

ORIGINAL ARTICLE

Open Access



A cluster positioning architecture and relative positioning algorithm based on pigeon flock bionics

Zhongliang Deng¹, Hang Qi^{1*} , Chengfeng Wu², Enwen Hu¹ and Runmin Wang¹

Abstract

Unmanned clusters can realize collaborative work, flexible configuration, and efficient operation, which has become an important development trend of unmanned platforms. Cluster positioning is important for ensuring the normal operation of unmanned clusters. The existing solutions have some problems such as requiring external system assistance, high system complexity, poor architecture scalability, and accumulation of positioning errors over time. Without the aid of the information outside the cluster, we plan to construct the relative position relationship with north alignment to adopt formation control and achieve robust cluster relative positioning. Based on the idea of bionics, this paper proposes a cluster robust hierarchical positioning architecture by analyzing the autonomous behavior of pigeon flocks. We divide the clusters into follower clusters, core clusters, and leader nodes, which can realize flexible networking and cluster expansion. Aiming at the core cluster that is the most critical to relative positioning in the architecture, we propose a cluster relative positioning algorithm based on spatiotemporal correlation information. With the design idea of low cost and large-scale application, the algorithm uses intra-cluster ranging and the inertial navigation motion vector to construct the positioning equation and solves it through the Multidimensional Scaling (MDS) and Multiple Objective Particle Swarm Optimization (MOPSO) algorithms. The cluster formation is abstracted as a mixed direction-distance graph and the graph rigidity theory is used to analyze localizability conditions of the algorithm. We designed the cluster positioning simulation software and conducted localizability tests and positioning accuracy tests in different scenarios. Compared with the relative positioning algorithm based on Extended Kalman Filter (EKF), the algorithm proposed in this paper has more relaxed positioning conditions and can adapt to a variety of scenarios. It also has higher relative positioning accuracy, and the error does not accumulate over time.

Keywords: Cluster positioning architecture, Cluster relative positioning, Multidimensional scaling, Multiple objective particle swarm optimization, Unmanned aerial vehicles positioning, Localizability analysis, Rigid graph

Introduction

At present, unmanned platforms with the characteristics of flexible configuration, multi-function, and miniaturization are widely used in detection, inspection, delivery, and other scenarios. However, due to the limitation of cost, volume, and other factors, one unmanned platform is difficult to meet the requirements in complex

environments and tasks. Unmanned cluster as an important development trend, is more efficient through the collaboration among members (Gautam & Mohan, 2012; Tahir et al., 2019). The advantages of unmanned clusters are as follows:

- (1) Parallel perception, distributed computing and execution capabilities, and better fault tolerance and robustness. Multiple Unmanned Aerial Vehicles (UAV) can realize multi-dimensional parallel perception through the complementary collocation

*Correspondence: chgmqh@163.com

¹ Beijing University of Posts and Telecommunications, Beijing 100876, China
Full list of author information is available at the end of the article

among heterogeneous sensors and distributed execution of total tasks through task splitting and reasonable allocation. In addition, when some UAVs fail, other UAVs in the cluster can replace them to complete the scheduled tasks, to improve the fault tolerance and robustness of the system.

- (2) A higher capacity limit to complete the tasks that are difficult for a single machine to complete. Through collaboration, unmanned cluster systems achieve capabilities beyond the simple superposition of individual unmanned nodes, so it has better function expansion and can adapt to more types of tasks.
- (3) First, through reasonable system design and coordination, a low-cost unmanned cluster system can be used to replace a single complex system with high cost, which is more economical. Secondly, based on the design concept of miniaturization, integration, modularization, and collaboration, it can greatly reduce the design cycle and the cost of building a new task platform.

Navigation and positioning are important for ensuring the normal operation of unmanned clusters. They form the basis for the research and development of unmanned clusters together with cluster architecture design, task allocation, collaborative control, self-organizing networks, and other technologies, and become a key research direction (Gyagenda et al., 2022; WU et al., 2022). The existing cluster navigation and positioning methods mainly face the following problems:

- 1) Need the assistance of an external positioning system, resulting in poor flexibility and easy interference. For example, radio positioning based on Global Navigation Satellite System (GNSS) (Yoo & Ahn, 2003) and road-based transmitters (Shamaei & Kassar, 2019), and visual positioning based on landmarks (Conte & Doherty, 2008).
- 2) The constructed relative coordinate system has no physical meaning and cannot be used for cluster control and mission planning. For example, the positioning algorithm of principal components analysis, MDS (Strohmeier et al., 2018), Fast Clustering MDS (Fan et al., 2020), vMDS (Kumar et al., 2016), etc.
- 3) The multi-source fusion algorithm can achieve high positioning accuracy but requires many types of sensors, which leads to a complex system and high cost, and it is difficult to apply to large-scale clusters. For example, positioning systems based on binocular vision cameras/Inertial Measurement Unit (IMU)/

Ultra-Wide Band (UWB)/Lidar often need to be equipped with high-performance computing boards (Chen & Gao, 2019).

- 4) Accumulated errors during long-term operation can cause the divergence of positioning results. Most exist in the positioning algorithm based on the inertial navigation system and without external information assistance, such as the EKF positioning algorithm based on IMU and relative distance or angle (Masinjila, 2016).
- 5) As the number of cluster nodes increases, the resource occupancy and computational complexity of the positioning algorithm will increase exponentially. This results in the poor scalability of the localization algorithm and cannot adapt to large-scale clusters, such as partially centralized algorithms (Panzieri et al., 2006).

Compared with traditional wireless sensor networks, unmanned cluster networks are more complex (Doriya et al., 2015), mainly reflected in the following points:

- (1) Cluster nodes are often in a relative motion state, resulting in dynamic changes in the cluster geometry, and the relative position configuration between nodes will affect the overall positioning accuracy of the cluster.
- (2) There is a conflict between the growing number of clusters and limited radio performance. With the increase of the number of cluster nodes, the distance between nodes becomes larger, but the communication distance is limited, which makes it difficult for unmanned cluster networks to achieve global direct connectivity. The network structure is characterized by regionalization and multi-hop connectivity.
- (3) With the movement between nodes, the topology relationship of the cluster network and the communication connection will change, and it is difficult to ensure that all nodes can establish stable connections.
- (4) Relative positioning is more critical than absolute positioning for cluster control and maintenance. At the same time, the north and east directions of the relative coordinate system should be consistent with the geodetic coordinate system, such as the North East Down (NED) coordinate system
- (5) Considering the robustness of cluster positioning and flexible deployment, the relative cluster positioning algorithm should use the observations

inside the cluster and avoid using external signal sources.

- (6) Considering the economy, to adapt to large-scale cluster applications it is necessary to reduce the number and types of sensors on a single node as much as possible, as well as the computing power required for positioning solutions.

Based on the above analysis, we plan to construct the relative position relationship with north alignment to adopt formation control and achieve robust cluster relative positioning without the aid of the information outside the cluster. This paper proposes a hierarchical unmanned cluster positioning architecture based on the bionic concept of pigeon flock and adopts the design of virtual general leader+core cluster+follower cluster to realize a flexible and scalable unmanned cluster. At the same time, considering the robustness of positioning and limited hardware resources, this paper proposes a relative positioning algorithm for unmanned clusters that only relies on internal ranging and inertial navigation data. The algorithm uses the spatial-temporal correlation information to solve the localization and reduces the dimension of the solution space through the MDS method. At the same time, the MOPSO algorithm is used to calculate the dimensionality reduction objective function to obtain the localization result. The algorithm constructs the relative coordinate relationship in the NED coordinate system, which is convenient for cluster control. At the same time, this paper uses hybrid graph theory to analyze the localizability of the proposed algorithm and obtains the node state in the scene that meets the localization conditions, which can provide theoretical support for formation control. Finally, the paper selects some typical scenarios for simulation test analysis. The simulation results show that the cluster relative positioning algorithm proposed in this paper has good positioning accuracy and error divergence suppression characteristics, and verifies the localizability analysis conclusion.

Cluster positioning architecture and model

Many creatures in nature show cluster phenomenon, such as fish, birds, wolves, insects, and so on. Individuals in a cluster exhibit certain group behaviors through mutual communication and cooperation, such as cooperative hunting by wolves, formation flying of birds, and nectar collection by bees. These biological swarm behaviors formed through long-term evolution provide inspiration and reference for us to design the unmanned cluster positioning architecture.

Individuals in biological groups often follow simple behavioral rules and have only limited ability to sense,

plan, and communicate. Individuals adjust their behaviors by interacting with neighboring peers and obtaining the information about the surrounding environment, and finally achieve a unified cluster behavior. The cluster system has strong robustness and will not cause fatal effects to the whole system due to the failure of some individuals (Garnier et al., 2007).

A pigeon flock is a collection of a large number of autonomous individuals. Through the interaction between individuals, the entire pigeon flock presents a complex macroscopic emergent behavior. The positioning and navigation of the pigeon group show three characteristics (Luo & Duan, 2017):

- (1) Limited by the vision and communication distance of the pigeons, the individuals in the pigeon group follow a topological distance interaction method, that is, each pigeon only interacts with a certain number of individuals around it, which makes the information interaction of the whole pigeon group simple and efficient.
- (2) The pigeon group presents as a hierarchical network in the flight, high-level individuals play a leading role, and the behavior of low-level individuals is affected by high-level individuals. At the same time, the rule of the pigeon group formation is not a common single-leader system. The lower-class pigeons only need to communicate in real-time with the nearest higher-class pigeons and follow. This network structure enables the group to respond quickly when dealing with external stimuli or avoiding obstacles.
- (3) In high-level pigeons, a general leader will control the overall flight track of the cluster, that is, the absolute positioning and navigation of the cluster are often completed by the general leader pigeon. At the same time, pigeons have different navigation methods in different stages of flight. In the early stage, the geomagnetic field is used to determine the general direction, and the direction is corrected by landmarks in the later stage. The altitude of the sun will also affect its navigation.

Cluster hierarchical positioning architecture based on pigeon flock autonomous behavior

Inspired by the hierarchical behavior and navigation methods of pigeon flocks, we propose a bionic unmanned cluster localization architecture based on the autonomous behavior of pigeon flocks. The cluster uses a hierarchical architecture, and the nodes are divided into follower clusters, core clusters, and the general leader.

