Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)

Cubature Kalman Particle Filters

Shan Ganlin, Chen Hai, Ji Bing, Zhang Kai
Department of Optics and Electronics Engineering Shijiazhuang Mechanical Engineering College
Shijiazhuang, China
wodezhangkai@126.com

***Abstract*—To resolve the tracking problem of nonlinear/non-Gaussian systems effectively, this paper proposes a novel combination of the cubature kalman filter(CKF) with the particle filters(PF), which is called cubature kalman particle filters(CPF). In this algorithm, CKF is used to generate the importance density function for particle filter. It linearizes the nonlinear functions using statistical linear regression method through a set of Gaussian cubature points. It need not compute the Jacobian matrix and is easy to be implemented. Moreover, it makes efficient use of the latest observation information into system state transition density, thus greatly improving the filter performance. The simulation results are compared against the widely used unscented particle filter(UPF), and have demonstrated that CPF has higher estimation accuracy and less computational load.**

Keywords- particle filter; cubature kalman filter; importance density function

I. INTRODUCTION

Over the past years, PF has been widely applied with great success in solving nonlinear/non-Gaussian filtering problem, such as vision tracking, robot localization, image processing, data fusion, navigating and so on. PF is a technique for implementing a recursive Bayesian filter by Monte Carlo simulations. The basic idea of PF[1] is to represent the required probability density function (PDF) by a set of random samples with associated weights and to compute estimations based on these samples and weights. There are mainly three problems in PF. One common problem is the particle degeneracy, that is to say, after a few iterations, only few particles keep high weights and the estimation may become unreliable. Fortunately, importance resampling has been developed by [2] to overcome this drawback. The objective of resampling is to eliminate samples with low importance weights and multiply samples with high importance weights. Several resampling methods have been developed[2], such as multinomial resampling, residual resampling and systematic resampling. The choice of the important density function is the second problem in PF, which is generally hard to design. The popular choice is the transition prior density as it simplies many calculations, but it doesn't adopt the latest measurements, the proposal distribution is very inefficient sometimes, and the estimation result is poor. To overcome this drawback, the methods of local linearization are used to generate the proposal importance distribution, it's shown that proposals based on EKF and UKF have better performance. These filters are known as extended kalman particle filter(EPF) and unscented kalman particle filter(UPF)[3,4,5]. Subsequently, we also get the Gaussian Hermite particle filter(GHPF)[6], quadrature kalman particle filter(QPF)[7], improved unscented kalman particle filter(IUPF)[8] respectively by using GHF, QKF, IUKF. Finally, instead of analytical solution or numerical approximation of a given nonlinear and/or non-Gaussian problem, it performs a considerable amount of computations in order to approximate the posterior PDF, the central idea is to represent the required PDF by a set of random samples with associated weights. In this paper, we propose a new particle filter, the cubature kalman particle filter(CPF), which uses the cubature kalman filter(CKF)[9] to generate the important density function. Because CKF has superior performance than EKF and UKF[10], so the new CPF approximates more accuracy.

II. PARTICLE FILTER

Nonlinear discrete time dynamic system can be modeled by

$$x_{k+1} = f_k(x_k) + w_k \tag{1}$$

$$z_k = h_k(x_k) + e_k \tag{2}$$

where $x_k \in R^{n\times 1}$ is the state of the system, $w_k \in R^{q\times 1}$ is the process noise caused by disturbances and modeling errors, $z_k \in R^{m\times 1}$ is the observation vector, and $e_k \in R^{r\times 1}$ is additive measurements noise. f_k and h_k are vector–valued functions. Let $X_{0:k} = \{x_0, x_1, \cdots, x_k\}$ and $z_{1:k} = \{z_0, z_1, \cdots, z_k\}$ be stacked vector of states and observations up to time step k . The Bayesian estimation is to estimate the posterior condition PDF $p(x_{k+1} \mid z_{1:k+1})$ recursively in the discrete time domain using prediction and updating procedures as follows:

