

Article

Double-Layer Cubature Kalman Filter for Nonlinear Estimation [†]

Feng Yang ^{1,2,3,*} , Yujuan Luo ^{1,2} and Litao Zheng ^{1,2}

¹ School of Automation, Northwestern Polytechnical University, Xi'an 710129, China; lyjnwpu@mail.nwpu.edu.cn (Y.L.); zhenglitao@mail.nwpu.edu.cn (L.Z.)

² Key Laboratory of Information Fusion Technology, Ministry of Education, Xi'an 710129, China

³ CETC Key Laboratory of Data Link Technology, No. 20 Institute of CETC, Xi'an 710000, China

* Correspondence: yangfeng@nwpu.edu.cn; Tel.: +86-29-8843-1306

[†] This paper is an extended version of our paper published in "Feng Yang; Yujuan Luo; Litao Zheng; Shaodong Chen; Jie Zou. Double-layer Cubature Kalman Filter. In 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), Hangzhou, China, 24–27 October 2018; doi:10.1109/ICCAIS.2018.8570334".

Received: 13 January 2019; Accepted: 20 February 2019; Published: 26 February 2019



Abstract: The cubature Kalman filter (CKF) has poor performance in strongly nonlinear systems while the cubature particle filter has high computational complexity induced by stochastic sampling. To address these problems, a novel CKF named double-Layer cubature Kalman filter (DLCKF) is proposed. In the proposed DLCKF, the prior distribution is represented by a set of weighted deterministic sampling points, and each deterministic sampling point is updated by the inner CKF. Finally, the update mechanism of the outer CKF is used to obtain the state estimations. Simulation results show that the proposed algorithm has not only high estimation accuracy but also low computational complexity, compared with the state-of-the-art filtering algorithms.

Keywords: nonlinear estimation; deterministic sampling strategy; cubature Kalman filter; cubature particle filter

1. Introduction

Nonlinear estimation has been widely used in many fields, such as information fusion, computer vision, engineering, and economics [1–4]. Some classic estimation algorithms are put forward to deal with nonlinear systems. These filters can be classified as two categories: point estimation (e.g., the extended Kalman filter [5], unscented Kalman filter [6,7] and cubature Kalman filter [8–10]), and density estimation (e.g., the particle filter [11,12]). The former uses moment and density function to characterize variables [13] as the latter approximates posterior density function on the basis of bayesian formula [14]. The extended Kalman filter (EKF) [5] has been widely used in dealing with nonlinear systems. By using Taylor series expansion, it linearizes nonlinear system equations and can be applied in nonlinear systems. However, as it must obey the Gaussian distribution, in strongly nonlinear [15] and non-Gaussian systems, the EKF will perform poorly [16]. Based on unscented transform (UT), the unscented Kalman filter (UKF) [6,7] can obtain a higher filtering accuracy than the EKF. However, the weights of the sigma points in the UKF are prone to be negative in high-dimension systems, leading to numerical instability as well as low estimation accuracy. The cubature Kalman filter (CKF) [8–10] uses the spherical-radial cubature rule to generate some weighted sampling points to approximate integral in Bayesian estimation. It has estimation accuracy up to three order and is widely used in high order nonlinear systems [17]. Therefore, the CKF algorithm is mainly discussed in this paper. However, in non-Gaussian systems, the CKF will degrade. Further, it has numerical instability in the process of practical applications [18].

As the estimation accuracy of the conventional CKF only reaches three order, reference [19] proposed a high-order cubature Kalman filter in order to augment the estimation accuracy. In this filtering method, multiple integral and moment matching are respectively adopted to derive arbitrary-order spherical rule and the radial rule in the frame of points-based Gauss approximation. Consequently, a high-order spherical-radial cubature rule is attained and then employed to calculate Gaussian weighted integrals. As a result, the high-order cubature Kalman filter is established which can improve the estimation accuracy greatly. However, high estimation accuracy is accompanied by large computational load, especially in the problem with a high dimensionality [20]. It is unsuitable for systems which require high real-time performance accordingly [21]. The above methods still suffer from high measurement equation nonlinearity.

The square-root cubature Kalman filter (SRCKF) [18] adopts orthogonal triangular decomposition to avoid the root operation in matrix. As a result, it has merits such as better numerical instability than the CKF as well as high efficiency. The iterated cubature Kalman filter (ICKF) [22] combines the Gauss-Newton iterated method with the CKF, reducing the estimation error. The recursive update extended Kalman filter (RUEKF) [23] incorporates nonlinear measurements into a Kalman filter by applying the update gradually. Specifically, it recalculates Jacobian matrix and keeps the measurement nonlinear while updating [23]. One of the advantages is that its computational load is not excessive since the filter is free from the curse of dimensionality. Above all, it is not necessary to linearize the update equation. On the basis of the similar approach, reference [24] reported the recursive update cubature Kalman filter (RUCKF), in which statistical linearization technique based on sigma transformation is used [24]. The RUCKF does not rely on Jacobian matrix as its Kalman gain is computed by state prior distribution rather than state estimate.

In nonlinear system models with non-Gaussian noise, the particle filter (PF) [11,12], which is based on Bayesian theory and Monte Carlo method, is an effective method to deal with the problem of nonlinear estimation. In the PF algorithm, the probability distribution is approximated by a large number of random samples [25] for which, the weights and the positions of the samples are adjusted to approximate the posterior probability distribution function (PDF) on the basis of the measurement [26,27]. Since the method is not interfered by system linearity and not subject to the Gaussian distribution, it can deal with nonlinear, non-Gaussian system models [28]. However, as the key component of the PF is to approximate importance PDF by selecting particles, the performance of this method is affected by the curse of importance PDF. To solve this problem, a cubature particle filter (CPF) [29,30] has been proposed, where the estimation of the CKF is adopted as the proposal distribution. It greatly improves the performance as compared with the conventional PF. Nevertheless, the CPF suffers from heavy computational load as the result of the use of a large number of particles.

Cascade structure algorithm is used to increase the system efficiency in [31,32]. Reference [31] applied orthogonal genetic-algorithm (OGA) to a cascade finite impulse response (FIR) realization to obtain the multiplierless design. Reference [32] proposed a Map/INS/Wi-Fi integrated system based on a cascaded Particle/Kalman filter framework structure to decrease the system computational burden.

Following the above line of thinking, this paper proposes a new double-layer cubature Kalman filter (DLCKF) algorithm which uses the weighted sampling points to represent the posterior density function at the previous moment. Besides, the weighted sampling points and their corresponding weights are updated by the inner CKF and the latest measurements, respectively. In addition, each sampling point is weighted fusion to attain the initial value. Finally, the final value is obtained by employing the outer CKF to update the initial value. The proposal algorithm is applicable for large process noise and low measurement noise.

This paper is an extended and revised version of the conference paper [33]. It provides new simulations, improvements, and further details, including mainly the following: (a) more detailed derivations and discussions about the correlated algorithms are added; (b) the fundamentals of the proposed algorithm is revised to be more comprehensible; (c) the weight of the updated cubature point

in the outer CKF of the DLCKF is added as well as the flowsheet of the DLCKF is revised for better formulations; (d) four algorithms (UKF, ICKF, UPF, RUCKF) and applications in real-world scenario are added in the simulation to demonstrate the excellent performance of the DLCKF.

