2012 International Conference on Image, Vision and Computing (ICIVC 2012)
IPCSIT vol. 50 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V50.29

Improved Extend Kalman Particle Filter Based On Markov Chain Monte Carlo for Nonlinear State Estimation

Huajian Wang +,

Dept. of Communication and Engineering Engineering University of China Armed Police Force Xi'An Shaanxi, China 710086

Abstract. Considering the problem of poor tracking accuracy and particle degradation in the standard particle filter algorithm, a new improved extend kalman particle filter algorithm based on markov chain monte carlo(MCMC) is discussed. The algorithm uses the Extended Kalman filter to generate the proposal distribution that can integrate with the current observation and introduces MCMC technique after the resampling step to figure out the problem of sample impoverishment, so it can obtain a relatively good tracking performance by using fewer particles. Meanwhile, the algorithm is optimized by MCMC sampling method, which makes the particles more diverse. The simulation results show that the improved extend kalman particle filter algorithm based on MCMC solves particle degradation effectively and improves tracking accuracy.

Keywords: Target tracking, Particle filter, Extend Kalman Filter, Markov chain Monte Carlo

1. Introduction

With increasingly improved computer operation ability, in the fields of target tracking [1-3], fault diagnosis [4], financial data analysis [5], computer vision tracking [6-7] and so on, Particle Filter (PF) is paid more attention by the majority of researchers in recent years. The basic idea of Particle Filter is to achieve Bayesian filtering by non-parametric Monte Carlo simulation method[8]. The priori information and the posteriori information are described by sample rather than function. With the number of the sample points increasing, the Monte Carlo simulation properties is approximately equal to the posterior probability density function. Thus, the Particle Filtering estimation approachs the Optimal Bayesian estimation. The Particle Filter is applicable to any state model or measurement model, unrestricted to Nonlinear / non-Gaussian problems.

A common problem of Particle Filter is the degeneracy phenomenon, where after a few iteration, the particles lose of diversity, as all but one of the particles importance weights are very close to '1'. Based on this, it is important to avoid the degeneracy phenomenon by choosing the proposal distribution reason-ably. Many researchers have been studying on how to generate the proposal distribution since Gordon firstly proposed Bootstrap filter in 1993. The Extended Kalman Particle Filter was introduced by Freitas and Doucet [9-10]. It is a linearization technique based on a first-order Taylor series expansion. Since high-order Taylor series are eliminated, the algorithm with relatively low computation causes Linearization error and results in lower filtering accuracy Merwe proposed the Unscented Particle Filter. Although the proposal distribution for Particle Filter generated by UKF shares more with the support of the probability density function in true state, and also greatly improving the filter accuracy, it costs too much execution time and is poor in real-time.

The paper discusses a new improved Particle Filter algorithm, which combines the Markov chain Monte Carlo (MCMC) and Extended Kalman Particle Filter. The algorit-hm uses Extend Kalman filter to generate a

+ Corresponding author. Tel.: +86138 9289 4967.
E-mail address: (whj-winner@ 163.com).

proposal distribution, which can integrate latest observation informa-tion, and then get posterior probability density function that is more in line with true state.. Meanwhile, the algorithm is optimized by MCMC sampling method, which makes the particles more diverse. The simulation results show that the improved Extend Kalman Particle Filter solves particle degradation effectively and improves tracking accuracy.

2. Classical Particle FilterAlgorithm

Consider the general nonlinear dynamic system:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{w}_k \tag{1}$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \tag{2}$$

where:

- $\mathbf{x}_k \in \mathbb{R}^n$ is the base state of the system.
- $\mathbf{z}_k \in \mathbb{R}^p$ is the measurement.
- $\mathbf{w}_k$ is a i.i.d process noise vector.
- $\mathbf{Q}_k$ is the process noise $\mathbf{w}_k$ covariance matrix.
- $\mathbf{v}_k$ is a i.i.d measurement noise vector
- $\mathbf{R}_k$ is the measurement noise $\mathbf{v}_k$ covariance matrix.
- $\mathbf{f}$ is the system dynamics function.
- $\mathbf{h}$ is the measurement function.