Prediction: Using (1) to obtain the predictive PDF of x_{k+1} via the Chapman-Kolmogorov equation for a known posterior PDF $p(x_k \mid z_{1:k})$ at time k :

$$p(x_{k+1} \mid z_{1:k}) = \int p(x_{k+1} \mid x_k)\, p(x_k \mid z_{1:k}) dx_k \tag{3}$$

Updating: At time $k+1$, the predictive PDF (3) is updated by the information contained in the measurement z_{k+1} via Bayesian formula:

$$p(x_{k+1} \mid z_{1:k+1}) = \frac{p(z_{k+1} \mid x_{k+1})\, p(x_{k+1} \mid z_{1:k})}{p(z_{k+1} \mid z_{1:k})} \tag{4}$$

where the normalizing constant is calculated as follows:

$$p(z_{k+1} \mid z_{1:k}) = \int p(z_{k+1} \mid x_{k+1})\, p(x_{k+1} \mid z_{1:k}) dx_{k+1}$$

which depends on the likelihood function of $p(z_{k+1} \mid x_{k+1})$.

A. Generic particle filter

Particle filter uses sequential Monte Carlo methods to approximate the integrals found in the recursive Bayesian filtering method. Specifically, particle filters use a set of weighted particles that are samples drawn from the posterior distribution in order to approximate the required integrals as discrete sums. Given a set of N_p random samples, $\{x_{0:k}^{(i)}, w_{0:k}^{(i)} : i = 1, \ldots, N_p\}$ the posterior distribution is approximated as

$$\hat{p}(x_k \mid z_{1:k}) = \frac{1}{N_p} \sum_{i=1}^{N_p} w_k^{(i)} \delta(x_k - x_k^{(i)}) \tag{5}$$

In practice, the particle set is finite and the major drawback of this algorithm is the degeneracy of the particle set. To avoid the problem, resampling is introduced to work with the sequential importance sampling(SIS). Combining SIS with a resampling method produces the generic particle filter. The implementation of the PF consists of three important operations: 1) generation of particles(sampling step). 2) computation of the particle weights (importance step). 3) resampling. An outline for the generic particle filter[11] is given in Table Ⅰ.

Table I. The generic particle filter[8]

- **Initialization**: at time $k = 0$.

1. For $i = 1, \cdots, N$, sampling from the prior $x_0^{(i)} \sim p(x_0)$.
2. For $i = 1, \cdots, N$,
   calculate $w_0^{(i)} = p(z_1 \mid x_0^{(i)})$,
   calculate the total weight $w_T = \sum_{i=1}^{N} w_0^{(i)}$,
   normalize $w_0^{(i)} = w_0^{(i)} / w_T$.

- **Prediction and Update**: For each time $k \geq 1$.

1. For $i = 1, \cdots, N$, Sample $x_k^{(i)} \sim q(x_k \mid x_{0:k-1}^{(i)}, z_{1:k})$, calculate the importance weights:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(z_k \mid x_k^{(i)}) p(x_k^{(i)} \mid x_{k-1}^{(i)})}{q(x_k^{(i)} \mid x_{0:k-1}^{(i)}, z_{1:k})}$$

2. Calculate the total weight $w_T = \sum_{i=1}^{N} w_k^{(i)}$.
3. For $i = 1, \cdots, N$, normalize $w_k^{(i)} = w_k^{(i)} / w_T$.
4. If $(N_{eff} < N_{th})$, set the weights $w_k^{(i)} = 1/N$, apply resampling algorithm

$$[\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}] = \mathrm{Re}\,sample[\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}].$$

