sensors

MDPI

Article

Adaptive Interacting Multiple Model Algorithm Based on Information-Weighted Consensus for Maneuvering Target Tracking

Ziran Ding [1], Yu Liu [1,2,*], Jun Liu [1,*], Kaimin Yu [3], Yuanyang You [4], Peiliang Jing [5] and You He [1]

[1] Research Institute of Information Fusion, Naval Aviation University, Yantai 264001, China; DZR19931201@163.com (Z.D.); heyou_f@126.com (Y.H.)
[2] School of Electronic and Information Engineering, Beihang University, Beijing 100191, China
[3] The First Training Base, Naval Aviation University, Huludao 125001, China; dongyongdi725@163.com
[4] The Second Training Base, Naval Aviation University, Changzhi 046000, China; yyyy-abc@163.com
[5] China Ordnance Test Center, Huayin 714200, China; jingpeiliang@nudt.edu.cn
* Correspondence: liuyu77360132@126.com (Y.L.); 18615042187@163.com (J.L.); Tel.: +86-535-663-5877 (Y.L.)

Received: 5 May 2018; Accepted: 17 June 2018; Published: 22 June 2018

Abstract: Networked multiple sensors are used to solve the problem of maneuvering target tracking. To avoid the linearization of nonlinear dynamic functions, and to obtain more accurate estimates for maneuvering targets, a novel adaptive information-weighted consensus filter for maneuvering target tracking is proposed. The pseudo measurement matrix is computed with unscented transform to utilize the information form of measurements, which is necessary for consensus iterations. To improve the maneuvering target tracking accuracy and get a unified estimation in each sensor node across the entire network, the adaptive current statistical model is exploited to update the estimate, and the information-weighted consensus protocol is applied among neighboring nodes for each dynamic model. Based on posterior probabilities of multiple models, the final estimate of each sensor is acquired with weighted combination of model-conditioned estimates. Experimental results illustrate the superior performance of the proposed algorithm with respect tracking accuracy and agreement of estimates in the whole network.

Keywords: state estimation; maneuvering target tracking; multiple sensor fusion; consensus; interacting multiple model

1. Introduction

The maneuvering target tracking problem has drawn immense attention for many years in areas such as aircraft surveillance, radar tracking, and road vehicle navigation [1,2]. In the military field, to avoid being attacked, most of the enemy's targets have strong maneuvers. In this situation, traditional tracking algorithms are not as effective as they usually are. Once the targets maneuver, they will be out of detection soon. Maneuvering target tracking has become a common situation in the military field, the research of which is of great significance. However, as the target's maneuverability and the anti-tracking awareness of fighters are increased, tracking maneuvering targets becomes increasingly difficult.

In maneuvering target tracking, the uncertainty of the target motion model leads to large errors for tracking results. To solve the problem, a variety of methods, including the white noise model with adjustable level, the variable dimension model, input estimation model, Singer model, Jerk model, multiple model and interacting multiple model (IMM), have been proposed. Among them, IMM [3] has won great popularity. In maneuvering target tracking, the IMM algorithm has been shown to be one of the most cost-effective estimation schemes [4]. In IMM, the model set includes several models to deal with the varying characteristics of target motion, as it is assumed that the target trajectory evolves according to some predetermined models. However, multiple models sometimes cannot fully contain the target's motion, and excessive numbers of models will generate larger calculations. On the basis of IMM, the interacting multiple adaptive model (IMAM) [5] uses two adaptive current statistical models to form a model set. The IMAM has better adaptability for model uncertainty, and it can achieve better accuracy than IMM when the model set of IMM cannot contain the motion of the target.

The standard extended Kalman filter (EKF) has become the method of choice in a number of nonlinear estimations. The EKF uses Taylor series expansions to handle nonlinear states and observation transition functions. However, when the system is highly nonlinear, the linearization scheme will cause additional linearization errors [6]. The unscented Kalman filter (UKF) approximates the posterior probability distribution by a selected set of sigma points, which are propagated according to the unscented transformation [7]. The estimated mean and covariance are obtained by weighted sum of these unscented points. As for highly nonlinear systems, UKF is more accurate and less computational than EKF. The Particle Filter (PF) has a superior performance for solving nonlinear problem than EKF and UKF, and can even be applied to non-Gaussian problems [8]. However, it costs too much in computational resources.

Many studies focus on maneuvering target tracking under nonlinear system. Reference [9] uses the current statistical model to estimate the state of the maneuvering target, in which the adaptive process noise covariance is applied to reduce the influence of limited acceleration. To handle nonlinear systems, the current statistical model is based on UKF. Reference [10] proposed an efficient IMM-UKF algorithm. Apart from the advantages of IMM and UKF, the cubature-principle-assisted IMM method can deal with the non-Gaussian problem, and the new adaptive UKF algorithm addresses sensor faults. Based on PF, Reference [11,12] performed a very similar study. They both fused the information of multiple predefined models to obtain a more accurate result, and utilized PF to estimate the target. The difference is that [11] allocates particles on the basis of posterior probability of each model, while [12] achieves allocation according to predictive power. The allocation of particles can provide more resources to the model which is closer to the true target motion, and then this model can obtain a more accurate estimate at the next time. Reference [13] proposed Particle Learning (PL), showing that PL outperforms existing PF. In PL, particles are used to approximate joint posterior distribution of the state and static parameters, so they can describe the feature of target more accurately. Due to the large number of particles used in the computing process, the defect of these particle methods is their heavy calculation, which would be particularly obvious in the multi-sensor system. References [10–13] are only based on a single sensor, so they cannot utilize the superiority of sensor networks which fuse the information of multiple sensors.

In many tracking scenarios, multiple sensors are used to provide more complementary measurement information and higher tolerance about individual sensor failure. There are three main kinds of multi-sensor data fusion architectures: centralized, hierarchical distributed and distributed [14]. Centralized architecture is the traditional architecture in which measurements from all sensors are sent to the fusion center. The optimal estimation is obtained by processing and fusing the complete measurement sets of the whole system [15]. However, the computation and communication burden imposed on the fusion center are severe, only powerful equipment can accomplish this task. On the other hand, a center node's failure results in failure of the whole system. In a hierarchical distributed architecture, multiple sensors are managed in a hierarchy. The low-level sensors handle measurements and send results to higher level sensors. The higher-level sensors only need to transmit processing results; thus, they have a lower computation burden and require less communication bandwidth. The distributed architecture has no fusion center. Each sensor node only exchanges local state estimation information with its neighbor nodes. This architecture requires less communication bandwidth and has a lower computation burden in each sensor, and it is strongly tolerant for single sensor failure [16,17]. However, the information from neighbor sensors is limited, which will cause a decline in the estimation accuracy. Through the consensus iterations, the estimated value of each sensor gradually tends to the global optimal estimation [18,19], which ensures a consensual estimate of the situation and improves estimation accuracy for each sensor [20].

To solve the model uncertainty of the target by the input estimation technique, an adaptive particle filter for maneuvering target tracking [21] is proposed. The multiplicative measurement model is used to deal with the measurement noise (related to state) in the sensor network. In addition, all measurements of sensors are sent to the fusion center. In sensor networks, IMM is still a great algorithm to deal with the maneuver of the target [22]. The likelihoods of models are communicated to the central-level tracker along with the measurements, so the effects of sensor biases can be reduced. The distributed state estimation with model uncertainty is considered by [23]. The distributed mixture Kalman filter is an effective method to handle nonlinear system and maneuvering target. Based on a distributed architecture, average consensus is executed to improve the accuracy of all sensors.

In this paper, an adaptive interacting multiple model algorithm based on information-weighted consensus (IMAM-UICF) is proposed. This algorithm further improves the estimation accuracy of tracking maneuvering target on the basis of IMAM, and the consensus filter is the key to the improvement. Each sensor in the network uses IMAM to get a more realistic model hypothesis of target motion. The information filter can estimate the target of a nonlinear system by embedding the unscented transformation. Then each sensor node maintains a local estimate and interacts with neighbor nodes. After receiving information from neighbor nodes, each node corrects its local estimate according to the consensus protocol. The algorithm can accomplish the maneuvering target tracking task under nonlinear systems with high estimation accuracy.

The remainder of this paper is organized as follows. The sensor communication topology and system model are defined in Section 2. The nonlinear interacting multiple adaptive model is described in Section 3. Our algorithm IMAM-UICF is proposed in Section 4. Experimental evaluation in Section 5 demonstrates the effectiveness of the proposed algorithm and comparisons with other methods. Finally, the conclusions are provided in Section 6.

2. Problem Formulation

2.1. Communication Network

The communication network topology between multiple sensors is represented by the adjacency matrix based on graph theory. A graph is defined as $G = (N, E)$, where $N = \{1, 2, \cdots, n\}$ is the set of sensor nodes and $E \subseteq N \times N$ is a set of edges in graph G. The edges represent the available communication channels between two nodes. $N_i = \{j \in N, (j, i) \in E\}$ denotes the set of neighbor sensors that can communicate with sensor i.

Adjacency matrix $\boldsymbol{A} = [a_{ij}]_{i,j=1}^{n}$ is defined as $a_{ij} = \begin{cases} 1 & if(j,i) \in E \\ 0 & otherwise \end{cases}$, where $a_{ij} = 1$ indicates that there is a communication link between node i and node j, and $a_{ij} = 0$ indicates that node i cannot communicate with node j. In Figure 1, for example, there are 4 sensors in the network.

The corresponding network topology can be denoted by the following adjacency matrix:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1}$$

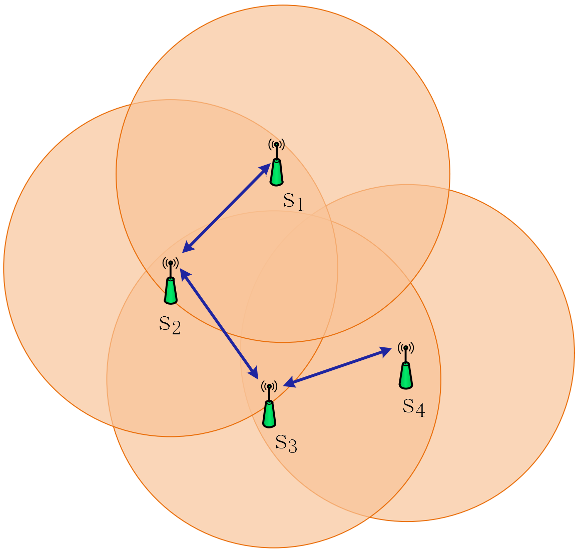

Figure 1. Communication network. s_i represents the number of the sensor. The dotted line denotes the communication range of the sensor. The blue arrow represents the communication between the sensors.

