ELEC6229: Optimal Online Nonlinear/Non-Gaussian State Estimation

Zhikun Zhu, ID: 29356822

***Abstract*— This report summarizes several main techniques and their implementations for nonlinear state estimation: extended Kalman filter(EKF), grid-based filter (GBF), and particle filter(PF). Their performance was benchmarked with a classical nonlinear system and the results have been discussed appropriately.**

I. Introduction and content arrangement

Following is the state model of the benchmark nonlinear system:

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

$$z_k = \frac{x_k^2}{20} + n_k \quad (2)$$

where $v_{k-1} \sim \mathcal{N}(0,10)$ and $n_k \sim \mathcal{N}(0,1)$ are i.i.d Gaussian noise. The initial state x_0 is a Gaussian random variable with zero mean and unit variance. The main task is to estimate the state x for $k = 1,2,...,100$ by the state measurement z_k. We are also required to simulate the state for $k = 1,50,100$ using Monte Carlo method.

Monte Carlo simulation for the state at $k = 1,50,100$ is introduced in section II. And the estimation results for the EKF, GBF and PFare discussed in section III. Their performance and relative merits will be discussed in section IV. Finally, there is a conclusion of this coursework.

II. Monte Carlo simulation for task 1

According to the state model of the system, the states at $k = 1,50,100$ are simulated using Monte Carlo (MC) method. Here I choose $N = 5000$ for the number of runs. The simulation result is shown in Fig. 1 with 50-bins histogram.

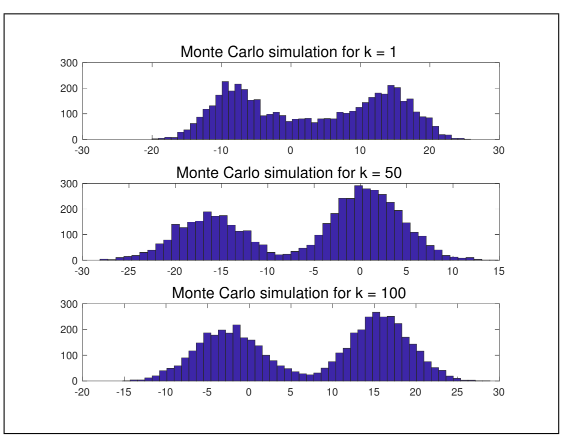

Fig. 1. MC simulation result of state at $k = 1,50,100$.

It is clear that there are two trends for the state at $k = 1,50,100$, which should be taken into consideration when we estimation the state using PF or GBF.

Besides, according to the process and observation equations of Eq. 1 and Eq. 2, respectively, the state estimation would fail if we use traditional Kalman filter because of the high degree of nonlinearity of both processes. Besides, according to [1], the Kalman filter is not sufficient for the non-Gaussian feature of the example. Whereas, [2] argue that Kalman filter also the optimal filter for the non-Gaussian environment since there is no assumption that the noise follows Gaussian distribution in the derivation of Kalman filter.

III. Nonlinear filter estimation results

In this section, the estimation result (solid line) of EKF, GBF, and PF will be compared with the real state (dashed line). Since the noise for both observation and state are continuous, then the possible state of the system is non-discrete, then the approximate GBF is used here to estimate the system state.

According to [1], the performance of the grid-based method is worse than the particle filter since the fixed position of grid points significantly contributes to the error. Such problem can be optimized by dynamic grid arrangement, or more simply, higher resolution by increasing grid points. Here the grid points range from -25 to 25 with a step of 0.1, thus 500 grids points in total. Equivalently, the number of particles for PF is chosen to be 500. However, using massive grid points or particles is time-consuming despite the estimation result may be accurate.

A. *Extended kalman filter and iterated EKF*

Extended Kalman filter is a linear approach to the nonlinear system, it is based on the traditional Kalman filter and the 1-st order expansion of the state equation using Taylor expansion. The difference between EKF and linearized Kalman filter is that the nominal state is updated to be the previous posterior state.

