Signal reconstruction in wireless sensor networks based on a cubature Kalman particle filter*

Huang Jin-Wang(黄锦旺) and Feng Jiu-Chao(冯久超)†

School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, *China*

(Received 15 August 2013; revised manuscript received 7 February 2014; published online 15 May 2014)

For solving the issues of the signal reconstruction of nonlinear non-Gaussian signals in wireless sensor networks (WSNs), a new signal reconstruction algorithm based on a cubature Kalman particle filter (CKPF) is proposed in this paper. We model the reconstruction signal first and then use the CKPF to estimate the signal. The CKPF uses a cubature Kalman filter (CKF) to generate the importance proposal distribution of the particle filter and integrates the latest observation, which can approximate the true posterior distribution better. It can improve the estimation accuracy. CKPF uses fewer cubature points than the unscented Kalman particle filter (UKPF) and has less computational overheads. Meanwhile, CKPF uses the square root of the error covariance for iterating and is more stable and accurate than the UKPF counterpart. Simulation results show that the algorithm can reconstruct the observed signals quickly and effectively, at the same time consuming less computational time and with more accuracy than the method based on UKPF.

Keywords: cubature rule, particle filter, signal reconstruction, chaotic signals

PACS: 05.45.–a, 05.45.Vx, 84.40.Ua **DOI:** 10.1088/1674-1056/23/7/070504

1. Introduction

Wireless sensor networks (WSNs) are a recent hot research topic. They have great applications in many areas.[1,2] A WSN system is composed of a large number of nodes via wireless communication. It can process complicated signal processing jobs through the cooperation of nodes. A node is a small unit which is limited in energy, communication bandwidth and computational ability.[3,4] Thus, we need to consider these limitations when designing an application. There are already some researches on signal reconstruction. For example, in Ref. [5], the authors used an extended Kalman filter (EKF) to realize signal construction. In Ref. [6], the authors estimated the dynamic stochastic process via iteration and realized the signal reconstruction. In Ref. [7], the authors use UKF for signal reconstruction. A Kalman filter is the optimal estimator for a Gaussian signal, but for nonlinear and non-Gaussian signal estimation, we usually use a particle filter.[8,9] A particle filter is a Monte Carlo estimation algorithm.[10] It uses a set of random samples (particles) with associated weights to represent the required posterior density function and computes estimates based on these samples and weights. Generating the importance proposal distribution is the key part of a particle filter. A generic particle filter uses state evolution to generate the importance proposal distribution. The particles will not track the state variable well when the measurement is changed dramatically, and this will deteriorate the performance of the filter.[11] Many studies tried to solve the issue of particle degeneration and sample impoverishment. Different techniques were used to solve this problem and many improved algorithms were proposed. A cubature Kalman particle filter (CKPF) is such an improved particle filter, which uses a cubature Kalman filter[12,13] to generate the importance proposal distribution and integrates the latest observation. Thus, it can approximate the posterior distribution better and improve the performance of the filter. In this paper, we use the CKPF for signal reconstruction, and propose a new signal reconstruction algorithm. It has fewer cubature points than the unscented Kalman particle filter[14,15] (UKPF) when generating the importance proposal distribution, and has less computational overheads. The algorithm also uses the square root of the error variance for iterating and has the merit of stability and accuracy. Computer simulation is used to evaluate the performance of our algorithm in the last section. The results confirm that the algorithm can reconstruct the signal quickly and effectively. Meanwhile it has fewer computational overheads and is more accurate than the unscented Kalman particle filter counterpart.

2. Signal reconstruction model

Consider a scenario that has N nodes, all the nodes observe a source signal s and send the observations to a fusion center for signal reconstruction. The observation of node n can be expressed as

$$x_n(k) = a_n s(k) + v_n(k), \quad n = 1, 2, \ldots, N, \tag{1}$$

*Project supported by the National Natural Science Foundation of China (Grant No. 60872123), the Joint Fund of the National Natural Science Foundation and the Guangdong Provincial Natural Science Foundation, China (Grant No. U0835001), the Fundamental Research Funds for the Central Universities of Ministry of Education of China (Grant No. 2012ZM0025), the South China University of Technology, China, and the Fund for Higher-level Talent in Guangdong Province, China (Grant No. N9101070).

