sensors

MDPI

Article

Centralized Multi-Sensor Square Root Cubature Joint Probabilistic Data Association

Yu Liu [1,2,*], Jun Liu [2,*], Gang Li [3], Lin Qi [2], Yaowen Li [3] and You He [2]

[1] School of Electronic and Information Engineering, Beihang University, Beijing 100191, China

[2] Research Institute of Information Fusion, Naval Aeronautical and Astronautical University, Yantai 264001, China; 3278pirate@163.com (L.Q.); heyou_f@126.com (Y.H.)

[3] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China; gangli@mail.tsinghua.edu.cn (G.L.); lyw17@mails.tsinghua.edu.cn (Y.L.)

* Correspondence: liuyu77360132@126.com (Y.L.); 18615042187@163.com (J.L.); Tel.: +86-535-663-5877 (Y.L.)

Received: 8 September 2017; Accepted: 2 November 2017; Published: 5 November 2017

Abstract: This paper focuses on the tracking problem of multiple targets with multiple sensors in a nonlinear cluttered environment. To avoid Jacobian matrix computation and scaling parameter adjustment, improve numerical stability, and acquire more accurate estimated results for centralized nonlinear tracking, a novel centralized multi-sensor square root cubature joint probabilistic data association algorithm (CMSCJPDA) is proposed. Firstly, the multi-sensor tracking problem is decomposed into several single-sensor multi-target tracking problems, which are sequentially processed during the estimation. Then, in each sensor, the assignment of its measurements to target tracks is accomplished on the basis of joint probabilistic data association (JPDA), and a weighted probability fusion method with square root version of a cubature Kalman filter (SRCKF) is utilized to estimate the targets' state. With the measurements in all sensors processed CMSCJPDA is derived and the global estimated state is achieved. Experimental results show that CMSCJPDA is superior to the state-of-the-art algorithms in the aspects of tracking accuracy, numerical stability, and computational cost, which provides a new idea to solve multi-sensor tracking problems.

Keywords: multi-sensor tracking; data association; cubature Kalman filter; state estimation; centralized filtering

1. Introduction

The centralized state estimation method for multiple-target tracking with multiple sensors can integrate detecting information from different sensors according to certain rules, which will make the tracking results more accurate than that of a single sensor [1–3]. Multi-sensor fusion is an important data processing method in the field of target tracking, especially under nonlinear conditions, and the fused result with multiple sensors is often better for multiple-target tracking, which has been widely concerned by scholars both at home and abroad [3–12].

The state-of-the-art techniques for multiple-target tracking mainly contains two categories. The first category is the tracking method based on a random finite set; the other is the method based on measurement to track association. The former method can directly track multiple targets without associating the detected measurements with the tracks in the surveillance area. However, this kind of method faces complex integral operations, which are often difficult to find an exact solution. Although approximation methods have been a good choice for this problem, the computational cost is too great, and improper approximation will lead to degradation of the tracking accuracy, thus making it a long way from being extensively utilized in practical applications [4]. The latter method has been widely applied in various tracking systems, and is still the most common method for centralized tracking [5–14].

In the second category, the sequential multi-sensor joint probabilistic data association algorithm (MSJPDA) is a classical and effective method for multi-sensor tracking, which achieves better tracking performance compared with its parallel counterparts [10,11]. Unfortunately, current MSJPDA algorithms are often proposed to solve association problems under linear circumstances, and the nonlinear environment is seldom involved in the literature. As is known, the nonlinear equation can be linearized according to Taylor expansion, then a MSJPDA algorithm based on the extended Kalman filter (EKF), the so called MSJPDA-EKF is derived [12–14]. However, it is essential for MSJPDA-EKF to calculate the Jacobian matrix in the linearization step in each iteration, and the ignorance of higher orders may introduce large linearization errors, which may result in degraded performance, and even divergence [1,2,14]. In addition, the estimated state will affect the calculation of association probability in the next iteration, thus causing changes in the computation of weights of joint events, which may eventually makes false correlations, and even lead to divergence [15]. To conquer the shortcomings of MSJPDA-EKF, the MSJPDA algorithm based on the unscented Kalman filter (MSJPDA-UKF) is proposed. MSJPDA-UKF can acquire higher estimation accuracy than MSJPDA-EKF, and is relatively more stable [16]. However, reasonable adjustments are needed in MSJPDA-UKF to achieve the ideal performance, and a numerical instability phenomenon is easy to occur in the situation of high dimensional estimation [16–18].

To address the high dimensional nonlinear estimation problem, the cubature Kalman filter (CKF) based on the third-degree spherical-radial cubature rule has been recently proposed [19–25]. In the procedure of CKF, a certain number of typical cubature points are selected to approximate the posterior probability, then the mean and covariance are captured through a nonlinear transfer function to achieve accurate estimation of the target state [19]. CKF and UKF are both moment-matching filters which deterministically select a set of weighted sample points to approximate the posterior probability density. Compared with EKF, CKF and UKF do not have to calculate the Jacobian matrix in each iteration, thereby reducing the computational cost. At the same time, they both avoid the negative influence of the truncation error in the process of nonlinear state estimation, which is particularly effective in the case of state estimation with strong nonlinearity [19,21–23]. In UKF, to achieve a better estimation performance, the adjustment of the scaling parameter is an essential step, as different scaling parameters may yield completely distinct estimation results. The selection of the scaling parameter is of great significance to the performance of UKF in some sense [16,19,21,25]. However, there is no parameter to be adjusted for CKF, and the selected sampling points and weights are only related to the dimension of the target state, which can be calculated in advance to reduce the computational complexity. Furthermore, CKF is more accurate with respect to high dimensional estimation, and easy for design and implementation [19,21–24]. Especially in the case of high dimension, EKF and UKF suffer from the curse of dimensionality or divergence, or both, but CKF still performs well [19,21,25]. Unfortunately, in practice, arithmetic operations performed on finite word-length digital computers may introduce large errors; then the symmetry and positive definiteness of the covariance are often not satisfied. Moreover, perhaps the loss of positive definiteness is more perilous as it terminates the running of the CKF. Then, the square root version of CKF, the so called SRCKF is proposed to address this problem [19,21,23,24]. In SRCKF, only the square root factors are propagated during the iterations. Therefore, the square rooting operations of the matrix are avoided, thus reducing the computation cost. The symmetry and positive definiteness of the covariance are preserved, and the numerical accuracy is also improved [19,21,25].

In this paper, to deal with the target tracking problem in a nonlinear system under a cluttered environment, a novel centralized multi-sensor square root cubature joint probabilistic data association algorithm (CMSCJPDA) is proposed. In CMSCJPDA, measurements of each single sensor are sequentially processed to compute the association probabilities based on the similar rules in JPDA, then SRCKF and the validated measurements are chosen to estimate the state of targets according to weighed state fusion. Eventually, the accurate estimation of multiple targets' states with multiple sensors is accomplished.

The rest of this paper is organized as follows: Section 2 describes the problem of tracking multiple targets with multiple sensors. Then, in Section 3, a simple overview of the numerical integral approximation method based on spherical-radial principle is introduced. The main idea of CMSCJPDA is detailed in Section 4, and an algorithm flow is given. The numerical experiments are designed in Section 5, and the comparison and analysis of CMSCJPDA against several existing methods is also presented in this section. Concluding remarks of this paper and future work are given in Section 6.

2. Problem Formulation

Consider sensors are applied to track N_t targets in a cluttered environment. For arbitrary target t $(1 \leq t \leq N_t)$, $\mathrm{X}^t(k)$ denotes the state of target t at discrete time instant k. For brevity, without the control input term, the discrete-time state equation of a nonlinear system is:

$$\mathrm{X}^t(k+1) = f^t\left[k, \mathrm{X}^t(k)\right] + \mathrm{V}^t(k) \tag{1}$$

where k is the discrete time instant, $f^t(\cdot)$ is a known nonlinear function, and $\mathrm{V}^t(k)$ is independent zero-mean Gaussian process noise with covariance:

$$\mathrm{E}\left\{\mathrm{V}^t(k)\left[\mathrm{V}^t(l)\right]^{\mathrm{T}}\right\} = \mathrm{Q}^t(k)\delta(k,l) \tag{2}$$

where $\delta(k,j)$ is the Kronecker delta function.