2.2. System Modeling

The state variable method is a valuable method to describe the dynamic system. The relationship between input and output of the system is described in the time domain using the state transfer model and output observation model. The input can be described by the state equation. The state equation consists of a determined time function and a stochastic process, and the stochastic process represents unpredictable noise. The output is a function of the state vector. The output is usually disturbed by the stochastic observation error and can be described by the measurement equation.

State equation:

$$\boldsymbol{x}(k) = f(\boldsymbol{x}(k-1)) + G\overline{\boldsymbol{a}}(k-1) + \boldsymbol{w}(k-1) \tag{2}$$

where $\boldsymbol{x}(k)$ represents the estimated state of target at the filtering time k, and the process noise is subject to $\boldsymbol{w}(k-1) \sim N(0, \boldsymbol{Q}(k-1))$. $f(\cdot)$ denotes state transfer function and G denotes input control matrix, $\overline{\boldsymbol{a}}(k-1)$ is maneuvering acceleration mean.

Measurement equation:

$$\boldsymbol{z}(k) = h(x(k)) + \boldsymbol{r}(k) \tag{3}$$

where $\boldsymbol{z}(k)$ represents the measurement of sensor, the measurement noise $\boldsymbol{r}(k)$ is subject to $\boldsymbol{r}(k) \sim N(0, \boldsymbol{R}(k))$, and $h(\cdot)$ denotes the measurement transfer function.

When networked multiple sensors observe a target, they get different measurements due to their different observation abilities. Assuming that there are n sensors, the measurement in Equation (3) should be rewritten as:

$$\boldsymbol{z}_i(k) = h_i(\boldsymbol{x}(k)) + \boldsymbol{r}_i(k), i = 1, 2, \cdots, n \tag{4}$$

How to fuse multiple measurements from different sensors to obtain the global optimal estimation is the main problem that consensus-based distributed filtering algorithms take into consideration.

When a maneuvering target is tracked, the motion state of the target is changeable, that is, there are multiple possible state transfer functions $f^r(\cdot), r = 1, 2, \cdots, m$.

Then the state equation of the target should be rewritten as:

$$\boldsymbol{x}(k) = f^r(\boldsymbol{x}(k-1)) + G^r\overline{\boldsymbol{a}}(k-1) + \boldsymbol{w}^r(k-1), r = 1, 2, \cdots, m \tag{5}$$

where $r = 1, 2, \cdots, m$ denotes the r_{th} model, and m is the number of models.

3. Nonlinear Interacting Multiple Adaptive Model

3.1. Interacting Multiple Adaptive Model

IMAM [5] is an interacting multiple model method that selects two ACS models with different maneuver frequencies to form a model set.

In the current statistical (CS) model, the fixed maximum acceleration is one reason that results in the increase of the model's error after the target maneuvers. Adaptive current statistical (ACS) model is a model that can adaptively adjust the maximum acceleration. It can improve the CS model's ability to respond to maneuvering targets, and ensure that the CS model has an accurate description of maneuvering targets.

3.1.1. Current Statistical Model

In the IMM [24], the system model is composed of multiple motion models. Target state is respectively estimated by each model in the tracking process, and the probability of each model is constantly adjusted at the same time. Considering the probability of each mode as the corresponding weight, the final estimate of IMM is the weighted sum of the estimated value of each model.

Model Interaction

The transition probability of the moving state of the target is defined as

$$\boldsymbol{P} = \begin{bmatrix} P(1,1) & P(1,2) & \cdots & P(1,m) \\ P(2,1) & P(2,2) & \cdots & P(2,m) \\ \vdots & \vdots & \ddots & \vdots \\ P(m,1) & P(m,2) & \cdots & P(m,m) \end{bmatrix} \tag{6}$$

where m is the number of models in the model set.

$u_{j|r}(k-1\big|k-1)$ is the probability that the model r is converted from the model j.

$$u_{j|r}(k-1|k-1) = \frac{1}{\overline{C_j}}\boldsymbol{P}(j,r)u_r(k-1) \tag{7}$$

where $u_r(k-1)$ is the probability of model r at time $k-1$, $\overline{C_j} = \sum\limits_{r=1}^{m} \boldsymbol{P}(j,r)u_r(k-1)$.

$\boldsymbol{x}^{or}(k-1|k-1)$ is the state estimation of model r at time $k-1$, $\boldsymbol{P}^{or}(k-1|k-1)$ is the corresponding state covariance. The input of the mode r after the interactive calculation is as follows

$$\boldsymbol{x}^{or}(k-1|k-1) = \sum_{j=1}^{m} \boldsymbol{x}_j(k-1\big|k-1)u_{j|r}(k-1\big|k-1) \tag{8}$$

$$\begin{aligned} \boldsymbol{P}^{or}(k-1|k-1) = \sum_{j=1}^{m} \{ & \boldsymbol{P}_j(k-1|k-1) + [\boldsymbol{x}_j(k-1|k-1) - \boldsymbol{x}^{or}(k-1|k-1)] \\ & [\boldsymbol{x}_j(k-1|k-1) - \boldsymbol{x}^{or}(k-1|k-1)]^{\mathrm{T}} \} u_{j|r}(k-1|k-1) \end{aligned} \tag{9}$$

Model-Conditioned Filtering

Use state vector $\boldsymbol{x}^{or}(k-1|k-1)$, covariance $\boldsymbol{P}^{or}(k-1|k-1)$ and measurement $\boldsymbol{z}(k)$ as the input of model r at time k. Each model obtains output $\boldsymbol{x}_r(k|k)$ and $\boldsymbol{P}_r(k|k)$ based on its own model and input.

Model-Conditioned Filtering

Assuming that the innovation $\boldsymbol{\nu}_r(k)$ obeys the Gauss distribution, the likelihood function is $\Lambda_r(k)$.

$$\Lambda_r(k) = \frac{1}{\sqrt{|2\pi \boldsymbol{S}_r(k)|}} \exp[-\frac{1}{2}(\boldsymbol{\nu}_r(k))^{\mathrm{T}}(\boldsymbol{S}_r(k))^{-1}\boldsymbol{\nu}_r(k)] \tag{10}$$

where $\boldsymbol{\nu}_r(k) = \boldsymbol{z}(k) - \boldsymbol{H}_r(k)\boldsymbol{x}_r(k|k-1)$, and $\boldsymbol{S}_r(k)$ is its covariance, $\boldsymbol{S}_r(k) = \boldsymbol{H}_r(k)\boldsymbol{P}_r(k\Big|k-1)(\boldsymbol{H}_r(k))^{\mathrm{T}} + \boldsymbol{R}^{-1}$.

The posteriori model probability of model r is calculated as follows:

$$u_r(k) = \frac{1}{C}\Lambda_r(k)\overline{C_r} \tag{11}$$

where $C = \sum\limits_{r=1}^{m} \Lambda_r(k)\overline{C_r}$.

State Combination

The combined state $\boldsymbol{x}(k|k)$ and its covariance $\boldsymbol{P}(k|k)$ are calculated as

$$\boldsymbol{x}(k|k) = \sum_{r=1}^{m} \boldsymbol{x}_r(k|k)u_r(k) \tag{12}$$

$$\boldsymbol{P}(k|k) = \sum_{r=1}^{m} u_r(k)\{\boldsymbol{P}_r(k|k) + [\boldsymbol{x}_r(k|k) - \boldsymbol{x}(k|k)]\cdot [\boldsymbol{x}_r(k|k) - \boldsymbol{x}(k|k)]^{\mathrm{T}}\} \tag{13}$$

3.1.2. Current Statistical Model

In the current statistical model, the process noise covariance matrix is time-varying. For example, there is an estimate of one-dimensional coordinate $\boldsymbol{x} = [x\,\dot{x}\,\ddot{x}]^{\mathrm{T}}$, its process noise covariance matrix is

$$\boldsymbol{Q}(k) = 2\alpha\sigma_a^2(k)\boldsymbol{q} \tag{14}$$

where α is the maneuver frequency, $\boldsymbol{q}$ is a constant matrix related to α and the sampling period T, and its expression can be referred to literature [25].

$\sigma_a^2(k)$ represents the variance of the maneuvering acceleration at time k, and it is adaptively adjusted at each moment. Its value is

$$\sigma_a^2(k) = \begin{cases} \frac{4-\pi}{\pi}[a_{\max} - \overline{a}(k)]^2, \overline{a}(k) > 0 \\ \frac{4-\pi}{\pi}[a_{-\max} - \overline{a}(k)]^2, \overline{a}(k) < 0 \end{cases} \tag{15}$$

where $\overline{a}(k) = \hat{\ddot{x}}(k|k-1)$. i is the maximum acceleration. In the traditional current statistical model, $a_{\max}$ is a fixed value given in advance according to experience.

3.1.3. Adaptive Current Statistical Model

Structure membership function [5]:

$$u_\delta(k) = \begin{cases} 1 - e^{-\nu(k)^2/(R_1 \cdot c)}, |\nu(k)| < 3R_1 \\ 1 - e^{-\nu(k)^4/(R_1 \cdot c)}, |\nu(k)| \geq 3R_1 \end{cases} \tag{16}$$

where $\nu(k)$ represents the innovation at time k, R_1 denotes the position measurement noise, and c denotes the position estimation covariance

In the ACS model, the maximum acceleration $a_{\max}$ adaptively changes according to the following equation:

$$a_{\max} = \overline{a}(k) + 2u_\delta(k) \cdot |\nu(k)| \tag{17}$$

The ACS model adjusts the maximum acceleration $a_{\max}$ at each moment according to the innovation, making it include the true acceleration of the target motion as much as possible. This method improves the response of the CS to the maneuver, which results in a better tracking effect. Selecting two ASC models, IMAM quickly adapts to the mutation of the target movement. Comparing to the traditional IMM, IMAM has a better performance for maneuvering target tracking.

3.2. Nonlinear Interacting Multiple Adaptive Model

In our algorithm, the Unscented Transformation (UT) of UKF [26] is chosen to deal with nonlinear problems. In nonlinear systems, UT transfers statistical characteristics through a set of random sampling points, so the prediction can be done. The predicted state $\boldsymbol{x}(k|k-1)$, predicted covariance $\boldsymbol{P}(k|k-1)$ and predicted measurement $\boldsymbol{z}(k|k-1)$ can be obtained by using the prediction of sigma points and corresponding weight. The details of UT are explained below.

