

Article

Strong Tracking Spherical Simplex-Radial Cubature Kalman Filter for Maneuvering Target Tracking

Hua Liu and Wen Wu *

Ministerial Key Laboratory of JGMT, Nanjing University of Science and Technology, Nanjing 210094, China; peterliuh@126.com

* Correspondence: wuwen@njust.edu.cn; Tel.: +86-139-5102-5028

Academic Editors: Xue-Bo Jin, Shuli Sun, Hong Wei and Feng-Bao Yang

Received: 20 February 2017; Accepted: 28 March 2017; Published: 31 March 2017

Abstract: Conventional spherical simplex-radial cubature Kalman filter (SSRCKF) for maneuvering target tracking may decline in accuracy and even diverge when a target makes abrupt state changes. To overcome this problem, a novel algorithm named strong tracking spherical simplex-radial cubature Kalman filter (STSSRCKF) is proposed in this paper. The proposed algorithm uses the spherical simplex-radial (SSR) rule to obtain a higher accuracy than cubature Kalman filter (CKF) algorithm. Meanwhile, by introducing strong tracking filter (STF) into SSRCKF and modifying the predicted states' error covariance with a time-varying fading factor, the gain matrix is adjusted on line so that the robustness of the filter and the capability of dealing with uncertainty factors is improved. In this way, the proposed algorithm has the advantages of both STF's strong robustness and SSRCKF's high accuracy. Finally, a maneuvering target tracking problem with abrupt state changes is used to test the performance of the proposed filter. Simulation results show that the STSSRCKF algorithm can get better estimation accuracy and greater robustness for maneuvering target tracking.

Keywords: maneuvering target tracking; spherical simplex-radial rule; cubature Kalman filter; fading factor; strong tracking filter

1. Introduction

Maneuvering target tracking has drawn increasing attention because of its widespread application in areas such as radar tracking, aircrafts surveillance, and spacecraft orbit control [1,2]. For maneuvering target tracking, many algorithms are developed and grouped into two types. One type is to improve the accuracy of the motion model, such as multiple-model (MM) methods [3], optimization of multiple model neural filter [4], current statistical (CS) model [5,6], and so on. The other type is to detect the target maneuverability and then to cope with it effectively, such as strong tracking filter (STF) [7], tracking algorithm based on maneuvering detection [8], and so on. In these methods, the performance of the filter is an important factor affecting the performance of these methods. Therefore, improving the accuracy of the filter is also a useful method to improve the performance of maneuvering target tracking. Thus, a large number of nonlinear filters have been developed. Among these algorithms, the extended Kalman filter (EKF) [9] is one of the earliest and most widely used nonlinear filters. The EKF uses a linearization technique, based on the first-order Taylor series expansion, and approximates the nonlinear system. However, EKF has some limitations, such as complex Jacobian matrix calculations and poor accuracy in estimating the states of the strongly nonlinear system.

As better alternatives to the EKF, many nonlinear filters based on the idea of Bayesian theory have been proposed. One popular approach for the nonlinear non-Gaussian filtering problem is to use sequential Monte Carlo methods. The most famous method is known as particle filter (PF) [10–13]. The key idea of PF is to represent the posterior distribution by a set of random samples

and to calculate estimates based on these samples and weights. Although the PF can provide good performance, the computational cost is very high and suffers from the curse of the dimensionality problem. These shortcomings restrict their applications in a real-time system. A different approach for nonlinear filtering is based on the point-based filtering technique that approximates intractable integrals encountered by a set of deterministically sampled points. Compared with the Monte Carlo numerical integration that relies on randomly sampled points, the deterministic point-based method has lower computational complexity with high accuracy. The type of filter includes the unscented Kalman filter (UKF) [14], Gauss-Hermite filter (GHF) [15], central difference filter (CDF) [16], etc. Among these methods, the well-known filter is UKF. The UKF uses unscented transform (UT) to capture the mean and covariance of a Gaussian density. It is shown that the UKF has better performance than the EKF. Besides its higher approximation accuracy, this UKF can avoid the cumbersome evaluation of Jacobian and Hessian matrices, making the algorithm easier to implement. Nevertheless, the unscented transform of the UKF is potentially unstable [17], which restricts its practical applications. Apart from the aforementioned filters, the cubature Kalman filter (CKF) has been proposed [17,18] by Arasaratnam and Haykin. Making use of the third-degree spherical-radial cubature rule, the CKF is reported to be more flexible in implementation form and more stable than UKF. In addition, Jia et al. [19] proposed the high-degree CKF where the number of sample points increases rapidly with the increase of the degree or state dimension. To further improve estimation accuracy with low complexity, a new nonlinear filter named spherical simplex-radial cubature filter (SSRCKF) is developed in [20]. The new class of CKF is based on the simplex spherical radial (SSR) rule, which improves the accuracy of CKF with only two more cubature points necessary.

Although the SSRCKF can achieve good accuracy in tracking non-maneuvering or weak maneuvering targets, it may lose the tracking ability to the abrupt state change when the system reaches the stable state. This is because the reaction of the gain matrix is delayed to the sudden change of the prediction error. To tackle the problem mentioned above, a new algorithm called strong tracking spherical simplex-radial cubature Kalman filter (STSSRCKF) is proposed in this paper. The STSSRCKF is developed based on the combination of strong tracking filter (STF) [7,21,22] and SSRCKF. The new algorithm using the strong tracking idea and the fading factor based on the residual to modify the prior covariance matrix quickly. Thus, the gain of the filter can be adjusted in real time to enhance tracking capacity for the maneuvering target. In addition, the algorithm can also keep a normal tracking accuracy for weak maneuvering targets. Compared with the STF, strong tracking unscented Kalman filter (STUKF) [23], strong tracking cubature Kalman filter (STCKF) [24] and SSRCKF, the proposed algorithm has a good accuracy and robust advantage over a wide range of maneuver. The performance of the proposed filter is demonstrated by the simulation.