The measurement equation of sensor i for target t is:

$$\mathrm{Z}^{i,t}(k) = h^i\left[k, \mathrm{X}^t(k)\right] + \mathrm{W}^i(k) \tag{3}$$

where $i = 1, 2, \cdots, N_s$ represents the label of sensors. $\mathrm{Z}^{i,t}(k)$ is the measurement vector of sensor i for target t at time instant k. $h^i(\cdot)$ represents a known nonlinear function. $\mathrm{W}^i(k)$ is the zero-mean, independent of process noise in Equation (3) and independent from sensor to sensor, with covariance:

$$\mathrm{E}\left\{\mathrm{W}^i(k)\left[\mathrm{W}^i(l)\right]^{\mathrm{T}}\right\} = \mathrm{R}^i(k)\delta(k,l) \tag{4}$$

3. Numerical Approximation of the Multi-Dimensional Weighted Integral

3.1. Third-Degree Spherical-Radial Rule

Before introducing the CMSCJPDA method, a brief description of the numerical integral approximation based on spherical-radial principle is given. Consider the following multi-dimensional Gaussian weighted integral:

$$\boldsymbol{I}(f) = \int\limits_{\boldsymbol{R}^n} f(\boldsymbol{x}) \exp\left(-\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\right) \mathrm{d}\boldsymbol{x} \tag{5}$$

where $f(\cdot)$ is an arbitrary function, $\boldsymbol{R}^n$ denotes the integral region. As is known, the key to addressing the problem of nonlinear filtering based on Bayesian theory is to compute the first-order and second-order moments. In other words, the core of the Bayesian filter is how to compute Gaussian weighted integral which is of the form *nonlinear function* × *Gaussian density* that is illustrated in Equation (5).

Let $\boldsymbol{x} = r\boldsymbol{y}$, where $\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y} = 1$, So that $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x} = r^2$ for $r \in [0, \infty)$. Then the integral Equation (5) can be rewritten as:

$$\boldsymbol{I}(f) = \int\limits_0^\infty \int\limits_{U_n} f(r\boldsymbol{y}) r^{n-1} \exp\left(-r^2\right) \mathrm{d}\sigma(\boldsymbol{y}) \mathrm{d}r \tag{6}$$

where $U_n = \left\{ \boldsymbol{y} \in \boldsymbol{R}^n \middle| \boldsymbol{y}^{\mathrm{T}} \boldsymbol{y} = 1 \right\}$ is the surface of the sphere and $\sigma(\cdot)$ is the spherical surface measure or the area element on U_n. Therefore, we can split Equation (6) into two integrals:

$$\boldsymbol{I}(f) = \int\limits_0^\infty S(r) r^{n-1} \exp\left(-r^2\right) \mathrm{d}r \tag{7}$$

$$S(r) = \int\limits_{U_n} f(r\boldsymbol{y}) \mathrm{d}\sigma(\boldsymbol{y}) \tag{8}$$

where $\boldsymbol{I}(f)$ is the radial weighted integral that is shown in Equation (7), $S(r)$ is the spherical integral with the unit weighting function $\omega(\boldsymbol{y}) = 1$ that is shown in Equation (8).

Then the above integrals described in Equations (7) and (8) can be solved according to the third-degree spherical cubature rule and radical rule, respectively:

$$\int\limits_{U_n} f(\boldsymbol{y}) \mathrm{d}\sigma(\boldsymbol{y}) \approx \omega \sum_{i=1}^{2n} f[\boldsymbol{u}]_i \tag{9}$$

$$\int\limits_a^b f(\boldsymbol{x}) \omega(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \approx \sum_{i=1}^{m} \omega_i f(\boldsymbol{x}_i) \tag{10}$$

where $[\boldsymbol{u}]_i$ represents the i-th element of generator $\boldsymbol{u}$, and $\boldsymbol{u}$ is a generator if $\boldsymbol{u} = (u_1, u_2, \ldots, u_r, 0, \ldots 0) \in \mathbb{R}^n$, where $u_i > u_{i+1} > 0$, $i = 1, 2, \ldots, (r-1)$. For simplicity, the $(n-r)$ zero coordinates are ignored and notation $[u_1, u_2, \ldots, u_r]$ is used to represent a complete fully-symmetric set of points that can be obtained by permutating and changing the sign of generator $\boldsymbol{u}$ in all possible ways. For example, $[1] \in \mathbb{R}^2$ denotes the following set of points:

$$[1] = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\} \tag{11}$$

where $\omega(x)$ is a known weighting function which is non-negative on the interval $[a,\ b]$.

As is shown in Equation (9), the spherical cubature rule indicates that the spherical integral can be approximated by a weighted sum of function values at the sample points, which are located at the intersection of the unit sphere and its axes. The radical rule described in Equation (10) indicates that an m-point Gaussian quadrature is equal to the sum of $(2m-1)$ polynomials [21].

Especially, a standard Gaussian weighted integral can be computed based on third-degree spherical-radial rule as follows [25]:

$$\boldsymbol{I}_N(f) = \int\limits_{R^n} f(\boldsymbol{x}) N(\boldsymbol{x}\,;\, \boldsymbol{0}, \mathbf{I}) \mathrm{d}\boldsymbol{x} \approx \sum_{i=1}^{2n_x} \omega_i f(\boldsymbol{\xi}_i) \tag{12}$$

where:

$$\boldsymbol{\xi}_i = \sqrt{\frac{2n_x}{2}}[1]_i,\ \omega_i = \frac{1}{2n_x},\ i = 1, 2, \cdots, 2n_x \tag{13}$$

where $\boldsymbol{\xi}_i$ is the cubature point, ω_i is the corresponding weight. $[1]_i$ denotes the i-th row or column of the generator $[1]$.

Actually, the weighting function term $\omega(\cdot)$ of the integrand is often not subject to the standard Gaussian distribution. In other words, the Gaussian weighted integral in nonlinear filtering cannot be solved directly by Equation (12). Fortunately, the following equation makes it possible to work out the nonstandard Gaussian weighted integrals which occur in nonlinear condition:

$$\int_{R^n} f(\boldsymbol{x})N(\boldsymbol{x};\ \mu,\ \Sigma)\mathrm{d}\boldsymbol{x} = \int_{R^n} f\left(\sqrt{\Sigma}\boldsymbol{x}+\mu\right)N(\boldsymbol{x};\ \boldsymbol{0},\ \mathbf{I})\mathrm{d}\boldsymbol{x} \tag{14}$$

Through Equation (14), which is proved in Appendix A, the nonstandard Gaussian weighted integrals can be transformed into the standard ones. Then, the approximation of posterior mean and error covariance can be addressed by Equation (12). Thus, an iteration of the time and the measurement updates in the Bayesian filter is completed.

3.2. Accuracy Analysis

Assume an n-dimensional vector $\boldsymbol{x} \sim N(\overline{\boldsymbol{x}},\ \boldsymbol{P})$, where $\overline{\boldsymbol{x}}$ is the mean of $\boldsymbol{x}$, and $\boldsymbol{P}$ is the corresponding covariance. We extend the nonlinear function $f(\boldsymbol{x})$ at $\overline{\boldsymbol{x}}$ based on Taylor extension:

$$f(\boldsymbol{x}) = f(\overline{\boldsymbol{x}}) + D_{\Delta \boldsymbol{x}}f + \frac{D^2_{\Delta \boldsymbol{x}}f}{2!} + \frac{D^3_{\Delta \boldsymbol{x}}f}{3!} + \frac{D^4_{\Delta \boldsymbol{x}}f}{4!} + \cdots \tag{15}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x} - \overline{\boldsymbol{x}}$, $D_{\Delta\boldsymbol{x}}f = [\Delta\boldsymbol{x}]^{\mathrm{T}}\nabla f(\boldsymbol{x})|_{\boldsymbol{x}=\overline{\boldsymbol{x}}}$, $\nabla f(\boldsymbol{x})$ is the partial derivative of $f(\boldsymbol{x})$.

