

Article

Iterated Orthogonal Simplex Cubature Kalman Filter and Its Applications in Target Tracking

Zhaoming Li , Xinyan Yang, Lei Li and Hang Chen *

School of Space Information, Space Engineering University, Beijing 101416, China; lizmspace@163.com (Z.L.); yangxyspace@163.com (X.Y.); lilspace@163.com (L.L.)

* Correspondence: hitchenhang@foxmail.com

Abstract: In order to increase a nonlinear system's state estimate precision, an iterated orthogonal simplex cubature Kalman filter (IOSCKF) is presented in this study for target tracking. The Gaussian-weighted integral is decomposed into a spherical integral and a radial integral, which are approximated using the spherical simplex-radial rule and second-order Gauss–Laguerre quadrature rule, respectively, and result in the novel simplex cubature rule. To decrease the high-order error terms, cubature points with appropriate weights are taken from the cubature rule and processed using the provided orthogonal matrix. The structure supporting the nonlinear Kalman filter incorporates the altered points and weights and the calculation steps; from this, the updated time and measurement can be inferred. The Gauss–Newton iteration is employed repeatedly to adjust the measurement update until the termination condition is met and the IOSCKF is attained. The proposed algorithms are applied in target tracking, including CV target tracking and spacecraft orbit tracking, and the simulation results reveal that the IOSCKF can achieve higher accuracy compared to the CKF, SCKF, and OSCKF. In spacecraft orbit tracking simulation, compared with the SCKF, the position tracking accuracy and velocity tracking accuracy of the OSCKF are increased by 2.21% and 1.94%, respectively, which indicates that the orthogonal transformation can improve the tracking accuracy. Furthermore, compared with the OSCKF, the position tracking accuracy and velocity tracking accuracy of the IOSCKF are increased by 2.71% and 2.97%, respectively, which indicates that the tracking accuracy can be effectively improved by introducing iterative calculation into the measurement equation, thus verifying the effectiveness of the method presented in this paper.



Citation: Li, Z.; Yang, X.; Li, L.; Chen, H. Iterated Orthogonal Simplex Cubature Kalman Filter and Its Applications in Target Tracking. *Appl. Sci.* **2024**, *14*, 392. <https://doi.org/10.3390/app14010392>

Academic Editor: Atsushi Mase

Received: 30 October 2023

Revised: 23 December 2023

Accepted: 27 December 2023

Published: 31 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: cubature Kalman filter; spherical simplex-radial rule; orthogonal transformation; target tracking; nonlinear system

1. Introduction

For several decades, target tracking has had substantial applications in many fields, such as navigation guidance, military application, and sensor networks [1,2]. Target tracking is a process whereby the position and velocity of a target's motion are estimated using discrete measurements with random noise output by sensors. Mathematical models of target tracking are generally described in different coordinate systems, where the process model is established in the Cartesian coordinate system and the measurements are generally implemented in the spherical coordinate system [3–5]. Thus, target tracking can be considered to be a typical nonlinear filtering problem [6,7]. The posterior probability density function (pdf) is very important in nonlinear filtering theory. Because pdf includes every piece of information on the state vector, it provides an integral filtering solution. Therefore, nonlinear filtering should focus on the calculation of posterior pdf, as the posterior mean and covariance of the state cannot be propagated using the nonlinear function directly.

In nonlinear systems, there contain two types of approaches to estimate the posterior pdf; one is optimal estimation, while the other is suboptimal estimation [8]. The former

mainly includes a particle filter [9] and makes no assumptions on the pdf, which leads to a large amount of computation and is not suitable for applications in engineering. For the latter method, the posterior pdf is assumed to have a Gaussian distribution, and the typical representation, the nonlinear Kalman filter, is primarily studied in practical applications [10].

In a nonlinear Kalman filter, the recursive estimation of the state vector is divided into two processes consisting of a time update and a measurement update, and a unified filtering calculation framework is given. The key problem in the filtering framework is in the approximation of the Gaussian-weighted integral of multi-dimensional nonlinear function [8], and two approximate approaches, including the approximation of the nonlinear function and that of the Gaussian pdf, are mainly taken. The extended Kalman filter (EKF) [11] belongs to the former, and the basic idea of EKF is to use the first-order Taylor expansion to linearize the nonlinear system equations, and then, the conventional Kalman filter is used to estimate the state [12]. It is required to calculate the Jacobian matrix of the system, which demands that the model be smooth and differentiable. The EKF filtering accuracy is reduced or even divergent for a highly nonlinear system, and an improved algorithm, named iterated EKF (IEKF) [13], is proposed. The latest measurements are fully utilized through iterative calculation in the measurement update of IEKF to improve the estimation accuracy; however, the inherent disadvantage of EKF is still present in IEKF, that is, the differential operation of the Jacobian matrix. In general, the approximation of pdf is easier than that of the nonlinear function, so the approximation is performed using the deterministic sampling points in the latter method, and results in the unscented Kalman filter (UKF) [14], Gauss–Hermite–Kalman filter (GHKF) [15], sparse-grid Kalman filter (SGKF) [16], and cubature Kalman filter (CKF) [17,18]. These filters are all derivative-free filters, that is, there is no need to calculate the Jacobian matrix to linearize the nonlinear equation [19]. Specifically, a set of sigma points is used to approximate the posterior mean and covariance in UKF, and Chang [20] proposes using the orthogonal transformation of the sigma points for improving the approximation accuracy without increasing the computational complexity, thus reducing the high-order error term. However, there exist three adjustable parameters in UKF, and the inappropriate choice of parameters may reduce the filtering accuracy and the numerical stability. The computation amount of GHKF increases exponentially with the system state, which may cause the curse of dimensionality [16]. In order to reduce the computational complexity, the sparse-grid integral rule can be utilized to derive the SGKF, and the number of integral points is increased polynomially [17]. However, the CKF is widely used in engineering [21–24]. The CKF helps the intractable Gaussian-weighted integral to decompose and is composed of a spherical integral and a radial integral, and a set of cubature points with equal weights are deduced [25]. It can achieve more filtering accuracy and numerical stability compared with UKF. Similar to the IEKF, the principle of iteration can also be expanded to these filters' measurement update processes [26–29].

Based on the CKF, Jia [30] puts forward the high-degree cubature Kalman filter (HDCKF) by using arbitrary degree symmetric spherical rule and moment matching method, which achieve higher accuracy than the CKF. And then, Wang [31] employs a transformation group of regular simplexes to approximate the spherical integral, and proposes the spherical simplex-radial cubature Kalman filter (SSRCKF). Compared with the HDCKF, the SSRCKF has more accurate performance, whereas the moment matching method cannot guarantee optimality when it is used in the abovementioned approaches' radial integrals. Furthermore, Singh [32] uses the high-order Gauss–Laguerre quadrature rule instead of the moment matching method to calculate the radial integral, and proposes the high-degree cubature quadrature Kalman filter (HDCQKF), which could improve the radial integral accuracy effectively, so as to enhance the filtering accuracy. The advantage of the high-degree cubature Kalman filter is that it can increase the filtering accuracy, although it will increase the computational complexity and reduce the numerical stability for the high-dimensional system, and thus limit the application of filters.

To further improve the nonlinear filtering accuracy without using the high-degree cubature rule, a new iterated orthogonal simplex cubature Kalman filter (IOSCKF) is proposed in this paper. The way to calculate the intractable integral is to combine the spherical simplex rule with the second-order Gauss–Laguerre quadrature rule. By introducing the orthogonal matrix to reduce the higher-order moments of the Taylor series expansion which appear during the approximation of the mean and covariance, the results in the novel cubature points with corresponding weights are improved. The time update is the method used to reduce the computational complexity for the linear process equation. It is equivalent to a simplification of the time update of the standard linear Kalman filter. The posterior state and covariance matrix, in the measurement update, perform Gaussian Newton iteration until the result is optimal. On target tracking problems, the IOSCKF’s performance was tested, and the simulation results verify the effectiveness of the proposed algorithm.