The remainder of this paper is organized as follows. The overview of the background theory is presented in Section 2. The proposed algorithm is developed in Section 3. Simulation results and performance comparisons are presented in Section 4. Finally, conclusions are provided in Section 5.

2. A Review of UKF and CKF

The nonlinear discrete-time system is represented by

$$\begin{cases} \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \end{cases} \quad (1)$$

where $k \in N$ denotes discrete time, $\mathbf{f}(\cdot)$ represents the nonlinear function, $\mathbf{h}(\cdot)$ represents the measurement function. $\mathbf{x}_k \in \mathbf{R}^n$ is the state vector of system, $\mathbf{z}_k \in \mathbf{R}^m$ is the measurement, $\mathbf{w}_k \in \mathbf{R}^n$ is the process noise vector, and $\mathbf{v}_k \in \mathbf{R}^m$ is the measurement noise vector. The process \mathbf{w}_k and measurement noise \mathbf{v}_k are uncorrelated zero-mean Gaussian white sequences and have zero cross-correlation with each other, represented as $\mathbf{w}_k \sim N(0, \mathbf{Q}_k)$ and $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$, respectively.

Under the Gaussian assumption in the Bayesian filtering framework, the key problem of the nonlinear filtering problem is to calculate the multi-dimensional integrals. However, in most cases,

the multi-dimensional integrals cannot be solved analytically. As a result, several approximation methods have been proposed, such as the unscented transformation (UT) and the cubature rule.

The UT with $2n + 1$ the sigma points χ_i and corresponding weights is chosen as

$$\left\{ \begin{array}{l} \chi_0 = \mathbf{x}_{k|k} \\ \chi_i = \mathbf{x}_{k|k} + \left[\sqrt{(n + \lambda)\mathbf{P}_{k|k}} \right]_i \quad (i = 1, \dots, n) \\ \chi_i = \mathbf{x}_{k|k} - \left[\sqrt{(n + \lambda)\mathbf{P}_{k|k}} \right]_i \quad (i = n + 1, \dots, 2n) \\ \omega_m^{(0)} = \lambda / (n + \lambda) \\ \omega_c^{(0)} = \lambda / (n + \lambda) + (1 - \alpha^2 + \beta) \\ \omega_m^{(i)} = \omega_c^{(i)} = 1/2(n + \lambda) \quad (i = 1, \dots, 2n) \end{array} \right. \quad (2)$$

where $[\mathbf{P}_{k|k}]_i$ is the i th column of the matrix square root of $\mathbf{P}_{k|k}$, n is the dimension of state. $\lambda = \alpha^2(n + \kappa) - n$ is the scaling parameter; α determines the spread of the sigma points around $\mathbf{x}_{k|k}$. The positive constants β and κ are used as parameters of the method.

The third-degree cubature rule with $2n$ cubature points and weights is given by:

$$\left\{ \begin{array}{l} \chi_i = \mathbf{x}_{k|k} + \left[\sqrt{(n/2)\mathbf{P}_{k|k}} \right]_i \quad (i = 1, \dots, n) \\ \chi_i = \mathbf{x}_{k|k} - \left[\sqrt{(n/2)\mathbf{P}_{k|k}} \right]_i \quad (i = n + 1, \dots, 2n) \\ \omega_m^{(i)} = \omega_c^{(i)} = 1/2n \quad (i = 1, \dots, 2n) \end{array} \right. \quad (3)$$

As indicated above, the main difference between the UT used in UKF and the third-degree cubature rule used in CKF is that the UT has one more point in the center with a tune parameter κ . If the parameter κ is set to zero, the sigma points set will evolve into the cubature points set and the UKF becomes identical to the CKF. For UKF, the scaling parameter κ is always set to $n - 3$. Based on this point, for high-dimensional problems ($n > 3$), it will lead to the negative weight of the center point. The presence of the negative weight may lead the covariance matrix to become non-positively defined. Thus, the cubature rule is more stable than the UT. In summary, the CKF is virtually a special case of UKF and the CKF has better numerical stability than UKF.

3. Strong Tracking Spherical Simplex-Radial Cubature Kalman Filter

The heart of the spherical simplex cubature Kalman filter is the spherical-radial cubature rule. The spherical-radial cubature rule does not approximate the nonlinear function, but it can approximate the integral of the form (nonlinear function \times Gaussian) using weighted quadrature point sets. The integral with the standard Gaussian distribution $N(\mathbf{x}; 0, \mathbf{I})$ can be approximated by the quadrature

$$\int_{\mathbb{R}^n} f(\mathbf{x})N(\mathbf{x}; 0, \mathbf{I})d\mathbf{x} \approx \sum_{i=1}^m \omega_i f(\gamma_i) \quad (4)$$

where m is the total number of quadrature points in the state-space \mathbb{R}^n , $\{\gamma_i, \omega_i\}_i^m$ is a set of quadrature points and corresponding weights. The general Gaussian integral $\int_{\mathbb{R}^n} f(\mathbf{x})N(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})d\mathbf{x}$ can be approximated by the following transformation

$$\begin{aligned} \int_{\mathbb{R}^n} f(\mathbf{x})N(\mathbf{x}; \hat{\mathbf{x}}, \mathbf{P})d\mathbf{x} &= \int_{\mathbb{R}^n} f(\sqrt{\mathbf{P}}\mathbf{x} + \hat{\mathbf{x}})N(\mathbf{x}; 0, \mathbf{I})d\mathbf{x} \\ &\approx \sum_{i=1}^m \omega_i f(\sqrt{\mathbf{P}}\gamma_i + \hat{\mathbf{x}}) \end{aligned} \quad (5)$$

The computational complexity of the numerical integration is proportional to the number of quadrature points, and the accuracy of the numerical integration rule is usually assessed by the polynomial approximation degrees.