The rest of this paper is organized as follows. In Section 2, the fundamentals of the CKF and CPF are briefly reviewed. Section 3 presents a new DLCKF algorithm based on the CKF. Then, simulation results are provided to demonstrate the performance of the proposed algorithm in Section 4. At last, some conclusions are given in Section 5.

2. Problem Formulation

2.1. Cubature Rule

The problem of non-linear integrating in Gauss estimation can be expressed as

$$I(f) = \int_{R^{n_x}} f(x) \exp(-x^T x) dx \quad (1)$$

In order to compute the above numerical integration, the following two steps are taken. First, the integration is transformed into a form of spherical-radial cubature, and then three-degree spherical-radial cubature rule is designed. The specific method is as follows:

In the cubature rule, the key is the transformation from vector x to radius r and direction vector y . Supposing $x = ry$, $y^T y = 1$, then $x^T x = r^2$, $r \in [0, +\infty)$. So, Equation (1) in the spherical-radial coordinate system can be expressed as

$$I(f) = \int_0^\infty \int_{U_{n_x}} f(ry) r^{n-1} \exp(-r^2) d(\sigma y) dr \quad (2)$$

where U_{n_x} denotes sphere surface with radius 1, that is $U_{n_x} = \{y \in R^{n_x} | y^T y = 1\}$, $\sigma(\cdot)$ represents spherical measure. So, Equation (2) can be rewritten. m_r points and m_s points are generated by the radial rule and the spherical rule respectively [18].

So, $m_r \times m_s$ points produced by spherical-radial cubature rule can be expressed as

$$I(f) = \int_{R^{n_x}} f(x) \exp(-x^T x) dx \approx \sum_{j=1}^{m_s} \sum_{i=1}^{m_r} a_i b_j f(r_i y_j) \quad (3)$$

For the three-degree spherical-radial cubature rule, there are $m_r = 1$ and $m_s = 2n_x$. Therefore, $2n$ points are needed. Accordingly, calculating a standard Gauss weight integral can be expressed as

$$I(f) = \int_{R^{n_x}} f(x) N(x; 0, I_{n_x}) dx \approx \sum_{i=1}^m w_i f(\chi_i) \quad (4)$$

where $m = 2n_x$; $\chi_i = \sqrt{m/2} [1]_i$; $w_i = 1/m$, $(i = 1, 2, \dots, m)$, $[1] \in R^{n_x}$ and

$$[1]_i \in \left\{ \left[\begin{array}{c|c|c|c|c|c} 1 & & 0 & & -1 & & 0 \\ 0 & & 0 & & 0 & & 0 \\ \vdots & \dots & \vdots & & \vdots & \dots & \vdots \\ 0 & & 1 & & 0 & & -1 \end{array} \right] \right\} \quad (5)$$

Given $x \sim N(x; \hat{x}, P)$, $P = SS^T$, then the Equation (8) can be transformed into

$$I(f) = \int_{R^{n_x}} f(x) N(x; \hat{x}, P) dx \approx \sum_{i=1}^m w_i f(S\chi_i + \hat{x}) \quad (6)$$

2.2. Cubature Kalman Filter

Consider the discrete-time state stochastic system, which is also used in [20,24] :

$$x_k = f(x_{k-1}) + \omega_{k-1} \quad (7)$$

$$z_k = h(x_k) + v_k, \quad (8)$$

where k denotes the discrete time index, $x_k \in R^m$ is the state vector, $z_k \in R^n$ is the measurement vector. $w_{k-1} \in R^m$ and $v_k \in R^n$ denote the process noise vector and the measurement noise vector. The process noise and the measurement noise are zero-mean random variables with known probability density functions. The functions $f(\cdot)$ and $h(\cdot)$ are assumed to be known.

The process of the CKF Algorithm 1 is summarized as follows:

Algorithm 1 The CKF algorithm [8] for one round

Input: $\hat{x}_{k-1}, P_{k-1}, f, h$

Output: \hat{x}_k, P_k

1: Time update:

Generate m cubature points and corresponding weights:

$$\zeta_{j,k-1} = chol(P_{k-1})\chi_i + \hat{x}_{k-1}, w_j = 1/m$$

Propagate each cubature point:

$$\tilde{\zeta}_{j,k|k-1} = f(\zeta_{j,k-1})$$

Obtain predicted state and predicted covariance:

$$\hat{x}_{k|k-1} = \sum_{j=1}^m w_j \tilde{\zeta}_{j,k|k-1}, P_{k|k-1} = \sum_{j=1}^m w_j \left(\tilde{\zeta}_{j,k|k-1} - \hat{x}_{k|k-1} \right) \left(\tilde{\zeta}_{j,k|k-1} - \hat{x}_{k|k-1} \right)^T + Q$$

2: Measure update:

Generate m integration points and corresponding weights:

$$\zeta_{j,k|k-1} = chol(P_{k|k-1})\chi_i + \hat{x}_{k|k-1}, w_j = 1/m$$

Propagate the integration point:

$$\varepsilon_{j,k|k-1} = h(\zeta_{j,k|k-1})$$

3: Obtain predicted measurement, covariance and mutual covariance:

$$\hat{z}_{k|k-1} = \sum_{j=1}^m w_j \varepsilon_{j,k|k-1}, P_{zz} = \sum_{j=1}^m w_j \left(\varepsilon_{j,k|k-1} - \hat{z}_{k|k-1} \right) \left(\varepsilon_{j,k|k-1} - \hat{z}_{k|k-1} \right)^T + R, P_{xz} = \sum_{j=1}^m w_j \left(\tilde{\zeta}_{j,k|k-1} - \hat{x}_{k|k-1} \right) \left(\varepsilon_{j,k|k-1} - \hat{z}_{k|k-1} \right)^T$$

4: Obtain state estimation and corresponding covariance:

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k \left(z_k - \hat{z}_{k|k-1} \right), P_k = P_{k|k-1} - K_k P_{zz} K_k^T$$

2.3. Cubature Particle Filter

Based on particle filter (PF), the cubature particle filter (CPF) [29,30] adopts the CKF as the importance probability distribution function. Although it is suitable for Gaussian systems and non-Gaussian systems, it may result in heavy computational load owing to the use of a large number of particles. The process of the CPF can be summarized as follows:

- (1) Based on the prior probability distribution $p(x_0)$, select N particles $\{x_0^{(i)}\}_{i=1}^N$;
- (2) Use CKF to update each particle;
- (3) Use latest measurements to update each weight $w_k^{(i)} = p(y_k | \hat{x}_k^{(i)}) p(\hat{x}_k^{(i)} | x_{k-1}^{(i)}) / \pi(\hat{x}_k^{(i)} | x_{k-1}^{(i)}, y_k)$, and then normalize them;
- (4) Calculate the number of effective particles N_{eff} , if $N_{eff} < N_{threshold}$, resample;
- (5) Obtain state estimation \hat{x}_k and covariance P_k by accumulating particles at time k ;
- (6) Get state estimation at each time by repeating steps (2) to (5).

