

Article

A High-Order Kalman Filter Method for Fusion Estimation of Motion Trajectories of Multi-Robot Formation

Miao Wang ¹, Weifeng Liu ¹ and Chenglin Wen ^{2,*}

¹ School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China; 200611024@sust.edu.cn (M.W.); liuwf@sust.edu.cn (W.L.)

² School of Automation, Guangdong University of Petrochemical Technology, Maoming 525000, China

* Correspondence: wencil@gdupt.edu.cn

Abstract: Multi-robot motion and observation generally have nonlinear characteristics; in response to the problem that the existing extended Kalman filter (EKF) algorithm used in robot position estimation only considers first-order expansion and ignores the higher-order information, this paper proposes a multi-robot formation trajectory based on the high-order Kalman filter method. The joint estimation method uses Taylor expansion of the state equation and observation equation and introduces remainder variables on this basis, which effectively improves the estimation accuracy. In addition, the truncation error and rounding error of the filtering algorithm before and after the introduction of remainder variables, respectively, are compared. Our analysis shows that the rounding error is much smaller than the truncation error, and the nonlinear estimation performance is greatly improved.

Keywords: multi-robot formation; trajectory estimation; higher-order Kalman filter; fusion estimation



Citation: Wang, M.; Liu, W.; Wen, C. A High-Order Kalman Filter Method for Fusion Estimation of Motion Trajectories of Multi-Robot Formation. *Sensors* **2022**, *22*, 5590. <https://doi.org/10.3390/s22155590>

Academic Editor: Biswanath Samanta

Received: 21 June 2022

Accepted: 22 July 2022

Published: 26 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As a typical complex system, swarm dynamics systems have been widely studied since first being proposed. A swarm is composed of many simple intelligent individuals. The interaction between individuals based on basic mechanical rules can stimulate highly coordinated swarm behavior [1–4]. The dynamic behavior of swarming in nature shows amazing charm. The system composed of interconnected and constantly moving individuals emerges as colorful and highly coordinated swarming behavior, which provides a rich source of ideas for the cognition and optimization of industrial and social groups. It has considerable prospects in the fields of multi-agent collaboration [5], UAV formation [6], multi-robot collaboration [7], and intelligent grids [8].

In a multi-robot formation system, because the robot can detect the external environment it is a control system with pattern recognition, complex task execution and allocation, autonomous behavior decision-making, and other functions necessary to operate in a hostile environment and complete a variety of complex tasks. Therefore, a multi-robot formation system exhibits the characteristics of high consistency and collaboration with the outside world. In this relatively complex control system, it is necessary to obtain its precise state quantity first in order for control to be applied and to achieve the purpose of completing complex tasks. Therefore, it is necessary to perform more accurate state estimation for multi-robot motion.

Among the state estimation methods, in 1960 R. E. Kalman et al. [9,10] proposed a recursive state estimation method suitable for a system in which the model is linear and the noise obeys a Gaussian distribution. Kalman filtering (KF) is a time domain filtering method; when the system process noise and measurement noise are mutually independent Gaussian white noise with zero mean, a Kalman filter is the optimal unbiased estimator in the sense of minimum variance [11].

In general, the motion equation of a multi-robot formation and the measurement equation of its sensor are both nonlinear equations. The motion equation has weak nonlinear characteristics because only simple operations such as turning and going straight are available. The sensor observes the robot target, which shows strong nonlinear characteristics. For nonlinear state estimation problems, the traditional filtering methods are mainly extended Kalman filter (EKF), unscented Kalman filter (UKF) and cubature Kalman filter (CKF). EKF uses a Taylor series first-order approximation expansion of the nonlinear equations, which introduces linearization errors and has low estimation accuracy for estimating systems with strong nonlinearity [12]. Over time, Julier et al. [13] proposed another nonlinear filtering method, the unscented Kalman filter. The core of the UKF algorithm is the UT transformation, which uses an ensemble of sigma sampling points to approximate the nonlinear system model [14–16]. The UKF algorithm is more computationally intensive than the EKF algorithm, and its filtering accuracy is better, at least to the second-order accuracy of the Taylor expansion. However, the shortcomings of the UKF algorithm are obvious in that it can only be applied to Gaussian systems, and a problem with the non-positive definite error covariance matrix is caused by the emergence of negative weight in the implementation of the algorithm [17]. The frequency of this problem becomes greater as the order of the system increases and the nonlinearity grows stronger. In order to better solve the problem of UKF being non-positive definite when dealing with high-dimensional covariance matrices, resulting in divergent filtering results, Ienkaram proposed a weight-selective UKF filtering method called Cubature Kalman Filter (CKF) [18,19]. CKF optimizes the sigma point sampling method and weight distribution in UKF using a spherical integral and radial integral. Compared with traditional EKF, it has improved estimation accuracy and improved stability of filtering methods [20].

In recent years, in the treatment of nonlinear estimation problems, the structure of the extended Kalman filter (EKF) is used to design recursive estimators as a new idea to solve the problem. The main idea is to transform the linear error of the nonlinear equation after doing the first-order Taylor expansion into the form of a product of the scaling matrix associated with the problem solved and some time-varying positive definite matrix, then set the upper bound of the covariance matrix [21–23], and get the upper bound of the minimum covariance matrix by numerically solving the Ricatti equation, and finally use recursive filtering, which cannot be accurately resolved to get this scaling matrix, and then the process of solving the inverse matrix in the covariance matrix obtained by the Ricatti equation is more complicated, which greatly increases the computational burden in some high-dimensional occasions and cannot meet the real-time requirements [24]. The method cannot accurately analyze the scaling matrix, and the process of solving the inverse matrix in the covariance matrix obtained by the Ricatti equation is relatively complicated. The other is to borrow the Taylor expansion idea, expand the higher-order term with the original equation, design the new state and measurement equation, and use recursive filtering to get the required state estimate [25,26]. It can choose the expansion order reasonably according to the nonlinear strength and weakness of the actual problem, which is a good balance between the requirements of arithmetic power and estimation accuracy, and has a good real-time filtering feature [27,28]. At the same time, this method has also been extended to correlation entropy filtering [29], Kronecker product [30] and lithium battery life prediction [31], which all show good estimation performance.

In traditional state estimation for multi-robot operation, the nonlinear state estimation is performed separately for each robot using methods such as EKF, UKF, or CKF. In addition to the problems already mentioned in this paper, performing individual state estimation for each robot reduces the estimation accuracy [32–34]. In recent years, the use of trigger probability in the multi-robot formation problem as the connection relationship between robots has gradually become a mainstream idea [24]. The reason for real-time estimation is that at a certain moment the connection relationship between multiple robots is deterministic, and there is only a connection or no connection, which is equivalent to a binary connection.

On this basis, this paper addresses the multi-robot formation state estimation problem by proposing a centerless multi-robot state joint estimation method based on high-order extended Kalman filter (HEKF). This improves the original multi-robot connection relationship based on trigger probability and improves the multi-robot state estimation accuracy. The main algorithm of this paper is as follows. In view of the nonlinear characteristics of multi-robot motion and observation, Taylor's higher-order expansion of the robot's state equation and measurement equation is performed to solve the problem of different nonlinear degrees of the two nonlinear equations. The high-order Kalman filtering algorithm is improved by expanding the truncation error with the state for dimensional estimation and fusing the state information of neighboring node robots to improve the multi-robot state estimation accuracy. A numerical simulation experiment proves the effectiveness of the algorithm. Therefore, we choose the second method based on high-order Kalman filtering. The main work of this paper includes:

- After Taylor expansion of the nonlinear state equation and observation equation, the remainder variable is introduced and the original EKF algorithm and the high-order Kalman filtering algorithm are changed to discard the truncation error. Only the first-order and low-order items are retained, reducing the estimation error and improving the estimation accuracy.
- A dynamic model of the remainder variables is introduced into the state equation and observation equation as hidden variables, and the changed pseudo-linear state equation and observation equation are rewritten into higher-order linear forms through dimension expansion.
- The new high-order linear equation is used to obtain the state estimation value using the recursive filtering algorithm, and the effectiveness of the algorithm is analyzed.

The rest of this paper is organized as follows. Section 2 proposes state equations and observation equations for the multi-robot formation operating system and designs a recursive estimator based on the EKF structure, while Section 3 presents the recursive estimator form derived from the second section, introduces the remainder variables into the state equation and the observation equation, and presents a new high-order linear filter derived through dynamic modeling, then analyzes its performance. In Section 4, an indoor multi-robot simulation numerical experiment is used to verify the effectiveness of the algorithm proposed in this paper. Finally, conclusions are provided in Section 5. In addition, the important mathematical symbols in Appendix A.

2. Problem Description

The movement of the robot is generally carried out by operations such as going straight and turning, and the observation system is generally composed of radar or visual sensors. Therefore, both the robot's motion and the observation system have nonlinear characteristics. In a multi-robot formation environment, there is a certain degree of motion consistency between the robots, and thus a certain coupling relationship. The dynamic equations can be described by the following N node coupling equations:

$$x_i(k+1) = f(x_i(k)) + c \sum_{j=1}^N \Gamma x_j(k) + w_i(k) \quad (1)$$

$$y_i(k+1) = h(x_i(k+1)) + v_i(k+1) \quad (2)$$

where $x_i(k) \in R^n$ and $y_i(k+1) \in R^q$ denote the state vector and the measurement vector, respectively, i and k denote the node index and the time instant, respectively, $f(\cdot)$ and $h(\cdot)$ are known nonlinear functions that are assumed to be continuously differentiable, $c > 0$ is the overall coupling strength, and Γ is the inner-coupling matrix. The process noise, $w_i(k)$, and the measurement noise, $v_i(k+1)$, are assumed to be mutually uncorrelated zero-mean white Gaussian noise with the covariances $Q_i(k)$ and $R_i(k+1)$, respectively.

The EKF structure is used to design the recursive estimation of the multi-robot formation trajectories (1) and (2):

$$\hat{x}_i(k+1|k) = f(\hat{x}_i(k|k)) + c \sum_{j=1}^N \Gamma \hat{x}_j(k|k) \quad (3)$$

$$\hat{x}_i(k+1|k+1) = \hat{x}_i(k+1|k) + K_i(k+1)[y_i(k+1) - h(\hat{x}_i(k+1|k))] \quad (4)$$

where $\hat{x}_i(k+1|k)$ and $\hat{x}_i(k+1|k+1)$ denote the predicted and updated estimates at time instant $k+1$, respectively, and $K_{i,k+1}$ is the gain matrix to be determined

As in the EKF, we define the updated estimation error and the corresponding covariance as follows:

$$e_i(k+1|k+1) = x_i(k+1) - \hat{x}_i(k+1|k+1) \quad (5)$$

$$P_i(k+1|k+1) = E\{e_i(k+1|k+1)e_i^T(k+1|k+1)\} \quad (6)$$

3. Algorithm Description

In the second section, we use the EKF structure to design the recursive filter. This idea generally expands the nonlinear function to obtain the first-order terms and truncation errors. The traditional EKF algorithm directly discards the truncation errors and keeps only the first-order terms. In the transfer function with strong non-linearity a large amount of information will be lost, which makes the estimation result inaccurate. In recent years, in the study of nonlinear filtering problems, the following two ideas have mainly been adopted in the design of recursive filters using the EKF structure. The first is to transform the truncation error into the form of a product of the scaling matrix and a time-varying positive definite matrix related to the problem, then set the upper bound of the covariance matrix, obtain the upper bound of the minimum covariance matrix by numerically solving the Riccati equation, and finally use recursive filtering, which cannot accurately analyze the scaling matrix, before obtaining the covariance matrix through the Riccati equation. The inverse matrix process is more complicated, and in occasions with high dimensions the computational burden is greatly increased to the point that real-time requirements cannot be met. The second is to use the idea of Taylor expansion to expand the high-order term while using the original equation to expand the dimension, design new state and measurement equations, use recursive filtering to obtain the required state estimation value, and avoid first above method. The "dimension disaster" caused by inversion in the dimensional situation, as well as the expansion order, can be reasonably selected according to the nonlinear strong and weak characteristics in the actual problem, which takes into account the requirements of computing power and estimation accuracy and has a good real-time filtering feature. A block diagram of the algorithm is shown in Figure 1.

