection of X_t on $\mathcal{F}_t^Y$. In addition, for any given function $f : \mathbb{R}^n \to \mathbb{R}$, the best estimation of $f(X_t)$ is defined by

$$\pi_t(f) \equiv \mathbb{E}[f(X_t)|\mathcal{F}_t^Y].$$

The core goal of filter theory is to solve $\pi_t(f)$ by analytical or numerical methods. And there are two kinds of filtering problem, linear filter and non-linear filter, which depends on the form of $a(\cdot), b(\cdot), h(\cdot)$ in (1). In the next two sections, we will first introduce DMZ equation, which allows us to solve $\pi_t(f)$ by a PDE even though $a(\cdot), b(\cdot), h(\cdot)$ are non-linear. For linear case, Kalman and Busy formulated the SDE of $\pi_t(1)$, see (18); and also showed that the cost of $\widehat{X}_t \equiv \mathbb{E}[X_t|\mathcal{F}_t^Y]$

$$\mathbb{E}\left[(X_t - \widehat{X}_t)^T(X_t - \widehat{X}_t)\right].$$

satisfies a Riccati equation, see (19).

3 Non-linear Filtering Theory

3.1 Stratonovich and Zakai equations

In this subsection, we first focus on Kushner and Stratonovich's results, they have shown that $\pi_t(f)$ is governed by a PDE, which gives a analytical way to solve $\pi_t(f)$. Before introducing their conclusions, the following assumptions are necessary.

- (X_t, Y_t) in equation (1) has unique $\mathcal{F}_t$-adapted solution.
- $\det(k(t)) \neq 0$ for all t.
- a, b, h are bounded functions.

In fact, if $a(t,x,u), b(t,x,u)$ in (1) are uniformly Lipschitz respect to x, u and bounded for $t \in [0,T]$, it can conclude that there exists unique solution $X = (X_t)_{t\geq 0}$ with a $L^2(\mathbb{P})$ bounded initial X_0. In addition, we define the operator $\mathscr{L}$ by

$$\mathscr{L}_t^u f(x) = \sum_{i=1}^n a_i(t,x,u)\frac{\partial f}{\partial x_i}(x) + \frac{1}{2}\sum_{i,j=1}^n b_i(t,x,u)b_j(t,x,u)\frac{\partial^2 f}{\partial x_i \partial x_j}(x).$$

where $f \in C^2(\mathbb{R}^n)$. One important technique for computing $\pi_t(f)$ is to find proper measure $\mathbb{Q}$, such that X_t and $\mathcal{F}_t^Y$ are independent under $\mathbb{Q}$. And Girsanov told us that some special Ito processes are in fact Wiener process after changing the measure.

Theorem 3.1.1 (Girsanov)**.** *Let $X = (X_t)_{t\geq 0}$ be an n-dimensional Ito process governed by*

$$X_t = \int_0^t F_s ds + W_t. \tag{2}$$

where $F = (F_t)_{t\geq 0}$ is n-dimensional and satisfies

$$\mathbb{E}_{\mathbb{P}}\left[\int_0^t ||F_s||^2 ds\right] < \infty, \mathbb{E}_{\mathbb{P}}\left[\exp\left(\frac{1}{2}||F_s||^2\right) ds\right] < \infty. \tag{3}$$

and define measure $\mathbb{Q}$ by Radon-Nikodym theorem

$$\frac{d\mathbb{Q}}{d\mathbb{P}}\Big|_t = \exp\left[-\int_0^t F_s \cdot dW_s - \frac{1}{2}\int_0^t ||F_s||^2 ds\right]. \tag{4}$$

then X is a Wiener process under measure $\mathbb{Q}$.

For the proof of Theorem 3.1.1, see §5.5 of [24]. Based on Girsanov's result and (1), it concludes that

Theorem 3.1.2 (Kallianpur-Striebel)**.** *For given time T, define $\mathbb{Q}_T \ll \mathbb{P}$ by*

$$\frac{d\mathbb{Q}_T}{d\mathbb{P}} = \exp\left[-\int_0^T k^{-1}(s)h(s,X_s,u_s)d\widetilde{W}_s - \frac{1}{2}\int_0^T ||k^{-1}(s)h(s,X_s,u_s)||^2 ds\right] := \Lambda_t \tag{5}$$

Under $\mathbb{Q}_T$, $(W_t, \widetilde{Y}_t)_{t\in[0,T]}$ is a $(m+p)$ dimensional $\mathcal{F}_t$-Wiener process, where

$$\widetilde{Y}_t = \int_0^t k^{-1}(s)h(s, X_s, u_s)ds + \widetilde{W}_t = \int_0^t k^{-1}(s)dY_s. \tag{6}$$

In addition, we have $\mathbb{P} \ll \mathbb{Q}_T$ as

$$\frac{d\mathbb{P}}{d\mathbb{Q}_T} = \exp\left[\int_0^T k(s)^{-1}h(s, X_s, u_s)d\widetilde{Y}_s - \frac{1}{2}\int_0^T ||k(s)^{-1}h(s, X_s, u_s)||^2 ds\right]. \tag{7}$$

For the proof of above theorem, it is just a special case of Girsanov's result. Next, denote

$$\sigma_t(f) := \mathbb{E}_{\mathbb{Q}}[f(X_t)|\mathcal{F}_t^Y]. \tag{8}$$

Due to Bayes formula, it concludes that

$$\pi_t(f) = \mathbb{E}_{\mathbb{P}}[f(X_t)|\mathcal{F}_t^Y] = \frac{\mathbb{E}_{\mathbb{Q}}[f(X_t)\Lambda_t|\mathcal{F}_t^Y]}{\mathbb{E}_{\mathbb{Q}}[\Lambda_t|\mathcal{F}_t^Y]} = \frac{\sigma_t(f)}{\sigma_t(1)}.$$

Based on the measure $\mathbb{Q}_t$, we have the following result from [38].

Proposition 3.1.1 (Zakai equation)**.** *Let $f \in C^2(\mathbb{R}^n)$ with bounded derivatives and σ_t defined by (8), then it concludes*

$$\sigma_t(f) = \sigma_0(f) + \int_0^t \sigma_s(\mathscr{L}_s^u f)ds + \int_0^t \sigma_s(k^{-1}(s)h_s^u f)d\widetilde{Y}_s. \tag{9}$$

Proof. According to Ito's rule, it concludes that

$$\begin{aligned} df(X_t)\Lambda_t = \Lambda_t \mathscr{L}_t^u f(X_t)dt + \Lambda_t(\bigtriangledown f(X_t)) \cdot b(t, X_t, u_t)dW_t \\ + \Lambda_t f(X_t)(k^{-1}(t)h(t, X_t, u_t))d\widetilde{Y}_t. \end{aligned}$$

Since all above integrands are $L^2(\mathcal{B}_{[0,t]} \times \mathbb{Q}_t)$, taking the conditional expectation of $\mathbb{E}_{\mathbb{Q}_t}[\,\cdot\,|\mathcal{F}_t^Y]$ and apply Lemma 3.1.1, we can obtain (9). □

Lemma 3.1.1. *Suppose $F = (F_t)_{t\geq 0}$ such that $\mathbb{E}_{\mathbb{P}}[\int_0^t F_s^2 ds] < \infty$ for any finite $t \geq 0$, and W_t is a $\mathcal{F}_t$ Wiener process, then it concludes that*

$$\mathbb{E}_{\mathbb{P}}\left[\int_0^t F_s dW_s \middle| \mathcal{F}_t^W\right] \equiv \int_0^t \mathbb{E}_{\mathbb{P}}\left[F_s|\mathcal{F}_s^W\right] dW_s. \tag{10}$$

where $\mathcal{F}_t^W := \sigma\{W_s : s \leq t\}$.

The proof of Lemma 3.1.1 see §7.2 in [36]. Now, we have concluded the SDE of $\sigma_t(f)$, so $\pi_t(f)$ can be simply derived as a corollary from above proposition by applying Ito's rule. And the following conclusion are given by Kushner [23] and Stratonovich [35].

Proposition 3.1.2 (Kushner-Stratonovich equation)**.** *Let $f \in C^2(\mathbb{R}^n)$ with bounded derivatives, then it concludes*

$$\pi_t(f) = \pi_0(f) + \int_0^t \pi_s(\mathscr{L}_s^u f)ds + \int_0^t k(s)^{-1}[\pi_s(h_s^u f) - \pi_s(h_s^u)\pi_s(f)]d\widehat{B}_s. \tag{11}$$

where $h_s^u(x) = h(s, x, u_s)$ and $d\widehat{B}_s = k(s)^{-1}dY_s - \pi_s(k(s)^{-1}h_s^u)ds$.

As we can see, PDE (9) has simpler form than (11), another problem arise. For example, in order to solve $\pi_t(f)$ from (11), we need the explicit expression of $\pi_t(\mathscr{L}_s^u f)$ and $\pi_t(h_s^u f)$. Unfortunately, they even can not be expressed as the function of $\pi_t(f)$ in general case. One way to solve this trouble is to consider the Kolmogorov forward equation, that is, let $p_t(x)$ be the conditional density related to observation filtration $\mathcal{F}_t^Y$ under measure $\mathbb{P}$, i.e.

$$\pi_t(f) = \mathbb{E}_{\mathbb{P}}[f(X_t)|\mathcal{F}_t^Y] = \int_{\mathbb{R}^n} f(x)p_t(x)dx.$$

Next, taking it into (11), and formally integrating by parts, we conclude

$$dp_t(x) = \mathscr{L}_t^u p_t(x)dt + p_t(x)[k^{-1}(t)(h(t, x, u_t) - \pi_t(h_t^u))](d\widetilde{Y}_t - \pi_t(k^{-1}(t)h_t^u)dt).$$

Similarly, if we denote the conditional density $q_t(x)$ under measure $\mathbb{Q}$ and (9), it concludes that

$$dq_t(x) = \mathscr{L}_t^u q_t(x)dt + q_t(x)(k^{-1}(t)h(t, x, u_t))d\widetilde{Y}_t. \tag{12}$$

$$p_t(x) = \frac{q_t(x)}{\int_{\mathbb{R}^n} q_t(x)dx}. \tag{13}$$

The advantage of this method is clear, by (12), the SDE of $q_t(x)$ is linear. And we will borrow this idea in §6.1. Although it is still hard to deal with, some numerical approximation such as particle filters and spectral methods can give optimal estimation of $\pi_t(f)$, and we will introduce these techniques in §6.4.

3.2 Approximation of Zakai equation

In this subsection, we will introduce some important results about the numerical approximation of Zakai equation, which is essential in exactly solving real-life filtering problems. In detail, there are several useful methods such as implicit Euler approximation and spectral method (Hermite polynomial) [17], and splitting up method [3], and polygonal approximation [15]. The following is a summary of these results. First, we ignore the control u and let $k(t) \equiv 1$, or multiply $k^{-1}(t)$ on each side of (1), and assume the operator $\mathscr{L}_t^u$ is uniformly elliptic, that is, for any $\xi \in \mathbb{R}^n$, there exits $\beta > 0$ such that

$$\sum_{i,j=1}^n b_i(t, x)b_j(t, x)\xi_i\xi_j \geq \beta||\xi||^2.$$

In addition, suppose

$$a(t, x) \in L^\infty([0, \infty) \times \mathbb{R}^n; \mathbb{R}^n), b(t, x) \in L^\infty\left([0, \infty) \times \mathbb{R}^n; \mathbb{R}^{n\times n}\right)$$

where $a(t,x), b(t,x)$ are Lipschitz in x, uniformly in t. And let

$$h(t,x) \in L^\infty([0,\infty) \times \mathbb{R}^n; \mathbb{R}^m)$$

By the variation formulation of PDE from Lions [25], we can regard $\mathscr{L}_t$ as an operator in Hilbert space $\mathcal{H} := L^2(\mathbb{R}^n)$ and rewrite it in divergence form

$$\mathscr{L}_t := \sum_{i,j=1}^n \frac{\partial}{\partial x_i}\left(b_i b_j \frac{\partial}{\partial x_j}\right) + \sum_{i=1}^n \left(a_i - \frac{\partial b_i b_j}{\partial x_j}\right)\frac{\partial}{\partial x_i}.$$

Hence the adjoint of $\mathscr{L}_t$ is

$$\mathscr{L}_t^* := \sum_{i,j=1}^n \frac{\partial}{\partial x_i}\left(b_i b_j \frac{\partial}{\partial x_j}\right) - \sum_{i=1}^n \frac{\partial}{\partial x_i}\left(a_i - \frac{\partial b_i b_j}{\partial x_j}\right).$$

Then define the operator $\mathscr{B}_t$ by

$$\mathscr{B}_t f(x) := f(x)h(t,x).$$

As a result, the conditional density $q_t(x)$ in (12) satisfies the following SDE

$$dq_t(x) - \mathscr{L}_t^* q_t(x)dt = \mathscr{B}_t q_t(x) \cdot d\widetilde{W}_t. \tag{14}$$

We will focus on the existence and uniqueness of solution (14), and the following theorem is valid.

Theorem 3.2.1 (Uniqueness of density solutions)**.** *With above conditions of $a(t,x), b(t,x), h(t,x)$, then for each $p_0(\cdot) \in \mathcal{H}$, these exists one unique solution $p_t(\cdot)$ of (14), such that*

$$p_t(\cdot) \in L^2([0,T]; H^1(\mathbb{R}^n)) \bigcap L^2((\Omega,\mathcal{F},\mathbb{P}); C([0,T];\mathcal{H})) \tag{15}$$

For the proof of above theorem, notice that the linear operator $\mathscr{L}_t$ satisfies the maximal monotone condition for some $\lambda > 0$

$$\langle(\mathscr{L}_t + \lambda \boldsymbol{I})\phi, \phi\rangle \geq \beta||\phi||^2, \ \forall \phi \in H^1(\mathbb{R}^n)$$

Hence, we can construct the resolvent of $\mathscr{L}_t$ by $J_\lambda := (\boldsymbol{I} + \lambda\mathscr{L})^{-1}$ and Yosida approximation $\mathscr{L}_t^\lambda := \lambda^{-1}(I - J_\lambda)$, see §6 of [4], then apply Hille–Yosida theorem, see appendix of [6] to conclude Theorem 3.2.1.

Now we summary the implicit Euler approximation. For $m, n \in \mathbb{N}^+$ and time interval $[0,T]$, let $\Delta t = T/m$ and $y_k := y(k\Delta t)$, then for $k = 1, \cdots, m$, $q_{m,n}^k$ is the solution of

$$q^k - q^{k-1} - \Delta t\left(\mathscr{L}_t^n(y_{k-1}) + \rho \boldsymbol{I}\right) q^k = \Delta t \rho p^{k-1} + \mathscr{B}_t^n(y_{k-1})p^{k-1}(y_k - y_{k-1}). \tag{16}$$

where $\mathscr{L}_t^n, \mathscr{B}^n$ are the Yosida approximation of $\mathscr{L}_t^*, \mathscr{B}_t$ with $\lambda = 1/n$, and $\rho > 0$ such that

$$\langle \mathscr{L}_t^n(y)\phi,\phi\rangle - \frac{1}{2}||\mathscr{B}_t^n(y)\phi||^2 + \rho||\phi||^2 \geq \frac{1}{2}\beta||\phi||^2, \forall \phi \in H^1(\mathbb{R}^n), y \in \mathbb{R}^m.$$

In [17], it shows that the following approximation result.

Theorem 3.2.2. *For given $\{q_0^n(x)\}_{n=1}^{\infty}$ such that $\lim_{n\to\infty} ||q_0^n - q_0||_{L^2(\mathbb{R}^n)} = 0$, $\{q_{m,n}^k\}$ generated by (16) with initial q_0^n converges to solution $q_t(x)$ of (14) as $m, n \to \infty$. Moreover, define $q_{m,n}^t = q_{m,n}^k$ for $t \in [k\Delta t, (k+1)\Delta t)$, which converges to q_t strongly in $L^2([0,T] \times \Omega; H^1(\mathbb{R}^n))$.*

The approximation technique of Zakai equation is indeed useful for solving filtering problems, and (16) provides us a feasible way to numerical approximate the conditional density $q_t(x)$. In §7.2, we will use this approximation result to show the effectiveness of RNNs filter.

4 Kalman Filtering Theory

4.1 Kalman-Bucy filter

In this subsection, we choose special kinds of a, b, h, k defined by

$$\begin{cases} a(t, X_t, u_t) := A(t)X_t + B(t)u_t. \\ b(t, X_t, u_t) := C(t). \\ h(t, X_t) := H(t)X_t. \\ k(t) := K(t). \end{cases} \tag{17}$$

where $A(t), B(t), C(t), H(t), K(t)$ are non-random matrices with dimension of $n \times n, n \times k, n \times m, p \times n$ and $p \times p$. Furthermore, we give the following assumptions.

(i) X_0 is a Gaussian random variable.

(ii) (X_t, Y_t) in equation (1) has unique $\mathcal{F}_t$-adapted solution.

(iii) $\det(K(t)) \neq 0$ for all t.

(iv) $A(t), B(t), C(t), H(t), K(t), K(t)^{-1}$ are continuous.

In addition, let $f \equiv 1$ so that $\pi_t(f) \equiv \widehat{X}_t$. Then we have the following result.