In the unscented transformation, $(2n_x+1)$ sigma points are usually selected for the target state

$$\begin{cases} \xi_0(k-1|k-1) = \boldsymbol{x}^{or}(k-1|k-1) & p=0 \\ \xi_p(k-1\Big|k-1) = \boldsymbol{x}^{or}(k-1\Big|k-1) + (\sqrt{(n_x+\kappa)\boldsymbol{P}^{or}(k-1|k-1)}\,)_p & p=1,\cdots,n_x \\ \xi_{p+n_x}(k-1\Big|k-1) = \boldsymbol{x}^{or}(k-1\Big|k-1) - (\sqrt{(n_x+\kappa)\boldsymbol{P}^{or}(k-1|k-1)}\,)_p & p=1,\cdots,n_x \end{cases} \tag{18}$$

where κ is a scale parameter, usually $\kappa = n_x(\tau^2-1)$, the range of the parameter τ is $0.0001 \leq \tau \leq 1$; $(\sqrt{(n_x+\kappa)\boldsymbol{P}^{or}(k-1|k-1)}\,)_p$ is the p_{th} row or the p_{th} column of the root mean square matrix $\sqrt{(n_x+\kappa)\boldsymbol{P}^{or}(k-1|k-1)}$; n_x is the dimension of the state vector.

These sigma points are symmetric distributions of the mean target state, and they can describe the random quantity of the Gauss distribution well. The weights corresponding to sigma points are

$$W_h = \begin{cases} \frac{\kappa}{(n_x+\kappa)} & h=0 \\ \frac{1}{2(n_x+\kappa)} & h=1,2,\cdots,2n_x \end{cases} \tag{19}$$

A set of precise selected sigma points are transformed through the known nonlinear function. These points are used to transfer the statistical characteristics of random quantities. These characteristics are the probability distribution's statistical moments: mean and covariance.

The prediction of state sigma points are

$$\xi_h(k|k-1) = f^r(\xi_h(k-1|k-1)) \tag{20}$$

According to the nonlinear function propagation method of the unscented transformation, the predicted state $\boldsymbol{x}(k|k-1)$ and the predicted covariance of state $\boldsymbol{P}(k|k-1)$ can be obtained by using the prediction of state sigma points and corresponding weight

$$\boldsymbol{x}(k|k-1) = \sum_{h=0}^{2n_x} W_h \xi_h(k|k-1) + G^r \overline{a}(k-1) \tag{21}$$

$$\boldsymbol{P}(k|k-1) = \sum_{h=0}^{2n_x} W_h \Delta\boldsymbol{x}_h(k\Big|k-1)\Delta\boldsymbol{x}_h^{\mathrm{T}}(k\Big|k-1) + \boldsymbol{Q}^r(k-1) \tag{22}$$

where $\Delta\boldsymbol{x}_h(k|k-1) = \xi_h(k|k-1) - \boldsymbol{x}(k|k-1)$.

The prediction of measurement sigma points are $\varsigma_h(k|k-1) = h(k,\xi_h(k|k-1))$.

The weighted sum of the predicted measurement sigma points is the predicted measurement

$$\boldsymbol{z}(k|k-1) = \sum_{h=0}^{2n_x} W_h \varsigma_h(k|k-1) \tag{23}$$

The cross-correlation covariance of state and measurement is

$$\boldsymbol{P}^{xz}(k|k-1) = \sum_{h=0}^{2n_x} W_h \Delta \boldsymbol{x}_h \left(k \middle| k-1\right) \Delta \boldsymbol{z}_h^{\mathrm{T}} \tag{24}$$

where $\Delta \boldsymbol{z}_h = \varsigma_h(k|k-1) - \boldsymbol{z}(k|k-1)$.

The research in [20] shows that, compared to common Kalman filtering, the consensus protocol can play a better role in Information Filtering (IF). Therefore, IF is selected as the basic filtering method of every model in the model set. In information filtering, update of the information matrix and information vector are based on the linear measurement transfer matrix. In order to update information in a nonlinear system, a pseudo measurement matrix is defined [27]. According to $\boldsymbol{P}^{xz}(k|k-1) = \boldsymbol{P}(k|k-1)\boldsymbol{H}^{\mathrm{T}}(k)$, we get $\boldsymbol{H}^{\mathrm{T}}(k) = \left(\boldsymbol{P}(k|k-1)\right)^{-1}\boldsymbol{P}^{xz}\left(k \middle| k-1\right)$.

Then the update of information matrix and information vector [20] are

$$\boldsymbol{Y} = \frac{1}{n}\left(\boldsymbol{P}(k|k-1)\right)^{-1} + \boldsymbol{H}^{\mathrm{T}}(k)\boldsymbol{R}^{-1}\boldsymbol{H}(k) \tag{25}$$

$$\boldsymbol{y} = \frac{1}{n}(\boldsymbol{P}(k|k-1))^{-1}\boldsymbol{x}(k|k-1) + \boldsymbol{H}^{\mathrm{T}}(k)\boldsymbol{R}^{-1}(\boldsymbol{H}(k)\boldsymbol{x}(k|k-1) + \boldsymbol{z}(k) - \boldsymbol{z}(k|k-1)) \tag{26}$$

where n is the number of sensors in the sensor network.

In linear systems, the innovation $v(k) = z(k) - z(k|k-1) = x_z(k) - x_z(k|k-1)$ represents the difference between the measurement and the predicted measurement in the x direction. In Equation (16), the innovation is used to calculate membership function. Then we adaptively adjust the maximum acceleration in the x direction according to the membership function. However, in nonlinear systems, the innovation $\boldsymbol{v}(k) = \boldsymbol{z}(k) - \boldsymbol{z}(k|k-1) = \left|\begin{matrix}\rho_z(k) \\ \theta_z(k)\end{matrix}\right| - \left|\begin{matrix}\rho_z(k|k-1) \\ \theta_z(k|k-1)\end{matrix}\right|$ cannot be used to calculate membership function. Through theoretical analysis and testing, we find that using $[\rho_z(k) - \rho_z(k|k-1)] \times \cos[\theta_z(k) - \theta_z(k|k-1)]$ instead of the innovation to calculate the membership function will return results with larger errors. In contrast, using $\rho_z(k) \times \cos\theta_z(k) - \rho_z(k|k-1) \times \cos\theta_z(k|k-1)$ instead of the innovation, we get results that describe the difference between the measurement and the predicted measurement more accurately.

Based on the above three methods, the proposed algorithm has the ability to track maneuvering targets in a nonlinear system. In addition, errors caused by nonlinear problems are controlled within a very small range.

4. Adaptive Interacting Multiple Model Algorithm Based on Information-Weighted Consensus

Considering the changeable moving state of a maneuvering target, we use the concept of IMAM to get a set of system models that are closer to the actual situation. This can compensate for the decline of tracking performance caused by the target maneuver. At the same time, considering the limited detection and survivability of single sensor, we build a multi-sensor network system to track the target. Multi-sensor data fusion can improve the tracking accuracy, and the distributed architecture ensures the robustness and flexibility of the system. The consensus protocol is combined with the distributed architecture, and it overcomes the shortage of distributed structure that improves the consensus of multiple sensors' estimation situations and estimate accuracy [28].

4.1. Average Consensus

Average consensus is an effective method to fuse the information of multiple distributed sensors. It is the most popular consensus protocol and has many applications in filter algorithms. Average consensus makes the estimates converge to their mean. For example, there is a network of n

nodes, and each node has an estimate value a_i. $a_i(0) = a_i$ is the initial value of each node, then runs the iterative formula for L times:

$$a_i(l) = a_i(l-1) + e \sum_{s \in N_i} (a_s(l-1) - a_i(l-1)) \tag{27}$$

Before iteration l, each node sends its estimate value $a_i(l-1)$ to its neighbor nodes s ($s \in N_i$) and receives neighbor node estimate value $a_s(l-1)$. Then new estimate value $a_i(l)$ is calculated as Equation (27). After L iterations, the estimate values of all nodes converge to the average of their initial values $\frac{1}{n}\sum\limits_{i=1}^{n} a_i(0)$. In addition, theoretically, when $L \to \infty$, all final value $a_i(L)$ is equal to $\frac{1}{n}\sum\limits_{i=1}^{n} a_i(0)$. Based on all estimates a_i, $\frac{1}{n}\sum\limits_{i=1}^{n} a_i(0)$ is the optimal estimate that nodes in the communication network could get. The final estimates of all nodes $a_i(L)$ are close to the mean $\frac{1}{n}\sum\limits_{i=1}^{n} a_i(0)$, this is the reason they have low disagreement.

The value of consensus weight e has a limit $(0, \frac{1}{\Delta_{\max}})$, where $\Delta_{\max}$ is the maximum degree of the communication network graph. In addition, each pair of nodes could have different weights, even the weights can be time-varying. So e should be replaced by $e_{i,s}(k)$. Between 0 and $\frac{1}{\Delta_{\max}}$, there are many options for its value. In our algorithm, we choose Metropolis weights as consensus weights. Metropolis weights can provide a fast convergence rate [29]. They also have good effect in distributed architecture, because nodes do not need to know the communication graph and even the number of nodes. Their values are

$$e_{i,s}(k) = \begin{cases} \frac{1}{1+\max\{d_i(k), d_s(k)\}} & if \quad s \in N_i \\ 1 - \sum\limits_{s \in N_i} e_{i,s}(k) & if \quad i = s \\ 0 & otherwise \end{cases} \tag{28}$$

where $d_i(k)$ represents the number of sensor i's neighbors at time k.

4.2. Interacting Multiple Adaptive Model-Unscented Information Consensus Filter

To state the proposed method more clearly, the algorithm flow of IMAM-UICF is illustrated in Algorithm 1.