3.1. Taylor Expansion of Nonlinear System and Introduction of Remainder Variables

Among the nonlinear filtering algorithms, the extended Kalman filter (EKF) algorithm is widely used. Its main idea is perform a first-order Taylor expansion of the nonlinear state equation around the filter estimated value. The main process is as follows.

Let $f_i(k) = f(x_i(k)) + c \sum_{j=1}^N \Gamma x_j(k)$; then, $x_i(k+1)$ is subjected to a first-order Taylor expansion at the filter estimate $\hat{x}_i(k|k)$ to obtain the following expression for $x_i(k+1)$:

$$\begin{aligned} x_i(k+1) &= f_i(k) + w_i(k) \\ &= f_i(\hat{x}_i(k|k)) + \left. \frac{\partial f_i}{\partial x} \right|_{x_i(k)=\hat{x}_i(k|k)} (x_i(k) - \hat{x}_i(k|k)) + T(f_i(\hat{x}_i(k|k))) + w_i(k) \\ &= A_i(\hat{x}_i(k|k), k)x_i(k) + \beta_i(\hat{x}_i(k|k), \xi_i(k)) + w_i(k) \\ &= A_i(\hat{x}_i(k|k), k)x_i(k) + f_i(\hat{x}_i(k|k)) - A_i(\hat{x}_i(k|k), k)\hat{x}_i(k|k) + T(f(\hat{x}_i(k|k))) + w_i(k) \end{aligned} \quad (7)$$

where the state transition matrix is $A_i(\hat{x}_i(k|k), k) = \frac{\partial f_i}{\partial x} \Big|_{x_i(k)=\hat{x}_i(k|k)}$, and $\beta_i(\hat{x}_i(k|k), \xi_i(k)) = f_i(\hat{x}_i(k|k)) - A_i(\hat{x}_i(k|k))\hat{x}_i(k|k) + T(f_i(\hat{x}_i(k|k)))$ is the state remainder variables.

In the same way, the first-order Taylor expansion of the nonlinear observation function $h_i(x_i(k + 1))$ at the state prediction value $\hat{x}_i(k + 1|k)$ can be obtained:

$$\begin{aligned}
 y_i(k + 1) &= h(x_i(k + 1)) + v_i(k + 1) \\
 &= h(\hat{x}_i(k + 1|k)) + \frac{\partial h_i}{\partial x} \Big|_{x_i(k+1)=\hat{x}_i(k+1|k)} (x_i(k + 1) - \hat{x}_i(k + 1|k)) \\
 &\quad + T(h(\hat{x}_i(k + 1|k))) + v_i(k + 1) \\
 &= h(\hat{x}_i(k + 1|k)) + C_i(\hat{x}_i(k + 1|k), k + 1)(x_i(k + 1) - \hat{x}_i(k + 1|k)) \\
 &\quad + T(\hat{x}_i(k + 1|k)) + v_i(k + 1) \\
 &= C_i(\hat{x}_i(k + 1|k), k + 1)x_i(k + 1) + \gamma(\hat{x}_i(k + 1|k)) + v_i(k + 1)
 \end{aligned}
 \tag{8}$$

where the observation matrix is $C_i(\hat{x}_i(k + 1|k), k + 1) = \frac{\partial h_i}{\partial x} \Big|_{x_i(k+1)=\hat{x}_i(k+1|k)}$, and $\gamma_i(\hat{x}_i(k + 1|k)) = y_i(k + 1) - h(\hat{x}_i(k + 1|k)) - C_i(\hat{x}_i(k + 1|k), k + 1)\hat{x}_i(k + 1|k) + T(h(\hat{x}_i(k + 1|k)))$ is the observation remainder.

From this, we can see that in the original extended Kalman filtering algorithm, which uses only the first-order term information of the nonlinear function, the high-order term information and the truncation error are directly discarded, which makes the estimation accuracy low in occasions with a high degree of nonlinearity and leads to inaccurate estimation results. In certain extreme cases, the filtering results may even directly diverge. Here, we introduce remainder variables into the state equation and observation equation, respectively, to compensate for the loss of high-order information in the original filtering algorithm.

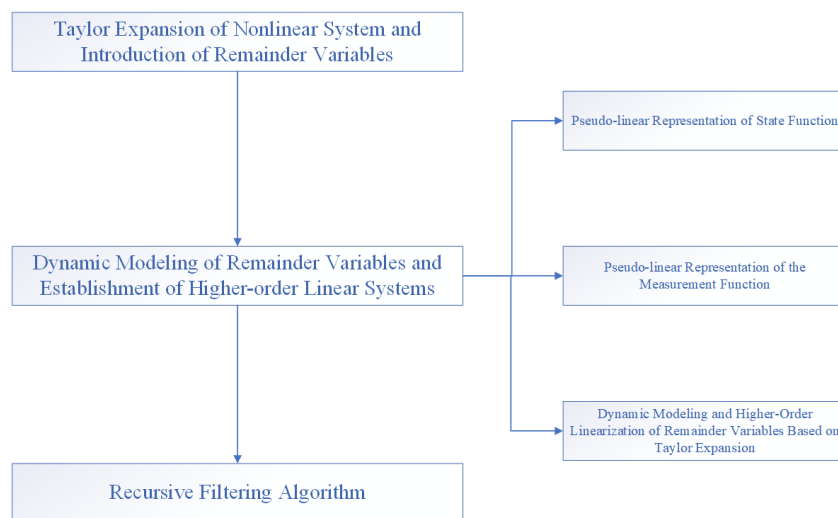


Figure 1. Algorithm block diagram.

3.2. Dynamic Modeling of Remainder Variables and Establishment of Higher-Order Linear Systems

The remainder variable is obtained by integrating the constant term with the truncation error after Taylor expansion of the nonlinear state function and the observation function around the filter estimation value and the one-step prediction value, respectively. By using the remainder variable information to improve the estimation accuracy, the original state equation and observation equation need to be expanded and rewritten to establish a high-order Kalman filter.

Lemma 1. The n -order Taylor formula of the binary function $z = f(x, y)$ at (x_0, y_0) :

Assuming that $z = f(x, y)$ is continuous in a neighborhood of (x_0, y_0) and has continuous partial derivatives up to order m ; if $(x_0 + h, y_0 + k)$ is any point in this neighborhood, there are

$$\begin{aligned}
 f(x_0 + h, y_0 + k) &= f(x_0, y_0) + (h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y}) f(x_0, y_0) + \frac{1}{2!} (h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y})^2 f(x_0, y_0) \\
 &+ \dots + \frac{1}{n!} (h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y})^n f(x_0, y_0) + \frac{1}{(n+1)!} (h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y})^{n+1} f(x_0 + \theta h, y_0 + \theta k) \quad (9) \\
 &(0 < \theta < 1)
 \end{aligned}$$

where

$$(h \frac{\partial}{\partial x} + k \frac{\partial}{\partial y})^m f(x_0, y_0) = \sum_{p=0}^m C_m^p h^p k^{m-p} \frac{\partial^m f(x_0, y_0)}{\partial x^p \partial y^{m-p}} \quad (10)$$

Lemma 2. Taylor expansion of a multivariate function at x_k :

$$\begin{aligned}
 f(x^1, x^2, \dots, x^n) &= f(x_k^1, x_k^2, \dots, x_k^n) + \sum_{i=1}^n (x^i - x_k^i) f'_{x^i}(x_k^1, x_k^2, \dots, x_k^n) \\
 &+ \frac{1}{2!} \sum_{i=1}^n \sum_{j=1}^n (x^i - x_k^i)(x^j - x_k^j) f''_{ij}(x_k^1, x_k^2, \dots, x_k^n) + o(f) \quad (11)
 \end{aligned}$$

Assume that $X = [x^1, x^2, \dots, x^n]^T$

$$f(X) = f(X_k) + [\nabla f(X_k)]^T (X - X_k) + \frac{1}{2!} [X - X_k]^T H(X_k) [X - X_k] + T(f) \quad (12)$$

where

$$\nabla f(X_k) = \left[\frac{\partial f(X_k)}{\partial x^1}, \frac{\partial f(X_k)}{\partial x^2}, \dots, \frac{\partial f(X_k)}{\partial x^n} \right]^T \quad (13)$$

$$H(X_k) = \begin{bmatrix} \frac{\partial^2 f(X_k)}{\partial x^1 \partial x^1} & \frac{\partial^2 f(X_k)}{\partial x^1 \partial x^2} & \dots & \frac{\partial^2 f(X_k)}{\partial x^1 \partial x^n} \\ \frac{\partial^2 f(X_k)}{\partial x^2 \partial x^1} & \frac{\partial^2 f(X_k)}{\partial x^2 \partial x^2} & \dots & \frac{\partial^2 f(X_k)}{\partial x^2 \partial x^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(X_k)}{\partial x^n \partial x^1} & \frac{\partial^2 f(X_k)}{\partial x^n \partial x^2} & \dots & \frac{\partial^2 f(X_k)}{\partial x^n \partial x^n} \end{bmatrix} \quad (14)$$

3.2.1. Pseudo-Linear Representation of State Function $f_i(x_{i,k})$

According to the above lemma, the state transition function $f_i(x_{i,k})$ in Formula (1) can be expressed as a polynomial form using Taylor expansion, as follows.

For simplicity, the following derivation takes two state variables, x_1 and x_2 :

$$\begin{aligned}
 f_i(x(k)) &= a_{i,0} + (a_{i,1,0} x_1(k) + a_{i,0,1} x_2(k)) \\
 &+ (a_{i,2,0} x_1^2(k) + a_{i,1,1} x_1(k) x_2(k) + a_{i,0,2} x_2^2(k)) \\
 &+ (a_{i,3,0} x_1^3(k) + a_{i,2,1} x_1^2(k) x_2(k) + a_{i,1,2} x_1(k) x_2^2(k) + a_{i,3,0} x_2^3(k)) \\
 &+ \dots + \sum_{\substack{l_1+l_2=l \\ l_1, l_2 < l}} a_{i,l_1,l_2} x_1^{l_1}(k) x_2^{l_2}(k) \\
 &+ \dots + \sum_{\substack{r_1+r_2=r \\ r_1, r_2 < r}} a_{i,r_1,r_2} x_1^{r_1}(k) x_2^{r_2}(k) \\
 &+ \beta(x_1(k)) + \beta(x_2(k)) \quad (15)
 \end{aligned}$$

where $a_{i,0}$ is a constant, $\sum_{\substack{l_1+l_2=l \\ l_1, l_2 < l}} a_{i,l_1,l_2} x_1^{l_1}(k) x_2^{l_2}(k)$, $0 < l_1, l_2 < l (l = 1, 2, \dots, r)$ are the weighted sum of all l order tensor terms, a_{i,l_1,l_2} , $l_1 + l_2 = l, 0 \leq l_1, l_2 \leq l (l = 1, 2, \dots, r)$ is the weight of the corresponding term, and $\beta(x_1(k)), \beta(x_2(k))$ are the respective remainder variables of the state.