(1) We use the third-degree spherical-radial rule to approximate the multi-dimensional integral $\boldsymbol{I}(f)$, then:

$$\boldsymbol{I}_{\mathrm{CKF}}(f) = \frac{1}{2n_x}\sum_{i=1}^{2n_x} f(\boldsymbol{x}_i) = \frac{1}{2n_x}\sum_{i=1}^{2n_x} f(\overline{\boldsymbol{x}}) + D_{\Delta \boldsymbol{x}_i}f + \frac{D^2_{\Delta \boldsymbol{x}_i}f}{2!} + \frac{D^3_{\Delta \boldsymbol{x}_i}f}{3!} + \frac{D^4_{\Delta \boldsymbol{x}_i}f}{4!} + \cdots \tag{16}$$

where $\boldsymbol{x}_i$ is the sampled cubature point, $\Delta\boldsymbol{x} = \boldsymbol{x}_i - \overline{\boldsymbol{x}}$. Through proper simplification, the final result can be written as:

$$\boldsymbol{I}_{\mathrm{CKF}}(f) = f(\overline{\boldsymbol{x}}) + \left(\frac{\nabla^{\mathrm{T}}\boldsymbol{P}\nabla}{2!}\right)f + \frac{n^{k-1}}{(2k)!}\sum_{i=1}^{n_x}\left(a_{i1}\frac{\partial}{\partial x_1} + \cdots + a_{in}\frac{\partial}{\partial x_n}\right)^{2k} f(x)\Bigg|_{\boldsymbol{x}=\overline{\boldsymbol{x}}} \tag{17}$$

where $k = 1,2,3,\cdots$, $a_{ij} = \left[\sqrt{\boldsymbol{P}}\right]_{ij}$, $i,j = 1,2,\cdots,n$, $\frac{\partial}{\partial x_i}f$ is the partial derivative of $f(\boldsymbol{x})$ at the ith component of $\boldsymbol{x}$. The interested reader can be referred to paper [26] for details in the extension.

(2) Like the derivation process of $\boldsymbol{I}_{\mathrm{CKF}}(f)$, we use UKF to operate the approximation, and the final result is:

$$\boldsymbol{I}_{\mathrm{UKF}}(f) = f(\overline{\boldsymbol{x}}) + \left(\frac{\nabla^{\mathrm{T}}\boldsymbol{P}\nabla}{2!}\right)f + \frac{(n+\kappa)^{k-1}}{(2k)!}\sum_{i=1}^{n_x}\left(a_{i1}\frac{\partial}{\partial x_1} + \cdots + a_{in}\frac{\partial}{\partial x_n}\right)^{2k} f(x)\Bigg|_{\boldsymbol{x}=\overline{\boldsymbol{x}}} \tag{18}$$

where κ is the scaling parameter, and the remaining notations are the same as Equation (17). To capture the kurtosis of the prior density as correctly as possible, κ is suggested to be $\kappa = 3 - n_x$ [16].

This paper is to tackle the high-dimensional estimation problem, so here we just discuss the case in which the state dimension $n_x > 3$. As is shown in Equations (17) and (18), the first two terms in both equations are the same, the only difference is in the last term. If $n_x > 3$, $n^{k-1} > (n+\kappa)^{k-1}$, and $2k$ is an even number, then $\boldsymbol{I}_{\mathrm{CKF}}(f) > \boldsymbol{I}_{\mathrm{UKF}}(f)$, which indicates that CKF is more accurate than UKF in high-dimensional estimation.

We choose the stability factor $I = \sum_i|\omega_i|/\sum_i \omega_i$ defined as the measure of the numerical stability of the multi-dimensional integral, where ω_i is the sampling weight. It is proven that the sampling rule implemented in a finite-precision arithmetic machine introduces a large amount of round-off errors when the stability factor I is larger than unity [19].

In UKF, if $n_x > 3$, and $\kappa + n_x = 3$, so $\kappa = 3 - n_x < 0$, then $\omega_0 = 1 - n_x/3 < 0$. The stability factor is $I_{\text{UKF}} = \frac{2n_x}{3} - 1 > 1$, and it scales linearly with state dimension n_x, which indicates that UKF introduces significant perturbations in numerical estimates for the moment integral. However, in CKF, the stability factor I_{CKF} always meets $I_{\text{CKF}} = 1$, which shows that CKF is more accurate than UKF for high-dimensional state estimation.

4. Centralized Multi-Sensor Square Root Cubature Joint Probabilistic Data Association

In CMSCJPDA, measurements of each sensor are processed in sequence, then the multi-sensor multi-target tracking problem can be reduced to several single-sensor multi-target tracking problems, which are easier to solve.

Assume there are $m^t_{i,\,k}$ validated measurements for target t obtained by sensor i at time instant k. $l_i \left(0 \le l_i \le m^t_{i,\,k}\right)$ is the label of validated measurements from sensor i, $l_i = j\ (j \neq 0)$ represents the jth measurements in the validated region. Especially, $l_i = 0$ indicates that there is no measurement in the validated region. $\mathrm{Z}^{i,t}_{l_i}(k)$ represents the l_ith validated measurement, $\hat{\mathrm{Z}}^t_{i|i-1}(k|k)$ represents the predicted measurement, which will be discussed in detail later. $\beta^t_{l_i,i}(k)$ represents the association probability that the measurement l_i originated from target t, $\mathrm{K}^t_i(k)$ is the filtering gain. The state estimate and the corresponding error covariance after processing the measurements of the ith sensor are denoted by $\hat{\mathrm{X}}^t_i(k|k)$ and $\mathrm{P}^t_i(k|k)$, respectively. $\hat{\mathrm{X}}^t_0(k|k)$ and $\mathrm{P}^t_0(k|k)$ represent the initial estimation, and $\hat{\mathrm{X}}^t(k|k)$ and $\mathrm{P}^t(k|k)$ represent the final estimation at time instant k, respectively. Then, the state update is as follows:

$$\hat{\mathrm{X}}^t_i(k|k) = \hat{\mathrm{X}}^t_{i-1}(k|k) + \mathrm{K}^t_i(k) \sum_{l_i=0}^{m^t_{i,\,k}} \beta^t_{i,\,l_i}(k) \left[\mathrm{Z}^{i,t}_{l_i}(k) - \hat{\mathrm{Z}}^t_{i|i-1}(k|k)\right] \tag{19}$$

where:

$$\begin{cases} \hat{\mathrm{X}}^t_0(k|k) = \hat{\mathrm{X}}^t(k|k-1),\ \hat{\mathrm{X}}^t(k|k) = \hat{\mathrm{X}}^t_{N_s}(k|k) \\ \mathrm{P}^t_0(k|k) = \mathrm{P}^t(k|k-1),\ \mathrm{P}^t(k|k) = \mathrm{P}^t_{N_s}(k|k) \end{cases} \tag{20}$$

Then, the update of the error covariance is:

$$\begin{aligned} \mathrm{P}^t_i(k|k) = \mathrm{P}^t_{i-1}(k|k) - \left[1 - \beta^t_{0,i}(k)\right] \mathrm{K}^t_i(k) \mathrm{S}^t_i(k) \left[\mathrm{K}^t_i(k)\right]^{\mathrm{T}} \\ + \sum_{l_i=0}^{m^t_{i,\,k}} \beta^t_{i,\,l_i}(k) \hat{\mathrm{X}}^t_{i,\,l_i}(k|k) \left[\hat{\mathrm{X}}^t_{i,\,l_i}(k|k)\right]^{\mathrm{T}} - \hat{\mathrm{X}}^t_i(k|k) \left[\hat{\mathrm{X}}^t_i(k|k)\right]^{\mathrm{T}} \end{aligned} \tag{21}$$