3.1. Review of the Third-Degree Spherical Simplex-Radial Cubature Rule

The SSRCKF algorithm has the same structure as the general Gaussian approximation filters, such as the CKF, but uses the third-degree spherical simplex-radial cubature rule to calculate the Gaussian weight integral $I(\mathbf{f}) = \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x})N(\mathbf{x}; 0, \mathbf{I})d\mathbf{x}$. By using the spherical simplex-radial cubature rule, the SSRCKF method can get more accurate estimation than CKF. In the third-degree spherical simplex-radial cubature rule, the following integral is considered [19]:

$$I(\mathbf{f}) = \int_{\mathbb{R}^n} \mathbf{f}(\mathbf{x}) \exp(-\mathbf{x}^T \mathbf{x}) d\mathbf{x} \quad (6)$$

where $\mathbf{f}(\cdot)$ is arbitrary nonlinear function, \mathbb{R}^n is the integral domain. To calculate the above integral, let $\mathbf{x} = r\mathbf{s}$ ($\mathbf{s}^T \mathbf{s} = 1, r = \sqrt{\mathbf{x}^T \mathbf{x}}$). Equation (6) can be transformed into the spherical-radial coordinate system

$$I(\mathbf{f}) = \int_0^\infty \int_{U_n} \mathbf{f}(r\mathbf{s}) r^{n-1} \exp(-r^2) d\sigma(\mathbf{s}) dr \quad (7)$$

where $\mathbf{s} = [s_1, s_2, \dots, s_n]^T$, $U_n = \{\mathbf{s} \in \mathbb{R}^n : s_1^2 + s_2^2 + \dots + s_n^2 = 1\}$ is the spherical surface, and $\sigma(\cdot)$ is the area element on U_n . Then, the Equation (7) can be decomposed into the spherical integral $S(r) = \int_{U_n} \mathbf{f}(r\mathbf{s}) d\sigma(\mathbf{s})$ and the radial integral $I(\mathbf{f}) = \int_0^\infty S(r) r^{n-1} \exp(-r^2) dr$.

3.1.1. Spherical Simplex Rule

As can be seen from the literature [25], the spherical integral $\int_{U_n} \mathbf{f}(r\mathbf{s}) d\sigma(\mathbf{s})$ can be approximated by the transformation group of the regular n -simplex with vertices \mathbf{a}_j . P_0 The third-degree spherical simplex rule with $2n + 2$ quadrature points is given by

$$\begin{aligned} S(r) &= \frac{A_n}{2(n+1)} \sum_{j=1}^{n+1} (\mathbf{f}(r\mathbf{a}_j) + \mathbf{f}(-r\mathbf{a}_j)) \\ &= \sum_{j=1}^{N_s} \omega_{s,j} \mathbf{f}(ry_j) \end{aligned} \quad (8)$$

where $A_n = 2\sqrt{\pi^n} / \Gamma^n(1/2)$, $N_s = 2n + 2$.

3.1.2. Radial Rule

The radial integral $\int_0^\infty S(r) r^{n-1} \exp(-r^2) dr$ can be calculated by the following moment matching equation

$$\int_0^\infty S(r) r^{n-1} \exp(-r^2) dr = \sum_{i=1}^{N_r} \omega_{r,i} S(r_i) \quad (9)$$

where $S(r) = r^l$ is a monomial in r , with l an even integer. Using the moment method with the minimum number of points, the third-degree radial rule ($N_r = 1$) can be derived. From Equation (9) we can obtain the moments' equations as

$$\begin{cases} \omega_{r,1} r_1^0 = \frac{1}{2} \Gamma\left(\frac{n}{2}\right) \\ \omega_{r,1} r_1^2 = \frac{1}{2} \Gamma\left(\frac{n+2}{2}\right) = \frac{n}{4} \Gamma\left(\frac{n}{2}\right) \end{cases} \quad (10)$$

By solving Equation (10), the points and weights for the third-degree radial rule are given by

$$\begin{cases} r_1 = \sqrt{n/2} \\ \omega_{r,1} = \Gamma(n/2)/2 \end{cases} \quad (11)$$

3.1.3. Spherical Simplex-Radial Rule

By using Equations (7), (8) and (11), the third-degree spherical simplex-cubature rule ($N_r = 1$, $N_s = 2n + 2$) is given by

$$\begin{aligned} & \int_{\mathbb{R}^n} f(\mathbf{x}) \mathcal{N}(\mathbf{x}, \mathbf{0}, \mathbf{I}) d\mathbf{x} \\ &= \frac{1}{\sqrt{\pi^n}} \int_{\mathbb{R}^n} f(\sqrt{2}\mathbf{x}) \exp(-\mathbf{x}^T \mathbf{x}) d\mathbf{x} \\ &\approx \frac{1}{\sqrt{\pi^n}} \sum_{i=1}^{N_r} \sum_{j=1}^{N_s} \omega_{r,i} \omega_{s,j} f(\sqrt{2}r_i \mathbf{s}_j) \\ &= \frac{1}{2(n+1)} \times \left(\sum_{j=1}^{n+1} f(\sqrt{n}\mathbf{a}_j) + \sum_{j=n+2}^{2n+2} f(-\sqrt{n}\mathbf{a}_j) \right) \\ &= \sum_{k=1}^m \omega_k f(\boldsymbol{\zeta}_k) \end{aligned} \quad (12)$$

where $m = 2n + 2$, $\boldsymbol{\zeta}_k = \sqrt{n}[\mathbf{a}, -\mathbf{a}]_k$ and $\omega_k = 1/(2n + 2)$ are the corresponding weights.

The steps of SSRCKF algorithm for the nonlinear system can be found in the literature [17].