Theorem 1. Let $x^l(k) = x_1^{l_1}(k) x_2^{l_2}(k)$, $l_1 + l_2 = l, 0 < l_j < l (l = 1, 2, \dots, r)$ be a set of l -order hidden variables corresponding to the original system variable $x(k)$.

Theorem 2. The weight vector corresponding to the l -order latent variable vector is then

$$a_i^{(l)} = [a_{i;1}^{(l)} \quad a_{i;2}^{(l)} \quad \dots \quad a_{i;n_l}^{(l)}] = [a_{i;l,0} \quad a_{i;l-1,1} \quad \dots \quad a_{i;0,l}] \quad (i = 1, 2) \tag{16}$$

We use the following two states as an example:

$$\begin{bmatrix} x_1^{(1)}(k+1) \\ x_2^{(1)}(k+1) \end{bmatrix} = \begin{bmatrix} a_1^{(1)} & a_1^{(2)} & \dots & a_1^{(l)} & \dots & a_1^{(r)} & a_1(\beta(x_1)) & a_1(\beta(x_2)) \\ a_2^{(1)} & a_2^{(2)} & \dots & a_2^{(l)} & \dots & a_2^{(r)} & a_2(\beta(x_1)) & a_2(\beta(x_2)) \end{bmatrix} \begin{bmatrix} x^{(1)}(k) \\ x^{(2)}(k) \\ \vdots \\ x^{(l)}(k) \\ \vdots \\ x^{(r)}(k) \\ \beta(x_1(k)) \\ \beta(x_2(k)) \end{bmatrix} + \begin{bmatrix} w_1^{(1)}(k) \\ w_2^{(1)}(k) \end{bmatrix} \tag{17}$$

Define $x(k) = x^{(1)}(k) = \begin{bmatrix} x_1^{(1)}(k) \\ x_2^{(1)}(k) \end{bmatrix}$, $A^{(l)} = \begin{bmatrix} a_1^{(l)} \\ a_2^{(l)} \end{bmatrix}$, $w(k) = \begin{bmatrix} w_1^{(1)}(k) \\ w_2^{(1)}(k) \end{bmatrix}$

Then,

$$\begin{aligned} x^{(1)}(k+1) &= \begin{bmatrix} A^{(1)} & A^{(2)} & \dots & A^{(l)} & \dots & A^{(r)} & A(\beta) \end{bmatrix} \begin{bmatrix} x^{(1)}(k) \\ x^{(2)}(k) \\ \vdots \\ x^{(l)}(k) \\ \vdots \\ x^{(r)}(k) \\ \beta(x(k)) \end{bmatrix} + w^{(1)}(k) \\ &= A^{(1)} x^{(1)}(k) + \sum_{l=2}^r A^{(l)} x^{(l)}(k) + A(\beta) \beta(x(k)) + w^{(1)}(k) \end{aligned} \tag{18}$$

Remark 1. Compared with the original state model, $x^{(1)}(k) = [x_1(k) \quad x_2(k)]^T$, the above equation is in a pseudo-linear form with higher-order latent variables $x^{(l)}(k) (l = 1, 2, \dots, r)$ and remainder variables $\beta(x(k))$. There is only a change in representation; as there is no essential difference, this is called pseudo-linearization.

3.2.2. Pseudolinear Representation of the Measurement Function

Similarly, the measurement function in formula (2) is expressed as a polynomial form by Taylor expansion, as follows:

$$\begin{aligned}
 h_i(x(k+1)) &= h_{i,0} + (h_{i,1,0}x_1(k+1) + h_{i,0,1}x_2(k+1)) \\
 &+ (h_{i,2,0}x_1^2(k+1) + h_{i,1,1}x_1(k+1)x_2(k+1) + h_{i,0,2}x_2^2(k+1)) \\
 &+ (h_{i,3,0}x_1^3(k+1) + h_{i,2,1}x_1^2(k+1)x_2(k+1) + h_{i,1,2}x_1(k+1)x_2^2(k+1) + h_{i,3,0}x_2^3(k+1)) \\
 &+ \dots + \sum_{\substack{l_1+l_2=l \\ l_1, l_2 < l}} h_{i,l_1,l_2}x_1^{l_1}(k+1)x_2^{l_2}(k+1) \\
 &+ \dots + \sum_{\substack{r_1+r_2=r \\ r_1, r_2 < r}} h_{i,r_1,r_2}x_1^{r_1}(k+1)x_2^{r_2}(k+1) \\
 &+ \gamma(x_1(k+1)) + \gamma(x_2(k+1))
 \end{aligned} \tag{19}$$

where $\gamma(x(k+1))$ denotes the remainder variables of the measurement function:

$$\begin{aligned}
 &\begin{bmatrix} y_1^{(1)}(k+1) \\ y_2^{(2)}(k+1) \end{bmatrix} \\
 &= \begin{bmatrix} h_1^{(1)} & h_1^{(2)} & \dots & h_1^{(l)} & \dots & h_1^{(r)} & h_1(\gamma(x_1)) & h_1(\gamma(x_2)) \\ h_2^{(1)} & h_2^{(2)} & \dots & h_2^{(l)} & \dots & h_2^{(r)} & h_2(\gamma(x_1)) & h_2(\gamma(x_2)) \end{bmatrix} \begin{bmatrix} x^{(1)}(k+1) \\ x^{(2)}(k+1) \\ \vdots \\ x^{(l)}(k+1) \\ \vdots \\ x^{(r)}(k+1) \\ \gamma(x_1(k+1)) \\ \gamma(x_2(k+1)) \end{bmatrix} + \begin{bmatrix} v_1^{(1)}(k+1) \\ v_2^{(1)}(k+1) \end{bmatrix} \\
 &= \begin{bmatrix} H^{(1)} & H^{(2)} & \dots & H^{(l)} & \dots & H^{(r)} & H(\gamma) \end{bmatrix} \begin{bmatrix} x^{(1)}(k+1) \\ x^{(2)}(k+1) \\ \vdots \\ x^{(l)}(k+1) \\ \vdots \\ x^{(r)}(k+1) \\ \gamma(x(k+1)) \end{bmatrix} + v^{(1)}(k+1) \\
 &= H^{(1)}x^{(1)}(k+1) + \sum_{l=2}^r H^{(l)}x^{(l)}(k+1) + H(\gamma)\gamma(x(k+1)) + v^{(1)}(k+1)
 \end{aligned} \tag{20}$$

3.2.3. Dynamic Modeling and Higher-Order Linearization of Remainder Variables Based on Taylor Expansion

In order to convert the pseudo-linearized model established above into a real high-order linearized form, it is necessary to expand the dimension and model using the high-order latent variables and remainder variables as new variables of the system.

In order to solve the problem in Remark 1, the high-order latent variable $x^{(l)}(k)$ and the remainder variables $\beta(x(k))$ and $\gamma(x(k+1))$ are used as time-varying parameters and the linear coupling relationship between the l -order latent variable $x^{(l)}(k+1)$ and the u -order latent variable $x^{(u)}(k)$ is established, along with the dynamic model relationship of the remainder variables $\beta(x(k))$ and $\gamma(x(k+1))$:

$$x^{(l)}(k+1) = A_l^{(u)}(k)x^{(u)}(k) \quad (l, u = 2, 3, \dots, r) \tag{21}$$

$$\beta(x(k+1)) = A(\beta)\beta(x(k)) \tag{22}$$

$$\gamma(x(k+1)) = A(\gamma)\gamma(x(k)) \tag{23}$$

Then, the latent variables and remainder variables are used as the extension of the original state vector to realize the linear representation of the state model. Here, $A_l^{(u)}(k)$

and $\beta(x(k+1))$ can be identified according to the input information of the original state model; when there is no prior information, the settings are as follows:

$$A_l^{(u)}(k) = \begin{cases} I, & l = u \\ 0, & l \neq u \end{cases} A(\beta) = I, A(\gamma) = I \quad (24)$$

Then,

$$\begin{bmatrix} x^{(1)}(k+1) \\ x^{(2)}(k+1) \\ \vdots \\ x^{(l)}(k+1) \\ \vdots \\ x^{(r)}(k+1) \\ \beta(x(k+1)) \\ \gamma(x(k+1)) \end{bmatrix} = \begin{bmatrix} A_u(k+1, k) & A(\beta) & 0 \\ 0 & A(\beta) & 0 \\ 0 & 0 & A(\gamma) \end{bmatrix} \begin{bmatrix} x^{(1)}(k) \\ x^{(2)}(k) \\ \vdots \\ x^{(l)}(k+1) \\ \vdots \\ x^{(r)}(k) \\ \beta(x(k)) \\ \gamma(x(k)) \end{bmatrix} + \begin{bmatrix} w^{(1)}(k) \\ w^{(2)}(k) \\ \vdots \\ w^{(l)}(k) \\ \vdots \\ w^{(r)}(k) \\ w_\beta(k) \\ w_\gamma(k) \end{bmatrix} \quad (25)$$

$$\text{Denote } X(k) = \left[(x^{(1)}(k))^T \quad (x^{(2)}(k))^T \quad \dots \quad (x^{(r)}(k))^T \quad (\beta(x(k)))^T \quad (\gamma(x(k)))^T \right]^T$$

$$A(k+1, k) = \begin{bmatrix} A_u(k+1, k) & A(\beta) & 0 \\ 0 & A(\beta) & 0 \\ 0 & 0 & A(\gamma) \end{bmatrix}, \quad (26)$$

$$A_u(k+1, k) = \begin{bmatrix} A_1^{(1)}(k+1, k) & A_1^{(2)}(k+1, k) & \dots & A_1^{(u)}(k+1, k) & \dots & A_1^{(r)}(k+1, k) \\ A_2^{(1)}(k+1, k) & A_2^{(2)}(k+1, k) & \dots & A_2^{(u)}(k+1, k) & \dots & A_2^{(r)}(k+1, k) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_l^{(1)}(k+1, k) & A_l^{(2)}(k+1, k) & \dots & A_l^{(u)}(k+1, k) & \dots & A_l^{(r)}(k+1, k) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ A_r^{(1)}(k+1, k) & A_r^{(2)}(k+1, k) & \dots & A_r^{(u)}(k+1, k) & \dots & A_r^{(r)}(k+1, k) \end{bmatrix} \quad (27)$$

$$W(k) = \left[(w^{(1)}(k))^T \quad (w^{(2)}(k))^T \quad \dots \quad (w^{(l)}(k))^T \quad \dots \quad (w^{(r)}(k))^T \quad (w_\beta(k))^T \quad (w_\gamma(k))^T \right]^T \quad (28)$$

The argmented state equation can be described as

$$X(k+1) = A(k+1, k)X(k) + W(k) \quad (29)$$

Based on the above-mentioned argmented state variables, the linearized description model of the observation equation is similarly described as follows:

$$\begin{aligned}
 & \begin{bmatrix} y_1^{(1)}(k+1) \\ y_2^{(2)}(k+1) \end{bmatrix} \\
 &= \begin{bmatrix} h_1^{(1)} & h_1^{(2)} & \dots & h_1^{(l)} & \dots & h_1^{(r)} & h_1(\beta(x)) & h_1(\gamma(x)) \\ h_2^{(1)} & h_2^{(2)} & \dots & h_2^{(l)} & \dots & h_2^{(r)} & h_2(\beta(x)) & h_2(\gamma(x)) \end{bmatrix} \begin{bmatrix} x^{(1)}(k+1) \\ x^{(2)}(k+1) \\ \vdots \\ x^{(l)}(k+1) \\ \vdots \\ x^{(r)}(k+1) \\ \beta(x(k+1)) \\ \gamma(x(k+1)) \end{bmatrix} + \begin{bmatrix} v_1^{(1)}(k+1) \\ v_2^{(1)}(k+1) \end{bmatrix} \\
 &= \begin{bmatrix} H^{(1)} & H^{(2)} & \dots & H^{(l)} & \dots & H^{(r)} & H(\beta) & H(\gamma) \end{bmatrix} \begin{bmatrix} x^{(1)}(k+1) \\ x^{(2)}(k+1) \\ \vdots \\ x^{(l)}(k+1) \\ \vdots \\ x^{(r)}(k+1) \\ \beta(x(k+1)) \\ \gamma(x(k+1)) \end{bmatrix} + v^{(1)}(k+1)
 \end{aligned} \tag{30}$$

Denote $H(k+1) = [H_u(k+1) \ H(\beta) \ H(\gamma)]$,
 $H_u(k+1) = [H^{(1)} \ H^{(2)} \ \dots \ H^{(u)} \ \dots \ H^{(r)}]$, $V(k+1) = \begin{bmatrix} v_1(k+1) \\ v_2(k+1) \end{bmatrix}$ where,
 $H(\beta) = 0, H(\gamma) = I$.

