Variants of Kalman Filter and Noise Covariances Estimation

A project report of the course "EE251 Signal Detection, Parameter Estimation, and Statistical Learning" at the School of Information Science and Technology (SIST) of Shanghaitech University

Junyan Su
sujy@shanghaitech.edu.cn
student ID:94232173

***Abstract*—Kalman filter is a useful estimation algorithm that can be used in guidance, navigation and control of vehicles like spacecraft [1]. In the class, we have learned the basic and extended form of Kalman filter. This paper gives a brief review of some basic concepts of Kalman filtering such as the basic dynamical model, formulations, intuitions and some limitations. Then in the next section we will talks about some nonlinear variants of the Kalman filters like extended Kalman filter and some modifications used to improve the performance. Another kind of nonlinear Kalman filter called Unscented Kalman filter will be talked. Finally, we will introduce an efficient noise covariances estimator called autocovariance least square method.[2]**

I. Introduction

Kalman Filter is a powerful and practical tool used in many area like control and signal processing. This filter use the concept of successive estimation. First, it uses a model to predict the next signal and then it uses the measurement that comes to correct the prediction weighted by the estimated variance and true variance of the noise in measurement. This algorithm is better than those only based on a single measurement alone since it make use of the information of the model we make.

The very first version introduced by Kalman[3] is linear quadratic estimation. It is optimal in linear system only. In practice, however, most of the systems are nonlinear. To cope with this problem, a lot of extensions and generalizations have been developed. In this paper, we will talk about two forms of nonlinear Kalman filter that is widely used in many fields. i.e. extended Kalman filter(EKF) and unscented Kalman filters(UKF).

Another issue that is essential for Kalman filter is how to get a good estimate of the noise covariances efficiently. An unbiased practical estimator called autocovariance least-squares method(ALS)[2] is introduced in this article.

II. Simple Kalman Filter

In the class, we have learned about a basic simple Kalman filter in the chapter13 of the textbook[4]. So to begin with, let us review some concepts of the Kalman filter.

A. Dynamical Signal Models

The problem in Kalman Filter want to solve derive from the estimation problem. Take a DC level in WGN problem as an example:

$$x[n] = A + w[n]$$

where A is the parameter that we want to estimate, and w[n] is the White Gaussian Noise with variance σ^2. But in practical, the parameter A is not necessarily a constant. It may varies over time. So in this context, our problem becomes

$$x[n] = A[n] + w[n]$$

In this situation, we may want to use a dynamical signal model to use as much information we get as possible. To illustrate, we use a simple model same as [4]i.e. the *first-order Gauss-Markov* process:

$$s[n] = as[n-1] + u[n]$$

where $u[n]$ is WGN with variance σ_u^2, and $s[-1] \sim N(u_s, \sigma_s^2)$ is independent of u[n] for all $n \geq 0$. Note that it is quite simple model to illustrate the basic concepts of Kalman filter. Other more complicated model like the *second-order Gauss-Markov* process also can be considered to apply Kalman filter.

B. Formulation

Now we can introduce the formulation of the Kalman filter.

The signal s[n] and observation x[n] iterate over time with the following equation

$$s[n] = as[n-1] + u[n]$$

$$x[n] = s[n] + w[n]$$

with $u[n] \sim N(0, \sigma_u^2)$, $w[n] \sim N(0, \sigma_n^2)$ and $s[-1] \sim N(\mu_s, \sigma_s^2)$. Also u[n],w[n] and s[n] are independent of each other.

Follow the derivation in section 13.4 of the textbook[4], we can obtain the formula as below. The basic idea of derivation is applying Minimum Mean Square Error(MMSE) estimate and orthogonality principle to the prolem, and making use of the properties of dynamical signal models.

Prediction:

$$\hat{s}[n|n-1] = a\hat{s}[n-1|n-1]$$

Minimum Prediction MSE:

$$M[n|n-1] = a^2 M[n-1|n-1] + \sigma_u^2$$

Kalman Gain:

$$K[n] = \frac{M[n|n-1]}{\sigma_n^2 + M[n|n-1]}$$

Correction:

$$\hat{s}[n|n] + K[n](x[n] - \hat{s}[n|n-1])$$

Minimum MSE:

$$M[n|n] = (1 - K[n])M[n|n-1]$$

This formulas are beautiful since they are easy to implement and fit the our intuitions quite well. In each step, we predict the next signal with our dynamical signal model and corresponding variance. Then we correct the estimation with current observation x[n]. Note that the variance M[n] can be calculated offline. i.e. whenever our model for the process is settled, we can calculate all these M[i] no matter what observation x[n] is.