The Particle Filter is the approximate numerical solution method, which is based on Monte Carlo and recursive Bayesian estimation methods. The basis of the methods is to use a set of random samples particles and associated importance weights to represent the posterior probability density, and to calculate state estimates. With the sample points increasing, the Monte Carlo simulation properties is approximately equal to the posterior probability density function. Thus, the Particle Filtering estimation approachs the Optimal Bayesian estimation.

In order to develop the details of the algorithm, let $\{\mathbf{x}_k^i, \mathbf{w}_k^i\}$, $i = 1, 2, ..., N$ denote a Random Measure.

where:

- $\mathbf{x}_k^i$ is a samples particle at time k
- $\mathbf{w}_k^i$ is a associated importance weights at time k. The weights are normalized such that $\sum_{i=1}^{N} \mathbf{w}_k^i = 1$.

Then, the posterior density at time k can be approximated as

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} \mathbf{w}_k^i \boldsymbol{\delta}(\mathbf{x}_k - \mathbf{x}_k^i) \tag{3}$$

where:

- $\mathbf{w}_k^i \propto \mathbf{w}_{k-1}^i \dfrac{\mathbf{p}(\mathbf{z}_k / \mathbf{x}_k^i)\mathbf{p}(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i)}{\mathbf{q}(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i, \mathbf{z}_k)}$
- $\mathbf{q}(\mathbf{x}_k^i / \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ is the proposal distribution function.
- $\mathbf{p}(\mathbf{z}_k / \mathbf{x}_k^i)$ is measurement likelihood function.

Below we give a fairly standard algorithm for a Particle Filter that solves this standard filtering problem.

- For $i = 1, ..., N$, Draw the states $\mathbf{x}_0^i$ from the prior $\mathbf{p}(x_0)$,

and set $\mathbf{x}_0^i$ and $\mathbf{p}(x_0)$ initial values

Compute the weights $\mathbf{w}_{k-1}^i = 1/N$

- For each time $k \geq 1$, for $i = 1, ..., N$, evaluate the importance weights up to a normalizing constant:

$$\tilde{\mathbf{w}}_k^i \propto \tilde{\mathbf{w}}_{k-1}^i \frac{\mathbf{p}(\mathbf{z}_k / \tilde{\mathbf{x}}_k^i)\mathbf{p}(\tilde{\mathbf{x}}_k^i / \tilde{\mathbf{x}}_{k-1}^i)}{\mathbf{q}(\tilde{\mathbf{x}}_k^i / \tilde{\mathbf{x}}_{k-1}^i, \mathbf{z}_k)} \tag{4}$$

- For $i = 1, ..., N$, normalize the importance weights:

$$\tilde{\mathbf{w}}_k^i = \tilde{\mathbf{w}}_k^i / \sum_{j=1}^{N} \tilde{\mathbf{w}}_k^j \tag{5}$$

where:

the proportionality constant is such that $\sum_{i=1}^{N} \tilde{\mathbf{w}}_k^i = 1$

- Resampling

Compute $N_{eff} = 1 / \sum_{i=1}^{N} (\tilde{\mathbf{w}}_k^i)^2$.

If $N_{eff} < N_{th}$ then resample the particles: duplicates particle with large weights and suppress particles with low weights. The resulting particles are denoted $\tilde{\mathbf{x}}_{\mathbf{k}}^{i}$ and their weights $\tilde{\mathbf{w}}_{k}^{i} = 1/N$

Otherwise, rename particles, that is, set $\mathbf{x}_{\mathbf{k}}^{i} \rightarrow \tilde{\mathbf{x}}_{\mathbf{k}}^{i}$

- For $k = k+1$ go back.

A common problem of Particle Filter is the degeneracy phenomenon. Because True distribution $p(\mathbf{x}_{\mathbf{k}}^{i} / \mathbf{x}_{\mathbf{k-1}}^{i}, \mathbf{z}_{\mathbf{k}})$

usually couldn't be get. In Practice, the Optimal proposal distribution function $\mathbf{q}(\mathbf{x}_{\mathbf{k}}^{i} / \mathbf{x}_{\mathbf{k-1}}^{i}, \mathbf{z}_{\mathbf{k}})$ is replaced by State transfer function $\mathbf{p}(\mathbf{x}_{\mathbf{k}}^{i} \mid \mathbf{x}_{\mathbf{k-1}}^{i})$, that can't integrate latest observation information. The covariance of the importance weights can only increase over time, all but one particle will have negligible weight. This degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the approximation to $\mathbf{p}(\mathbf{x}_{\mathbf{k}} / \mathbf{z}_{\mathbf{1:k}})$ is almost zero. So that the particles can't contribute to the actual posterior probability distribution. There are two methods to deal with the problem: choosing reasonable the proposal distribution function and resampling method.