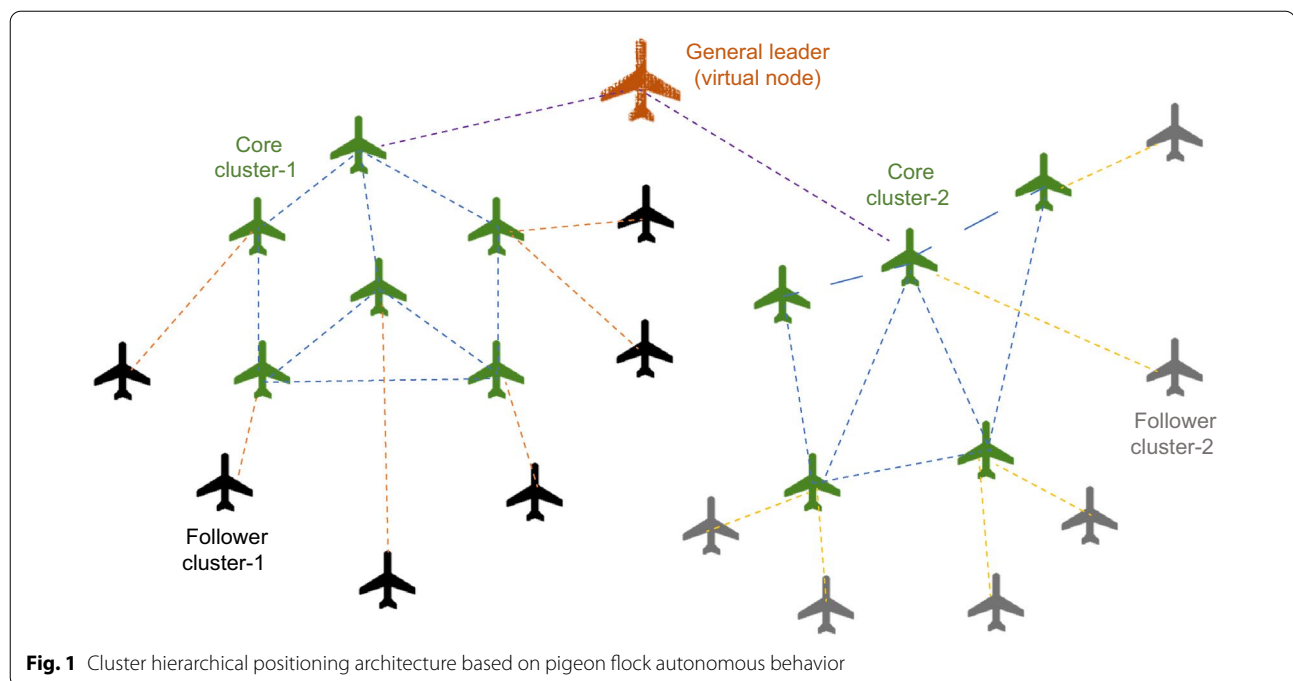


Fig. 1 Cluster hierarchical positioning architecture based on pigeon flock autonomous behavior

The meanings and interrelationships of each level are shown in the Fig. 1.

The spatial relationship between the core cluster and the follower cluster is not a fixed front-to-back formation, and the follower cluster can be distributed around the core cluster. The arrangement here is only to facilitate the description of the overall cluster hierarchical structure. The members of different clusters are marked with different colors. The connections between nodes represent interactions.

A core cluster is like a high-level pigeon in a flock, except that its responsibilities are carried out by a cluster of nodes rather than a single node. The reason is related to its cluster positioning algorithm, and the specific algorithm will be explained in the following sections. The unmanned nodes of the core cluster establish and maintain a stable relative position relationship through mutual measurement and information exchange within the cluster. In the core cluster, all nodes can establish a stable pairwise communication relationship and exchange information. There can be multiple core clusters in a cluster, and connections can also be established between core clusters.

The nodes in the follower cluster are like the low-level pigeons. The nodes in each follower cluster need to be registered in a core cluster (often choose the closest one), and all the follower nodes of a core cluster form a follower cluster. In a follower cluster, a node does not need to interact with other follower nodes to locate itself but uses the core cluster to determine its position, and its

track control and task assignment are also undertaken by the core cluster. The nodes in the core cluster and the follower cluster can be transformed into each other.

The general leader node is equivalent to the general leader pigeon in the pigeon group and at the highest level in the entire cluster. Because the core clusters are at the same level, we need a higher-level role to coordinate all core clusters, and this role also assumes the responsibility of absolute positioning of the entire cluster. However, considering the anti-interference and survivability of the cluster, it is defined as a virtual character in the architecture of this paper. The reason why an entity node is not directly designated as the general leader node is that the attack or interference of the entity general leader node will cause confusion and failure in the entire cluster. This paper does not discuss the implementation of the leader node in detail.

The pigeon flock positioning architecture proposed in this paper adopts a heterogeneous layered networking, which not only has the robustness brought by decentralization, but also is easier to obtain global information than the distributed architecture. At the same time, compared with the traditional pigeon flock architecture, the architecture proposed in this paper has following advantages.

The traditional pigeon group architecture designates one node as the general leader node. The general leader node is in an absolute leadership position, can contribute the most weight to the decision-making of the cluster, and has the most followers. Once the general leader

node fails or is attacked, the entire cluster will be greatly affected. In this paper, the functions of the general leader node, such as absolute positioning and navigation and formation configuration control, are shared by other core clusters to virtualize the general leader. The advanced node network composed of core clusters functions as the general leader, which can significantly improve the anti-damage ability of the entire cluster organization and avoid the overall collapse caused by the failure of the traditional physical general leader node.

The functions of high-level nodes in the traditional pigeon cluster architecture are also realized by a small cluster of UAVs, rather than a single physical node. This cluster is the core cluster mentioned in the article, which brings two benefits: (1) Multiple nodes bring redundancy. Even if some nodes fail, the remaining nodes can still ensure the functional integrity of the core cluster. (2) In traditional methods, a single high-level node needs to carry high-precision inertial sensors to ensure high dead reckoning accuracy, to assist and improve the positioning accuracy of its following nodes. The core cluster composed of multiple nodes can build a stable relative position relationship through cooperative positioning without expensive and cumbersome high-precision inertial navigation devices. In this regard, we propose a cluster relative positioning algorithm based on spatial-temporal correlation information, which is suitable for core clusters. See Sect. [Core cluster localization algorithm](#) for details.

The architecture design considers the positioning accuracy, communication load and system complexity. The core cluster composed of a small number of nodes adopts full connectivity networking mode to obtain the global information within the cluster, and it is easier to obtain the global optimal solution when positioning. We limit the number of the core cluster members to avoid the network communication pressure caused by the full connectivity networking mode. After the core cluster constructs a local relative positioning architecture, it can be used as anchor nodes to broadcast its own position information and ranging signals to the surrounding nodes, and the surrounding nodes calculate their own positions accordingly and become the following nodes of the core cluster. If one-way broadcast is used for positioning between core and follower cluster, theoretically, there is no upper limit for the number of follower nodes and no channel congestion caused by more nodes, and the system capacity can adapt to the large-scale cluster nodes.

Parallel multiple core clusters are used to expand the coverage of clusters and improve flexibility. One single core cluster can only cover a limited range, and multiple

clusters can cover more and control the member number of a single core cluster, relieving the communication pressure brought by the full connecting networking mode. At the same time, more flexible formation configuration can be used among multiple clusters to adapt different task requirements, while single clusters are limited by the coverage of communication equipment and tend to form a huge circle when the number of cluster nodes increases. Multiple core clusters are relatively independent and distributed computing, which reduces the complexity of the overall positioning solution of core clusters. Clusters are calculated independently, which reduces the complexity of the overall clusters' location. Multiple core clusters can also serve more follower nodes, facilitating cluster expansion. By merging coordinate systems, core clusters can form a unified relative coordinate system, which is explained in Sect. [Core cluster localization algorithm](#) (5).

Unmanned cluster observation model

Unmanned vehicles, ships, and other unmanned clusters are often at the same height, so system modeling is often analyzed in two-dimensional space, while unmanned aerial vehicles, underwater detectors, etc. have a higher degree of freedom in movement space and need to be modeled and analyzed in three-dimensional space. Since the height information of nodes can often be obtained through laser altimetry, air pressure/water pressure gauge, etc., the clusters in three-dimensional space can be transformed into those in two-dimensional space by eliminating the height information (Martinelli et al., 2005). To simplify the analysis model, this paper analyzes the cluster localization algorithm in a two-dimensional space.

Assuming that each node constructs a NED coordinate system with itself as the origin, and its north alignment can be achieved by using the strap-down compass. Each node is equipped with inertial sensors capable of measuring angular velocity and acceleration and an onboard communication module, which can measure the distance to other nodes in addition to two-way information exchange with surrounding nodes.

$d_{ij}(t)$ represents the distance between nodes i and j at time t .

$$d_{ij}(t) = d_{ji}(t) = |P_i(t) - P_j(t)| = |(x_i(t) - x_j(t), y_i(t) - y_j(t))| \quad (1)$$

$P_i(t) = (x_i(t), y_i(t))$ represents the coordinates of node i at time t .

The NED coordinate system is established with the position at time $t - 1$ as the origin, and $ins_N(t)$ represents the relative motion vector of node N in this coordinate system from time $t - 1$ to time t , that is, the relative

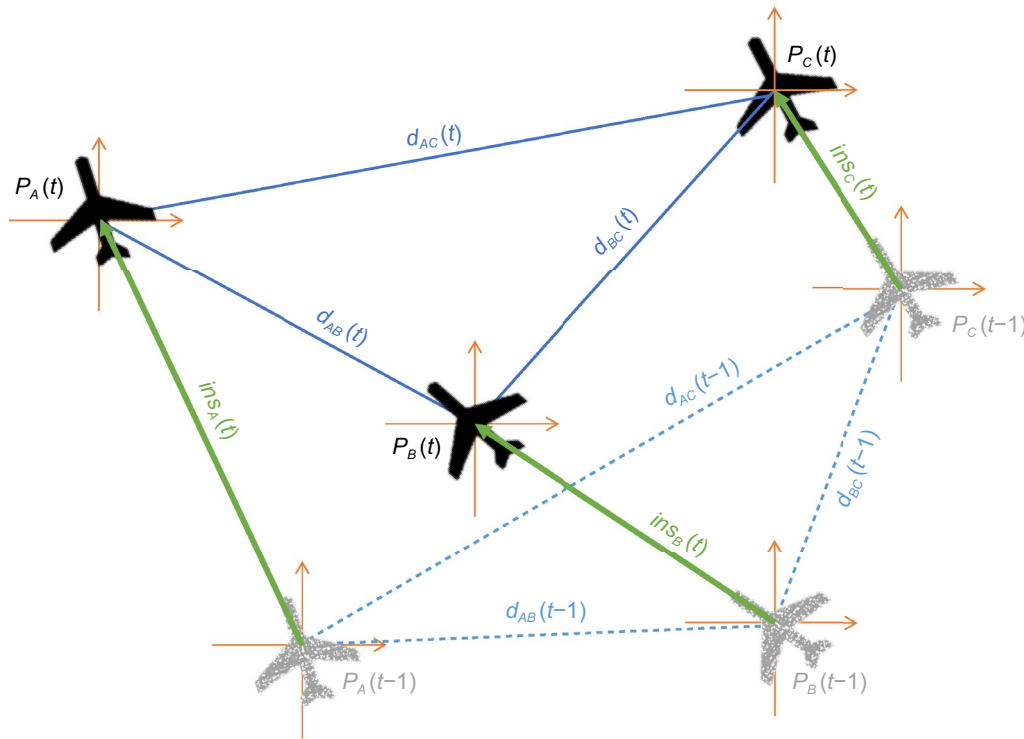


Fig. 2 Observational model of a three-member unmanned cluster

displacement vector obtained by the inertial navigation system track reckoning at the adjacent positioning time.

$$ins_N(t) = \int_{t-1}^t \mathbf{v}_N dt \quad (2)$$

where \mathbf{v}_N represents the vector of node N .