3.2. Strong Tracking Filter

To improve the performance of EKF, a concept of STF was proposed by Zhou and Frank [7]. They proved that a filter can obtain the strong tracking estimation of the state can have the strong tracking performance only if the filter satisfies the orthogonal principle [7]. In strong tracking, the time-varying suboptimal fading factor is incorporated, which online adjusts the covariance of the predicted state. In this way, the algorithm has the ability to track abrupt state change and strong robustness against mode uncertainties. The algorithm has the following steps [21]:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}_k(\hat{\mathbf{x}}_{k-1|k-1}) \\ \mathbf{P}_{k|k-1} &= \lambda_k \mathbf{F}_{k|k-1} \mathbf{P}_{k-1|k-1} \mathbf{F}_{k|k-1}^T + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})) \\ \mathbf{P}_{k|k} &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_{k|k-1} \end{aligned} \quad (13)$$

where $\mathbf{F}_{k|k-1}$ and \mathbf{H}_k are the process matrix and measure matrix, respectively. The suboptimal time-varying fading factor λ_k is given by

$$\lambda_k = \begin{cases} c_k, & c_k \geq 1 \\ 1, & c_k < 1 \end{cases}, c_k = \frac{\text{tr}[\mathbf{N}_k]}{\text{tr}[\mathbf{M}_k]} \quad (14)$$

$$\mathbf{N}_k = \mathbf{V}_k - \mathbf{H}_k \mathbf{Q}_{k-1} \mathbf{H}_k^T - \beta \mathbf{R}_k \quad (15)$$

$$\mathbf{M}_k = \mathbf{H}_k \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T \mathbf{H}_k^T \quad (16)$$

$$\mathbf{V}_k = \begin{cases} \mathbf{v}_0 \mathbf{v}_0^T & k = 0 \\ \frac{\rho \mathbf{V}_{k-1} + \mathbf{v}_k \mathbf{v}_k^T}{1 + \rho} & k \geq 1 \end{cases} \quad (17)$$

where $\text{tr}[\cdot]$ is the trace operation, $\mathbf{v}_k = \mathbf{z}_k - \hat{\mathbf{z}}_{k|k-1}$ denotes the measurement residual vector; $\beta \geq 1$ is the softening factor, which can improve the smoothness of state estimation; $0 < \rho \leq 1$ is the forgetting factor. In generally, the parameters β and ρ are chosen as 4.5 and 0.95, respectively [26,27].

3.3. Equivalent Expression of the Fading Factor

As we know, STF need calculate the linearization of the nonlinear measurement matrix (Hessian matrix). However, SSRCKF is not necessary to compute the Hessian matrix. So we give the equivalent expression of STF, which need not calculate the Hessian matrix. Suppose $\mathbf{P}_{k|k-1}^l$ is the state error covariance matrix before introducing fading factor, $\mathbf{P}_{zz,k|k-1}^l$ is the measurement covariance matrix and $\mathbf{P}_{xz,k|k-1}^l$ is cross-covariance matrix, Equations (15) and (16) have the following equivalent expressions:

$$\mathbf{N}_k = \mathbf{V}_k - (\mathbf{P}_{xz,k|k-1}^l)^T (\mathbf{P}_{k|k-1}^l)^{-1} \mathbf{Q}_{k-1} (\mathbf{P}_{k|k-1}^l)^{-1} (\mathbf{P}_{xz,k|k-1}^l) - \beta \mathbf{R}_k \quad (18)$$

$$\mathbf{M}_k = \mathbf{P}_{zz,k|k-1}^l - \mathbf{V}_k + \mathbf{N}_k + (\beta - 1) \mathbf{R}_k \quad (19)$$

The new fading factor can be obtained through Equations (14) and (17)–(19). It can be verified from Equations (18) and (19) that the calculation of suboptimal fading factor in the Equation expression does not need to compute any Jacobian matrix.

3.4. Steps of the STSSRCKF

Based on the previous sections, the strong tracking spherical-simplex cubature Kalman filtering (STSSRCKF) can adjust the prediction error covariance matrix by introducing a suboptimal factor. Hence, the robustness and real-time tracking ability are provided in the STSSRCKF algorithm. The initial state is assumed to be Gaussian distribution with $\hat{\mathbf{x}}_{0|0}$ and $\mathbf{P}_{0|0}$. The computation steps of the third-degree strong tracking spherical simplex-radial cubature Kalman filter is summarized as follows:

Step 1. Give the state estimate $\hat{\mathbf{x}}_{k-1|k-1}$ and the error covariance matrix $\mathbf{P}_{k-1|k-1}$;

Step 2. State estimate prediction:

The cubature points are obtained as

$$\chi_{j,k|k-1}^l = \hat{\mathbf{x}}_{k|k-1} + \text{chol}(\mathbf{P}_{k|k-1})^T \boldsymbol{\zeta}_j \quad (20)$$

where $\text{chol}(\cdot)$ is the Cholesky factorization.

Propagate the cubature points, the predicted state $\mathbf{x}_{k|k-1}$, and the predicted covariance $\mathbf{P}_{k|k-1}^l$ without the fading factor are given as

$$\chi_{j,k|k-1}^* = f(\chi_{j,k-1}) \quad (21)$$

$$\hat{\mathbf{x}}_{k|k-1} = \sum_{j=1}^m \omega_j \chi_{j,k|k-1}^* \quad (22)$$

$$\mathbf{P}_{k|k-1}^l = \sum_{j=1}^m \omega_j (\chi_{j,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1})(\chi_{j,k|k-1}^* - \hat{\mathbf{x}}_{k|k-1})^T + \mathbf{Q}_{k-1} \quad (23)$$

where \mathbf{Q}_{k-1} is the covariance matrix of process noise.

Step 3. Calculation of the fading factor λ_k :

Using the predicted state $\hat{\mathbf{x}}_{k|k-1}$ and the predicted covariance $\mathbf{P}_{k|k-1}^l$, the innovation covariance $\mathbf{P}_{zz,k|k-1}^l$ and the cross covariance $\mathbf{P}_{xz,k|k-1}^l$ can be calculated as

$$\mathbf{z}_{j,k|k-1}^l = \mathbf{h}(\boldsymbol{\chi}_{j,k|k-1}^*) \quad (24)$$

$$\hat{\mathbf{z}}_{k|k-1}^l = \sum_{j=1}^m \omega_j \mathbf{z}_{j,k|k-1}^l \quad (25)$$

$$\mathbf{P}_{xz,k|k-1}^l = \sum_{j=1}^m \omega_j (\boldsymbol{\chi}_{j,k|k-1}^l - \hat{\mathbf{x}}_{k|k-1}) (\mathbf{z}_{j,k|k-1}^l - \hat{\mathbf{z}}_{k|k-1}^l)^T \quad (26)$$

$$\mathbf{P}_{zz,k|k-1}^l = \sum_{j=1}^m \omega_j (\mathbf{z}_{j,k|k-1}^l - \hat{\mathbf{z}}_{k|k-1}^l) (\mathbf{z}_{j,k|k-1}^l - \hat{\mathbf{z}}_{k|k-1}^l)^T + \mathbf{R}_k \quad (27)$$

The fading factor λ_k can be calculated by using Equations (14) and (17)–(19).