C. Optimality

The Kalman is the optimal linear filter if the following condition holds:

- The model matches the real system quite well and the real system is linear.
- Then noise entering is white and independence between signal and noise holds.
- The covariances of the noise are exactly known.

As can be seen, there are certain limitations for Kalman filter's optimality. In the real world, it is quite difficult to find the optimal filter since neither of those conditions can be fully satisfied. Linear models are only approximations of the actual process near the operating point. The noise covariance matrix is unknown and not easy to estimate. In the following section, we will talk about some variants created to cope with those limitations.

D. More Than Textbook

In the textbook[4], the derivation of Kalman filter is under Bayes's rule and the assumption that all estimates have independent, Gaussian noise. There is a misunderstanding that Kalman filter can only be applied under Gaussian assumptions. The original derivation[3], however, did not apply Bayes' rule and did not require any noise distribution information except mean and covariance.

III. Extended Kalman Filter

Notice that our simple Kalman filter is limited. It works well only for linear system. When our system become nonlinear,which is the most case in practice, there should be a more general extension for the filter. Suppose we have a general dynamical signal model:

$$s_k = f(s_{k-1}, u_k) + w_k$$

$$x_k = h(s_k) + v_k$$

where f(s,u) and h(s) are two functions corresponding the system and w_k and v_k are two zero mean Gaussian noises. In addition, u_k is the control signal.

And more generally, the noise signal does not have to be additive, our system becomes

$$s_k = f(s_{k-1}, u_k, w_k)$$

$$x_k = h(s_k, v_k)$$

The basic idea of extended Kalman filter is quite straight forward. That is, to linearize the system by a Taylor expansion and apply the simple Kalman filter showing above. More details can be seen in section 13.7 of the textbook[4]. Note that unlike simple form of Kalman filter, the estimated variance $M[n|n]$ has to be calculated online since we have to wait for the estimated signal to calculate the Taylor expansion and then the estimated variance.

Also note that the extended Kalman filter described in the textbook is a first-order extended Kalman filter(EKF). Higher order EKFs may be obtained by retaining more terms of the Taylor series expansions. For example, second and third order EKFs have been described. However, higher order EKFs tend to only provide performance benefits when the measurement noise is small.

The disadvantage of the extended Kalman filter is quite obvious: it is in general is not an optimal estimator any more. And due to the linearization, the estimate will not be accurate when system is not near linear. In order to improve the performance of the extended Kalman filter, there are certain modifications.[5]

A. Iterated extended Kalman filter

The iterated extended Kalman filter improves the linearization of the extended Kalman filter by recursively modifying the centre point of the Taylor expansion. This reduces the linearization error at the cost of increased computational requirements.

B. Robust extended Kalman filters

The extended Kalman filter arises by linearizing the signal model about the current state estimate and using the linear Kalman filter to predict the next estimate. This attempts to produce a locally optimal filter, however, it is not necessarily stable because the solutions of the underlying Riccati equation are not guaranteed to be positive definite.

As stated in [6], we can employ the H_∞ results from robust control. Robust filters are obtained by adding a positive definite term to the design Riccati equation. The additional term is parametrized by a scalar which the designer may tweak to achieve a trade-off between mean-square-error and peak error performance criteria.

Another way is the faux algebraic techique. The similar structure of the Extended Kalman filter maintains while the stability is achieved by selecting carefully a positive definite solution to a faux algebraic Riccati equation.

C. Invariant extended Kalman filter

The invariant extended Kalman filter (IEKF)[7] is another version of the extended Kalman filter (EKF) for nonlinear systems possessing symmetries (or invariances).It has the advantages of both the EKF and the symmetry-preserving filters that introduced recently.

Instead of using a linear correction term based on a linear output error, the IEKF uses a geometrically adapted correction term based on an invariant output error; in the same way the gain matrix is not updated from a linear state error, but from an invariant state error. The main benefit is that the gain and covariance equations converge to constant values on a much bigger set of trajectories than equilibrium points that is the case for the EKF, which results in a better convergence of the estimation.

IV. Unscented Kalman Filters

As stated in [8], "The extended Kalman filter (EKF) is probably the most widely used estimation algorithm for non-linear systems. However, more than 35 years of experience in the estimation community has shown that is difficult to implement, difficult to tune, and only reliable for systems that are almost linear on the time scale of the updates. Many of these difficulties arise from its use of linearization."

