Optimal Online Nonlinear/Non-Gaussian State Estimation

Yaodong Cui (29746396)

Abstract—**In many application areas, we need to estimate nonlinear and non-Gaussian latent states in real-time. In this report, we will explore three methods for online nonlinear/non-Gaussian state estimation, namely extended Kalman filter, grid-based filter and particle filter. We will compare and evaluate these three methods and improvements.**

Index Terms—**extended Kalman filter, grid-based filter, particle filter,Bayesian, nonlinear/non-Gaussian**

I. A INTRODUCTION OF OF KALMAN FILTER, GRID BASED METHOD AND PARTICLE FILTER

A. *Kalman filter and extended kalman filter*

Kalman filter is an algorithm that can utilize a set of inaccurate measurements to produce an estimation of the unknown latent variables. This estimation is often better than estimations produced by algorithms that only consider single measurement. The Kalman filter is the optimal estimates for the linear model. It was developed around 1960 by Rudolf E. Kálmán. [1]

However, the Kalman filter assumes the observed model is linear and the posterior density is Gaussian. This is often not the case in reality. Therefore, an extended version of the Kalman filter for the nonlinear model was developed. For extended Kalman filter we define the state space model shown in equation 1 & 2.

$$x(k+1) = f[x(k), u(k)] + G_1 v(k) \tag{1}$$

$$z(k) = g[x(k), u(k)] + G_2 n(k) \tag{2}$$

where x is the state, u is the input, $f(a,b)$ is the system function, v is the process noise, G_1 is the process noise gain, y is the measurement, $g(a,b)$ is the measurement function, G_2 is the measurement noise gain, n is the measurement noise.

Steps of extended Kalman algorithm:

- 1. Initialize the state estimate $\widehat{x_k}$ and error covariance $\widehat{P_k}$. (executed only once)
- 2. Calculate the predicted state estimate.

$$\widehat{x_k} = f[x_{k-1}, u_{k-1}] \tag{3}$$

- 3. Predict the error covariance.

$$\widehat{P_k} = A_k P_{k-1} A_k^T + Q_{k-1} \tag{4}$$

- 4. Calculate the Kalman Filter gain.

$$K_k = \widehat{P_k} C^T [C_k \widehat{P_k} C^T + R_{k-1}]^{-1} \tag{5}$$

- 5. Update the current state estimate.

$$x_k = \widehat{x_k} + K_k(z_k - g[\widehat{x_k}]) \tag{6}$$

- 6. Update the state estimate error covariance.

$$P_k = [I - K_k C]\widehat{P_k} \tag{7}$$

where A and C are Jacobian matrices of function f and g. This process can be understood as the local linearization of the non-linear function around the current estimate. The initial state estimate can be a guessed value, the Kalman algorithm can quickly correct the estimate and stable it around the true value. This is due to the presence of the Kalman gain in the estimate update process. The Kalman gain is a weight that determined the importance of the measurement and state estimate in estimate update.

The extended Kalman filter solved the problem of the nonlinear system through linearizing around the current estimates. However, the extended Kalman filter assumes the noise and posterior density are Gaussian and require to have an accurate system model.

B. *particle filter*

Those restraints can be lifted using a sequential Monte Carlo methods, such as particle filters. Particle filters employ Monte Carlo simulations to approximate posterior density, enabling the tracking of models that have non-linear state functions and non-Gaussian measurements.

Steps of particle filter algorithm:

- 1. Initialize the state estimate $\widehat{x_k}$ and measurements $\widehat{P_k}$. (executed only once)
- 2. Calculate the predicted state estimate, using previous particles.

$$\widehat{x_k^i} = f[x_{k-1}^i, u_{k-1}] + V_k \tag{8}$$

- 3. Calculate the predicted measurements.

$$\widehat{Z_k^i} = g[\widehat{x_k^i}] \tag{9}$$

- 4. Calculate the particle weights using importance sampling and normalize it.

$$W_k^i = W_{k-1}^i \sigma(Z_k^i - \widehat{Z_k^i}) \tag{10}$$

$$W_k^i = W_k^i / sum(W_k^i) \tag{11}$$

- 5. Re-sample particles x_k^i according to the weights and reset weights to 1/N.
- 6. Update the state estimate.

$$X_k = mean(x_k^i) \tag{12}$$

One of the biggest obstacles of particle filters is degeneracy problem. This will cause weight to concentrate on one particle after a few iterations. The degeneracy problem can be solved by adopting suitable importance density and resampling.