3. Extended Kalman Particle Filter

Freitas and Doucet used Extend Kalman filter to generate a proposal distribution, which can integrate latest observation information, which makes the resulting particles closer to truth sample, real-time updates the weight of each particle, so as to improve the filtering effect.

Below we give a standard algorithm for the Extend Kalman Particle Filter that solves this standard filtering problem.

- For $i = 1,...,N$, Draw the states $\mathbf{x}_{0}^{i}$ from the prior $\mathbf{p}(x_0)$, and set $\mathbf{x}_{0}^{i}$ and $\mathbf{p}(x_0)$ initial values

Compute the weights $\mathbf{w}_{\mathbf{k-1}}^{i} = 1/N$

- For each time $k \geq 1$, for each $i = 1,...,N$, update the particle with EKF

$$\hat{\mathbf{x}}_{\mathbf{k/k-1}}^{i} = \mathbf{f}(\mathbf{x}_{\mathbf{k-1}}^{i}, \mathbf{0}) \tag{6}$$

$$\mathbf{P}_{\mathbf{k/k-1}}^{i} = \mathbf{F}_{\mathbf{k}} \mathbf{P}_{\mathbf{k-1}}^{i} \mathbf{F}_{\mathbf{k}}^{T} + \mathbf{Q}_{\mathbf{k-1}} \tag{7}$$

$$\mathbf{S}_{\mathbf{k}}^{i} = \mathbf{H}_{\mathbf{k}} \mathbf{P}_{\mathbf{k/k-1}}^{i} \mathbf{H}_{\mathbf{k}}^{T} + \mathbf{R}_{\mathbf{k}} \tag{8}$$

$$\mathbf{K}_{\mathbf{k}}^{i} = \mathbf{P}_{\mathbf{k/k-1}}^{i} \mathbf{H}_{\mathbf{k}}^{T} (\mathbf{S}_{\mathbf{k}}^{i})^{-1} \tag{9}$$

$$\bar{\mathbf{x}}_{\mathbf{k}}^{i} = \hat{\mathbf{x}}_{\mathbf{k/k-1}}^{i} + \mathbf{K}_{\mathbf{k}}^{i} (\mathbf{y}_{\mathbf{k}} - \mathbf{h}(\hat{\mathbf{x}}_{\mathbf{k/k-1}}^{i}, \mathbf{0})) \tag{10}$$

$$\hat{\mathbf{P}}_{\mathbf{k}}^{i} = (\mathbf{I} - \mathbf{K}_{\mathbf{k}}^{i} \mathbf{H}_{\mathbf{k}}) \mathbf{P}_{\mathbf{k/k-1}}^{i} \tag{11}$$

where:

F_k is the Jacobian matrices of the process model.

$$\mathbf{F}_{\mathbf{k}} = \frac{\partial f(x)}{\partial x} |_{\mathbf{x}=\mathbf{x}_{\mathbf{k-1}}} \tag{12}$$

H_k is the Jacobian matrices of the measurement model.

$$\mathbf{H}_{\mathbf{k}} = \frac{\partial h(x)}{\partial x} |_{\mathbf{x}=\hat{\mathbf{x}}_{\mathbf{k/k-1}}} \tag{13}$$

Sample particles:

$$\mathbf{x}_{\mathbf{k}}^{i} \sim \mathbf{q}(\mathbf{x}_{\mathbf{k}}^{i} / \mathbf{x}_{\mathbf{0:k-1}}^{i}, \mathbf{z}_{\mathbf{1:k}}) = \mathbf{N}(\mathbf{x}_{\mathbf{k}}; \bar{\mathbf{x}}_{\mathbf{k}}^{i}, \hat{\mathbf{P}}_{\mathbf{k}}^{i}) \tag{14}$$