It is true that when the system is highly non-linear, i.e. the function f and h are highly nonlinear, the performance of EKF will be quite poor. This is because the higher orders of Taylor expansion are neglected during the linearization in EKF. In the UKF, we apply an unscented transformation that will maintain higher order information of nonlinear function f and h. It means that the error is smaller so the performance of UKF is better than EKF in highly nonlinear conditions.

A. The Unscented Transformation

The basic idea of unscented transformation(UT) is to pick a set of point properly. This set of point (called sigma point) is the input of the function f, then we can calculate the output set and this set contains most of the information we need in function f. Therefore, we can approximate the non-linear function with high accuracy and efficiency.

In general, a set of sigma points S consists of p+1 vectors and their associated weights $S = \{i = 0, 1, \vdots, p : x^{(i)}, W^{(i)}\}$. The weights $W^{(i)} \in R$ with subject to the condition:

$$\sum_{i=0}^{p} W^{(i)} = 1$$

Then we can compute the transformed sigma points as following:

$$z^{(i)} = f[x^{(i)}]$$

$$\bar{z} = \sum_{i=0}^{p} W^{(i)} z^{i}$$

$$\Sigma_z = \sum_{i=0}^{p} W^{(i)} \{z^{(i)} - \bar{z}\}\{z^{(i)} - \bar{z}\}^T$$

By choosing the set of sigma points properly, we get transformed data with mean and covariance matching the nonlinear function f and minimizing the error of the higher order. Further, if the number of sigma points is larger, we can match higher order like first four moments of a Guassian exactly. And then apply it into simple Kalman filter. Then we will introduce a set of sigma points that mentioned in [8] and talk about how it can be used in detail.

B. The Unscented Kalman Filters

Note that if we choose the sigma point differently, these formula can be changed, so there is no unique way to implement a UKF.

Here is an example of how to use UT in Kalman filter. [9]

1) prediction: Consider a augmented version of mean and covariance of the process noise.

$$s_{k-1|k-1} = \begin{bmatrix} \hat{s}^T_{k-1|k-1} & E[w^T_k] \end{bmatrix}^T$$

$$P_{k-1|k-1} = \begin{bmatrix} P_{k-1|k-1} & 0 \\ 0 & Q_k \end{bmatrix}$$

Then we can construct a set of 2L+1 sigma points

$$s^{(0)}_{k-1|k-1} = s_{k-1|k-1}$$

$$s^{(i)}_{k-1|k-1} = s_{k-1|k-1} + D_i, i = 1, ..., L$$

$$s^{(i)}_{k-1|k-1} = s_{k-1|k-1} - D_{i-L}, i = L+1, ..., 2L$$

$$D = \sqrt{(L+\lambda)P_{k-1|k-1}}$$

where D_i denotes the i^{th} column of the matrix D. And the square root of a matrix $D = \sqrt{A}$ is defined as $A = D \times D$. The matrix square root should be calculated using numerically efficient and stable methods such as the Cholesky decomposition.

Then we can apply the UT metioned before to get $\hat{s}_{k|k-1}$ and corresponding predicted covariance $P_{k|k-1}$ with

$$W^{(0)}_s = \frac{\lambda}{L+\lambda}$$

$$W^{(0)}_c = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta)$$

$$W^{(i)}_s = W^{(i)}_c = \frac{1}{2(L+\lambda)}$$

$$\lambda = \alpha^2(L+\kappa) - L$$

$$s^{(i)}_{k|k-1} = f(s_{k-1|k-1})$$

$$\hat{s}_{k|k-1} = \sum_{i=0}^{2L} W^{(i)}_s s^{(i)}_{k|k-1}$$

$$P_{k|k-1} = \sum_{i=0}^{2L} W^{(i)}_c [s^{(i)}_{k|k-1} - \hat{s}_{k|k-1}][s^{(i)}_{k|k-1} - \hat{s}_{k|k-1}]^T$$

where α, β, κ are the parameters that we can tune.