The remainder of this paper is organized as follows: a review of Gaussian approximation filter is described in Section 2, the improved spherical simplex-radial cubature rule is deduced in Section 3, the iterated orthogonal simplex cubature Kalman filter is derived in Section 4, the target tracking simulations and results are presented in Section 5, and the conclusions are drawn in Section 6.

2. Brief Review of Gaussian Approximation Filter

The following discrete nonlinear system model is defined:

$$\begin{cases} x_k = f(x_{k-1}) + w_{k-1} \\ z_k = h(x_k) + v_k \end{cases} \tag{1}$$

where k is the time step, $x_k \in \mathbf{R}^n$ denotes the state vector at time step k , $z_k \in \mathbf{R}^p$ represents the measurement vector at time step k , and $f(\cdot)$ and $h(\cdot)$ are the known nonlinear state function and measurement function, respectively. The process noise w_k and measurement noise v_k are uncorrelated zero-mean Gaussian white noise, with the covariance matrices being Q_k and R_k , respectively.

The key problem in the nonlinear Kalman filter is calculating the multi-dimensional vector integral of the form “nonlinear function \times Gaussian pdf” [8], that is, calculating the integral $I_N = \int_{\mathbf{R}^n} g(x)N(x; \bar{x}, P_x)dx$ in the Cartesian coordinate system, where $g(x)$ denotes the arbitrary nonlinear function, and $N(x; \bar{x}, P_x)$ represents the Gaussian pdf, with the mean and covariance being \bar{x} and P_x , respectively.

This Gaussian-weighted integral I_N can be approximated using various quadrature rules, and the integral with respect to $N(x; \mathbf{0}, I)$ can be approximated by the following quadrature rule:

$$I_N = \int_{\mathbf{R}^n} g(x)N(x; \mathbf{0}, I)dx \approx \sum_{i=1}^{N_p} \omega_i g(\sigma_i) \tag{2}$$

where I is the identity matrix, N_p denotes the total number of points, and σ_i and ω_i represent the quadrature points and weights, respectively.

The integral with respect to the more general Gaussian distribution $N(x; \bar{x}, P_x)$ can be obtained using the equation below:

$$\begin{aligned} I_N &= \int_{\mathbf{R}^n} g(x)N(x; \bar{x}, P_x)dx = \int_{\mathbf{R}^n} g(\bar{x} + \sqrt{P_x}x)N(x; \mathbf{0}, I)dx \\ &\approx \sum_{i=1}^{N_p} \omega_i g(\bar{x} + \sqrt{P_x}\sigma_i) \end{aligned} \tag{3}$$

where $\sqrt{P_x}$ denotes the square root of P_x , which can be solved by the Cholesky decomposition, and $\bar{x}^{(i)} = \bar{x} + \sqrt{P_x}\sigma_i$ represents the integral point.

The Gaussian approximation filter, which is deduced using the numerical integration method in Equation (3), is summarized below [30].

Time Update:

The predicted state \hat{x}_k^- and error covariance P_k^- are calculated, respectively, as follows:

$$\hat{x}_k^- = \sum_{i=1}^{N_p} \omega_i f(\hat{x}_{k-1}^{(i)}) \tag{4}$$

$$P_k^- = \sum_{i=1}^{N_p} \omega_i (f(\hat{x}_{k-1}^{(i)}) - \hat{x}_k^-) (f(\hat{x}_{k-1}^{(i)}) - \hat{x}_k^-)^T + Q_{k-1} \tag{5}$$

where $\hat{x}_{k-1}^{(i)} = \hat{x}_{k-1}^+ + \sqrt{P_{k-1}^+} \sigma_i$ stands for the integral points generated using the numerical rule.

Measurement Update:

The updated state \hat{x}_k^+ and corresponding error covariance P_k^+ are estimated, respectively, as follows:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (z_k - \hat{z}_k) \tag{6}$$

$$P_k^+ = P_k^- + K_k P_z K_k^T \tag{7}$$

where $K_k = P_{xz} P_z^{-1}$ denotes the Kalman filtering gain, $\hat{z}_k = \sum_{i=1}^{N_p} \omega_i h(\hat{x}_k^{(i)})$ represents the predicted measurement, $\hat{x}_k^{(i)} = \hat{x}_k^- + \sqrt{P_k^-} \sigma_i$ denotes the integral points generated by the numerical rule, $P_z = \sum_{i=1}^{N_p} \omega_i (h(\hat{x}_k^{(i)}) - \hat{z}_k) (h(\hat{x}_k^{(i)}) - \hat{z}_k)^T + R_k$ denotes the innovation covariance matrix, and $P_{xz} = \sum_{i=1}^{N_p} \omega_i (\hat{x}_k^{(i)} - \hat{x}_k^-) (h(\hat{x}_k^{(i)}) - \hat{z}_k)^T$ represents the cross-covariance matrix.

3. The Improved Spherical Simplex-Radial Cubature Rule

In this section, an improved spherical simplex-radial cubature rule is proven to be more accurate than the traditional conventional spherical-radial rule. Before introducing the improved cubature rule, the definition of the d th-degree rule is given first for the integral

Definition 1 ([30]).

$$\int_{\mathbf{R}^n} g(x) \omega_g(x) dx \approx \sum_i \omega_i g(\gamma_i) \tag{8}$$

where $x = (x_1 \ x_2 \ \dots \ x_n)^T \in \mathbf{R}^n$ and $\omega_g(x)$ is a given weighing function. Equation (8) is a d th-degree rule if it is accurate for $g(x)$, whose components are linear combinations of monomials $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$ with a total degree up to d . $\alpha_1, \alpha_2, \dots, \alpha_n$ are nonnegative integers and $\alpha_1 + \alpha_2 + \dots + \alpha_n \leq d$, and there is at least one monomial of degree $d + 1$ for which Equation (8) is not exact.

Specifically, an intractable integral of the form $I_g = \int_{\mathbf{R}^n} g(x) e^{-x^T x} dx$ is considered first in the cubature rule. Let $x = r\mathbf{y}$ with $\mathbf{y}^T \mathbf{y} = 1$, where \mathbf{y} means the direction vector and the radius $r \geq 0$. As a consequence, $x^T x = r^2$; then, the integral I_g can be written anew as in a spherical-radial coordinate system as follows [12]:

$$I_g = \int_0^\infty \int_{U_n} g(r\mathbf{y}) r^{n-1} e^{-r^2} d\sigma(\mathbf{y}) dr \tag{9}$$

where U_n is the sphere surface defined by $U_n = \{\mathbf{y} \in \mathbf{R}^n : \mathbf{y}^T \mathbf{y} = 1\}$, and $\sigma(\cdot)$ is the area element on U_n . The integral I_g is decomposed into the spherical integral $S(r) = \int_{U_n} \mathbf{g}(r\mathbf{y})d\sigma(\mathbf{y})$ and radial integral $R = \int_0^\infty S(r)r^{n-1}e^{-r^2}dr$, respectively.

Because of the difficulty in achieving the analytical solutions, the approximate solution method of I_g requires numerical integration, which should be approximated using the numerical integration method. From the third-degree spherical-radial cubature rule proposed in [17], we obtain

$$S(r) \approx \sum_{i=1}^{N_s} \omega_{s,i} \mathbf{g}(r\mathbf{y}_i) \tag{10}$$

$$R \approx \sum_{j=1}^{N_r} \omega_{r,j} S(r_j) \tag{11}$$

and I_g is approximated using the following formula:

$$I_g = \int_0^\infty r^{n-1}e^{-r^2}dr \int_{U_n} \mathbf{g}(r\mathbf{y})d\sigma(\mathbf{y}) \approx \sum_{j=1}^{N_r} \sum_{i=1}^{N_s} \omega_{r,j} \omega_{s,i} \mathbf{g}(r_j \mathbf{y}_i) \tag{12}$$

where \mathbf{y}_i and $\omega_{s,i}$ denote the points and weights to calculate the spherical integral, and the number of points is N_s . r_j and $\omega_{r,j}$ represent the points and weights to calculate the radial integral, and the number of points is N_r . The total number of points is $N_r N_s$ if $r_j \neq 0$, or $(N_r - 1)N_s + 1$ if one of r_j is zero.