Theorem 4.1.1 (Kalman-Bucy)**.** *Under suitable conditions on the control u_t, it concludes that*

$$d\widehat{X}_t = A(t)\widehat{X}_t dt + B(t)u_t dt + S_t(K(t)^{-1}H(t))dZ_t. \tag{18}$$

$$\frac{dS_t}{dt} = A(t)S_t + S_t A(t)^T + (S_t H(t))^T (K(t)K(t)^T)^{-1}(S_t H(t)) + C(t)C(t)^T. \tag{19}$$

where $S_t = \mathbb{E}[(X_t - \widehat{X}_t)(X_t - \widehat{X}_t)^T]$ and $dZ_t = K(t)^{-1}(dY_t - H(t)\widehat{X}_t dt)$ is called the innovation Winer process.

The proof of Theorem 4.1.1, see [22]. For the proper condition of control u_t, one can ensure that if u_t is non-random and bounded locally, the above proposition is valid, such as $u \equiv 0$, i.e. no control case. For more general kinds of u_t, define $\mathcal{F}_t^{Y,u} := \sigma\{Y_s^u : s \leq t\}$, we have the following results from §7.3 and §7.6 of [36].

Theorem 4.1.2. *If $u \in \bigcap_{t>0} L^1(\mathcal{B}_{[0,t]} \times \mathbb{P})$ and $\mathcal{F}_t^{Y,u} = \mathcal{F}_t^{Y,0}$ for $t \geq 0$, then (18) and (19) hold.*

In fact, we can give some more concrete expression of u.

Theorem 4.1.3. *Let $\alpha(t, x) : [0, \infty) \times \mathbb{R}^n \to \mathbb{R}^k$ is Lipschitz and $\widetilde{X} = (\widetilde{X}_t)_{t \geq 0}$ satisfies*

$$d\widetilde{X}_t = (A(t)\widetilde{X}_t + B(t)\alpha(t, \widetilde{X}_t))dt + S_t(K(t)^{-1}H(t))^T K(t)^{-1}(dY_t^u - H(t)\widetilde{X}_t dt). \tag{20}$$

where the control $u_t = \alpha(t, \widetilde{X}_t)$ and $\widetilde{X}_0 = \widehat{X}_0$. Then there exists unique $(X_t^u, Y_t^u, \widetilde{X}_t)_{t \geq 0}$ such that u satisfies Theorem 4.1.2 and $\widetilde{X}_t = \widehat{X}_t$.

Another view to find proper control is by constructing the cost function (Linear-Quadratic cost, Gaussian) as the following

$$J_t(u) = \mathbb{E}\left[\int_0^t (w(u_s) + v(X_s^u))ds + z(X_t^u)\right].$$

where $X^u = (X_t^u)_{t\geq 0}$ is the solution of linear filtering problem for given control $u = (u_t)_{t\geq 0}$. Our goal is to find the optimal control u^* which minimizing the cost function $J_t(u)$. Next, define matrices $P(s), Q(s), R$ such that

(i) $P(s), Q(s)$ are continuous on $[0, t]$.

(ii) $P(s), R$ are semi-definite and symmetrical on $[0, t]$.

(iii) $Q(s)$ are positive-definite and symmetrical on $[0, t]$.

Then let $v(X_s^u) = (X_s^u)^T P(s) X_s^u$, $w(u_s) = u_s^T Q(s) u_s$ and $z(X_t^u) = (X_t^u)^T R X_t^u$, the following result is valid.

Theorem 4.1.4 (LQG control). *Suppose $F = (F_s)_{s\in[0,t]}, M = (M_s)_{s\in[0,t]}$ is the unique solution of*

$$\begin{cases} \frac{d}{ds}F_s = A(s)F_s + F_sA(s)^T - F_sH(s)^{-1}(K(s)K(s)^T)^{-1}H(s)F_s + C(s)C(s)^T. \\ \frac{d}{ds}M_s = -A(s)M_s - M_sA(s)^T + M_sB(s)Q(s)^{-1}B(s)^TM_s - P(s). \end{cases} \tag{21}$$

with boundary value $F_0 = \mathrm{Cov}(X_0)$ and $M_t = R$, then $u_s^* = -Q(s)^{-1}B(s)^TM_s\widehat{X}_s$, where

$$\begin{aligned} d\widehat{X}_s =& (A(s)B(s)Q(s)^{-1}B(s)^TM_s)\widehat{X}_s ds \\ &+ F_s(K(s)^{-1}H(s))^TK(s)^{-1}(dY_s^{u^*} - H(s)\widehat{X}_s ds). \end{aligned} \tag{22}$$

and $\widehat{X}_0 = \mathbb{E}[X_0]$, $\widehat{X}_s = \widehat{X}_s^{u^*}$ and $F_s = S_s$ for $s \in [0, t]$.

4.2 Discrete-time Kalman filter

For numerical approximation and real-life application, the observation information is usually discrete, in other words, what we known is a finite time sequence of information. Hence, we will consider the discrete-time filtering model.

$$\begin{cases} \boldsymbol{x}_k = A_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{u}_k + \boldsymbol{w}_k. & \text{Signal} \\ \boldsymbol{y}_k = C_k\boldsymbol{x}_k + \boldsymbol{n}_k. & \text{Observation} \end{cases} \tag{23}$$

where $k = 1, \cdots, K$. In addition, $\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k \in \mathbb{R}^n$ and $\boldsymbol{y}_k, \boldsymbol{n}_k \in \mathbb{R}^m$ for all k; hence $A_k \in \mathbb{R}^{n\times n}$ and $C_k \in \mathbb{R}^{m\times n}$ are constant matrix. The initial distribution of $\boldsymbol{x}_0 \sim \mathcal{N}(\bar{\boldsymbol{x}}_0, \boldsymbol{P}_0)$ and noise $\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q}_k), \boldsymbol{n}_k \sim \mathcal{N}(0, \boldsymbol{R}_k)$ are known. As a result, we have the following sequence of observed events.

$$\overbrace{\underbrace{\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{u}_1, \boldsymbol{y}_1, \cdots, \boldsymbol{u}_{k-1}, \boldsymbol{y}_{k-1}}_{\widehat{\boldsymbol{x}}_{k-1}}, \boldsymbol{u}_k, \boldsymbol{y}_k}^{\widehat{\boldsymbol{x}}'_{k-1}}, \cdots$$

where $\widehat{\boldsymbol{x}}_k$ be the best estimation of $\boldsymbol{x}_k$ based on information $\mathcal{F}_k := \sigma\{\boldsymbol{x}_0, \boldsymbol{y}_i, \boldsymbol{u}_i : i \leq k\}$ and $\widehat{\boldsymbol{x}}'_{k-1}$ be the best estimation of $\boldsymbol{x}_k$ based on $\mathcal{F}_k$. In addition, we denote

$$\boldsymbol{x}_{0:K} = (\boldsymbol{x}_0, \cdots, \boldsymbol{x}_K).$$

as well $\boldsymbol{v}_{1:K}, \boldsymbol{y}_{0:K}$. Based on Bayes rule, the best estimation $\widehat{\boldsymbol{x}}_{0:K}$ of $\boldsymbol{x}_{0:K}$ is

$$\widehat{\boldsymbol{x}}_{0:K} := \arg\max_{\widehat{\boldsymbol{x}}_{0:K}} p(\widehat{\boldsymbol{x}}_{0:K}|\mathscr{F}_K) = \arg\max_{\widehat{\boldsymbol{x}}_{0:K}} p(\boldsymbol{y}_{0:K}|\widehat{\boldsymbol{x}}_{0:K}) p(\widehat{\boldsymbol{x}}_{0:K}|\boldsymbol{u}_{1:K}, \bar{\boldsymbol{x}}_0).$$

Moreover, since

$$p(\boldsymbol{y}_{0:K}|\widehat{\boldsymbol{x}}_{0:K}) = \prod_{k=0}^{K} p(\boldsymbol{y}_k|\widehat{\boldsymbol{x}}_k).$$

$$p(\widehat{\boldsymbol{x}}_{0:K}|\boldsymbol{u}_{1:K}, \bar{\boldsymbol{x}}_0) = p(\widehat{\boldsymbol{x}}_0|\bar{\boldsymbol{x}}_0) \prod_{k=1}^{K} p(\widehat{\boldsymbol{x}}_k|\widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k).$$

Hence, we define the cost function $J(\boldsymbol{x}_{0:K})$ as the following

$$J(\widehat{\boldsymbol{x}}_{0:K}) = \sum_{k=0}^{K} [J_v(\widehat{\boldsymbol{x}}_k) + J_y(\widehat{\boldsymbol{x}}_k)].$$

where

$$J_v(\widehat{\boldsymbol{x}}_k) := \begin{cases} \frac{1}{2}(\widehat{\boldsymbol{x}}_0 - \boldsymbol{x}_0)^T P \boldsymbol{P}_0^{-1}(\widehat{\boldsymbol{x}}_0 - \boldsymbol{x}_0). & k = 0 \\ \frac{1}{2}(\widehat{\boldsymbol{x}}_k - A_{k-1}\widehat{\boldsymbol{x}}_{k-1} - \boldsymbol{u}_k)^T \boldsymbol{Q}_k^{-1}(\widehat{\boldsymbol{x}}_k - A_{k-1}\widehat{\boldsymbol{x}}_{k-1} - \boldsymbol{u}_k). & \text{otherwise} \end{cases}$$
$$J_y(\widehat{\boldsymbol{x}}_k) := \frac{1}{2}(\boldsymbol{y}_k - C_k\widehat{\boldsymbol{x}}_k)^T \boldsymbol{R}_k^{-1}(\boldsymbol{y}_k - C_k\widehat{\boldsymbol{x}}_k).$$

And we can conclude that $J(\widehat{\boldsymbol{x}}_{0:K}) = -\ln[p(\boldsymbol{y}_{0:K}|\widehat{\boldsymbol{x}}_{0:K}) p(\widehat{\boldsymbol{x}}_{0:K}|\boldsymbol{u}_{1:K}, \bar{\boldsymbol{x}}_0)]$. Hence, $\widehat{\boldsymbol{x}}_{0:K} = \arg\min J(\widehat{\boldsymbol{x}}_{0:K})$. For the purpose of simpleness, let

$$H = \begin{pmatrix} 1 & & & & \\ -A_0 & 1 & & & \\ & \ddots & \ddots & & \\ & & -A_{K-1} & 1 & \\ C_0 & & & & \\ & C_1 & & & \\ & & \ddots & & \\ & & & & C_K \end{pmatrix} W = \begin{pmatrix} \boldsymbol{P}_0 & & & & & & \\ & \boldsymbol{Q}_1 & & & & & \\ & & \ddots & & & & \\ & & & \boldsymbol{Q}_K & & & \\ & & & & \boldsymbol{R}_0 & & \\ & & & & & \ddots & \\ & & & & & & \boldsymbol{R}_K \end{pmatrix}$$

and we have

$$J(\boldsymbol{x}_{0:K}) = \frac{1}{2}(\boldsymbol{z} - H\widehat{\boldsymbol{x}}_{0:K})^T W^{-1}(\boldsymbol{z} - H\widehat{\boldsymbol{x}}_{0:K}).$$

where $\boldsymbol{z} = (\boldsymbol{x}_0, \boldsymbol{u}_{1:K}, \boldsymbol{y}_{0:K})^T$. After computing the partial derivative of $\widehat{\boldsymbol{x}}_{0:K}$ and letting it be zero, it concludes that

$$(H^T W^{-1} H)\widehat{\boldsymbol{x}}_{0:K} = H^T W^{-1} \boldsymbol{z}. \tag{24}$$

Moreover, the existence of solution (24) requires that $\mathrm{rank}(H^T H) = \mathrm{rank}(H) = n(K+1)$. In other words, we need

$$\mathrm{rank}\ [C_0^T, A_0^T C_1^T, \cdots, A_0^T \cdots A_{K-1}^T C_K^T] = n.$$

which is the observability theorem of Kalman, see §3.1 of [2].

4.3 Recurrent Kalman filter

Although we can solve the discrete-time linear filtering problem by solving equation (24), computing the inverse of a large dimensional matrix is a heavy work for computer, therefore it is necessary to introduce an effective method in this subsection. Beside, when estimating one state $\widehat{\boldsymbol{x}}_k$ at time t_k, we use the future data. Therefore, this method can not be used on-line. Hence, we refer the following change about matrix H, W and vector $\boldsymbol{z}$.

$$\boldsymbol{H} = \begin{pmatrix} 1 & & & & \\ C_0 & & & & \\ -A_0 & 1 & & & \\ & C_1 & & & \\ & & \ddots & \ddots & \\ & & & -A_{K-1} & 1 \\ & & & & C_K \end{pmatrix} \quad \boldsymbol{W} = \begin{pmatrix} \boldsymbol{P}_0 & & & & & & \\ & \boldsymbol{R}_0 & & & & & \\ & & \boldsymbol{Q}_1 & & & & \\ & & & \boldsymbol{R}_1 & & & \\ & & & & \ddots & & \\ & & & & & \boldsymbol{Q}_K & \\ & & & & & & \boldsymbol{R}_K \end{pmatrix}$$

and $\boldsymbol{z} = (\boldsymbol{x}_0, \boldsymbol{y}_0, \boldsymbol{v}_1, \boldsymbol{y}_1, \cdots, \boldsymbol{v}_K, \boldsymbol{y}_K)^T$. Next, denote

$$\boldsymbol{z}_k := \begin{pmatrix} \widehat{\boldsymbol{x}}_{k-1} \\ \boldsymbol{u}_k \\ \boldsymbol{y}_k \end{pmatrix}, H_k := \begin{pmatrix} 1 & \\ -A_{k-1} & 1 \\ & C_k \end{pmatrix}, W_k := \begin{pmatrix} \widehat{\boldsymbol{P}}_{k-1} & & \\ & \boldsymbol{Q}_k & \\ & & \boldsymbol{R}_k \end{pmatrix}$$

The solution of Kalman filter is

$$(H_k^T W_k^{-1} H_k) \begin{pmatrix} \widehat{\boldsymbol{x}}'_{k-1} \\ \widehat{\boldsymbol{x}}_k \end{pmatrix} = H_k^T W_k^{-1} \boldsymbol{z}_k.$$

Hence we can use the following update process to compute $\widehat{\boldsymbol{x}}_k$.

$$\text{Predictor}: \begin{cases} \check{\boldsymbol{P}}_k = A_{k-1}\widehat{\boldsymbol{P}}_{k-1}A_{k-1}^T + \boldsymbol{Q}_k. \\ \check{\boldsymbol{x}}_k = A_{k-1}\widehat{\boldsymbol{x}}_{k-1} + \boldsymbol{u}_k. \end{cases} \tag{25}$$

$$\text{Kalman gain}: \left\{ \boldsymbol{K}_k = \check{\boldsymbol{P}}_k C_k^T (C_k \check{\boldsymbol{P}}_k C_k^T + \boldsymbol{R}_k)^{-1}. \right. \tag{26}$$

$$\text{Corrector}: \begin{cases} \widehat{\boldsymbol{P}}_k = (\boldsymbol{1} - \boldsymbol{K}_k C_k)\check{\boldsymbol{P}}_k. \\ \widehat{\boldsymbol{x}}_k = \check{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - C_k\check{\boldsymbol{x}}_k). \end{cases} \tag{27}$$

4.4 Extended Kalman filter

Although the technique of Kalman filter only deal with linear filtering problem, it is still useful for some non-linear case by linearizing both the signal and observation process. In this subsection, consider the following stochastic differential equation.

$$\begin{cases} \boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k). \\ \boldsymbol{y}_k = \boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{n}_k). \end{cases} \tag{28}$$

Next, consider the Taylor expansion of $\boldsymbol{f}(\cdot,\cdot,\cdot)$ and $\boldsymbol{g}(\cdot,\cdot)$, it concludes that

$$\boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k) \approx \underbrace{\boldsymbol{f}(\widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k, \boldsymbol{0})}_{\check{\boldsymbol{x}}_k} + \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\Big|_{\widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k, \boldsymbol{0}}}_{\boldsymbol{F}_{k-1}} (\boldsymbol{x}_{k-1} - \widehat{\boldsymbol{x}}_{k-1}) + \underbrace{\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}}\Big|_{\widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k, \boldsymbol{0}} \boldsymbol{w}_k}_{\boldsymbol{w}'_k}.$$

$$\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{n}_k) \approx \underbrace{\boldsymbol{g}(\check{\boldsymbol{x}}_k, \boldsymbol{0})}_{\check{\boldsymbol{y}}_k} + \underbrace{\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\Big|_{\check{\boldsymbol{x}}_k, \boldsymbol{0}}}_{\boldsymbol{G}_k} (\boldsymbol{x}_{k-1} - \check{\boldsymbol{x}}_{k-1}) + \underbrace{\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{n}}\Big|_{\check{\boldsymbol{x}}_k, \boldsymbol{0}} \boldsymbol{n}_k}_{\boldsymbol{n}'_k}.$$

In this way, we can use Kalman filtering theory to deal with non-linear cases and the conditional distribution of

$$p(\boldsymbol{x}_k | \boldsymbol{x}_{k-1}, \boldsymbol{u}_k) \approx \mathcal{N}(\boldsymbol{f}(\widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k, \boldsymbol{0}) + \boldsymbol{F}_{k-1}(\boldsymbol{x}_{k-1} - \widehat{\boldsymbol{x}}_{k-1}), \underbrace{\mathbb{E}[\boldsymbol{w}'_k (\boldsymbol{w}'_k)^T]}_{\boldsymbol{Q}'_k}).$$

$$p(\boldsymbol{y}_k | \boldsymbol{x}_k) \approx \mathcal{N}(\boldsymbol{g}(\check{\boldsymbol{x}}_k, \boldsymbol{0}) + \boldsymbol{G}_k(\boldsymbol{x}_{k-1} - \check{\boldsymbol{x}}_{k-1}), \underbrace{\mathbb{E}[\boldsymbol{n}'_k (\boldsymbol{n}'_k)^T]}_{\boldsymbol{R}'_k}).$$

As a result, the solution of EKF can be obtained from (25), (26) and (27) as the following

$$\text{Predictor} : \begin{cases} \check{\boldsymbol{P}}_k = \boldsymbol{F}_{k-1} \widehat{\boldsymbol{P}}_{k-1} \boldsymbol{F}_{k-1}^T + \boldsymbol{Q}'_k. \\ \check{\boldsymbol{x}}_k = \boldsymbol{f}(\widehat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_k, \boldsymbol{0}). \end{cases} \tag{29}$$

$$\text{Kalman gain} : \{ \ \boldsymbol{K}_k = \check{\boldsymbol{P}}_k \boldsymbol{G}_k^T (\boldsymbol{G}_k \check{\boldsymbol{P}}_k \boldsymbol{G}_k^T + \boldsymbol{R}_k)^{-1}. \tag{30}$$

$$\text{Corrector} : \begin{cases} \widehat{\boldsymbol{P}}_k = (\boldsymbol{1} - \boldsymbol{K}_k \boldsymbol{G}_k) \check{\boldsymbol{P}}_k. \\ \widehat{\boldsymbol{x}}_k = \check{\boldsymbol{x}}_k + \boldsymbol{K}_k (\boldsymbol{y}_k - \boldsymbol{G}_k \boldsymbol{g}(\check{\boldsymbol{x}}_k, \boldsymbol{0})). \end{cases} \tag{31}$$

5 Application of Kalman Filter in Pose Estimation

Based on discussion of linear filtering theory before, we will give an important application of EKF in the field of robotics [2], that is, pose estimation. Since Lie theory plays an important role in representing pose of one vehicle, some algebraic methods have been used in filtering theory. For instance, invariant systems on Lie groups has been used to improve classical EKF in [30]. Moreover, for infinite dimensional non-linear filtering problems, which sets a problem for practical application, some algebraic methods can find the finite dimensional representation of these problems, see [27]. Hence, we will first introduce Lie theory as preliminaries.