Step 4. Measurement updating modified by the fading factor:

The modified prediction covariance $\mathbf{P}'_{k|k-1}$ can be updated by

$$\mathbf{P}'_{k|k-1} = \lambda_k (\mathbf{P}_{k|k-1}^l - \mathbf{Q}_{k-1}) + \mathbf{Q}_{k-1} \quad (28)$$

By utilizing the predicted state estimate $\hat{\mathbf{x}}_{k|k-1}$ and the modified predicted covariance $\mathbf{P}'_{k|k-1}$ with the fading factor λ_k , the modified predicted measurement $\hat{\mathbf{z}}'_{k|k-1}$, the modified cross covariance and the modified innovation covariance $\mathbf{P}'_{zz,k|k-1}$ can be calculated as follows

$$\boldsymbol{\chi}'_{j,k|k-1} = \text{chol}(\mathbf{P}'_{k|k-1}) \boldsymbol{\zeta}_i + \hat{\mathbf{x}}_{k|k-1} \quad (29)$$

$$\mathbf{z}'_{j,k|k-1} = \mathbf{h}(\boldsymbol{\chi}'_{j,k|k-1}) \quad (30)$$

$$\hat{\mathbf{z}}'_{k|k-1} = \sum_{j=1}^m \omega_j \mathbf{z}'_{j,k|k-1} \quad (31)$$

$$\mathbf{P}'_{xz,k|k-1} = \sum_{j=1}^m \omega_j (\boldsymbol{\chi}'_{j,k|k-1} - \hat{\mathbf{x}}_{k|k-1}) (\mathbf{z}'_{j,k|k-1} - \hat{\mathbf{z}}'_{k|k-1})^T \quad (32)$$

$$\mathbf{P}'_{zz,k|k-1} = \sum_{j=1}^m \omega_j (\mathbf{z}'_{j,k|k-1} - \hat{\mathbf{z}}'_{k|k-1}) (\mathbf{z}'_{j,k|k-1} - \hat{\mathbf{z}}'_{k|k-1})^T + \mathbf{R}_k \quad (33)$$

Step 5. Estimation results:

The state estimate $\hat{\mathbf{x}}_k$ and the covariance \mathbf{P}_k at time k are calculate as follows

$$\mathbf{K}_k = \mathbf{P}'_{xz,k|k-1} (\mathbf{P}'_{zz,k|k-1})^{-1} \quad (34)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}'_{k|k-1}) \quad (35)$$

$$\mathbf{P}_{k|k} = \mathbf{P}'_{k|k-1} - \mathbf{K}_k \mathbf{P}'_{zz,k|k-1} \mathbf{K}_k^T \quad (36)$$

The STSSRCKF combines the advantages of STF and SSRCKF. Then the STSSRCKF has strong robustness against model uncertainties and good real-time state tracking capability [28]. Moreover, the STSSRCKF algorithm eliminates the cumbersome evaluation of Jacobian/Hessian matrices, its numerical stability and estimated accuracy are significantly improved.

4. Simulation and Results

The effectiveness of the proposed algorithm will be illustrated through two examples of maneuvering target tracking. Taking the root mean square error (RMSE) and accumulative RMSE (ARMSE), the study compared the STSSRCKF algorithm with the EKF algorithm and the SSRCKF algorithm.

4.1. Tracking Model and Measurement Model

The constant acceleration (CA) model is a common tool for tracking target modeling. The state Equation of CA model in two-dimensional case is described as follow:

$$\mathbf{X}_k = \text{diag}[\Phi_{CA}, \Phi_{CA}]\mathbf{X}_{k-1} + \mathbf{G}_{CA}\mathbf{V}_{k-1} + \mathbf{w}_{k-1} \quad (37)$$

where $\mathbf{X}_{k-1} = [x_{k-1}, \dot{x}_{k-1}, \ddot{x}_{k-1}, y_{k-1}, \dot{y}_{k-1}, \ddot{y}_{k-1}]^T$ is the target state at time $k-1$, (x_{k-1}, y_{k-1}) , $(\dot{x}_{k-1}, \dot{y}_{k-1})$ and $(\ddot{x}_{k-1}, \ddot{y}_{k-1})$ represent the target position, velocity and acceleration in the x and y coordinate at time $k-1$, respectively; $\text{diag}[\Phi_{CA}, \Phi_{CA}]$ is the state transition matrix, \mathbf{G}_{k-1} is the state input matrix, \mathbf{V}_{k-1} is the process noise, \mathbf{w}_{k-1} is zero-mean white Gaussian noise and its corresponding covariance matrix is \mathbf{Q}_{ca} . Φ_{CA} , \mathbf{G}_{k-1} are described as:

$$\Phi_{CA} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (38)$$

$$\mathbf{G}_{CA} = \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix} \quad (39)$$

where T is the sampling interval.

In radar tracking system, the target motion is usually modelled in Cartesian coordinates, whereas the target's position and azimuth are obtained in polar coordinate. The radar is located at the origin, and provides range and bearing measurements. The measurement model can be established as

$$\mathbf{z}_k = \begin{pmatrix} \sqrt{x_k^2 + y_k^2} \\ \text{atan2}(y_k, x_k) \end{pmatrix} + \mathbf{v}_k \quad (40)$$

where $\text{atan2}(\cdot)$ is the four-quadrant inverse tangent function, \mathbf{v}_k is the white Gaussian measurement noise with zero mean and covariance $\mathbf{R}_k = \text{diag}([\sigma_r^2, \sigma_\theta^2])$. σ_r and σ_θ denote the standard deviation of range measurement noise and bearing angle measurement noise, respectively.