3.1. Spherical Simplex Rule

The proposed third-degree spherical rule and spherical simplex rule to calculate Equation (10) are listed below, respectively.

The third-degree spherical rule is

$$S_s(r) \approx \sum_{i=1}^{N_s} \omega_{s,i} \mathbf{g}(r\mathbf{y}_i) = \frac{A_n}{2n} \sum_{i=1}^n [\mathbf{g}(r\mathbf{e}_i) + \mathbf{g}(-r\mathbf{e}_i)] \tag{13}$$

where $A_n = 2\sqrt{\pi^n}/\Gamma(n/2)$ denotes the surface area of the unit sphere, and $\Gamma(z) = \int_0^\infty e^{-t}t^{z-1}dt$ is the Gamma function. The vector \mathbf{e}_i represents the unit column vector with the i th element being 1. The number of cubature points needed is $N_s = 2n$.

The third-degree spherical simplex rule [31] consisting of $N_s = 2(n + 1)$ points is

$$S_{ss}(r) = \frac{A_n}{2(n + 1)} \sum_{i=1}^{n+1} [\mathbf{g}(r\mathbf{a}_i) + \mathbf{g}(-r\mathbf{a}_i)] \tag{14}$$

where $\mathbf{a}_j = (a_{j,1} \ a_{j,2} \ \dots \ a_{j,n})^T, j = 1, 2, \dots, n + 1$ denote the vertices of the n -dimensional simplex, and the elements are defined as follows.

$$a_{j,i} = \begin{cases} -\sqrt{\frac{n+1}{n(n-i+2)(n-i+1)}}, i < j \\ \sqrt{\frac{(n+1)(n-j+1)}{n(n-j+2)}}, i = j \\ 0, i > j \end{cases} \tag{15}$$

Although the aforementioned two spherical rules share the same accuracy of the third degree, through extensive simulations, it has been proven that spherical simplex rules have higher accuracy than spherical rules [31], and that is the reason the spherical simplex rule is chosen to approximate $S(r)$ in this paper. For more details, please refer to the related references.

3.2. Radial Rule

Then, the calculation of the radial integral R is considered. Two methods, including the moment matching method and Gauss–Laguerre quadrature rule, may be used. To construct the $(2m + 1)$ th-degree radial rule, the two methods lead to identical points and weights when m is odd and $(m + 1)/2$ points are used. However, they may be different when m is even and $(m/2 + 1)$ points are used [30]. In the CKF and SSRCKF, the radial integral is calculated using the former one, and results in the following third-degree radial rule with only one point (that is, $m = 1, N_r = 1$).

$$R = \frac{1}{2} \Gamma\left(\frac{n}{2}\right) S\left(\sqrt{\frac{n}{2}}\right) \tag{16}$$

In this paper, the second-order Gauss–Laguerre quadrature rule is adopted to approximate the radial integral. This method improves the accuracy of radial integration.

For radial integral $R = \int_0^\infty S(r)r^{n-1}e^{-r^2}dr$, let $r^2 = t$ to obtain $r = \sqrt{t}$, which takes the place of the radial integral, and for $R = \frac{1}{2} \int_0^\infty S(\sqrt{t})t^{\frac{n-2}{2}}e^{-t}dt$, let $q(t) = S(\sqrt{t})$, and $\beta = \frac{n}{2} - 1$. Then, R is transformed into $R = \frac{1}{2} \int_0^\infty q(t)t^\beta e^{-t}dt$, and this form can be handily approximated using the Gauss–Laguerre quadrature [32] rule as follows:

$$R = \frac{1}{2} \int_0^\infty q(t)t^\beta e^{-t}dt \approx \frac{1}{2} \sum_{j=1}^m A_j q(t_j) \tag{17}$$

where t_j denotes the quadrature points, with A_j being the corresponding weights.

The approximation accuracy depends on the order m . The m -order Chebyshev–Laguerre polynomial’s solutions are the quadrature points, as shown in the following:

$$L_m^\beta(t) = (-1)^m t^{-\beta} e^t \frac{d^m}{dt^m} (t^{\beta+m} e^{-t}) = 0 \tag{18}$$

and the weights are calculated using the following equation.

$$A_j = \frac{m! \Gamma(\beta + m + 1)}{t_j \left[L_m^\beta(t_j) \right]^2} \tag{19}$$

On account of m being set at 2, the solutions are obtained are $t_1 = n/2 + 1 + \sqrt{n/2 + 1}$, $t_2 = n/2 + 1 - \sqrt{n/2 + 1}$, $A_1 = n\Gamma(n/2)/(2n + 4 + 2\sqrt{2n + 4})$, and $A_2 = n\Gamma(n/2)/(2n + 4 - 2\sqrt{2n + 4})$, which are substituted into Equation (17) to deduce the following radial rule.

$$R = \frac{n\Gamma\left(\frac{n}{2}\right)}{4n + 8 + 4\sqrt{2n + 4}} S\left(\sqrt{\frac{n}{2} + 1 + \sqrt{\frac{n}{2} + 1}}\right) + \frac{n\Gamma\left(\frac{n}{2}\right)}{4n + 8 - 4\sqrt{2n + 4}} S\left(\sqrt{\frac{n}{2} + 1 - \sqrt{\frac{n}{2} + 1}}\right) \tag{20}$$

From Equation (20) we see that the number of points is $N_r = 2$, which leads to the radial rule to achieve the fifth-degree accuracy. Compared with third-degree radial rule used in the CKF and SSRCKF Equation (16), the fifth-degree accuracy is more accurate. It is noted that the two points can be solved using the moment matching method as well, with one of the points set to zero. However, it is pointed out that the two points generated by the Gauss–Laguerre quadrature rule have higher radial accuracy than those generated using the moment matching method [30].

3.3. Spherical Simplex-Radial Cubature Rule

By substituting Equation (14) into Equation (20), we obtain the following improved spherical simplex-radial cubature rule which is composed of $N_p = N_r N_s = 4(n + 1)$ points.

$$\begin{aligned}
 I_N &= \int_{\mathbb{R}^n} g(x) N(x; \bar{x}, P_x) dx \\
 &= \frac{n}{4(n+1)(n+2+\sqrt{2n+4})} \sum_{i=1}^{n+1} \left[g\left(\bar{x} + \sqrt{(n+2+\sqrt{2n+4})P_x} a_i\right) + g\left(\bar{x} - \sqrt{(n+2+\sqrt{2n+4})P_x} a_i\right) \right] + \\
 &\frac{n}{4(n+1)(n+2-\sqrt{2n+4})} \sum_{i=1}^{n+1} \left[g\left(\bar{x} + \sqrt{(n+2-\sqrt{2n+4})P_x} a_i\right) + g\left(\bar{x} - \sqrt{(n+2-\sqrt{2n+4})P_x} a_i\right) \right]
 \end{aligned} \tag{21}$$

If we define matrix $a = [a_1 \ a_2 \ \dots \ a_{n+1}]$, and the calculation methods of the cubature points and weights are extracted from cubature rule (21), we obtain the following.

$$\bar{x}^{(i)} = \begin{cases} \bar{x} + \sqrt{(n+2+\sqrt{2n+4})P_x} [a \ -a]_i, i = 1, 2, \dots, 2n+2 \\ \bar{x} + \sqrt{(n+2-\sqrt{2n+4})P_x} [a \ -a]_i, i = 2n+3, \dots, 4n+4 \end{cases} \tag{22}$$

$$\omega_i = \begin{cases} \frac{n}{4(n+1)(n+2+\sqrt{2n+4})}, i = 1, 2, \dots, 2n+2 \\ \frac{n}{4(n+1)(n+2-\sqrt{2n+4})}, i = 2n+3, \dots, 4n+4 \end{cases} \tag{23}$$

The order of the quadrature rule determines the accuracy of the cubature rule. The relationship of the number of quadrature points and the accuracy is proportional [8,32]. The radial rule proposed in Section 3.2 uses two points to achieve fifth-degree accuracy. In contrast to the third degree adopted by the CKF and SSRCKF, when only one point is used, fifth-degree is more accurate. And the spherical simplex rule performs better than the spherical rule; therefore, the improved spherical simplex-radial cubature rule, which is shown in Equation (21), can achieve higher accuracy than the conventional CKF and SSRCKF.