5.1 Lie group and Lie algebra

Definition 5.1.1 (Lie group)**.** *A C^∞-manifold G is called a Lie group if G is a group and its two groups operators (multiplication and inverse)*

$$\begin{cases} \circ : G \times G \to G & (a, b) \to a \circ b \\ ()^{-1} : G \to G & a \to a^{-1} \end{cases} \tag{32}$$

are C^∞.

Two important Lie groups special orthogonal group $SO(3)$ and special Euclidean group $SE(3)$ defined as the following.

$$SO(3) := \{C \in \mathbb{R}^{3\times3} |\ CC^T = \boldsymbol{I}, \det(C) = 1\}.$$
$$SE(3) := \left\{ \begin{pmatrix} C & r \\ 0 & 1 \end{pmatrix} \middle|\ C \in SO(3), r \in \mathbb{R}^3 \right\}.$$

Next, we will give the definition of Lie algebra, which plays an important role in solving the filter estimation.

Definition 5.1.2 (Lie algebra)**.** *Let k is a field and L is a vector space over k, if there exists a bilinear operator $[\cdot, \cdot] : L \times L \to L$ such that for $x, y, z \in L$*

1. $[x, y] = -[y, x]$.
2. $[x, [y, z]] + [z, [x, y]] + [y, [z, x]] = 0$.

then L is called a Lie algebra.

Remark 5.1.1. *For a given Lie group G, we denote $\mathfrak{g}$ as the Lie algebra associated to G, which is the tangent space of G at identity. In addition, the Lie bracket $[\cdot, \cdot]$ of $\mathfrak{g}$ is defined by*

$$[x, y] = xy - yx \quad \text{for } \forall x, y \in \mathfrak{g} \tag{33}$$

According to the above remark, we denote $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$ as the Lie algebra of $SO(3)$ and $SE(3)$. Then it concludes

$$\mathfrak{so}(3) := \{C = -C^T |\ C \in \mathbb{R}^{3\times 3}\}.$$
$$\mathfrak{se}(3) := \left\{ \begin{pmatrix} C & r \\ 0 & 0 \end{pmatrix} \middle|\ C \in \mathfrak{so}(3), r \in \mathbb{R}^3 \right\}.$$

Next, we define two operators $(\cdot)^\wedge$ and $(\cdot)^\vee$ as in [2] and [34]. Since the freedom degree of $\mathfrak{so}(3)$ is 3, define $(\cdot)^\wedge : \mathbb{R}^3 \to \mathfrak{so}(3)$ such that for $\phi = (a, b, c)^T \in \mathbb{R}^3$

$$(\phi)^\wedge = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix}$$

Similarly, since the freedom degree of $\mathfrak{se}(3)$ is 6, for $\xi = (\rho^T, \phi^T)^T \in \mathbb{R}^6$, where $\rho, \phi \in \mathbb{R}^3$, $(\cdot)^\wedge : \mathbb{R}^6 \to \mathfrak{se}(3)$ such that

$$(\xi)^\wedge = \begin{pmatrix} (\phi)^\wedge & \rho \\ 0 & 0 \end{pmatrix}$$

It is easy to see $(\cdot)^\wedge$ is bijective, hence let $(\cdot)^\vee$ as its inverse. In addition, define $\exp(\cdot) : \mathfrak{g} \to G$ such that for $(\phi)^\wedge \in \mathfrak{g}$

$$\exp((\phi)^\wedge) := \sum_{n=0}^{\infty} \frac{((\phi)^\wedge)^n}{n!}.$$

If denote $\phi = ||\phi|| a \in \mathbb{R}^3$ such that $a^T a = 1$, it can show that

$$\begin{aligned} \exp((\phi)^\wedge) &= \boldsymbol{I}_{3\times 3} + \left(\sum_{i=0}^{\infty} (-1)^i \frac{||\phi||^{2i+1}}{(2i+1)!} \right) (a)^\wedge - \left(\sum_{j=0}^{\infty} (-1)^i \frac{||\phi||^{2i}}{(2i)!} \right) (a)^\wedge (a)^\wedge \\ &= \cos(||\phi||) \boldsymbol{I}_{3\times 3} + (1 - \cos(||\phi||)) a a^T + \sin(||\phi||)(a)^\wedge. \end{aligned} \tag{34}$$

where we use $(a)^\wedge (a)^\wedge (a)^\wedge = -(a)^\wedge$ and $aa^T = \boldsymbol{I} + (a)^\wedge (a)^\wedge$. Hence, $\exp((\phi)^\wedge) = \exp((||\phi|| + 2k\pi)(a)^\wedge)$ for $k \in \mathbb{Z}$ and $\exp(\cdot)$ is just surjection. In order to define its inverse $\ln(\cdot) : G \to \mathfrak{g}$ such that for $C \in G$

$$\ln(C) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} (C - \boldsymbol{I}_{3\times 3})^n.$$

and let $||(\ln(C))^\vee|| < \pi$, therefore $\ln(C)$ is uniquely determined.

5.2 Rotation, pose and quaternions

In this subsection, we will introduce the background of pose estimation. First, given two right-hand frames $\mathcal{F}_1, \mathcal{F}_2$ in $\mathbb{R}^3$ with the same origin. Next, a 3-dimensional vector v has its coordinates $v_i = (x_i, y_i, z_i)$ in $\mathcal{F}_i$, $i = 1, 2$. Then there exists an orthogonal matrix $C_{2,1} \in \mathbb{R}^{3\times 3}$

such that $v_2 = C_{2,1}v_1$. Then the matrix $C_{2,1}$ is the rotation from $\mathcal{F}_1$ to $\mathcal{F}_2$. Similarly, we can define $C_{1,2}$ in the same way and $C_{1,2} = C_{2,1}^T$. Hence, a rotation is one element in $SO(3)$. Next, suppose $\mathcal{F}_1$ and $\mathcal{F}_2$ do not have the same origin, denote $\vec{r}_{2,1} = \vec{0}_2 - \vec{0}_1$ where $\vec{0}_i$ is the origin of $\mathcal{F}_i$ in $\mathcal{F}_1$, $i = 1, 2$. Therefore, it concludes that $\vec{v}_1 = \vec{r}_{2,1} + \vec{v}_{v,0_2} = \vec{r}_{2,1} + C_{1,2}\vec{v}_2$, or in another way

$$\begin{pmatrix} \vec{v}_1 \\ 1 \end{pmatrix} = \begin{pmatrix} C_{1,2} & \vec{r}_{2,1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \vec{v}_2 \\ 1 \end{pmatrix} \tag{35}$$

Since $C_{1,2} \in SO(3)$, the matrix in (35) belongs to $SE(3)$, which is called a pose. Moreover, we will introduce quaternions as a powerful tool to deal with rotation problem. Given two vector $\vec{P}$ and $\vec{Q}$, which have the same starting point, let $\vec{P}'$ gotten by rotating $\vec{P}$ around $\vec{Q}$ by degree θ in anti-clockwise, then

$$\vec{P}' = \cos(\theta)\vec{P} + (1 - \cos(\theta))(\vec{P} \cdot \vec{A})\vec{A} + \sin(\theta)\vec{P} \times \vec{A} \tag{36}$$

where $\vec{A} = \vec{Q}/||\vec{Q}||$. And (36) is called the Euler Rotation formula in §6.2 of [2]. Next, we will introduce quaternion as a tool to deal with the computation of rotation.

Definition 5.2.1 (Quaternion)**.** *A quaternion $q = (\phi^T, \theta)^T \in \mathbb{R}^4$, where $\phi \in \mathbb{R}^3$. The multiplication operator $\circ$ between two quaternions q_1, q_2 defined by*

$$q_1 \circ q_2 := \begin{pmatrix} \theta_1 \boldsymbol{I}_{3\times3} - (\phi_1)^\wedge & \phi_1 \\ -\phi_1^T & \theta_1 \end{pmatrix} q_2 = q_1 \begin{pmatrix} \theta_2 \boldsymbol{I}_{3\times3} + (\phi_2)^\wedge & \phi_2 \\ -\phi_2^T & \theta_2 \end{pmatrix} \tag{37}$$

In addition, the conjugate of $q = (\phi^T, \theta)^T$ is $\bar{q} = (\phi^T, -\theta)^T$ and the inverse defined by $q^{-1} = \bar{q}/||q||^2$, since identity is $(0, 0, 0, 1)^T$. In this way, all quaternions combined with above multiplication operator is a group.

Back to our rotation model, if we regard $\vec{P}$ as a quaternion $p = (\vec{P}^T, 1)^T$ and denote $q = (\phi^T, s)$ as a unit quaternion, we can show that if $\phi = \sin(\theta/2)\vec{A}$ and $s = \cos(\theta/2)$

$$q \circ p \circ q^{-1} = \begin{pmatrix} \vec{P}' \\ 1 \end{pmatrix}$$

5.3 Jacobian and adjoint operator

In this subsection, let $\xi = (\rho^T, \phi^T)^T \in \mathbb{R}^6$ and

$$\exp((\xi)^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} \begin{pmatrix} (\phi)^\wedge & \rho \\ 0 & 0 \end{pmatrix}^n = \begin{pmatrix} \exp((\phi)^\wedge) & J_l((\phi)^\wedge)\rho \\ 0 & 1 \end{pmatrix}$$

where $J_l((\phi)^\wedge) = \sum_{n=1}^{\infty} (\phi)^n/(n+1)!$ is the left Jacobian of $(\phi)^\wedge$. In addition, if denote $\phi = ||\phi||a$, it concludes that

$$J_l((\phi)^\wedge) = \frac{\sin(||\phi||)}{||\phi||} \boldsymbol{I}_{3\times3} + \left(1 - \frac{\sin(||\phi||)}{||\phi||}\right) aa^T + \left(\frac{1 - \cos(||\phi||)}{||\phi||}\right) (a)^\wedge \tag{38}$$

$$J_l^{-1}((\phi)^\wedge) = \frac{||\phi||}{2} \cot\left(\frac{||\phi||}{2}\right) \boldsymbol{I}_{3\times3} + \left(1 - \frac{||\phi||}{2} \cot\left(\frac{||\phi||}{2}\right)\right) aa^T - \frac{||\phi||}{2} (a)^\wedge \tag{39}$$

The proof of (38) and (39) is similar as (34). If denote $C = \exp((\phi)^\wedge)$, we get $J_l((\phi)^\wedge) = \int_0^1 C^\alpha d\alpha$. Besides, the right Jacobian of $(\phi)^\wedge$ is defined by $J_r((\phi)^\wedge) = J_l(-(\phi)^\wedge)$ and we can show that $J_l(\phi) = CJ_r(\phi)$. Next, we will introduce the adjoint operator of $SE(3)$ and $\mathfrak{se}(3)$. For a given $T \in SE(3)$, denote

$$\mathrm{Ad}(SE(3)) := \left\{ \mathrm{Ad}(T) = \begin{pmatrix} C & (r)^\wedge C \\ 0 & C \end{pmatrix} \middle| \, T = \begin{pmatrix} C & r \\ 0 & 1 \end{pmatrix} \in SE(3) \right\}$$

Let $T_1, T_2 \in \mathrm{Ad}(SE(3))$, we get

$$\mathrm{Ad}(T_1)\mathrm{Ad}(T_2) = \begin{pmatrix} C_1C_2 & (C_1r_2 + r_1)^\wedge C_1C_2 \\ 0 & C_1C_2 \end{pmatrix} = \mathrm{Ad}(T_1T_2).$$
$$\mathrm{Ad}(T)^{-1} = \begin{pmatrix} C^T & (-C^Tr)^\wedge C^T \\ 0 & C^T \end{pmatrix} = \mathrm{Ad}(T^{-1}).$$

Hence, $\mathrm{Ad}(\cdot) : SE(3) \to \mathrm{Ad}(SE(3))$ is a group homomorphism. In fact, we can show the smoothness of $\mathrm{Ad}(\cdot)$, therefore it is actually a Lie group homomorphism, see §3.3 in [10]. Similarly, let $\xi = (\rho^T, \phi^T)^T \in \mathbb{R}^6$ and $(\xi)^\wedge \in \mathfrak{se}(3)$, we can define the operator $\mathrm{ad}(\cdot)$ by

$$\mathrm{ad}((\xi)^\wedge) = \begin{pmatrix} (\phi)^\wedge & (\rho)^\wedge \\ 0 & (\phi)^\wedge \end{pmatrix}$$

As a results, we can define $\mathrm{ad}(\mathfrak{se}(3))$ as above and show $\mathrm{ad}(\cdot) : \mathfrak{se}(3) \to \mathrm{ad}(\mathfrak{se}(3))$ is a Lie algebra homomorphism. In fact, we can show

$$[\mathrm{ad}((\xi_1)^\wedge), \mathrm{ad}((\xi_2)^\wedge)] = \mathrm{ad}(\mathrm{ad}(\xi_1)^\wedge \xi_2).$$

Finally, we have the following commutative diagram.

$$\begin{tikzcd} \mathfrak{se}(3) \arrow[r, "\exp"] \arrow[d, "\mathrm{ad}"] & SE(3) \arrow[d, "\mathrm{Ad}"] \\ \mathrm{ad}(\mathfrak{se}(3)) \arrow[r, "\exp"] & \mathrm{Ad}(SE(3)) \end{tikzcd}$$

We need to show $\mathrm{Ad}(\exp(\cdot)) \equiv \exp(\mathrm{ad}(\cdot))$. For $\xi \in \mathfrak{se}(3)$, we have

$$\mathrm{Ad}(\exp((\xi)^\wedge)) = \mathrm{Ad}\left(\exp\begin{pmatrix} (\phi)^\wedge & \rho \\ 0 & 0 \end{pmatrix}\right) = \begin{pmatrix} C & (r)^\wedge C \\ 0 & C \end{pmatrix}$$

where $C = \exp((\phi)^\wedge)$ and $r = J_l((\phi)^\wedge)\rho$. On the other hand, we compute

$$\exp(\mathrm{ad}((\xi)^\wedge)) = \sum_{n=0}^{\infty} \frac{1}{n!} \begin{pmatrix} (\phi)^\wedge & (\rho)^\wedge \\ 0 & (\phi)^\wedge \end{pmatrix}^n = \begin{pmatrix} C & K \\ 0 & C \end{pmatrix}$$

where

$$K = \sum_{m,n=0}^{\infty} \frac{1}{(n+m+1)!}((\phi)^\wedge)^n(\rho)^\wedge((\phi)^\wedge)^m.$$

In addition, we have

$$
\begin{aligned}
(r)^\wedge C = (J_l((\phi)^\wedge)\rho)^\wedge C &= \int_0^1 (C^\alpha \rho)^\wedge C d\alpha = \int_0^1 C^\alpha (\rho)^\wedge C^{1-\alpha} d\alpha \\
&= \int_0^1 \left(\sum_{n=0}^{\infty} \frac{(\alpha(\phi)^\wedge)^n}{n!} \right) (\rho)^\wedge \left(\sum_{m=0}^{\infty} \frac{((1-\alpha)(\phi)^\wedge)^m}{m!} \right) d\alpha \\
&= \sum_{m,n=0}^{\infty} \frac{1}{n!m!} \left(\int_0^1 \alpha^n (1-\alpha)^m d\alpha \right) ((\phi)^\wedge)^n (\rho)^\wedge ((\phi)^\wedge)^m = K
\end{aligned}
$$

Here, we use $(C\rho)^\wedge = C(\rho)^\wedge C^T$ and $\int_0^1 \alpha^n (1-\alpha)^m d\alpha = n!m!/(n+m+1)!$.