Algorithm 1. IMAM-UICF for sensor i at time instant k.

```
Input: state $\mathbf{x}_{i,r}(k-1|k-1)$ and error covariance $\boldsymbol{P}_{i,r}(k-1|k-1)$, process noise covariance $\boldsymbol{Q}^r(k-1)$,
hybrid probability $u_{i,j|r}(k-1|k-1)$, measurement $z_i(k)$;
Output: state $\mathbf{x}_{i,r}(k|k)$ and error covariance $\boldsymbol{P}_{i,r}(k|k)$, process noise covariance $\boldsymbol{Q}^r(k)$, hybrid probability
$u_{i,j|r}(k|k)$, combination state $\mathbf{x}_i(k|k)$ and error covariance $\boldsymbol{P}_i(k|k)$;

Step 1. Interacting
$\mathbf{x}_i^{or}(k-1|k-1) = \sum_{j=1}^{m} \mathbf{x}_{i,j}(k-1|k-1) u_{i,j|r}(k-1|k-1)$
  $\boldsymbol{P}_i^{or}(k-1|k-1) = \sum_{j=1}^{m} \{ \boldsymbol{P}_{i,j}(k-1|k-1) + [\mathbf{x}_{i,j}(k-1|k-1) - \mathbf{x}_i^{or}(k-1|k-1)]$
  $[\mathbf{x}_{i,j}(k-1|k-1) - \mathbf{x}_i^{or}(k-1|k-1)]^{\mathrm{T}} \} u_{i,j|r}(k-1|k-1)$
```

```
Step 2. Filtering
  (1) prediction (based on the UT)
    $\boldsymbol{x}_{i,r}(k|k-1) = \sum_{h=0}^{2n_x} W_h \xi_h(k|k-1) + G^r \overline{a}(k-1)$
    $\boldsymbol{z}_{i,r}(k|k-1) = \sum_{h=0}^{2n_x} W_h \varsigma_h(k|k-1)$
    $\boldsymbol{P}_{i,r}(k|k-1) = \sum_{h=0}^{2n_x} W_h \Delta\boldsymbol{x}_{i,h}(k\Big|k-1)\Delta\boldsymbol{x}_{i,h}^{\mathrm{T}}(k\Big|k-1) + \boldsymbol{Q}^r(k-1)$
  (2) update process noise covariance
    $\sigma_a^2(k) = \begin{cases} \frac{4-\pi}{\pi}[a_{\max} - \overline{a}(k)]^2, \overline{a}(k) > 0 \\ \frac{4-\pi}{\pi}[a_{-\max} - \overline{a}(k)]^2, \overline{a}(k) < 0 \end{cases}$
    $\boldsymbol{Q}^r(k) = 2\alpha\sigma_a^2(k)\boldsymbol{q}$
  (3) define pseudo measurement matrix
    $\boldsymbol{P}_{i,r}^{xz}(k|k-1) = \sum_{h=0}^{2n_x} W_h \Delta\boldsymbol{x}_{i,h}(k|k-1)\Delta\boldsymbol{z}_{i,h}^{\mathrm{T}}$
    $\boldsymbol{H}_{i,r}^{\mathrm{T}}(k) = (\boldsymbol{P}_{i,r}(k|k-1))^{-1}\boldsymbol{P}_{i,r}^{xz}(k\Big|k-1)$
  (4) update information matrix and information vector
    $\boldsymbol{Y}_{i,r}^0 = \frac{1}{n}(\boldsymbol{P}_{i,r}(k|k-1))^{-1} + \boldsymbol{H}_{i,r}^{\mathrm{T}}(k)\boldsymbol{R}_i^{-1}\boldsymbol{H}_{i,r}(k)$
    $\boldsymbol{y}_{i,r}^0 = \frac{1}{n}(\boldsymbol{P}_{i,r}(k|k-1))^{-1}\boldsymbol{x}_{i,r}(k|k-1) + \boldsymbol{H}_{i,r}^{\mathrm{T}}(k)\boldsymbol{R}_i^{-1}(\boldsymbol{H}_{i,r}(k)\boldsymbol{x}_{i,r}(k|k-1) + \boldsymbol{z}_i(k) - \boldsymbol{z}_{i,r}(k|k-1))$
  (5) perform consensus on $\boldsymbol{Y}_{i,r}^0$ and $\boldsymbol{y}_{i,r}^0$
    for $l = 1 : L$ do
      (a) send $Y_{i,r}^{l-1}$ and $y_{i,r}^{l-1}$ to all neighbours $s \in N_i$
      (b) receive $Y_{s,r}^{l-1}$ and $y_{s,r}^{l-1}$ from all neighbours $s \in N_i$
      (c) update consensus terms:
        $\boldsymbol{Y}_{i,r}^l = \boldsymbol{Y}_{i,r}^{l-1} + e_{i,s}(k)\sum_{s\in N_i}(\boldsymbol{Y}_{s,r}^{l-1} - \boldsymbol{Y}_{i,r}^{l-1})$
        $\boldsymbol{y}_{i,r}^l = \boldsymbol{y}_{i,r}^{l-1} + e_{i,s}(k)\sum_{s\in N_i}(\boldsymbol{y}_{s,r}^{l-1} - \boldsymbol{y}_{i,r}^{l-1})$
    end for
  (6) compute a posteriori state estimate and covariance
    $\boldsymbol{x}_{i,r}(k\Big|k) = (\boldsymbol{Y}_{i,r}^L)^{-1}\boldsymbol{y}_{i,r}^L$
    $\boldsymbol{P}_{i,r}(k\Big|k) = (n\boldsymbol{Y}_{i,r}^L)^{-1}$
Step 3. Update mode probability
  $\Lambda_{i,r}(k) = \frac{1}{\sqrt{|2\pi\boldsymbol{S}_{i,r}(k)|}}\exp[-\frac{1}{2}(\boldsymbol{\nu}_{i,r}(k))^{\mathrm{T}}(\boldsymbol{S}_{i,r}(k))^{-1}\boldsymbol{\nu}_{i,r}(k)]$
  $u_{i,r}(k) = \frac{1}{C}\Lambda_{i,r}(k)\overline{C_{i,r}}$
Step 4. Combination
  $\boldsymbol{x}_i(k|k) = \sum_{r=1}^{m}\boldsymbol{x}_{i,r}(k|k)u_{i,r}(k)$
  $\boldsymbol{P}_i(k|k) = \sum_{r=1}^{m}u_{i,r}(k)\{\boldsymbol{P}_{i,r}(k|k) + [\boldsymbol{x}_{i,r}(k|k) - \boldsymbol{x}_i(k|k)]\cdot\ [\boldsymbol{x}_{i,r}(k|k) - \boldsymbol{x}_i(k|k)]^{\mathrm{T}}\}$
```

4.3. Distributed Architecture

In our algorithm, multiple sensors are used to track the target and they cooperate in the distributed architecture. Multiple fusion nodes process data from their own sensors and communicate with neighbor nodes. The data from other sensors can provide more information to improve upon the local results [30,31]. We perform distributed computing on each model separately so that more accurate model probabilities can be obtained. In this case, the model interaction of IMAM can play a greater role. The structure of the algorithm is shown in Figure 2. To improve the tracking effect and the agreement between sensors, distributed computing is based on a consensus protocol.

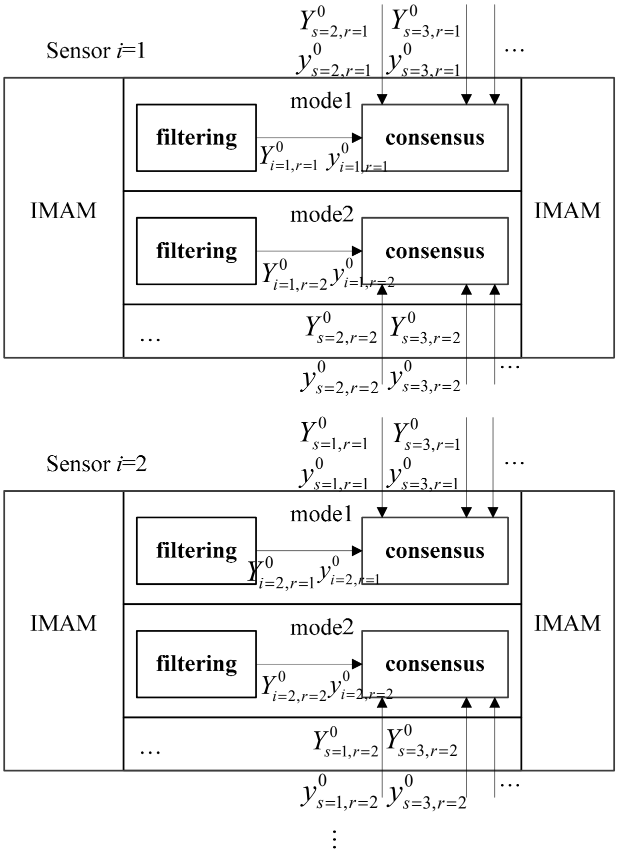

Figure 2. Structure of Interacting Multiple Adaptive Model-Unscented Information Consensus Filter.

5. Experimental Evaluation

5.1. Verification Experiment

To test the effectiveness of the proposed IMAM-UICF, a simple maneuvering target tracking problem is considered. The target motion switches between constant velocity model and constant acceleration model. A 6-D vector is chosen to express the state of target, including 2-D position(x, y), 2-D velocity$(\upsilon_\mathrm{x}, \upsilon_\mathrm{y})$ and 2-D accelerated velocity(a_x, a_y). The initial truth state of target is $\boldsymbol{x}(0) = [10080, 8000, -5.8, -11.6, 0, 0]^\mathrm{T}$. From 1s to 60s, target moves at constant velocity. From 61s to 110s, target moves at constant acceleration. From 111s to 170s, target moves at constant velocity. From 171s to 220s, target moves at constant acceleration. From 221s to 300s, target moves at constant velocity. The target motion can be described by constant velocity model

$$\boldsymbol{x}(k+1) = \begin{bmatrix} 1\,0\,T\,0\,0\,0 \\ 0\,1\,0\,T\,0\,0 \\ 0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \end{bmatrix} \boldsymbol{x}(k) + \begin{bmatrix} 0.5T^2\,0 \\ 0\,0.5T^2 \\ T \quad 0 \\ 0 \quad T \\ 0 \quad 0 \\ 0 \quad 0 \end{bmatrix} w_{CV}(k) \tag{29}$$

and constant acceleration model

$$\boldsymbol{x}(k+1) = \begin{bmatrix} 1 & 0 & T & 0 & 0.5T^2 & 0 \\ 0 & 1 & 0 & T & 0 & 0.5T^2 \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}(k) + \begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{w}_{CA}(k) \tag{30}$$

where $T = 1s$ is the sampling interval.

A sensor network consisting of five sensor nodes is deployed to track the target. Sensors can measure distance and direction of the target according to the following equation

$$\boldsymbol{z}_i(k) = \begin{bmatrix} \sqrt{x^2 + y^2} \\ \arctan(\frac{y}{x}) \end{bmatrix} + \boldsymbol{r}_i(k) \quad i = 1, 2, \cdots, 5 \tag{31}$$

The five sensors have different detection capabilities, so measurement noise covariance $\boldsymbol{R}_i$ of sensor i is $\sqrt{i} \times diag([30^2, (0.3\pi/180)^2])$. On the other hand, the five sensors are all static, so the position of sensor nodes is not considered. In addition, it is assumed that the target moves within all sensors' detection range.

Considering the limited energy and communication bandwidth in actual situations, the number of consensus iterations is set to $L = 8$. The communication channels among five nodes can be expressed by adjacency matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{32}$$

Model Parameters

In IMAM, two CS models are chosen to estimate state of target and interact. The model parameters of five sensors are the same.