Note that the key point for nonlinear single sensor multi-target tracking problem is to compute the association probability $\beta^t_{i,\,l_i}(k)$ and nonlinear state estimation $\hat{\mathrm{X}}^t_i(k|k)$. JPDA is thought to be an effective method for multiple targets tracking with single sensor, so we choose JPDA to compute $\beta^t_{i,\,l_i}(k)$ here. The remaining problem is to obtain $\hat{\mathrm{X}}^t_i(k|k)$. Although CKF is a good method to deal with the nonlinear state estimation problem, it is easily influenced by limited computer word-length and numerical errors, which may lead to loss of positive definiteness and symmetry of the error covariance matrix [19,21–24]. The effective way to preserve both properties and improve the numerical stability is to design a square root version of the CKF. Despite the fact that the square root cubature Kalman filter (SRCKF) is reformulated to propagate the square roots of the covariance matrices, both CKF and SRCKF are algebraically equivalent to each other, the interested reader is referred to [19]. In the proposed method, we choose SRCKF to update the state of the targets after the correlation step.

4.1. Computation of Association Probabilities

The probability $\beta^{t}_{i,\ l_i}(k)$ that measurement l_i originated from target t can be calculated by summing over all feasible events for which it is true [1]:

$$\beta^{t}_{l_i,i}(k) = \sum_{m=1}^{n^t_i(k)} \hat{\omega}^{m}_{l_i,\ t}(\theta_m(k))\Pr\{\theta_m(k)\big|Z^t_i(k)\} \tag{22}$$

where $n^t_i(k)$ is the number of joint events, $\theta_m(k)$ denotes the event m, and $\hat{\omega}^{m}_{l_i,\ t}(\theta_m(k))$ is the following binary element.

$$\hat{\omega}^{m}_{l_i,\ t}(\theta_m(k)) = \begin{cases} 1, & \text{if } \theta_m(k) \text{ occurs} \\ 0, & \text{else} \end{cases} \tag{23}$$

$\Pr\{\theta_m(k)\big|Z^t_i(k)\}$ represents the posterior probability of event m, which can be computed by Equation (24):

$$\begin{aligned}\Pr\{\theta_m(k)\big|Z^t_i(k)\} &= \tfrac{1}{C}\tfrac{\phi[\theta_m(k)]!}{V^{\phi[\theta_m(k)]}} \\ &\cdot \prod_{l_i=1}^{m^t_{i,\ k}} N^{i,\ t}_{l_i}\left[Z^{i,t}_{l_i}(k)\right]^{\tau_{l_i}[\theta_m(k)]} \prod_{t=1}^{N_t} \left(P^t_{\mathrm{D}}\right)^{\delta_t[\theta_m(k)]}\left(1-P^t_{\mathrm{D}}\right)^{1-\delta_t[\theta_m(k)]}\end{aligned} \tag{24}$$

where C is a constant, $\phi[\theta_m(k)]$ is the total number of false measurements in event m. V is the volume of the entire surveillance region, and P^t_{D} is the detection probability of target t. The target detection indicator $\delta_t[\theta_m(k)]$ is defined as:

$$\delta_t[\theta_m(k)] = \sum_{l_i=1}^{m_{k_i}} \hat{\omega}^{m}_{l_i,\ t}(\theta_m(k)) = \begin{cases} 1, \text{ if target } t \text{ is detected} \\ 0, \text{ otherwise} \end{cases} \tag{25}$$

which indicates whether any measurement is correlated with target t in event m. In the same way, it is also easy to define measurement association indicator:

$$\tau_{l_i}[\theta_m(k)] = \sum_{t=1}^{N_t} \hat{\omega}^{m}_{l_i,\ t}(\theta_m(k)) = \begin{cases} 1, \text{ if measurement } l_i \text{ is associated with a target} \\ 0, \text{ otherwise} \end{cases} \tag{26}$$

which indicates whether measurement l_i is correlated with a target in event m. $N^{i,\ t}_{l_i}\left[Z^{i,t}_{l_i}(k)\right]$ is a conditional probability density function under Gaussian assumption, i.e.:

$$\begin{aligned}N^{i,\ t}_{l_i}\left[Z^{i,t}_{l_i}(k)\right] &= \frac{1}{\sqrt{\left|2\pi S^t_i(k)\right|}}\exp\left\{-\tfrac{1}{2}\left[Z^{i,t}_{l_i}(k) - \hat{Z}^t_{i|i-1}(k|k)\right]^{\mathrm{T}}\right. \\ &\left.\cdot\left[S^t_i(k)\right]^{-1}\left[Z^{i,t}_{l_i}(k) - \hat{Z}^t_{i|i-1}(k|k)\right]\right\}\end{aligned} \tag{27}$$

4.2. State Update

During the update of state, the data processing in first sensor is a little different from that in other sensors. Firstly, the specific procedures for the state update in sensors with label $i > 1$ are as follows:

Step 1: Calculate the cubature points (j = 1, 2, ... , $2n_x$):

$$\hat{X}^{t}_{i|i-1}(k|k) = \hat{X}^{t}_{i-1|i-1}(k|k) \tag{28}$$

$$X^t_{j,\ i|i-1}(k|k) = S^t_{i|i-1}(k|k)\xi_j + \hat{X}^t_{i|i-1}(k|k) \tag{29}$$

where n_x is the dimension of the state. $\mathrm{S}^t_{i|i-1}(k|k)$ is the square root factor of $\mathrm{P}^t_{i-1}(k|k)$, i.e.:

$$\mathrm{P}^t_{i-1}(k|k) = \mathrm{S}^t_{i|i-1}(k|k)\left[\mathrm{S}^t_{i|i-1}(k|k)\right]^{\mathrm{T}} \tag{30}$$

Equations (28) and (29) indicate that the estimated state $\hat{\mathrm{X}}^t_{i-1|i-1}(k|k)$ and error covariance $\mathrm{P}^t_{i-1}(k|k)$ of the previous sensor are regarded as the predicted state $\hat{\mathrm{X}}^t_{i|i-1}(k|k)$ and error covariance $\mathrm{P}^t_{i|i-1}(k|k)$ of the next sensor, which is the greatest difference of the sensors with label $i > 1$ compared with the first sensor with label $i = 1$ in the process of state estimation. That is to say, the time update is not needed in sensors with label $i > 1$, and the estimation of the previous sensor is used as the prediction of the next sensor.

Step 2: Evaluate the prediction of cubature points:

$$\mathrm{Z}^t_{j,i|i-1}(k|k) = h^i\left[k, \mathrm{X}^t_{j,\ i|i-1}(k|k)\right] \tag{31}$$

Step 3: Estimate the prediction of the corresponding measurement:

$$\hat{\mathrm{Z}}^t_{i|i-1}(k|k) = \frac{1}{2n_x}\sum_{j=1}^{2n_x}\mathrm{Z}^t_{j,\ i|i-1}(k|k) \tag{32}$$

Step 4: Estimate the square-root of the innovation covariance:

$$\mathrm{S}^t_{zz,\ i|i-1}(k|k) = \mathrm{Tria}\left(\left[\mathcal{Z}^t_{i|i-1}(k|k) \quad \mathrm{S}_{R^i(k)}\right]\right) \tag{33}$$

where $\mathrm{S}^t_{zz,\ i|i-1}(k|k)$ is a lower triangular matrix. $\mathrm{S}_{R^i(k)}$ denotes a square root factor of $\mathrm{R}^i(k)$ that $\mathrm{R}^i(k) = \mathrm{S}_{R^i(k)}\left[\mathrm{S}_{R^i(k)}\right]^{\mathrm{T}}$, and the weighed, centred matrix:

$$\mathcal{Z}^t_{i|i-1}(k|k) = \tfrac{1}{\sqrt{2n_x}}\left[\mathrm{Z}^t_{1,\ i|i-1}(k|k) - \hat{\mathrm{Z}}^t_{i|i-1}(k|k),\ \mathrm{Z}^t_{2,\ i|i-1}(k|k) - \hat{\mathrm{Z}}^t_{i|i-1}(k|k),\ \cdots,\ \mathrm{Z}^t_{2n_x,\ i|i-1}(k|k) - \hat{\mathrm{Z}}^t_{i|i-1}(k|k)\right] \tag{34}$$

Note that $\mathrm{S} = \mathrm{Tria}(\mathrm{A})$ denotes the QR decomposition, where S is a lower triangular matrix. The relationship of matrices S and A is as follows: Let B be an upper triangular matrix obtained from the QR decomposition on A^{T}. Then, the lower triangular matrix S is obtained as $\mathrm{S} = \mathrm{R}^{\mathrm{T}}$.