4. Iterated Orthogonal Simplex Cubature Kalman Filter

4.1. Orthogonal Transformation of Cubature Points

It is suggested from the invariant theory and the orthogonal transformation property that the transformed points $O \times [a \ -a]$, with weights shown in Equation (23), remain to constitute a numerical cubature rule if O is an $n \times n$ orthogonal matrix. The orthogonally transformed points are able to reduce the higher-order terms of the Taylor series expansion that appeared while approximating the mean and covariance [20], and it is observed that more accurate estimation is obtained with the transformed points compared to the ordinary cubature points, which lead to the approximation accuracy of the cubature rule becoming further enhanced. In other words, the new set of cubature points can be modified as follows:

$$\bar{x}^{(i)} = \begin{cases} \bar{x} + \sqrt{(n+2+\sqrt{2n+4})P_x} \gamma_i, i = 1, 2, \dots, 2n+2 \\ \bar{x} + \sqrt{(n+2-\sqrt{2n+4})P_x} \gamma_i, i = 2n+3, \dots, 4n+4 \end{cases} \tag{24}$$

where $\gamma = [\gamma_1 \ \gamma_2 \ \dots \ \gamma_{2n+2}] = O \times [a \ -a]$, and the orthogonal matrix O can be designed using different methods. As an effective design method, the $n \times n$ matrix O is constructed in this paper as $O = [O_1 \ O_2 \ \dots \ O_n]$, and $O_i = (O_{i,1} O_{i,2} \dots O_{i,n})^T$, whose elements are defined as follows:

$$\begin{cases} O_{i,2r-1} = \sqrt{\frac{2}{n}} \cos\left[\frac{(2r-1)i\pi}{n}\right] \\ O_{i,2r} = \sqrt{\frac{2}{n}} \sin\left[\frac{(2r-1)i\pi}{n}\right] \end{cases}, r = 1, 2, \dots, \max\left(p \in \mathbf{Z} \mid p \leq \frac{n}{2}\right) \tag{25}$$

where $O_{i,n} = (-1)^i / \sqrt{n}$ if n is odd.

The information of high-order terms can be reduced to a low level using the orthogonal matrix, which means the non-local sampling problem can be significantly reduced with a rare computational complexity increase.

4.2. Iterated Measurement Update Process

The prior state \hat{x}_k^- and covariance matrix P_k^- are used in the measurement update process to calculate the posterior state \hat{x}_k^+ and covariance matrix P_k^+ , which are revised using the latest measurement z_k via Kalman gain K_k . This means an approximation accuracy of \hat{x}_k^+ to the real state is bound to be better than \hat{x}_k^- , and the filtering accuracy will be improved if we can use \hat{x}_k^+ and P_k^+ directly in the measurement update. Therefore, to iteratively make use of the measurement information, the Gauss–Newton iteration is introduced into the updating process of the mean and covariance of the state. Under the assumption of Gaussian variable independence, there exist $x_k : N(\hat{x}_k^-, P_k^-)$ and $z_k : N(h(x_k), R_k)$, and the likelihood function of x_k and z_k is defined as follows [33].

$$p(x_k|z_k) = C \exp \left\{ -\frac{1}{2} \left[(x_k - \hat{x}_k^-)^T (P_k^-)^{-1} (x_k - \hat{x}_k^-) + (z_k - h(x_k))^T R_k^{-1} (z_k - h(x_k)) \right] \right\} \quad (26)$$

where C is a constant, and the cost function is defined as follows.

$$J(x_k) = (x_k - \hat{x}_k^-)^T (P_k^-)^{-1} (x_k - \hat{x}_k^-) + [z_k - h(x_k)]^T R_k^{-1} [z_k - h(x_k)] \quad (27)$$

The maximum likelihood estimate of Equation (26) is equivalent to the minimum value of the cost function (27), which is usually solved by the nonlinear iteration method. The linearized measurement equation and the Gauss–Newton iteration are used to achieve the iterative formula for solving the minimum value of J_{x_k} as follows:

$$\hat{x}_k^{i+1} = \hat{x}_k^- + P_k^- (H_k^i)^T \left[H_k^i P_k^- (H_k^i)^T + R_k \right]^{-1} \left[z_k - h(\hat{x}_k^i) - H_k^i (\hat{x}_k^- - \hat{x}_k^i) \right] \quad (28)$$

where $H_k^i = \partial h(x_k) / \partial x_k |_{x_k = \hat{x}_k^i}$ denotes the Jacobian matrix, and i represents the iteration index.

The analogous representation that $P_{xz} = P_k^- H_k^T$ and $P_z = H_k P_k^- H_k^T + R_k$ is obtained using the statistical linear error approach, which is used instead of the Jacobian matrix to reduce the computational complexity and avoid the linearization error caused by the Taylor first-order expansion. Furthermore, the enhanced Gauss–Newton iteration is displayed as follows.

$$\hat{x}_k^{i+1} = \hat{x}_k^- + P_{xz}^i (P_z^i)^{-1} \left[z_k - h(\hat{x}_k^i) - (P_{xz}^i)^T (P_k^-)^{-1} (\hat{x}_k^- - \hat{x}_k^i) \right] \quad (29)$$

4.3. The Iterated Orthogonal Simplex Cubature Kalman Filter

The enhanced Gauss–Newton iteration and the aforementioned altered cubature points are combined to create the IOSCKF method within the nonlinear Kalman filter architecture, and the specific calculation steps are as follows.

Step 1: Filter Initialization

$$\begin{cases} \hat{x}_0^+ = E(x_0) \\ P_0^+ = E \left[(x_0 - \hat{x}_0^+) (x_0 - \hat{x}_0^+)^T \right] \end{cases} \quad (30)$$

Step 2: Time Update

\hat{x}_{k-1}^+ and P_{k-1}^+ take the place of \bar{x} and P_x in Equation (24) to calculate the cubature points $\hat{x}_{k-1}^{(i)}$, which are disseminated nonlinearly using $f(\cdot)$, and the prior state \hat{x}_k^- and covariance P_k^- are calculated using the weights in Equation (23) on the basis of Equations (4) and (5), respectively.

Step 3: Measurement Update

Take the prior state $\hat{x}_k^0 = \hat{x}_k^-$ and covariance matrix $P_k^0 = P_k^-$ as the initial values of the iterated measurement update, and the estimated state and covariance matrix in the j th iteration express \hat{x}_k^j and P_k^j ($j = 1, 2, \dots, N_{\max}$), respectively. The cubature points $\hat{x}_k^{(i),j-1}$ are calculated using \hat{x}_k^{j-1} and P_k^{j-1} instead of \bar{x} and P_x in Equation (24), and propagated nonlinearly using $h(\cdot)$. The predicted measurement \hat{z}_k^j , the measurement covariance matrix $P_{z_r}^j$ and the cross covariance matrix P_{xz}^j are calculated using the weights in Equation (23) on the basis of the measurement update process of the Gaussian approximation filter, respectively. The Kalman filtering K_k^j and covariance matrix P_k^j are estimated to be equivalent to the Gaussian approximation filter, respectively, while the posterior state \hat{x}_k^j is calculated using the following iterated form.

$$\hat{x}_k^j = \hat{x}_k^- + K_k^j \left[z_k - h(\hat{x}_k^{j-1}) - (P_{xz}^j)^T (P_k^-)^{-1} (\hat{x}_k^- - \hat{x}_k^{j-1}) \right] \tag{31}$$

The above iteration format obtains a posterior estimate higher than the last one after each iteration step, and the system model $h(\cdot)$ is known and unchanged, so the whole iterative calculation process is convergent and stable.