The observation model of the unmanned cluster is shown in the Fig. 2.

The notations used in this section and their meanings are summarized in Table 1.

This paper does not specifically define the network protocol implementation mode of the cluster system, but only makes a general assumption, that is, the nodes complete time synchronization with the cluster while accessing the network. The ranging tasks at the same time are uniformly allocated to adjacent time slots to ensure that

the required ranging information can be completed as quickly as possible, reducing the impact of incomplete synchronization of ranging.

Cluster relative positioning algorithm

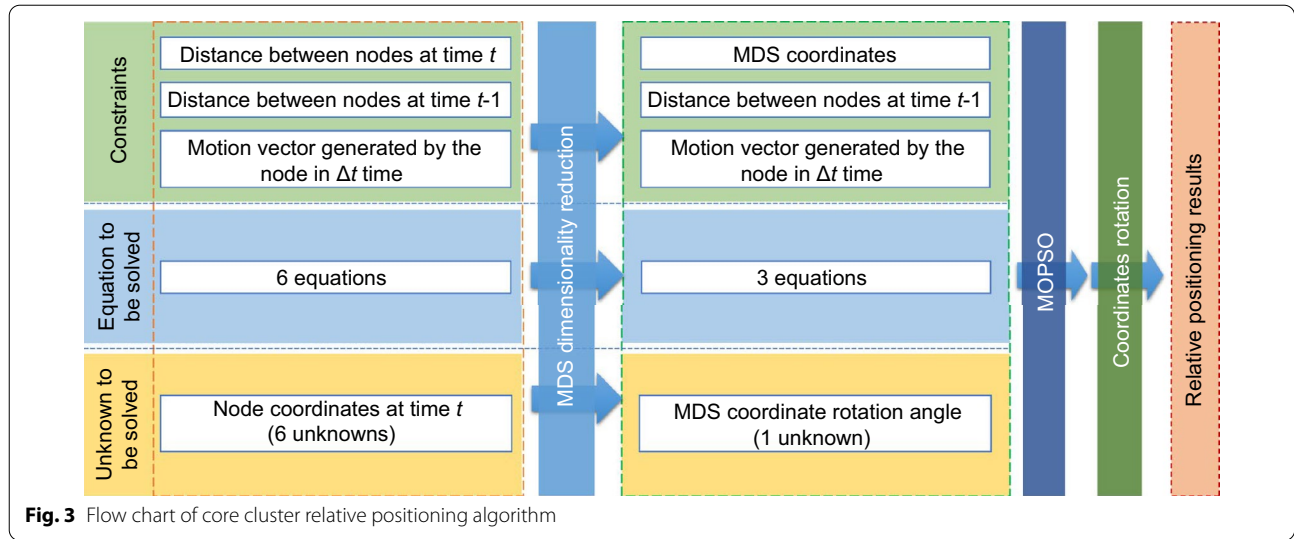
From the positioning architecture proposed in Sect. [Cluster positioning architecture and model](#), cluster positioning can be divided into relative positioning and absolute positioning. Absolute positioning is based on relative positioning and is mainly completed by the general leader node. The relative positioning of the core cluster is completed autonomously, and after the completion, it acts as an anchor node in the positioning of the follower cluster. Therefore, the key to cluster positioning is the relative positioning algorithm of the core cluster. In this regard, this paper proposes a positioning algorithm that uses spatial-temporal correlation information and solves it by the MDS + MOPSO algorithm.

Core cluster localization algorithm

Taking the core cluster composed of 3 nodes as an example, the algorithm process, constraint relationship, objective function, and the number of unknowns of the entire relative positioning are summarized as follows. As shown

Table 1 Notations used in this chapter

Notations	Meaning
$d_{ij}(t)$	Euclidean distance between nodes i and j
$P_i(t)$	Coordinate $(x_i(t), y_i(t))$ of node i at time t
$ins_i(t)$	Relative motion vector of node i from time $t - 1$ to t in NED coordinate system



in the Fig. 3, based on the cluster observation model, we first build the core cluster positioning equations, which are composed of six objective functions. To reduce the difficulty of solution, we use MDS algorithm to lower the dimensions. The unknown parameters are changed from 6 (number of nodes \times 2) to 1 (rotation angle), reducing the search space dimension. At the same time, the objective functions of the equations are changed from 6 to 3. We use the reduced dimension objective function as the fitness function of MOPSO algorithm to solve the rotation angle, and then the rotation matrix can be constructed. Finally, the relative positioning results are obtained by coordinate transformation using the rotation matrix. Next, we will introduce the location algorithm in detail.

Constructing the cluster positioning equation

Based on the cluster observation model, we can give a set of equations for calculating the position of the core cluster consisting of three nodes. By combining the distance observation information of the current and previous moment, the equations are obtained as follows:

$$\begin{aligned}
 d_{AB}(t) &= |P_A(t) - P_B(t)| \\
 d_{AC}(t) &= |P_A(t) - P_C(t)| \\
 d_{BC}(t) &= |P_B(t) - P_C(t)| \\
 d_{AB}(t-1) &= |P_A(t-1) - P_B(t-1)| \\
 d_{AC}(t-1) &= |P_A(t-1) - P_C(t-1)| \\
 d_{BC}(t-1) &= |P_B(t-1) - P_C(t-1)|
 \end{aligned} \quad (3)$$

$P_N(t-1)$ can be calculated from $P_N(t)$ and $ins_N(t)$. So, the equation can be transformed into:

$$\begin{aligned}
 d_{AB}(t) &= |P_A(t) - P_B(t)| \\
 d_{AC}(t) &= |P_A(t) - P_C(t)| \\
 d_{BC}(t) &= |P_B(t) - P_C(t)| \\
 d_{AB}(t-1) &= |(P_A(t) - ins_A(t)) - (P_B(t) - ins_B(t))| \\
 d_{AC}(t-1) &= |(P_A(t) - ins_A(t)) - (P_C(t) - ins_C(t))| \\
 d_{BC}(t-1) &= |(P_B(t) - ins_B(t)) - (P_C(t) - ins_C(t))|
 \end{aligned} \quad (4)$$

In the equation constructed by motion vector and ranging information, the unknown variables are $P_A(t), P_B(t), P_C(t)$. Because the analysis is performed in a two-dimensional space, there are 6 unknowns, and it is not easy to converge to the global optimal solution when solving in a high-dimensional solution space. Therefore, we introduce a multi-dimensional calibration method for dimensionality reduction, which makes it easier for the positioning results to be converged to the optimal solution.

Dimensionality reduction by MDS

The essence of MDS is to map the similarity measures of several analysis objects from a high-dimensional space of unknown dimension to a lower-dimensional space, and fit the similarity between them in the lower-dimensional space (Niu et al., 2010; Yi & Ruml, 2004). Corresponding to unmanned cluster positioning, that is, mapping the Euclidean distance between nodes from the distance measurement dimensional space to the two-dimensional coordinate space, and obtaining the relative coordinates of each node (Chen et al., 2013).

Firstly, the distance matrix D between nodes is constructed with the ranging information between nodes at time t . where $d_{AB}(t)$ is abbreviated as d_{AB} .

$$\mathbf{D} = \begin{bmatrix} 0 & d_{AB} & d_{AC} \\ d_{BA} & 0 & d_{BC} \\ d_{CA} & d_{CB} & 0 \end{bmatrix} \quad (5)$$

$$\mathbf{D}^2 = \begin{bmatrix} 0 & d_{AB}^2 & d_{AC}^2 \\ d_{BA}^2 & 0 & d_{BC}^2 \\ d_{CA}^2 & d_{CB}^2 & 0 \end{bmatrix} \quad (6)$$

Let the coordinates of node N be $P_N(t) = (x_N(t), y_N(t))$, abbreviated as $P_N = (x_N, y_N)$, then $d_{AB}^2 = x_A^2 + y_A^2 + x_B^2 + y_B^2 - 2x_Ax_B - 2y_Ay_B$, $I_N^2 = x_N^2 + y_N^2$, and the matrix \mathbf{R} can be constructed as.

$$\mathbf{R} = \begin{bmatrix} I_A^2 & I_A^2 & I_A^2 \\ I_B^2 & I_B^2 & I_B^2 \\ I_C^2 & I_C^2 & I_C^2 \end{bmatrix} \quad (7)$$

The coordinate matrix \mathbf{X} of the nodes is constructed as.

$$\mathbf{X} = \begin{bmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \end{bmatrix} \quad (8)$$

And $\mathbf{D}^2 = \mathbf{R} + \mathbf{R}^T - 2\mathbf{X}^T\mathbf{X}$.

Dual centralized of \mathbf{D}^2 , and receive a positive definite symmetric matrix \mathbf{B} (Borg & Groenen, 2005), which is.

$$\mathbf{B} = -\frac{1}{2}\mathbf{J}\mathbf{D}^2\mathbf{J} \quad (9)$$

$$\mathbf{J} = \mathbf{E} - n^{-1}\mathbf{I} = \begin{bmatrix} 1 - \frac{1}{n} & -\frac{1}{n} & -\frac{1}{n} \\ -\frac{1}{n} & 1 - \frac{1}{n} & -\frac{1}{n} \\ -\frac{1}{n} & -\frac{1}{n} & 1 - \frac{1}{n} \end{bmatrix} \quad (10)$$

where $n = 3$.

$$\mathbf{B} = -\frac{1}{2}\mathbf{J}(\mathbf{R} + \mathbf{R}^T - 2\mathbf{X}^T\mathbf{X})\mathbf{J} \quad (11)$$

Because of $\mathbf{R}\mathbf{J} = 0$, $\mathbf{J}\mathbf{R}^T = 0$, so

$$\mathbf{B} = -\mathbf{J}\mathbf{X}^T\mathbf{X}\mathbf{J} = \mathbf{V}\mathbf{U}\mathbf{V}^T = \mathbf{V}\sqrt{\mathbf{U}}(\mathbf{V}\sqrt{\mathbf{U}})^T \quad (12)$$

Eigenvalue decomposition of matrix \mathbf{B} , because in two-dimensional space, retain the two largest eigenvalues λ_1, λ_2 and corresponding eigenvectors q_1, q_2 to calculate the 2D coordinates of the node. The relative coordinates of each node after centralization are $\mathbf{J}\mathbf{X}^T = \mathbf{V}\sqrt{\mathbf{U}}$, where $\mathbf{U} = \text{diag}(\lambda_1, \lambda_2)$ and $\mathbf{V} = [q_1, q_2]$.

The coordinates obtained after MDS only represent the distance relationship between nodes, and its coordinate system will change per location calculation, which has no actual physical meaning, and is different from the NED

coordinate system by a rotation angle. Therefore, the original problem of solving 6 unknowns is transformed into solving a rotation angle, and the solution space dimension is significantly reduced.