4.2. Simulation of the STSSRCKF

Example 1. In this simulation, the sampling interval is $T = 1$ s and simulation time is 100s. The Monte Carlo simulations are carried out 200 times. The RMSE of the target position at time k and the accumulative RMSE (ARMSE) of estimated position at all times are defined in Equations (41) and (42):

$$\text{RMSE}_{\text{pos}}(k) = \sqrt{\frac{1}{M} \sum_{m=1}^M ((x_k - \hat{x}_{m,k})^2 + (y_k - \hat{y}_{m,k})^2)} \quad (41)$$

$$\text{ARMSE}_{\text{pos}} = \sqrt{\frac{1}{N} \sum_{k=1}^N (\text{RMSE}_{\text{pos}}^2(k))} \quad (42)$$

where M is the number of Monte Carlo runs, (x_k, y_k) is the actual value of the target position at time k and $(\hat{x}_{m,k}, \hat{y}_{m,k})$ is the estimated position at time k in m th Monte-Carlo. The RMSE and the accumulative RMSE in the velocity and acceleration can be defined in the same way.

This example considers a two-dimensional simulation scenario including one motion mode of high maneuver. The initial location of the target is $(x, y) = (100 \text{ m}, 400 \text{ m})$, its initial velocity is $(v_x, v_y) = (15 \text{ m/s}, 20 \text{ m/s})$, and its initial acceleration is $(a_x, a_y) = (0 \text{ m/s}^2, 0 \text{ m/s}^2)$. The target makes a uniform motion during the first 150 s. Then, it takes a high maneuver with the acceleration $(a_x, a_y) = (15 \text{ m/s}^2, 25 \text{ m/s}^2)$ up to the end of this simulation at $t = 200 \text{ s}$. In this simulation, the initial value $\hat{x}_{0|0}$ and the initial covariance matrix $P_{0|0}$ are set to be $[100 \text{ m}, 15 \text{ m/s}, 0 \text{ m/s}^2, 400 \text{ m}, 20 \text{ m/s}, 0 \text{ m/s}^2]^T$ and $\text{diag}[(50 \text{ m})^2, (20 \text{ m/s})^2, (1 \text{ m/s}^2)^2, (50 \text{ m})^2, (10 \text{ m/s})^2, (1 \text{ m/s}^2)^2]^T$, respectively. The standard deviation of range measurement noise σ_r is 30 m and the standard deviation of bearing angle measurement noise σ_θ is 10 mrad.

The example is executed to examine the performance among the SSRCKF, STF, STUKF, STCKF and STSSRCK methods. The RMSEs of the position, velocity and acceleration using the five filters are shown in Figures 1–3. It can be shown that the STF, STUKF, STCKF and STSSRCK methods can converge quickly when the target engages in high maneuvering. The SSRCKF algorithm only has a good performance for uniform motion. However, the performance of SSRCKF decreases seriously when the target engages in high maneuvering. This is because that the prediction covariance cannot be adjusted timely when the target state suddenly changes. The STF algorithm has the fourth speed of convergence, which is due to the fact that the linear approximation in the STF may introduce errors in the state which may lead the state to diverge. As can be seen from Figures 1–3, when the target is making uniform motion within the first 100 s, the five methods have a similar performance. When the maneuver starts at $t = 101 \text{ s}$, it obviously shows that STF, STUKF, STCKF and STSSRCKF have the ability to convergence. The main reason is that the fading factor can adjust the prediction covariance and the corresponding filter gain in real time, which makes these algorithms converge in a short time. We can also see that the RMSE of the proposed algorithm is lower than that of STUKF and STCKF. It means that estimate precision of the proposed algorithm is higher than that of the two algorithms. It is demonstrated that the proposed algorithm can effectively track the abrupt motion state of the target.

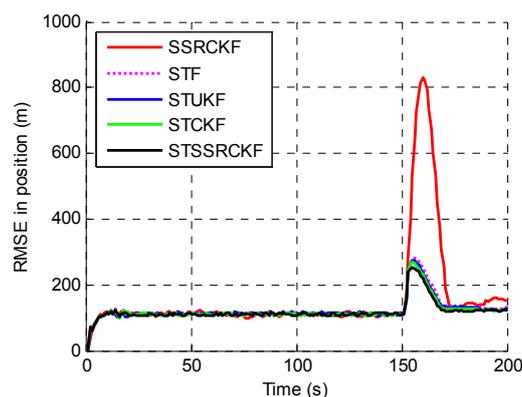


Figure 1. Root mean square error (RMSE) of the estimated position.

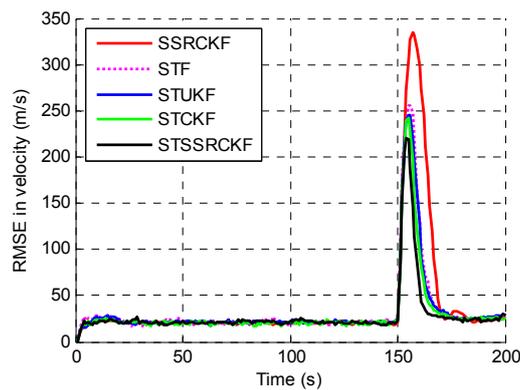


Figure 2. RMSE of the estimated velocity.

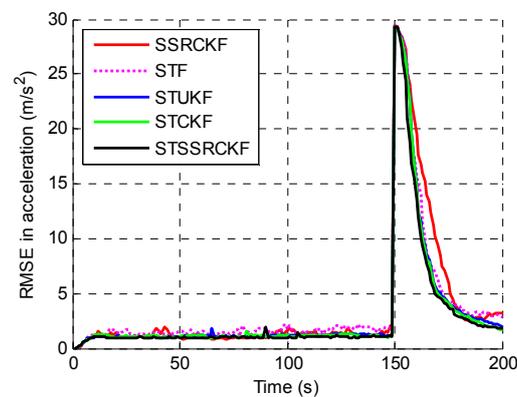


Figure 3. RMSE of the estimated acceleration.