C. *grid-based filter*

Grid-based filter is another method that utilize sequential Monte Carlo simulations. Unlike particle filter, grid-based filter do not need down-sampling. However, grid-based filter is more computational ($O(N^2)$) compared to particle filter ($O(N)$). Like the filters mentioned above, grid-based filter can be divided into two stages, the prediction and update. The equations for calculating the weights shown in equation. 13.

$$w_{k|k-1}^i = \sum_{j=1}^{N_s} w_{k-1|k-1}^j p(x_k^i | x_{k-1}^j) \tag{13a}$$

$$W_k^i = W_{k-1}^i \sigma(Z_k^i - \widehat{Z_k^i}) \tag{13b}$$

II. Comparison of different filters

In this section, we will compare the performance of extended Kalman filter (EKF), grid-based filter and particle filter. We will use the root-mean-square error (RMSE) as performance metrics. For the reliable measure of the performance of these algorithms, a number of independent simulations will be conducted and the performance metrics will be averaged. The benchmark problem is shown in equation 14.

$$x_k = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8cos(1.2(k-1)) + v_{k-1} \tag{14a}$$

$$z_k = \frac{x_k^2}{20} + n_k \tag{14b}$$

where v and n are independent zero mean white Gaussian noises.[3]

A. Task 1

The Kalman filter assumes the state space model to be linear and the posterior density to be Gaussian. However, this discrete time system (equation 14) is a nonlinear system. Therefore, under this state-space model the assumptions made by Kalman filter does not hold. Kalman filter will perform poorly.

The probability density function (PDF) of x_1, x_{50} and x_{100} can be obtained using Monte Carlo simulations. The distributions of states at k=1, k=2, k=50 and k=100 during 1000 simulations are shown in Fig. 1.

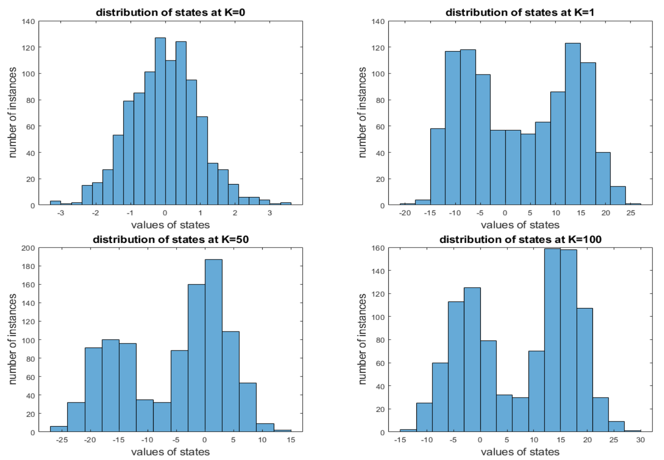

Figure 1: Distribution of state's value of x_0, x_1, x_{50} and x_{100} respectively, during 1000 simulations

From Fig. 1, we notice the distributions of k=2, k=50 and k=100 appears to be Gaussian mixture distributions (GMM). We use MATLAB's $fitgmdist()$ function to approximate the GMMs of x_1, x_{50} and x_{100} and calculate their mean and mixing proportion, as shown in Table. I.

Table I: The mean and mixing proportion of the Gaussian mixture distributions (GMMs) of states x_0, x_1, x_{50} and x_{100}

states	mean	Mixing proportion
x0	0.016	1
x1	[-6.51, 12.55]	[0.52, 0.47]
x2	[0.84, -16.74]	[0.61, 0.39]
x3	[-2.44, 15.74]	[0.45, 0.56]

B. Task 2 & 3

When the actual state information (without process noise) are known, we can use Root Mean Squared Error (RMSE) as performance metric. The average of RMSE from 50 independent simulations are shown in Table II.

Table II: Average of RMSE of three different filters during 50 simulations

Algorithm	Panels in figure. 2	RMSE
Extended Kalman Filter	Panel 1	19.38
Grid-Based filter	Panel 2	10.84
Particle filter	Panel 3	5.32

From table II, it is clear particle filters outperform any other algorithms with an RMSE of 5.32, while grid-based filter followed closely behind it. The extended Kalman filter performs significantly worse than the other two algorithms. The line chart of true state and estimated state of these three filter during 1 simulation are shown in Fig. 2.