The iteration termination condition is set as shown below.

$$\|\hat{x}_k^j - \hat{x}_k^{j-1}\| \leq \varepsilon \text{ or } j = N_{\max} \tag{32}$$

where ε and N_{\max} are the pre-set threshold and maximum iteration times.

The number of iterations in the end is presumed to be N , whereupon the estimated state and covariance matrix at time k are $\hat{x}_k^+ = \hat{x}_k^N$ and $P_k^+ = P_k^N$, respectively. Note that the calculational burden of the ICKF and CKF is within the same order of magnitude if N_{\max} is not too large, and in general, N_{\max} is no more than three.

Considering that all parameters in the algorithm are definite values, the algorithm has low sensitivity and strong robustness to parameters.

5. Simulations and Results

The proposed IOSCKF algorithm is applied in CV target tracking and spacecraft orbit tracking, respectively, in this section.

5.1. CV Target Tracking Simulation

The target is assumed to be in constant velocity (CV) motion; the state equation of the CV model in a two-dimensional case is described as follows:

$$X_k = F_{CV} X_{k-1} + G w_{k-1} \tag{33}$$

where $X_k = (x_k \ \dot{x}_k \ y_k \ \dot{y}_k)^T$ represents the position and velocity of the target at time index k , w_k denotes the process noise, and F_{CV} and G denote the state transformation matrix and the noise-driven matrix, which are defined, respectively as follows:

$$F_{CV} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \tag{34}$$

where T is the sampling interval.

In the target tracking system, the bearings-only measurement equation is written as follows.

$$Z_k = \arctan\left(\frac{y_k - y_r}{x_k - x_r}\right) + v_k \tag{35}$$

where Z_k is the radar measurement at time k , $(x_r \ y_r)$ is the location of the radar, and v_k is the measurement noise.

The proposed algorithm has the following two degraded forms: the cubature Kalman filter with the improved spherical simplex-radial rule (SCKF), and the SCKF with orthogonal transformation (OSCKF). In order to evaluate the performance of the proposed algorithm, it is compared to the general CKF, SSRCKF, SCKF, and OSCKF. This simulation shows the radar location $(x_r \ y_r) = (200 \text{ m} \ 300 \text{ m})$, the simulation time is 40 s, $T = 1$, the initial position of the target is $(x_0 \ y_0) = (100 \text{ m} \ 200 \text{ m})$ and the initial velocity is $(\dot{x}_0 \ \dot{y}_0) = (2 \text{ m/s} \ 20 \text{ m/s})$. The initial state and covariance matrix in the simulation are set to $\hat{x}_0^+ = (100 \text{ m} \ 2 \text{ m/s} \ 200 \text{ m} \ 20 \text{ m/s})^T$ and $P_0^+ = \text{diag}[0.01 \ 0.01 \ 0.01 \ 0.01]$, respectively. The standard deviation of the measurement noise is 0.1 deg. The maximum iteration time N_{\max} is set to 3. The Monte Carlo simulation is implemented 500 times, and the position accuracy is evaluated using the root mean square error (RMSE), defined as follows:

$$RMSE_{pos} = \sqrt{\frac{1}{N} \sum_{i=1}^N \left[(x_k - \hat{x}_{i,k}^+)^2 + (y_k - \hat{y}_{i,k}^+)^2 \right]} \tag{36}$$

where N is the execution number of Monte Carlo simulations, $(x_k \ y_k)$ is the real position of the target, and $(\hat{x}_{i,k}^+ \ \hat{y}_{i,k}^+)$ is the estimated position at time k in the i th Monte Carlo simulation. The velocity RMSE is defined similarly.

The simulation results are shown in Figure 1. The position RMSE and velocity RMSE of various filters are shown in Figure 1a,b, respectively, and the partial magnification is given to show the details of the curve. It can be seen from the figure that all the four filters have the ability to track the target trajectory, the tracking accuracy of the conventional CKF is significantly lower than that of the other filters, and the proposed IOSCKF achieves higher target tracking accuracy than the other filters. The average RMSE of the various filters is shown in Table 1; compared with the CKF, the estimated position accuracy of the SCKF is improved by 20.77%, indicating that the second-order Gauss–Laguerre rule used in the proposed modified spherical simplex-radial cubature rule has higher accuracy than the spherical-radial cubature rule. The accuracy of the OSCKF is 1% greater than that of the SCKF due to the orthogonal transformation of the cubature points reducing the high-order error components. While the filtering accuracy is further enhanced by the repetitive calculation of the measurement update, the IOSCKF’s accuracy is 0.5% higher than that of the OSCKF. Through the analysis of the simulation results, the validity of the proposed method is verified.

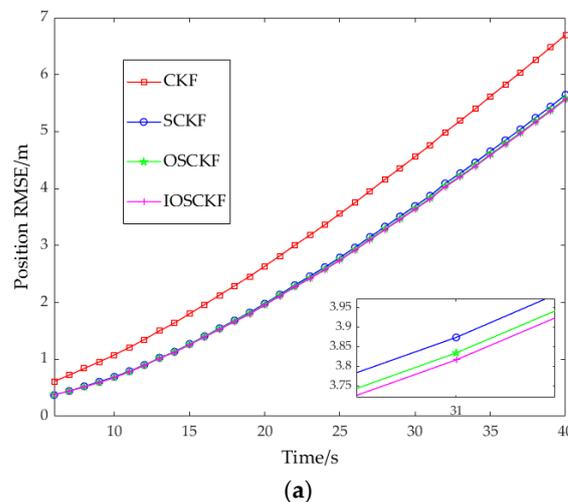


Figure 1. Cont.

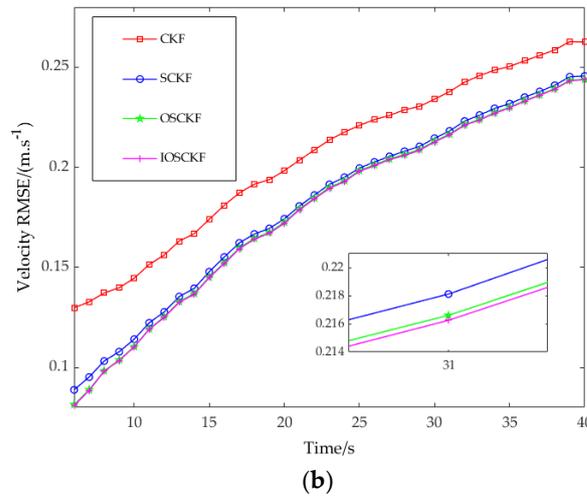


Figure 1. The RMSE of various filters. (a) The position RMSE. (b) The velocity RMSE.

Table 1. Average RMSE of various filters.