Then, the linearized representation of the nonlinear measurement model equation is as follows:

$$Y(k+1) = H(k+1)X(k+1) + V(k+1) \tag{31}$$

3.3. Recursive Filtering Algorithm

Considering the linearized state model and measurement model Equations (30) and (32) shown above, its statistical characteristics are as follows:

$$E\{W(k)W^T(k)\} = Q_w(k) \tag{32}$$

$$E\{V(k+1)V^T(k+1)\} = R_V(k+1) \tag{33}$$

$$E\{W(k)V^T(k+1)\} = 0 \tag{34}$$

Step 1: Set the initial values of the new system; then, according to the initial value $x(0)$ of $x(0)$, the following formula is satisfied:

$$E\{x(0)\} = \hat{x}_0 \tag{35}$$

$$E\{[x(0) - x_0][x(0) - x_0]^T\} = P_0 \tag{36}$$

Then, X_0 satisfies the following characteristics:

$$X(0) = \left[(x^{(1)}(0))^T \ (x^{(2)}(0))^T \ \dots \ (x^{(r)}(0))^T \ (\beta(x(0)))^T \ (\gamma(x(0)))^T \right]^T \tag{37}$$

$$\bar{P}_0 = \text{diag}\{P_0^{(1)} \ P_0^{(2)} \ \dots \ P_0^{(r)} \ P_0(\beta_0) \ P_0(\gamma_0)\} \tag{38}$$

$$E\{[X(0) - \hat{X}_0][X(0) - \hat{X}_0]^T\} = \bar{P}_0 \tag{39}$$

where $\bar{P}_0 = \text{diag}\{P_0^{(1)} \quad P_0^{(2)} \quad \dots \quad P_0^{(r)} \quad P_0(\beta_0) \quad P_0(\gamma_0)\} \geq 0$ is a positive semi-definite matrix.

Step 2: Recursive filtering

Assuming that $y(1), y(2), \dots, y(k)$ has been obtained, that is, $\hat{X}(k|k)$ and $\bar{P}(k|k)$ are known, the new higher-order Kalman filter is designed as follows:

$$\begin{aligned} \hat{X}(k+1|k+1) &= E\{X(k+1)|\hat{X}_0, y(1), y(2), \dots, y(k), y(k+1)\} \\ &= E\{X(k+1)|\hat{X}(k|k), y(k+1)\} \end{aligned} \quad (40)$$

The corresponding covariance matrix is as follows:

$$\bar{P}(k+1|k+1) = E\{[X(k+1) - \hat{X}(k+1|k+1)][X(k+1) - \hat{X}(k+1|k+1)]^T\} \quad (41)$$

Step 3: Time Update

(1) Obtain the following based on $\hat{X}(k|k)$ and $A(k+1, k)$:

$$\hat{X}(k+1|k) = A(k+1, k)\hat{X}(k|k) \quad (42)$$

(2) Obtain $\bar{P}(k+1|k)$ based on $\bar{P}(k|k)$ and $Q_W(k)$:

$$\bar{P}(k+1|k) = A(k+1|k)\bar{P}(k|k)A^T(k+1|k) + Q_W(k) \quad (43)$$

Step 4: Observe Update

(3) According to the relevant information of $\bar{P}(k+1|k)$ and the observed value, the gain matrix $K(k+1)$ is obtained:

$$\hat{Y}(k+1|k) = H(k+1)\hat{X}(k+1|k) \quad (44)$$

$$K(k+1) = \bar{P}(k+1|k)H^T(k+1) * [H(k+1)\bar{P}(k+1|k)H^T(k+1) + R_V(k+1)]^{-1} \quad (45)$$

(4) A higher-order Kalman filter is obtained from the remainders of the actual and predicted observations of $\hat{X}(k+1|k)$ and $K(k+1)$:

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)[Y(k+1) - H(k+1)\hat{X}(k+1|k)] \quad (46)$$

(5) Calculate the update error covariance matrix:

$$\bar{P}(k+1|k+1) = (I - K(k+1)H(k+1))\bar{P}(k+1|k) \quad (47)$$

$\hat{X}(k+1|k+1)$ is the obtained state estimate.

4. Performance Analysis

4.1. Projection Matrix Analysis

Rewrite the expanded state variables in Equation (46) as the original state variables and all higher-order variables and remainder variables in the following form:

$$\begin{aligned} \begin{bmatrix} \hat{x}(k+1|k+1) \\ \beta(\hat{x}(k+1|k+1)) \end{bmatrix} &= \begin{bmatrix} \hat{x}(k+1|k) \\ \beta(\hat{x}(k+1|k)) \end{bmatrix} \\ &+ \begin{bmatrix} K_x(k+1) \\ K_\beta(k+1) \end{bmatrix} * \left(\begin{bmatrix} h_x(k+1) & \gamma(h_x(k+1)) \end{bmatrix} \begin{bmatrix} e(k+1|k) \\ \beta(\hat{x}(k+1|k)) \end{bmatrix} + v(k+1) \right) \end{aligned} \quad (48)$$

where $K_x(k+1)$ and $K_\beta(k+1)$ are the gain matrices corresponding to the original variables and the remainder augmented-dimensional variables, respectively, and $h_x(k+1)$ and

$\gamma(h_x(k+1))$ are the observation matrices corresponding to the original variables and the remaining augmented-dimensional variables, respectively. Let the projection operator be

$$P_e = [I_x \quad O_\beta] \quad (49)$$

where I_x and O_β are matrices that match the dimensions of the original and remainder augmented variables, respectively.

Then,

$$\hat{x}(k+1|k+1) = P_e * \hat{X}(k+1|k+1) = [I_x \quad O_\beta] * \begin{bmatrix} \hat{x}(k+1|k+1) \\ \beta(\hat{x}(k+1|k+1)) \end{bmatrix} \quad (50)$$

Substituting Equation (48) into Equation (50), we obtain

$$\begin{aligned} \hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + K_x(k+1) * [h_x(k+1)e(k+1|k) \\ &\quad + \gamma(h_x(k+1))\beta(\hat{x}(k+1|k)) + v(k+1)] \\ &= \hat{x}(k+1|k) + K_x(k+1) * h_x(k+1)e(k+1|k) \\ &\quad + K_x(k+1) * \gamma(h_x(k+1))\beta(\hat{x}(k+1|k)) + K_x(k+1)v(k+1) \end{aligned} \quad (51)$$

It is found that after projection through the projection matrix, only the original system state estimation value is retained, which reduces the complexity of the algorithm and the actual computational burden, and includes more information from the model, improving the estimation accuracy.

4.2. Covariance Matrix Analysis

Bring Equation (48) into Equation (47) and at the same time divide the covariance matrix into blocks according to the original variables and the remaining dimension expansion variables, then write as follows:

$$\begin{aligned} &\begin{bmatrix} P_x(k+1|k+1) & P_{x\beta}(k+1|k+1) \\ P_{\beta x}(k+1|k+1) & P_\beta(k+1|k+1) \end{bmatrix} \\ &= \left(\begin{bmatrix} I_x & O \\ O & I_\beta \end{bmatrix} - \begin{bmatrix} K_x(k+1) \\ K_\beta(k+1) \end{bmatrix} \begin{bmatrix} h_x(k+1) & \gamma(h_x(k+1)) \end{bmatrix} \right) * \begin{bmatrix} P_x(k+1|k) & P_{x\beta}(k+1|k) \\ P_{\beta x}(k+1|k) & P_\beta(k+1|k) \end{bmatrix} \end{aligned} \quad (52)$$

where $P_x(k+1|k+1)$ is the covariance matrix of the original variable, $P_{x\beta}(k+1|k+1)$ is the covariance matrix of the original variable and the remainder variable, $P_{\beta x}(k+1|k+1)$ is the covariance matrix of the remainder variable and the original variable, and $P_\beta(k+1|k+1)$ is the covariance matrix of the remainder variable.

Then, the original variable covariance matrix can be calculated by the projection matrix as follows:

$$\begin{aligned} &P_x(k+1|k+1) \\ &= P_e * \bar{P}(k+1|k+1) * P_e^T \\ &= [I_x \quad O_\beta] * \begin{bmatrix} P_x(k+1|k+1) & P_{x\beta}(k+1|k+1) \\ P_{\beta x}(k+1|k+1) & P_\beta(k+1|k+1) \end{bmatrix} * \begin{bmatrix} I_x \\ O_\beta \end{bmatrix} \\ &= [I_x \quad O_\beta] * \left(I - \begin{bmatrix} K_x(k+1) \\ K_\beta(k+1) \end{bmatrix} * \begin{bmatrix} h_x(k+1) & \gamma(h_x(k+1)) \end{bmatrix} \right) \\ &\quad * \begin{bmatrix} P_x(k+1|k) & P_{x\beta}(k+1|k) \\ P_{\beta x}(k+1|k) & P_\beta(k+1|k) \end{bmatrix} * \begin{bmatrix} I_x \\ O_\beta \end{bmatrix} \\ &= (I_x - K_x(k+1) * h_x(k+1)) * P_x(k+1|k) - K_x(k+1) * \gamma(h_x(k+1)) * P_{x\beta}(k+1|k) \end{aligned} \quad (53)$$

In the experiment, it was found that $K_x(k+1) * \gamma(h_x(k+1)) * P_{x\beta}(k+1|k)$ is a positive-definite matrix. Compared with the original EKF algorithm, the posterior covariance matrix becomes smaller with the application of higher-order residual information, and the algorithm makes full use of the higher-order information of the model. Theoretically, the state estimation accuracy will be higher.

5. Simulation Experiments

For multi-robot formation systems, the state equation is as follows [24,35]:

$$\xi_i(k+1) = \xi_i(k) + \phi_i(k) \cos \theta_i(k) + c \sum_{j=1}^N \Gamma \xi_j(k) + \omega_i^x(k) \tag{54}$$

$$\psi_i(k+1) = \psi_i(k) + \phi_i(k) \sin \theta_i(k) + c \sum_{j=1}^N \Gamma \psi_j(k) + \omega_i^y(k) \tag{55}$$

$$\theta_i(k+1) = \theta_i(k) + \delta_i(k) + c \sum_{j=1}^N \Gamma \theta_j(k) + \omega_i^\theta(k) \tag{56}$$

where $(\xi_i(k), \psi_i(k))$ and $\theta_i(k)$ represent the position and direction of the i -th robot, respectively, and $\phi_i(k)$ and $\delta_i(k)$ represent the linear velocity and angular velocity, respectively. Suppose that $\phi_i(k) = 0.15$, $\delta_i(k) = 0.3$, $\omega_i(k) = (\omega_i^x(k), \omega_i^y(k), \omega_i^\theta(k))$ are Gaussian white noise with zero mean variance $Q_i(k)$.