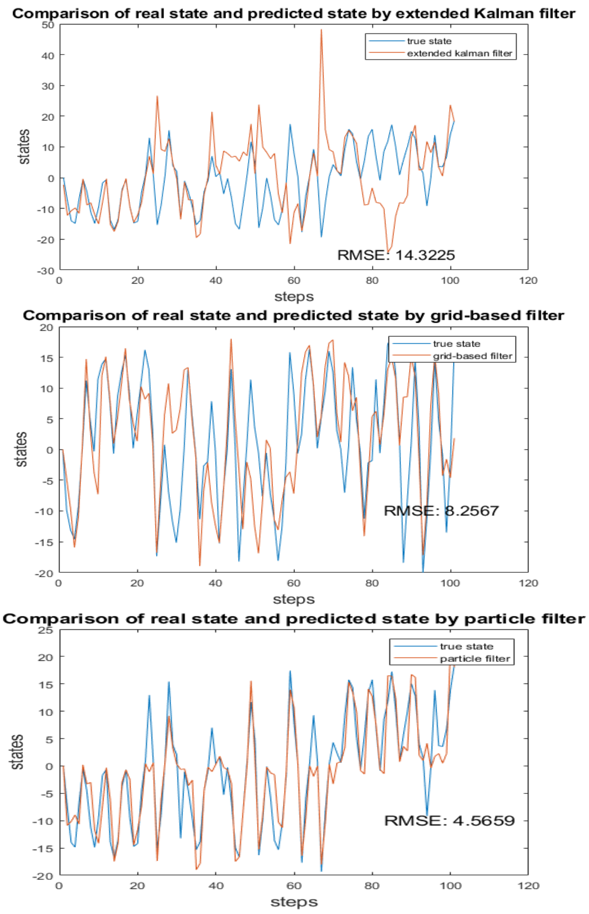

Figure 2: Line charts of the true states (blue line) and estimated state (orange line) of extended Kalman filter, grid-based filter and particle filter in 1 simulation.

There are three panels in this figure. The top one are Line charts of extended Kalman filter. The middle one are Line charts of grid-based filter.The bottom one are Line charts of particle filter.

Particle filters employ sequential Monte Carlo (SMC) simulations to approximate posterior density, which given enough particles, can approximate the non-linear state functions and non-Gaussian measurement. As shown in Task 1, the state-space system is non-linear and possible non-Gaussian.

Although the grid-based filter is based on SMC and is more computational than particle filter, its RMSE is larger than particle filter. This may due to the fixed resolution of grid-based filters. This fixed structure can prevent degeneracy problem but will cause more irrelevant particles contributing to the state estimations.

The poor performance of extended Kalman filter is caused by the approximation of the nonlinear system using local linearization. This linearization error may cause failure of convergence if the states change violently.

C. Task 4

In this section, we will explore methods to improve the performance of the three filtering techniques in Task 2 &3. More specifically, we will compare and evaluate the iterated Kalman filter (IEKF), grid-based filter using different state estimation strategy and sampling importance resampling (SIR) particle filter to their original method (Table. III).

Table III: Average of RMSE of three different filters and their improvement methods during 50 simulations

Algorithm	Figures	RMSE
Extended Kalman Filter	Figure. 3 left	20.01
Iterated Kalman Filter	Figure. 3 right	14.53
Grid-Based filter	Figure. 4 left	10.84
Average Grid-Based filter	Figure. 4 right	9.54
Particle filter	Figure. 5 left	5.32
SIR Particle filter	Figure. 5 right	5.08

From the Fig. 3, we notice that compared to the EKF algorithm the iterated Kalman filter reduce the RMSE from 18.5 to 13.7.

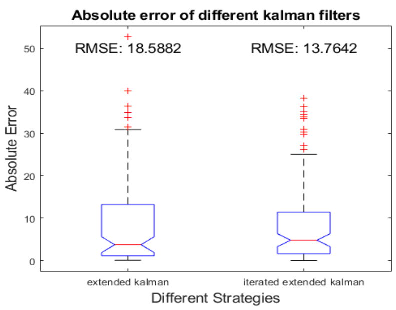

Figure 3: Box plot of the absolute error of extended Kalman filter and iterated extended Kalman filter in 1 simulation.