III. THE CUBATURE KALMAN PARTICLE FLTER

The choice of proposal or importance distribution is a critical design issues in implementing PF. The performance of PF depends on the proposal importance function heavily. The optimal proposal importance distribution is given by $q(x_k \mid x_{0:k-1}, z_{1:k}) = p(x_k \mid x_{0:k-1}, z_{1:k})$ and fully exploits the information in both $x_{0:k-1}$ and $z_{1:k}$ [12]. However, it is impossible to sample from this distribution due to the unknown distribution $p(x_k \mid x_{0:k-1}, z_{1:k})$. The most popular choice of proposal function is the transmission prior function $q(x_k \mid x_{0:k-1}, z_{1:k}) = p(x_k \mid x_{k-1})$ due to its convenience. But since this way has not incorporate the latest information $z_{1:k}$, the performance depends heavily on the variance of observation noise. The third choice is to use local linearization to generate the proposal importance distribution. Then, EKF, UKF, IUKF, QKF and GHF were used to generate this proposal distribution. Since all these filters use the latest information $z_{1:k}$, the choice of the method of local linearization is better than the transmission prior function. CKF linearizes the nonlinear functions using statistical linear regression method through a set of Gaussian cubature points that parameterize the Gaussian density. Ref.[10] has been proved that CKF has higher estimation accuracy than EKF and UKF, therefore, we will use CKF to generate proposal distributions in this paper.

A. The cubature kalman filter

1) Spherical-radial rule

CKF uses the spherical-radial rule to find the points and weights. For the third-degree spherical-radial rule, it entails a total of $2n$ cubature points when the dimension of the random variable equals n. The cubature points and its corresponding weights will be given as:

$$\xi_i = \sqrt{\frac{m}{2}}[1]_i \tag{6}$$

$$\omega_i = \frac{1}{m}, i = 1, 2, \cdots, m = 2n \tag{7}$$

For example, $[1] \in R^2$ represents the following set of points:

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}$$

2) The cubature kalman filter

This subsection summarizes CKF algorithm that computes both the time update and measurement update steps at each time-step. The cubature-point set $\{\xi_i, \omega_i\}$ should be calculated using (6) and (7) at first. CKF is depicted in the Table Ⅱ.

Table II. The cubature kalman filter

- **Time update step:**

1. Assume at time k that the posterior density function $p(x_{k-1} \mid D_{k-1}) = \mathbb{N}(\hat{x}_{k-1|k-1}, P_{k-1|k-1})$ is known. Factorize

$$P_{k-1|k-1} = S_{k-1|k-1}(S_{k-1|k-1})^T \tag{8}$$

here we can use Cholesky decomposition, the singular value decomposition to factorize the covariance $P_{k-1|k-1}$.

2. Evaluate the cubature points $(i = 1, 2, \ldots, m)$

$$X_{i,k-1|k-1} = S_{k-1|k-1}\xi_i + \hat{x}_{k-1|k-1} \tag{9}$$

where $m = 2n$.

3. Evaluate the propagated cubature points $(i = 1, 2, \ldots, m)$

$$X^*_{i,k|k-1} = f(X_{i,k-1|k-1}, u_{k-1}) \tag{10}$$

4. Estimate the predicted state

$$\hat{x}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} X^{*}_{i,k|k-1} \quad (11)$$

5. Estimate the predicted error covariance

$$P_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} X^{*}_{i,k|k-1}X^{*T}_{i,k|k-1} - \hat{x}_{k|k-1}\hat{x}^{T}_{k|k-1} + Q_{k-1} \quad (12)$$

- **Measurement update step:**

1. Factorize

$$P_{k|k-1} = S_{k|k-1}(S_{k|k-1})^{T} \quad (13)$$

2. Evaluate the cubature points $(i = 1, 2, \ldots, m)$

$$X_{i,k|k-1} = S_{k|k-1}\xi_i + \hat{x}_{k|k-1} \quad (14)$$

3. Evaluate the propagated cubature points

$$Z_{i,k|k-1} = h(X_{i,k|k-1}, u_k) \quad (15)$$

4. Estimate the predicted measurement

$$\hat{z}_{k|k-1} = \frac{1}{m}\sum_{i=1}^{m} Z_{i,k|k-1} \quad (16)$$

5. Estimate the innovation covariance matrix

$$P_{zz,k|k-1} = \frac{1}{m}\sum_{i=1}^{m} Z_{i,k|k-1}Z^{T}_{i,k|k-1} - \hat{z}_{k|k-1}\hat{z}^{T}_{k|k-1} + R_k \quad (17)$$