Theorem 5.3.1 (Baker-Campbell-Hausdorff formula)**.** *For given $a, b \in \mathfrak{g}$, it has*

$$
\ln(\exp(a)\exp(b)) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} \sum_{\substack{r_i + s_i > 0 \\ 1 \le i \le n}} \frac{(\sum_{i=1}^n r_i + s_i)^{-1}}{\prod_{i=1}^n r_i! s_i!} [a^{r_1} b^{t_1} \cdots a^{r_n} b^{t_n}] \tag{40}
$$

where

$$
\begin{aligned}
&[a^{r_1} b^{t_1} \cdots a^{r_n} b^{t_n}] \\
&= \underbrace{[a, \cdots [a,}_{r_1} \underbrace{[b, \cdots [b,}_{s_1} \underbrace{[a, \cdots [a,}_{r_n} \underbrace{[b, \cdots, [b, b] \cdots]}_{s_n} \cdots] \cdots] \cdots] \cdots] \cdots] \cdots]
\end{aligned} \tag{41}
$$

The proof of Theorem 5.3.1, see §3.2 of [10]. In general, if we can omit a or b when its order larger than 2, it has the following approximation

$$
\ln(\exp(a)\exp(b)) \approx b + \sum_{n=0}^{\infty} \frac{B_n}{n!} \mathbf{ad}_b^n(a).
$$

$$
\ln(\exp(a)\exp(b)) \approx a + \sum_{n=0}^{\infty} (-1)^n \frac{B_n}{n!} \mathbf{ad}_a^n(b).
$$

where the operator $\mathbf{ad}_b(a) := [b, a]$. In addition, we can show that for given $(\phi_1)^\wedge, (\phi_2)^\wedge \in SO(3)$, it concludes that

$$
\ln(\exp((\phi_1)^\wedge)\exp((\phi_2)^\wedge))^\vee \approx \begin{cases} J_l(\phi_2)^{-1}\phi_1 + \phi_2 & \text{if } \phi_1 \text{ samll} \\ \phi_1 + J_r(\phi_1)^{-1}\phi_2 & \text{if } \phi_2 \text{ samll} \end{cases} \tag{42}
$$

Now, we can compute the derivative of $C = \exp((\phi)^\wedge) \in SO(3)$, suppose $\widetilde{C} = \exp((\phi + \delta\phi)^\wedge) \in SO(3)$, where $\delta\phi$ is a small perturbed rotation, due to (42), we conclude the following approximation

$$
\ln(C^T \widetilde{C})^\vee = \ln(C^T \exp((\phi + \delta\phi)^\wedge))^\vee \approx \ln(C^T C \exp((J_r(\phi)\delta\phi)^\wedge))^\vee = J_r(\phi)\delta\phi.
$$

Similarly, we can also obtain

$$
\ln(\widetilde{C} C^T)^\vee \approx J_l(\phi)\delta\phi.
$$

If we denote $\delta C_r := C^T\widetilde{C}$ and $\delta C_l := \widetilde{C}C^T$, let $\delta\phi$ tend to zero, it concludes

$$\begin{cases} dC_r = |\det(J_r(\phi))|d\phi. \\ dC_l = |\det(J_l(\phi))|d\phi. \end{cases}$$

On the other hand, $\det(J_l(\phi)) = \det(CJ_r(\phi)) = \det(J_r(\phi))$, hence it can simplify as

$$dC = |\det(J_\phi)|d\phi. \tag{43}$$

5.4 Random variable on matrix Lie group

For a Gaussian random variable C in $SO(3)$, it can be defined by

$$C = \bar{C}\exp((\boldsymbol{\epsilon})^\wedge).$$

where $\bar{C} \in SO(3)$ is noise-free and $\boldsymbol{\epsilon}$ is a three dimensional Gaussian random variable with covariance matrix $\boldsymbol{\Sigma}$. It concludes that

$$\begin{aligned} 1 &= \int_{\mathbb{R}^3} \underbrace{\frac{1}{\sqrt{(2\pi)^3\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\epsilon}\right)}_{p(\boldsymbol{\epsilon})} d\boldsymbol{\epsilon} \\ &= \int_{\mathbb{R}^3} \underbrace{\frac{1}{\sqrt{(2\pi)^3\det(\boldsymbol{\Sigma})}}\exp\left(-\frac{1}{2}(\ln(\bar{C}^TC)^\vee)^T\boldsymbol{\Sigma}^{-1}\ln(\bar{C}^TC)^\vee\right)\frac{1}{\det(J_{\boldsymbol{\epsilon}})}}_{p(C)} dC \end{aligned}$$

The last equation is derived by (43), moreover, we can show the Gaussian rand variable C has mean of $\bar{C}$ and covariance of Σ. In fact, let M be the mean of C and we have

$$\int \ln(M^TC)^\vee p(C)dC \equiv 0.$$

The let $M \equiv \bar{C}$, it concludes that

$$\int \ln(\bar{C}^TC)^\vee p(C)dC = \int_{\mathbb{R}^3} \boldsymbol{\epsilon}p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} = 0.$$

By similar method, we can show that C has covariance of Σ. Moreover, for a Gaussian random variable $T \in SE(3)$, it is defined by

$$T = \bar{T}\exp((\boldsymbol{\xi})^\wedge).$$

where $\bar{T} \in SE(3)$ is a noise-free pose and $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_T)$ is a six dimensional Gaussian random variable, and we can also show that $T \sim \mathcal{N}(\bar{T}, \boldsymbol{\Sigma}_T)$ with the similar deduction as $SO(3)$.

5.5 Pose estimation

Let $\mathcal{F}_w$ be a non-moving frame and $\mathcal{F}_v$ be the moving frame of vehicle in $\mathbb{R}^3$. Suppose we have n observation of the moving vehicle at time $\{t_i\}_{i=0}^n$, and denote $\mathcal{F}_v^i$ to be the position of frame $\mathcal{F}_v$. In addition, let $p_j \in \mathbb{R}^3$ be a fixed point and its coordinate $\vec{r}_j^w$ in $\mathcal{F}_w$ is known. Besides, its coordinates $\vec{r}_{i,j}^v$ can be obtained from the vehicle, which is called the measurement and influenced by some noise. Our main goal is to estimate the distance between our vehicle and non-moving frame, that is $\vec{r}_j := \vec{r}_j^w - \vec{r}_{i,j}^v$.

Based on pose transformation formula in §5.2, it concludes that

$$\begin{pmatrix} \vec{r}_{i,j}^v \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} C_{i,w} & -C_{i,w}\vec{r}_j \\ 0 & 1 \end{pmatrix}}_{T_i} \underbrace{\begin{pmatrix} \vec{r}_j^w \\ 1 \end{pmatrix}}_{\vec{p}_j}$$

where T_i is the pose transition matrix between frame $\mathcal{F}_w$ and $\mathcal{F}_v^i$, for $i = 0, \cdots, n$. In addition, suppose the measurements are also perturbed by some noises, hence the observation $\boldsymbol{y}_{i,j}$ of $\vec{r}_{i,j}^v$ defined by

$$\boldsymbol{y}_{i,j} = D^T T_i \vec{p}_j + \boldsymbol{n}_{i,j}. \tag{44}$$

where $\boldsymbol{n}_{i,j} \sim \mathcal{N}(0, \boldsymbol{R}_{i,j})$ and

$$D^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Next, the following input about vehicle motion is prior known.

(i) The initial pose T_0, translation velocity $\vec{v}_i$ and angular velocity $\vec{\omega}_i$. Denote $\boldsymbol{s}_i = (\vec{v}_i, \vec{\omega}_i)^T \in \mathbb{R}^6$. Hence the prior known inputs are $\boldsymbol{v} = \{T_0, \boldsymbol{s}_0, \cdots, \boldsymbol{s}_n\}$.

(ii) The measurements $\vec{r}_{i,j}^v$ of point $\{\vec{P}_j\}_{j=1}^m$ are known. Denote by $\boldsymbol{y} = \{\vec{r}_{i,j}^v\}_{i=0,j=1}^{n,m}$.

The kinematics differential equation for this model can be derived by the following discussion. For $C(t) = \exp((\phi(t))^\wedge) \in SO(3)$ and $t \geq 0$, the derivative can be computed by

$$\dot{C}(t) = \frac{d}{dt}\exp((\phi(t))^\wedge) = \int_0^1 \exp(\alpha(\phi)^\wedge)(\dot{\phi})^\wedge \exp((1-\alpha)(\phi)^\wedge) d\alpha.$$

it concludes that

$$\dot{C}C^T = \int_0^1 (C^\alpha \dot{\phi})^\wedge d\alpha = (J_l(\dot{\phi}))^\wedge.$$

Hence, we get $\dot{C}(t) = (J_l(\dot{\phi}(t)))^\wedge C(t)$ and denote $w_\phi = J_l(\dot{\phi})$. Then take the $\vee$ operator on both sides, we get

$$\dot{\phi} = J_l(\phi)^{-1} w_\phi. \tag{45}$$

Similarly, for $T(t) = \exp((\xi(t))^\wedge)$ and $t \geq 0$, we can conclude that $\dot{T}(t) = (J_l(\dot{\xi}(t)))^\wedge T(t)$. Moreover, consider $C(t) = \exp((\delta\phi(t))^\wedge)\bar{C} \in SO(3)$, where $(\delta\phi(t))^\wedge \in \mathfrak{se}(3)$ is a small perturbed rotation and $\bar{C} = \exp((\phi)^\wedge)$ is relative large constant rotation, then we have

$$C(t) \approx (\boldsymbol{I}_{3\times 3} + (\delta\phi)^\wedge)\bar{C}.$$

Next, let $\phi' = \ln(C(t))^\vee$ and we get according to (42)

$$\phi' = J_l(\phi)^{-1}\delta\phi + \phi.$$

In addition, we compute

$$J_l(\phi') = \int_0^1 \exp((\delta\phi)^\wedge C)^\alpha d\alpha \approx \underbrace{\int_0^1 C^\alpha d\alpha}_{J_l(\phi)} \underbrace{- \int_0^1 \alpha \left(J_l(\alpha\phi)J_l(\phi)^{-1}\delta\phi\right)^\wedge C^\alpha d\alpha}_{\delta J}.$$

Besides, we have

$$\begin{aligned}
\dot{\phi}' &= \dot{\phi} + \left(\frac{d}{dt}J_l(\phi)^{-1}\right)\delta\phi + J_l(\phi)^{-1}\frac{d}{dt}\delta\phi \\
&= \dot{\phi} - J_l(\phi)^{-1}\left(\frac{d}{dt}J_l(\phi)\right)J_l(\phi)^{-1}\delta\phi + J_l(\phi)^{-1}\frac{d}{dt}\delta\phi \\
&\approx (J_l(\phi) + \delta J)^{-1}(w_\phi + \delta w).
\end{aligned}$$

The last approximation is derived from (45), and we conclude that

$$\begin{aligned}
\frac{d}{dt}\delta\phi &= \left(\frac{d}{dt}J_l(\phi)\right)J_l(\phi)^{-1}\delta\phi - \delta J\dot{\phi}' + \delta w \\
&= \left(\frac{\partial w_\phi}{\partial\phi} + (w_\phi)^\wedge J_l(\phi)\right)J_l(\phi)^{-1}\delta\phi - \delta J\dot{\phi}' + \delta w \\
&= (w_\phi)^\wedge\delta\phi + \delta w + \frac{\partial w_\phi}{\partial\phi}J_l(\phi)^{-1}\delta\phi - \delta J\dot{\phi}'.
\end{aligned}$$

The second equation of above due to [16], finally compute

$$\begin{aligned}
\frac{\partial w_\phi}{\partial\phi}J_l(\phi)^{-1}\delta\phi &= \left(\int_0^1 \frac{\partial}{\partial\phi}(C^\alpha\dot{\phi})d\alpha\right)J_l(\phi)^{-1}\delta\phi \\
&= \int_0^1 \alpha(C^\alpha\dot{\phi})^\wedge J_l(\alpha\phi)J_l(\phi)^{-1}\delta\phi d\alpha \\
&= -\int_0^1 \alpha\left(J_l(\alpha\phi)J_l(\phi)^{-1}\delta\phi\right)^\wedge C^\alpha\dot{\phi}d\alpha \\
&= \delta J\dot{\phi}.
\end{aligned}$$

Now, we omit terms which contains more than two orders of small perturbed rotation and conclude that

$$\frac{d}{dt}\delta\phi = (w_\phi)^\wedge\delta\phi + \delta w.$$

Similarly, for $T = \exp((\delta\xi)^\wedge)\bar{T} \in SE(3)$, we can repeat the above inducement and conclude the kinematics differential equations as the following.

$$\begin{cases} \dot{\bar{T}} = (J_l(\dot{\xi}))^\wedge \bar{T}. & \text{nominal kinematics} \\ \frac{d}{dt}\delta\xi = \mathrm{ad}((\xi)^\wedge)\delta\xi + \delta w. & \text{perturbation kinematics} \end{cases} \tag{46}$$

Combined with our motion model, we transform (46) into discrete equations.

$$\begin{cases} \bar{T}_k = \exp(\Delta t_k \boldsymbol{s}_k^\wedge)\bar{T}_{k-1}. & \text{nominal kinematics} \\ \delta\xi_k = \exp(\Delta t_k \mathrm{ad}(\boldsymbol{s}_k))\delta\xi_{k-1} + \boldsymbol{w}_k. & \text{perturbation kinematics} \end{cases} \tag{47}$$

where $\boldsymbol{w}_k \sim \mathcal{N}(0, \boldsymbol{Q}_k)$ for $k = 0, \cdots, n$. Next, the measurements (44) can be linearized as the following.

$$\begin{aligned} \boldsymbol{y}_{i,j} := \bar{\boldsymbol{y}}_{i,j} + \delta\boldsymbol{y}_{i,j} &= D^T(\exp((\delta\xi_i)^\wedge)T_i)\vec{p}_j + \boldsymbol{n}_{i,j} \\ &\approx D^T(\boldsymbol{I}_{4\times4} + (\delta\xi_i)^\wedge)T_i\vec{p}_j + \boldsymbol{n}_{i,j} \\ &= \underbrace{D^T T_i \vec{p}_j}_{\bar{\boldsymbol{y}}_{i,j}} + \underbrace{D^T(T_i\vec{p}_j)^\odot \delta\xi_i + \boldsymbol{n}_{i,j}}_{\delta\boldsymbol{y}_{i,j}}. \end{aligned}$$

Here, for a given $\xi = (\phi, \theta)^T \in \mathbb{R}^4$,

$$(\xi)^\odot = \begin{pmatrix} \theta\boldsymbol{I}_{3\times3} & -(\phi)^\wedge \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{4\times6}$$

and $(\xi)^\wedge\vec{p} = (\vec{p})^\odot\xi$ for $\forall\vec{p} \in \mathbb{R}^4$. Finally, we denote

$$\boldsymbol{F}_{k-1} = \exp(\Delta t_k \mathrm{ad}(\boldsymbol{s}_k)), \boldsymbol{G}_{j,k} = D^T(T_i\vec{p}_j)^\odot, \boldsymbol{G}_k = \begin{pmatrix} \boldsymbol{G}_{1,k} \\ \vdots \\ \boldsymbol{G}_{m,k} \end{pmatrix}, \boldsymbol{y}_k = \begin{pmatrix} \boldsymbol{y}_{1,k} \\ \vdots \\ \boldsymbol{y}_{m,k} \end{pmatrix}$$

and $\boldsymbol{R}_k = \mathrm{diag}(\boldsymbol{R}_{1,k}, \cdots, \boldsymbol{R}_{m,k})$. Then the solution of pose estimation can be obtained from EKF (29), (30) and (31).

5.6 Numerical results

In this subsection, we choose stand $\mathbb{R}^3$ coordinate system as $\mathcal{F}_w$ and time sequence $\{t_i = i \in \mathbb{Z}\}_{i=0}^{100}$ with $\Delta t_i = t_i - t_{i-1} = 0.01$ for all $i = 1, \cdots, 100$, then let the $\{p_j \in \mathbb{R}^3\}_{j=1}^{20}$ generated by $\mathcal{N}(0, \mathrm{diag}(1,1,1))$ be 20 fixed points in $\mathbb{R}^3$. From last subsection, our filtering problem is

$$\begin{cases} \bar{T}_k = \exp(\Delta t_k(\boldsymbol{s}_k)^\wedge)\bar{T}_{k-1}. \\ \delta\xi_k = \exp(\Delta t_k \mathrm{ad}(\boldsymbol{s}_k))\delta\xi_{k-1} + \boldsymbol{w}_k. \\ T_k = \exp((\delta\xi_k)^\wedge)\bar{T}_k. \\ \boldsymbol{y}_k = D^T T_k \boldsymbol{p} + \boldsymbol{n}_k. \end{cases} \tag{48}$$

where $\boldsymbol{p}$ is a 4×20 dimensional matrix and the noise of measurement $\boldsymbol{n}_k$ denoted by

$$\boldsymbol{p} = \begin{pmatrix} \vec{p}_1 & \cdots & \vec{p}_2 \\ 1 & \cdots & 1 \end{pmatrix}, \boldsymbol{n}_k = \begin{pmatrix} \boldsymbol{n}_{1,k} \\ \vdots \\ \boldsymbol{n}_{m,k} \end{pmatrix}$$

and for $k = 1, \cdots, 100$, the translation and angular velocity is denoted by

$$\vec{v}_k = \begin{pmatrix} \frac{1}{1+k} \\ \frac{1}{1+2k} \\ \frac{1}{1+3k} \end{pmatrix}, \vec{\omega}_i = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{k^2/1000+k+1} \end{pmatrix}$$

and $\boldsymbol{s}_k = (\vec{v}_k, \vec{\omega}_k)^T$. Moreover, the Gaussian noise $\boldsymbol{w}_k$ with covariance matrix

$$\begin{pmatrix} 0.1^2 \boldsymbol{I}_{3\times 3} & \\ & 0.05^2 \boldsymbol{I}_{3\times 3} \end{pmatrix}$$

and $\boldsymbol{n}_{i,j} \sim \mathcal{N}(\boldsymbol{0}, 0.15^2 \boldsymbol{I}_{3\times 3})$. The initial pose is defined by

$$T_0 = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 1 & 0 & 0.5 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Finally, we simulate the pose $\{T_k\}_{k=1}^{100}$ and measurement $\{\boldsymbol{y}_k\}_{k=0}^{100}$ by (48) and then obtain the estimation results $\{\widehat{T}_k\}_{k=1}^{100}$ EKF. In order to show accuracy of our estimation results, we compute $||\widehat{T}_k - T_k||_2$ for $k = 0, \cdots, 100$.