In a multi-robot formation, Γ is a time-varying matrix related to the formation and Γ is the formation; if Γ is an upper triangular matrix, it means that the formation has a fixed movement formation, as shown in Figure 2. If $\Gamma = \Gamma(k)$, it means that the formation is time-varying and has a random switching topology, as shown in Figure 3. We use the extended Kalman filter (EKF) and high-order Kalman filter methods to estimate the second-order Kalman filter algorithm (SOKF), including the remainder extended Kalman filter algorithm (REKF), as well as the second-order remainder Kalman filter algorithm (SEORKF) for comparison.

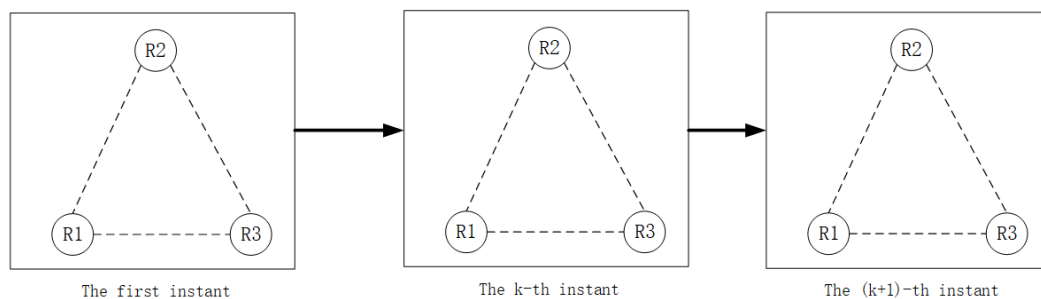


Figure 2. Multi-robot formation with fixed formation.

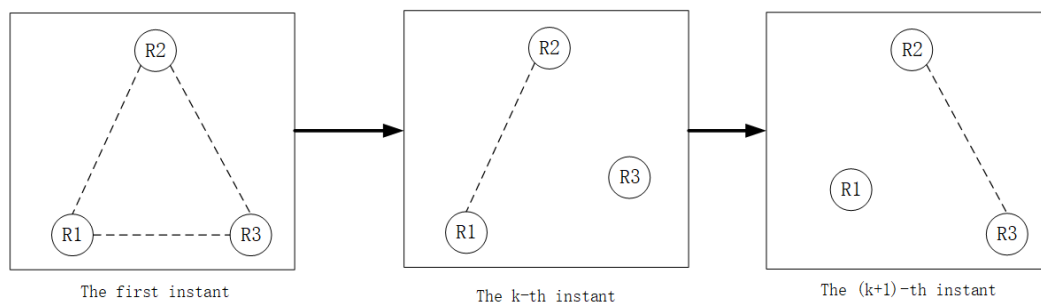


Figure 3. Multi-robot formation in a random topology formation.

As can be seen from the above figure, in a robot formation with a fixed formation, the robot maintains the formation at the first moment and continues to move with a unique nonlinear model. However, in a formation with a random switching topology, at the first moment it maintains a fixed formation structure. With the constraints of different communication or task division, the formation at each moment will be different. There are arbitrary formations or individual formations, and they maintain their own nonlinear motion.

The visual measurement is provided by the following equation:

$$p_i(k) = \frac{\gamma_u}{z_f^c} [-(s_x - \xi_i(k)) \sin \theta_i(k) + (s_x - \psi_i(k)) \cos \theta_i(k) - d_2] + p_0 + v_i^p(k) \quad (57)$$

$$q_i(k) = \frac{\gamma_v}{z_f^c} [-(s_y - \xi_i(k)) \cos \theta_i(k) - (s_y - \psi_i(k)) \sin \theta_i(k) - d_1] + q_0 + v_i^q(k) \quad (58)$$

where $(p_i(k), q_i(k))$ defines the coordinates of the robot in the image plane, (d_1, d_2) are its frame coordinates, z_f^c is the distance from the optical center of the camera to the robot, and γ_u and γ_v are variable pixel magnification factors; for a visual tracking system, the feature points are placed on the ceiling, (s_x, s_y) are the coordinates of the feature points in the world frame, and (p_0, q_0) is the camera image coordinate principal point; $v_i(k) = (v_i^p(k), v_i^q(k))$ is white Gaussian noise with zero mean variance $R_i(k)$.

In the simulation, the following parameters of the visual tracking system are adopted: $d_1 = -0.0668, d_2 = 0.0536, z_f^c = 2.1050, \gamma_u = 902.13283, \gamma_v = 902.50141, p_0 = 347.20436, q_0 = 284.34705$. The process noise covariance matrix is $Q_i(k+1) = \text{diag}\{0.01, 0.01, 0.01\}$, and the measurement noise covariance matrix is $R_i(k) = \text{diag}\{25^2, 25^2\}$ ($i = 1, 2, 3$).

To illustrate the tracking performance of the proposed filter, we use the root mean square error (MSE) of the three robots. Over 50 Monte-Carlo runs were obtained, and the table below shows the mean squared error for each robot position:

$$\begin{aligned} \text{MSE}_{\xi}(k) &= \frac{1}{3} \sum_{i=1}^3 \frac{1}{50} \sum_{m=1}^{50} (\xi_i(k) - \hat{\xi}_i^{(m)}(k))^2 \\ \text{MSE}_{\psi}(k) &= \frac{1}{3} \sum_{i=1}^3 \frac{1}{50} \sum_{m=1}^{50} (\psi_i(k) - \hat{\psi}_i^{(m)}(k))^2 \end{aligned} \quad (59)$$

where $(\hat{\xi}_i^{(m)}(k), \hat{\psi}_i^{(m)}(k))$ represents the position estimate of the i -th robot at the m -th Monte Carlo run.

5.1. Trajectory Estimation of Multi-Robot in Fixed Formation

In a multi-robot formation, Γ is a time-varying matrix related to the formation, Γ is the formation, and when Γ is an upper triangular matrix, it means that the formation has a fixed movement formation to complete a certain task. Table 1 shows the multi-robot trajectory estimation MSE with fixed formation, where 'Improved' refers to the improved estimation accuracy compared with EKF method.

The original nonlinear filtering algorithms based on Taylor expansion, such as extended Kalman filtering and high-order Kalman filtering methods, directly discard the truncation error. The estimation accuracy has a great influence. Table 2 shows the truncation errors generated by the EKF and second-order Kalman filtering methods for estimating the fixed formation trajectories of multi-robot. It can be seen from the table below that both methods have considerable truncation errors.

In the high-order Kalman filtering method, the remainder variables are introduced into the algorithm to make full use of the nonlinear model information, which can effectively avoid the influence of the truncation error on the estimation results; however, there will always be rounding errors. Here, the rounding error results generated by the first-order remainder extended Kalman filtering algorithm and the second-order remainder Kalman

filtering algorithm with a fixed formation of robots are compared and analyzed, as show in Table 3, with the result that the rounding error is much smaller than the truncation error.

Table 1. MSE for multi-robot trajectory estimation in fixed formation.

R	MSE of R1		MSE of R2		MSE of R3	
	x	y	x	y	x	y
Method						
EKF	1.9042	1.9836	0.1045	0.2788	1.0844	1.0605
UKF	0.4113	1.1985	0.0642	0.1266	0.4825	0.9536
(Improved)	(78.40%)	(39.58%)	(38.56%)	(54.59%)	(55.51%)	(10.08%)
CKF	0.6523	1.3720	0.0857	0.1382	0.5028	0.9783
(Improved)	(65.74%)	(30.83%)	(17.99%)	(50.43%)	(53.63%)	(7.75%)
REKF	0.7566	0.7721	0.0638	0.0625	0.3927	0.9391
(Improved)	(60.27%)	(61.08%)	(38.94%)	(77.58%)	(63.79%)	(11.45%)
SOKF	0.8428	1.7616	0.0793	0.2597	0.7113	1.0198
(Improved)	(55.74%)	(11.19%)	(24.11%)	(6.85%)	(34.41%)	(3.84%)
SORKF	0.2473	0.4087	0.0480	0.0597	0.0457	0.1956
(Improved)	(87.01%)	(78.98%)	(54.07%)	(78.59%)	(95.79%)	(81.56%)

Table 2. Truncation error of trajectory estimation in fixed-formation multi-robot.

R	Truncation Error of R1		Truncation Error of R2		Truncation Error of R3	
	x	y	x	y	x	y
Method						
EKF	1.9536	1.9243	0.1085	0.3117	1.0726	1.0734
SOKF	0.8729	1.7157	0.0905	0.2391	0.6910	0.9819
(Reduced)	(55.32%)	(10.84%)	(16.59%)	(23.29%)	(35.58%)	(8.52%)

Table 3. Rounding error of trajectory estimation in fixed-formation multi-robot swarm.

R	Rounding Error of R1		Rounding Error of R2		Rounding Error of R3	
	x	y	x	y	x	y
Method						
REKF	0.0818	0.0847	0.0179	0.0138	0.0445	0.1845
SORKF	0.0333	0.0487	0.0142	0.0125	0.0130	0.0320
(Reduced)	(59.29%)	(10.84%)	(20.67%)	(9.42%)	(70.79%)	(82.66%)

As follows, in the fixed formation, Figure 4 shows the real trajectory of the fixed multi-robot formation; Figure 5a,b shows the R1 estimation error in X-Coordinate and Y-Coordinate respectively; Figure 6 shows the histogram of R1 estimation error; Figure 7a,b shows the R2 estimation error in X-Coordinate and Y-Coordinate respectively; Figure 8 shows the histogram of R2 estimation error; Figure 9a,b shows the R3 estimation error in X-Coordinate and Y-Coordinate respectively; Figure 10 shows the histogram of R3 estimation error.

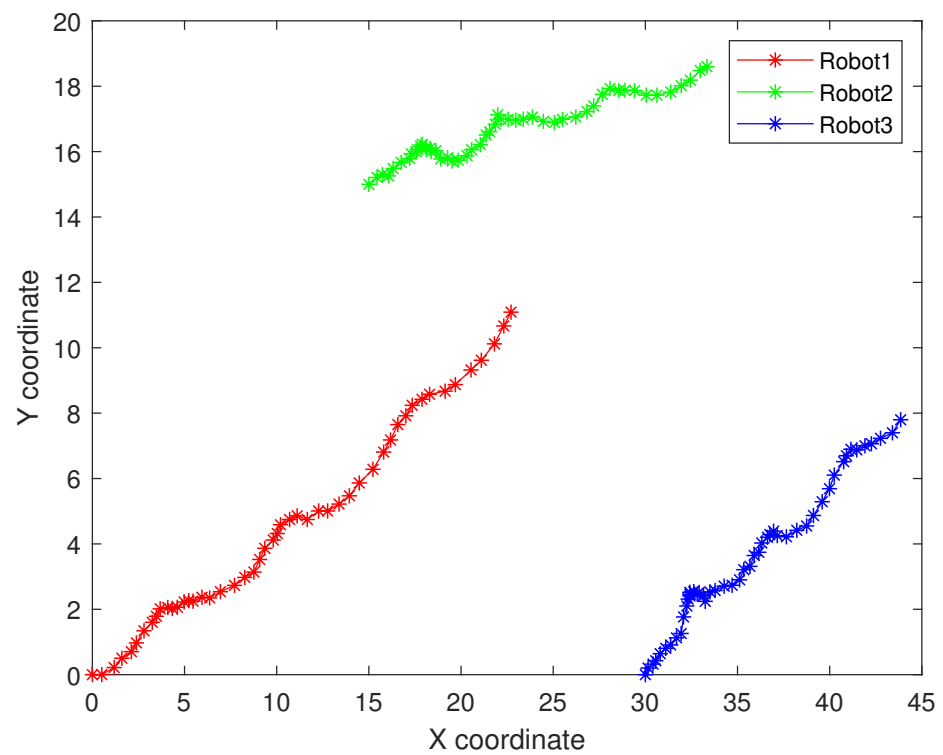


Figure 4. The real trajectory of the fixed multi-robot formation.