Step 5: Estimate the cross-covariance matrix:

$$\mathrm{P}^t_{xz,i|i-1}(k|k) = \mathcal{X}^t_{i|i-1}(k|k)\left[\mathcal{Z}^t_{i|i-1}(k|k)\right]^{\mathrm{T}} \tag{35}$$

where:

$$\mathcal{X}^t_{i|i-1}(k|k) = \tfrac{1}{\sqrt{2n_x}}\left[\mathrm{X}^t_{1,\ i|i-1}(k|k) - \hat{\mathrm{X}}^t_{i|i-1}(k|k), \mathrm{X}^t_{2,\ i|i-1}(k|k) - \hat{\mathrm{X}}^t_{i|i-1}(k|k),\ \cdots,\ \mathrm{X}^t_{2n_x,\ i|i-1}(k|k) - \hat{\mathrm{X}}^t_{i|i-1}(k|k)\right] \tag{36}$$

Step 6: Estimate the filter gain:

$$\mathrm{K}^t_i(k) = \mathrm{P}^t_{xz,i|i-1}(k|k)/\mathrm{S}^t_{zz,\ i|i-1}(k|k)/\left[\mathrm{S}^t_{zz,\ i|i-1}(k|k)\right]^{\mathrm{T}} \tag{37}$$

Step 7: Update the nonlinear state:

$$\hat{\mathrm{X}}^t_{l_i,\ i}(k|k) = \hat{\mathrm{X}}^t_{i-1}(k|k) + \mathrm{K}^t_i(k)\left[\mathrm{Z}^{i,t}_{l_i}(k) - \hat{\mathrm{Z}}^t_{i|i-1}(k|k)\right] \tag{38}$$

Equations (28)–(38) give a solution to the estimation of the target state and error covariance at time instant k with sensor label $i > 1$.

Now the estimation problem in the first sensor with label $i = 1$ is considered, the process of the measurement update is the same as that in sensors with label $i > 1$, but the time update is quite different. In the sensor with label $i = 1$, the estimation of prediction of the state and error covariance is essential. The current cubature points are:

$$\mathrm{X}_j^t(k-1|k-1) = \mathrm{S}^t(k-1|k-1)\xi_j + \hat{\mathrm{X}}^t(k-1|k-1) \tag{39}$$

The propagated cubature points are:

$$\mathrm{X}_j^t(k|k-1) = f^t\left[k-1, \mathrm{X}_j^t(k-1|k-1)\right] \tag{40}$$

Then the predicted state is:

$$\hat{\mathrm{X}}^t(k|k-1) = \frac{1}{2n_x}\sum_{j=1}^{2n_x}\mathrm{X}_j^t(k|k-1) \tag{41}$$

and the predicted error covariance is:

$$\begin{aligned}\mathrm{P}^t(k|k-1) = {} & \tfrac{1}{2n_x}\sum_{j=1}^{2n_x}\left[\mathrm{X}_j^t(k|k-1) - \hat{\mathrm{X}}^t(k|k-1)\right] \\ & \cdot\left[\mathrm{X}_j^t(k|k-1) - \hat{\mathrm{X}}^t(k|k-1)\right]^{\mathrm{T}} + \mathrm{Q}^t(k)\end{aligned} \tag{42}$$

Equations (39)–(42) describe the time update in the first sensor with label $i = 1$. With the measurement update process composed of Equations (28)–(38) applied, the estimated state $\hat{\mathrm{X}}_1^t(k|k)$ and error covariance $\mathrm{P}_1^t(k|k)$ can be achieved respectively after the data of the first sensor is processed.

To state the proposed method more clearly, the algorithm flow of CMSCJPDA is illustrated in Table 1.

Table 1. Algorithm flow of CMSCJPDA.

```
Input: The estimated state $\hat{X}^t(k-1|k-1)$ and error covariance $P^t(k-1|k-1)$ at time instant $k-1$;
Output: The estimated state $\hat{X}^t(k|k)$ and error covariance $P^t(k|k)$ at time instant $k$;
Step 1. Time update ($i = 1$)
  Compute the predicted state $\hat{X}^t(k|k-1)$ and error covariance $P^t(k|k-1)$ according to Equations (39)–(42);
Step 2. Initialization
  $\hat{X}_0^t(k|k) = \hat{X}^t(k|k-1)$, $P_0^t(k|k) = P^t(k|k-1)$;
Step 3. Measurement update
  For $t = 1 : N_t$
    For $i = 1 : N_s$
      Compute the estimation of state $\hat{X}_i^t(k|k)$ and error covariance $P_i^t(k|k)$ with the validated
      measurements of sensor $i$ according to Equations (19)–(38);
    End
  End
Step 4. Establish the final estimation
  $\hat{X}^t(k|k) = \hat{X}_{N_s}^t(k|k)$, $P^t(k|k) = P_{N_s}^t(k|k-1)$
```

5. Numerical Experiments

In this section, the numerical results and the effectiveness of the proposed CMSCJPDA algorithm are discussed in two simulation scenarios: crossing target tracking and maneuvering target tracking. The performance of CMSCJPDA is compared with MSJPDA-EKF and MSJPDA-UKF in two scenarios.

For a fair comparison, we make 50 independent Monte Carlo runs. In each run, the measurement noises are generated independently, and the corresponding noisy position measurements are used to initialize the estimate, which guarantees consistency of the initialization of the filter and makes the final estimate unbiased. Except for the initialization, the process of each Monte Carlo run is the same, and the mean value of 50 runs is considered as the final result. The total number of steps per run is 100. $T = 1$ s is the sampling interval. The nonparametric model is used for the probability mass function (PMF) of the number of false measurements. The expected number of false measurements in the validation gate is $m = 2$. The detection probabilities of all targets are assumed to be $P_D = 0.9$. The probability mass is assumed to be $P_G = 0.9997$. Due to the large initial error, we display the numerical results after the tenth step for clarity.

Targets in the surveillance area are observed by three two-dimensional sensors. The measurement equation of sensor i is:

$$\mathrm{Z}^i(k) = \begin{bmatrix} \sqrt{\left(x(k) - x_{pi}\right)^2 + \left(y(k) - y_{pi}\right)^2} \\ \arctan\left(\frac{y(k)-y_{pi}}{x(k)-x_{pi}}\right) \end{bmatrix} + \mathrm{W}^i(k) \tag{43}$$

where $i = 1,2,3$ is the sensor label. $(x(k),\ y(k))$ is the true state of the targets, $\left(x_{pi},\ y_{pi}\right)$ is the position of sensor i. Other parameters of the sensors are set as shown in Table 2.

Table 2. Sensor position and parameter setting.