6. Estimate the cross-convariance matrix

$$P_{xz,k|k-1} = \frac{1}{m}\sum_{i=1}^{m} X_{i,k|k-1}Z^{T}_{i,k|k-1} - \hat{x}_{k|k-1}\hat{z}^{T}_{k|k-1} \quad (18)$$

7. Estimate the Kalman gain

$$W_k = P_{xz,k|k-1}P^{-1}_{zz,k|k-1} \quad (19)$$

8. Estimate the update state

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + W_k(z_k - \hat{z}_{k|k-1}) \quad (20)$$

9. Estimate the corresponding error covariance

$$P_{k|k} = P_{k|k-1} - W_k P_{zz,k|k-1} W_k^{T} \quad (21)$$

B. *The cubature kalman particle filter*

The cubature kalman filter substituting unscented transformation for linearization means relinearizing the measurement equation around more accurate state. The distribution generated by CKF matches the true posterior distribution better than EKF and UKF. So CKF is used to generate more accurate proposal distribution for particle filter. The cubature kalman particle filter is depicted in Table III.

Table III. The cubature kalman particle filter

- **Initialization: at time** $k = 0$

1. For $i = 1, \cdots, N$, sampling from the prior $x_0^{(i)} \sim p(x_0)$.

$$\hat{x}_0^{(i)} = E[x_0^{(i)}] \quad (22)$$

$$P_0^{(i)} = E[(x_0^{(i)} - \hat{x}_0^{(i)})(x_0^{(i)} - \hat{x}_0^{(i)})^{T}] \quad (23)$$

$$S_0^{(i)} = chol\{P_0^{(i)}\} \quad (24)$$

where $\hat{x}_0$ is the initial value of the fixed state estimation, P_0 is the initial value of matrix square-root of the state covariance.

2. For $i = 1, \cdots, N$,

calculate $w_0^{(i)} = p(z_0 \mid x_0^{(i)})$,

calculate the total weight $w_T = \sum_{i=1}^{N} w_0^{(i)}$,

normalize $w_0^{(i)} = w_0^{(i)} / w_T$.

where w_0 is the initial important weights of support points.

- **Prediction and Update**: For each time $k \geq 1$.

1. For $i = 1, \cdots, N$, update the particles $x_{k-1}^{(i)}$ with CKF, then obtain $\{\hat{x}_k^{(i)}, P_k^{(i)}\}$.

2. For $i = 1, \cdots, N$, sample the new particles $\{x_k^{(i)}\}_{i=1}^{N}$ from the proposal distribution function:

$$x_k^{(i)} \sim q(x_k^{(i)} \mid x_{0:k-1}^{(i)}, z_{1:k}) \quad (25)$$

$$q(x_k^{(i)} \mid x_{0:k-1}^{(i)}, z_{1:k}) \approx \mathbb{N}(x_k^{(i)}; \hat{x}_k^{(i)}, P_k^{(i)}) \quad (26)$$

that is to say, draw a sample from importance distribution:

$$x_k^{(i)} \sim \mathbb{N}(x_k^{(i)}; \hat{x}_k^{(i)}, P_k^{(i)}) \quad (27)$$

where $P_k^{(i)} = S_k^{(i)}(S_k^{(i)})^{T}$, It incorporates the current observa- tions to proposal so improve the precision of particle sampling.