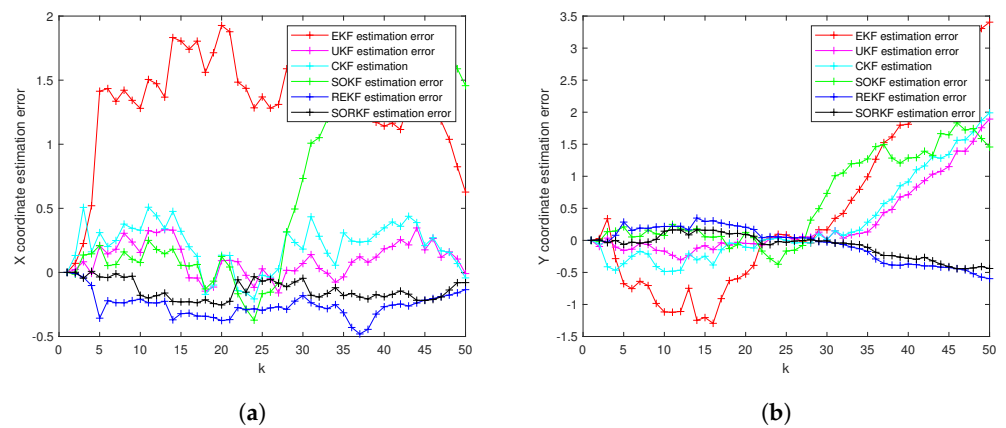


Figure 5. R1 estimation error of multi-robot in fixed formation. (a) X-Coordinate estimation error; (b) Y-Coordinate estimation error.

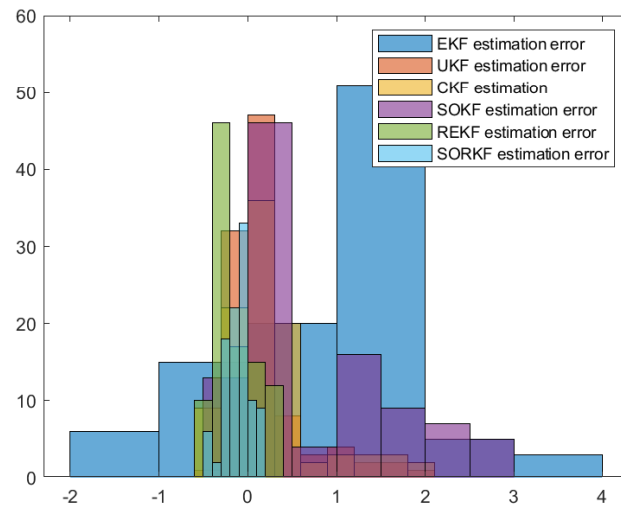


Figure 6. Histogram of R1 positioning estimation error in fixed formation.

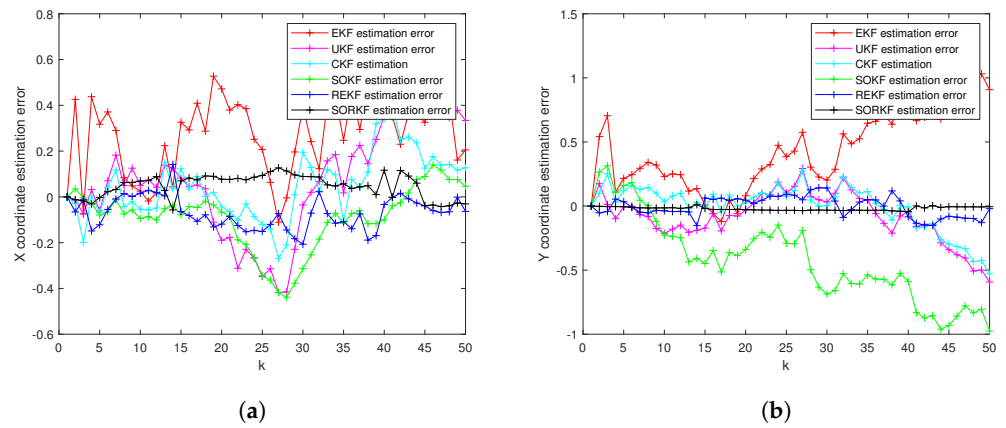


Figure 7. R2 estimation error of multi-robot in fixed formation. (a) X-Coordinate estimation error; (b) Y-Coordinate estimation error.

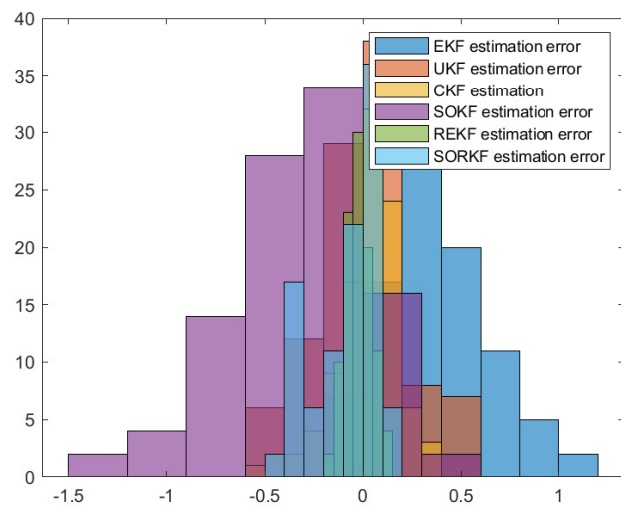


Figure 8. Histogram of R2 positioning estimation error in fixed formation.

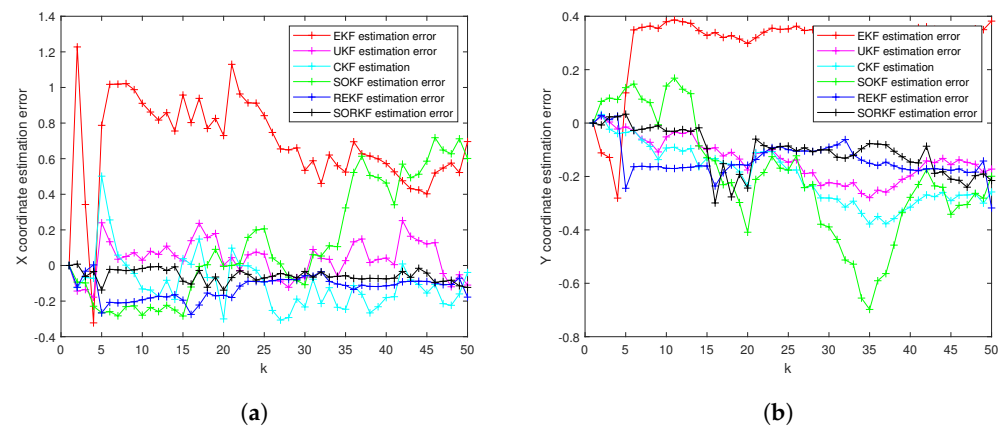


Figure 9. R3 estimation error of multi-robot in fixed formation. (a) X-Coordinate estimation error; (b) Y-Coordinate estimation error.

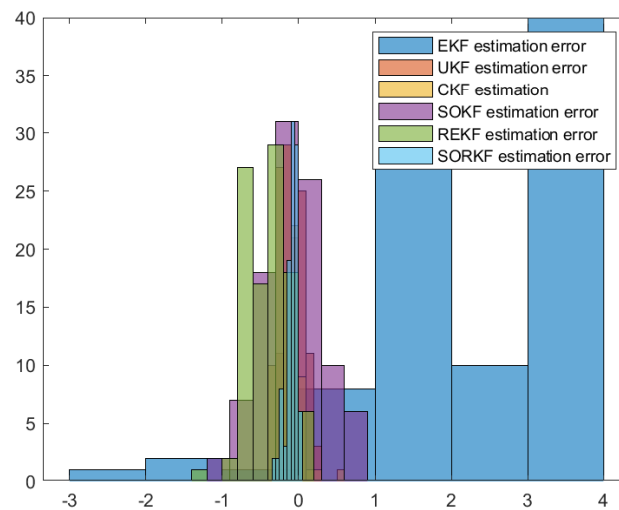


Figure 10. Histogram of R3 positioning estimation error in fixed formation.

5.2. Multi-Robot Formation Trajectory Estimation with Random Topology Structure

In the multi-robot formation, Γ is the time-varying matrix related to the formation, that is, $\Gamma = \Gamma(k)$, indicating that the formation has a random switching topology. Table 4 shows MSE for multi-robot trajectory estimation and Table 5 shows the truncation error of trajectory estimation. The truncation error of the trajectory estimation extended Kalman algorithm and the second-order Kalman filtering algorithm in Table 6 shows the rounding error when the trajectory estimation is performed by introducing the remainder variables into the above two algorithms, and the results are similar to the trajectory estimation under the fixed formation.

As follows, in random topology formation, Figure 11 shows the real trajectory; Figure 12a,b shows the R1 estimation error in X-Coordinate and Y-Coordinate respectively; Figure 13 shows the histogram of R1 estimation error; Figure 14a,b shows the R2 estimation error in X-Coordinate and Y-Coordinate respectively; Figure 15 shows the histogram of R2 estimation error; Figure 16a,b shows the R3 estimation error in X-Coordinate and Y-Coordinate respectively; Figure 17 shows the histogram of R3 estimation error.

Table 4. MSE for multi-robot trajectory estimation in random topology formation.

R	MSE of R1		MSE of R2		MSE of R3	
	x	y	x	y	x	y
Method						
EKF	0.0458	0.0886	0.1895	0.9687	0.1523	0.3874
UKF	0.0431	0.0632	0.1382	0.4635	0.0757	0.0877
(Improved)	(5.90%)	(28.67%)	(27.07%)	(52.15%)	(50.29%)	(77.36%)
CKF	0.0362	0.0649	0.0825	0.2875	0.0586	0.0732
(Improved)	(20.96%)	(26.75%)	(56.46%)	(70.51%)	(61.52%)	(81.10%)
REKF	0.0390	0.0426	0.1391	0.2143	0.0061	0.0338
(Improved)	(14.85%)	(51.92%)	(26.60%)	(77.88%)	(95.99%)	(91.28%)
SOKF	0.0433	0.0451	0.1497	0.4809	0.0412	0.0850
(Improved)	(5.46%)	(49.10%)	(21.00%)	(50.36%)	(72.95%)	(66.75%)
SORKF	0.0340	0.0366	0.0191	0.0433	0.0044	0.0325
(Improved)	(25.76%)	(62.08%)	(89.92%)	(95.53%)	(97.11%)	(91.61%)

Table 5. Truncation error of trajectory estimation in random topology formation.

R	Truncation Error of R1		Truncation Error of R2		Truncation Error of R3	
	x	y	x	y	x	y
Method						
EKF	0.0629	0.0905	0.2299	1.0499	0.1919	0.4147
SOKF	0.0408	0.0492	0.1579	0.4622	0.0606	0.1625
(Reduced)	(35.14%)	(45.64%)	(29.16%)	(55.98%)	(68.42%)	(60.82%)

Table 6. Rounding error of trajectory estimation in random topology formation.