†Corresponding author. E-mail: fengjc@scut.edu.cn

© 2014 Chinese Physical Society and IOP Publishing Ltd

where $x_n(k)$ is the observation of node n, $s(k)$ is the source signal, a_n is the constant coefficient, $v_n(k)$ is the additive white Gaussian noise, and k is the discrete time instant.

As the bandwidth is limited, the observed signal needs to be quantified before sending it to the fusion center. It is assumed that the probability distribution function of the signal is unknown, we use the uniform quantization here. The quantization noise is $q_n(k)$, the signal in the fusion center is[16]

$$y_n(k) = a_n s(k) + v_n(k) - q_n(k), \quad n = 1, 2, \ldots, N. \tag{2}$$

For rounding quantization, the mean and the variance of the quantizing noise are as follows:[17]

$$m_{q_n} = 0, \tag{3}$$

$$\sigma_{q_n} = \Delta^2/12, \tag{4}$$

where Δ is the step of the quantization and $\Delta = L/(2^{M_n})$; L is the amplitude of signal; M_n is the number of quantization bits. Equation (2) can be rewritten as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{U}, \tag{5}$$

where $\boldsymbol{Y} = [y_1, y_2, \ldots, y_N]^{\mathrm{T}}$, $\boldsymbol{A} = [a_1, a_2, \ldots, a_N]^{\mathrm{T}}$ and $\boldsymbol{U} = [v_1 - q_1, v_2 - q_2, \ldots, v_N - q_N]^{\mathrm{T}}$.

Assume that the chaotic function is already known as

$$s(k) = f(s(k-1)). \tag{6}$$

With state equation (6) and observation equation (5), we can use some kind of adaptive filter to realize the signal reconstruction.

3. Signal reconstruction algorithm

The particle filter algorithm is based on sequential Monte Carlo methods and the optimal recursive Bayesian filtering. It uses a set of particles with associated weights to approximate the posterior distribution. The most common strategy for a particle filter is to sample from the probabilistic model of the state evolution. This will fail when the new measurement changes greatly. Such a situation does indeed arise in several areas of application. In Ref. [15], the authors use an unscented Kalman filter to generate the importance proposal distribution. It allows the filter to integrate the latest measurement and improve the approximation of the posterior distribution. Thus it can make the estimation more accurately. The CKPF follows the same approach, but replace the UKF proposal by a CKF proposal. The CKPF has fewer cubature points than the UKPF and has fewer computational overheads. It uses the square root of the error covariance for iterating and is more stable and accurate.

Consider a weighted integral function

$$\boldsymbol{I}(f) = \int_{R^n} f(x) w(x)\, \mathrm{d}x, \tag{7}$$

where $f(\cdot)$ is a nonlinear function, R^n is the domain of integration, $w(x)$ is the weighted function. This integral is typically intractable and we need to find a suboptimal solution. We can use numerical integration to approximate this integral. Equation (7) can be approximated by

$$\boldsymbol{I}(f) \approx \sum_{j=1}^{m} \omega_j f(\zeta_j), \tag{8}$$

where ζ_j is the integral point, ω_j is the corresponding weight, m is the number of integral points.

The third-degree spherical-radial cubature rule provides an effective method to calculate the integral points and weights. The integral points and corresponding weights are set as

$$\begin{cases} \zeta_j = \sqrt{n_w}[\mathbf{1}]_j, \\ \omega_j = \dfrac{1}{m}, \quad j = 1, 2, \ldots, \quad m = 2n_w, \end{cases} \tag{9}$$

where n_w is the dimension of the vector, $[\mathbf{1}]_j$ is the j-th element of the integral point set $[\mathbf{1}]$, which is defined as

$$[\mathbf{1}] = \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ -1 \end{pmatrix} \right\}. \tag{10}$$

We can now use these cubature points and weights to calculate the time update and measurement update, obtaining the cubature Kalman filter. Here we use it to generate the importance proposal distribution of the particle filter.