3. For each $i = 1, \cdots, N$, calculate the importance weights:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{p(z_k \mid x_k^{(i)})\, p(x_k^{(i)} \mid x_{k-1}^{(i)})}{q(x_k^{(i)} \mid x_{0:k-1}^{(i)}, z_{1:k})} \quad (28)$$

where $p(z_k \mid x_k^{(i)})$ is the likelihood function of the measurement z_k, $p(x_k^{(i)} \mid x_{k-1}^{(i)})$ is decided by system equation, and the $q(x_k^{(i)} \mid x_{0:k-1}^{(i)}, z_{1:k})$ is calculated by the $\{\hat{x}_k^{(i)}, P_k^{(i)}\}$,which is obtained by CKF, the detail calculated way is given by:

$$\bar{w}_k^{(i)} = w_k^{(i)} \Big/ \sum_{i=1}^{N} w_k^{(i)} \quad (29)$$

$$\hat{z}_k^{(i)} = h(\hat{x}_k^{(i)}, measnoise) \quad (30)$$

$$p(z_k \mid x_k^{(i)}) = \exp\left(\frac{(z_k - \hat{z}_k^{(i)})(z_k - \hat{z}_k^{(i)})^{T}}{2R}\right) \quad (31)$$

$$p(x_k^{(i)} \mid x_{k-1}^{(i)}) = \exp\left(\frac{(x_k^{(i)} - \hat{x}_{k-1}^{(i)})(x_k^{(i)} - \hat{x}_{k-1}^{(i)})^{T}}{2Q}\right) \quad (32)$$

$$q(x_k^{(i)} \mid x_{0:k-1}^{(i)}, z_{1:k}) = \frac{1}{\sqrt{\det(P_k^{(i)})}} \exp\left(\frac{(x_k^{(i)} - \hat{x}_k^{(i)})(x_k^{(i)} - \hat{x}_k^{(i)})^{T}}{2P_k^{(i)}}\right) \quad (33)$$

where $\bar{w}_k^{(i)}$ is the normalize weights, $measnoise$ is defined by e_k in equation (2).

4. If $N_{eff} < N_{th}$, ($N_{eff} = 1 \Big/ \sum_{i=1}^{N} (\bar{w}_k^{(i)})^2$ is sample volume, N_{th} is a some given threshold value), set he weights $\bar{w}_k^{(i)} = 1/N$ and apply resampling algorithm

$$[\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}] = \mathrm{Re}\,sample[\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}]. \quad (34)$$

in this paper, we use residual sampling, see[4].

- **Output**:

The experience probability distribution of filtering distribution, system state estimate and covariance are given respectively:

$$p(x_k \mid z_{1:k}) = \sum_{i=1}^{N} \bar{w}_k^{(i)} \delta(x_k - x_k^{(i)}) \quad (35)$$

$$x_k = E(x_k \mid z_{1:k}) \approx \sum_{i=1}^{N} \bar{w}_k^{(i)} x_k^{(i)} \quad (36)$$

$$P_k = \sum_{i=1}^{N} \bar{w}_k^{(i)} (x_k - x_k^{(i)})(x_k - x_k^{(i)})^{T} \quad (37)$$

IV. SIMULATION AND ANALYSIS

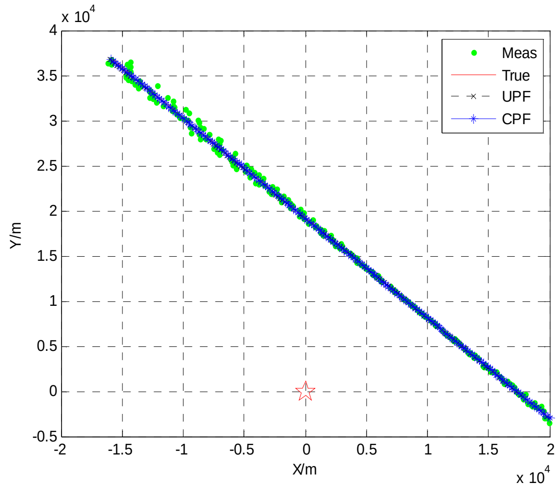
In order to illustrate the performance of CPF proposed in this paper, simulations were carried out and the results are presented in this section. The background of simulations is target tracking problem in a planar surface at nearly constant speed, and the system models were shown as follows

$$X_k = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{k-1} + pro\_noise \qquad (38)$$