- Importance weights

evaluate the importance weights up to a normalizing constant:

$$\tilde{\mathbf{w}}_{\mathbf{k}}^{i} \propto \tilde{\mathbf{w}}_{\mathbf{k-1}}^{i} \frac{\mathbf{p}(\mathbf{z}_{\mathbf{k}} / \tilde{\mathbf{x}}_{\mathbf{k}}^{i}) \mathbf{p}(\tilde{\mathbf{x}}_{\mathbf{k}}^{i} / \tilde{\mathbf{x}}_{\mathbf{k-1}}^{i})}{\mathbf{q}(\tilde{\mathbf{x}}_{\mathbf{k}}^{i} / \tilde{\mathbf{x}}_{\mathbf{k-1}}^{i}, \mathbf{z}_{\mathbf{k}})} \tag{15}$$

normalize the importance weights:

$$\tilde{\mathbf{w}}_{\mathbf{k}}^{i} = \tilde{\mathbf{w}}_{\mathbf{k}}^{i} / \sum_{j=1}^{N} \tilde{\mathbf{w}}_{\mathbf{k}}^{j} \tag{16}$$

where:

the proportionality constant is such that $\sum_{i=1}^{N} \tilde{\mathbf{w}}_{\mathbf{k}}^{i} = 1$

- Resampling

Compute $N_{eff} = 1 / \sum_{i=1}^{N} (\tilde{\mathbf{w}}_{\mathbf{k}}^i)^2$.

If $N_{eff} < N_{th}$ then resample the particles: duplicates particle with large weights and suppress particles with low weights. The resulting particles are denoted $\tilde{\mathbf{x}}_{\mathbf{k}}^i$ and their weights $\tilde{\mathbf{w}}_k^i = 1/N$

Otherwise, rename particles, that is, set $\mathbf{x}_{\mathbf{k}}^i \to \tilde{\mathbf{x}}_{\mathbf{k}}^i$

- For $k = k+1$ go back.

4. Target Tracking Algorithm Based on Improved Extend Kalman Particle Filter

4.1 MCMC Steps

The resampling step reduces the effects of the degener-acy problem. The basic idea of resampling step is that the particles which have high weights $\mathbf{w}_k^i$ are repeated many times, and which have lower weights $\mathbf{w}_k^i$ are removed. This leads to a loss of diversity among the particles as the resultant sample will contain many repeated points, and so as to loss of the diversity of particle, reducing the estimation, leading to particle sampling impoverishment.

So another approach is to use MCMC steps for each particle, to make the particles more diversification. The basic idea of MCMC steps is to construct a Markov chain, which uses Markov kernel function $\kappa(x_{0:k} / \tilde{x}_{0:k})$ to generate a set new sample particle $\{x^{*i}_{k}, i = 1, 2, ..., N\}$ instead of a set re- sample particle $\left\{\tilde{x}_k^i, i = 1, 2, ..., N\right\}$. Under the condition for $\int \kappa(x_{0:k} / \tilde{x}_{0:k}) p(\tilde{x}_{0:k} / z_{1:k}) d\tilde{x}_{0:k} = p(x_{0:k} / z_{1:k})$, new particles $x^{*i}_{0:k} = \{\tilde{x}^{\,i}_{0:k-1}, x^{*i}_{k}\}$ are still subject to the $p(x_{0:k} / z_{1:k})$ distribution.

The paper uses Metropolis-Hasting Algorithm to implement MCMC steps.

- Sample $u \sim U_{[0,1]}$, $U_{[0,1]}$ is uniformly distribution in the interval $[0,1]$.
- Sample from the proposal distribution $x^{*i}_{k} \sim p(x_k^i \mid x_{k-1}^i)$.
- Calculate the acceptance ratio

if $u \le min\left\{1, \dfrac{p(x_k^i \mid x^{*i}_{k})}{p(x_k^i \mid \tilde{x}_k^i)}\right\}$, set $x^{*i}_{0:k} = \{\tilde{x}^{\,i}_{0:k-1}, x^{*i}_{k}\}$,

else $x^{*i}_{0:k} = \tilde{x}_{0:k}^i$.

4.2 Extend Kalman Particle Filter Based on MCMC steps

The basic idea of Extend Kalman Particle Filter Based on MCMC steps is to uses Extend Kalman filter to generate a proposal distribution, which can update particles' mean and variance with $(\overline{x}_k^i, \hat{P}_k^i)$. Meanwhile, the algorithm is optimiz-ed by MCMC sampling method after Extend Kalman Particle Filter steps, which makes the particles more diversification.