In model 1, the initial state $x^1(0|0)$ and covariance $\boldsymbol{P}^1(0|0)$ of each sensor is equal, $\boldsymbol{x}^1(0\big|0) = [10000, 7900, -6, -10, 0, 0]^{\mathrm{T}}$ and $\boldsymbol{P}^1(0|0) = diag([0.1, 0.1, 0.1, 0.1, 0.1, 0.1])$. The maneuver frequency is $\alpha = 0.1$. The process noise variance matrix Q^1 is set to

$$Q^1 = \begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \\ 1 & 0 \\ 0 & 1 \end{bmatrix} [0.04\,0.09] \begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \\ 1 & 0 \\ 0 & 1 \end{bmatrix}^{\mathrm{T}} \tag{33}$$

And state transition matrix

$$F^1 = \begin{bmatrix} 1 & 0 & T & 0 & (\alpha T - 1 + e^{-\alpha T})/\alpha^2 & 0 \\ 0 & 1 & 0 & T & 0 & (\alpha T - 1 + e^{-\alpha T})/\alpha^2 \\ 0 & 0 & 1 & 0 & (1 - e^{-\alpha T})/\alpha & 0 \\ 0 & 0 & 0 & 1 & 0 & (1 - e^{-\alpha T})/\alpha \\ 0 & 0 & 0 & 0 & e^{-\alpha T} & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-\alpha T} \end{bmatrix} \tag{34}$$

In model 2, the initial state $x^2(0|0)$ and covariance $P^2(0|0)$ of each sensor is equal, $x^2(0\big|0) = [10000, 7900, -6, -10, 0, 0]^{\mathrm{T}}$ and $P^2(0|0) = diag([0.1, 0.1, 0.1, 0.1, 0.1, 0.1])$. The maneuver frequency is $\alpha = 0.0045$. The process noise variance matrix Q^2 is equal to (33), and state transition matrix F^2 is calculated as Equation (34).

The true motion of maneuvering target is uncertain. At each time step, the model probability is used to represent the likelihood that the target is moving in a certain model. The initial model probability is chosen to be $u_1(0) = 0.5$ and $u_2(0) = 0.5$. At the same time, the possibility of model conversion also needs to be considered. The transition probability is set to

$$P = \begin{bmatrix} 0.90 & 0.10 \\ 0.10 & 0.90 \end{bmatrix} \tag{35}$$

Results and Analysis

The true trajectory and the estimated trajectory of multiple sensors are shown in Figure 3. The root mean square errors of all sensors' estimated position and estimated velocity are shown in Figures 4 and 5, in which each color corresponds to the root mean square error (RMSE) of each sensor. After a few times, the RMSEs converge to small values. Every sensor has a good track result. That proves our algorithm has an effective tracking performance. One advantage of the algorithm is that it can improve the consensus of multiple sensors' estimation situations. The disagreement between all sensors is used to evaluate this performance. The result is shown in Figure 6.

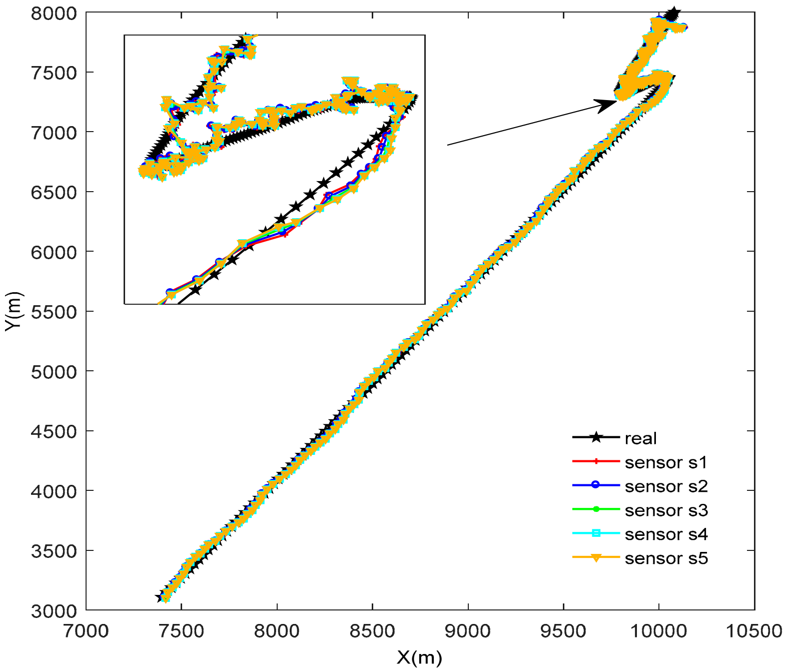

Figure 3. Real and estimated target trajectory.

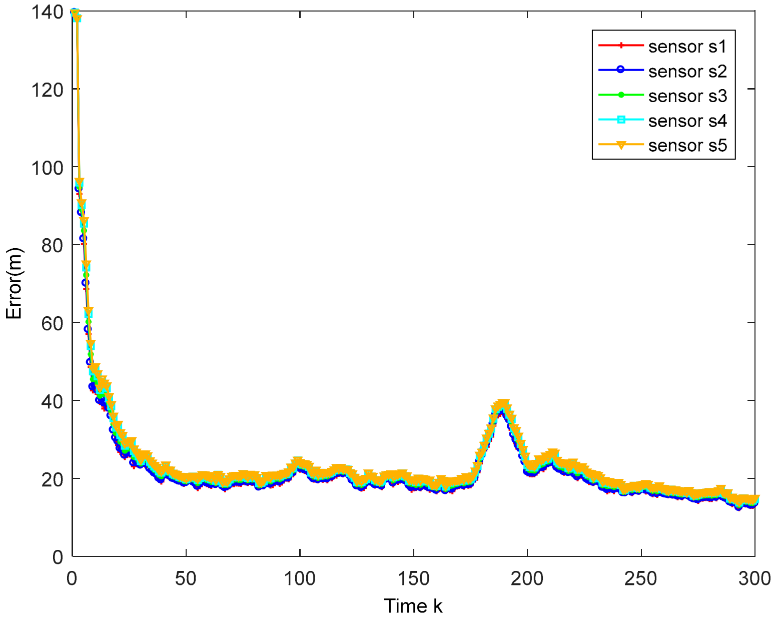

Figure 4. Position RMSE of different sensors.

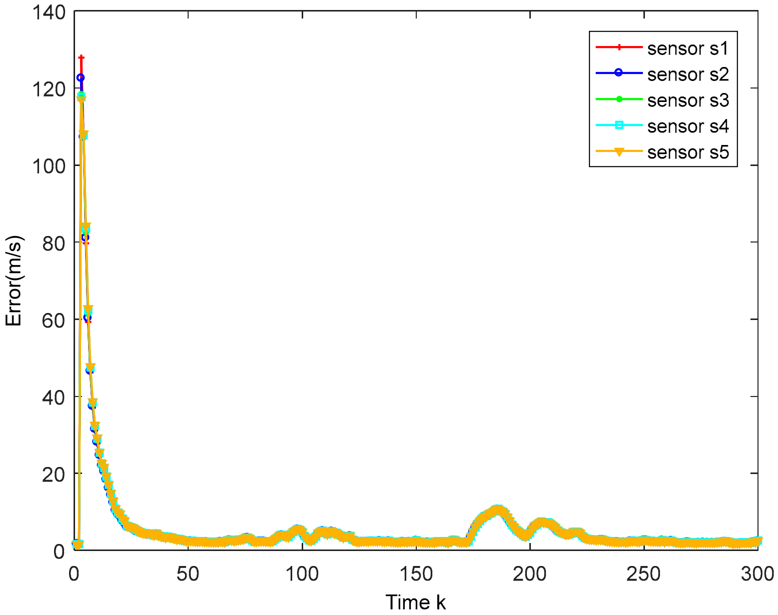

Figure 5. Velocity RMSE of different sensors.

5.2. Comparison Experiment

To validate the superiority of the proposed algorithm, the unscented transformation-based interacting multiple adaptive model filter (IMAM), interacting multiple adaptive model based on distributed unscented information filter (IMAM-DUIF) [5,32] and interacting multiple model-unscented information consensus filter (IMM-UICF) are chosen to be compared. In the IMAM method, we choose the best estimate throughout the whole sensor network to compare. IMAM-DUIF updates the state with neighboring information and there is no consensus protocol among them. In addition, IMM-UICF is the proposed method without adaptation.

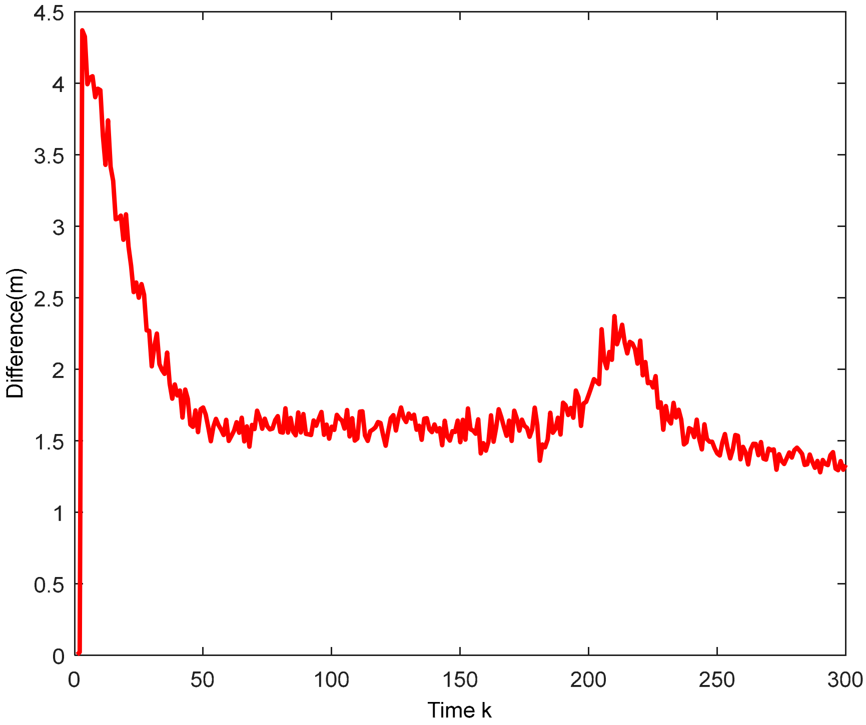

Figure 6. The disagreement between sensors.

In this scenario, the target moves with a complex maneuvering motion, including two strong maneuvers, which is a great challenge for sensors in the network. From 1 s to 110 s, the target dives and climbs, which are typical actions of military aircraft. From 111 s to 180 s, the target follows a snake maneuver. The details of the two maneuvers are shown in Tables 1 and 2. Finally, from 181 s to 250 s, target moves at constant velocity. The initial truth state of the target is $x(0) = [12000, 8000, -200, 0, 0, 0]^{\mathrm{T}}$. To obtain more measurement from target, the sampling interval is set to $T = 0.5$s.

Table 1. Diving and climbing.

Time (s)	**1–51**	**51–61**	**61–81**	**81—85**	**85–87**	**87–91**	**91–96**	**96–110**
Model	**CV**	**CT**	**CV**	**CT**	**CA**	**CA**	**CA**	**CT**
a_x a_y	0 0	$\omega = 0.12$	0 0	$\omega = -0.4$	0 20	0 15	-5 10	$\omega = 0.06$

Where $\omega = 0.12$ is the angular velocity.