From the state evolution function (6) and measurement function (5) obtained above, we apply the CKPF to realize signal reconstruction. The process of the algorithm is as follows.

1) **Initialization**

Draw the states (particles) s_0^i $(1 \le i \le N)$ from the prior $\boldsymbol{P}_0^i$

$$\bar{s}_0^i = E[s_0^i], \tag{11}$$

$$\boldsymbol{S}_0^i = \mathrm{sqrt}\{\boldsymbol{P}_0^i\} = \mathrm{sqrt}\{E[(s_0^i - \bar{s}_0^i)(s_0^i - \bar{s}_0^i)^{\mathrm{T}}]\}, \tag{12}$$

where $E[\cdot]$ is the expectation, $\mathrm{sqrt}\{\cdot\}$ is the square root oper-

ation; $\bar{s}_0^i$ is the initial value, $\boldsymbol{S}_0^i$ is the square root of the error covariance.

2) **Importance sampling**

Update the particles with the CKF using the measurement $\boldsymbol{y}_k$. For particle $\boldsymbol{s}_k^i$, the updating process is:

2.1) Calculate cubature points and propagated cubature points

$$\boldsymbol{W}_{j,k-1|k-1}^i = \boldsymbol{S}_{k-1|k-1}^i \zeta_j + \bar{\boldsymbol{s}}_{k-1|k-1}^i, \tag{13}$$

$$\boldsymbol{W}_{j,k|k-1}^{i*} = f(\boldsymbol{W}_{j,k-1|k-1}^i), \tag{14}$$

where $\boldsymbol{W}_{j,k-1|k-1}^i$ are the cubature points, $\boldsymbol{W}_{j,k|k-1}^{i*}$ are the propagated cubature points, $j = 1,2,\ldots,m$, $m = 2n_w$. The n_w is the dimension of the state vector. The ζ_j is defined as in Eqs. (9) and (10).

2.2) Update the state vector and the square root of covariance

$$\bar{\boldsymbol{s}}_{k|k-1}^i = \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{W}_{j,k|k-1}^{i*}, \tag{15}$$

$$\boldsymbol{\chi}_{k|k-1}^{i*} = \frac{1}{\sqrt{m}}[\boldsymbol{W}_{1,k|k-1}^{i*} - \bar{\boldsymbol{s}}_{k|k-1}^i \cdots \boldsymbol{W}_{m,k|k-1}^{i*} - \bar{\boldsymbol{s}}_{k|k-1}^i], \tag{16}$$

$$\boldsymbol{S}_{k|k-1}^i = \mathrm{qr}\{[\boldsymbol{\chi}_{k|k-1}^{i*} \sqrt{\boldsymbol{Q}_{k-1}}]\}, \tag{17}$$

where $\boldsymbol{Q}_{k-1}$ is the covariance of the process equation, $\mathrm{qr}\{\cdot\}$ is the QR decomposition, $\boldsymbol{\chi}_{k|k-1}^{i*}$ is the weighted, centered (prior mean is subtracted off) matrix with dimensions $n_w \times 2n_w$; $\sqrt{\boldsymbol{Q}_{k-1}}$ is the square root of measurement error covariance, its dimension is $n_w \times n_w$. We denote matrix $\boldsymbol{B} = [\boldsymbol{\chi}_{k|k-1}^{i*} \sqrt{\boldsymbol{Q}_{k-1}}]$, its dimension is $n_w \times 3n_w$. Let $\boldsymbol{R}$ be an upper triangular matrix obtained from the QR decomposition on $\boldsymbol{B}^{\mathrm{T}}$, the expression in Matlab is $[\boldsymbol{Q}\,\boldsymbol{R}] = \mathrm{qr}(B',0)$, then $\boldsymbol{S}_{k|k-1}^i = \boldsymbol{R}^{\mathrm{T}}$ is obtained, it is a lower triangular matrix with dimensions $n_w \times n_w$.