The improved Extend Kalman Particle Filter implement-ation is as follows.

- Generate to a threshold ε, $\varepsilon \sim U[0,1]$.
- For each time $k \ge 1$, for each $i = 1, ..., N$, update the particle with EKF

$$\tilde{\mathbf{x}}_{\mathbf{k/k-1}}^i = \mathbf{f}(\mathbf{x}_{\mathbf{k-1}}^i, \mathbf{0}) \tag{17}$$

$$\tilde{\mathbf{P}}_{\mathbf{k/k-1}}^i = \mathbf{F}_{\mathbf{k}} \mathbf{P}_{\mathbf{k-1}}^i \mathbf{F}_{\mathbf{k}}^T + \mathbf{Q}_{\mathbf{k-1}} \tag{18}$$

$$\mathbf{S}_{\mathbf{k}}^i = \mathbf{H}_{\mathbf{k}} \tilde{\mathbf{P}}_{\mathbf{k/k-1}}^i \mathbf{H}_{\mathbf{k}}^T + \mathbf{R}_{\mathbf{k}} \tag{19}$$

$$\mathbf{K}_{\mathbf{k}}^i = \tilde{\mathbf{P}}_{\mathbf{k/k-1}}^i \mathbf{H}_{\mathbf{k}}^T (\mathbf{S}_{\mathbf{k}}^i)^{-1} \tag{20}$$

$$\overline{\mathbf{x}}_{\mathbf{k}}^{*i} = \tilde{\mathbf{x}}_{\mathbf{k/k-1}}^i + \mathbf{K}_{\mathbf{k}}^i (\mathbf{y}_{\mathbf{k}} - \mathbf{h}(\tilde{\mathbf{x}}_{\mathbf{k/k-1}}^i, \mathbf{0})) \tag{21}$$

$$\hat{\mathbf{P}}_{\mathbf{k}}^{*i} = (\mathbf{I} - \mathbf{K}_{\mathbf{k}}^i \mathbf{H}_{\mathbf{k}}) \tilde{\mathbf{P}}_{\mathbf{k/k-1}}^i \tag{22}$$

where:

F_k is the Jacobian matrices of the process model.

$$\mathbf{F}_{\mathbf{k}} = \frac{\partial f(x)}{\partial x} |_{\mathbf{x}=\mathbf{x}_{\mathbf{k-1}}} \tag{23}$$

H_k is the Jacobian matrices of the measurement model.

$$\mathbf{H}_{\mathbf{k}} = \frac{\partial h(x)}{\partial x} |_{\mathbf{x}=\hat{\mathbf{x}}_{\mathbf{k/k-1}}} \tag{24}$$

- Resampling

$$\mathbf{x}_k^{*i} \sim \mathbf{q}(\mathbf{x}_k^i / \mathbf{x}_{0:k-1}^{*i}, \mathbf{z}_{1:k}) = \mathbf{N}(\mathbf{x}_{k;} \overline{\mathbf{x}}_k^{*i}, \hat{\mathbf{P}}_k^{*i}) \quad (25)$$

- Calculate the acceptance ratio

$$if\, u \le min\left\{1, \frac{\mathbf{p}(\mathbf{z}_k / \mathbf{x}_k^{*i})\mathbf{p}(\mathbf{x}_k^{*i} / \tilde{\mathbf{x}}_{k-1}^i)\mathbf{q}(\tilde{\mathbf{x}}_k / \tilde{\mathbf{x}}_{0:k-1}^i, \mathbf{z}_{1:k})}{\mathbf{p}(\mathbf{z}_k / \tilde{\mathbf{x}}_k^i)\mathbf{p}(\tilde{\mathbf{x}}_k^i / \tilde{\mathbf{x}}_{k-1}^i)\mathbf{q}(\mathbf{x}^{*i}_k / \tilde{\mathbf{x}}_{0:k-1}^i, \mathbf{z}_{1:k})}\right\}, set\, \mathbf{x}^{*i}_{0:k} = \{\tilde{\mathbf{x}}^i_{0:k-1}, \mathbf{x}^{*i}_k\},\ \mathbf{P}^{*i}_{0:k} = \{\mathbf{P}^i_{0:k-1}, \mathbf{P}^{*i}_k\},$$

$$else\, \mathbf{x}^{*i}_{0:k} = \tilde{\mathbf{x}}^i_{0:k},\ \mathbf{P}^{*i}_{0:k} = \mathbf{P}^i_{0:k}$$

- Repeat the above steps

5. Simulation Results

In this section, estimation performance and computation-al cost of comparisons between PF, EKF-PF and EKF-PF-MCMC are made through simulated examples on the same problem.