3. Double-Layer Cubature Kalman Filter (DLCKF)

Since CPF needs a large number of random particles to approximate the posterior density function of the state, it results in heavy computational load. In contrast to CPF, the proposed DLCKF uses deterministic weighted sampling points to approximate the posterior density function. The core idea of DLCKF is using the inner CKF to update each sampling point, and then the weight of each sampling point is updated with the latest measurement. Next, weighted summation of the updated sampling points is performed to obtain the initial state estimation for the next time. The obtained initial state estimation is used as the predictive value to run the outer CKF algorithm. Finally, the state estimation is obtained.

The proposed DLCKF consists of inner-layer CKF and outer-layer CKF. The process of DLCKF is given as follows:

Initial value is $\hat{x}_0 = E[x_0]$, and initial covariance can be expressed as $P_0 = \text{cov}(x_0)$

At time $k - 1$, the $2n_x$ cubature points $\{\xi_{i,k-1}\}_{i=1}^{2n_x}$ and the corresponding weight value $\{w_i\}_{i=1}^{2n_x}$ are obtained by cubature rule. Then the inner-layer CKF is used to update each cubature point.

The inner-layer CKF:

The cubature point $\chi_{j,i} = \sqrt{m/2}[1]_{j,i}$ with corresponding weight $w_{j,i} = 1/m$ are obtained by three-degree spherical-radial cubature rule, where m denotes the total number of cubature points, which is twice the state dimension, that is $m = 2n_x$, and n_x is the dimension of the state. $[1] \in R^{n_x}$ can be written as

$$[1]_{j,i} \in \left\{ \begin{array}{c|c|c|c|c|c} 1 & & 0 & & -1 & & 0 \\ 0 & & 0 & & 0 & & 0 \\ \vdots & \cdots & \vdots & & \vdots & \cdots & \vdots \\ 0 & & 1 & & 0 & & -1 \end{array} \right\} \quad (9)$$

Then, based on the cubature point $\{\chi_{j,i}\}_{j=1}^m$ and its weight $\{w_{j,i}\}_{j=1}^m$, the cubature point $\{\xi_{j,i,k-1}\}_{j=1}^m$ of the inner-layer CKF is obtained. Then, we have the time update and measurement update steps of the inner-layer CKF as follows.

Time update:

$$S_{i,k-1} = \text{chol}(P_{i,k-1}) \quad (10)$$

$$\xi_{j,i,k-1} = S_{i,k-1} \cdot \chi_{j,i} + \xi_{i,k-1} \quad (11)$$

$$\xi_{j,i,k|k-1} = f(\xi_{j,i,k-1}) + \omega_{j,i,k-1} \quad (12)$$

$$\hat{x}_{i,k|k-1} = \sum_{j=1}^m w_{j,i} \xi_{j,i,k|k-1} \quad (13)$$

$$P_{i,k|k-1} = \sum_{j=1}^m w_{j,i} (\xi_{j,i,k|k-1} - \hat{x}_{i,k|k-1}) (\xi_{j,i,k|k-1} - \hat{x}_{i,k|k-1})^T + Q \quad (14)$$

Measurement update:

$$S_{i,k|k-1} = \text{chol}(P_{i,k|k-1}) \quad (15)$$

$$\zeta_{j,i,k|k-1} = S_{i,k|k-1} \cdot \chi_{j,i} + \hat{x}_{i,k|k-1} \quad (16)$$

$$\varepsilon_{j,i,k|k-1} = h(\zeta_{j,i,k|k-1}) + v_{j,i,k|k-1} \quad (17)$$

$$\hat{z}_{i,k|k-1} = \sum_{j=1}^m w_{j,i} \varepsilon_{j,i,k|k-1} \quad (18)$$

$$P_{i,zz} = \sum_{j=1}^m w_{j,i} \left(\varepsilon_{j,i,k|k-1} - \hat{z}_{i,k|k-1} \right) \left(\varepsilon_{j,i,k|k-1} - \hat{z}_{i,k|k-1} \right)^T + R \quad (19)$$

$$P_{i,xz} = \sum_{j=1}^m w_{j,i} \left(\zeta_{j,i,k|k-1} - \hat{x}_{i,k|k-1} \right) \left(\varepsilon_{j,i,k|k-1} - \hat{z}_{i,k|k-1} \right)^T \quad (20)$$

$$K_{i,k} = P_{i,xz} P_{i,zz}^{-1} \quad (21)$$

$$\hat{x}_{i,k} = \hat{x}_{i,k|k-1} + K_{i,k} \left(z_{i,k} - \hat{z}_{i,k|k-1} \right) \quad (22)$$

$$P_{i,k} = P_{i,k|k-1} - K_{i,k} P_{i,zz} K_{i,k}^T \quad (23)$$

The outer-layer CKF:

After the cubature point updated by the inner CKF, its weight can be calculated as follows.

$$w_i = w_i \frac{p(z_k | \hat{x}_{i,k}) p(\hat{x}_{i,k} | \hat{x}_{i,k-1})}{q(\hat{x}_{i,k} | z_{1:k})} \quad (24)$$

Normalize the weight as follows:

$$w_i = \frac{w_i}{\sum_{i=1}^{2n_x} w_i} \quad (25)$$

Then, the initial estimation with corresponding covariance at time k can be expressed as

$$\hat{x}_{k|k-1} = \sum_{i=1}^{2n_x} w_i \hat{x}_{i,k} \quad (26)$$

$$P_{k|k-1} = \sum_{i=1}^{2n_x} w_i \left(\hat{x}_{i,k} - \hat{x}_{k|k-1} \right) \left(\hat{x}_{i,k} - \hat{x}_{k|k-1} \right)^T + Q \quad (27)$$

Based on $\hat{x}_{k|k-1}$ and $P_{k|k-1}$, the cubature points can be updated as:

$$S_{k|k-1} = \text{chol} \left(P_{k|k-1} \right) \quad (28)$$

$$\zeta_{i,k|k-1} = S_{k|k-1} \cdot \chi_i + \hat{x}_{k|k-1} \quad (29)$$

$$\varepsilon_{i,k|k-1} = h \left(\zeta_{i,k|k-1} \right) + v_{i,k|k-1} \quad (30)$$

$$\hat{z}_{k|k-1} = \sum_{i=1}^{2n_x} w_i \varepsilon_{i,k|k-1} \quad (31)$$

$$P_{zz} = \sum_{i=1}^{2n_x} w_i \left(\varepsilon_{i,k|k-1} - \hat{z}_{k|k-1} \right) \left(\varepsilon_{i,k|k-1} - \hat{z}_{k|k-1} \right)^T + R \quad (32)$$

$$P_{xz} = \sum_{i=1}^{2n_x} w_i \left(\hat{x}_{i,k} - \hat{x}_{k|k-1} \right) \left(\varepsilon_{i,k|k-1} - \hat{z}_{k|k-1} \right)^T \quad (33)$$

$$K_k = P_{xz} P_{zz}^{-1} \quad (34)$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k \left(z_k - \hat{z}_{k|k-1} \right) \quad (35)$$

$$P_k = P_{k|k-1} - K_k P_{zz} K_k^T \quad (36)$$

Repeating Equations from Equation (10) to (36) state estimation \hat{x} of the proposed DLCKF at each time can be obtained.