The difference between EFK and IEFK is that the EKF computes the state estimates as approximate conditional mean, while the IEKF computes the state estimates as a maximum a posteriori (MAP) estimate.[5] The IEKF will update the new states and error covariance through several iterations (equation 15).

$$K_{k,i} = \widehat{P}_k C_{k,i}^T [C_{k,i} \widehat{P}_k C_{k,i}^T + R_{k-1}]^{-1} \tag{15a}$$

$$P_{k,i} = [I - K_{k,i} C_{k,i}] \widehat{P}_k \tag{15b}$$

$$x_{k,i+1} = \widehat{x}_k + K_{k,i}(z_k - g[\widehat{x_{k,i}}] - A_{k,i}(\widehat{x}_k - \widehat{x_{k,i}})) \tag{15c}$$

Where $C_{k,i}$ are Jacobian matrices of function g at $x_{k,i+1}$. This recalculation of the measurement equation around the updated state will reduce the linearization error.

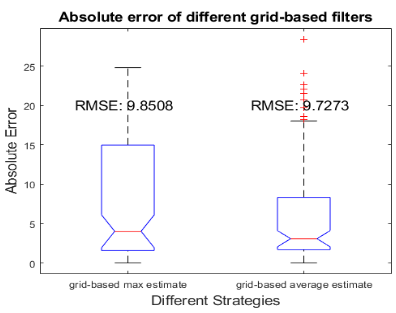

Figure 4: Box plot of the absolute error of grid-based filter and its improvement methods in 1 simulation.

Two methods can be used for calculating the estimates from the grid. The first method is to use the state with the biggest weight. The second is to use the weighted mean of all the estimated states, namely $x_k = W_k^i \widehat{x_k^i}$.The second method can provide moderate improvements, as can be seen in Fig. 4.

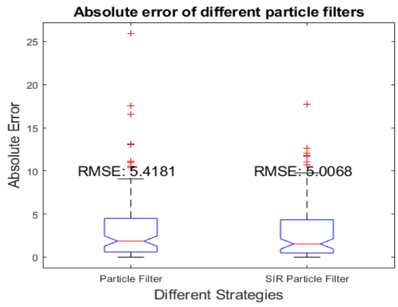

Figure 5: Box plot of the absolute error of SIR particle filter and regularized particle filter in 1 simulation.

The SIR filter is derived from sequential importance sampling, with the importance density chosen as prior and perform re-sample at every steps. Therefore, its weights are calculated using equation. 5. A small improvements of SIR filter can be seen from Fig. 5.

$$W_k^i \propto W_{k-1}^i p(Z_k^i - \widehat{x_k^i}) \tag{16}$$

III. Conclusions

To summarize, we explored and evaluate three methods of on-line nonlinear/non-Gaussian state estimations and their improvements. In situations like nonlinear/non-Gaussian state estimation, particle filter outperformed other two methods with a big margin. However, if the assumptions of Kalman filter or grid-based filter hold, they may be the optimal solutions.

Most of the objectives of this coursework have been met. However, there are some aspects that can be further improved. One of these aspects is to find a good performance metric for evaluating the different filters. In this report, we use RMSE. This is due to RMSE is widely used and can be viewed as benchmarks. However, RMSE can only provide a limited picture of the filter's performance. Other performance metrics, for instance, the Kullback–Leibler (KL) information metric maybe a better alternative. [6]

Although good performances are achieved by certain filters discussed in this report, this does not guarantee their success in reality. In the real world, an accurate system model may not be available, or the noise may change over time. Therefore, when solving a real-world problem, it is very important to confirm assumptions made are true.

References

[1] Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems". Journal of Basic Engineering. 82: 35. doi:10.1115/1.3662552.

[2] Mark Silbert, Thomas Mazzuchib, Shahram Sarkanib, "Comparison of a grid-based filter to a Kalman filter for the state estimation of a maneuvering target," Proceedings Volume 8137, Signal and Data Processing of Small Targets 2011; 81370J (2011); doi: 10.1117/12.892477

[3] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," IEEE Transactions on Signal Processing, vol.50, no.2, pp. 174-188, 2002.

[4] D. Simon, Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches. John Wiley & Sons, 2006.

[5] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software. John Wiley & Sons, 2001.

[6] G. Hendeby and R. Karlsson, "Target tracking performance evaluation A general software environment for filtering," in Proc. IEEE Aerospace Conf., Big Sky, MT, 2007