Sensor Labe	Sensor Position (m)	Ranging Error (m)	Angle Error (rad)
1	(0, 0)	100	0.01
2	(−500, −500)	200	0.02
3	(−500, 500)	300	0.03

Performance metrics: To compare the tracking performance of different algorithms, the root mean square error (RMSE) is chosen as the metric. The root mean square position error of target t at time instant k is defined as:

$$\mathrm{RMSE}_{\mathrm{pos}}(k,\ t) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(x_i^t(k) - \hat{x}_i^t(k|k)\right)^2} \tag{44}$$

where M is the total number of Monte Carlo simulations. $x_i^t(k)$ and $\hat{x}_i^t(k|k)$ are true and estimated positions of target t in the x direction in the n-th Monte Carlo run. Similarly, we may also define formulas of the root mean square velocity error.

5.1. Crossing Targets Tracking

5.1.1. Simulation Scenario

Consider a two-crossing target tracking case. The state model is:

$$\mathrm{X}(k+1) = \mathrm{F}(k)\mathrm{X}(k) + \Gamma(k)\mathrm{V}(k) \tag{45}$$

where the component of process noise $q_1 = q_2 = 0.01$. The state transition matrix $\mathrm{F}(k)$ is defined as:

$$\mathrm{F}(k) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{46}$$

and the process noise distribution matrix is:

$$\Gamma(k) = \begin{bmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{bmatrix} \tag{47}$$

The initial states of two targets are $\boldsymbol{X}_1(0)$ = [−29,500 m, 400 m/s, 34,500 m, 2,212,400 m/s]$^{\mathrm{T}}$, $\boldsymbol{X}_2(0)$ = [−26,500 m, 296 m/s, 34,500 m, −400 m/s]$^{\mathrm{T}}$. After about 31 s, the two targets cross at the point (−17,000 m, 22,000 m).

5.1.2. Results and Analysis

The true and filtered tracks of two crossing targets are illustrated in Figure 1, which shows the tracking performance of three algorithms in a single run.

To give a more explicit picture of the estimated results, the root mean square position error of three data association algorithms in the x and y directions is shown in Figures 2 and 3, while the root mean square velocity error in the x and y directions is shown in Figures 4 and 5. As is shown, the three algorithms can effectively estimate the state of both targets. However, CMSCJPDA and MSJPDA-UKF, which are based on deterministic sampling, are of higher tracking accuracy than MSJPDA-EKF, and CMSCJPDA is slightly more accurate than MSJPDA-UKF, which just validates the argument in [19]. With the square root version of CKF applied in the proposed method, the square rooting operations of covariance matrix is avoided and the tracking performance is effectively enhanced.

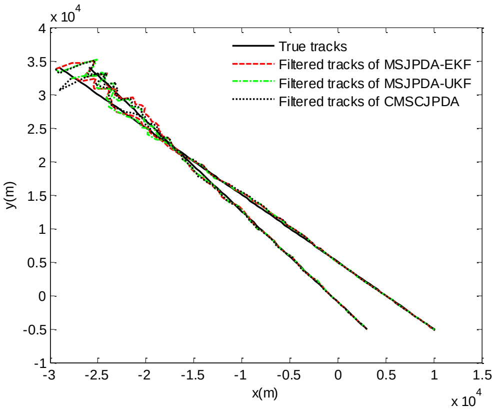

Figure 1. True tracks and filtered tracks of two crossing targets.

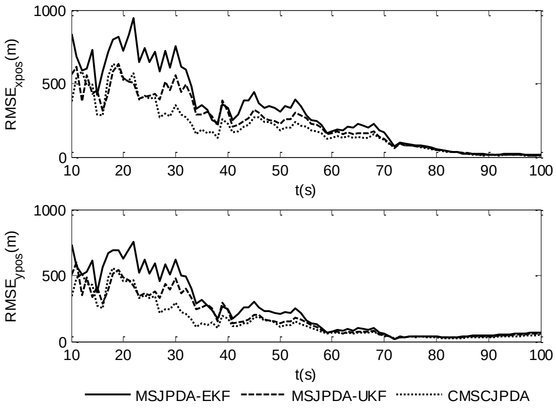

Figure 2. Root mean square position error of target 1.

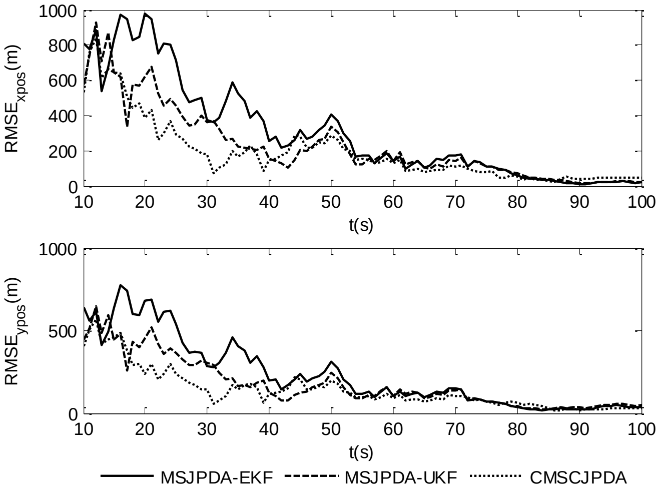

Figure 3. Root mean square position error of target 2.

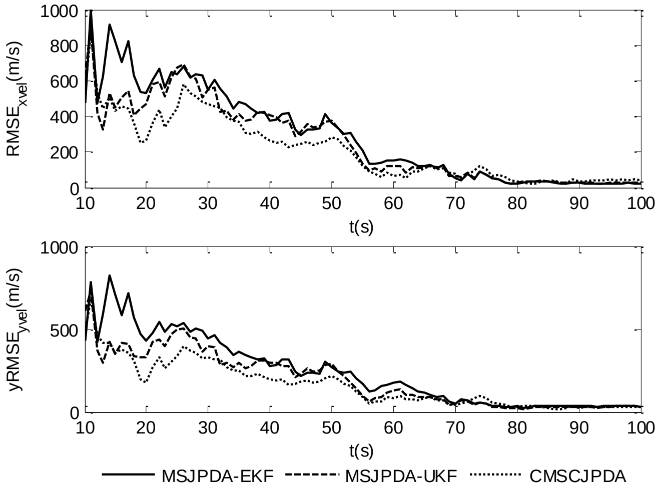

Figure 4. Root mean square velocity error of target 1.

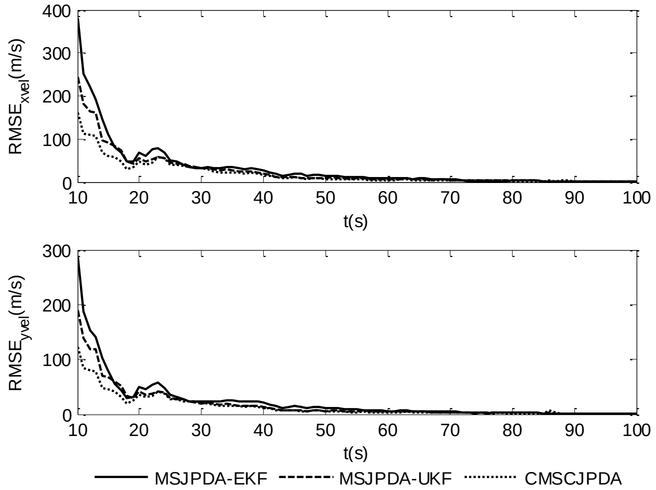

Figure 5. Root mean square velocity error of target 2.

To further validate the effectiveness of the proposed algorithm, numerical stability and computational complexity of the three algorithms are compared in Table 3, which gives the average divergence times (ADT) and average time consumption (ATC) in 50 Monte Carlo runs. As is shown, the ADT of MSJPDA-EKF is twice more than that of MSJPDA-UKF, and nearly four times more than that of CMSCJPDA, to some extent CMSCJPDA is relatively more stable in terms of convergence. As for the computational cost, the ATC of CMSCJPDA is almost half that of MSJPDA-EKF, and less than that of MSJPDA-UKF. In a word, compared with MSJPDA-EKF and MSJPDA-UKF, CMSCJPDA can achieve relatively higher tracking accuracy, and is more stable and less computationally complex under a nonlinear cluttered environment.