The coordinates obtained by MDS are represented as $P_N'(t) = (x_N'(t), y_N'(t))$, abbreviated as $P_N' = (x_N', y_N')$, and the relationship between the target $P_N(t)$ can be expressed as follows.

$$\begin{bmatrix} x_N \\ y_N \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} x_N' \\ y_N' \end{bmatrix} \quad (13)$$

where α is the rotation angle to be solved, and the objective function is rewritten as.

$$\begin{aligned} f_1 &= d_{AB}(t-1) = |(R_\alpha P_A'(t) - \text{ins}_A(t)) - (R_\alpha P_B'(t) - \text{ins}_B(t))| \\ f_2 &= d_{AC}(t-1) = |(R_\alpha P_A'(t) - \text{ins}_A(t)) - (R_\alpha P_C'(t) - \text{ins}_C(t))| \\ f_3 &= d_{BC}(t-1) = |(R_\alpha P_B'(t) - \text{ins}_B(t)) - (R_\alpha P_C'(t) - \text{ins}_C(t))| \end{aligned} \quad (14)$$

There is only one unknown variable (the rotation angle α) in the objective function.

Solution using MOPSO

In order to solve this multi-objective function composed of nonlinear equations, we introduce the MOPSO algorithm.

Particle swarm optimization is an evolutionary algorithm, which was originally inspired by the regularity of birds flocking activities, and then used swarm intelligence to establish a simplified model (Shi & Eberhart, 1998). It makes the movement of the whole group in the problem-solving space evolve from disorder to order. The advantage of Particle Swarm Optimization (PSO) is that it is not easy to fall into the local optimal solution, has strong versatility, and can solve complex optimization problems (Marini & Walczak, 2015).

The algorithm randomly distributes a certain number of particles in the feasible region of the problem space, and each particle flies at a certain speed. During the flight, the particle adjusts its own state by combining its current best position and the best position of the population, and then flies to a better area to finally achieve the purpose of searching for the optimal solution (Wei & Li, 2004).

In the single-objective PSO algorithm, since there is only one objective function, the position of the global best particle (g_{best}) and the best position of individual particle (p_{best}) can be uniquely determined simply by comparing their fitness values. In the MOPSO (Reyes-Sierra & Coello, 2006), when selecting p_{best} , if each objective function value of the particle position is optimal, it should be the optimal particle position. If it cannot strictly compare which is better, one of them is randomly selected as the optimal position. Regarding

the selection of g_{best} , there are many non-inferior solutions that can be used as the global optimal one. Therefore, saving these non-inferior solutions and selecting a better one are the core of particle swarm optimization for multi-objective optimization problems. The choice of g_{best} is the core problem of multi-objective particle swarm optimization. Coello and Lechuga (2002) proposed a method, where the objective space is divided into hypercubes, and each hypercube is assigned a fitness value depending on its particles number. The more particles, the less fitness value is. Then roulette-wheel selection is applied on the hypercubes to select one. At the end, g_{best} is randomly selected from this hypercube.

Meanwhile, MOPSO adopts an external repository (name is Archive) to maintain the diversity of the population, Archive stores the non-dominated solution set for each iteration (Mostaghim & Teich, 2003). The algorithm flow is as follows.

- 1) Initialize the particle swarm. Set the population size N , factor parameters, etc. Randomly generate the position X_i and velocity V_i of each particle;
- 2) Divide the target space and calculate the crowding degree according to the number of particles in the grid;
- 3) Calculate the objective function value of the particle. Update the individual optimal position p_{best} of the particle;
- 4) Calculate the non-dominated solution of the population and update the Archive set. If the number of non-dominated solutions exceeds the size of the external repository, random deletion is performed according to the degree of congestion;
- 5) Update the global optimal particle g_{best} ;
- 6) Update the velocity and position of each particle. The particle velocity and position update equations are as follows (Fallah-Mehdipour et al., 2010):

$$v(t+1) = \omega v(t) + c_1 r_1 (p(t) - x(t)) + c_2 r_2 (g(t) - x(t)) \quad (15)$$

$$x(t+1) = x(t) + v(t+1) \quad (16)$$

Among them, ω is the inertia weight; c_1, c_2 are the individual experience coefficient and social experience coefficient, respectively; r_1, r_2 are random numbers in the range $[0, 1]$; $p(t)$ and $g(t)$ are the individual optimal solution and the global optimal solution, respectively.

7) Terminate the program if the termination condition is satisfied, otherwise go to step 3.

Use the objective functions after MDS dimensionality reduction, that is, the equation set finally obtained in Sect. [Core cluster localization algorithm 1\(2\)](#), as the

fitness function of MOPSO, where only the rotation angle α is the unknown to be solved.

Relative coordinates calculation

Finally, target $P_N(t)$ can be obtained by constructing rotation matrix R and MDS coordinate $P_N'(t)$ use Eq. (13).

So far, we have constructed the relative position relationship within the core cluster. The relative coordinate system is the NED coordinate system, which takes the centroid of the core cluster node as the origin.

Summary and discussion

Since MDS algorithm requires the complete ranging information between two nodes to build a distance matrix, it is difficult to ensure this condition when there are many nodes involved. In two-dimensional space, the minimum number of nodes in the system is 3, and the number of ranging between nodes to be maintained is c_n^2 , where n is the number of nodes.

When the fault of individual node in the cluster results in the loss or unreliability of ranging information, the common ideas are: 1) Discard the observations related to this node. 2) Use algorithm to estimate and recover the lost or damaged measurement information, for example, matrix completion algorithm based on norm regularization (Xiao et al., 2015), Multidimensional Scaling Map (MDSMAP) (Shang et al., 2003), etc. 3) Improve the weight of reliable nodes in positioning, such as node reordering and edge reordering algorithms (Hamaoui, 2019). Although ideas 2 and 3 can bring some compensation, they will also introduce errors to some extent. Since the positioning result of the core cluster will affect the following cluster, any error in the core cluster should be avoided as far as possible.

Considering the complexity and reliability of engineering implementation, we suggest that the number of nodes in the core cluster should be limited to about 3–5. More than three will bring some redundancy to the core cluster positioning. When a node does not meet the conditions, it will be discarded with the relevant observations, and the remaining nodes can still ensure the minimum demand.

Another issue worth discussing is the coordinate system merging between multiple core clusters, coordinate conversion can be performed through the common nodes. Compared with the traditional methods (Moore et al., 2004) which need 2 to 3 common points, because the coordinate system constructed by each core cluster in this paper is NED coordinate system, there is no need to consider the rotation and mirror of the coordinate system

in coordinate merging, so only one common point is needed to calculate the required translation.

Follower cluster localization algorithm

Considering that the number of nodes in the follower cluster is significantly larger than that of the core cluster, one should use a localization algorithm that is insensitive to the number of nodes.

For example, with the help of the robust relative coordinate relationship of the core cluster, the core cluster nodes can be used as anchor nodes to broadcast ranging signals and self-position information. After receiving the location information and relative distances of multiple core cluster nodes, the follower cluster node uses the trilateration method to calculate its own location. This algorithm is simple and of good scalability, and not affected by the number of nodes to be located.

In addition to the above algorithm, other methods can also be used to locate the nodes in the follower cluster. Have given the relative coordinates of the core clusters, the positioning of the follower clusters will be easy. The positioning of the follower clusters is not the focus of this paper, so only a brief discussion is given.

Localizability analysis

The localizability is also called observability. Satisfying the localizability condition is the premise to ensure the reliable localization of the cluster. When the cluster state is fully observable, the localization problem has a unique and reliable solution (Arrichiello et al., 2011). Since the formation configuration, motion state, and observation constraints of the cluster will affect the positioning results, it is necessary to clarify the conditions for effective cluster positioning and design a reasonable swarm distribution configuration and motion control strategy to avoid the unlocatable state. Furthermore, the cluster observation information can be optimized through localizability analysis, and redundant observations can be eliminated while ensuring the cluster positioning capability (Frew et al., 2005).

By abstracting the cluster network into an undirected graph consisting of nodes, directions and distance constraints, the cluster localization algorithm proposed in this paper is analyzed by the rigidity theory of graphs. Since the observation error does not affect the cluster localizability, to simplify the analysis process the systematic measurement error is not considered in this analysis.

Abstracted as a mixed direction-distance graph

Rigid Graph is a special network configuration. Intuitively, a rigid graph maintains the stable structure of the entire graph by constraining some edges in the graph (Whiteley, 1996). In cluster localization, rigidity theory studies the relationship between the uniqueness of network node location and network structure from the perspective of graph theory (Eren et al., 2004). At present, some scholars have used graph theory for locatable analysis (Cano & Ny, 2021), but most of them build abstract graphs for a single type of distance/direction observation (Liu et al., 2021; Yang et al., 2013) without considering the constraints between nodes at different times, which has limitations and is difficult to apply to the cluster positioning algorithm proposed in this paper.

First, the cluster observation model is abstracted into an undirected graph composed of distance constraints, direction constraints, and nodes. The specific methods are as follows.

Taking the core cluster consisting of three nodes as an example, due to the joint analysis of spatiotemporal information, the node set V consists of 6 nodes at time t and $t - 1$, $V = \{v_1, v_2, \dots, v_6\}$. The observation information between nodes is abstracted as the edge of the graph, and the set of edges is defined as E , $E = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$. The cluster network can be represented by an undirected connected graph $G = (V, E)$.

There are two types of observations, the ranging information $d_{AB}(t)$ and $d_{AB}(t - 1)$ between nodes at time t and $t - 1$, and the motion vector $ins_A(t)$ of the node from time t to time $t - 1$. The set of distance constraints that define the graph is L , and the direction constraint is D . Then the ranging information can be abstracted into elements in L . The motion vector contains both the distance and direction information between the two nodes. Therefore, a motion vector observation is abstracted as a distance constraint plus a direction constraint. Set $E = \{L, D\}$, use $|N|$ to denote the number of elements in the set, and $|L| = 9$, $|D| = 3$.

The frame F_p represents the mapping pair (G, P) from the abstract set G to R^2 . It corresponds to an implementation of the graph. where $P = (p_1, p_2, \dots, p_6)$ is the mapping of node V in R^2 space, that is, the coordinates of the node in two-dimensional space, and the mapping satisfies the constraints in E .

From the perspective of graph theory, the problem of cluster localizability is whether the unique realization of the graph of a given network structure can be obtained through the distance information and direction

Table 2 Notations used in this section

Notations	Meaning
G	Abstract diagram of cluster
V	Set of all nodes
E	Set of all edges, $E = \{L, D\}$
L	Set of all distance constraints
D	Set of all directional constraints
$ N $	Number of elements in set N
v_i	Node i
L_n	Distance constraint n
D_n	Directional constraint n
p_i	Mapping coordinates of node i in R^2 space
$R(G, P)$	Rigid matrix of frame (G, P)

information between nodes. A sufficient and necessary condition for a cluster to be uniquely localized is that (G, P) is globally rigid, and the problem of cluster localizability is transformed into a global rigidity determination problem based on distance and direction constraints.

The notations used in this section and their meanings are summarized in Table 2.