Table 2. Snake maneuver.

Time(s)	**110–126**	**126–134**	**134–145**	**145–157**	**157–161**	**161–163**	**163–178**	**178–180**
Model	**CV**	**CA**	**CA**	**CA**	**CA**	**CA**	**CA**	**CV**
a_x	0	5	-8	10	0	-10	-5	0
a_y	0	-10	18	-20	30	-8	0	0

The constant velocity (CV) model and constant acceleration (CA) model have been described in Equations (29) and (30). The coordinate turn (CT) model is

$$x(k+1) = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega T)}{\omega} & \frac{\cos(\omega T)-1}{\omega} & 0 \\ 0 & 1 & \frac{1-\cos(\omega T)}{\omega} & \frac{\sin(\omega T)}{\omega} & 0 \\ 0 & 0 & \cos(\omega T) & -\sin(\omega T) & 0 \\ 0 & 0 & \sin(\omega T) & \cos(\omega T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0.5T^2\,0\,0 \\ 0\,0.5T^2\,0 \\ T \quad 0 \quad 0 \\ 0 \quad T \quad 0 \\ 0 \quad 0 \quad 1 \end{bmatrix} w_{CT}(k) \tag{36}$$

The sensor network in verification experiment is also used to estimate the state of target in this experiment. The sensor parameters, communication channels and related assumptions are all the same. IMAM-DUIF, IMM-UICF and IMAM-UICF are implemented by the sensor network, while IMAM is implemented by the sensor 1. Sensor 1 has the best detection capabilities over the whole network, so the comparison can better illustrate that the track performance of sensor network is superior to single sensor.

Model Parameters

In IMAM, two CS models are chosen to estimate state of target and interact. The model parameters of the five sensors are also the same.

In model 1, the initial state $x^1(0\big|0) = [11600, 7900, -180, 0, 0, 0]^{\mathrm{T}}$ and covariance $P^1(0|0) = diag([0.1, 0.1, 0.1, 0.1, 0.1, 0.1])$. The maneuver frequency is $\alpha = 0.1$. The state transition matrix F^1 is calculated as Equation (34). In addition, the process noise variance matrix Q^1 is set to

$$Q^1 = \begin{bmatrix} 0.5T^2\,0 \\ 0\,0.5T^2 \\ T \quad 0 \\ 0 \quad T \\ 1 \quad 0 \\ 0 \quad 1 \end{bmatrix} [0.25\,0.25] \begin{bmatrix} 0.5T^2\,0 \\ 0\,0.5T^2 \\ T \quad 0 \\ 0 \quad T \\ 1 \quad 0 \\ 0 \quad 1 \end{bmatrix}^{\mathrm{T}} \tag{37}$$

In model 2, the initial state $x^2(0\big|0) = [11600, 7900, -180, 0, 0, 0]^{\mathrm{T}}$ and covariance $P^2(0|0) = diag([0.1, 0.1, 0.1, 0.1, 0.1, 0.1])$. The maneuver frequency is $\alpha = 0.0001$. The state transition matrix F^2 is calculated as Equation (34). In addition, the process noise variance matrix Q^2 is equal to (37).

The initial model probability is chosen to be $u_1(0) = 0.5$ and $u_2(0) = 0.5$. In addition, the transition probability is set to

$$P = \begin{bmatrix} 0.90 & 0.10 \\ 0.10 & 0.90 \end{bmatrix} \tag{38}$$

As for IMM, a CV model and a CT model are chosen.

In the CV model, the initial state $x^1(0\big|0) = [11600, 7900, -180, 0]^{\mathrm{T}}$ and covariance $P^1(0|0) = diag([0.1, 0.1, 0.1, 0.1])$. The state transition matrix F^1 can be seen in Equation (29), and the process noise variance matrix Q^1 is set to

$$Q^1 = \begin{bmatrix} 0.5T^2\,0 \\ 0\,0.5T^2 \\ T \quad 0 \\ 0 \quad T \end{bmatrix} [0.25\,0.25] \begin{bmatrix} 0.5T^2\,0 \\ 0\,0.5T^2 \\ T \quad 0 \\ 0 \quad T \end{bmatrix}^{\mathrm{T}} \tag{39}$$

In CT model, the initial state $x^2(0\big|0) = [11600, 7900, -180, 0, 0]^{\mathrm{T}}$ and covariance $P^2(0|0) = diag([0.1, 0.1, 0.1, 0.1, 0.1])$. The state transition matrix F^2 can be seen in Equation (36), and the process noise variance matrix Q^2 is set to

$$Q^2 = \begin{bmatrix} 0.5T^2\,0\,0 \\ 0\,0.5T^2\,0 \\ T \quad 0 \quad 0 \\ 0 \quad T \quad 0 \\ 0 \quad 0 \quad 1 \end{bmatrix} [0.25\,0.25] \begin{bmatrix} 0.5T^2\,0\,0 \\ 0\,0.5T^2\,0 \\ T \quad 0 \quad 0 \\ 0 \quad T \quad 0 \\ 0 \quad 0 \quad 1 \end{bmatrix}^{\mathrm{T}} \tag{40}$$

Results and Analysis

Considering that there are five sensors tracking the target, we only display the results of sensor 1 as a reference. In addition, sensor 1 is the chosen one in IMAM. The simulation results are shown in Figures 7–10. Figure 7 shows that four methods are all effective in estimating the maneuvering target state. Figures 8 and 9 are the root mean square position and velocity error of IMAM, IMAM-DUIF, IMM-UICF and IMAM-UICF, respectively. Even if the target has several strong maneuvers, they can get a relatively accurate estimate in a short time. The two figures demonstrate that all four methods can track the maneuvering target, but the best performance of the single sensor is poorer than that of the remaining three methods. Compared to IMAM-DUIF and IMM-UICF, IMAM-UICF is more accurate, especially for position estimation. For networked algorithms, it is essential to compare the disagreement of estimates among sensors throughout the sensor network, and the relatively high disagreement in the estimates goes against the purpose of tracking targets in a distributed way. Figure 10 shows that IMAM-UICF and IMM-UICF both have relatively low disagreements, while the estimates of sensors in IMAM-UICF are closer to each other than IMM-UICF. In addition, the two methods both perform better than IMAM-DUIF in terms of agreement.

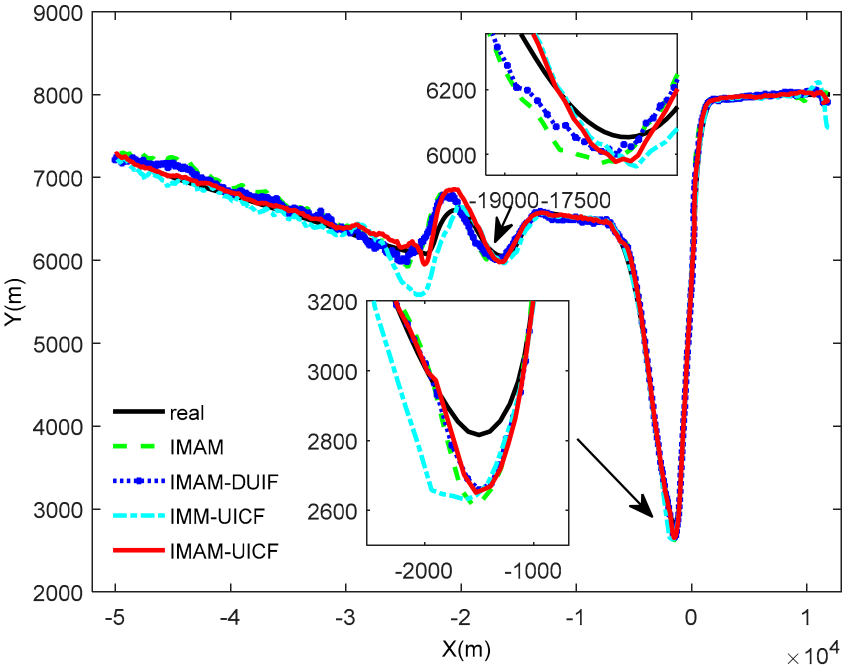

Figure 7. The trajectories of real and the four algorithms.

To further validate the performance of IMAM-UICF, the accumulative root mean square error (ARMSE) of position estimates is defined in Equation (41).

$$ARMSE_{\mathrm{pos}} = \sqrt{\frac{1}{\mathrm{MT}}\sum_{i=1}^{\mathrm{M}}\sum_{k=1}^{\mathrm{T}}\left((\hat{x}_i(k) - x_i(k))^2 + (\hat{y}_i(k) - y_i(k))^2\right)} \tag{41}$$

where M is the number of Monte Carlo runs, and T is the total number of simulation steps.$(\hat{x}_i(k), \hat{y}_i(k))$ represents the estimated position in i^{th} simulation, and $(x_i(k), y_i(k))$ represents the actual state. For simplicity, the definition of velocity ARMSE is not given here, which is similar to position ARMSE.

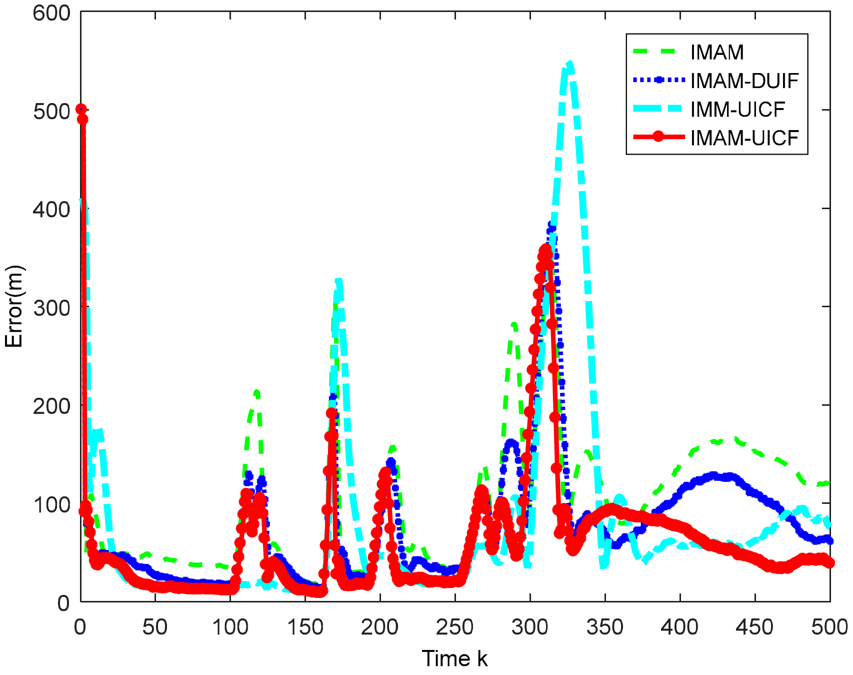

Figure 8. Position RMSE of different sensors.