Table 3. Performance comparison of the three algorithms.

Algorithms	Average Divergence Times	Average Time Consumption (s)
MSJPDA-EKF	0.87	0.423
MSJPDA-UKF	0.42	0.346
CMSCJPDA	0.27	0.257

5.2. Maneuvering Targets Tracking

5.2.1. Simulation Scenario

Assume there are two maneuvering targets in the surveillance area. The initial states of two targets are $\boldsymbol{X}_1(0)$ = [20,000 m, −600 m/s, 1800 m, 500 m/s]$^{\mathrm{T}}$, $\boldsymbol{X}_2(0)$ = [4000 m, 600 m/s,1800 m, 200 m/s]$^{\mathrm{T}}$. The two targets move at a constant speed in a straight line for 30 s. Then, from 30 s to 50 s, they both move with constant accelerations of $a_{1x} = -30$ m/s^2, $a_{1y} = 20$ m/s^2, $a_{2x} = 30$ m/s^2, $a_{2y} = 20$ m/s^2 in the x and y directions, respectively. From 50 s to 70 s, they still move with constant accelerations, and the accelerations for both targets in the x and y directions are $a_{x1} = -30$ m/s^2, $a_{y1} = -20$ m/s^2, $a_{x2} = 30$ m/s^2, $a_{y2} = -20$ m/s^2, respectively. After 70 s, they both move at a constant speed again until 100 s. The true trajectories of both targets are illustrated in Figure 6.

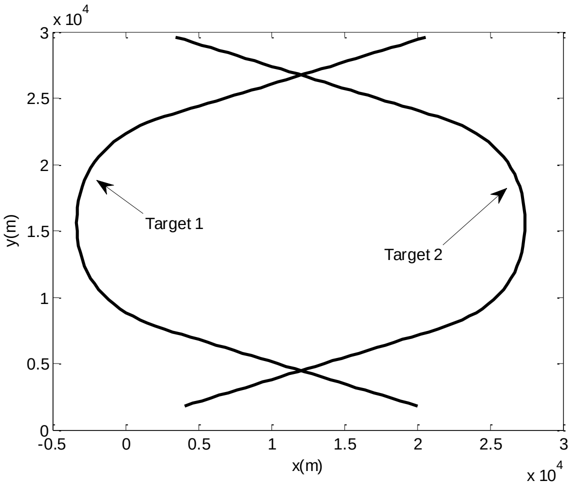

Figure 6. The trajectories of two cross-maneuvering targets.

5.2.2. Results and Analysis

The root mean square (RMS) position error of two targets is compared in Figures 7 and 8. The results show that the estimation error of CMSCJPDA and MSJPDA-UKF is much smaller than that of MSJPDA-EKF in the whole simulation. When t is about 40 s, the estimation error of MSJPDA-EKF is about 200 m larger compared with CMSCJPDA and MSJPDA-UKF. Moreover, with the new association strategy for maneuvering target tracking, CMSCJPDA has relatively higher accuracy than MSJPDA-UKF. At some time instants, the RMS position error of CMSCJPDA is about 50 m smaller than that of MSJPDA-UKF, and it converges relatively faster.

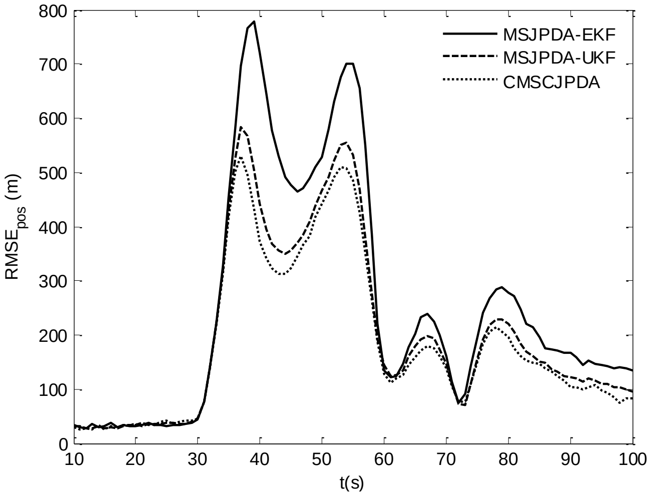

Figure 7. Root mean square position error of target 1.

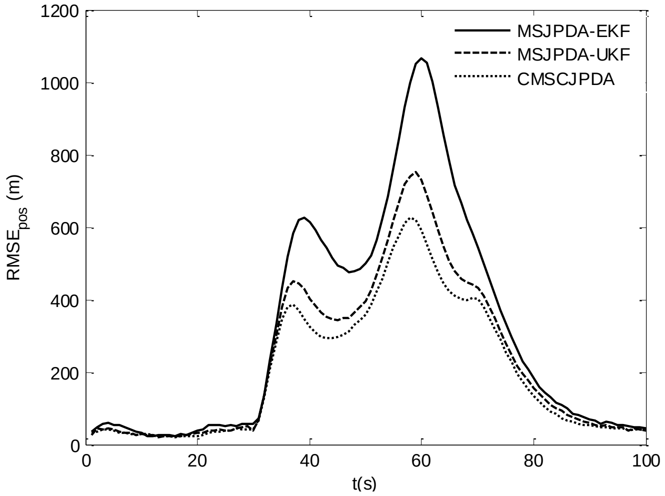

Figure 8. Root mean square position error of target 2.

Table 4 evaluates the tracking performance of the three algorithms in the aspects of mean RMS position error (MRMSE) and correct association rate (CAR). The results indicate that both CMSCJPDA and MSJPDA-UKF are more accurate than MSJPDA-UKF, and CMSCJPDA is slightly more accurate than MSJPDA-UKF. Meanwhile, the CAR of CMSCJPDA is the highest among the three algorithms, which is 11.1% higher than MSJPDA-UKF and 28.7% higher than MSJPDA-EKF, and CMSCJPDA is more time-saving.

Table 4. Performance comparison of three algorithms.

Algorithms	MRMSE (m)	CAR (%)	TC (s)
MSJPDA-EKF	206.2	47.6	0.735
MSJPDA-UKF	132.6	65.2	0.563
CMSCJPDA	104.3	76.3	0.432

6. Conclusions

In this paper, we propose a centralized multi-sensor square root cubature joint probabilistic data association algorithm, which is based on the spherical-radial principle and the sequential update scheme. Compared with the state-of-the-art method in two tracking scenarios, the proposed method is much more accurate than MSJPDA-EKF, and a slightly better than MSJPDA-UKF by using the square root version of cubature Kalman filter to estimate the target state. Furthermore, the experimental results also show that the proposed method is less prone to divergence than MSJPDA-EKF and MSJPDA-UKF, and is more stable with respect to its numerical characteristics. Although the proposed method is less time consuming among the compared methods, due to complicated calculation of the association probability, the joint probabilistic data association-based methods are of significant computational cost, which is still less practical in real applications. Currently, the sensors in our experiment are assumed to be synchronized sampling, and there is no time difference in measurements from different sensors. However, the actual applications can rarely meet this condition, and in the future we will pay more attention to reducing the computational cost and improving the real-time performance. At the same time, the tracking problem with asynchronous sampling and changing the number of targets also deserves consideration.

Acknowledgments: The authors give their sincere thanks to the editors and the anonymous reviewers for their constructive comments of the manuscript. This study was financially co-supported by the State Key Program for Basic Research of China (no. 613XXXXX), the National Natural Science Foundation of China (nos. 61471383, 91538201, 61531020, and 61671463), the Young Elite Scientist Sponsorship Program by CAST (YESS) (no. 2015QNRC001), and China Postdoctoral Science Foundation (2017M610032).