Rigidity matrix and rigidity analysis

For a direction-length framework (G, P) , we obtain a dot product for each equation of distance constraints and direction constraints (Clinch, 2018). By taking derivatives of these dot products at $t = 0$, we obtain the Jacobian matrix $R(G, P)$, which is the rigidity matrix of (G, P) . The matrix R has $2|V|$ columns and $|E|$ rows, one row for each constraint, and one pair of columns for each node, as shown in the Fig. 4. The row identified by L_n on the left corresponds to the distance constraint, and the row identified by D_n corresponds to the direction constraint. In the column where the corresponding nodes (V_1 and V_2) are located, the pair of columns' elements of the L_n row

are $(p_i - p_j)(p_j - p_i)$, the pair of columns' elements of the D_n row are $(p_i - p_j)^\perp, (p_j - p_i)^\perp$, where $(x, y)^\perp = (y, -x)$.

The abstract rules for the linear dependencies between rows or columns in matrices are described in the theory of matroids proposed by Whitney(1992). Given a matrix M , if the linear combination of row vectors is not zero, then the subset of rows of M is linearly independent.

Looking for the row subset of R whose linear combination is zero, after excluding the case where the nodes are in the same position, we can get the following 3 combinations: (where j, k, m, n are unknowns)

$$(1) \quad L_1 + jL_7 + kL_8 + mD_1 + nD_2 - L_4$$

$$\begin{aligned}
 \Delta x_{AB} + j\Delta x_{AA'} + m\Delta y_{AA'} &= 0 \\
 \Delta y_{AB} + j\Delta y_{AA'} - m\Delta x_{AA'} &= 0 \\
 -\Delta x_{A'B'} - j\Delta x_{AA'} - m\Delta y_{AA'} &= 0 \\
 -\Delta y_{A'B'} - j\Delta y_{AA'} + m\Delta x_{AA'} &= 0 \\
 -\Delta x_{AB} + k\Delta x_{BB'} + n\Delta y_{BB'} &= 0 \\
 -\Delta y_{AB} + k\Delta y_{BB'} - n\Delta x_{BB'} &= 0 \\
 \Delta x_{A'B'} + k\Delta x_{BB'} - n\Delta y_{BB'} &= 0 \\
 \Delta y_{A'B'} + k\Delta y_{BB'} + n\Delta x_{BB'} &= 0
 \end{aligned} \tag{17}$$

After linear combination of the equations, we get:

$$\begin{aligned}
 1 + 5 : j\Delta x_{AA'} + m\Delta y_{AA'} + k\Delta x_{BB'} + n\Delta y_{BB'} &= 0 \\
 2 + 6 : j\Delta y_{AA'} - m\Delta x_{AA'} + k\Delta y_{BB'} - n\Delta x_{BB'} &= 0 \\
 1 + 3 : \Delta x_{AB} - \Delta x_{A'B'} &= 0 \\
 2 + 4 : \Delta y_{AB} - \Delta y_{A'B'} &= 0
 \end{aligned} \tag{18}$$

After further derivation:

$$\begin{aligned}
 \Delta x_{AA'} &= \Delta x_{BB'} \\
 \Delta y_{AA'} &= \Delta y_{BB'} \\
 m + n &= 0 \\
 j + k &= 0
 \end{aligned} \tag{19}$$

	v_A		v_B		v_C		$v_{A'}$		$v_{B'}$		$v_{C'}$	
L_1	Δx_{AB}	Δy_{AB}	$-\Delta x_{AB}$	$-\Delta y_{AB}$	0	0	0	0	0	0	0	0
L_2	Δx_{AC}	Δy_{AC}	0	0	$-\Delta x_{AC}$	$-\Delta y_{AC}$	0	0	0	0	0	0
L_3	0	0	Δx_{BC}	Δy_{BC}	$-\Delta x_{BC}$	$-\Delta y_{BC}$	0	0	0	0	0	0
L_4	0	0	0	0	0	0	$\Delta x_{A'B'}$	$\Delta y_{A'B'}$	$-\Delta x_{A'B'}$	$-\Delta y_{A'B'}$	0	0
L_5	0	0	0	0	0	0	$\Delta x_{A'C'}$	$\Delta y_{A'C'}$	0	0	$-\Delta x_{A'C'}$	$-\Delta y_{A'C'}$
L_6	0	0	0	0	0	0	0	0	$\Delta x_{B'C'}$	$\Delta y_{B'C'}$	$-\Delta x_{B'C'}$	$-\Delta y_{B'C'}$
L_7	$\Delta x_{AA'}$	$\Delta y_{AA'}$	0	0	0	0	$-\Delta x_{AA'}$	$-\Delta y_{AA'}$	0	0	0	0
L_8	0	0	$\Delta x_{BB'}$	$\Delta y_{BB'}$	0	0	0	0	$-\Delta x_{BB'}$	$-\Delta y_{BB'}$	0	0
L_9	0	0	0	0	$\Delta x_{CC'}$	$\Delta y_{CC'}$	0	0	0	0	$-\Delta x_{CC'}$	$-\Delta y_{CC'}$
D_1	$\Delta y_{AA'}$	$-\Delta x_{AA'}$	0	0	0	0	$-\Delta y_{AA'}$	$\Delta x_{AA'}$	0	0	0	0
D_2	0	0	$\Delta y_{BB'}$	$-\Delta x_{BB'}$	0	0	0	0	$-\Delta y_{BB'}$	$\Delta x_{BB'}$	0	0
D_3	0	0	0	0	$\Delta y_{CC'}$	$-\Delta x_{CC'}$	0	0	0	0	$-\Delta y_{CC'}$	$\Delta x_{CC'}$

Fig. 4 Rigidity matrix $R(G, p)$

$$(2) \quad L_1 + jL_7 + kL_8 - L_4$$

$$\begin{aligned} \Delta x_{AB} + j\Delta x_{AA'} &= 0 \\ \Delta y_{AB} + j\Delta y_{AA'} &= 0 \\ -\Delta x_{A'B'} - j\Delta x_{AA'} &= 0 \\ -\Delta y_{A'B'} - j\Delta y_{AA'} &= 0 \\ -\Delta x_{AB} + k\Delta x_{BB'} &= 0 \\ -\Delta y_{AB} + k\Delta y_{BB'} &= 0 \\ \Delta x_{A'B'} - k\Delta x_{BB'} &= 0 \\ \Delta y_{A'B'} - k\Delta y_{BB'} &= 0 \end{aligned} \quad (20)$$

After linear combination of the equations, we get:

$$\begin{aligned} 1 + 5 : j\Delta x_{AA'} + k\Delta y_{BB'} &= 0 \\ 2 + 6 : j\Delta y_{AA'} + k\Delta x_{BB'} &= 0 \\ 1 + 3 : \Delta x_{AB} - \Delta x_{A'B'} &= 0 \\ 2 + 4 : \Delta y_{AB} - \Delta y_{A'B'} &= 0 \end{aligned} \quad (21)$$

After further derivation:

$$\begin{aligned} \Delta x_{AA'} &= \Delta x_{BB'} \\ \Delta y_{AA'} &= \Delta y_{BB'} \\ j + k &= 0 \end{aligned} \quad (22)$$

$$(3) \quad L_1 + mD_1 + nD_2 - L_4$$

$$\begin{aligned} \Delta x_{AB} + m\Delta y_{AA'} &= 0 \\ \Delta y_{AB} - m\Delta x_{AA'} &= 0 \\ -\Delta x_{A'B'} - m\Delta y_{AA'} &= 0 \\ -\Delta y_{A'B'} + m\Delta x_{AA'} &= 0 \\ -\Delta x_{AB} + n\Delta y_{BB'} &= 0 \\ -\Delta y_{AB} - n\Delta x_{BB'} &= 0 \\ \Delta x_{A'B'} - n\Delta y_{BB'} &= 0 \\ \Delta y_{A'B'} + n\Delta x_{BB'} &= 0 \end{aligned} \quad (23)$$

After linear combination of the equations, we get:

$$\begin{aligned} 1 + 5 : m\Delta y_{AA'} + n\Delta y_{BB'} &= 0 \\ 2 + 6 : -m\Delta x_{AA'} - n\Delta x_{BB'} &= 0 \\ 1 + 3 : \Delta x_{AB} - \Delta x_{A'B'} &= 0 \\ 2 + 4 : \Delta y_{AB} - \Delta y_{A'B'} &= 0 \end{aligned} \quad (24)$$

After further derivation:

$$\begin{aligned} \Delta x_{AA'} &= \Delta x_{BB'} \\ \Delta y_{AA'} &= \Delta y_{BB'} \\ m + n &= 0 \end{aligned} \quad (25)$$

So, it is known that when $ins_A(t) = ins_B(t)$, the rank of matrix \mathbf{R} is 11, that is, there are 11 linearly independent observations.

In the same way, $ins_A(t) = ins_C(t)$ can be deduced from $(L_2 + jL_7 + kL_9 + mD_1 + nD_3 - L_5), (L_2 + jL_7 + kL_9 - L_5), (L_2 + mD_1 + nD_3 - L_5), ins_B(t) = ins_C(t)$ can be deduced from $(L_3 + jL_8 + kL_9 + mD_2 + nD_3 - L_6), (L_3 + jL_8 + kL_9 - L_6), (L_3 + mD_2 + nD_3 - L_6)$.

Since any non-empty frame can be translated in two-dimensional space, the rank of the rigid matrix \mathbf{R} of a rigid frame is $2|V| - 2$, i.e. guaranteeing a unique solution requires $2|V| - 2$ independent constraints, or requires $2|V| - 2$ equivalent spanning constraints (Jackson & Keavash, 2011). Knowing that $|V| = 6$, to ensure the rigidity of the frame \mathbf{F}_p , the rank of the rigid matrix \mathbf{R} needs to be 10. From the above inference, when any pair of motion vectors in $ins_A(t), ins_B(t), ins_C(t)$ are equal, the rank of \mathbf{R} is 11, then (\mathbf{G}, \mathbf{P}) is redundant rigidity. When the three motion vectors are all equal, the rank of \mathbf{R} is 9, and the frame \mathbf{F}_p is non-rigid, that is, the localization solution of the cluster cannot be uniquely determined.