To quantitatively describe the tracking performance, the ARMSEs of the five methods in estimating different target parameters are listed in Table 1. As shown, the STSSRCKF provided the best result in terms of estimation. The STCKF also performed well, followed by the STUKF and STF. The SSRCKF provided the worst estimate. We can also draw the conclusion that the STSSRCKF has the highest tracking accuracy of the position, velocity and acceleration.

Table 1. Tracking performance comparison.

Filters	Position ARMSE/m	Velocity ARMSE/(m/s)	Acceleration ARMSE/(m/s ²)
SSRCKF	152.1	31.2	6.9
STF	129.7	28.4	6.2
STUKF	124.5	27.5	5.9
STCKF	123.1	26.7	5.8
STSSRCKF	119.3	25.1	5.6

The program is made on the Intel Core (TM) i5-4430 3.0GHZ CPU with 4.00G RAM. Table 2 shows the computational complexity and the computational time of SSRCKF, STF, STUKF, STCKF and STSSRCK for each run. Apart from STF, the computational complexity of different filters is mainly determined by the number of points they use. The computational complexity of STCKF as well as STUKF differs only by one points. The computational complexity of SSRCKF and STSSRCKF is $O((2n + 2)^3)$, where n denotes the dimension of state. In addition, we can see that the computational complexity of STF is the lowest. Because there is a clear formula in STF to calculate the Jacobian matrix, the computational complexity of STF is much smaller than other four algorithms. It is also shown that the computational time of the SSRCKF is 0.07 s for each run. However, the computational time

of the STSSRCKF is 0.15 s, which is greater than that of the SSRCKF. This is because the STSSRCKF needs to calculate the suboptimal fading factor at each time step. At present, the time consumption is acceptable. The STSSRCKF needs more computational time than the SSRCKF, but considering the significant performance improvement gained from the STSSRCKF, this increased computational time is not substantial.

From this simulation, we can conclude that the STSSRCKF can perform the best in terms of the balance between computational complexity and estimation accuracy.

Table 2. Computational complexity and computational time of different filters.

Filters	Computational Complexity	Computational Time (s)
SSRCKF	$O\{(2n + 2)^3\}$	0.07
STF	$O\{n^2\}$	0.02
STUKF	$O\{(2n + 1)^3\}$	0.14
STCKF	$O\{(2n)^3\}$	0.14
STSSRCKF	$O\{(2n + 2)^3\}$	0.15

Example 2. This example evaluates the proposed algorithm in tracking a target with weak maneuver and medium maneuver. Therefore, two simulations are simulated as follows. Assume that there is a target making uniform at first. The initial location of the target is $(x, y) = (5000 \text{ m}, 5000 \text{ m})$, its initial velocity is $(v_x, v_y) = (150 \text{ m/s}, 80 \text{ m/s})$, and its initial acceleration is $(a_x, a_y) = (0 \text{ m/s}^2, 0 \text{ m/s}^2)$.

Case 1: Simulation of medium maneuvering target tracking. The target moves with initial acceleration until $t = 150 \text{ s}$. Then, it maneuvers with acceleration of $(a_x(151), a_y(151)) = (5 \text{ m/s}^2, 5 \text{ m/s}^2)$ up to end of this simulation at $t = 200 \text{ s}$.

Case 2: Simulation of weak maneuvering target tracking. The initial position, velocity and acceleration of the target are the same as those in Case 1. The target also moves with initial acceleration until $t = 150 \text{ s}$. Then, it maneuvers with acceleration of $(a_x(151), a_y(151)) = (0.5 \text{ m/s}^2, 0.5 \text{ m/s}^2)$ up to end of this simulation at $t = 200 \text{ s}$.

Table 3 lists the accumulative RMSEs of the five methods in estimation the three target parameters. As can be seen from Table 3, the STSSRCKF algorithm also has a good tracking performance for a weak or medium maneuvering target.

Table 3. ARMSEs in simulation of medium and weak maneuvering target.

Simulation	Filters	Position ARMSE/m	Velocity ARMSE/(m/s)	Acceleration ARMSE/(m/s ²)
Case 1	SSRCKF	101.6	21.2	5
	STF	95.3	20.2	4.5
	STUKF	88.5	16.4	4.1
	STCKF	87.4	16.8	4.1
	STSSRCKF	81.1	15.9	3.7
Case 2	SSRCKF	50.5	8.2	1.8
	STF	65.1	10.8	2.4
	STUKF	57.3	8.8	2.2
	STCKF	56.3	8.3	2.2
	STSSRCKF	53.4	8.4	2.1

5. Conclusions

To implement higher tracking accuracy for a maneuvering target, a new method has been proposed based on the STF and SSRCKF algorithms. Firstly, the time-varying suboptimal fading factor is introduced in order to adjust the prediction covariance and the corresponding filter gain in

real time. Secondly, in the proposed method, the spherical simplex-cubature rule takes the place of calculating nonlinear function Jacobian matrix. In this way, STSSRCKF can converge rapidly in a short time. Thus, the proposed method has a high tracking accuracy for maneuvering target tracking. Simulation results show that the STSSRCKF can achieve higher accuracy and robustness than STF, STUKF, STCKF and SSRCKF, and indicate that it is suitable for maneuvering target tracking.

Author Contributions: All authors were involved in the study presented in this manuscript. Hua Liu provided insights into formulating the ideas and wrote the original manuscript of the paper. Wen Wu reviewed the manuscript and provided scientific advising to this study.