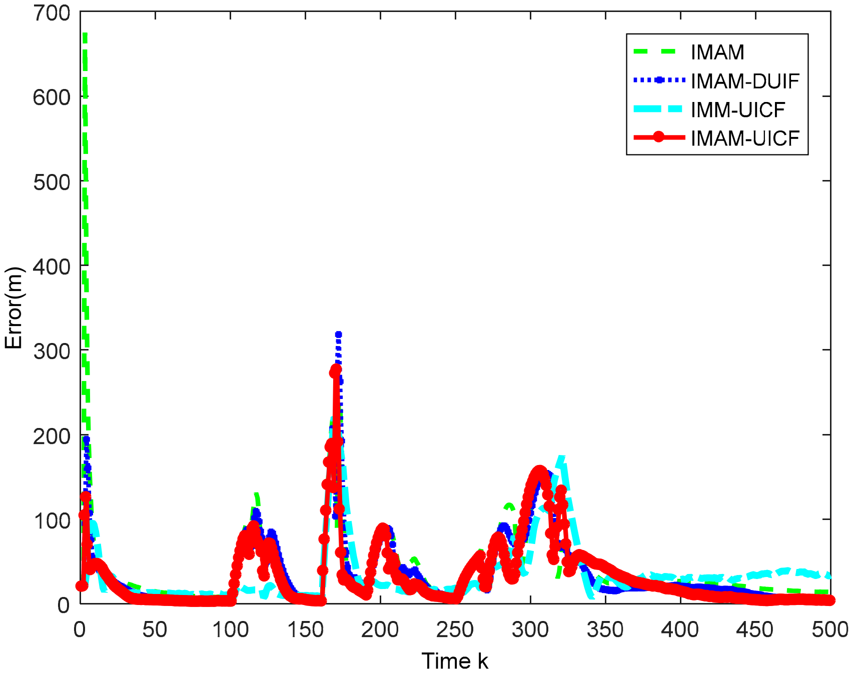

Figure 9. Velocity RMSE of different sensors.

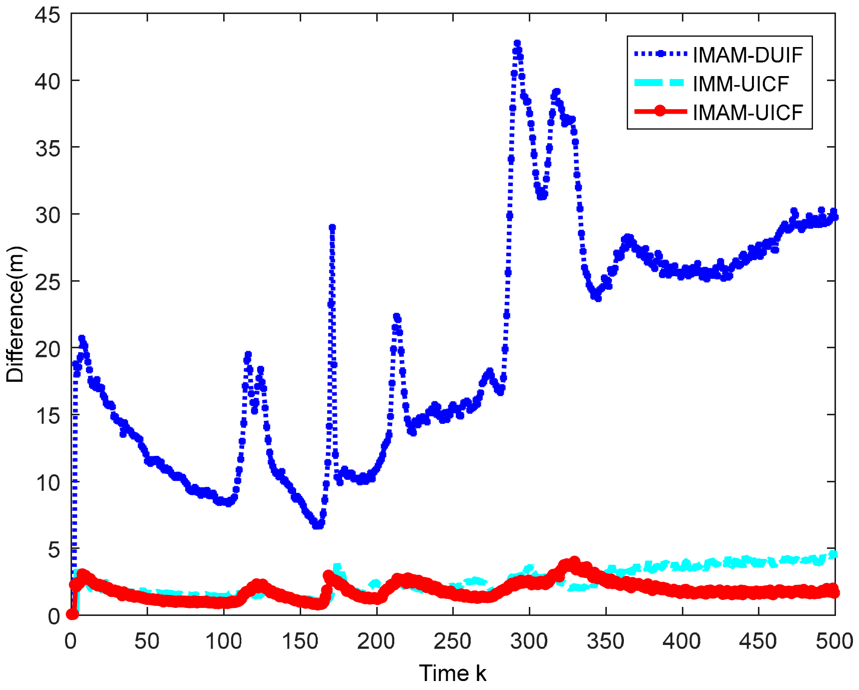

Figure 10. The disagreements of the three algorithms.

Table 3 shows that the two ARMSEs of IMM, IMAM-DUIF and IMM-UICF are larger than those of IMAM-UICF. In addition, the ARMSEs of IMAM-DUIF are smaller than those of IMAM, which shows the superiority of the sensor network. In particular, the position ARMSE of IMM-UICF is larger than that for the other methods. From the above comparisons, we can conclude that IMAM-UICF is superior among the three algorithms for maneuvering target tracking with respect to tracking accuracy and agreement among sensors. The time in Table 3 represents the computational cost of four methods. They are the calculation time of a single sensor at a sampling time. IMAM takes the shortest time, because the sensors do not need to communicate with neighbor sensors. Due to the iterative operations of consensus protocol, IMAM-UICF takes more time than IMAM-DUIF. In addition, the cost of IMAM-UICF is higher than IMM-UICF because of model adaptation.

Table 3. Performance comparison of different algorithms.

Algorithms	Position ARMSE (m)	Velocity ARMSE (m/s)	Time (s)
IMAM	124.0424	65.6603	0.001400000
IMAM-DUIF	103.8004	55.8458	0.001557773
IMM-UICF	137.0088	51.4543	0.001606565
IMAM-UICF	86.2178	50.7479	0.001806565

5.3. Experiment with Varying Numbers of Sensors n

This experiment aims to verify the advantages of the consensus protocol. In this experiment, n is varied from 5 to 20 at increments of 3. The results are shown in Figures 11 and 12. In the figures, when n is increased, the error of IMAM-UICF decreases, while the error of IMAM-DUIF shows almost no change. In the distributed architecture, each sensor communicates with its neighbor nodes. So each sensor can only get part of the information in the network, obtaining a local estimate. In IMAM-UICF, using consensus iteration, each node can indirectly fuse information from other nodes (those are not its neighbors). Thus, as the number of sensors is increased, the estimation results of IMAM-UICF are closer to the optimal global estimate.

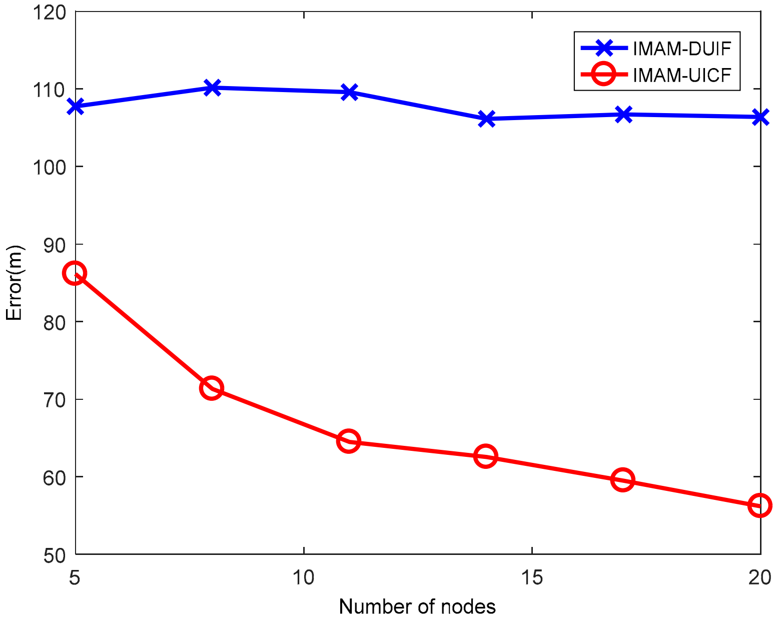

Figure 11. The position ARMSE of the two algorithms with varying number of sensors.

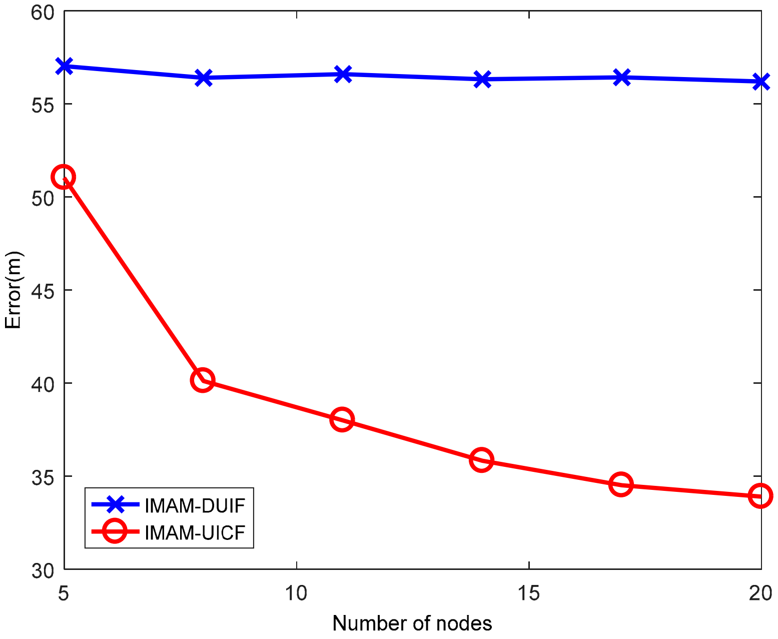

Figure 12. The velocity ARMSE of the two algorithms with varying number of sensors.

5.4. Experiment of Varying Measurement Noise r_i

This experiment aims to compare the performance of the three algorithms for varying measurement noise r_i. In this experiment, the standard deviation of r_i is varied from $[20, (0.2\pi/180)]^{\mathrm{T}}$ to $[68, (0.68\pi/180)]^{\mathrm{T}}$. The results are shown in Figures 13 and 14. In the figures, when the standard deviation of noise is small, the position ARMSE and velocity ARMSE of IMAM-UICF are the lowest among the 3 algorithms. With increased standard deviation of noise, the estimation accuracies of three methods decreases. When the standard deviation of noise is large, IMAM-DUIF has a higher position error than IMAM, and IMAM-UICF still has a better estimate in position and velocity. From the results, it can be seen that IMAM-UICF outperforms the other two methods for different measurements.