2) Correction: Similarly, we have the augmented matrix:

$$s_{k|k-1} = \begin{bmatrix} \hat{s}_{k|k-1}^T & E[v_k^T] \end{bmatrix}^T$$

$$P_{k|k-1} = \begin{bmatrix} P_{k|k-1} & 0 \\ 0 & R_k \end{bmatrix}$$

And apply UT to the similar set of 2L+1 sigma points and get corresponding measurement and measurement covariance, $\hat{z}_k$, $P_{z_k z_k}$ and $P_{x_k z_k}$. And finally get the UKF Kalman gain to finish the correction step.

$$K_k = P_{x_k z_k} P_{z_k z_k}^{-1}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - \hat{z_k})$$

$$P_{k|k} = P_{k|k-1} - K_k P_{z_k z_k} K_k^T$$

where z_k is the true measurement which is known.

V. Estimation For Noise Covariance

Note that another condition that need to be satisfied for Kalman filter is that the noise covariance matrix is exactly known. So in practice, if we want to implement a useful Kalman filter, it is necessary to estimate the covariance in an efficient way. One practical approach to do this is the autocovariance least-squares (ALS) technique presented in the [2]. It is unbiased and the covariance of the estimator converges to zero when the the number of samples goes large. Therefore, it is generally a good estimate for the noise covariance.

A. preliminary

Consider a normal linear, time-invariant, discrete-time model:

$$x_{k+1} = Ax_k + Bu_k + Gw_k$$

$$y_k = Cx_k + v_k$$

where $A \in R^{n\times n}, B \in R^{n\times m}, G \in R^{n\times g}, C \in R^{p\times n}$, and w_k and v_k are uncorrelated zero-mean Gaussian noise with covariances Q_w and R_v, respectively. In addition, u_k is the control input which is known. And our state estimator result in

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L[y_k - C\hat{x}_{k|k-1}]$$

where L denotes the estimator gain, note that this gain is general and is not necessarily optimal gain. Particularly, in Kalman filter, this gain is the Kalman gain. Our goal is to estimate the covariances of the noises Q_w, R_v. But before we go, let me introduce some notations that will be used in the following discussion.

1) Kronecker product & Kronecker sum: If $A \in R^{m\times n}$ and $B \in R^{p\times q}$, then the Kronecker product $A \otimes B$ is the mp $\times$ nq block matrix:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

Then we can define Kronecker sum.

$$A \oplus B = A \otimes I_m + I_n \otimes B$$

where $A \in R^{n\times n}, B \in R^{m\times m}$ and I_k denotes the $k \times k$ identity matrix.

2) Autocovariance & autocovariance matrix: The autocovariance is defined as the expectation of the data with some lagged version of itself.

$$r_j = E[y_k y_{k+j}^T]$$

Then we can define the autocovariance matrix (ACM) as

$$R(N) = \begin{bmatrix} r_0 & \cdots & r_{N-1} \\ \vdots & \ddots & \vdots \\ r_{n-1}^T & \cdots & r_0 \end{bmatrix}$$

where N is a user-defined parameter.

3) "vec" Operation: Define "vec" operation as columnwise stacking of a matrix into a vector. If z_k is the k^{th} column of a matrix Z.

$$vec(Z) = Z_s = [z_1^T \cdots z_k^T]^T$$

In the following derivation, we use the s subscript to denote the output of applying the "vec" operation.

4) System Transformation: The state estimate error is $\epsilon_k = x_k - \hat{x}_{k|k-1}$. Then our iterative equation becomes:

$$\epsilon_{k+1} = (A - ALC)\epsilon_k + Gw_k - ALv_k$$

Define $A' = A - ALC$, $G' = [G, -AL]$, $w_k' = [w_k^T, v_k^T]^T$, $y_k' = y_k - C\hat{x}_{k|k-1}$then the equations become:

$$\epsilon_{k+1} = A'\epsilon_k + G'w_k'$$

$$y_k' = C\epsilon_k + v_k$$

5) Assumptions: There are assumptions required for the following derivation

1) (A,C) is detectable, i.e. matrix A and C are known for the following derivation.
2) $A' = A - ALC$ is stable
3) $E[\epsilon_0] = 0, cov(\epsilon_0) = P$ where P is the solution of the Lyapunov equation $P = A'PA' + G'Q_w'G'^T$

B. The ALS estimator

First, we can compute the autocovariance of y_k

$$E[y'_k t'^T_k] = CPC + R_v$$

$$E[y'_{k+j} y'^T_k] = CA'^j PC^T - CA'^{j-1} ALR_v, j \geq 1$$

Now we can calculate the ACM and write it as the following form.