The initialization is crucial for the EKF. [1] suggest to use the covariance matrix of x_0 for the estimation error matrix P and mean of x_0 for initial state, which is 1 and 0 for this specific case.

The update function for the posterior state x_k^+, gain matrix K, and estimation error matrix P are given here:

$$x_k^- = f(x_{k-1}^+, v_{k-1}) \quad (3)$$
$$P_k^- = AP_{K-1}^+ A^T + LQL^T \quad (4)$$
$$A = \frac{\partial f}{\partial x}|_{x_{k-1}^+} \quad C = \frac{\partial h}{\partial x}|_{x_k^-} \quad (5)$$
$$K_k = P_k^- C^T (CP_k^- C^T + MRM^T)^{-1} \quad (6)$$
$$z_k = \frac{x_k^2}{20} + n_k \quad (7)$$
$$P_k^+ = (1 - K_k C)P_k^-(1 - K_k C)^T + K_k R K_k^T \quad (8)$$
$$x_k^+ = x_k^- + K_k(z_k - h(x_k^-)) \quad (9)$$

where $L = \frac{\partial f}{\partial v}|_{x_{k-1}^+} = 1$ and $M = \frac{\partial h}{\partial n}|_{x_k^-}$. The covariance matrix for noise in eq.1 and eq.2 are $Q = 10$ and $R = 1$, respectively.

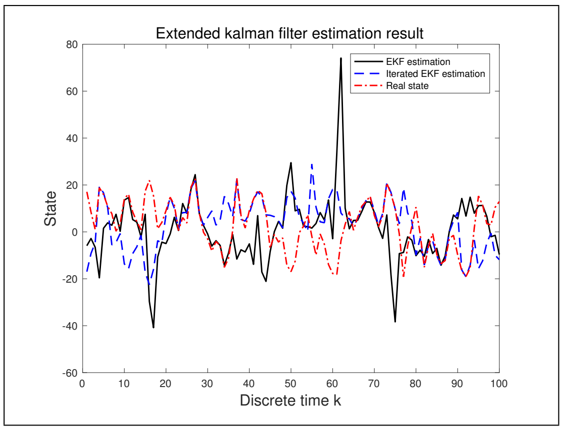

Fig. 2. Extented kalman filter estimation versus real state.

An improved EKF algorithm called iterated EKF is also implemented. The only difference between them is the later algorithm iteratively calculate eq. 5, eq. 6, eq. 8, and eq. 9 for N times (here $N = 100$), which is predefined.

The estimation results of EKF and iterated EKF are shown in Fig. 2. The estimation of EKF basically fails due to the high nonlinearity of the system and this is optimized by the iterated EKF.

B. Grid-based method

The grid-based method is based on the Bayesian estimation theory and estimate state based on the propagation of posterior PDF within the system, which scheme is a prediction and update process. The prediction (prior) of k is driven from the posterior of $k-1$ and the PDF of $p(x_k|x_{k-1}, z_{1:k-1})$. Here, assumption has been made that the prediction is under a Markov process, which indicate that the state x_k is only determined by the previous state x_{k-1}, and then $p(x_k|x_{k-1}, z_{1:k-1}) = p(x_k|x_{k-1})$. At the stage of update, the observation z_k arrives and the update is based on the Bayes' rule:

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{\int p(z_k|x_k)p(x_k|z_{1:k-1})dx_k} \quad (10)$$

Since the state space for the system defined in eq.1 is continuous, then we should discrete it by $\mathbb{S} = \{x_1, x_2, ..., x_{N_s}\}$ grid points, where $N_s = 500$. The prior knowledge of the system are:

1) $x_0 \sim \mathcal{N}(0, 1)$ and $p(x_0|z_0) = p(x_0)$.
2) $p(x_k|x_{k-1}) \sim \mathcal{N}(\frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k), 10)$.
3) $p(z_k|x_k) \sim \mathcal{N}(\frac{x_k^2}{20}, 1)$