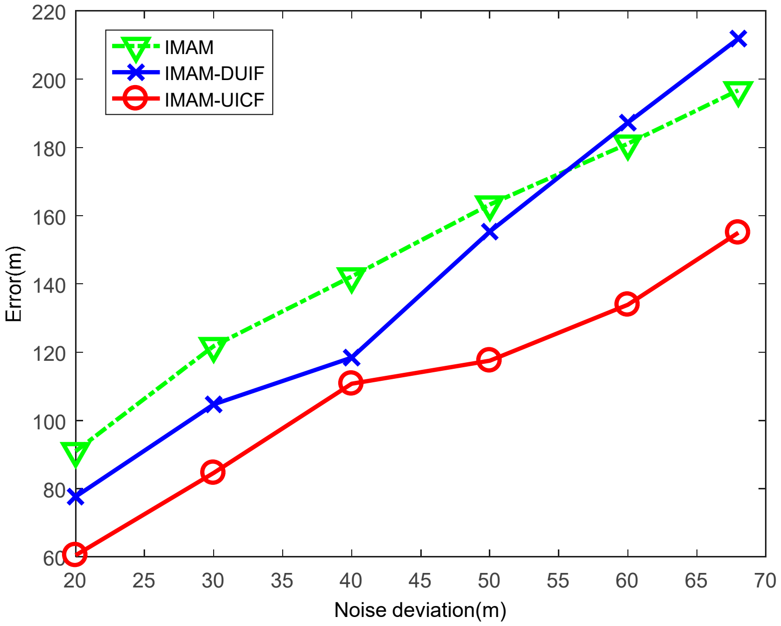

Figure 13. The position ARMSE of the three algorithms with varying measurement noise.

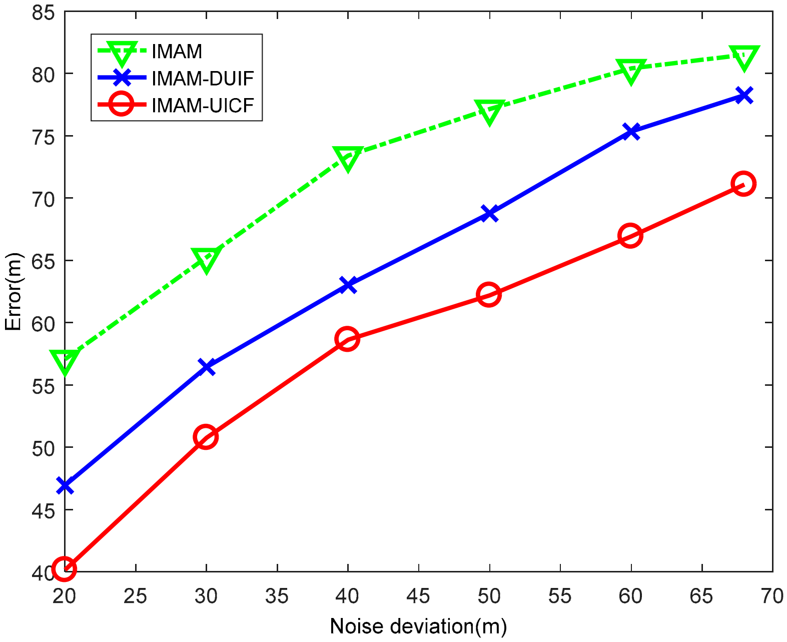

Figure 14. The velocity ARMSE of the three algorithms with varying measurement noise.

6. Conclusions

In this paper, an adaptive information-weighted consensus filter for maneuvering target tracking problem is proposed. Combined with adaptive current statistical models to handle the dynamic uncertainty, the true acceleration of a maneuvering target is more accurately estimated. The nonlinearity problem is solved using unscented transformation, which avoids the linearization errors of Taylor series expansions in the extended Kalman filter, and reduces the computational cost. The information-weighted consensus protocol and adaptive interacting multiple models are applied to improve the estimation accuracy and unify the estimates of maneuvering targets in the entire sensor network, which makes a shared situation picture in each sensor node. The simulation results illustrate that the proposed IMAM-UICF outperforms IMAM, IMAM-DUIF and IMM-UICF with respect to estimation accuracy and agreement of estimates in the entire sensor network.

Author Contributions: Conceptualization, Y.L. and Y.H.; Data curation, K.Y.; Formal analysis, Z.D.; Funding acquisition, Y.H.; Investigation, Y.Y.; Methodology, Z.D.; Project administration, Y.L.; Resources, J.L. and P.J.; Software, Y.H.; Validation, Z.D.; Visualization, J.L.; Writing – original draft, Z.D.; Writing – review & editing, J.L.

Funding: This study is co-supported by the National Natural Science Foundation of China. (Nos. 61471383, 91538201, 61531020, 61790550, 61671463 and 61790552).

Acknowledgments: The authors give their sincere thanks to the editors and the anonymous reviewers for their constructive comments of the manuscript.

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. Liu, H.; Wu, W. Interacting Multiple Model (IMM) Fifth-Degree Spherical Simplex-radial cubature Kalman filter for maneuvering target tracking. *Sensors* **2017**, *17*, 1374. [CrossRef] [PubMed]
4. Li, X.; Bar-Shalom, Y. Performance Prediction of the Interacting Multiple Model Algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **1993**, *29*, 755–771. [CrossRef]
5. Yang, Y.; Fan, X. Interacting multiple model algorithm based on adaptive current statistical model. *Syst. Eng. Electron.* **2016**, *38*, 977–983.
6. Vikranth, S.; Sudheesh, P.; Jayakumar, M. Nonlinear Tracking of Target Submarine Using Extended Kalman Filter (EKF). In *International Symposium on Security in Computing and Communication*; Springer: Singapore, 2016; Volume 625.
7. Yousefi, S.; Chang, X.W.; Champagne, B. Mobile Localization in Non-Line-of-Sight Using Constrained Square-Root Unscented Kalman Filter. *IEEE Trans. Veh. Technol.* **2015**, *64*, 2071–2083. [CrossRef]
8. Lamberti, R.; Petetin, Y.; Desbouvries, F.; Septier, F. Semi-independent resampling for particle filtering. *IEEE Signal Process. Lett.* **2018**, *25*, 130–134. [CrossRef]
9. Peng, X.; Yang, K.; Liu, C. Maneuvering Target Tracking Using Current Statistical Model Based Adaptive UKF for Wireless Sensor Network. *J. Commun.* **2015**, *10*, 579–588. [CrossRef]
10. Zhou, H.; Zhao, H.; Huang, H.; Zhao, X. A Cubature-Principle-Assisted IMM-Adaptive UKF Algorithm for Maneuvering Target Tracking Caused by Sensor Faults. *Appl. Sci.* **2018**, *10*, 1003. [CrossRef]
11. Martino, L.; Read, J.; Elvira, V.; Louzada, F. Cooperative Parallel Particle Filters for on-Line Model Selection and Applications to Urban Mobility. *Digit. Signal Process.* **2017**, *60*, 172–185. [CrossRef]
12. Urteaga, I.; Bugallo, M.F.; Djuric, P.M. Sequential Monte Carlo methods under model uncertainty. In Proceedings of the IEEE Statistical Signal Processing Workshop (SSP), Palma de Mallorca, Spain, 26–29 June 2016.
13. Carvalho, C.M.; Johannes, M.S.; Lopes, H.F.; Polson, N.G. Particle Learning and Smoothing. *Stat. Sci.* **2010**, *25*, 88–106. [CrossRef]
14. Lee, D.J. Unscented information filtering for distributed estimation and multiple sensor fusion. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI, USA, 18–21 August 2008; pp. 1–15.
15. Liu, Y.; Liu, J.; Li, G.; Qi, L.; Li, Y.; He, Y. Centralized multi-sensor square root cubature joint probabilistic data association. *Sensors* **2017**, *17*, 1–16. [CrossRef] [PubMed]
16. Yu, J.; Li, Y.; Cao, Y.; Xu, Y. An impedance-match design scheme for inductively active power filter in distribution networks. *Int. J. Electr. Power Energy Syst.* **2018**, *99*, 638–649. [CrossRef]
17. Das, S.; Moura, J.M. Distributed Kalman Filtering With Dynamic Observations Consensus. *IEEE Trans. Signal Process.* **2015**, *63*, 4458–4473. [CrossRef]
18. Yu, L.; You, H.; Haipeng, W. Squared-root Cubature Information Consensus Filter for non-linear Decentralized State Estimation in Sensor networks. *IET Radar Sonar Navig.* **2014**, *8*, 931–938. [CrossRef]

19. Keshavarz-Mohammadiyan, A.; Khaloozadeh, H. Consensus-based distributed unscented target tracking in wireless sensor networks with state-dependent noise. *Signal Process.* **2018**, *144*, 283–295. [CrossRef]
20. Katragadda, S.; Sanmiguel, J.C.; Cavallaro, A. Consensus protocols for distributed tracking in wireless camera networks. In Proceedings of the 2014 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–8.
21. Keshavarz-Mohammadiyan, A.; Khaloozadeh, H. Logarithm based Adaptive Particle Filter for Maneuvering Target Tracking in Wireless Sensor Networks with Multiplicative Noise. In Proceedings of the 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA), Qazvin, Iran, 27–28 January 2016; pp. 1–5.
22. Blair, W.D.; Ogle, T.L. Tracking Maneuvering Targets with Multiple Biased Sensors. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016.
23. Yu, Y. Consensus-Based Distributed Mixture Kalman Filter for Maneuvering Target Tracking in Wireless Sensor Networks. *IEEE Trans. Veh. Technol.* **2016**, *10*, 8669–8681. [CrossRef]
24. Bar-shalom, Y. *Multitarget-Multisensor Tracking: Principles and Techniques*; YBS Publishing: Storrs, CT, USA, 1995.
25. Chen, L.; Gong, X.; Shi, H.J. Maneuvering frequency adaptive algorithm of maneuvering target tracking. In Proceedings of the 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP), Beijing, China, 9–11 June 2013; pp. 445–458.
26. Yoon, J. Study on Nonlinear Filter Using Unscented Transformation Update. *J. Aerosp. Syst. Eng.* **2016**, *10*, 15–20. [CrossRef]
27. Sibley, G.; Sukhatme, G.S.; Matthies, L. The Iterated Sigma Point Filter with Applications to Long Range Stereo. In Proceedings of the 2nd Robotics: Science and Systems Conference, Philadelphia, Pennsylvania, 16–19 August 2006.
28. Kamal, A.; Farrell, J.; Roy-Chowdhury, A. Information weighted consensus filters and their application in distributed camera networks. *IEEE Trans. Autom. Control* **2013**, *58*, 3112–3125. [CrossRef]
29. Liu, G.; Tian, G. Square-Root Sigma-Point Information Consensus Filters for Distributed Nonlinear Estimation. *Sensors* **2017**, *17*, 800. [CrossRef] [PubMed]
30. Liggins, M.E.; Chong, C.Y.; Kadar, I.; Alford, M.G.; Vannicola, V.; Thomopoulos, S. Distributed fusion architectures and algorithms for target tracking. *Proc. IEEE* **1997**, *85*, 95–107. [CrossRef]
31. Bakr, M.A.; Lee, S. Distributed Multisensor Data Fusion under Unknown Correlation and Data Inconsistency. *Sensors* **2017**, *11*, 2472. [CrossRef] [PubMed]
32. Lin, W.; Nan, W.; Huayong, Z.; Lincheng, S. Distributed fusion estimation algorithm for multi-UAVs cooperative sensing. *Control Decis.* **2010**, *25*, 814–820.