2.3) Calculate the measurement cubature points, propagated measurement cubature points and update the measurement

$$\boldsymbol{W}_{j,k|k-1}^i = \boldsymbol{S}_{k|k-1}^i \zeta_j + \bar{\boldsymbol{s}}_{k|k-1}^i, \tag{18}$$

$$\boldsymbol{Y}_{j,k|k-1}^i = \boldsymbol{A}\boldsymbol{W}_{j,k|k-1}^i + \boldsymbol{U}, \tag{19}$$

$$\bar{\boldsymbol{y}}_{k|k-1}^i = \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{Y}_{j,k|k-1}^i. \tag{20}$$

2.4) Calculate the square root of covariance and cross-covariance

$$\boldsymbol{\Psi}_{k|k-1}^i = \frac{1}{\sqrt{m}}[\boldsymbol{Y}_{1,k|k-1}^i - \bar{\boldsymbol{y}}_{k|k-1}^i \cdots \boldsymbol{Y}_{m,k|k-1}^i - \bar{\boldsymbol{y}}_{k|k-1}^i], \tag{21}$$

$$\boldsymbol{S}_{y,k|k-1}^i = \mathrm{qr}\left\{\left[\boldsymbol{\Psi}_{k|k-1}^i \sqrt{\boldsymbol{R}_k}\right]\right\}, \tag{22}$$

$$\boldsymbol{\chi}_{k|k-1}^i = \frac{1}{\sqrt{m}}[\boldsymbol{W}_{1,k|k-1}^i - \bar{\boldsymbol{s}}_{k|k-1}^i \cdots \boldsymbol{W}_{m,k|k-1}^i - \bar{\boldsymbol{s}}_{k|k-1}^i], \tag{23}$$

$$\boldsymbol{P}_{xy,k|k-1}^i = \boldsymbol{\chi}_{k|k-1}^i \boldsymbol{\Psi}_{k|k-1}^{i\mathrm{T}}, \tag{24}$$

where $\mathrm{qr}\{\cdot\}$ is the QR decomposition, it is the same as that in Eq. (17).

2.5) Update the Kalman gain, state vector and square root of covariance

$$\boldsymbol{G}_k^i = (\boldsymbol{P}_{xy,k|k-1}^i / \boldsymbol{S}_{y,k|k-1}^{i\mathrm{T}}) / \boldsymbol{S}_{y,k|k-1}^i, \tag{25}$$

$$\bar{\boldsymbol{s}}_{k|k}^i = \bar{\boldsymbol{s}}_{k|k-1}^i + \boldsymbol{G}_k^i(\boldsymbol{y}_k - \bar{\boldsymbol{y}}_{k|k-1}^i), \tag{26}$$

$$\boldsymbol{S}_{k|k}^i = \mathrm{qr}\left\{\left[\boldsymbol{\chi}_{k|k-1}^i - \boldsymbol{G}_k^i \boldsymbol{\Psi}_{k|k-1}^i \;\; \boldsymbol{G}_k^i \sqrt{\boldsymbol{R}_k}\right]\right\}, \tag{27}$$

where $\boldsymbol{G}_k^i$ is the Kalman gain, $\bar{\boldsymbol{s}}_{k|k}^i$ is the estimated state vector, $\boldsymbol{S}_{k|k}^i$ is the square root of covariance, $\mathrm{qr}\{\cdot\}$ is the QR decomposition; it is the same as that in Eq. (17).