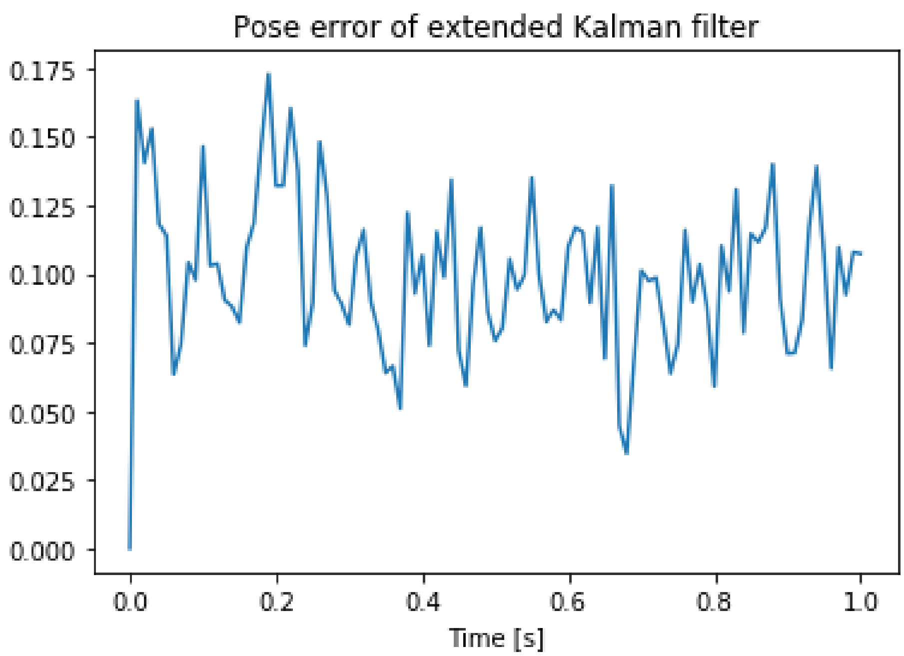


Figure 1: The error of pose $||\widehat{T}_k - T_k||_2$ for all $k = 0, \cdots, 100$

6 Application of Non-linear Filter in Finance.

In this section, we will focus on the application of non-linear filters, based on Black-Scholes Model, we consider the asset price $S = (S_t)_{t\geq 0}$ is a stochastic process governed by the following differential equation [33].

$$dS_t = \mu_t S_t dt + \sigma_t S_t dW_t.$$

where W_t is Wiener process and σ_t is called the 'volatility', which usually can not be observed, since σ_t depends on variable of factors such as the interest rate, national policy and even investors' psychology. As a result, it is meaningful to apply filter theory to estimate σ_t based on S_t, since the price of asset can easily obtained from the markets. In addition, for the purpose of simpleness, we regard the drift μ_t as a constant and assume σ_t satisfy ARV dynamics [9] (49). Moreover, since this financial model is a non-linear filtering problem (50), we will apply Monte-Carlo simulation (§6.3) and particle filters [1](§6.4). Besides, the inference of parameters (μ for example) is another critical task, and we will use EM (Expectation-Maximum) algorithm to solve this problem in (§6.2).

6.1 Mathematics model

First, we denote a stochastic process $y = (y_t)_{t\geq 0}$ such that $y_t := \log(S_t/S_0)$, then it concludes

$$dy_t = (\mu - \frac{1}{2}\sigma_t^2)dt + \sigma_t d\widetilde{W}_t.$$

Next, we choose the stochastic volatility model proposed by ARV dynamics, i.e. define a stochastic process $x = (x_t)_{t\geq 0}$ such that $x_t = \log(\sigma_t)$, which is governed by the following differential equations.

$$dx_t = (a + bx_t)dt + cdW_t. \tag{49}$$

where W_t is another Wiener process which is independent of $\widetilde{W}_t$. To summary, we obtain the following two stochastic processes $x = (x_t)_{t\geq 0}$ and $y = (y_t)_{t\geq 0}$ by

$$\begin{cases} dx_t = (a + bx_t)dt + cdW_t. \\ dy_t = (\mu - \frac{1}{2}e^{2x_t})dt + e^{x_t}d\widetilde{W}_t. \end{cases} \tag{50}$$

Based on our assumption, we know x is the signal process and y is our observation. Next, define the observed filtration $\mathcal{F}_t^y$ by

$$\mathcal{F}_t^y := \sigma\{y_s : s < t\}.$$

As we have shown in Proposition 3.1.2, we will construct a new 'ideal' measurable $\mathbb{Q}$ satisfying

$$\begin{aligned} \frac{d\mathbb{Q}}{d\mathbb{P}}\Big|_{\mathcal{F}_t^y} = \exp\Big[&- \int_0^t \left(\mu e^{-x_s} - \frac{1}{2}e^{x_s}\right) d\widetilde{W}_s - \frac{1}{2}\int_0^t \left(\mu e^{-x_s} - \frac{1}{2}e^{x_s}\right)^2 ds \\ &- \int_0^t \left(c^{-1}a + c^{-1}bx_t\right) dW_s - \frac{1}{2}\int_0^t \left(c^{-1}a + c^{-1}bx_t\right)^2 ds\Big]. \end{aligned} \tag{51}$$

By Theorem 3.1.1, we know that $(x_s, y_s)_{s\in[0,t]}$ is a $\mathcal{F}_t$-Wiener process under measure $\mathbb{Q}$.

6.2 Parameters inference

Our next step is to estimate parameters (a, b, c, μ), and we will discrete equation (50). In other words, suppose our model is finite time horizon, i.e. we only observe x, y in time interval $[0, T]$. Then we divide it into K pieces and $x_k = x_{k\Delta t}$ for $k = 1, \cdots, K$, where $\Delta t = T/K$. In addition, we will use EM (Expectation-Maximum) algorithm, which is a special case of MLE (Maximum-Likelihood-Estimation) method. From the last subsection, we have the following equations.

$$\begin{cases} x_k = (a + bx_{k-1})\Delta t + c\sqrt{\Delta t} w_k. \\ y_k = (\mu - \frac{1}{2}e^{2x_k})\Delta t + e^{x_k}\sqrt{\Delta t}\widetilde{w}_k. \end{cases} \tag{52}$$

where w_k and $\widetilde{w}_k$ are independent $\mathcal{N}(0, 1)$ for $k = 1, \cdots, K$. Moreover, for $t = 1, \cdots, M$, define the observation filtration $\mathcal{F}^y$ by

$$\mathcal{F}_t^y := \sigma\{y_k : k \leq t\}.$$

Due to equation (51), we define Radon-Nikodym function Λ_t by

$$\Lambda_t = \prod_{k=0}^{t} \lambda_k \tag{53}$$

where for $k = 1, \cdots, t$

$$\lambda_k := \frac{\phi(c^{-1}(x_k - (a + bx_{k-1})\Delta t))}{c\phi(x_k)} \frac{\phi(e^{-x_k}(y_k - (\mu - e^{2x_k}/2)\Delta t))}{e^{x_k}\phi(y_k)}.$$
$$\lambda_0 := \frac{\phi(e^{-x_0}(y_0 - (\mu_0 - e^{2x_0}/2)\Delta t))}{e^{x_0}\phi(y_0)}.$$

where $\phi(x)$ is the density function of $\mathcal{N}(0, 1)$. In addition, denote $q_t(x)$ to be the density function under measure $\mathbb{Q}_{\mathcal{F}_t^y}$, that is, $\mathbb{E}_{\mathbb{Q}}[f(x_t)|\mathcal{F}_t^y] = \int_{\mathbb{R}} f(x)q_t(x)dx$. Combined with conclusion in §3, it concludes that

Proposition 6.2.1. *The density $q_t(x)$ related to the measure $\mathbb{Q}_{\mathcal{F}_t^y}$ satisfies the following recurrence equation*

$$q_t(x) = \Delta(y_t, x) \int_{-\infty}^{\infty} \phi(c^{-1}(x - (a + bz)\Delta t))q_{t-1}(z)dz. \tag{54}$$

$q_0(t)$ is the density of x_0 and

$$\Delta(y_t, x) = \frac{\phi(e^{-x}(y_t - (\mu - e^{2x}/2)\Delta t))}{e^x c\phi(y_t)}. \tag{55}$$

As a conclusion of above proposition, we conclude the best estimation of $f(x_t)$ on $\mathcal{F}_t$ is

$$\mathbb{E}[f(x_t)|\mathcal{F}_t^y] = \frac{\int_{-\infty}^{\infty} f(z)q_t(z)dz}{\int_{-\infty}^{\infty} q_t(z)dz}.$$

The estimation parameters are (a, b, c, μ), and we will use iteration method to get the optimal estimation. For instance, we have obtain a set of parameters (a, b, c, μ), the following procedure is used to update a. Suppose we obtain a' after one iteration, consider the Radon–Nikodym derivatives

$$\Lambda_t^a = \prod_{k=0}^{t} \frac{\phi(c^{-1}(x_k - (a + bx_{k-1})\Delta t))}{c\phi(x_k)} \frac{\phi(e^{-x_k}(y_k - (\mu - e^{2x_k}/2)\Delta t))}{e^{x_k}\phi(y_k)}.$$

$$\Lambda_t^{a'} = \prod_{k=0}^{t} \frac{\phi(c^{-1}(x_k - (a' + bx_{k-1})\Delta t))}{c\phi(x_k)} \frac{\phi(e^{-x_k}(y_k - (\mu - e^{2x_k}/2)\Delta t))}{e^{x_k}\phi(y_k)}.$$

Hence, we conclude

$$\log\left(\frac{d\mathbb{P}^{a'}}{d\mathbb{P}^{a}}\Big|_{\mathcal{F}_t^y}\right) = -\frac{1}{2}\sum_{k=1}^{t}[(c^{-1}(x_k - (a' + bx_{k-1})\Delta t))^2 - R(a)].$$

where $R(a) = \sum_{k=1}^{t}(c^{-1}(x_k - (a + bx_{k-1})\Delta t))^2$.So the expectation step will be

$$\mathbb{E}\left[\log\left(\frac{d\mathbb{P}^{a}}{d\mathbb{P}^{a'}}\Big|_{\mathcal{F}_t}\right)\Big|\mathcal{F}_t^y\right] = \mathbb{E}\left[-\frac{1}{2}\sum_{k=1}^{t}[(c^{-1}(x_k - (a' + bx_{k-1})\Delta t))^2 - R(a)]\Big|\mathcal{F}_t^y\right].$$

The maximum step will be attended when

$$\frac{d}{da'}\left(\mathbb{E}\left[\log\left(\frac{d\mathbb{P}^{a}}{d\mathbb{P}^{a'}}\Big|_{\mathcal{F}_t}\right)\Big|\mathcal{F}_t^y\right]\right) = \mathbb{E}\left[c^{-2}\Delta t\sum_{k=1}^{t}(x_k - (a' + bx_{k-1})\Delta t)\Big|\mathcal{F}_t^y\right] = 0.$$

Therefore, the best estimation of a' will be

$$a' := \frac{1}{t\Delta t}\left(\mathbb{E}\left[\sum_{k=1}^{t} x_k\Big|\mathcal{F}_t^y\right] - b\Delta t\mathbb{E}\left[\sum_{k=1}^{t} x_{k-1}\Big|\mathcal{F}_t^y\right]\right) \tag{56}$$

Similarly, we can obtain the updating procedure of b, c, μ by

$$b' := \frac{\mathbb{E}\left[\sum_{k=1}^{t} x_k x_{k-1}\Big|\mathcal{F}_t^y\right] - a\Delta t\mathbb{E}\left[\sum_{k=1}^{t} x_{k-1}\Big|\mathcal{F}_t^y\right]}{\Delta t\mathbb{E}\left[\sum_{k=1}^{t} x_{k-1}^2\Big|\mathcal{F}_t^y\right]} \tag{57}$$

$$c' := \left\{\frac{1}{t}\mathbb{E}\left[\sum_{k=1}^{t}(x_k - (a + bx_{k-1})\Delta t)^2\Big|\mathcal{F}_t^y\right]\right\}^{\frac{1}{2}} \tag{58}$$

$$\mu' := \frac{\mathbb{E}\left[\sum_{k=1}^{t} e^{-2x_k}y_k\Big|\mathcal{F}_t^y\right] + \mathbb{E}\left[t\Delta t/2\Big|\mathcal{F}_t^y\right]}{\Delta t\mathbb{E}\left[\sum_{k=1}^{t} e^{-2x_k}\Big|\mathcal{F}_t^y\right]} \tag{59}$$

Before giving estimation, the following theorem is necessary.

Theorem 6.2.1. *Suppose $H(\cdot), F(\cdot), G(\cdot)$ are integrable functions such that*

$$S_t := \sum_{k=1}^{t} H(y_k)F(x_k)G(x_{k-1}). \tag{60}$$

Next denote the density function $L_t^{H,F,G}(x)$ such that

$$\mathbb{E}_{\mathbb{Q}}[\Lambda_t S_t f(x_t)|\mathcal{F}_t] = \int_{-\infty}^{\infty} f(x)L_t^{H,F,G}(x)dx. \tag{61}$$

where

$$\begin{aligned} L_t^{H,F,G}(x) = \Delta(y_t, x)\Bigg[&\int_{-\infty}^{\infty} \phi(c^{-1}(x-(a+bz)\Delta t))L_{t-1}^{H,F,G}(z)dz \\ &+ H(y_t)F(x)\int_{-\infty}^{\infty} \phi(c^{-1}(x-(a+bz)\Delta t))G(z)q_{t-1}(z)dz\Bigg]. \end{aligned} \tag{62}$$

As a result, with the help of above theorem, the updating process can be re-written as the following. Suppose (a_k, b_k, c_k, μ_k) it the estimation result of k-th iteration, the updating of $(a_{k+1}, b_{k+1}, c_{k+1}, \mu_{k+1})$ will be

$$a_{k+1} = \frac{\int_{-\infty}^{\infty}\left[L_t^{1,x,1}(z) - b_k L_t^{1,1,x}(z)\right]dz}{t\int_{-\infty}^{\infty} q_t(z)dz}$$

$$b_{k+1} = \frac{\int_{-\infty}^{\infty}\left[L_t^{1,x,x}(z) - a_k L_t^{1,1,x}(z)\right]dz}{\int_{-\infty}^{\infty} L_t^{1,1,x^2}(z)dz}$$

$$c_{k+1} = \left\{\frac{\int_{-\infty}^{\infty}\left[L_t^{1,x^2,1}(z) + a_k^2 q_t(z) + b_k^2 L_t^{1,1,x^2}(z) + 2a_k b_k L_t^{1,1,x}(z) - 2a_k L_t^{1,x,1}(z) - 2b_k L_t^{1,x,x}(z)\right]dz}{t\int_{-\infty}^{\infty} q_t(z)dz}\right\}^{\frac{1}{2}}$$

$$\mu_{k+1} = \frac{\int_{-\infty}^{\infty}\left[L_t^{1,e^{-2x},1}(z) + t q_t(z)/2\right]dz}{\int_{-\infty}^{\infty} L_t^{1,e^{-2x},1}(z)dz}$$

6.3 Monte-Carlo simulation for EM algorithm

Although we have given the analytical updating process of parameters (a, b, c, μ) in last subsection, it is nearly impossible to realize this method by computers, since calculating integration on infinite interval is a trouble. Hence, we will apply Monte-Carlo filter [20] method to numerically infer parameters (a, b, c, μ). Consider model (28) without control, suppose the density $d_{\boldsymbol{w}}(\cdot), d_{\boldsymbol{n}}(\cdot)$ of noises $\boldsymbol{w}_k, \boldsymbol{n}_k$ are known for $k = 0, \cdots K$ and functions $\boldsymbol{f}(\cdot), \boldsymbol{g}(\cdot)$ contain unknown parameters. Besides, if $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ are given, $\boldsymbol{n}_k$ can be uniquely decided by a function of $(\boldsymbol{x}_k, \boldsymbol{y}_k)$, denote by $\boldsymbol{n}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{y}_k)$. Moreover, assume that $\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{y})$

exists for all $(\boldsymbol{x}, \boldsymbol{y})$, we approximate our filtering problem by the following three parts.