Filters	Position RMSE/m	Velocity RMSE/(m/s)
CKF	2.9379	0.1852
SCKF	2.3277	0.1643
OSCKF	2.3045	0.1622
IOSCKF	2.2929	0.1619

5.2. Spacecraft Target Tracking Simulation

5.2.1. Spacecraft Orbit Dynamics Model

For spacecrafts operating in low Earth orbit, Earth’s J2 term non-spherical perturbation and atmospheric drag perturbation are the most important perturbations. In the J2000 geocentric inertial coordinate system (O-XYZ), considering the above two perturbations, the orbital dynamics model of the spacecraft is as follows:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = \frac{\mu x}{r^3} \left[J_2 \left(\frac{R_e}{r} \right) \left(7.5 \frac{z^2}{r^2} - 1.5 \right) - 1 \right] + f_{ax} + f_x \\ \dot{v}_y = \frac{\mu y}{r^3} \left[J_2 \left(\frac{R_e}{r} \right) \left(7.5 \frac{z^2}{r^2} - 1.5 \right) - 1 \right] + f_{ay} + f_y \\ \dot{v}_z = \frac{\mu z}{r^3} \left[J_2 \left(\frac{R_e}{r} \right) \left(7.5 \frac{z^2}{r^2} - 4.5 \right) - 1 \right] + f_{az} + f_z \\ r = \sqrt{x^2 + y^2 + z^2} \end{cases} \quad (37)$$

where $X = (x \ y \ z)^T$ is the position vector of the spacecraft in O-XYZ; $\dot{X} = (v_x \ v_y \ v_z)^T$ is the velocity vector of the spacecraft in O-XYZ; J_2 is the harmonic term coefficient; μ is the gravitational constant of the Earth; R_e is the radius of the Earth; and $(f_x \ f_y \ f_z)^T$ is the sum of the high-order non-spherical perturbation of the Earth, three-body gravitational perturbation, and solar luminous pressure perturbation on the three coordinate axes, which can be equivalent to Gaussian white noise with a zero mean value in this study. $f_a = (f_{ax} \ f_{ay} \ f_{az})^T$ is the atmospheric drag perturbation, and its specific expression is as follows:

$$f_a = -\frac{1}{2} \frac{c_d A}{m} \rho_d v_{rel} v_{rel} \quad (38)$$

where c_d is the atmospheric drag coefficient, A/m is the surface mass ratio of the spacecraft, ρ_d is the atmospheric density, and v_{rel} is the relative speed between the spacecraft and the atmosphere, assuming that the atmosphere rotates with the Earth, as follows:

$$v_{rel} = \dot{\mathbf{X}} - \boldsymbol{\omega} \times \mathbf{X} \tag{39}$$

where $\boldsymbol{\omega} = \omega_e \times (0 \ 0 \ 1)^T$ is the angular velocity vector of the Earth rotation and ω_e is the angular velocity of the Earth rotation. Using the fourth-order Runge–Kutta method, Equation (32) is written in the following discrete form:

$$\mathbf{X}_k = f(\mathbf{X}_{k-1}) + \boldsymbol{w}_{k-1} \tag{40}$$

where $\mathbf{X}_k = [x_k \ y_k \ z_k \ v_{x,k} \ v_{y,k} \ v_{z,k}]^T$ is the track state at time k , and \boldsymbol{w}_k is the system noise.

5.2.2. Ground Radar Measurement Model

The radar measurement model is built in the radar horizon coordinate system ($O - X_h Y_h Z_h$), and the orbit model is built in the radar horizon coordinate system ($O - XYZ$), so it is necessary to use the WGS84 Earth fixed coordinate system ($O - X_e Y_e Z_e$) to achieve the conversion from $O - XYZ$ to $O - X_h Y_h Z_h$. Assuming that the orbital state of the spacecraft in $O - XYZ$ is $(\mathbf{X} \ \dot{\mathbf{X}})^T$, the orbital state in $O - X_e Y_e Z_e$ is $(\mathbf{X}_f \ \dot{\mathbf{X}}_f)^T$, and the orbital state in $O - X_h Y_h Z_h$ is $(\mathbf{X}_h \ \dot{\mathbf{X}}_h)^T$, the conversion of the orbital state is completed in two steps.

Step 1: Transition from $O - XYZ$ to $O - X_e Y_e Z_e$

The conversion matrix is $\mathbf{M}_J^W = \mathbf{M}_{Pw} \mathbf{M}_{Ro} \mathbf{M}_{Nu} \mathbf{M}_{Pr}$, where \mathbf{M}_{Pr} is the precession matrix, \mathbf{M}_{Nu} is the nutation matrix, \mathbf{M}_{Ro} is the Earth rotation matrix, and \mathbf{M}_{Pw} is the pole shift matrix; then, we can obtain:

$$\mathbf{X}_f = \mathbf{M}_J^W \mathbf{X} \tag{41}$$

$$\dot{\mathbf{X}}_f = \mathbf{M}_J^W \dot{\mathbf{X}} + \dot{\mathbf{M}}_J^W \mathbf{X} \tag{42}$$

In these formulas, the derivative of the transformation matrix $\dot{\mathbf{M}}_J^W$ is shown as follows:

$$\dot{\mathbf{M}}_J^W = \frac{d(\mathbf{M}_{Pw} \mathbf{M}_{Ro} \mathbf{M}_{Nu} \mathbf{M}_{Pr})}{dt} \approx \mathbf{M}_{Pw} \frac{d(\mathbf{M}_{Ro})}{dt} \mathbf{M}_{Nu} \mathbf{M}_{Pr} \tag{43}$$

where $d(\mathbf{M}_{Ro})/dt$ is the derivative matrix of the Earth rotation matrix as follows:

$$\frac{d(\mathbf{M}_{Ro})}{dt} = \begin{bmatrix} 0 & \omega_e & 0 \\ -\omega_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \mathbf{M}_{Ro} \tag{44}$$

Equations (36) and (37) are written in matrix form as:

$$\begin{bmatrix} \mathbf{X}_f \\ \dot{\mathbf{X}}_f \end{bmatrix} = \begin{bmatrix} \mathbf{M}_J^W & \mathbf{0} \\ \dot{\mathbf{M}}_J^W & \mathbf{M}_J^W \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \dot{\mathbf{X}} \end{bmatrix} \tag{45}$$

Step 2: Transition from $O - X_e Y_e Z_e$ to $O - X_h Y_h Z_h$

The conversion matrix \mathbf{M}_W^H is:

$$\mathbf{M}_W^H = \begin{bmatrix} -\sin \lambda & \cos \lambda & 0 \\ -\sin \varphi \cos \lambda & -\sin \varphi \sin \lambda & \cos \varphi \\ \cos \varphi \cos \lambda & \cos \varphi \sin \lambda & \sin \varphi \end{bmatrix} \tag{46}$$

where λ is the radar geocentric longitude and φ is the radar geocentric latitude, which can be converted to the geocentric coordinates of the radar $\mathbf{X}_c = (x_c \ y_c \ z_c)^T$ in $O - X_e Y_e Z_e$; so, we obtain:

$$\mathbf{X}_h = \mathbf{M}_W^H (\mathbf{X}_f - \mathbf{X}_c) \tag{47}$$

$$\dot{\mathbf{X}}_h = \mathbf{M}_W^H \dot{\mathbf{X}}_f \tag{48}$$

This is written in matrix form as below:

$$\begin{bmatrix} \mathbf{X}_h \\ \dot{\mathbf{X}}_h \end{bmatrix} = \begin{bmatrix} \mathbf{M}_W^H & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_W^H \end{bmatrix} \begin{bmatrix} \mathbf{X}_f \\ \dot{\mathbf{X}}_f \end{bmatrix} - \begin{bmatrix} \mathbf{M}_W^H \mathbf{X}_c \\ \mathbf{0} \end{bmatrix} \tag{49}$$

Substituting Equation (40) into Equation (44) yields:

$$\begin{bmatrix} \mathbf{X}_h \\ \dot{\mathbf{X}}_h \end{bmatrix} = \begin{bmatrix} \mathbf{M}_W^H & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_W^H \end{bmatrix} \begin{bmatrix} \mathbf{M}_J^W & \mathbf{0} \\ \dot{\mathbf{M}}_J^W & \mathbf{M}_J^W \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \dot{\mathbf{X}} \end{bmatrix} - \begin{bmatrix} \mathbf{M}_W^H \mathbf{X}_c \\ \mathbf{0} \end{bmatrix} \tag{50}$$

Thus, the transformation of the orbital state from $O - XYZ$ to $O - X_h Y_h Z_h$ is achieved, as shown in Equation (45). In order to obtain the relationship between the measured value and the orbit state, we write \mathbf{X}_h and $\dot{\mathbf{X}}_h$ in the form of a specific vector as $\mathbf{X}_h = (x_h \ y_h \ z_h)^T$ and $\dot{\mathbf{X}}_h = (\dot{x}_h \ \dot{y}_h \ \dot{z}_h)^T$; then, the geometric relationship between the radar ranging value R , velocity measurement value \dot{R} , azimuth angle A , and pitch angle E and the orbit state is as follows:

$$\begin{cases} R = \sqrt{x_h^2 + y_h^2 + z_h^2} \\ A = \arctan \frac{y_h}{x_h} \\ E = \arctan \frac{z_h}{\sqrt{x_h^2 + y_h^2}} \end{cases} \tag{51}$$

Equation (46) can be written as the following discrete measurement equation:

$$\mathbf{Z}_k = \mathbf{h}(\mathbf{X}_k) + \mathbf{v}_k \tag{52}$$

where $\mathbf{Z}_k = (R_k \ A_k \ E_k)^T$ is the measured value at time k , and \mathbf{v}_k is the measured noise.