The process of the proposed DLCKF algorithm is summarized as Figure 1:

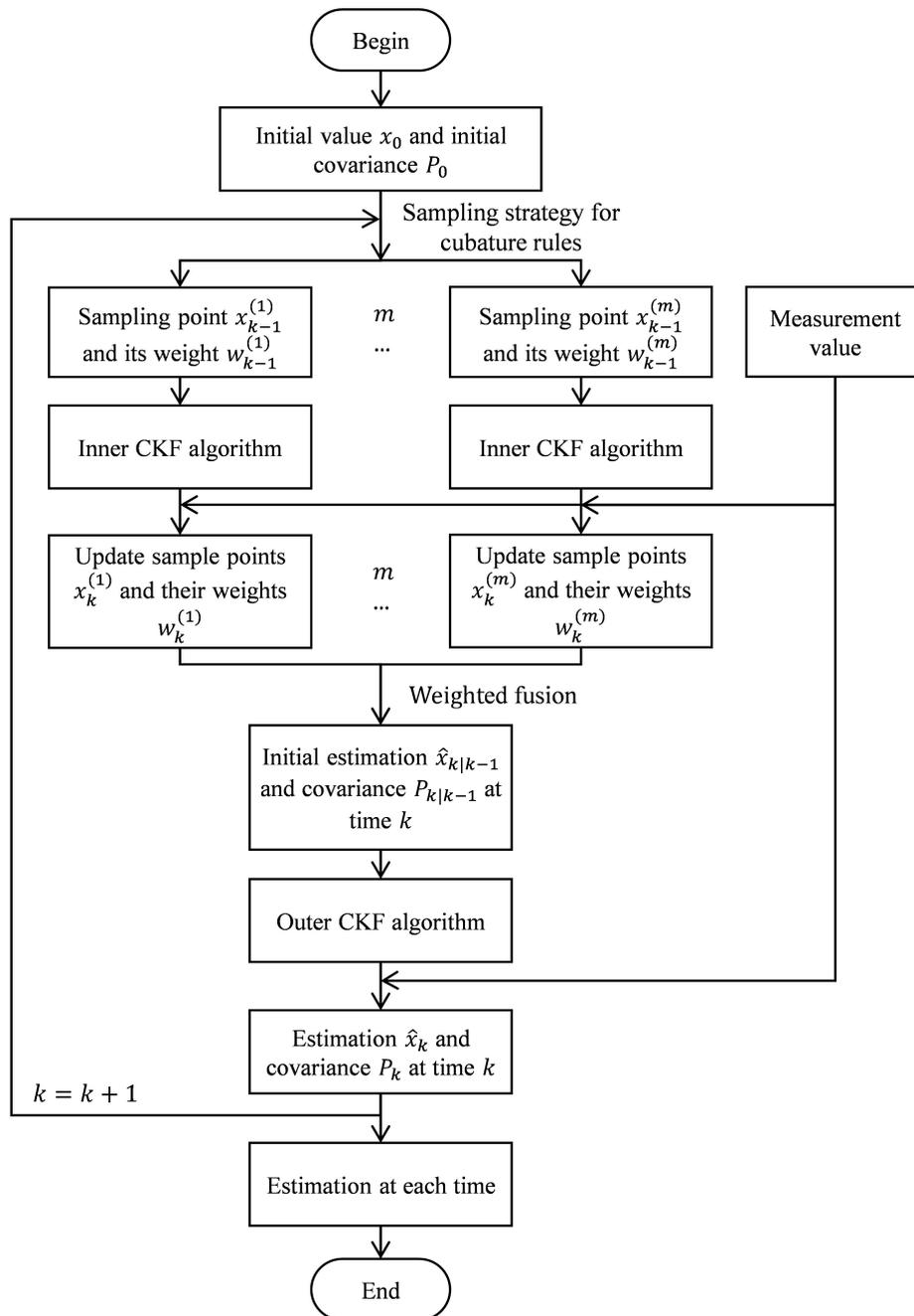


Figure 1. The flowsheet of DLCKF.

4. Simulation Experiments

In this section, to illustrate the effectiveness of the proposed algorithm, the DLCKF is compared with CKF [8–10], UKF [6,7], ICKF [22], CPF [29,30], UPF [34,35] and RUCKF [24] in two classical filtering applications and a real-world scenario. The root mean squared error (RMSE) is used to indicate the estimation performance. All simulations are executed on an Intel Core i5-4200H CPU 2.80 GHz personal computer with 8 GB of random-access memory.

4.1. Single-Dimensional Scenario

We first consider the following univariate nonstationary growth model, which has been also used in [36,37] for its high nonlinearity. Its state space model can be written as:

$$x_{k+1} = 0.5x_k + \sin(0.04\pi k) + 1 + \omega_k \quad (37)$$

$$z_{k+1} = 0.2x_{k+1}^2 + v_{k+1} \quad (38)$$

where process noise ω_k obeys gamma distribution $Ga(3,2)$, and measurement noise v_{k+1} obeys Gaussian distribution with mean of 0 with variance of $R = 10^{-5}$. The initial position is $x_0 = 3$. The iterative number of ICKF is 10 and the particle number of the CPF as well as UPF is set as 100. The Monte Carlo times are 300 and the simulation time is 30 s. The simulation results are shown in Figure 2.

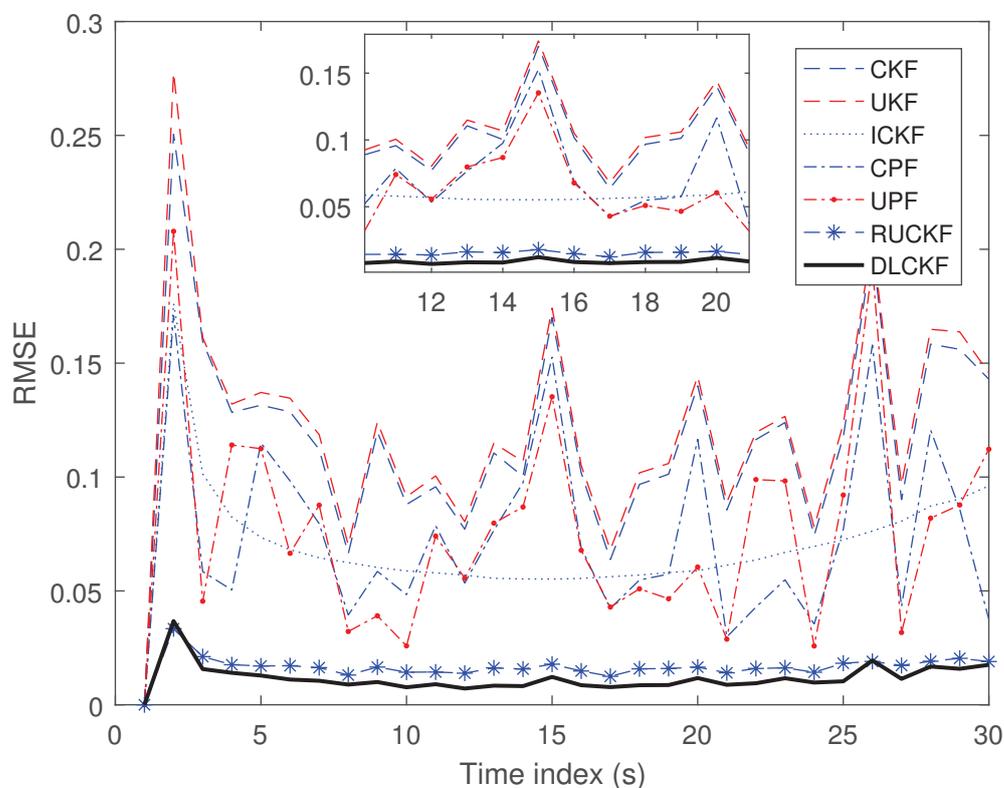


Figure 2. RMSE of 300 Monte Carlo simulations.