R	Rounding Error of R1		Rounding Error of R2		Rounding Error of R3	
	x	y	x	y	x	y
Method						
REKF	0.0165	0.0136	0.0543	0.0861	0.0124	0.0235
SORKF	0.0126	0.0121	0.0111	0.0110	0.0117	0.0169
(Reduced)	(23.64%)	(11.03%)	(79.56%)	(87.22%)	(5.65%)	(28.09%)

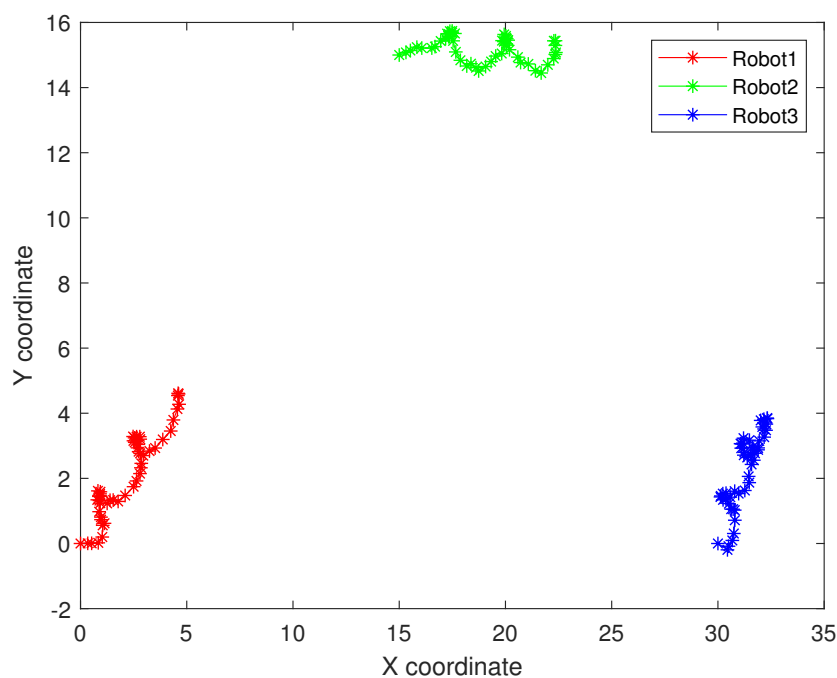


Figure 11. The real trajectory of the multi-robot swarm in random topology formation.

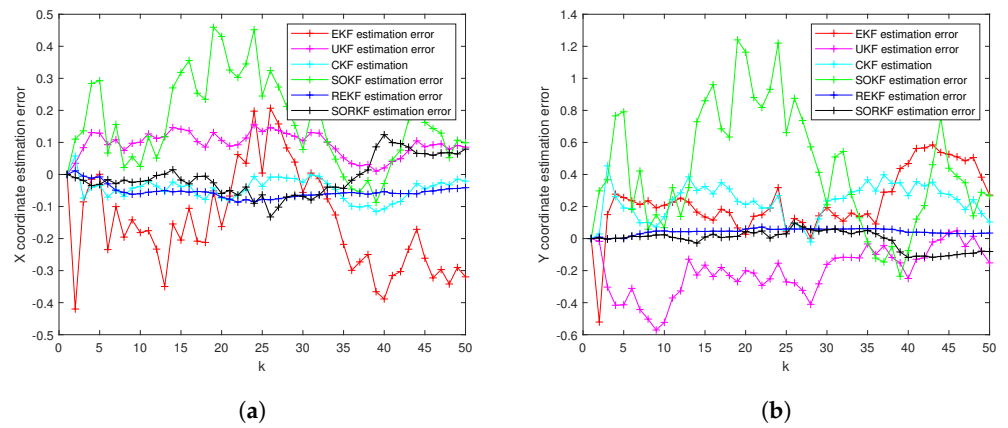


Figure 12. R1 estimation error of multi-robot swarm in random topology formation. (a) X-Coordinate estimation error; (b) Y-Coordinate estimation error.

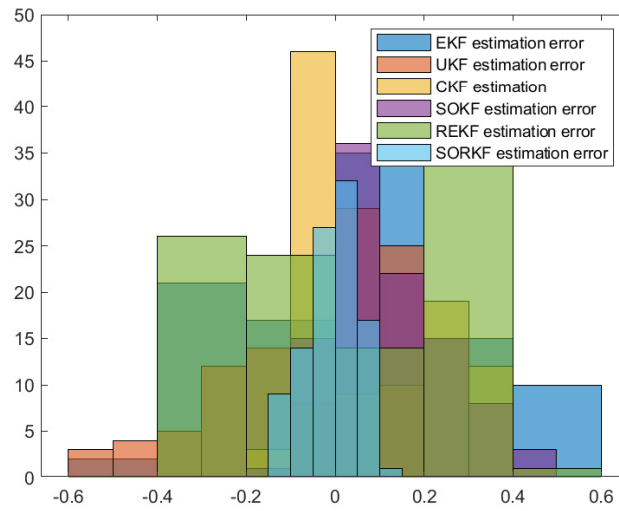


Figure 13. Histogram of R1 positioning estimation error in random topology formation.

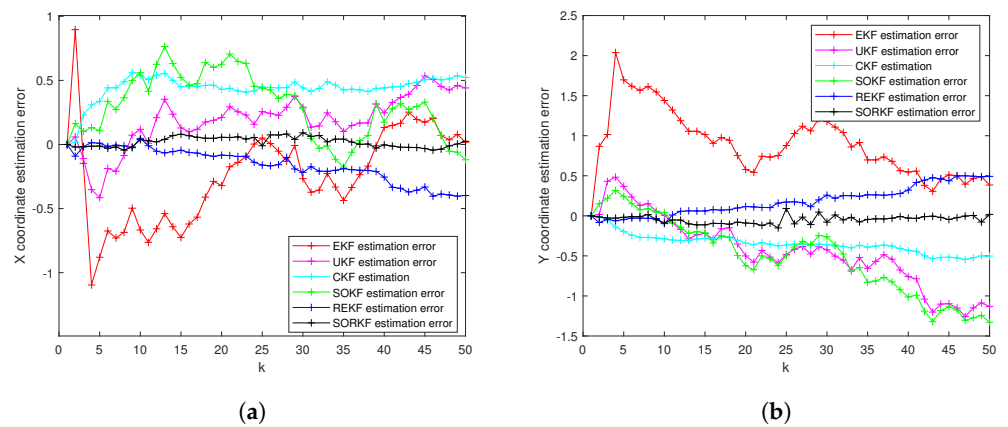


Figure 14. R2 estimation error of multi-robot swarm in random topology formation. (a) X-Coordinate estimation error; (b) Y-Coordinate estimation error.

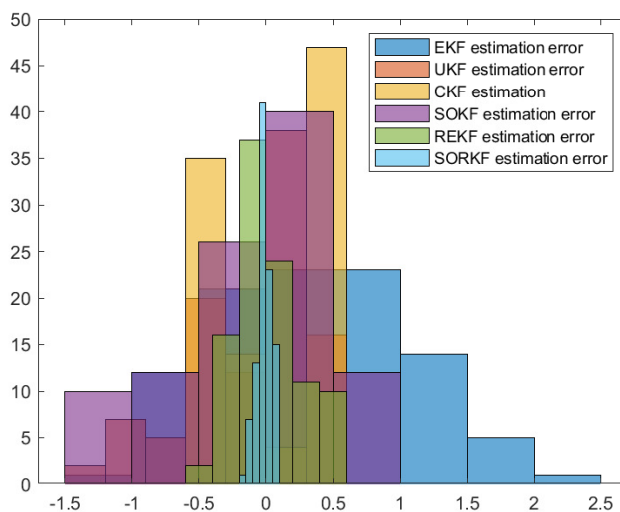


Figure 15. Histogram of R2 positioning estimation error in random topology formation.

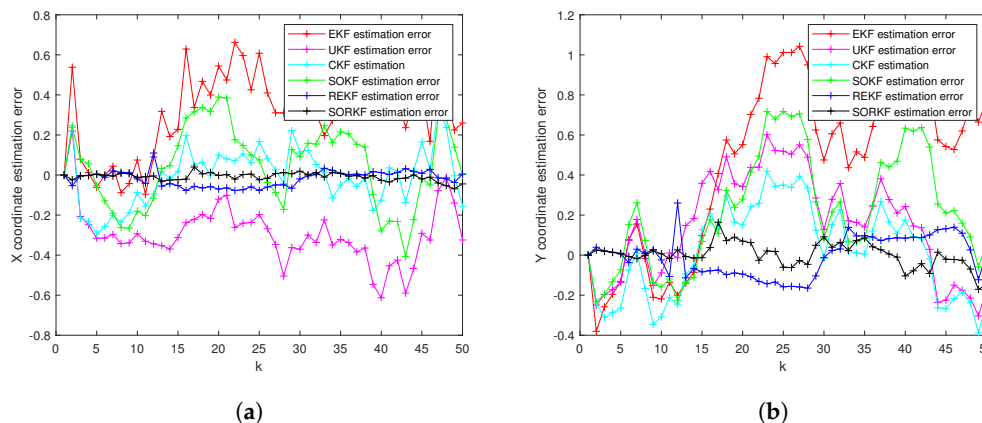


Figure 16. R3 estimation error of multi-robot swarm in random topology formation. (a) X-Coordinate estimation error; (b) Y-Coordinate estimation error.

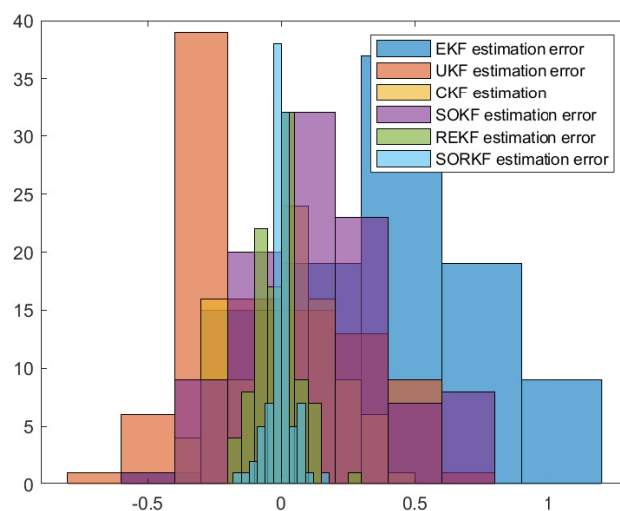


Figure 17. Histogram of R3 positioning estimation error in random topology formation.

As shown in the figure above, we use a fixed formation formation and a random topology formation to respectively estimate the actual trajectories of the three robots, use the estimation error of each robot in the x–y direction of the two-dimensional plane as a measure, and use the EKF and the second-order Kalman filtering(SOKF) methods,

respectively, to introduce the residual term variable in each of these two methods. From the results, it can be concluded that the estimation accuracy of EKF is the lowest, followed by the second-order Kalman filtering algorithm (SOKF), the remainder extended Kalman filtering Algorithm (REKF), and the remainder second-order Kalman filtering algorithm (SORKF). The other three filtering methods designed according to EKF structure, excepting EKF, significantly improve the estimation accuracy of EKF. Thus, we can see that for the robot in the motion model, the degree of non-linearity of the turning characteristics is relatively high. If first-order Taylor expansion is performed, the information contained in it cannot be completely extracted, which inevitably leads to a decrease in the estimation accuracy. The higher the degree of expansion performed, the higher the robot position estimation accuracy, and the more accurate the robot estimation estimation. In addition, we analyzed the truncation error of the EKF and second-order Kalman filtering(SOKF) algorithms. The truncation error has a great influence on the estimation accuracy. When the nonlinearity is strong, the model information cannot be fully utilized by directly discarding the truncation error, resulting in low estimation accuracy. After the variables are introduced, the rounding error of the estimation results is analyzed, and it is found that the rounding error is much smaller than the truncation error. The Taylor expansion of the nonlinear model and the use of remainder variables can make full use of the model information to improve the estimation accuracy and make the estimation more accurate.