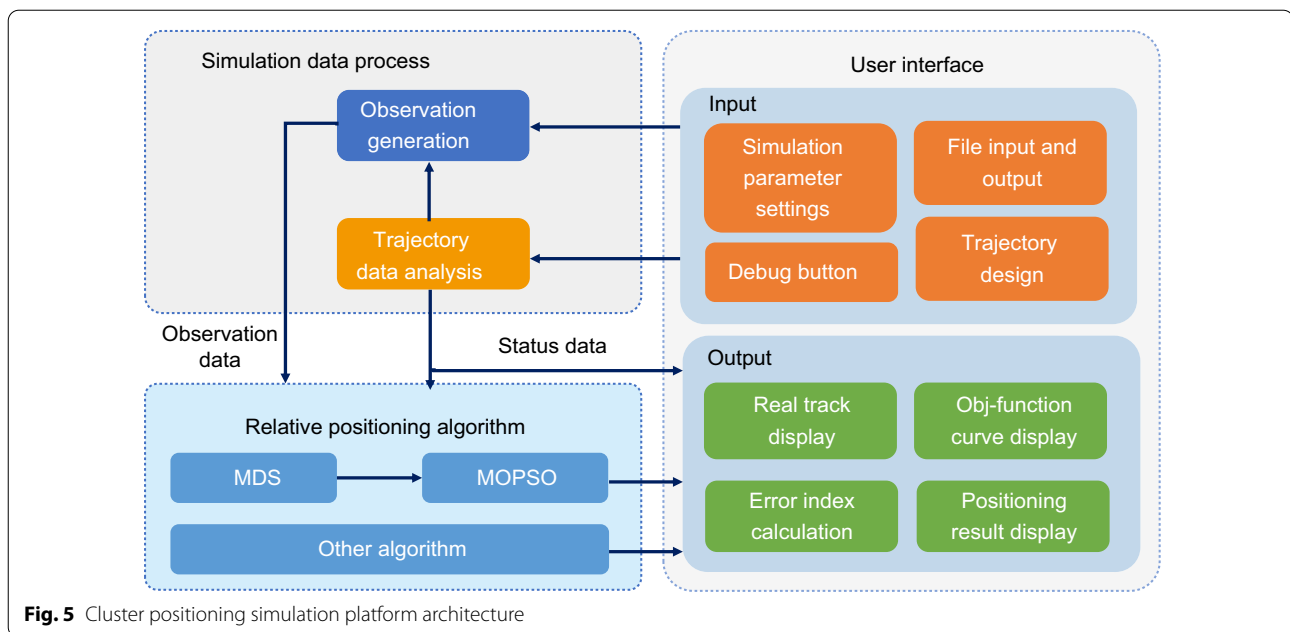
Results

Rough the above analysis, to ensure the localizability of the cluster relative localization algorithm proposed in this paper, the situation of same motion vectors of multiple nodes at the same time should be avoided. When the rank of the rigid matrix \mathbf{R} is less than $2|V| - 2$, the clusters are not localizable. As a special case, when all nodes remain stationary, the motion vector is zero, and the localizability condition cannot be satisfied either.

Experiment

To verify the localizability and positioning accuracy of the proposed cluster positioning algorithm, we designed a cluster positioning simulation software and conducted localizability tests and positioning accuracy tests in different scenarios. Through the simulation software, the motion trajectory of the cluster nodes is designed first, and the trajectory is imported into the analysis unit to generate the status information of the nodes at each moment. Further modeling and simulation of the observations are performed to obtain gyroscope and accelerometer measurements, magnetic compass measurements, and wireless ranging values. The cluster positioning algorithm unit uses the observation data to calculate the relative positioning result. The UI interface of the software can easily display and debug the simulation and help evaluate the results.

In the localizability simulation, we selected typical cluster motion scenarios for verification, such as parallel formation (constant velocity), parallel formation (non-constant velocity), cross-line formation, circular formation, and collinear formation (non-constant velocity). In the positioning accuracy simulation, we choose a three-node circular motion cluster scene to analyze the



accuracy of the core cluster relative positioning algorithm proposed in this paper and compare it with the traditional pure inertial navigation-based and EKF-based relative positioning algorithms.

Simulation platform

The architecture of the software simulation platform includes a user interface, a simulation data processing unit, and a positioning solution unit. The user interface includes input function panels such as parameter setting, file input/output, motion trajectory design, debugging buttons, and output function panels such as positioning result display and trajectory display. The simulation data processing unit can analyze the trajectory data and generate the original observation data. The relative positioning algorithm unit performs positioning operations according to the obtained observation information and state information and can select different positioning algorithms. The simulation platform can verify cluster positioning capability under various task scenarios. The overall architecture is shown in Fig. 5.

The soft interface of the simulation platform is shown in the Fig. 6.

Localizability simulation

In this section, we test the localizability of the proposed algorithm in different scenarios, prove the algorithm adaptability to multiple scenarios, and confirm the

conclusion of the localizability analysis in Sect. [Localizability analysis](#).

Simulation scene

In the scenario design, we selected several typical formation scenarios for verification, including parallel formation (constant velocity), parallel formation (non-constant velocity), cross-line formation, circular formation, collinear formation (non-constant velocity) and random sports formation. The various motion trajectories are shown in the Fig. 7. Different nodes in the figure are marked with different colors, and the start and end points of the trajectory are also marked.

Simulation parameters

Observation noise is not added to the simulation to avoid interference with the localizability analysis. The simulation interval is 1 s, and the output frequency of the positioning result is 1 Hz. The number of cluster nodes is three, and they are always at the same height.

Evaluation indicators

We choose the objective function curve and the two-dimensional positioning result plan to judge the localizability. By drawing the objective function curve participating in the MOPSO solution, we can judge whether there is a solution to the objective function equations. The two-dimensional positioning result

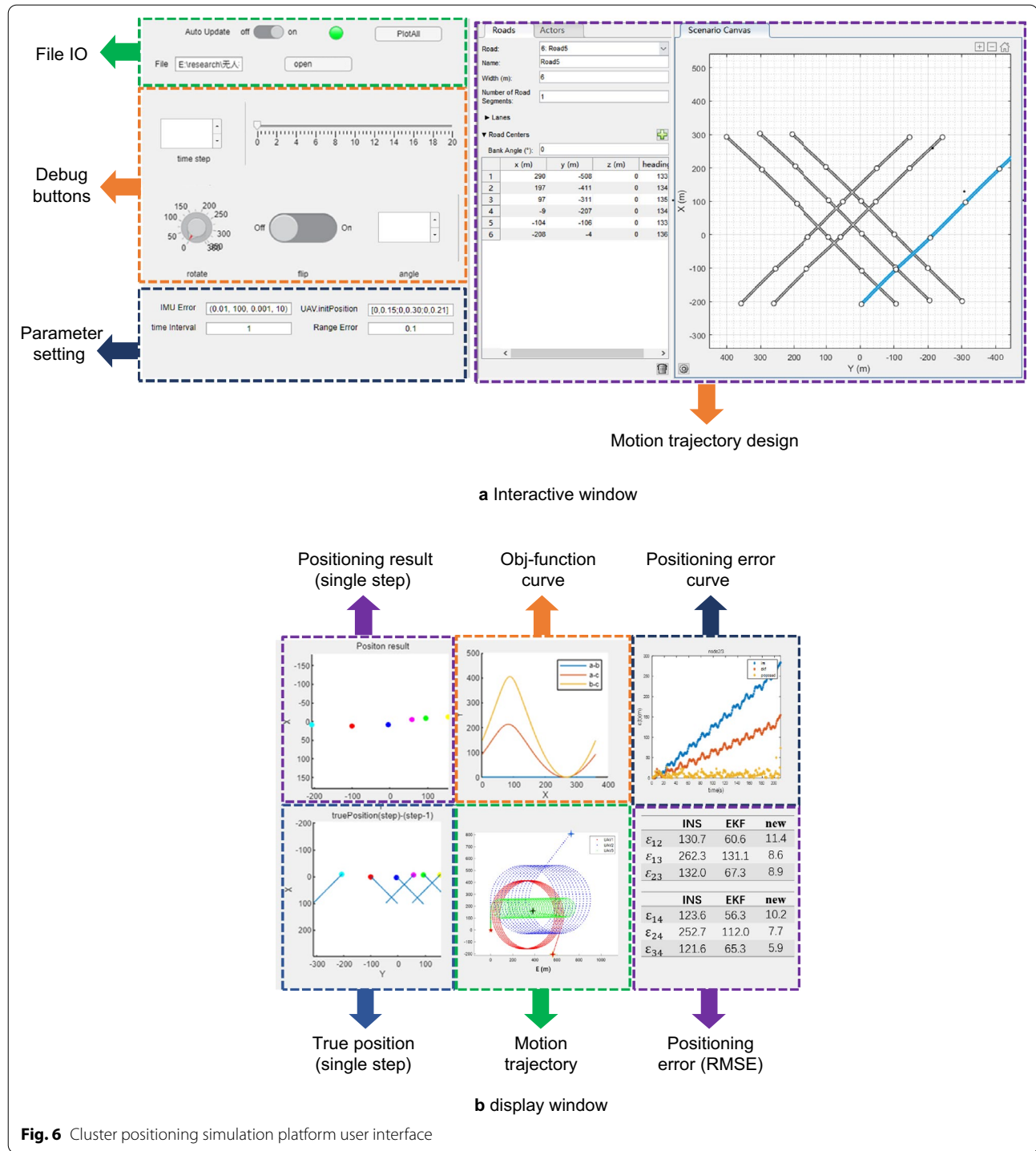


Fig. 6 Cluster positioning simulation platform user interface

plan is compared with the real position plan to verify if the relative positional relationship between nodes is consistent.

Simulation results

Limited to the length of the article, we only list the simulation results at time $t = 1$ in each scenario, and the complete results can be found in this url: <https://github.com/>