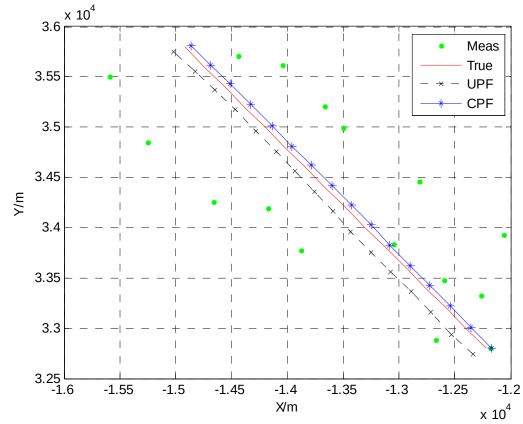
$$Z_k = \begin{bmatrix} L \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt{x^2+y^2} \\ \tan^{-1}(y/x) \end{bmatrix} + mea\_noise \qquad (39)$$

where $X_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]^T$ is the state vector, $Z_k = [L, \alpha]^T$ is the measurement vector, L is the distance from the radar sensor to the target, α is the target azimuth, dt is the sample interval time, pro_noise is the system noise, and mea_noise is the measure noise.

The simulation time is 200 steps, $dt = 1s$. The initial state of the target is $X_0 = [2000m, -180m/s, -3000m, 200m/s]^T$, the initial covariance is $P_0 = diag([10, 0.3, 5, 0.2])$, and $Q = diag([20, 0.001, 20, 0.001])$, $R = diag([5, 5e-4])$. The radar is set at $[x, y] = [0m, 0m]$, and its noise assumed uniform distribution, $n_L \sim unif(-15m, 15m)$, $n_\alpha \sim unif(-2^\circ, 2^\circ)$.



(a) The full tracking



(b) The local tracking (t=180s~195s)

Fig.1 Tracking result (N=200)

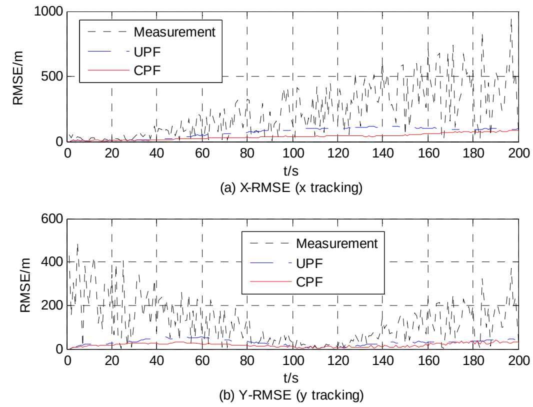


Fig.2 Comparison of the RMSE of position (N=200)

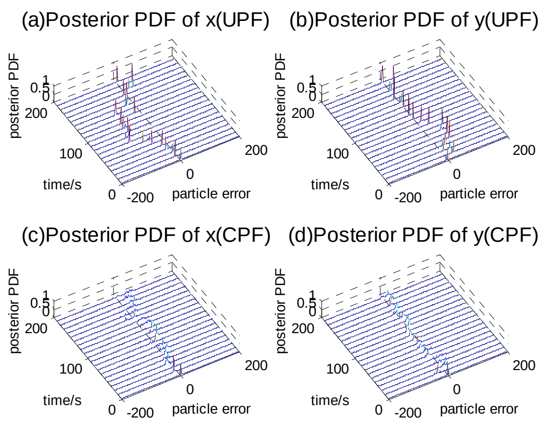


Fig.3 Comparison of the particle posterior PDF(N=200)

In the experiment, we compare CPF with UPF. After 50 times simulations with 200 particles, the full tracking result is shown in Fig.1(a), and the local tracking result during in the 180s~195s is shown in Fig.1(b), Fig.2 shows the RMSE of the state estimation at every time step, and Fig.3 shows the comparison of the particle posterior PDF. The RMSE of different particle filters using different numbers of particles are given in Table Ⅳ, *Time* is the computing time, *N* is the number of particles used in each method.