$$\begin{aligned}
&\{p_k^{(1)}, \cdots, p_k^{(m)}\} \sim p(\boldsymbol{x}_k|\mathcal{F}_{k-1}^{\boldsymbol{y}}) && \text{predictor}\\
&\{f_k^{(1)}, \cdots, f_k^{(m)}\} \sim p(\boldsymbol{x}_k|\mathcal{F}_{k}^{\boldsymbol{y}}) && \text{filter}\\
&\{s_{k|N}^{(1)}, \cdots, s_{k|N}^{(m)}\} \sim p(\boldsymbol{x}_k|\mathcal{F}_{N}^{\boldsymbol{y}}) && \text{smoother}
\end{aligned}$$

In detail, $\{p_k^{(1)}, \cdots, p_k^{(m)}\}$ are m independent samples generated by density $p(\boldsymbol{x}_k|\mathcal{F}_{k-1}^{\boldsymbol{y}})$, as well as others. Therefore, we have the following approximation

$$p(\boldsymbol{x}_k|\mathcal{F}_{k-1}^{\boldsymbol{y}}) \approx \frac{1}{m}\sum_{l=1}^{m} 1_{x \geq p_k^{(l)}}(x).$$

In fact, given $\{f_{k-1}^{(1)}, \cdots, f_{k-1}^{(m)}\}$ from $p(\boldsymbol{x}_{k-1}|\mathcal{F}_{k-1}^{\boldsymbol{y}})$ and $\{w_k^{(1)}, \cdots, w_k^{(m)}\}$ from $\boldsymbol{w}_k$, we have

$$\begin{aligned}
p(\boldsymbol{x}_k|\mathcal{F}_{k-1}^{\boldsymbol{y}}) &= \int\int p(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}, \boldsymbol{w}_k|\mathcal{F}_{k-1}^{\boldsymbol{y}}) d\boldsymbol{x}_{k-1} d\boldsymbol{w}_k\\
&= \int\int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{w}_k, \mathcal{F}_{k-1}^{\boldsymbol{y}}) p(\boldsymbol{w}_k|\boldsymbol{x}_{k-1}, \mathcal{F}_{k-1}^{\boldsymbol{y}}) p(\boldsymbol{x}_{k-1}|\mathcal{F}_{k-1}^{\boldsymbol{y}}) d\boldsymbol{x}_{k-1} d\boldsymbol{w}_k\\
&= \int\int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{w}_k) p(\boldsymbol{w}_k) p(\boldsymbol{x}_{k-1}|\mathcal{F}_{k-1}^{\boldsymbol{y}}) d\boldsymbol{x}_{k-1} d\boldsymbol{w}_k\\
&= \int\int \delta(\boldsymbol{x}_k - \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{w}_k)) p(\boldsymbol{w}_k) p(\boldsymbol{x}_{k-1}|\mathcal{F}_{k-1}^{\boldsymbol{y}}) d\boldsymbol{x}_{k-1} d\boldsymbol{w}_k. \qquad (63)
\end{aligned}$$

where $\delta(\cdot)$ is the Dirac function. Hence, if we denote $p_k^{(j)} := \boldsymbol{f}(f_{k-1}^{(j)}, w_k^{(j)})$, we conclude (63). Next, denote

$$\alpha_k^{(j)} := p(\boldsymbol{y}_k|p_k^{(j)}) = \left|\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{y}}(\boldsymbol{y}_k, p_k^{(j)})\right| d_{\boldsymbol{w}}(\boldsymbol{h}(\boldsymbol{y}_k, p_k^{(j)})). \qquad (64)$$

Hence, we compute

$$\begin{aligned}
p(\boldsymbol{x}_k = p_k^{(i)}|\mathcal{F}_k^{\boldsymbol{y}}) &= p(\boldsymbol{x}_k = p_k^{(i)}|\boldsymbol{y}_k, \mathcal{F}_{k-1}^{\boldsymbol{y}})\\
&= \lim_{\Delta\boldsymbol{y}\to 0} \frac{p(\boldsymbol{x}_k = p_k^{(i)}, \boldsymbol{y}_k \leq \boldsymbol{y} \leq \boldsymbol{y}_k + \Delta\boldsymbol{y}|\mathcal{F}_{k-1}^{\boldsymbol{y}})}{p(\boldsymbol{y}_k \leq \boldsymbol{y} \leq \boldsymbol{y}_k + \Delta\boldsymbol{y}|\mathcal{F}_{k-1}^{\boldsymbol{y}})}\\
&= \frac{p(\boldsymbol{y}_k|p_k^{(i)}) p(\boldsymbol{x}_k = p_k^{(i)}|\mathcal{F}_{k-1}^{\boldsymbol{y}})}{\sum_{l=1}^m p(\boldsymbol{y}_k|p_k^{(l)}) p(\boldsymbol{x}_k = p_k^{(l)}|\mathcal{F}_{k-1}^{\boldsymbol{y}})} = \frac{\alpha_k^{(i)}}{\sum_{l=1}^m \alpha_k^{(l)}}.
\end{aligned}$$

Now re-sampling $\{f_k^{(1)}, \cdots, f_k^{(m)}\}$ from $\{p_k^{(1)}, \cdots, p_k^{(m)}\}$ by

$$f_k^{(j)} = p_k^{(i)} \text{ with probability } \frac{\alpha_k^{(i)}}{\sum_{l=1}^m \alpha_k^{(l)}} \qquad (65)$$

Therefore, $p(\boldsymbol{x}_k|\mathcal{F}_k^{\boldsymbol{y}})$ can be approximated by

$$p(\boldsymbol{x}_k|\mathcal{F}_k^{\boldsymbol{y}}) \approx \frac{1}{\sum_{l=1}^m \alpha_k^{(l)}} \sum_{j=1}^m \alpha_k^{(j)} 1_{x \geq p_k^{(j)}}(x). \qquad (66)$$

Algorithm 6.3.1 (Monte-Carlo filter)**.** *The whole filtering process can be summarized as the following steps.*

Initial step: generate $\{f_0^{(1)}, \cdots, f_0^{(m)}\}$ *by* $p(\boldsymbol{x}_0)$.

For-loop step: for $k = \{1, \cdots, K\}$

1. *Generate* $\{w_k^{(1)}, \cdots, w_k^{(m)}\}$ *by* $d_{\boldsymbol{w}}(\cdot)$.
2. *Compute* $\{p_k^{(1)}, \cdots, p_k^{(m)}\}$ *by* $p_k^{(i)} = \boldsymbol{f}(f_{k-1}^{(i)}, w_k^{(i)})$.
3. *Compute* $\{\alpha_k^{(1)}, \cdots, \alpha_k^{(m)}\}$ *by (64).*
4. *Re-sample* $\{f_k^{(1)}, \cdots, f_k^{(m)}\}$ *from* $\{p_k^{(1)}, \cdots, p_k^{(m)}\}$ *by (66).*

Now denote $\{s_{1|N}^{(j)}, \cdots, s_{n|N}^{(j)}\}$ be the j-th sample of $p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n|\mathcal{F}_N^{\boldsymbol{y}})$ for $j = 1, \cdots, m$. In addition, let $p(\boldsymbol{x}_1 = s_{1|n-1}^{(j)}, \cdots, \boldsymbol{x}_{n-1} = s_{n-1|n-1}^{(j)}|\mathcal{F}_{n-1}^{\boldsymbol{y}}) = 1/m$ and generate $w_n^{(j)}$ by $\boldsymbol{w}_n$. Therefore, define

$$p_{i|n-1}^{(j)} = \begin{cases} s_{i|n-1}^{(j)} & i = 1, \cdots, n-1 \\ \boldsymbol{f}(s_{n-1|n-1}^{(j)}, w_n^{(j)}) & i = n \end{cases}$$

and we conclude the following updating process

$$\begin{aligned} & p(\boldsymbol{x}_1 = p_{1|n-1}^{(j)}, \cdots, \boldsymbol{x}_n = p_{n|n-1}^{(j)}|\mathcal{F}_n^{\boldsymbol{y}}) \\ &= \frac{p(\boldsymbol{y}_n|\boldsymbol{x}_1 = p_{1|n-1}^{(j)}, \cdots, \boldsymbol{x}_n = p_{n|n-1}^{(j)}, \mathcal{F}_{n-1}^{\boldsymbol{y}})p(\boldsymbol{x}_1 = p_{1|n-1}^{(j)}, \cdots, \boldsymbol{x}_n = p_{n|n-1}^{(j)}|\mathcal{F}_{n-1}^{\boldsymbol{y}})}{p(\boldsymbol{y}_{n-1}|\mathcal{F}_{n-1}^{\boldsymbol{y}})} \\ &= \frac{p(\boldsymbol{y}_n|p_{n|n-1}^{(j)})p(\boldsymbol{x}_1 = p_{1|n-1}^{(j)}, \cdots, \boldsymbol{x}_n = p_{n|n-1}^{(j)}|\mathcal{F}_{n-1}^{\boldsymbol{y}})}{p(\boldsymbol{y}_{n-1}|\mathcal{F}_{n-1}^{\boldsymbol{y}})} \end{aligned}$$

Since the value of $p_{n|n-1}^{(j)} \equiv p_n^{(j)}$ from Algorithm 6.3.1, the smoothing method can be derived by add the following step after (4) of Algorithm 6.3.1.

5. *Re-sample* $\{(s_{1|n}^{(j)}, \cdots, s_{n|n}^{(j)}), j = 1, \cdots, m\}$ *from* $\{(s_{1|n-1}^{(j)}, \cdots, s_{n-1|n-1}^{(j)}, p_n^{(j)}), j = 1, \cdots, m\}$ *by (66).*

Back to our parameters inference task and denote $\Theta = (a, b, c, \mu)$, since we can simulate the density of $p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n|\mathcal{F}_n^{\boldsymbol{y}})$ by Monte-Carlo simulation, now we numerically compute the updating process of EM algorithm in (56), (57), (58) and (59). In detail, $\mathbb{E}_{\mathbb{P}}[S_t|\mathcal{F}_t^{y}]$ in (60) can be approximated by

$$\frac{1}{m}\sum_{k=1}^{t}\sum_{j=1}^{m} H(y_k)F(s_{k|t}^{(j)})G(s_{k-1|t}^{(j)}).$$

For numerical experiment, let $a = 0.02, b = 0.01, c = 1, \mu = 2$ and $\Delta t = 1$, we simulate (x_t, y_t) for $t = 0, \cdots, 100$ by (52). Then choose $N = \{10, 20, \cdots, 100\}$ and apply **Algorithm 6.3.1** with $m = 500$ samples to approximate $p(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N|\mathcal{F}_N^{\boldsymbol{y}})$ for each N, let $\epsilon = 10^{-6}$ be the accuracy of EM algorithm, that is, we stop the updating process if

$$||\Theta_{k+1} - \Theta_k|| < \epsilon.$$

As a summary of the results from §6.2 and §6.3, the whole estimation process is shown below.

Algorithm 6.3.2 (Parameters estimation)**.** *As a summary of the results from §6.2 and §6.3, the whole estimation process is shown below.*

For-loop step: for $N \in \{10, 20, \cdots, 100\}$ and set $k = 0$.

1 *Generate $\{\boldsymbol{y}_0, \cdots, \boldsymbol{y}_N\}$ by (52) and Θ_k.*

2. *Apply Algorithm 6.3.1 by letting $K = N$ and giving Θ_k, obtain $\{(s_{1|N}^{(j)}, \cdots, s_{N|N}^{(j)}), j = 1, \cdots, 500\}$.*

3. *Updating Θ_{k+1} by (56), (57), (58) and (59).*

4. *If $||\Theta_{k+1} - \Theta_k|| < \epsilon$, breaking out; else back to step 1.*

We run Algorithm 6.3.2 for 100 times and compute the mean and 95% confidence interval of (a, b, c, μ), see Figure 2.

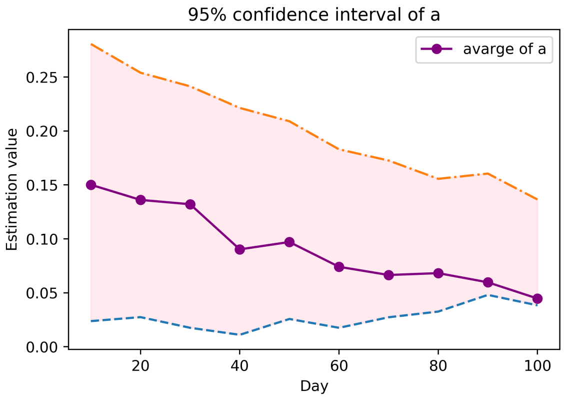
6.4 Particle filters

We have discussed Monte-Carlo filter method and re-sampling technique in §6.3, now, we will focus on one special Monte-Carlo filter, particle filter, and discuss the re-sampling method in detail. The key work is to estimate the conditional probability $p_t(x) := p(x|\mathcal{F}_t^y)$. Instead of using explicit expression (54), we will use the following approximation.

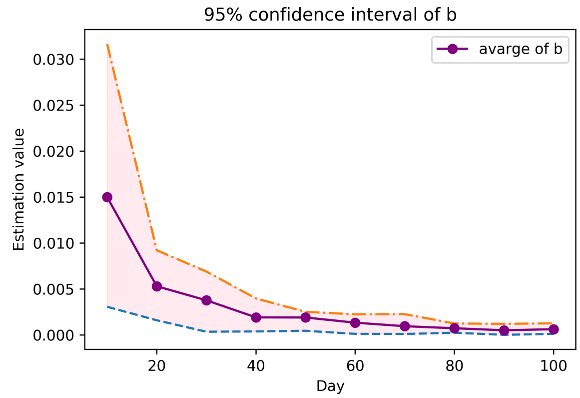
$$p_t^N(x) = \sum_{k=1}^{N} w_t^{(k)} \delta(x - x^{(k)}).$$

where $w_k \in (0, 1)$ such that $\sum_{k=1}^{N} w_k = 1$ and $\delta(\cdot)$ is Dirac function. Here, N is the number of particles and $x^{(k)}$ is our particle. The main trouble is that we do not know the explicit form of $p_t(x)$, hence denote another known positive density function $\pi_t(x) := \pi(x|\mathcal{F}_t^y)$ and concludes

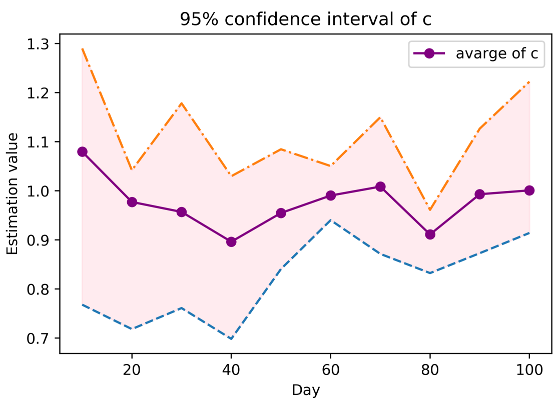
$$\begin{aligned}
\int f(x_{0:t}) p(x_{0:t}|y_{0:t}) dx_{0:t} &= \int f(x_{0:t}) \frac{p(x_{0:t}|y_{0:t})}{\pi(x_{0:t}|y_{0:t})} \pi(x_{0:t}|y_{0:t}) dx_{0:t} \\
&= \int f(x_{0:t}) \frac{p(x_{0:t}, y_{0:t})}{p(y_{0:t}) \pi(x_{0:t}|y_{0:t})} \pi(x_{0:t}|y_{0:t}) dx_{0:t} \\
&= \frac{\int f(x_{0:t}) \widetilde{w}(x_{0:t}) \pi(x_{0:t}|y_{0:t}) dx_{0:t}}{\int \widetilde{w}(x_{0:t}) \pi(x_{0:t}|y_{0:t}) dx_{0:t}}
\end{aligned}$$



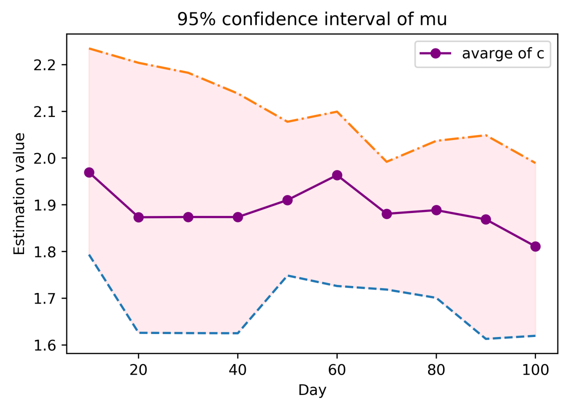
(a) Estimation result of a.



(b) Estimation result of b.



(c) Estimation result of c.



(d) Estimation result of μ.