The system models were taken as follows:

$$x_k = 1 + \sin(5\pi k) + 0.5 x_{k-1} + w_{k-1} \quad (26)$$

$$z_k = \begin{cases} 0.5x_k - 2 + v_k & k > 30 \\ 0.2x_k^2 + v_k & k \le 30 \end{cases} \quad (27)$$

where:

w_{k-1} is a Gamma $\varsigma_a(3,2)$ random variable modeling the process noise. The measurement noise v_k is drawn from a Gaussian distribution $N(0, 0.0001)$. The true initialization state of the target is $x_0 = 1.0$. The true initialization variance of the target is $P_0 = 0.75$. In every run, the total simulation time is 100 steps, $t = 1s$ and the number of particles N is 100, 200 or 400. The output of the algorithm is the mean of samples set that can be computed

$$\hat{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_k^i \quad (28)$$

The root mean square errors of each run is defined as

$$RMSE = \left(\frac{1}{T}\sum_{k=1}^{T}(\hat{x}_k - x_k)^2\right)^{1/2} \quad (29)$$

The estimated state mean and covariance of the RESE and the execution time are obtained in Table I.

As shown in Table I, apparently the proposed EKF-PF-MCMC algorithm outperforms the PF and EKF-PF. Specially, EKF-PF-MCMC achieves the better estimated accuracy with 200 sampling particles rather than PF with 400 sampling particles, which is close to estimate accuracy of EKF-PF with 400 sampling particles. The simulation results show that the EKF-PF-MCMC algorithm can achieve better filtering effect and improve real-time with less number of particles. The estimate results of different filters are taken as follows Fig.1 ~ Fig.3.

Table 1. The Rmse and execution time at different particles number

Algorithm	Particles Number	RMSE Mean	RMSE Variance	Execution Time
PF	100	0.028645	0.0073965	1.3780
	200	0.015278	0.0030354	2.4246
	400	0.004896	0.0010811	4.9783
EKF-PF	100	0.015442	0.0020565	5.1085
	200	0.006897	0.0006524	9.6718
	400	0.004377	0.0002322	20.4087
EKF-PF-MCMC	100	0.009090	0.0010950	10.5864
	200	0.004755	0.0004571	19.9857
	400	0.002796	0.0001816	42.1868

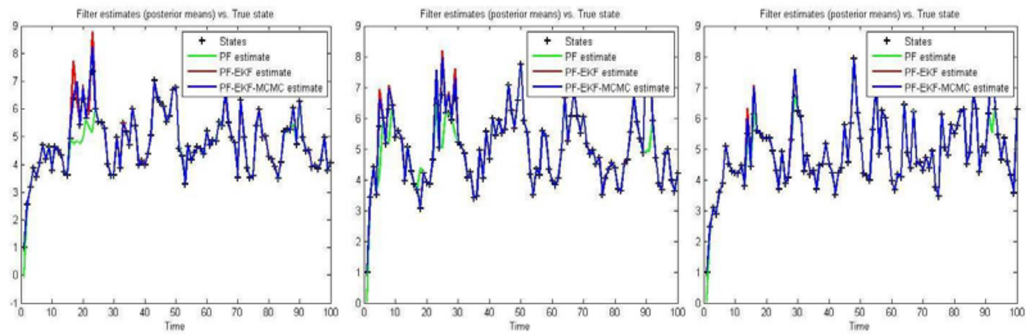

Fig. 1.Estimate results of different filters vs. true state (a) N=100; (b) N=200, (c) N=400