2.6) Sample particle

$$\hat{\boldsymbol{P}}_{k|k}^i = \boldsymbol{S}_{k|k}^i \boldsymbol{S}_{k|k}^{i\mathrm{T}}, \tag{28}$$

$$\hat{\boldsymbol{s}}_k^i \sim q(\boldsymbol{s}_k^i | \boldsymbol{s}_{0:k-1}^i, \boldsymbol{y}_{1:k}) = N(\bar{\boldsymbol{s}}_{k|k}^i, \hat{\boldsymbol{P}}_{k|k}^i). \tag{29}$$

Here $\hat{\boldsymbol{s}}_{0:k}^i \triangleq (\boldsymbol{s}_{0:k-1}^i, \hat{\boldsymbol{s}}_k^i)$ and $\hat{\boldsymbol{P}}_{0:k|k}^i \triangleq (\boldsymbol{P}_{0:k-1}^i, \hat{\boldsymbol{P}}_{k|k}^i)$.

3) **Evaluate the importance weights up to a normalizing constant and normalize the importance weights**

$$\omega_k^i = \omega_{k-1}^i \frac{p(\boldsymbol{y}_k|\bar{\boldsymbol{s}}_k^i)p(\bar{\boldsymbol{s}}_k^i|\boldsymbol{s}_{k-1}^i)}{q(\bar{\boldsymbol{s}}_k^i|\boldsymbol{s}_{0:k-1}^i, \boldsymbol{y}_{1:k})}, \tag{30}$$

$$\tilde{\omega}_k^i = \omega_k^i \left[\sum_{j=1}^{N} \omega_k^j\right]^{-1}, \tag{31}$$

where ω_k^i are the importance weights, $\tilde{\omega}_k^i$ are the normalized importance weights.

4) **Selection**

Multiply or suppress particles $(\hat{\boldsymbol{s}}_{0:k}^i, \hat{\boldsymbol{P}}_{0:k}^i)$ according to their importance weights $\tilde{\omega}_k^i$ respectively and obtaining N random particles $(\tilde{\boldsymbol{s}}_{0:k}^i, \tilde{P}_{0:k}^i)$.

5) **Update the state vector**

$$\bar{\boldsymbol{s}}_{k+1} = \sum_{i=1}^{N} \tilde{\omega}_k^i \tilde{\boldsymbol{s}}_k^i, \tag{32}$$

where $\bar{\boldsymbol{s}}$ is the reconstruction signal.

4. Simulation results

In this session, we use a computer simulation to evaluate the performance of our algorithm. The parameters are set as below. The source signal is a logistic map[18]

$$s(k) = 1 - 2s^2(k-1). \tag{33}$$

The constant coefficient matrix $\boldsymbol{A}$ is generated randomly and is uniform distributed. We assume that quantization bits and the SNR (signal-to-noise ratio) are the same for all nodes to facilitate the simulation. The number of particles is set to

200. MSE (mean squared error) is used here to evaluate the performance of our algorithm[19]

$$\text{MSE (dB)} = 10\lg\left(\sum_{j=1}^{J}|s_j-\bar{s}_j|^2/J\right), \tag{34}$$

where J is the length of the signal. As the mixing coefficient matrix $\boldsymbol{A}$ and source signals are randomly generated in every simulation, the MSE will vary under the same conditions, so we simulate 200 times for each condition and take the average.

Figure 1 shows the source signal, reconstruction signal and error at the condition $N = 3$, $M_i = 4$ and SNR = 15 dB. The reconstruction signal is similar to the source signal, which demonstrates that our algorithm can reconstruct the source signal effectively.

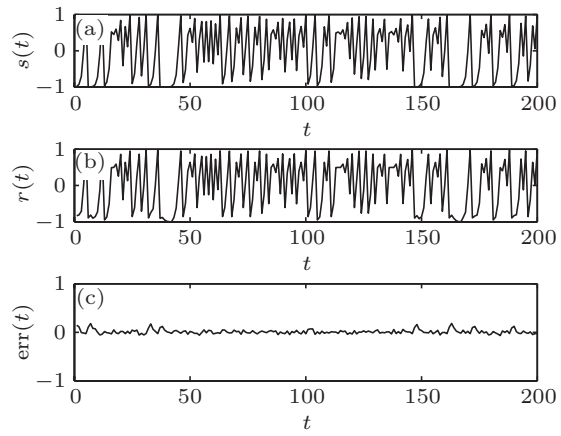


Fig. 1. Source signal and reconstruction signal. (a) $s(t)$, (b) $r(t)$, and (c) err(t).