Figure 2: Estimation result of (a, b, c, μ) from **Algorithm 6.3.2** with 95% confidence interval.

where $\widetilde{w}(x_{0:t}) = \frac{p(x_{0:t}, y_{0:t})}{\pi(x_{0:t}|y_{0:t})}$. Then by Monte Carlo simulation and large number law, we conclude

$$\frac{1}{N}\sum_{k=1}^{N} w_t^{(k)} f(x_{0:t}^{(k)}) \approx \int f(x_{0:t}) p(x_{0:t}|y_{0:t}) dx_{0:t}.$$

where $x^{(k)}$ are generated by distribution of $\pi_t(x)$ for $k = 1, \cdots, N$ and

$$w_t^{(k)} := \frac{\widetilde{w}(x_{0:t}^{(k)})}{\sum_{k=1}^{N} \widetilde{w}(x_{0:t}^{(k)})}. \tag{67}$$

The core task is to update weight $\widetilde{w}(x_{0:t+1})$ from $\widetilde{w}(x_{0:t})$, based on Bayes formula, it concludes that

$$\widetilde{w}(x_{0:t+1}) = \frac{p(x_{0:t+1}, y_{0:t+1})}{\pi(x_{0:t+1}|y_{0:t+1})} = \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)}{\pi(x_{t+1}|x_{0:t}, y_{0:t+1})} \frac{p(x_{0:t}, y_{0:t})}{\pi(x_{0:t}|y_{0:t+1})}$$

In addition, we assume that $\pi(x_{0:t}|y_{0:t+1}) = \pi(x_{0:t}|y_{0:t})$ and concludes that

$$\widetilde{w}(x_{0:t+1}) = \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)}{\pi(x_{t+1}|x_{0:t}, y_{0:t+1})}\widetilde{w}(x_{0:t})$$

Although $\pi_t(\cdot)$ can be any of probability distribution, it's better find a optimal one which relates to $p_t(\cdot)$. Here, we choose

$$\pi(x_{t+1}|x_{0:t}, y_{0:t+1}) = p(x_{t+1}|x_t).$$

With this assumption, we have

$$\widetilde{w}(x_{0:t+1}) = p(y_{t+1}|x_{t+1})\widetilde{w}(x_{0:t}).$$

However, there is one trouble about particle decreasing, i.e. some of the weights will tend to zero after several iterations. One way to solve this trouble is re-sampling [13]. In detail, consider the distribution of $w_t^{(k)}$ for $k = 1, \cdots, N$ and generate the integer $j \in \{1, \cdots, N\}$ such that $j \sim \{w_t^{(k)}\}_{k=1}^N$. Now, the whole step of particle filter is the following.

Algorithm 6.4.1 (Particle filter)**.** *The whole process of particle filter can be summarized by the following steps.*

Initial step: for each $i \in \{1, \cdots, N\}$. Generate $x_0^{(i)}$ based on the priori distribution of x_0.

For-loop step: for $t \in \{0, \cdots, T-1\}$ and for $i \in \{1, \cdots, N\}$.

1. *Re-sampling: Generate an integer $j \in \{1, \cdots, N\}$ based on distribution of $w_t^{(k)}$ for $k = 1, \cdots, N$.*
2. *Sampling: Generate $x_{t+1}^{(i)}$ based on distribution of $p(x_{t+1}|x_t^{(j)})$.*
3. *Updating: Compute $\widetilde{w}^{(i)}(x_{0:t+1}) = p(y_{t+1}|x_{t+1}^{(i)})$.*
4. *Normalizing: Using (67) to compute $w_{t+1}^{(i)}$.*

For the numerical experiment, we simulate (x_t, y_t) for $t = 0, \cdots, 100$ by the same method of §6.3 with parameter value $(a, b, c, \mu) = (0.02, 0.01, 1, 2)$. Next apply Algorithm 6.4.1 with 2000 particles and compare our estimation results with the true value, see Figure 3.

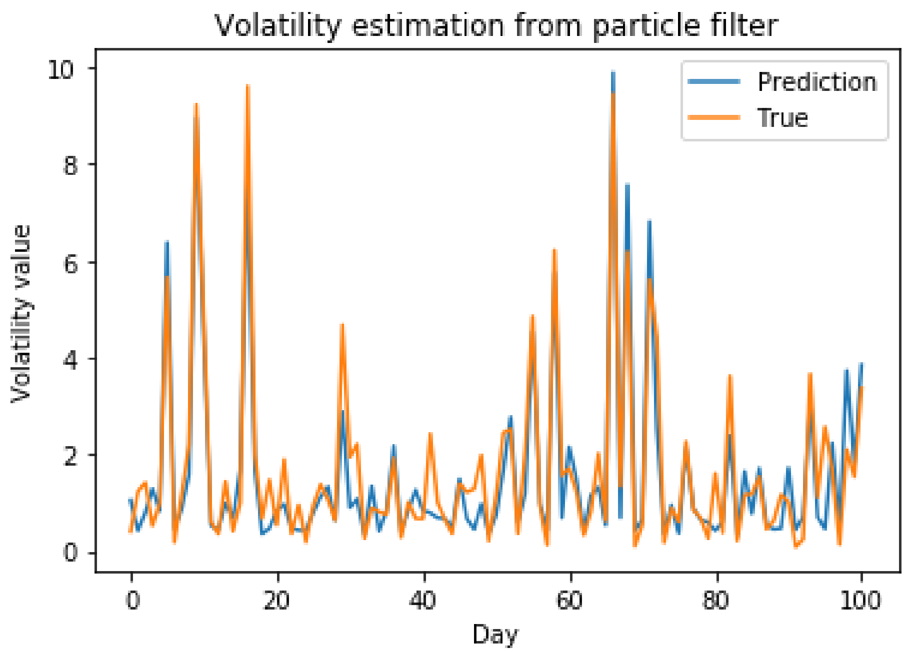


Figure 3: Simulation value (orange) and prediction value (blue) from particle filter.

7 Recurrent Neural Networks in Filtering Theory

Until now, we have introduced KF (EKF) and PF in solving both linear and non-linear filtering problems. Although the estimation of signal process is quite close to the true value in both pose and volatility estimation, these two cases all need the explicit expression of stochastic differential equations which govern signal and observation processes. However, such information may be not available sometimes, in detail, we have no idea about the expression of $a(\cdot), b(\cdot), h(\cdot), k(\cdot)$ in (1). So all these method can not be extended into more general cases. But neural networks provide a wider and reliable solution to deal with this trouble. In paper [26] and [12], they introduce some network optimal filters with layer structures of interconnected neurons and recurrent multilayer perceptions. In our paper, we will use RNNs to deal with filtering problems, which indeed has better performance than classical neural networks.

7.1 Structure of RNNs

A basic neuron unit in RNNs has the following form Figure (4). There are two inputs for one RNNs neuron, the active value from the forward unit a_t and input data from last layer x_t, where t means this neuron is the t-th unit in its layer. Then combining weight $W^t_{aa}, W^t_{ax}, b^t_a$ and getting $W^t_{aa}a_t + W^t_{ax}x_t + b^t_a$. Next compute the output active value by the activation $\tanh(\cdot)$, and we can use other activations such as sigmoid and 'relu' functions. Finally the output active value a_{t+1} is derived by $\tanh(W^t_{aa}a_t + W^t_{ax}x_t + b^t_a)$ and $h_t = W^t_{ya}a_{t+1} + b^t_y$. By connecting these RNNs units as a sequence, we obtain a simple RNNs layer. In application, it is usually construct several different kinds of layers and combine them as a whole neural

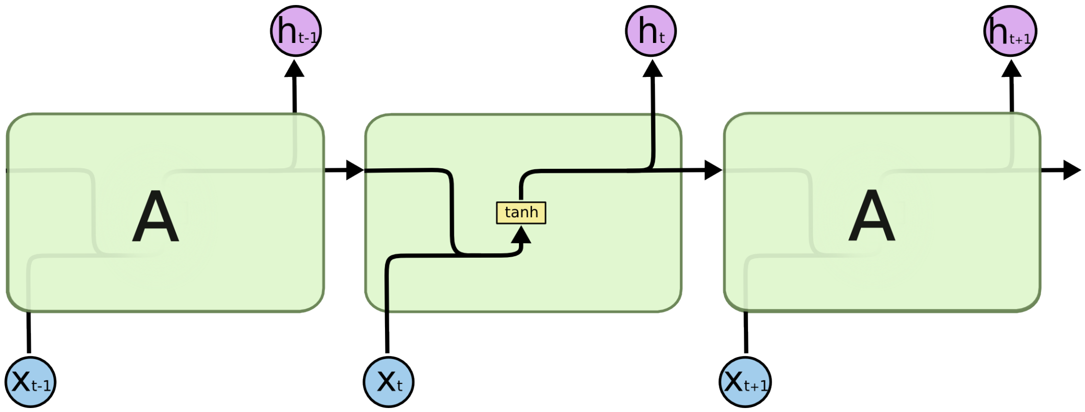


Figure 4: The structure of SimpleRNNs unit, cite from colah's blog, *Understanding LSTM Networks*.

networks.

The merit of RNNs is to use prior information to estimate the current states. However, from the structure of RNNs units, only the most recent information is used in the current states' estimation, in other words, the prediction results of traditional RNNs never depends on long-term information. One solution of this problem is to add the gate structure in each basic RNNs unit. And we introduce LSTM (Long Short Term Memory) networks as an example.

The basic unit of LSTM is shown in Figure 5. The main difference between RNNs and LSTM is the 'gate layer' structure. In fact, there are three stages in one LSTM unit.

1. Forget stage: For the gate information from last unit C_{t-1}, the gate structure will optionally forget unimportant information. In detail, for the two inputs a_t and x_t, first compute the 'forget gate' $f_t = \sigma(W^t_{fa}a_t + W^t_{fx}x_t + b^t_f)$, where $\sigma(\cdot)$ is the sigmoid function.

2. Optional remember stage: Based on current input a_t, x_t, gate structure will optionally remember important information. Compute the second 'input gate' by $i_t = \sigma(W^t_{ia}a_t + W^t_{ix}x_t + b^t_i)$ and $\widetilde{C}_t = \tanh(W^t_{ca}a_t + W^t_{cx}x_t + b^t_c)$, and combine with the gate information from last unit C_{t-1}, compute the update gate state $C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t$, where $\odot$ is Hadamard Product.

3. Output stage: Compute the 'output gate' $o_t = \sigma(W^t_{oa}a_t + W^t_{ox}x_t + b^t_o)$, moreover the forward unit and output $y_t, a_{t+1} = o_t \odot C_t$.

7.2 RNNs filter

Since the structure of RNNs layer has a sequential property, that is, the i-th output y_i is determined by a_i and x_i, where a_i depends on x_{i-1} and a_{i-1}. Besides, the discrete filtering problem has the form of (28), in detail, if we ignore the influence of noises, the observation

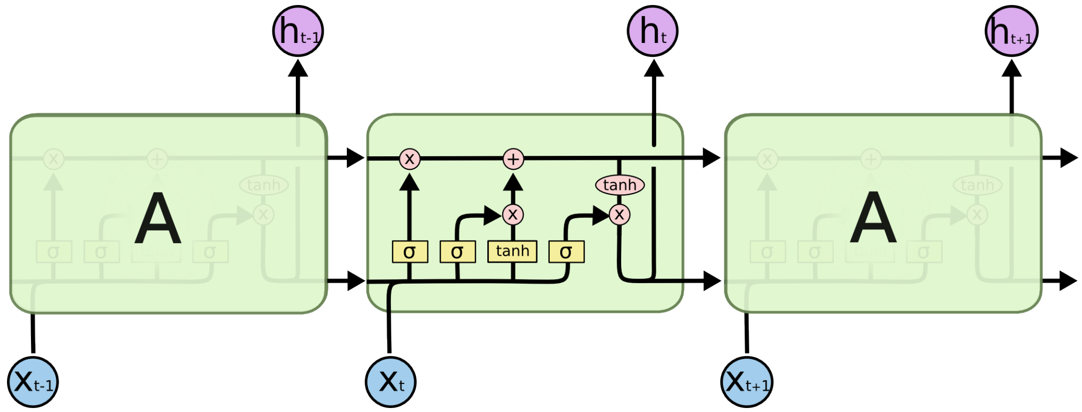


Figure 5: The structure of LSTM unit, cite from colah's blog, *Understanding LSTM Networks*.

$\boldsymbol{y}_k$ is determined by $\boldsymbol{x}_k$, and $\boldsymbol{x}_k$ depends on $\boldsymbol{x}_{k-1}$ and $\boldsymbol{u}_k$, which quite fits the RNNs model. In fact, we have the following result.

Theorem 7.2.1. *Suppose $(\boldsymbol{x}_t, \boldsymbol{y}_t)_{t=0}^K$ is defined by (28) and $f : \mathbb{R}^n \to \mathbb{R}^p$ is a Borel measurable function such that $\mathbb{E}[||f(\boldsymbol{x}_t)||^2] < \infty$ for all $t = 0, \cdots, K$, then for any $N \in \mathbb{N}^+$, there exists a RNNs filter $\boldsymbol{\alpha}(t; N)$ with N layers, such that the following function is monotone decreasing and converges to 0 as $N \to \infty$*

$$r(N) := \min_w \sum_{t=0}^{K} \mathbb{E}[||\boldsymbol{\alpha}(t; N) - \mathbb{E}[f(\boldsymbol{x}_t)|\mathcal{F}_t^{\boldsymbol{y}}]||^2] \tag{68}$$

where w is the set of all parameters in RNNs.

Proof. Based on the structure of RNNs, the output $\boldsymbol{\alpha}_t(t; N)$ can be regarded as a function of $(\boldsymbol{y}_t, \boldsymbol{a}_t)$, where $\boldsymbol{a}_t$ is N-dimensional vector with active value a_t^i from prior units in i-th layer. In detail, we write

$$\boldsymbol{\alpha}_t(t; N) := F(\boldsymbol{y}_t, \boldsymbol{a}_t).$$

On the other hand, the active value vector $\boldsymbol{a}_t$ depends on $\boldsymbol{y}_j$ for $j = 0, \cdots, t$ with parameters defined in §7.1, hence we denote Θ_t^N as the set of parameters in t-th units of all layers, and $F(\cdot)$ can be rewritten as $F(\boldsymbol{y}_{0:t}|\Theta_t^N)$. In addition, suppose $\{q_{m,n}^t\}_{m,n=1}^{\infty}$ be the solution of (16) by the implicit Euler approximation method in §3.2, where $q_{m,n}^k$ can be derived from a equation of $(\boldsymbol{y}_k, \boldsymbol{y}_{k-1})$. Therefore, our core task is to show that the following approximation, for any bounded and Borel measurable $G(\boldsymbol{y}_t, \boldsymbol{y}_{t-1})$

$$\lim_{N\to\infty} \min_{\Theta_t^N} ||G(\boldsymbol{y}_t, \boldsymbol{y}_{t-1}) - F(\boldsymbol{y}_{0:t}|\Theta_t^N)||_{L^2} = 0.$$

First, since $\mathbb{E}[||f(\boldsymbol{x}_t)||^2] < \infty$ for all $t = 0, \cdots, K$, we conclude the following by dominated convergence theorem

$$\mathbb{E}\left[f(\boldsymbol{x_t})|\mathcal{F}_t^{\boldsymbol{y}}\right] = \lim_{m,n\to\infty} \int f(\boldsymbol{x_t}) q_{m,n}^t(\boldsymbol{x_t}) d\boldsymbol{x_t}. \tag{69}$$

where $\mathbb{E}\left[f(\boldsymbol{x_t})|\mathcal{F}_t^{\boldsymbol{y}}\right]$ is function of $\boldsymbol{y}_{0:t}$ and the right side of above equation depends on $\boldsymbol{y}_{t-1:t}$, which is denoted by $G(\boldsymbol{y}_t, \boldsymbol{y}_{t-1})$. In addition, by Corollary 2.2 of [14], it shows that the p-dimensional functions derived from neural networks are dense in $L^2(\mathbb{R}^p, \mathbb{P})$, that is, (69) is valid. Besides, we have

$$h_t(N) := \min_{\Theta_t^N} ||G(\boldsymbol{y}_t, \boldsymbol{y}_{t-1}) - F(\boldsymbol{y}_{0:t}|\Theta_t^N)||_{L^2}.$$

is a monotone decreasing function of N with limit of 0. As a result, we conclude for each $t = 0, \cdots, K$

$$\begin{aligned}\min_{\Theta_t^N} \mathbb{E}[||\boldsymbol{\alpha}(t;N) - \mathbb{E}[f(\boldsymbol{x}_t)|\mathcal{F}_t^{\boldsymbol{y}}]||^2] &= \min_{\Theta_t^N} \mathbb{E}\left[\left|\left|F(\boldsymbol{y}_{0:t}|\Theta_t^N) - \int f(\boldsymbol{x_t}) q_{m,n}^t(\boldsymbol{x_t}) d\boldsymbol{x_t}\right|\right|^2\right] \\ &= \mathbb{E}[h_t(N)^2] \to 0 \text{ as } N \to \infty\end{aligned}$$

Finally, we conclude that

$$r(N) = \lim_{m,n\to\infty} \sum_{t=0}^{K} \mathbb{E}[h_t(N)^2].$$

Therefore, $r(N)$ is also a monotone decreasing function of N with limit of 0. □

7.3 Numerical result

In this subsection, we first apply RNNs (LSTM) filter on pose estimation model. Here, we construct 6 LSTM layers, with $400, 200, 100, 50, 25, 12$ units on each layer. For the training data, we first generate 20 fixed points $\{p_j\}_{j=1}^{20}$ by $\mathcal{N}(\mathbf{0}, \text{diag}(1,1,1))$ and then use the same method as in §5.6 and get 50 sets of pose data $\{T_k^l\}_{k,l=1}^{100,50}$ with their measurements $\{\boldsymbol{y}_k^l\}_{k=0,l=1}^{100,50}$. For the optimizer, we use Adam optimization algorithm to train our RNNs filter for 200 times with the leaning rate of 0.001. Now, for one test data generated by 48, we still use 2-norms of matrix to show the accuracy of prediction from RNNs filter, see Figure 6.

For the volatility estimation model, we construct 6 simple RNNs layers instead of LSTM layers, with $500, 200, 200, 200, 100, 1$ units on each layer. For the training data, we simulate 50 sets of $\{(x_t^i, y_t^i)\}_{t,i=1}^{100}$ by (52) and still use Adam optimization algorithm to train our RNNs filters for 200 times with the learning rate of 0.001. Our estimation results is shown in Figure 7.

8 Conclusion

In this section, we summary our numerical experiments of different filters and compare their performance separately. First, for both pose and volatility estimation, it is clear that EKF and particle filter have better accuracy than RNNs filter, see Figure 8 and 9. Next, we will discuss the advantages and weakness of each filter separately.