It can be seen from Figure 2 that the RMSE of the CKF is similar to that of the UKF because they both have three-order accuracy of Gaussian distribution. The performance of the CPF and the UPF is slightly better than those of the CKF and the UKF, since the former make certain improvements to the latter. The ICKF shows better performance than all above methods to some extent as the result of its iterative process which applies state estimation and covariance of each timestep as the initial value. The RUCKF is adaptable in single-dimensional scenario and therefore its RMSE is low. The proposed algorithm performs best at each timestep, which indicates the improvement of the DLCKF is more significant than that of others.

The number of the particles in the CPF and UPF is gradually increased from 100 to 500. The single run time and average RMSE of each algorithm are shown in Table 1.

Table 1. The run time and the average RMSE of each algorithm.

Algorithm	Run Time (s)	Average RMSE (m)
CKF	0.0013	0.0335
UKF	0.0010	0.0344
ICKF	0.0026	0.0186
CPF(100)	0.5057	0.0241
CPF(200)	1.0398	0.0172
CPF(300)	1.6181	0.0113
CPF(400)	2.2107	0.0111
CPF(500)	2.8181	0.0106
UPF(100)	0.5035	0.0257
UPF(200)	1.0319	0.0146
UPF(300)	1.6009	0.0139
UPF(400)	2.2084	0.0122
UPF(500)	2.8250	0.0103
RUCKF	0.0025	0.0044
DLCKF	0.0035	0.0039

Table 1 illustrates the performance of the run time and average RMSE of each algorithm. It can be seen that the CKF and the UKF take the least time cost, and the CPF and the UPF takes the longest time cost. In the CPF and the UPF, as the number of particles increases, the run time gradually becomes longer. In terms of RMSE, the DLCKF is much smaller than that of the other five methods. The ICKF and RUCKF have lower RMSE than the CKF due to their iteration. As the number of the CPF and UPF increases, their RMSE also decreases. When the particle number is 500, the RMSE of the CPF and the UPF is still twice that of the DLCKF. The above results all indicate that in single-dimensional scenario, the proposed DLCKF has better performance than the other algorithms.

4.2. Multi-Dimensional Scenario

Consider tracking a 2-D uniform linear motion [38], while the state space model expressed as:

$$x_{k+1} = Fx_k + \omega_k \quad (39)$$

$$z_{k+1} = h(x_{k+1}) + v_{k+1} \quad (40)$$

where, $x_k = [x_k, \dot{x}_k, y_k, \dot{y}_k]$ is the state variable that represents the position and velocity of the x axis and y axis, respectively, the process noise ω_k obeys a Gaussian distribution with zeros mean and variance matrix Q , and F and Q are written as

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (41)$$

$$Q = q \begin{bmatrix} T^3/3 & T^2/2 & 0 & 0 \\ T^2/2 & T & 0 & 0 \\ 0 & 0 & T^3/3 & T^2/2 \\ 0 & 0 & T^2/2 & T \end{bmatrix} \quad (42)$$

In Equation (40), $z_{k+1} = [r_{k+1}, \theta_{k+1}]^T$ denotes the measurement variable, which respectively represents the radial distance and azimuth angle of the target, v_{k+1} denotes the measurement noise, which is flicker noise and can be expressed as:

$$\begin{aligned} p(v_{k+1}) &= (1 - \varepsilon)p_1(v_{k+1}) + \varepsilon p_2(v_{k+1}) \\ &= (1 - \varepsilon)N(v_{k+1}; 0, R_1) + \varepsilon N(v_{k+1}; 0, R_2) \end{aligned} \quad (43)$$

The measurement equation $h(\cdot)$ can be written as:

$$h(x_{k+1}) = \left[\sqrt{x_k^2 + y_k^2} \quad \arctan(y_k, x_k) \right]^T \quad (44)$$

In Equation (43), R_1 and R_2 can be expressed as

$$R_1 = \begin{bmatrix} \sigma_{1r}^2 & 0 \\ 0 & \sigma_{1\varepsilon}^2 \end{bmatrix} \quad (45)$$

$$R_2 = \begin{bmatrix} \sigma_{2r}^2 & 0 \\ 0 & \sigma_{2\varepsilon}^2 \end{bmatrix} \quad (46)$$

In this simulation, the initial value of the state is (20,000 m, -160 m/s, 40,000 m, -150 m/s). The iterative times of the ICKF is set as 10, the particle number of the CPF and the UPF is 300. Other parameters are set as Table 2. The simulation time is 100 s, and the Monte Carlo times is set as 300.

Table 2. The Simulation parameters.

Parameter	T	q	σ_{1r}	$\sigma_{1\varepsilon}$	σ_{2r}	$\sigma_{2\varepsilon}$	ε
Value	1 s	$1 \text{ m}^2/\text{s}^3$	20 m	0.2°	200 m	2°	0.1

The RMSE of position can be expressed as:

$$\text{RMSE} = \sqrt{\text{RMSE}_x^2 + \text{RMSE}_y^2} \quad (47)$$

The result of simulation is as Figure 3.

Figure 3 shows the position RMSE of each algorithm. It can be seen that the ICKF, the CPF and the UPF are superior to the CKF and the UKF and the RUCKF is not suitable for the multi-dimensional scenario. The performance of the DLCKF is the best. As a result, the RMSE of the DLCKF is the lowest in the above filtering methods. This also shows that the proposed DLCKF performs greatly on the 2-D uniform linear motion.

In the 2-D uniform linear motion, the number of the particles in the CPF and UPF increases gradually from 300 to 1000. The single run time and the average RMSE of each algorithm are shown as Table 3.

From Table 3, it can be seen that in terms of the running time, the proposed DLCKF is slightly longer than that of the CKF and UKF, but it is far less than that of the CPF and UPF. In the CPF and UPF, as the number of particles increases, their computational time also increases. In terms of average RMSE, as the particle number of the CPF and UPF increases, their average RMSE also gradually reduces. However, compared with the DLCKF, the RMSE of the CPF and UPF is still sizable. All of this indicate that the proposed DLCKF has a great estimation effect in 2-D uniform linear motion.