Figure 2 shows the MSE versus number of nodes at the condition $M_i = 4$ and SNR = 15 dB. The performance gets better when the number of nodes increases, but the performance gains little when $N > 30$. An explanation of the result is that more sensors provide more information of the source, but excessive information by no means contributes to the performance. The CKPF algorithm is better than the UKPF counterpart at the same condition.

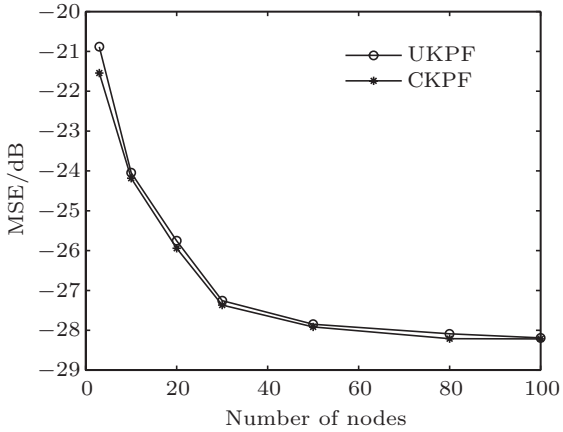


Fig. 2. MSE versus number of nodes.

Figure 3 shows MSE versus number of quantization bits M_i at the condition $N = 3$ and SNR = 15 dB. The performance gets better as the quantization bits increase. More quantization bits will bring greater overheads of communication, computation and storage. Improving performance via increasing the number of quantization bits is at the cost of consuming more resources of nodes. The CKPF algorithm has better performance when quantization bits are few.

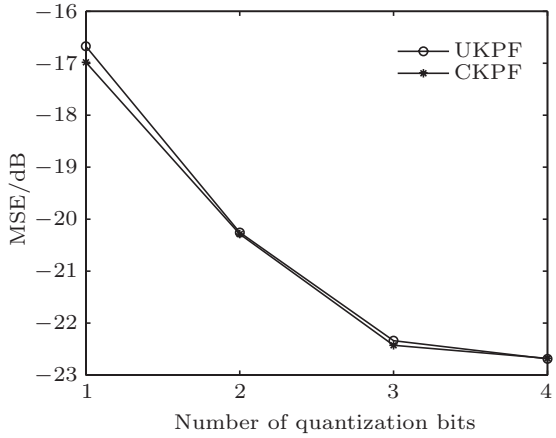


Fig. 3. MSE versus number of quantization bits.

Figure 4 shows the MSE versus SNR at the condition $N = 3$ and $M_i = 4$. The performance generally increases with SNR. The CKPF algorithm performs better when SNR$<$ 20 dB. When SNR$>$ 20 dB the two algorithms have the same performance. The performance gains little when SNR is high. The reason behind this observation is that when SNR is high, the effect of noise is dominated by the signals mixing.

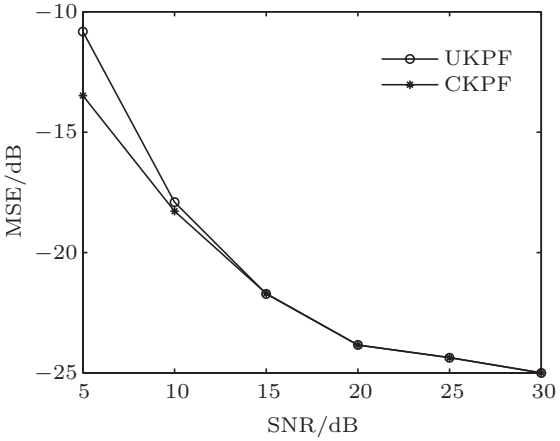


Fig. 4. MSE versus SNR.