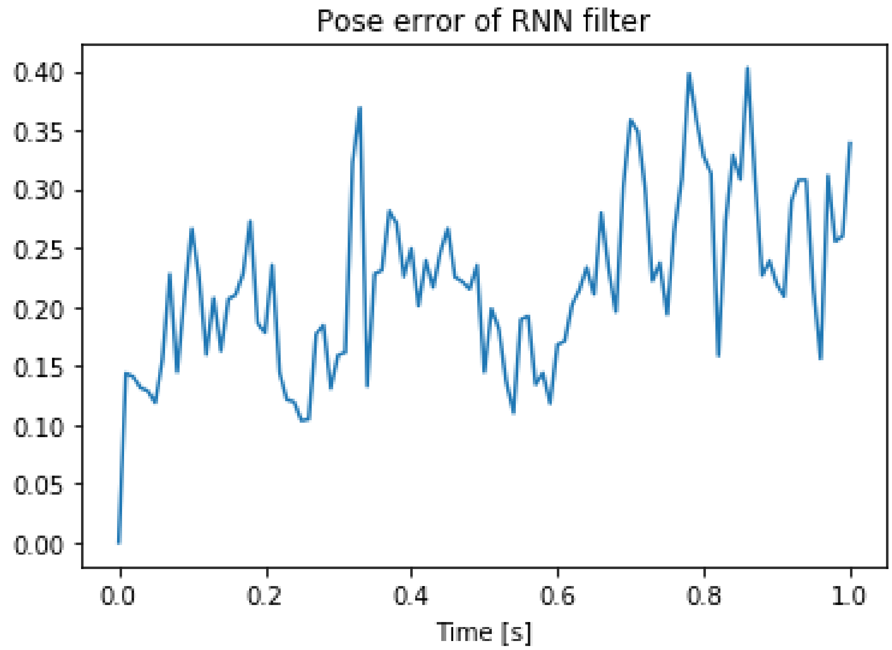


Figure 6: The error of pose $||\widehat{T}_k - T_k||_2$ for all $k = 0, \cdots, 100$

For (Extended) Kalman filter, it has efficient computation process, see (25), (26) and (27). Unlike RNNs filter, which needs the training process as preparation and will occupy a lot of computational sources, KF allows us to obtain the estimation of signal process quickly. Besides, KF has better accuracy compared than RNNs filter, see Figure 8. Even for some non-linear filtering problems, the EKF can also have good performance, such as pose estimation. On the other hand, the main weakness of KF is that it can not be widely generalized. Although we can use Taylor expansion as in §4.4 to approximate a non-linear filtering problems by linear filter, it depends on the smoothness of functions $\boldsymbol{f}(\cdot), \boldsymbol{g}(\cdot)$ in (28). In addition, KF also requires the noises of both signal and observation should be Gaussian. All these requirements limits the application of KF.

For particle filter introduced in §6.4, it overcomes some limitations of KF. First, the noise of signal and observation may not be Gaussian, any other known distributions are all suitable. Although we use Monte-Carlo simulation in computing the conditional probability, we can still obtain more accurate estimation results by increasing the number of particles. Therefore, particle filter has wider usage than KF. The main weakness is that we need the explicit expression of $\boldsymbol{f}(\cdot), \boldsymbol{g}(\cdot)$ in (28), since we need the distribution of $p(x_{t+1}|x_t)$ and $p(y_{t+1}|x_{t+1})$ in updating process of **Algorithm 6.4.1**. But in most of cases, all these information is priorly definite, hence PF is indeed a powerful tool for many different filtering problems in physics and engineer fields.

However, for some special and extreme cases, the informations about noises and SDE of signal and observation process are not available. Now RNNs filter provides us even more general application than PF and EKF. All we need is the training data and enough computa-

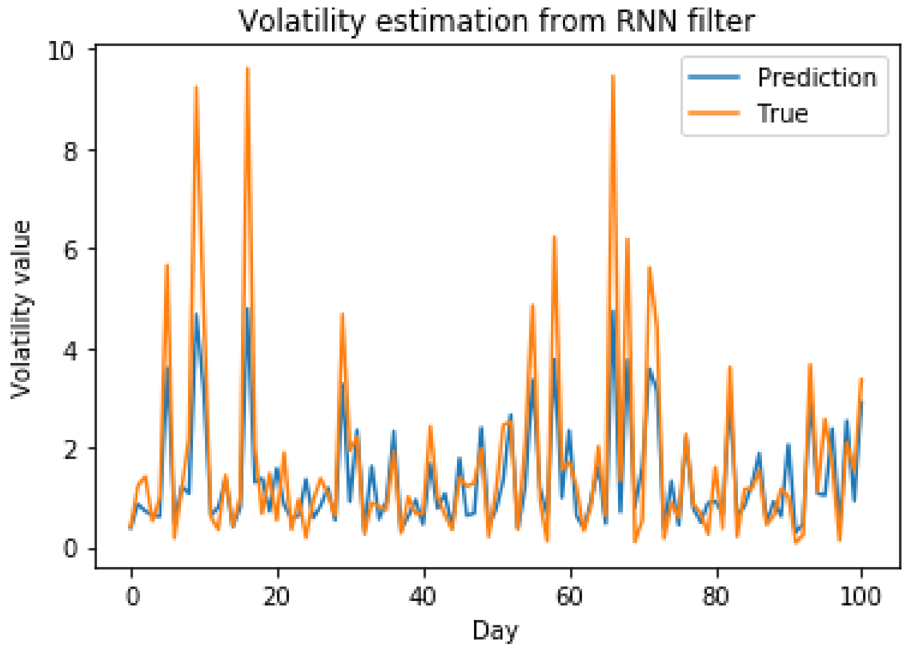


Figure 7: Simulation value (orange) and prediction value (blue) from RNNs filter.

tion resource, in fact, from **Theorem 7.2.1**, large amounts of training data and large-scale networks structure can both improve the performance of RNNs filter. On the contrary, it is clear to see that training RNNs filter occupy more time and computation power than EKF and PF in order to obtain the same performance. For example, due to the limited capacity of our computers, RNNs filter cause more estimation error than EKF and PF, see Figure 8 and 9. Furthermore, getting large amount of training data may not be an easy task. With limited amount of data, RNNs filter can not perform well.

For EM method and Monte-Carlo simulation in §6.2 and §6.3, there is another idea by using stochastic optimization. In detail, we can construct the likelihood function by

$$L(\Theta) = p(\boldsymbol{y}_1, \cdots, \boldsymbol{y}_K|\Theta) = \prod_{k=1}^{K} p(\boldsymbol{y}_k|\mathcal{F}^{\boldsymbol{y}}_{k-1}) \tag{70}$$

In addition, we have

$$p(\boldsymbol{y}_k|\mathcal{F}^{\boldsymbol{y}}_{k-1}) = \int p(\boldsymbol{y}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\mathcal{F}^{\boldsymbol{y}}_{k-1})d\boldsymbol{x}_k \approx \frac{1}{m}\sum_{j=1}^{m} p(\boldsymbol{y}_k|p_k^{(j)}) = \frac{1}{m}\sum_{j=1}^{m}\alpha_k^{(j)} \tag{71}$$

Hence, the log-likelihood function is

$$l(\Theta) = \sum_{k=1}^{K}\log(p(\boldsymbol{y}_k|\mathcal{F}^{\boldsymbol{y}}_{k-1})) \approx \sum_{k=1}^{K}\log\left(\sum_{j=1}^{m}\alpha_k^{(j)}\right) - K\log(m) \tag{72}$$

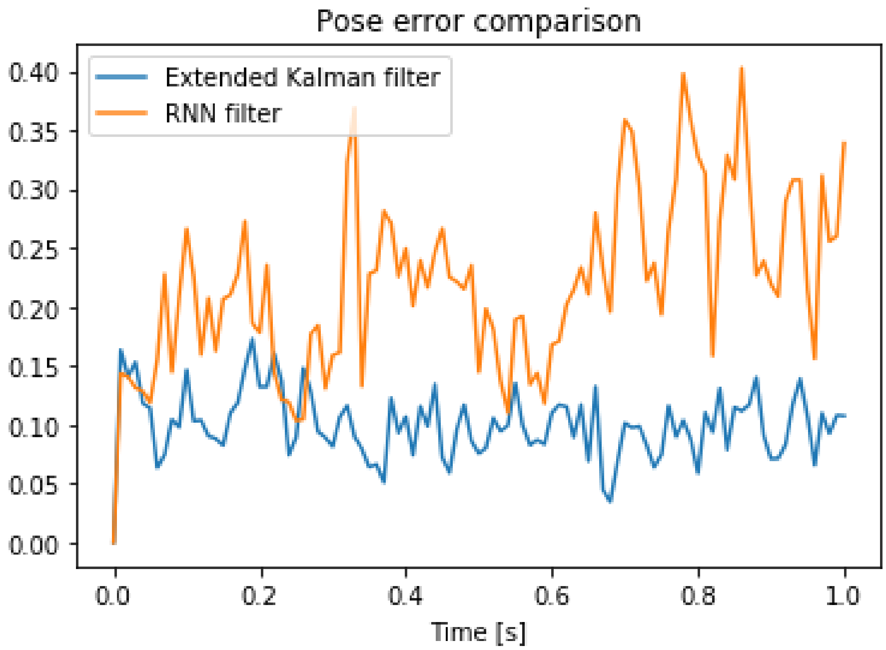


Figure 8: Comparison of RNNs filter (orange) and Extended Kalman filter (blue).

This is a classic stochastic optimization task, since the unknown parameters (a, b, c) will not directly appears in $l(\Theta)$, but these parameters indeed govern the SDE (52) of signal process. In paper [32], it introduces stochastic optimization frameworks for optimal control and reinforcement learning, which can be generalized to our model.

In the end, let's discuss some possible improvements about these filters. For EKF, since we use Taylor expansion to linearize the non-linear filtering equations, see §4.4, using the second-order terms of Taylor expansion can improve the performance of EKF, see [37]. However, the accuracy of these two methods depends on the coordinate system, see Picard [31]. Another improvement method is geometric intrinsic filter [7], which is coordinate invariant. Hence, it has better performance than EKF. As for PF, introducing neural networks in the scheme of prediction process can improve the accuracy of PF [11], meanwhile, neural networks can numerically infer the unknown parameters in (28), which extend the usage of PF into more general filtering problems. Finally, other neural networks technique instead of RNNs filter can also be applied in filtering problems, such as reinforcement learning [28], which has better performance in numerical inference.

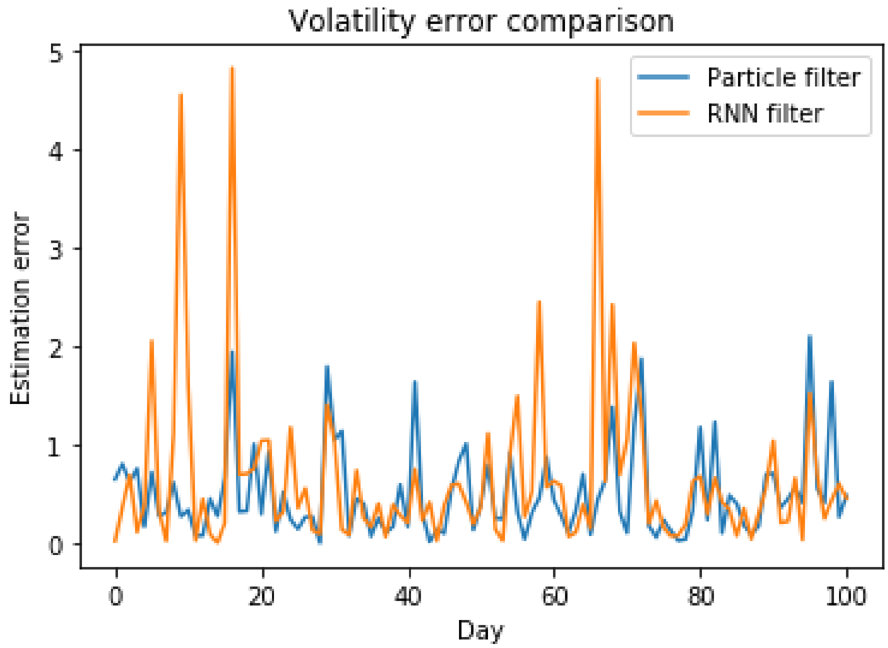


Figure 9: Comparison of RNNs filter (orange) and Particle filter (blue).

References

[1] Shin Ichi Aihara, Arunabha Bagchi, and Saikat Saha. On parameter estimation of stochastic volatility models from stock data using particle filter-application to aex index. *International journal of innovative computing, information and control*, 5(1):17–27, 2009.

[2] Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2017.

[3] Allan Bensoussan, Roland Glowinski, and A Rascanu. Approximation of the zakai equation by the splitting up method. *SIAM journal on control and optimization*, 28(6):1420–1431, 1990.

[4] Haim Brezis. *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

[5] Zhe Chen et al. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics: A Journal of Theoretical and Applied Statistics*, 182(1):1–69, 2003.

[6] Jean-Michel Coron. *Control and nonlinearity*. American Mathematical Soc., 2007.

[7] RWR Darling. Geometrically intrinsic nonlinear recursive filters i: algorithms. *arXiv preprint math/9809028*, 1998.

[8] Tyrone Edward Duncan. Probability densities for diffusion processes with applications to nonlinear filtering theory and detection theory. Technical report, STANFORD UNIV CA STANFORD ELECTRONICS LABS, 1967.

[9] Robert J Elliott, John van der Hoek, and Jorge Valencia. Nonlinear filter estimation of volatility. *Stochastic analysis and applications*, 28(4):696–710, 2010.

[10] Jaques Faraut. *Analysis of Lie Groups.* Cambridge University Press Cambridge, 2008.

[11] Helaine Cristina Morais Furtado, Haroldo Fraga de Campos Velho, and Elbert Einstein Nehrer Macau. Data assimilation: Particle filter and artificial neural networks. *Journal of Physics: Conference Series*, 135(1):012073, 2008.

[12] Simon Haykin, Paul Yee, and Eric Derbez. Optimum nonlinear filtering. *IEEE Transactions on signal processing*, 45(11):2774–2786, 1997.

[13] Jeroen D Hol. Resampling in particle filters. *Institutionen för systemteknik*, 2004.

[14] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[15] Yozhong Hu, Gopinath Kallianpur, and Jie Xiong. An approximation for the zakai equation. *Applied Mathematics and Optimization*, 45(1):23–44, 2002.

[16] Peter C Hughes. *Spacecraft attitude dynamics.* Courier Corporation, 2012.

[17] Kazufumi Ito. Approximation of the zakai equation for nonlinear filtering. *SIAM Journal on Control and Optimization*, 34(2):620–634, 1996.

[18] A. H. Jazwinski. Stochastic processes and filtering theory. *Academic Press*, 1970.

[19] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82D:35–45, 1960.

[20] Genshiro Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25, 1996.

[21] Andrei Nikolaevitch Kolmogorov. Sur l'interpolation et extrapolation des suites stationnaires. *CR Acad. Sci. Paris*, 208:2043–2045, 1939.

[22] Bernt Ksendal. Stochastic differential equations. *IEEE Transactions on Automatic Control*, 1989.

[23] H. J. Kushner. On the differential equations satisfied by conditional probability densities of markov processes. *Journal of the Society for Industrial and Applied Mathematics Series A Control*, 2(1), 1964.

[24] Jean-François Le Gall. *Brownian motion, martingales, and stochastic calculus.* Springer, 2016.

[25] Jacques Louis Lions. *Contrôle optimal de systemes gouvernés par des équations aux dérivées partielles.* Dunod, 1968.

[26] James Ting-Ho Lo. Neural network approach to optimal filtering. *World Congress of Nonlinear Analysts' 92*, pages 3705–3716, 2011.

[27] Steven I Marcus. Algebraic and geometric methods in nonlinear filtering. *SIAM Journal on Control and Optimization*, 22(6):817–844, 1984.

[28] Jun Morimoto and Kenji Doya. Reinforcement learning state estimator. *Neural computation*, 19(3):730–756, 2007.

[29] Richard Edgar Mortensen. *Optimal control of continuous-time stochastic systems*. University of California, Berkeley, 1966.

[30] Karmvir Singh Phogat and Dong Eui Chang. Discrete-time invariant extended kalman filter on matrix lie groups. *International Journal of Robust and Nonlinear Control*, 30(12):4449–4462, 2020.

[31] Jean Picard. Efficiency of the extended kalman filter for nonlinear systems with small noise. *SIAM Journal on applied mathematics*, 51(3):843–885, 1991.

[32] Warren B Powell. A unified framework for stochastic optimization. *European Journal of Operational Research*, 275(3):795–821, 2019.

[33] Steven E Shreve. *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer Science & Business Media, 2004.

[34] Alexander Meyer Sjøberg and Olav Egeland. Lie algebraic unscented kalman filter for pose estimation. *arXiv preprint arXiv:2005.00385*, 2020.

[35] Ruslan Leont'evich Stratonovich. Conditional markov processes. In *Non-linear transformations of stochastic processes*, pages 427–453. Elsevier, 1965.

[36] Ramon Van Handel. Stochastic calculus, filtering, and stochastic control. *Course notes., URL http://www. princeton. edu/rvan/acm217/ACM217. pdf*, 14, 2007.

[37] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press Cambridge, 1950.

[38] Moshe Zakai. On the optimal filtering of diffusion processes. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 11(3):230–243, 1969.