$$R[N] = OPO^T + T[\oplus_{i=1}^N G'Q'_w]T^T + \Phi[\oplus_{i=1}^N R_v]$$
$$+[\oplus_{i=1}^N R_v]\Phi^T + \oplus_{i=1}^N R_v$$

where

$$O = \begin{bmatrix} C \\ CA' \\ \vdots \\ C'^{N-1} \end{bmatrix}, T = \begin{bmatrix} 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 \\ \vdots & \ddots & & \vdots \\ CA'^{N-2} & \cdots & C & 0 \end{bmatrix}$$

$$\Phi = T[\oplus_{i=1}^N (-AL)]$$

Then we apply the "vec" operation to the Lyapunov equation and ACM R(N):

$$P_s = (A' \otimes A')P_s + (G'Q'_w)G'^T$$

$$[R(N)]_s = [(O \otimes O)(I_{n^2} - A' \otimes A')^{-1} +$$

$$(T \otimes T)\phi_{n,N}](G \otimes G)(Q_w)_s$$

$$+(R_v)_s\{[(O\otimes O)(I_{n^2} - A'\otimes A')^{-1} + (T\otimes T)\phi_{n,N}](AL\otimes AL)$$

$$+[\Phi \otimes \Phi + I_{p^2N^2}]\phi_{p,N}\}$$

where $\phi_{p,N} \in R^{(pN)^2 \times p^2}$ is a permutation matrix such that

$$(\oplus_{i=1}^N)_s = \phi_{p,N}(R_v)_s$$

Define

$$D = (O \otimes O)(I_{n^2} - A' \otimes A')^{-1} + (T \otimes T)\phi_{n,N}$$

$$\mathscr{A} = \begin{bmatrix} D(G \otimes G) & D(AL \otimes AL) \\ & +[\Phi \otimes \Phi + I_{p^2N^2}](\phi_{p,N}) \end{bmatrix}$$

$$x = [(Q_w)_s^T (R_v)_s^T]^T, b = R(N)_s$$

Then we get the standard form of least square estimation $\mathscr{A}x = b$. And to estimate the noise covariance $\hat{x}$, we can simply apply the well known $\hat{x} = \mathscr{A}^+\hat{b} = (\mathscr{A}^T\mathscr{A})^{-1}\mathscr{A}^T\hat{b}$ and we can finally get the corresponding covariance $\hat{x}$.

Another issue is how to get the autocovariance matrix R(N). Ideally, we need to calculate all the autocovariance $E[y'_l y'^T_{k+j}]$. In practice, we can approximate the expectation by the formula:

$$\hat{r}_j = \frac{1}{N_d - j} \sum_{i=1}^{N_d - j} y'_i y'^T_{i+j}$$

Then we can use the computed $\hat{r}_j$ to get the estimated $\hat{R}(N)$ and thus get $\hat{b}$.

VI. Conclusion

In this article, we summarize some variants of Kalman filters that can be apply to linear and nonlinear system. Also, we learn a practical way caller autocovariance least-squares method to efficiently estimate the noise covariance which is important in Kalman filter.

References

[1] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering*. American Institute of Aeronautics and Astronautics, Incorporated, 2000.

[2] B. J. Odelson, M. R. Rajamani, and J. B. Rawlings, "A new autocovariance least-squares method for estimating noise covariances," *Automatica*, vol. 42, pp. 303–308, Feb. 2006.

[3] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, pp. 35–45, 1960.

[4] S. M. Kay, *Fundamentals of Statistical Signal Processing*. Prentice Hall, 1993.

[5] Extended Kalman filter, "Extended kalman filter — Wikipedia, the free encyclopedia," 2018, [Online; accessed 18-Jan-2018]. [Online]. Available: https://en.wikipedia.org/wiki/Extended_Kalman_filter

[6] G. A. Einicke and L. B. White, "Robust extended kalman filtering," *IEEE Transactions on Signal Processing*, pp. 2596 – 2599, Sep 1999.

[7] S. . Bonnabel, P. Martin, and E. Salaun, "Invariant extended kalman filter: theory and application to a velocity-aided attitude estimation problem," *48th IEEE Conference on Decision and Control*, p. 12971304, 2009.

[8] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, pp. 401 – 422, 2004.

[9] Kalman filter, "Kalman filter — Wikipedia, the free encyclopedia," 2018, [Online; accessed 18-Jan-2018]. [Online]. Available: https://en.wikipedia.org/wiki/Kalman_filter