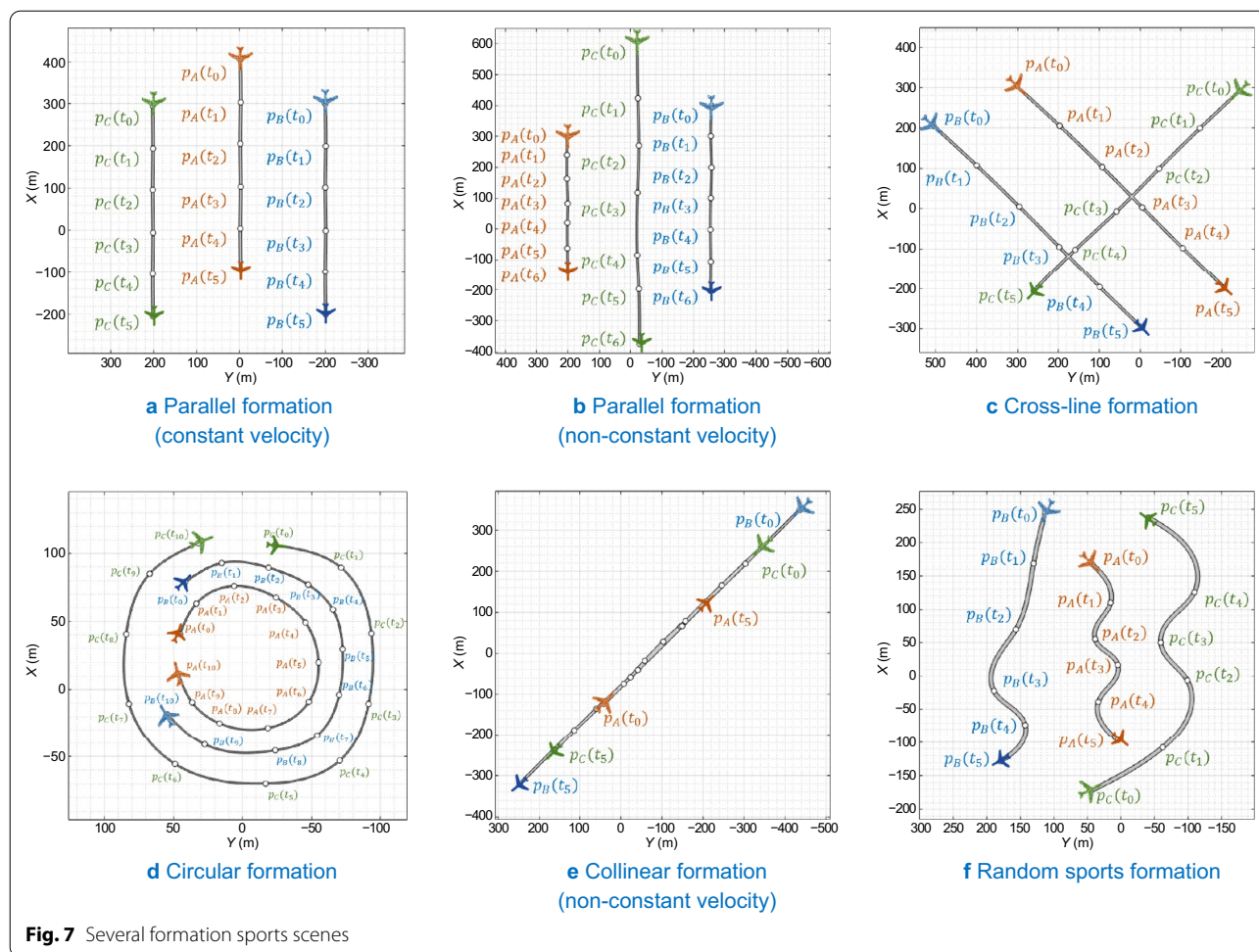


Fig. 7 Several formation sports scenes

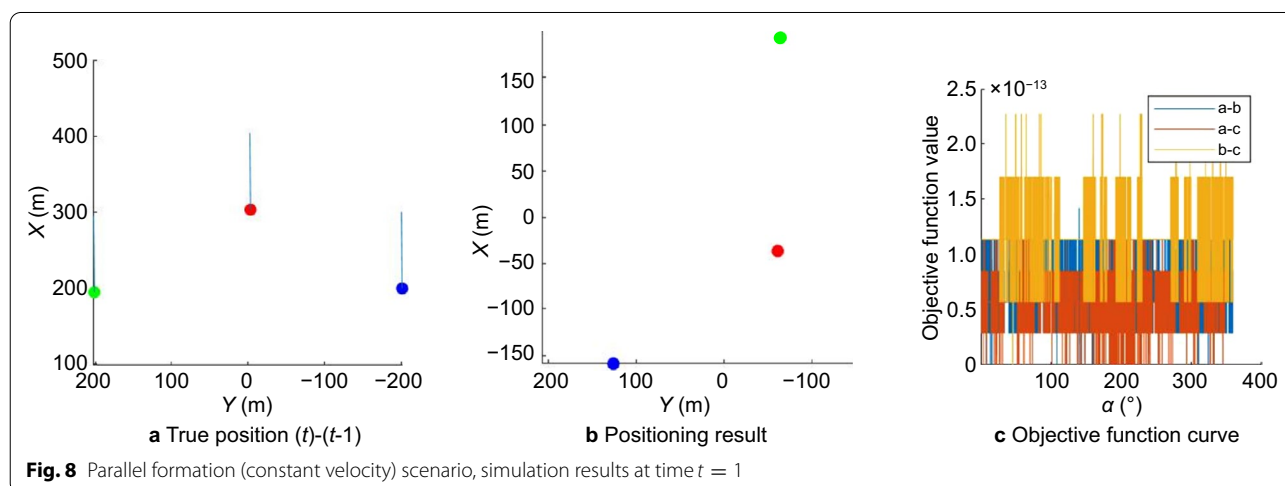
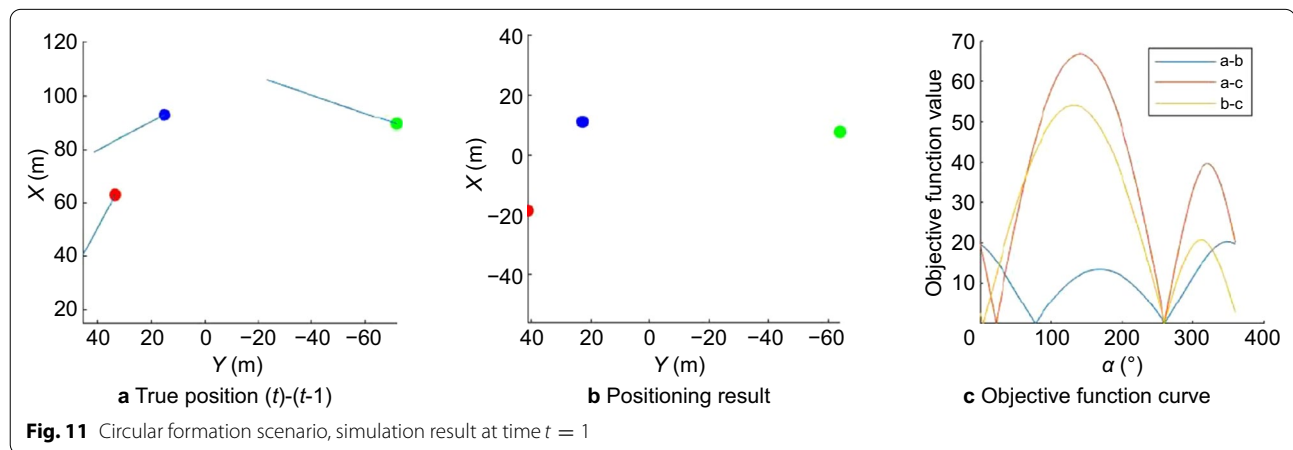
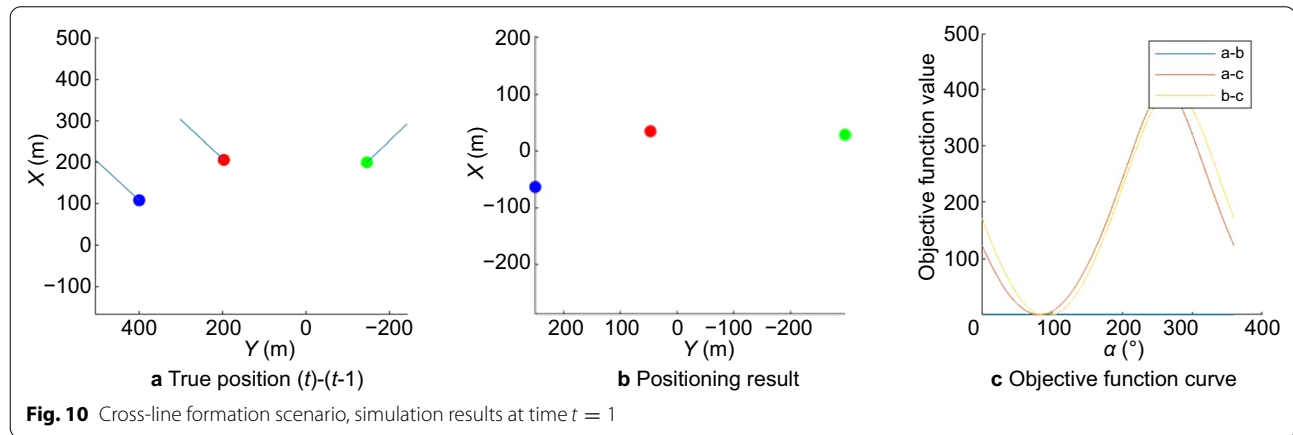
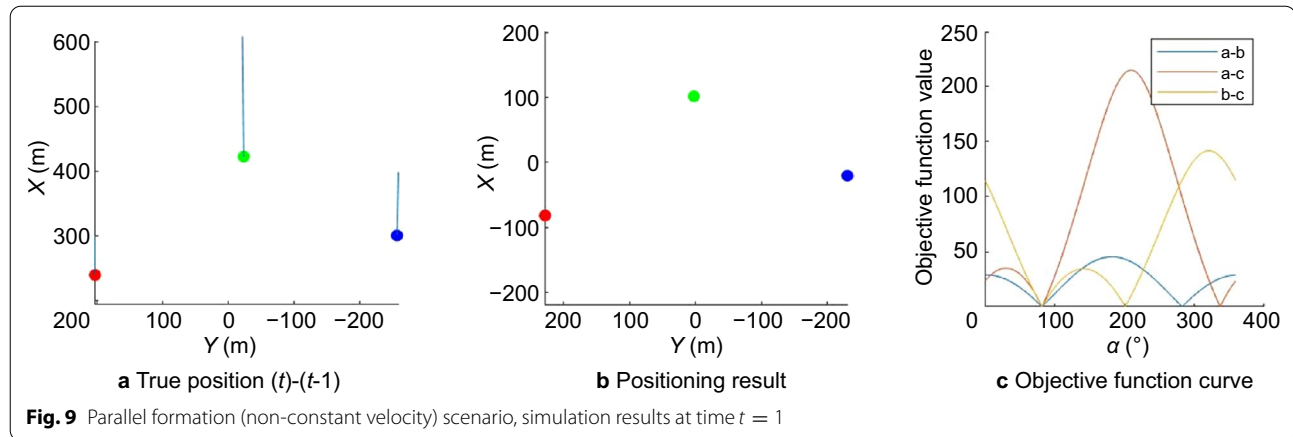