Then we can construct a recursive estimator for the probability/ weight w_k^i that at time k, the state is x_i, where $x_i \in \mathbb{S}$. The recursive predict and update equation are:

$$w_{k|k-1}^i = \sum_{j=1}^{N_s} w_{k-1|k-1}^j p(x_k^i|x_{k-1}^j) \quad (11)$$
$$w_{k|k}^i = \frac{w_{k|k-1}^i p(z_k|x_k^i)}{\sum_{j=1}^{N_s} w_{k|k-1}^j p(z_k|x_k^j)} \quad (12)$$

Here we estimate the state for $x_1, x_2, ..., x_100$, and the result posterior PDF for each time k is show in Fig. 3.

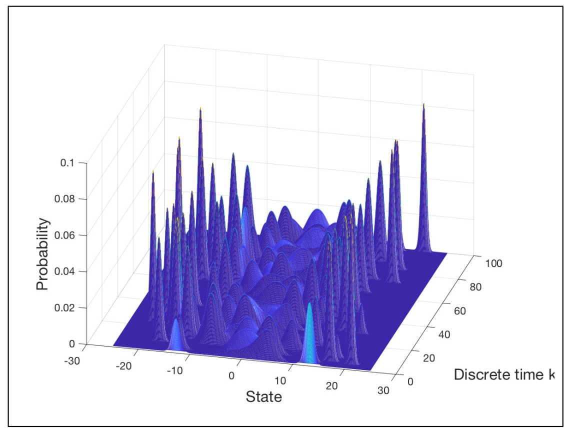

Fig. 3. GBF estimatied PDF for k = 1:100.

The estimation is based on the weighted sum of the posterior PDF:

$$\hat{x}_k = \sum_{i=1}^{N_s} w_{k|k}^i x_i \quad (13)$$

The estimation result is shown in Fig. 4. It is obvious that the performance of GBF is far more better than EKF.

C. Particle filter

Different from theGBF, particle filter utilizes Monte Carlo method to numerically implement the Bayes' recursive estimator. Instead of using grid points to approximate the posterior PDF, here we use MC sampling to approximate it.

Our prior knowledge of the system are same with grid-based method. Then, according to $p(x_0|z_0) = p(x_0)$, we can initially sampling the posterior state $x_{0,i}^+$ for $i = 1, 2, ..., N_s$ based on $p(x_0)$. PF is also a predict-update algorithm which can be devided into three part:

1) Prediction: processing the time propagation to obtain the prior particles of the next time slot based on $x_{k-1,i}^+$:

$$x_{k,i}^- = \frac{x_{k-1,i}^+}{2} + \frac{25x_{k-1,i}^+}{1 + {x_{k-1,i}^+}^2} + 8\cos(1.2k) + v_{k-1} \quad (14)$$

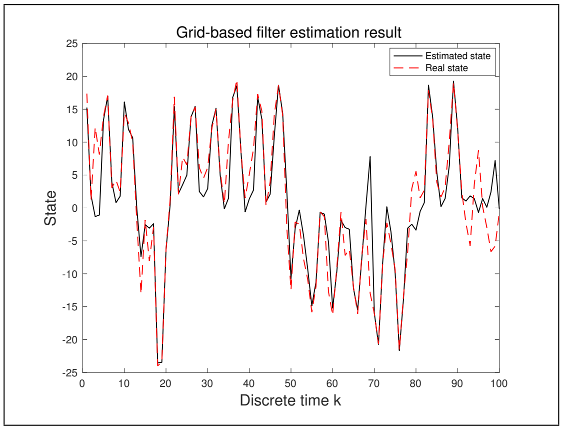

Fig. 4. GBF estimation versus real state.