Author Contributions: All the listed authors contributed to the research design; Y.L., J.L., G.L. and L.Q. collected the data; Y.L., J.L. and G.L. analyzed the data; Y.L. and J.L. contributed to the formula derivation; Y.L. and G.L. designed the simulation experiments; J.L. and L.Q. designed the figures and tables; Y.L. searched the literature; Y.L., J.L. and G.L. wrote the paper.

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

Appendix A. Proof of Equation (14)

For simplicity, define the weighting function $\omega_1(\boldsymbol{x}) = N(\boldsymbol{x};\ \mathbf{0},\ \mathbf{I})$ and $\omega_2(\boldsymbol{x}) = N(\boldsymbol{x};\ \upmu,\ \boldsymbol{\Sigma})$. Consider the left side of Equation (14). Since $\boldsymbol{\Sigma}$ is a positive definite matrix, then $\boldsymbol{\Sigma}$ can be factorized to be $\boldsymbol{\Sigma} = \sqrt{\Sigma}\left(\sqrt{\Sigma}\right)^{\mathrm{T}}$. Making a variable substitution by $\boldsymbol{x} = \sqrt{\boldsymbol{\Sigma}}\boldsymbol{y} + \upmu$, we have:

$$\begin{aligned}\int\limits_{R^n} f(\boldsymbol{x})\omega_2(\boldsymbol{x})\mathrm{d}\boldsymbol{x} &= \int\limits_{R^n} f\left(\sqrt{\Sigma}\boldsymbol{y}+\upmu\right)\frac{1}{\sqrt{|2\pi\Sigma|}}\exp\left\{-\left(\sqrt{\Sigma}\boldsymbol{y}\right)^{\mathrm{T}}\Sigma^{-1}\left(\sqrt{\Sigma}\boldsymbol{y}\right)\right\}\left|\sqrt{\Sigma}\right|\mathrm{d}\boldsymbol{y}\\ &= \int\limits_{R^n} f\left(\sqrt{\Sigma}\boldsymbol{y}+\upmu\right)\frac{1}{\sqrt{|2\pi\mathbf{I}|}}\exp\left(-\boldsymbol{y}^{\mathrm{T}}\boldsymbol{y}\right)\mathrm{d}\boldsymbol{y}\\ &= \int\limits_{R^n} f\left(\sqrt{\Sigma}\boldsymbol{y}+\upmu\right)N(\boldsymbol{y};\ \mathbf{0},\ \mathbf{I})\mathrm{d}\boldsymbol{y}\\ &= \int\limits_{R^n} f\left(\sqrt{\Sigma}\boldsymbol{x}+\upmu\right)\omega_1(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\end{aligned} \tag{A1}$$

Substituting $\omega_1(\boldsymbol{x})$ and $\omega_2(\boldsymbol{x})$ into Equation (A1), we have:

$$\int\limits_{R^n} f(\boldsymbol{x})N(\boldsymbol{x};\ \upmu,\ \Sigma)\mathrm{d}\boldsymbol{x} = \int\limits_{R^n} f\left(\sqrt{\Sigma}\boldsymbol{x}+\upmu\right)N(\boldsymbol{x};\ \mathbf{0},\ \mathbf{I})\mathrm{d}\boldsymbol{x} \tag{A2}$$

References

1. Bar-Shalom, Y. *Multitarget-Multisensor Tracking: Advanced Application*; Artech House: Norwood, MA, USA, 1990.
2. He, Y.; Wang, G.; Lu, D.; Peng, Y. *Multisensor Information Fusion with Applications*, 2nd ed.; Publishing House of Electronics Industry: Beijing, China, 2010.
3. Zhang, K.; Shan, G.; Ji, B.; Chen, H. Maneuvering targets tracking based on pose-aided nonlinear filter algorithm. *Acta Electron. Sin.* **2012**, *40*, 1670–1675.
4. Liu, J.; Liu, Y.; He, Y.; Sun, S. Multi-sensor fuzzy data association based on azimuth and velocity. *Syst. Eng. Electron.* **2016**, *38*, 2040–2047.
5. Ge, Q.; Wei, Z.; Cheng, T.; Chen, C.; Wang, X. Flexible fusion structure-based performance optimization learning for multisensory target tracking. *Sensors* **2017**, *17*, 1045. [CrossRef] [PubMed]

6. Liu, Z.; Pan, Q.; Jean, D.; Gregoire, M. Hybrid Classification System for Uncertain Data. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *10*, 2783–2790. [CrossRef]
7. Taghavi, E.; Tharmarasa, R.; Kirubarajan, T.; Bar-Shalom, Y.; Mcdonald, M. A practical bias estimation algorithm for multisensor-multitarget tracking. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 2–19. [CrossRef]
8. Fantacci, C.; Papi, F. Scalable multisensor multitarget tracking using the marginalized-glmb densit. *IEEE Signal Process. Lett.* **2016**, *23*, 863–867. [CrossRef]
9. Si, W.; Wang, L.; Qu, Z. A measurement-driven adaptive probability hypothesis density filter for multitarget tracking. *Chin. J. Aeronaut.* **2015**, *28*, 1689–1698. [CrossRef]
10. ONeil, S.D.; Pao, L.Y. Multisensor fusion algorithm for tracking. In Proceedings of the American Control Conference, San Francisco, CA, USA, 2–4 June 1993; pp. 859–863.
11. Pao, L.Y.; Frei, C.W. A comparison of parallel and sequential implementation of a multisensory tracking algorithm. In Proceedings of the American control Conference, Seattle, WA, USA, 21–23 June 1995; pp. 1683–1687.
12. Bar-shalom, Y. *Multitarget-Multisensor Tracking: Principles and Techniques*; YBS Publishing: Stors, CT, USA, 1995.
13. Zhang, J.; He, Y.; Xiong, W. Centralized interacted multiple model multisensory fuzzy joint probabilistic data association algorithm. *Acta Electron. Sin.* **2008**, *36*, 1655–1659.
14. Han, C.; Zhu, Y.; Duan, Z.; Han, D.; Liu, W.; Yu, X. *Multi-Source Information Fusion*, 2nd ed.; Tsinghua University Press: Beijing, China, 2010.
15. Guan, X.; Zhou, X.; Rui, G. Centralized multisensor unscented joint probabilistic data association algorithm. *Syst. Eng. Electron.* **2009**, *31*, 2602–2606.
16. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [CrossRef]
17. Gao, X.; Chen, J.; Tao, D.; Li, X. Multi-sensor centralized fusion without measurement noise covariance by variational Bayesian approximation. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 718–727. [CrossRef]
18. Wang, X.; Pan, Q. Overview of deterministic sampling filtering algorithms for nonlinear system. *Control Decis.* **2012**, *27*, 801–812.
19. Arasaratnam, I.; Haykin, S. Cubature Kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [CrossRef]
20. Arasaratnam, I.; Haykin, S. Cubature Kalman smoothers. *Automatica* **2011**, *47*, 2245–2450. [CrossRef]
21. Liu, Y.; He, Y.; Wang, H. Squared-root cubature information consensus filter for non-linear decentralised state estimation in sensor networks. *IET Radar Sonar Navig.* **2014**, *8*, 931–938.
22. 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*, 1161–1174. [CrossRef]
23. Liu, Y.; Dong, K.; Wang, H.; Liu, J.; He, Y.; Pan, L. Adaptive Gaussian sum squared-root cubature Kalman filter with split-merge scheme for state estimation. *Chin. J. Aeronaut.* **2014**, *27*, 1242–1250. [CrossRef]
24. Liu, G.; Florentin, W.; Irene, M. Square-root sigma-point information filtering. *IEEE Trans. Autom. Control* **2012**, *57*, 2945–2950. [CrossRef]
25. Jia, B.; Sun, T.; Xin, M. Iterative diffusion-based distributed cubature Gaussian mixture filter for multisensory estimation. *Sensors* **2016**, *16*, 1741. [CrossRef] [PubMed]
26. Sun, F.; Tang, L. Estimation precision comparison of cubature Kalman filter and unscented Kalman filter. *Control Decis.* **2013**, *28*, 303–308.