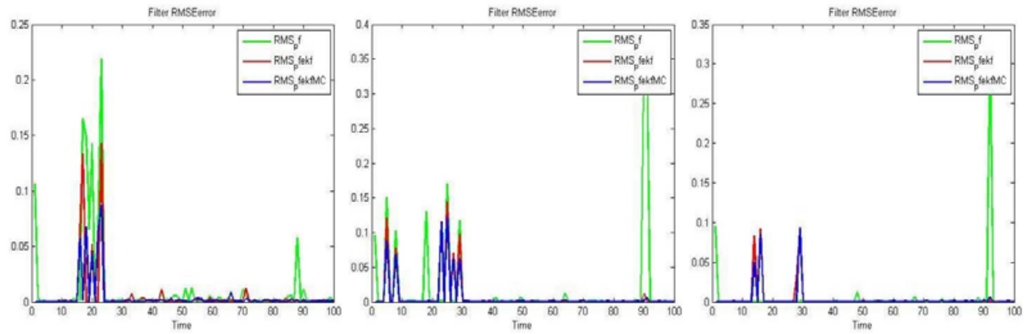

Fig. 2.The RMSE results of different nonliear filters (a) N=100; (b) N=200, (c) N=400

From Fig.4 ~ Fig.6, although EKF-PF-MCMC also brings a certain error at the initial filtering stage, it can adjust to converge quickly to true state based on observed values. However, PF has not considered the observation information, so filtering has not significantly improved. With the number of particles increasing, the situation is getting more serious.

6. Conclusion

In this paper we presented a EKF-PF-MCMC algorithm to estimate the state of the nonlinear and non-Gaussian system. The algorithm uses the Extended Kalman filter to generate the proposal distribution that can integrate with the current observation and introduces MCMC technique after the resampling step to figure out the problem of sample impoverishment, so it can obtain a relatively good tracking performance by using fewer particles. However, the algori-thm also has some deficiencies. More steps in MCMC leads to high computation and poor real-time compared to classic PF. In the future, we will pay more attention to improve the accuracy and reduce the execution time of the method.

7. Acknowledgements

This research work was supported by the foundation fund of China Armed Police Force No. WJY201114.

8. References

[1] Gordon N, Salmond D. Novel approach to non-linear and non-Gaussian Bayesian state estimation. *Proc of Institute Electric Engineering*, 1993, 140(2): 107-113.

[2] Buzzi S, Lops M, Venturino L, et al. Track-before-detect procedures in a multi-target environment. *IEEE Transactions on Aerospace and Electronic Systems,* 2008, 44(3): 1135-1150.

[3] Petar M.Djuric, Monica F.Bugallo. Improved Target Tracking with Particle Filtering. *Aerospace conference, 2009 IEE*E, 2009, Page(s): 1341-1348.

[4] Kadirkamanathan V, Li P, Jaward M H, et al, Particle filtering-based fault detection in non-linear stochastic system. *International Journal of Systems Science*,2002, 33(4): 259-265.

[5] Zhang Gaoyu, Li Qiongfei, Luo Qing, Zhou Zhizhao. High frequency financial time series forecasting via Particle Filtering. *Information Management, Innovation Manage-ment and Industrial Engineering*, 2009 Volume: 4, 2009,62- 65.

[6] QieongM/ang, JilinLiu, ZhigangWu. Objeet Tracking Using Genetie Evolution Based Kemel Particle Filter. *Leeture Notes in Computer Seience*, VOI.4040, 2006:466-473.

[7] Junlan Yang, Schonfeld, D., Mohamed, M. Robust Video Stabilization Based on Particle Filter Tracking of Projected Camera Motion. *Circuits and Systems for Video Technology, IEEE*, 2009 , Page(s): 945- 954.

[8] Doucet A, Gordon N. Sequential Monte Carlo methods in practice. *New York: Springer*- Verlag, 2001.

[9] De Freitas F G, Bayesian methods for LONDON neural networks. *Cambridge University*,1999.

[10] Doucet A, Godsilli S J, ABDRUEU C. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing (S0960-3174)* , 2000, 10(3): 197-208.

[11] van der Merwe R, Douce A, de Freitas N, et al. The unscented Particle Filter. *Technical Report of the Cambridge University Engineering Department CUED/FINFENG/TR, 380. England: Cambridge University Press*,2000·1-45.

[12] Hastings W K. Monte Carlo sampling methods using Markov chains and their Applications. *Biometrika* , 1970 , 57: 97-109.