Thus, Equations (35) and (47) constitute the state equation and measurement equation corresponding to Equation (11) for real-time filtering orbit determination.

5.2.3. Simulation Result and Analysis

Figure 2a shows the spacecraft ground simulator on which the algorithm of High-Precision Orbit Propagation (HPOP) is run, which takes into account the perturbation effects of the Earth’s high-order non-spherical gravity, atmospheric drag, solar light pressure, three-body gravity, and tides. Among them, the 21-layer Earth model, atmospheric drag coefficient $c_d = 2.2$, and spacecraft surface mass ratio $0.02 \text{ m}^2/\text{kg}$ are considered; the atmospheric density model adopts the Jacchia–Roberts model; and the solar luminous pressure coefficient $c_r = 1$. The orbital epoch of the spacecraft considered in the experiment is 16:00:00 (UTC) on 1 September 2016, and the six orbital roots are:

$$a = 7028.14 \text{ km}, e = 0, i = 80.7^\circ, \Omega = 63.8^\circ, \omega = 0^\circ, f = 107.1^\circ$$

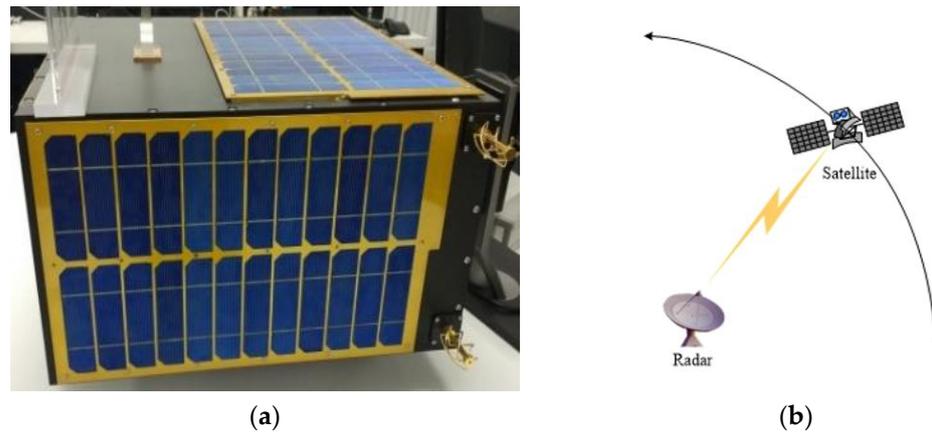


Figure 2. Schematic diagram of spacecraft tracking. (a) Spacecraft simulator. (b) Radar spacecraft tracking.

As shown in Figure 2b, the lowest measured elevation angle of the radar is 10° . It is assumed that the radar ranging error is 60 m and the angle error is 0.02° .

The algorithm proposed in this paper is compared with the SCKF and OSCKF, and the root mean square error (RMSE) is used to evaluate the spacecraft tracking accuracy. The Monte-Carlo simulation is run 200 times, and the results are shown in Figure 3. The mean positioning RMSE and velocity RMSE are listed in Table 2.

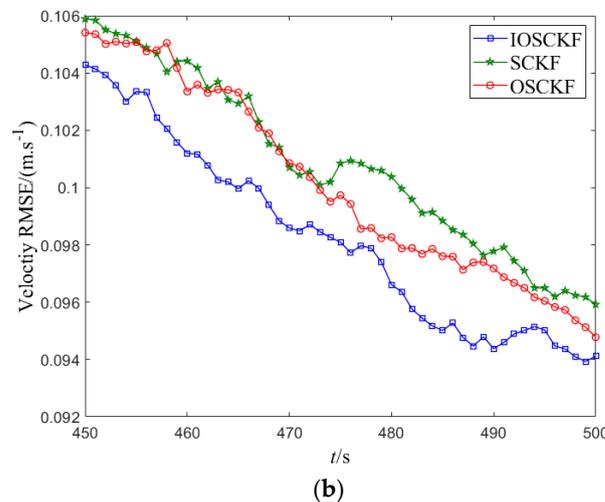
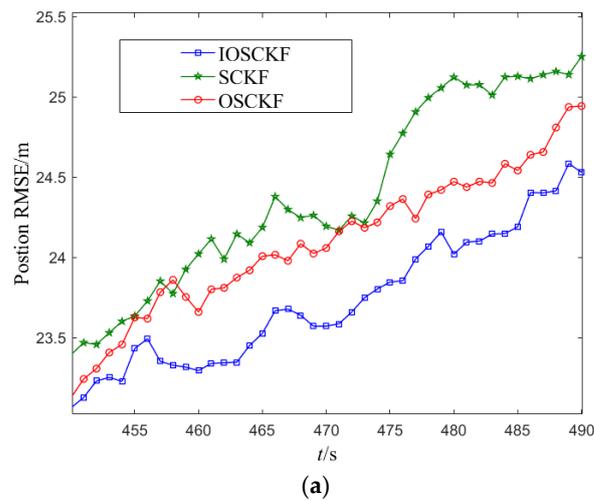


Figure 3. The spacecraft tracking RMSE of the various filters. (a) Position RMSE. (b) Velocity RMSE.

Table 2. Spacecraft tracking RMSE of the various filters.

Filters	Position RMSE/m	Velocity RMSE/(m/s)
SCKF	24.86	0.103
OSCKF	24.31	0.101
IOSCKF	23.65	0.098

It can be seen that the spacecraft tracking accuracy of the OSCKF is higher than that of the SCKF; specifically, the position tracking accuracy is increased by 2.21%, and the velocity tracking accuracy is increased by 1.94%, which indicates that the orthogonal transformation can improve the tracking accuracy. The spacecraft tracking accuracy of the IOSCKF is higher than that of the OSCKF; specifically, the position tracking accuracy is increased by 2.71%, and the velocity tracking accuracy is increased by 2.97%, which indicates that the tracking accuracy can be effectively improved by introducing iterative calculation into the measurement equation, thus verifying the effectiveness of the method presented in this paper.

6. Conclusions

A new iterated orthogonal simplex cubature Kalman filter (IOSCKF) is proposed to improve the nonlinear system filtering accuracy in target tracking. The new cubature points are determined using the orthogonal matrix after being modified using the spherical simplex-radial rule and the second-order Gauss–Laguerre quadrature rule. This lowers the high-order error terms. The Gauss–Newton iteration is used to maximize the posteriori state and covariance in the measurement update process until the iteration is finished. Target tracking simulations, including CV target tracking and spacecraft orbit tracking, are used to examine the performance of the proposed method, and the results reveal that the IOSCKF can achieve greater accuracy compared to the CKF, SCKF, and OSCKF. In spacecraft orbit tracking simulation, compared with the SCKF, the position tracking accuracy and velocity tracking accuracy of the OSCKF are increased by 2.21% and 1.94%, respectively, which indicate that the orthogonal transformation can improve the tracking accuracy. Furthermore, compared with the OSCKF, the position tracking accuracy and velocity tracking accuracy of the IOSCKF are increased by 2.71% and 2.97%, respectively, which indicate that the tracking accuracy can be effectively improved by introducing iterative calculation into the measurement equation, thus verifying the effectiveness of the method presented in this paper.