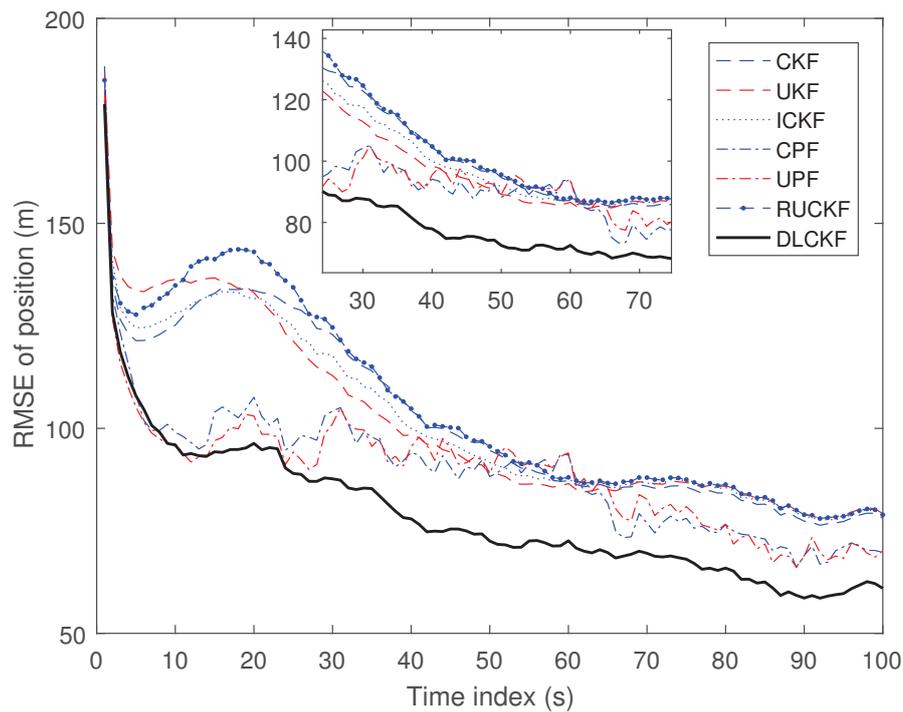


Figure 3. RMSE of position.

Table 3. Performance of each algorithm.

Algorithm	Run Time (s)	Average RMSE (m)
CKF	0.0187	102.8858
UKF	0.0190	97.3787
ICKF	0.0326	101.1720
CPF (300)	9.8903	89.3956
CPF (400)	12.7624	88.4210
CPF (500)	16.0016	87.0589
CPF (600)	19.0964	86.0255
CPF (700)	22.3376	84.6944
CPF (800)	25.6078	83.8060
CPF (900)	28.7884	83.1365
CPF (1000)	31.9711	83.0590
UPF (300)	9.9026	89.2771
UPF (400)	12.8477	87.3480
UPF (500)	16.0309	86.9156
UPF (600)	19.1404	85.6700
UPF (700)	22.3845	83.8380
UPF (800)	25.6139	83.4147
UPF (900)	28.8655	83.0882
UPF (1000)	32.3971	82.8012
RUCKF	0.1140	104.9231
DLCKF	0.2189	79.7820

4.3. Real-World Scenario

Measurement data of the real-world scenario is obtained by a real radar sensor. A given target appears in a $6 \text{ km} \times 2 \text{ km}$ area covered by the sensor. The given target is a car driving along a road in this area with a constant velocity. The initial position of the target is $(-1443.6 \text{ m}, -18.5454 \text{ m})$, the running speed of the target is almost 32 km/h , and scanning period is 8 s . All illustration of the sensor and the target is presented in Figure 4.

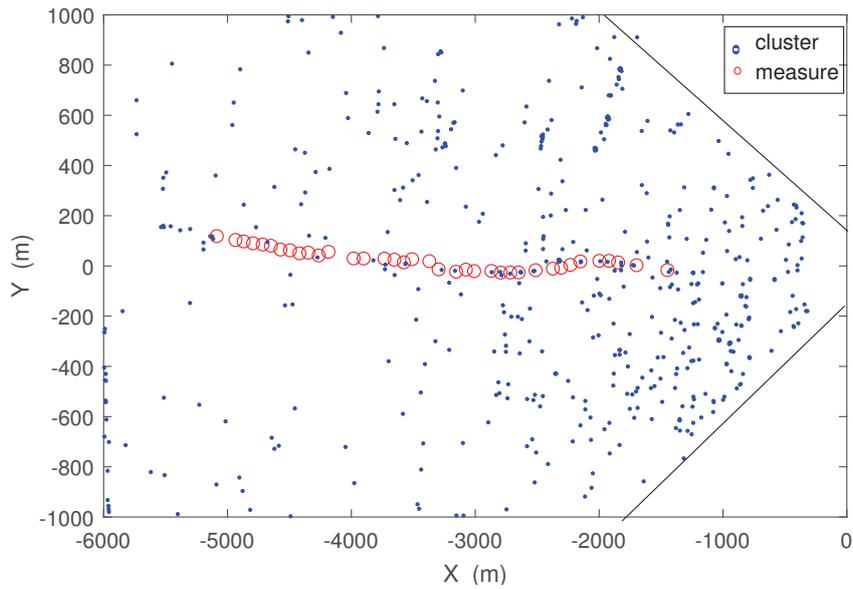


Figure 4. Real-world scenario.

In this experiment, a 2-D uniform linear motion model is applied and measurement value comes from the real radar sensor. We manually wipe off the measurements which are not from the certain target we concerned.

The results of experiments are shown in Figures 5 and 6.

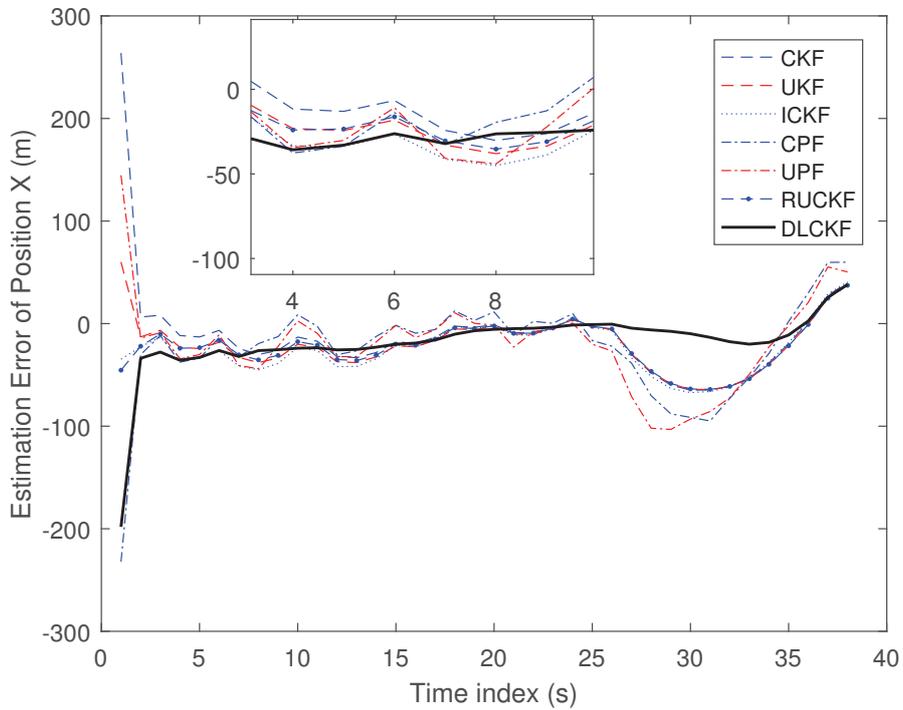


Figure 5. Estimation Error of Position X (m).