6. Conclusions and Future Work

This paper studies the multi-robot trajectory estimation problem, and proposes a fusion estimation method based on the high-order Kalman filter algorithm. The extended Kalman filter (EKF) algorithm used in the existing robot position estimation only considers first-order expansion and ignores the high-order information. To solve the problem, a joint trajectory estimation method with a multi-robot formation based on the high-order Kalman filtering method was adopted, the Taylor expansion of the state equation and the observation equation was carried out, and the remainder variables were introduced on this basis, effectively improving the estimation accuracy. Through a simulation, we found that for robot fixed formations and random topology formations the introduction of remainder variables improves the accuracy of position estimation by more than 50%, and in certain cases even more, compared to the EKF algorithm. At the same time, the truncation error in the estimation process of the EKF and the high-order Kalman filter algorithms was analyzed and compared with the rounding error of the estimation algorithm after the introduction of the remainder variable. We found that the rounding error is much smaller than the truncation error. After the remainder variable is introduced during filtering, the algorithm makes full use of the model information, and the estimation accuracy is greatly improved.

In future research, we may encounter more complex models. If the state model has strong nonlinear characteristics, the observation model has super strong nonlinear characteristics, and the modeling error is non Gaussian white noise, our method will be difficult to achieve, and will be completed with the help of the characteristic function filtering method [36,37]; At the same time, for the neural network model, we can also introduce the high-order Kalman filtering method to update the real-time parameters [38]; Finally, for the state estimation of non-cooperative targets, there must be multiple targets and sensors. We can extend the method in this paper to distributed filtering. Similarly, for the distributed model of Federated learning, we will also be applicable [39,40].

Author Contributions: Conceptualization, M.W., C.W. and W.L.; methodology, C.W.; software, M.W.; validation, M.W.; formal analysis, M.W.; investigation, M.W.; resources, M.W.; data curation, M.W.; writing—original draft preparation, M.W.; writing—review and editing, C.W. and W.L.; visualization, C.W.; supervision, C.W. and W.L.; project administration, C.W.; funding acquisition, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shaanxi Province key research and development program 2021GY-087.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A. Mathematical Symbols

The important mathematical symbols used in the text are described as follows:

Table A1. Mathematical Symbols.

Mathematical Symbols		Description
x	X	State vector
y	Y	Observation vector
\hat{x}	\hat{X}	State estimation
\hat{y}	\hat{Y}	Observation estimation
$f(\cdot)$	$h(\cdot)$	Nonlinear continuous function
	P	Covariance matrix
	K	Kalman gain
	A	State transfer matrix
	H	Observation matrix
	$\beta(\cdot)$	State remainder variable
	$\gamma(\cdot)$	Observation remainder variable

References

- Liu, H.; Xu, X.; Lu, J.A.; Chen, G.; Zeng, Z. Optimizing pinning control of complex dynamical networks based on spectral properties of grounded Laplacian matrices. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *51*, 786–796. [\[CrossRef\]](#)
- Qiu, H.X.; Duan, H.B. From collective flight in bird flocks to unmanned aerial vehicle autonomous swarm formation. *Chin. J. Eng.* **2017**, *39*, 317–322. [\[CrossRef\]](#)
- Cornejo, A.; Rubenstein, M.; Nagpal, R. Programmable self-assembly in a thousand-robot swarm. *Science* **2014**, *345*, 795–799.
- Vicsek, T.; Zafeiris, A. Collective motion. *Phys. Rep.* **2012**, *517*, 71–140. [\[CrossRef\]](#)
- Li, T.; Qiu, Q.; Zhao, C. A fully distributed protocol with an event-triggered communication strategy for second-order multi-agent systems consensus with nonlinear dynamics. *Sensors* **2021**, *21*, 4059. [\[CrossRef\]](#)
- Yan, J.; Yu, Y.; Wang, X. Distance-Based Formation Control for Fixed-Wing UAVs with Input Constraints: A Low Gain Method. *Drones* **2022**, *6*, 159. [\[CrossRef\]](#)
- Martinez, J.B.; Becerra, H.M.; Gomez-Gutierrez, D. Formation tracking control and obstacle avoidance of unicycle-type robots guaranteeing continuous velocities. *Sensors* **2021**, *21*, 4374. [\[CrossRef\]](#)
- Jiangfeng, Z.; Chunxiao, L.; Shuzhe, S.; Dadong, H.; Biao, W.; Jun, C.; Ying, T. Research on Decision-Making optimization for Operation System of Power Grid Framework under Smart Grid and Dijkstra Algorithm. In Proceedings of the 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA), Dalian, China, 29–31 October 2021; pp. 378–382. [\[CrossRef\]](#)
- Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [\[CrossRef\]](#)
- Dean, G. An introduction to Kalman filters. *Meas. Control* **1986**, *19*, 69–73. [\[CrossRef\]](#)
- Lee, T.S. Theory and application of adaptive fading memory Kalman filters. *IEEE Trans. Circuits Syst.* **1988**, *35*, 474–477. [\[CrossRef\]](#)
- Meinhold, R.J.; Singpurwalla, N.D. Robustification of Kalman filter models. *J. Am. Stat. Assoc.* **1989**, *84*, 479–486. [\[CrossRef\]](#)
- Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [\[CrossRef\]](#)
- Yang, F.; Zheng, L.T.; Wang, J.Q.; Pan, Q. Double Layer Unscented Kalman Filter. *Acta Autom. Sin.* **2019**, *45*, 1386–1391
- Perea, L.; Elosegui, P. New state update equation for the unscented Kalman filter. *J. Guid. Control Dyn.* **2008**, *31*, 1500–1504. [\[CrossRef\]](#)
- Chandra, K.; Gu, D.W.; Postlethwaite, I. Square Root Cubature Information Filter. *IEEE Sens. J.* **2013**, *13*, 750–758. [\[CrossRef\]](#)
- Cui, T.; Sun, X.; Wen, C. A Novel Data Sampling Driven Kalman Filter Is Designed by Combining the Characteristic Sampling of UKF and the Random Sampling of EnKF. *Sensors* **2022**, *22*, 1343. [\[CrossRef\]](#)
- Arasaratnam, I.; Haykin, S. Cubature kalman filters. *IEEE Trans. Autom. Control* **2009**, *54*, 1254–1269. [\[CrossRef\]](#)
- Arasaratnam, I.; Haykin, S. Square-root quadrature Kalman filtering. *IEEE Trans. Signal Process.* **2008**, *56*, 2589–2593. [\[CrossRef\]](#)
- Rvg, A.; Pcpmp, A.; Hkk, B.; Mcz, C. Nonlinear filtering for sequential spacecraft attitude estimation with real data: Cubature Kalman Filter, Unscented Kalman Filter and Extended Kalman Filter - ScienceDirect. *Adv. Sp. Res.* **2019**, *63*, 1038–1050.

21. Song, W.; Wang, J.; Zhao, S.; Shan, J. Event-triggered cooperative unscented Kalman filtering and its application in multi-UAV systems. *Automatica* **2019**, *105*, 264–273. [[CrossRef](#)]
22. Li, W.; Wang, Z.; Ho, D.; Wei, G. On Boundedness of Error Covariances for Kalman Consensus Filtering Problems. *IEEE Trans. Autom. Control* **2019**, *65*, 2654–2661. [[CrossRef](#)]
23. Hu, J.; Wang, Z.; Liu, S.; Gao, H. A variance-constrained approach to recursive state estimation for time-varying complex networks with missing measurements. *Automatica* **2016**, *64*, 155–162. [[CrossRef](#)]
24. Li, W.; Jia, Y.; Du, J. State estimation for stochastic complex networks with switching topology. *IEEE Trans. Autom. Control* **2017**, *62*, 6377–6384. [[CrossRef](#)]
25. Xiaohui, S.; Chenglin, W.; Tao, W. High-Order Extended Kalman Filter Design for a Class of Complex Dynamic Systems with Polynomial Nonlinearities. *Chin. J. Electron.* **2021**, *30*, 508–515. [[CrossRef](#)]
26. Sun, X.; Wen, T.; Wen, C.; Cheng, X.; Yunkai, W.U. High-Order Extended Strong Tracking Filter. *Chin. J. Electron.* **2021**, *30*, 1152–1158.
27. Xiaohui, S.; Chenglin, W.; Tao, W. A Novel Step-by-Step High-Order Extended Kalman Filter Design for a Class of Complex Systems with Multiple Basic Multipliers. *Chin. J. Electron.* **2021**, *30*, 313–321. [[CrossRef](#)]
28. SUN, X.; WEN, C.; WEN, T. Maximum Correntropy High-Order Extended Kalman Filter. *Chin. J. Electron.* **2022**, *31*, 190–198.
29. Wang, Q.; Sun, X.; Wen, C. Design method for a higher order extended Kalman filter based on maximum correlation entropy and a Taylor network system. *Sensors* **2021**, *21*, 5864. [[CrossRef](#)] [[PubMed](#)]
30. Liu, X.; Wen, C.; Sun, X. Design Method of High-Order Kalman Filter for Strong Nonlinear System Based on Kronecker Product Transform. *Sensors* **2022**, *22*, 653. [[CrossRef](#)] [[PubMed](#)]
31. Wang, J.; Wen, C. Real-Time Updating High-Order Extended Kalman Filtering Method Based on Fixed-Step Life Prediction for Vehicle Lithium-Ion Batteries. *Sensors* **2022**, *22*, 2574. [[CrossRef](#)] [[PubMed](#)]
32. Chow, S.M.; Ferrer, E.; Nesselroade, J.R. An unscented Kalman filter approach to the estimation of nonlinear dynamical systems models. *Multivar. Behav. Res.* **2007**, *42*, 283–321. [[CrossRef](#)] [[PubMed](#)]
33. Guo, L.; Wang, H. Minimum entropy filtering for multivariate stochastic systems with non-Gaussian noises. *IEEE Trans. Autom. Control* **2006**, *51*, 695–700.
34. Zhou, J.; Wang, H.; Zhou, D. PDF tracking filter design using hybrid characteristic functions. In Proceedings of the 2008 American Control Conference, Seattle, WA, USA, 11–13 June 2008; pp. 3046–3051.
35. Chen, X.; Jia, Y. Indoor localization for mobile robots using lampshade corners as landmarks: Visual system calibration, feature extraction and experiments. *Inter. J. Control Autom. Syst.* **2014**, *12*, 1313–1322. [[CrossRef](#)]
36. Wen, C.; Cheng, X.; Xu, D.; Wen, C. Filter design based on characteristic functions for one class of multi-dimensional nonlinear non-Gaussian systems. *Automatica* **2017**, *82*, 171–180. [[CrossRef](#)]
37. Wen, C.; Ge, Q.; Cheng, X.; Xu, D. Filters design based on multiple characteristic functions for the grinding process cylindrical workpieces. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4671–4679. [[CrossRef](#)]
38. Wen, T.; Xie, G.; Cao, Y.; Cai, B. A DNN-based channel model for network planning in train control systems. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 2392–2399. [[CrossRef](#)]
39. Ma, X.; Wen, C.; Wen, T. An asynchronous and real-time update paradigm of federated learning for fault diagnosis. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8531–8540. [[CrossRef](#)]
40. Xue, M.; Chenglin, W. An Asynchronous Quasi-Cloud/Edge/Client Collaborative Federated Learning Mechanism for Fault Diagnosis. *Chin. J. Electron.* **2021**, *30*, 969–977. [[CrossRef](#)]