Figure 5 shows the time consumed at the condition $N = 3$, $M_i = 4$ and SNR = 15 dB. The average time consumptions are 7.6814 ms and 7.1874 ms respectively for the two algorithms. The CKPF algorithm uses less cubature points than the UKPF algorithm, and has less computational overheads. It takes 93.57% of the time of the UKPF counterpart.

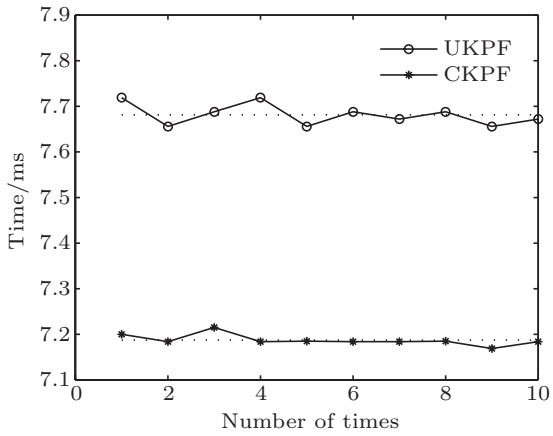


Fig. 5. Time consumption.

5. Conclusions

A new signal reconstruction algorithm based on the CKPF has been proposed in this paper. This new algorithm aims at solving the signal reconstruction problems of nonlinear non-Gaussian signal in wireless sensor networks. It uses the cubature Kalman filter to generate the importance proposal distribution of the particle filter, integrating the latest observation and approximating the posterior distribution, which improves the performance specifically when the observation changes greatly. The CKPF algorithm uses less cubature points than the UKPF and has less computational overheads. Meanwhile, it uses the square root of the variance for iterating and has the merit of stability and accuracy. Simulation results confirm that our algorithm can reconstruct source signals effectively, and the algorithm is more accurat and quicker than the UKPF counterpart.

References

[1] Qi H, Wang F B and Deng H 2013 *Acta Phys. Sin.* **62** 104301 (in Chinese)
[2] Zhang C, Fei S M and Zhou X P 2012 *Chin. Phys. B* **21** 120101
[3] Wang Z, Wang Q, Wei D B and Wang L 2012 *Acta Phys. Sin.* **61** 120505 (in Chinese)
[4] Tong X J, Wang Z and Zou K 2012 *Chin. Phys. B* **21** 020506
[5] Zhu Z and Leung H 2001 *IEEE Trans. Circ. Syst. I* **48** 979
[6] Ribeiro A, Giannakis G B and Roumeliotis S I 2006 *IEEE Trans. Signal Process.* **54** 4782
[7] Vercauteren T, Xiaodong W 2005 *IEEE Trans. Signal Process.* **53** 2997
[8] Leng H Z and Song J Q 2013 *Chin. Phys. B* **22** 030505
[9] Zhu Z Y and Yang G X 2010 *Acta Phys. Sin.* **59** 8316 (in Chinese)
[10] Masrour R, Bahmad L and Benyoussef A 2013 *Chin. Phys. B* **22** 057504
[11] Hlinka O, Hlawatsch F and Djuric P M 2013 *IEEE Signal Process. Mag.* **30** 61
[12] Arasaratnam I and Haykin S 2009 *IEEE Trans. Autom. Control* **54** 1254
[13] Arasaratnam I and Haykin S 2011 *Automatica* **47** 2245
[14] Soken H E and Hajiyev C 2012 *IEEE Trans. Aerosp. Electron. Syst.* **48** 2614
[15] Li W and Jia Y 2011 *IET Signal Process.* **5** 187
[16] Chen H B, Tse C K and Feng J C 2008 *IEEE Trans. Circ. Syst. II* **55** 947
[17] Na S and Neuhoff D L 2012 *IEEE Trans. Inf. Theory* **58** 3169
[18] Wang C J 2013 *Chin. Phys. B* **22** 060502
[19] Luo Z Q 2005 *IEEE Trans. Inf. Theory* **51** 2210