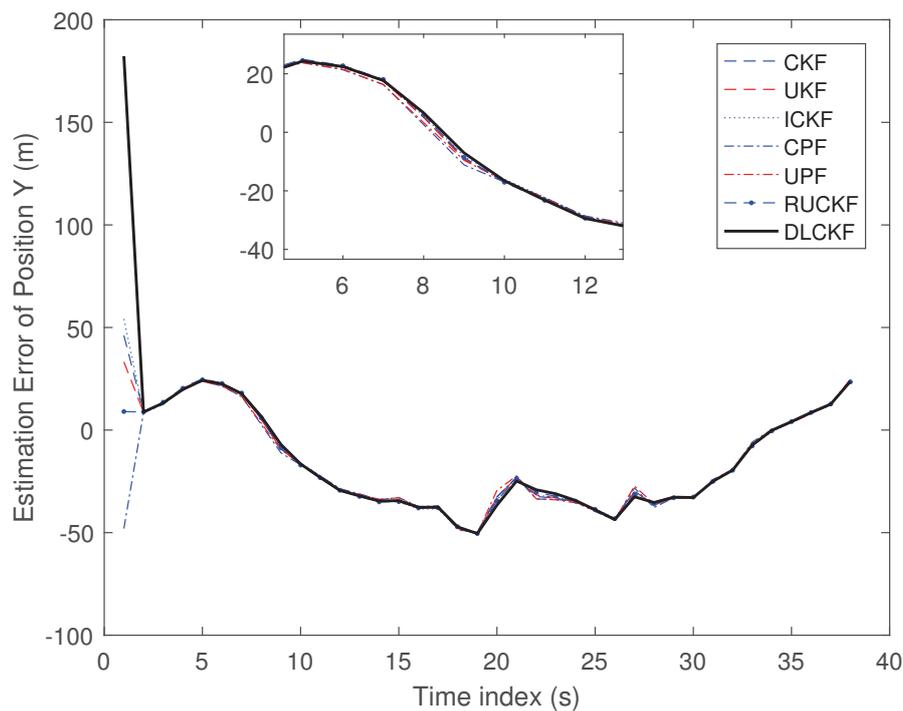


Figure 6. Estimation Error of Position Y (m).

Figures 5 and 6 shows the estimation error of position X, Y of each algorithm in real-world scenario. It can be seen that the CPF and the UPF are superior to the CKF, the UKF, the ICKF and the RUCKF. The DLCKF performs best, which indicates that the proposed DLCKF can obtain great estimation in the real-world scenario.

In this simulation, the number of the particles in the CPF and UPF is 300. The single run time and the average estimation error (AEE) of position X, Y of each algorithm are shown as Table 4.

Table 4. Performance of each algorithm.

Algorithm	Run Time (s)	AEE of Position X (m)	AEE of Position Y (m)
CKF	0.0099	−20.7860	−14.8852
UKF	0.0150	−21.0891	−15.3066
ICKF	0.0710	−22.6846	−14.8039
CPF	7.8441	−19.6904	−12.5453
UPF	8.0290	−19.9458	−13.3300
RUCKF	0.1504	−21.3505	−15.9280
DLCKF	0.1976	−18.3091	−11.3136

From Table 4, it can be seen that in terms of the running time, the proposed DLCKF has slightly longer timecost than that of the CKF and UKF, but it is far less than that of the CPF and UPF. In terms of average estimation error of position X and Y, the CPF and the UPF perform better compared with the CKF and the UKF, and the DLCKF performs best. All of those demonstrate that the proposed DLCKF has better performance than other algorithms in real-world scenario.

5. Conclusions

The DLCKF that is based on CKF is proposed in this paper, which uses the cubature points to approximate the prior density function of the state. The cubature points are updated by the inner-layer CKF, meanwhile, the weight of each cubature point in the outer-layer CKF is updated with the latest measurement. Finally, the state estimations at each time are obtained through the

update mechanism of the outer-layer CKF. The experiments results show that in one-dimensional and multi-dimensional simulation scenarios, the DLCKF performs best compared with the CKF, UKF, ICKF, CPF, UPF and RUCKF. In addition, the superiority of the DLCKF is its effectiveness of in a more challenging environment, which brings great challenges to classic algorithms. The experiments in real-world scenario confirm it. All of those indicate that the proposed DLCKF proposed can obtain an outstanding estimation accuracy with low time cost, compared to classic algorithms.

Author Contributions: F.Y., Y.L. and L.Z. conceived of and designed the algorithm. F.Y., Y.L. and L.Z. performed the experiments and simulations. F.Y. and Y.L. analyzed the data. F.Y. and Y.L. wrote the paper.

Funding: The paper is sponsored by the National Natural Science Foundation of China (No. 61374159), Shaanxi Natural Fund (No. 2018MJ6048), Space Science and Technology Fund, and the Foundation of CETC Key Laboratory of Data Link Technology (CLDL-20182316, CLDL-20182203).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Song, Y.; Nuske, S.; Scherer, S. A Multi-Sensor Fusion MAV State Estimation from Long-Range Stereo, IMU, GPS and Barometric Sensors. *Sensors* **2017**, *17*, 11. [[CrossRef](#)] [[PubMed](#)]
2. Mu, X.; Chen, J.; Zhou, Z.; Leng, Z.; Fan, L. Accurate Initial State Estimation in a Monocular Visual-Inertial SLAM System. *Sensors* **2018**, *18*, 506.
3. Yu, Q.; Xiong, R.; Lin, C.; Shen, W.; Deng, J. Lithium-Ion Battery Parameters and State-of-Charge Joint Estimation Based on H-Infinity and Unscented Kalman Filters. *IEEE Trans. Veh. Technol.* **2017**, *66*, 8693–8701. [[CrossRef](#)]
4. Chen, T.; Yse, F.; Ling, K.V.; Chen, X. Distributed State Estimation Using a Modified Partitioned Moving Horizon Strategy for Power Systems. *Sensors* **2017**, *17*, 2310. [[CrossRef](#)] [[PubMed](#)]
5. Reif, K.; Gunther, S.; Yaz, E.; Unbehauen, R. Stochastic stability of the continuous-time extended Kalman filter. *IEE Proc. Control Theory Appl.* **2000**, *147*, 45–52. [[CrossRef](#)]
6. Julier, S.J.; Uhlmann, J.K. Unscented Filtering and Nonlinear Estimation. *Proc. IEEE* **2004**, *92*, 401–422. [[CrossRef](#)]
7. Peng, S.; Chen, C.; Shi, H.; Yao, Z. State of Charge Estimation of Battery Energy Storage Systems Based on Adaptive Unscented Kalman Filter with a Noise Statistics Estimator. *IEEE Access* **2017**, *99*, 13202–13212. [[CrossRef](#)]
8. Arasaratnam, I.; Haykin, S. Cubature Kalman Filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [[CrossRef](#)]
9. Liu, H.; Wu, W. Strong Tracking Spherical Simplex-Radial Cubature Kalman Filter for Maneuvering Target Tracking. *Sensors* **2017**, *17*, 741. [[CrossRef](#)] [[PubMed](#)]
10. He, R.; Chen, S.; Wu, H.; Hong, L.; Chen, K. Stochastic Feedback Based Continuous-Discrete Cubature Kalman Filtering for Bearings-Only Tracking. *Sensors* **2018**, *18*, 1959. [[CrossRef](#)] [[PubMed](#)]
11. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [[CrossRef](#)]
12. Genshiro, K. Computational aspects of sequential Monte Carlo filter and smoother. *Ann. Inst. Stat. Math.* **2014**, *66*, 443–471.
13. Li, X.R.; Jilkov, V.P. A survey of maneuvering target tracking: approximation techniques for nonlinear filtering. In Proceedings of the 2004 SPIE Conference on Signal and Data Processing of Small Targets, San Diego, CA, USA, 25 August 2004; pp. 537–550.
14. Li, X.R.; Jilkov, V.P. A survey of maneuvering target tracking-part VIIb: Approximate nonlinear density filtering in mixed time. *Proc. SPIE Int. Soc. Opt. Eng.* **2010**, 7698, 76981E–76981E-12.
15. Kozlov, V.V.; Furta, S.D. Lyapunov's first method for strongly non-linear systems. *J. Appl. Math. Mech.* **1996**, *60*, 7–18. [[CrossRef](#)]
16. Simon, D.J. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; Wiley-Interscience: Hoboken, NJ, USA, 2006.