2) Wighting: weighting $x_{k,i}^-$ with the posterior $p(z_k|x_{k,i}^-)$:

$$q_i = \frac{p(z_k|x_{k,i}^-)}{\sum_j p(z_k|x_{k,j}^-)} \tag{15}$$

3) Update: resampling the prior particles with the probability q:

$$p(x_{k,i}^+ = x_{k,i}^-) = q_i \quad i,j = 1,2,\ldots,N_s \tag{16}$$

The estimation result for PF is shown in Fig. 5, which have $N_s = 500$ particles. It can perform better if we increase N_s but it requires more computation power since computational complexity for PF is $O(N_s)$.

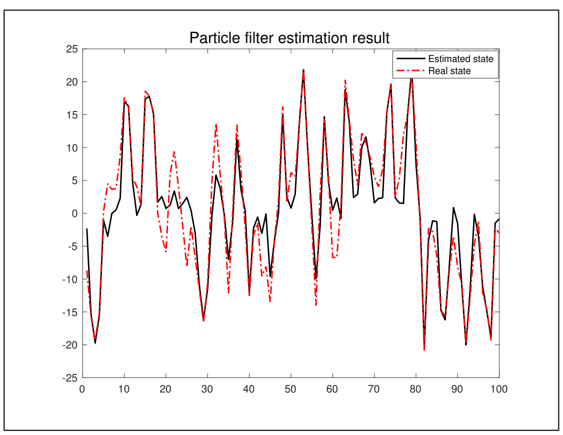

Fig. 5. Particle filter estimation versus real state.

IV. Comparision and discussion

The estimation errors for each iteration among all the filters mentioned before are shown in Fig. 6. According to the figure, we can have an intuitive perception that the performance of PF and GBF are better than EKF and iterated EKF.

Root mean square error (RMSE) is used to further measure their performance, which is shown in Table. 1. Besides, the time consumption is also recorded as a criterion to the performance.

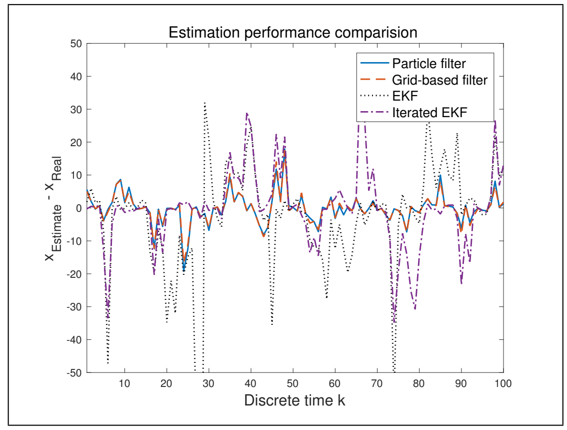

Fig. 6. Error difference with each iteration among three filters.

TABLE I
RMS error among filters been used

Filter	EKF	IEKF	GBF-500	GBF-100	PF-500	PF-100
RMSE	30.471	12.038	4.4304	4.9197	4.8932	5.4522
Time (s)	0.0023	0.0039	22.3033	0.8656	0.8242	0.1518

According to the table, GBF and PF have the minimum MMSE. However, PF is about 30 times faster than that of the GBF with large N_s. Then, generally, PF is assumed to be the best for its accuracy and real-time property. IEKF greatly improved the RMSE of EKF while their time-consumptions are on the same level. IEKF could be under the scenario that the requirement of the accuracy is low but sensitive to the time. The performances of these filters are tested on the same computer with the identical environment.

Besides, increasing N_s just sightly optimized the RMSE due to the impoverishment problems, which could be solved by algorithms like Markov Chain Monte Carlo (MCMC) and Particle Swarm Optimization (PSO). And an appropriate N_s should be considered dealing with practical problems.

In conclusion, the required EKF, GBF, PF, and iterated EKF are all well implemented. GBF based on the propagation of PDF is itself the optimal solution, and the room to optimize its efficiency. There are improved PF for specific problems like auxiliary PF. MCMC and PSO algorithm could be used to solve the impoverishment problem and optimize the performance of PF. The key considerations in nonlinear estimation are time consumption and accuracy. Their relative importance is different for disparate works, where various nonlinear filters could be implemented.

References

[1] M. 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.

[2] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.