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

References

1. Yepes, J.L.; Hwang, I.; Rotea, M. New algorithms for aircraft intent inference and trajectory prediction. *J. Guid. Control Dyn.* **2007**, *30*, 370–382. [[CrossRef](#)]
2. Liu, Z.X.; Zhang, Q.Q.; Li, L.Q.; Xie, W.X. Tracking multiple maneuvering targets using a sequential multiple target Bayes filter with jump Markov system models. *Neurocomputing* **2016**, *216*, 183–191. [[CrossRef](#)]
3. Li, X.R.; Jilkov, V.P. Survey of maneuvering target tracking. Part V: Multiple-model methods. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1255–1321.
4. Kazimierski, W.; Stateczny, A. Optimization of multiple model neural tracking filter for marine targets. In Proceedings of the 13th International Radar Symposium (IRS), Warsaw, Poland, 23–25 May 2012; pp. 543–548.
5. Zhou, H.R.; Kumar, K.S.P. A “Current” Statistical Model and Adaptive Algorithm for Estimating Maneuvering Targets. *AIAA J. Guid.* **1984**, *7*, 596–602.
6. Bingbing Jiang, W.S. Range tracking method based on adaptive “current” statistical model with velocity prediction. *Signal Process.* **2017**, 261–270. [[CrossRef](#)]
7. Zhou, D.H.; Frank, P.M. Strong tracking filtering of nonlinear time varying stochastic systems with coloured noise application to parameter estimation and empirical robustness analysis. *Int. J. Control* **1996**, *65*, 295–307. [[CrossRef](#)]
8. Li, X.R.; Jilkov, V.P. A Survey of Maneuvering Target Tracking—Part IV: Decision-Based Methods. In Proceedings of the SPIE Conference on Signal and Data Processing of Small Targets 2002, Orlando, FL, USA, 15 April 2002; pp. 4728–4760.
9. Schmidt, S.F. The Kalman filter—Its recognition and development for aerospace applications. *J. Guid. Control Dyn.* **1981**, *4*, 4–7. [[CrossRef](#)]
10. Djuric, P.M.; Kotecha, J.H.; Zhang, J.Q.; Huang, Y.F.; Ghirmai, T.; Bugallo, M.F.; Miguez, J. Particle filtering. *IEEE Signal. Proc. Mag.* **2003**, *20*, 19–38. [[CrossRef](#)]
11. Doucet, A.; Johansen, A.M. A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. Available online: http://automatica.dei.unipd.it/tl_files/utenti/lucaschenato/Classes/PSC10_11/Tutorial_PF_doucet_johansen.pdf (accessed on 31 March 2017).
12. Martino, L.; Read, J.; Elvira, V.; Louzada, F. Cooperative parallel particle filters for online model selection and applications to urban mobility. *Digit. Signal. Process.* **2017**, *60*, 172–185. [[CrossRef](#)]
13. Kotecha, J.H.; Djuric, P.A. Gaussian particle filtering. *IEEE Trans. Signal. Process.* **2003**, *51*, 2592–2601. [[CrossRef](#)]
14. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [[CrossRef](#)]
15. Arasaratnam, I.; Haykin, S.; Elliott, R.J. Discrete-time nonlinear filtering algorithms using Gauss-Hermite quadrature. *Proc. IEEE* **2007**, *95*, 953–977. [[CrossRef](#)]
16. Sadhu, S.; Srinivasan, M.; Bhaumik, S.; Ghoshal, I.K. Central difference formulation of risk-sensitive filter. *IEEE Signal. Process. Lett.* **2007**, *14*, 421–424. [[CrossRef](#)]
17. Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [[CrossRef](#)]
18. Arasaratnam, I.; Haykin, S. Cubature Kalman smoothers. *Automatica* **2011**, *47*, 2245–2250. [[CrossRef](#)]
19. Jia, B.; Xin, M.; Cheng, Y. High-degree cubature Kalman filter. *Automatica* **2013**, *49*, 510–518. [[CrossRef](#)]

20. Wang, S.Y.; Feng, J.C.; Tse, C.K. Spherical Simplex-Radial Cubature Kalman Filter. *IEEE Signal Process. Lett.* **2014**, *21*, 43–46. [[CrossRef](#)]
21. Wang, L.J.; Wu, L.F.; Guan, Y.; Wang, G.H. Online Sensor Fault Detection Based on an Improved Strong Tracking Filter. *Sensors* **2015**, *15*, 4578–4591. [[CrossRef](#)] [[PubMed](#)]
22. Ge, Q.B.; Shao, T.; Wen, C.L.; Sun, R.Y. Analysis on Strong Tracking Filtering for Linear Dynamic Systems. *Math. Probl. Eng.* **2015**, *2015*. [[CrossRef](#)]
23. Li, D.; Ouyang, J.; Li, H.Q.; Wan, J.F. State of charge estimation for LiMn2O4 power battery based on strong tracking sigma point Kalman filter. *J. Power Sources* **2015**, *279*, 439–449. [[CrossRef](#)]
24. Gao, M.; Zhang, H.; Zhou, Y.L.; Zhang, B.Q. Ground maneuvering target tracking based on the strong tracking and the cubature Kalman filter algorithms. *J. Electron. Imaging* **2016**, *25*, 023006. [[CrossRef](#)]
25. Jia, B.; Xin, M. Multiple sensor estimation using a new fifth-degree cubature information filter. *Trans. Inst. Meas. Control.* **2015**, *37*, 15–24. [[CrossRef](#)]
26. Jwo, D.J.; Wang, S.H. Adaptive fuzzy strong tracking extended kalman filtering for GPS navigation. *IEEE Sens. J.* **2007**, *7*, 778–789. [[CrossRef](#)]
27. Jwo, D.J.; Yang, C.F.; Chuang, C.H.; Lee, T.Y. Performance enhancement for ultra-tight GPS/INS integration using a fuzzy adaptive strong tracking unscented Kalman filter. *Nonlinear Dyn.* **2013**, *73*, 377–395. [[CrossRef](#)]
28. Han, P.; Mu, R.; Cui, N. Effective fault diagnosis based on strong tracking UKF. *Aircr. Eng. Aerosp. Technol.* **2011**, *83*, 275–282. [[CrossRef](#)]



© 2017 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 (<http://creativecommons.org/licenses/by/4.0/>).