17. Li, T.C.; Su, J.Y.; Liu, W.; Corchado, J.M. Approximate Gaussian conjugacy: Parametric recursive filtering under nonlinearity, multimodality, uncertainty, and constraint, and beyond. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 1913–1939. [[CrossRef](#)]
18. Hao, Y.L.; Yang, J.W.; Chen, L.; Hao, J.H. Square root cubature Kalman filter. *J. Proj. Rocket. Missiles Guid.* **2012**, *32*, 169–172.
19. Jia, B.; Xin, M.; Cheng, Y. *High-Degree Cubature Kalman Filter*; Pergamon Press, Inc.: Oxford, UK, 2013.
20. Zhang, Y.; Huang, Y.; Li, N.; Zhao, L. Embedded cubature Kalman filter with adaptive setting of free parameter. *Signal Process.* **2015**, *114*, 112–116. [[CrossRef](#)]
21. Zhang, Y.; Li, N.; Zhao, L.; Huang, Y. Interpolatory cubature Kalman filters. *IET Control Theory Appl.* **2015**, *9*, 1731–1739. [[CrossRef](#)]
22. Mu, J.; Cai, Y. Iterated cubature Kalman filter and its application. *Syst. Eng. Electron.* **2011**, *33*, 33–37.
23. Zanetti, R. Recursive Update Filtering for Nonlinear Estimation. *IEEE Trans. Autom. Control* **2012**, *57*, 1481–1490. [[CrossRef](#)]
24. Huang, Y.; Zhang, Y.; Li, N.; Zhao, L. Design of Sigma-Point Kalman Filter with Recursive Updated Measurement. *Circuits Syst. Signal Process.* **2015**, *35*, 1767–1782. [[CrossRef](#)]
25. Li, T.C.; Bolic, M.; Djuric, P.M. Resampling Methods for Particle Filtering: Classification, implementation, and strategies. *IEEE Signal Process. Mag.* **2015**, *32*, 70–86. [[CrossRef](#)]
26. Maurelli, F.; Szymon, K.; Petillot, Y.; Salvi, J. A Particle Filter Approach for AUV Localization. In Proceedings of the Oceans 2008, Quebec City, QC, Canada, 15–18 September 2008.
27. Bravo, F.G.; Vale, A.; Ribeiro, M.I. Navigation strategies for cooperative localization based on a particle-filter approach. *Integr. Comput. Aided Eng.* **2007**, *14*, 263–279. [[CrossRef](#)]
28. Wei, W.; Gao, S.; Zhong, Y.; Gu, C.; Hu, G. Adaptive Square-Root Unscented Particle Filtering Algorithm for Dynamic Navigation. *Sensors* **2018**, *18*, 2337. [[CrossRef](#)] [[PubMed](#)]
29. Sun, F.; Tang, L.J. Cubature particle filter. *Syst. Eng. Electron.* **2011**, *33*, 2554–2557.
30. Wang, D.; Yang, F.; Tsui, K.L.; Zhou, Q.; Bae, S.J. Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Spherical Cubature Particle Filter. *IEEE Trans. Instrum. Meas.* **2016**, *65*, 1282–1291. [[CrossRef](#)]
31. Ahmad, S.U.; Antoniou, A. Cascade-form multiplierless FIR filter design using orthogonal genetic algorithm. In Proceedings of the 2006 IEEE International Symposium on Signal Processing and Information Technology, Vancouver, BC, Canada, 27–30 August 2006.
32. Yu, C.; Lan, H.; Gu, F.; Yu, F.; El-Sheimy, N. A Map/INS/Wi-Fi Integrated System for Indoor Location-Based Service Applications. *Sensors* **2017**, *17*, 1272. [[CrossRef](#)] [[PubMed](#)]
33. Yang, F.; Luo, Y.; Zheng, L.; Chen, S.; Zou, J. Double-layer Cubature Kalman Filter. In Proceedings of the 2018 International Conference On Control Automation & Information Sciences (ICCAIS 2018), Hangzhou, China, 24–27 October 2018; pp. 49–54.
34. Merwe, R.V.D.; Doucet, A.; Freitas, N.D.; Wan, E.A. The Unscented Particle Filter. In Proceedings of the 13th International Conference on Neural Information Processing Systems, Denver, CO, USA, 28–30 November 2000; MIT Press: Cambridge, MA, USA, 2000; pp. 563–569.
35. Wang, X.; Li, T.; Sun, S. A Survey of Recent Advances in Particle Filters and Remaining Challenges for Multitarget Tracking. *Sensors* **2017**, *17*, 2707. [[CrossRef](#)] [[PubMed](#)]
36. Prucher, J.; Tronarp, F.; Karvonen, T.; Särkkä, S.; Straka, O. Student-t process quadratures for filtering of non-linear systems with heavy-tailed noise. In Proceedings of the 2017 20th International Conference on Information Fusion, Xi'an, China, 10–13 July 2017; pp. 1–8.
37. Zuo, J.Y.; Jia, Y.N.; Zhang, Y.Z.; Lian, W. Adaptive iterated particle filter. *Electron. Lett.* **2013**, *49*, 742–744. [[CrossRef](#)]
38. Zheng, B.; Fu, P.; Li, B.; Yuan, X. A Robust Adaptive Unscented Kalman Filter for Nonlinear Estimation with Uncertain Noise Covariance. *Sensors* **2018**, *18*, 808. [[CrossRef](#)] [[PubMed](#)]