Author Contributions: Methodology, Z.L. and H.C.; Software, Z.L. and X.Y.; Investigation, X.Y. and L.L.; Resources, L.L.; Data curation, L.L.; Writing—original draft, Z.L. and H.C.; Funding acquisition, H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (Grant 62005320).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (Grant 62005320).

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

References

1. Wu, H.; Chen, S.; Yang, B.; Chen, K. Feedback robust cubature Kalman filter for target tracking using an angle sensor. *Sensors* **2016**, *16*, 629. [[CrossRef](#)] [[PubMed](#)]
2. Alomair, R.A.; Hassan, S.Z.; Abdelrahman, M.A.E.; Amin, A.H.; El-Shewy, E.K. New solitary optical solutions for the NLSE with δ -potential through Brownian process. *Results Phys.* **2022**, *40*, 105814. [[CrossRef](#)]
3. Abdelwahed, H.G.; Alsarhana, A.F.; El-Shewy, E.K.; Abdelrahman, M.A.E. Higher-Order Dispersive and Nonlinearity Modulations on the Propagating Optical Solitary Breather and Super Huge Waves. *Fractal Fract.* **2023**, *7*, 127. [[CrossRef](#)]
4. El-Shewy, E.K.; Alharbi, Y.F.; Abdelrahman, M.A.E. On the dynamical stochastic electrostatic noise fluctuations in Zakharov model. *Chaos Solitons Fractals* **2023**, *170*, 113324. [[CrossRef](#)]
5. Li, J.-Q.; Zhao, R.-H.; Chen, J.-L.; Zhao, C.-Y.; Zhu, Y.-P. Target tracking algorithm based on adaptive strong tracking particle filter. *IET Sci. Meas. Technol.* **2016**, *10*, 704–710.
6. Wu, H.; Chen, S.; Yang, B.; Luo, X. Range-parameterised orthogonal simplex cubature Kalman filter for bearings-only measurements. *IET Sci. Meas. Technol.* **2016**, *10*, 370–374. [[CrossRef](#)]
7. Leong, P.H.; Arulampalam, S.; Lamahewa, T.A.; Abhayapala, T.D. A Gaussian-sum based cubature Kalman filter for bearings-only tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 1161–1176. [[CrossRef](#)]
8. Swati, S.B. Cubature quadrature Kalman filter. *IET Signal Process.* **2013**, *7*, 533–541.
9. Ahmed, N.; Rutten, M.; Bessell, T.; Kanhere, S.S.; Gordon, N.; Jha, S. Detection and tracking using particle-filter-based wireless sensor networks. *IEEE Trans. Mob. Comput.* **2010**, *9*, 1332–1344. [[CrossRef](#)]
10. Leong, P.H.; Arulampalam, S.; Lamahewa, T.A.; Abhayapala, T.D. Gaussian-sum cubature Kalman filter with improved robustness for bearings-only tracking. *IEEE Signal Process. Lett.* **2014**, *21*, 513–517. [[CrossRef](#)]
11. Reif, K.; Gunther, S.; Yaz, E.; Unbehauen, R. Stochastic stability of the discrete-time extended Kalman filter. *IEEE Trans. Autom. Control* **1999**, *44*, 714–728. [[CrossRef](#)]
12. Liu, G.; Tian, G. Square-root sigma-point information consensus filters for distributed nonlinear estimation. *Sensors* **2017**, *17*, 800. [[CrossRef](#)] [[PubMed](#)]
13. Bell, B.M.; Cathey, F.W. The iterated Kalman filter update as a Gauss-Newton method. *IEEE Trans. Autom. Control* **1993**, *38*, 294–297. [[CrossRef](#)]
14. Julier, S.J.; Uhlmann, J.K.; Durrant-Whyte, H.F. A new method for nonlinear transformation of means and covariances in filters and estimators. *IEEE Trans. Autom. Control* **2000**, *45*, 477–482. [[CrossRef](#)]
15. Arasaratnam, I.; Haykin, S.; Elliott, R.J. Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature. *Proc. IEEE* **2007**, *95*, 953–975. [[CrossRef](#)]
16. Jia, B.; Xin, M.; Cheng, Y. Sparse-grid quadrature nonlinear filtering. *Automatica* **2012**, *48*, 327–341. [[CrossRef](#)]
17. Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [[CrossRef](#)]
18. Arasaratnam, I.; Haykin, S.; Hurd, T.R. Cubature Kalman filtering for continuous-discrete systems: Theory and simulations. *IEEE Trans. Signal Process.* **2010**, *58*, 4977–4992. [[CrossRef](#)]
19. Potnuru, D.; Chandra, K.P.B.; Arasaratnam, I.; Gu, D.-W.; Mary, K.A.; Ch, S.B. Derivative-free square-root cubature Kalman filter for non-linear brushless DC motors. *IET Electr. Power Appl.* **2016**, *10*, 419–429. [[CrossRef](#)]
20. Chang, L.; Hu, B.; Li, A.; Qin, F. Transformed unscented Kalman filter. *IEEE Trans. Autom. Control* **2013**, *58*, 252–257. [[CrossRef](#)]
21. Zhang, L.; Yang, H.; Lu, H.; Zhang, S.; Cai, H.; Qian, S. Cubature Kalman filtering for relative spacecraft attitude and position estimation. *Acta Astronaut.* **2014**, *105*, 254–264. [[CrossRef](#)]
22. Li, W.; Jia, Y. Location of mobile station with maneuvers using an IMM-based cubature Kalman filter. *IEEE Trans. Ind. Electron.* **2012**, *59*, 4338–4348. [[CrossRef](#)]
23. 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](#)]
24. Zhang, K.; Shan, G. Model-switched Gaussian sum cubature Kalman filter for attitude angle-aided three-dimensional target tracking. *IET Radar Sonar Navig.* **2015**, *9*, 531–539. [[CrossRef](#)]
25. Zarei, J.; Shokri, E. Convergence analysis of non-linear filtering based on cubature Kalman filter. *IET Sci. Meas. Technol.* **2015**, *9*, 294–305. [[CrossRef](#)]
26. Wu, P.; Li, X.; Bo, Y. Iterated square root unscented Kalman filter for maneuvering target tracking using TDOA measurements. *Int. J. Control Autom. Syst.* **2013**, *11*, 761–767. [[CrossRef](#)]
27. Zhan, R.; Wan, J. Iterated unscented Kalman filter for passive target tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 1155–1163. [[CrossRef](#)]
28. Chang, G.; Xu, T.; Wang, Q. Alternative framework for the iterated unscented Kalman filter. *IET Signal Process.* **2017**, *11*, 258–264. [[CrossRef](#)]
29. Cui, B.; Chen, X.; Xu, Y.; Huang, H.; Liu, X. Performance analysis of improved iterated cubature Kalman filter and its application to GNSS/INS. *ISA Trans.* **2017**, *66*, 460–468. [[CrossRef](#)]
30. Jia, B.; Xin, M.; Cheng, Y. High-degree cubature Kalman filter. *Automatica* **2013**, *49*, 510–518. [[CrossRef](#)]
31. Wang, S.; Feng, J.; Tse, C.K. Spherical simplex-radial cubature Kalman filter. *IEEE Signal Process. Lett.* **2014**, *21*, 43–46. [[CrossRef](#)]

32. Singh, A.K.; Bhaumik, S. Higher degree cubature quadrature Kalman filter. *Int. J. Control Autom. Syst.* **2015**, *13*, 1097–1105. [[CrossRef](#)]
33. Wang, C.; Zhang, J.; Mu, J. Maximum likelihood-based iterated divided difference filter for nonlinear systems from discrete noisy measurements. *Sensors* **2012**, *12*, 8912–8929. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.