Table IV. Comparison of position RMSE with different numbers of particles

Filter	N	X-RMSE	Y-RMSE	Time(s)
UPF	50	80.2468	35.3318	0.2459
	100	58.3316	27.2073	0.2460
	200	55.5252	24.4551	0.2467
	500	52.3022	24.0646	0.2476
CPF	50	42.9117	19.3071	0.1856
	100	34.4937	17.1268	0.1858
	200	33.8370	16.0061	0.1864
	500	29.6364	14.2194	0.1874

Comparing the RMSE of the state estimate in Fig.2 , the particles posterior PDF in Fig.3 and the RMSE of the state estimate with the different numbers of particles in Table Ⅳ, it shows that the performance of CPF is superior to UPF solutions when same numbers of particles used, because the proposal distribution based on CKF taken into approximate the true posterior distribution is more precise than UKF. We also find that the number of particles is very influential in determining the results of the filter. Using more particles can produce more accurate results, however, it also requires more calculations to be performed. Table Ⅳ shows that using as few as 50 particles can allow the particle filtering algorithms to produce results that are comparable to UPF in scalar simulations, and the calculation cost is decreased a little.

V. CONCLUSION

In this paper, we proposed CPF algorithm to estimate the state of the nonlinear and non-Gaussian system. The simulation result proved that CPF has a better performance than UPF. Therefore, CPF is an effective nonlinear filtering algorithm.

REFERENCES

[1] M. Sanjeev Arulampalam, Simon Maskell, and Neil Gordon, A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesial Tracking, IEEE Trans. AES, Vol.55, No.2, pp.174-188, 2002.

[2] N.J.Gordon, D.J.Salmond, and A.F.M.Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” IEEE Proc.Inst. Elect. Eng.F, vol.40,no.2,pp.107-113,1993.

[3] R.vander Merwe, A.Doucet, N.de Freitas, et al. The unscented particle filter. Technical Report of the Cambridge University Engineering Department, UK:Cambridge University Press, 2000:1-45.

[4] S.J.Julier, J.K.Uhlmann. “Unscented filtering and nonlinear estimation,” Proc. of the IEEE, 2004,92(3):401-422.

[5] L.Smith, V.Aitken. “The Auxiliay Extended and Auxiliary Unscented Kalman Particle Filters,” in Canadian Conference on Electronical and Computer Engineering, April 22,2007, pp.107-113,1993.

[6] D.Guo, Wang X., and Chen R, “New sequential monte carlo methods for nonlinear dynamic systems,” Statistics and Computing, vol.15, no.2, pp.135-147,2005.

[7] Chunling.Wu, Chongzhao.Han, “Quadrature Kalman particle filter,” Journal of Systems Engineering and Electronics,vol.21,no.2, pp.175-179, April 2010.

[8] Wenyan,Guo,Chongzhao,Han,Minglei,“Improved unscented particle filter for nonlinear bayesian estimation,”Proceeddings of the 10th Interna-tional Conference on Information Fusion, July 9 2007,pp.1-6

[9] I.Arasaratnam, S.Haykin, “Cubature Kalman filters,” IEEE Trans. Automatic Control. vol.54, pp. 1254-1269, June. 2009.

[10] Mingjie Wan, Pengfei Li, Tao Li, “Tracking Maneuvering Target with Angle-Only Measurements Using IMM Algorithm Based on CKF ,” in IEEE International Conference On Communications and Mobile Computing, 2010,pp. 92-96.

[11] Deok-Jin LEE, Nonlinear Bayesian Filtering with Applications to Estimation and Navigation, the dissertation of Ph.D. pp 79-88.2005.

[12] Doucet, A, De. Freitas, N, and Gordon, N.J., Sequential Monte Carlo Methods in Practice, Springer, New York. 2001