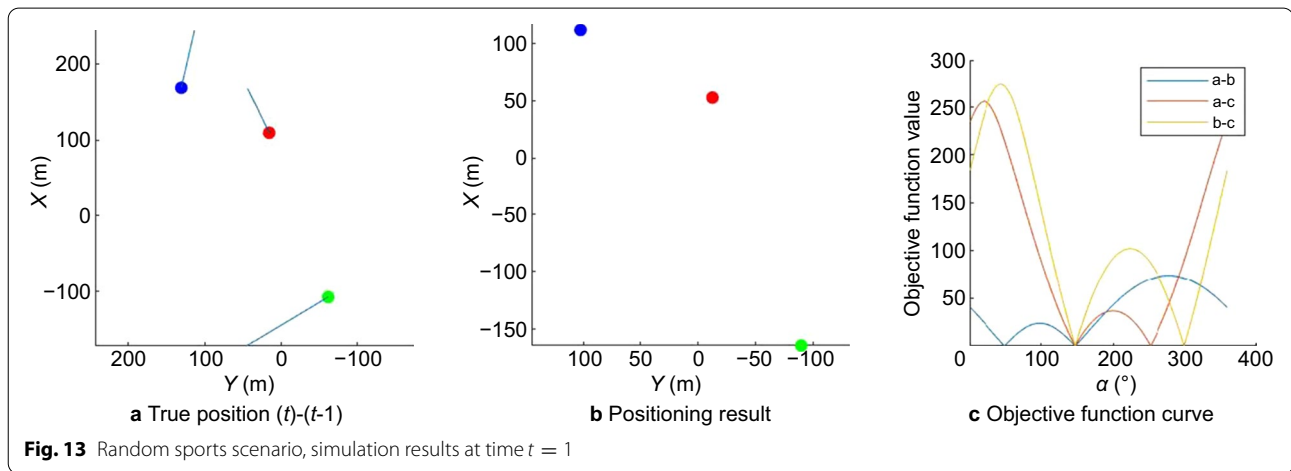
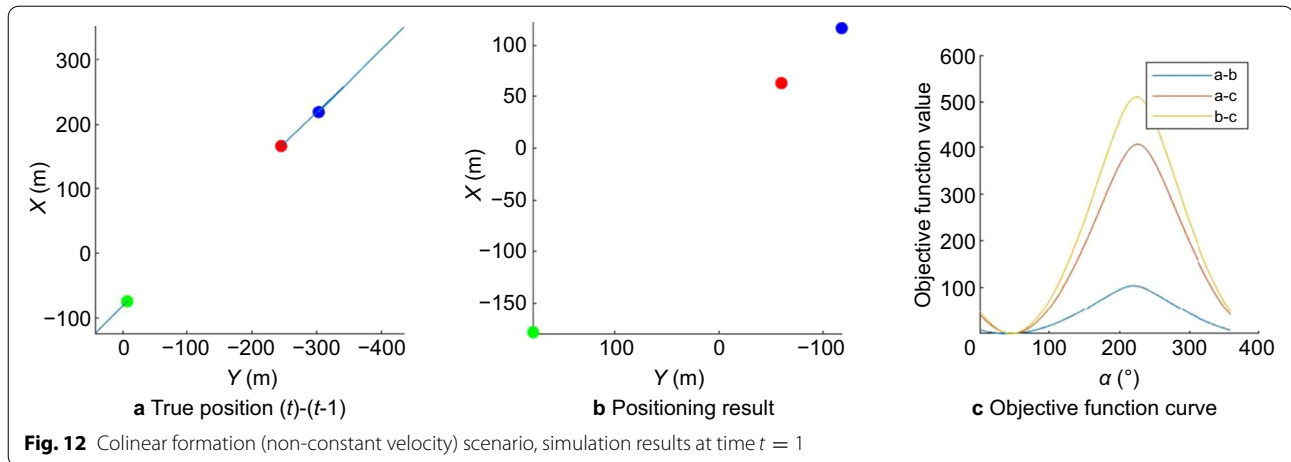
Fig. 8 Parallel formation (constant velocity) scenario, simulation results at time $t = 1$



[chgmqh/relative-positioning-simulation-results](#) (Figs. 8, 9, 10, 11, 12, 13).

From the simulation results, one can see the two-dimensional position result in the first scenario is quite different from the real position, and the objective function curve is disorganized. It suggests that the proposed

algorithm cannot effectively locate in this scenario. In other scenarios, the positioning is successful according to the objective function curve graph and the two-dimensional positioning results. The simulation results can corroborate the conclusion in Sect. [Localizability analysis](#).



Liu analyzed the locatable conditions of the mainstream EKF collaborative location algorithm based on ranging and IMU (Liu, 2015). Compared with the algorithm based on EKF, the proposed algorithm has more relaxed localizable conditions and can adapt to more scenarios, as shown in Table 3.

Positioning accuracy simulation

This experiment conducts a comparative analysis and verification of the performance of cluster relative positioning algorithm. The information required for positioning is obtained by using the motion trajectory simulation and observation volume generation functions of the simulation platform. In addition to the relative positioning algorithm proposed in this paper, the EKF positioning algorithm based on ranging and IMU and the relative positioning algorithm based on pure Inertial Navigation System (INS) are selected for comparison.

Simulation scene

We designed a cluster scene consisting of three nodes. They start from the origin and perform circular motions of different radii after a short acceleration motion. The simulation time is 210 s, and the five-pointed star and the cross star represent the starting and ending points, respectively. The motion trajectory of the node is shown in the Fig. 14.

Simulation parameters

The measured values of the inertial device and the ranging device are all obtained by applying the error model to the true value. The constant bias of the gyroscope is $0.01^\circ/\text{h}$, the random walk error is $0.001^\circ/\sqrt{\text{h}}$, the constant bias of the accelerometer is $100\mu\text{g}$, and the random walk error is $10\mu\text{g}/\sqrt{\text{Hz}}$. The sampling frequency of the inertial device is 10 Hz. The wireless ranging accuracy is 0.1 m, and the sampling frequency is 1 Hz. The output frequency of the positioning result is 1 Hz.

Table 3 Algorithm localizability condition comparison

	Colocalization algorithm based on EKF	Proposed algorithm
Unlocatable case (Two nodes as an example)	(1) Two nodes move at the same speed along a linear path parallel to each other (2) Two nodes move along the same straight path	Two nodes move in the same direction and speed

Evaluation indicators

The relative positioning error evaluation index is to calculate the modulus of the difference between the relative position of the two nodes under the positioning result and the real coordinates. The formula is expressed as follows

$$\varepsilon_{AB}(t) = |P_{AB} - P'_{AB}| \quad (26)$$

P_{AB} and P'_{AB} respectively represent the relative positions of the two nodes under the real coordinates and the positioning result.

$$\begin{aligned} P_A &= (x_A, y_A) \\ P_B &= (x_B, y_B) \\ P_{AB} &= (x_A - x_B, y_A - y_B) \\ P'_A &= (x'_A, y'_A) \\ P'_B &= (x'_B, y'_B) \\ P'_{AB} &= (x'_A - x'_B, y'_A - y'_B) \end{aligned} \quad (27)$$

At the same time, the Root Mean Square Error (RMSE) of the positioning is calculated to evaluate the overall positioning accuracy.

$$\text{RMSE}_{AB} = \frac{\sqrt{\sum_{t=1}^N \varepsilon_{AB}(t)}}{N} \quad (28)$$

Simulation results

The comparison of relative positioning accuracy over time between INS, EKF-based collaborative algorithm, and the proposed algorithm is shown in Fig. 15.

From the simulation results, the relative positioning error of the INS accumulates with time, showing a divergent state. The relative positioning algorithm based on EKF can reduce some positioning errors, but only suppress the speed of error divergence. Compared with the previous two methods, the proposed algorithm does not have the problem of error accumulation over time, and the overall positioning accuracy is higher. The Root Mean Square Error (RMSE) indicators in Table 4 of the three positioning algorithms can also confirm this.

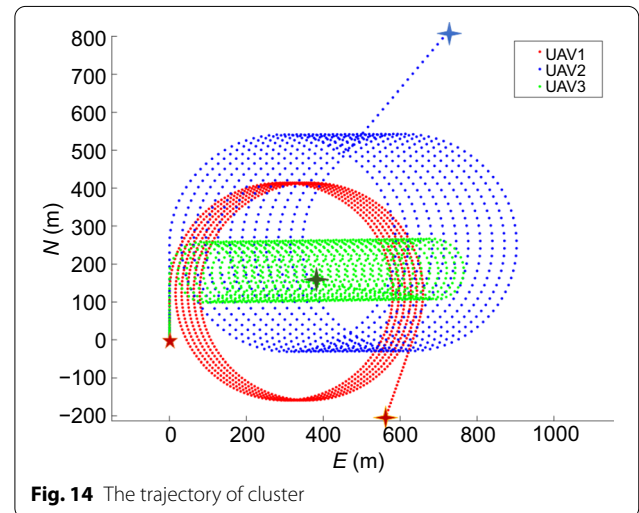
Time-consuming simulation

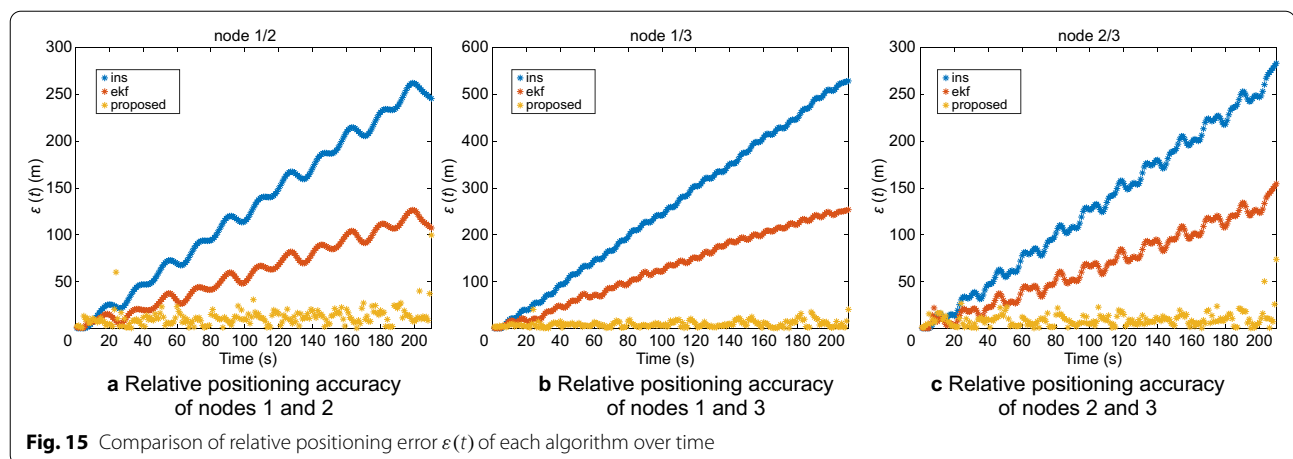
Although the simulation is not a real-time demonstration system, we can still estimate the single shot time consumption by dividing the total operation time by the number of simulation steps.

The simulation platform uses Lenovo PC, i5-10400 CPU, 8G memory, runs simulation program we designed, and calculates the execution time. Taking a 3 nodes core cluster as an example, the number of simulation steps is 5 and the simulation interval is 1 s. The relative location algorithm proposed in this paper takes 1.16 s in total, and 0.23 s in a single shot, which can fully meet the requirements of 3 Hz real-time positioning. However, the operation speed of EKF algorithm is obviously faster, and the single shot location takes about 0.024 s.

Conclusion and future work

Aiming at the problems in the localization of moving time-varying cluster networks, this paper proposes a hierarchical unmanned cluster localization architecture based on the concept of pigeon flock bionics. The design of virtual general leader + core cluster + follower cluster is adopted to realize a flexible and scalable unmanned cluster positioning system. At the same time, a new relative positioning algorithm based on spatial-temporal correlation observation for unmanned clusters is proposed, which only relies on mutual ranging and inertial measurement within the cluster for positioning

**Fig. 14** The trajectory of cluster



calculation. No external signal source is needed, which improves the deployment flexibility and system robustness of the cluster positioning system. The MDS algorithm is introduced to reduce the dimension of solution space and simplify the objective function as much as possible, which can shorten the subsequent calculation time and make it easier for the optimization search algorithm to find the optimal solution. The introduced MOPSO algorithm is suitable for solving multi-objective function problems and hardly falls into the local optimal solution.

We introduce the mixed graph rigidity theory to analyze the localizability of clusters, which can analyze abstract graphs with both distance and direction constraints. At the same time, we abstract the inertial navigation data as a mixed constraint of direction and distance, which expands the application object of localizability theoretical analysis based on graph theory. Therefore, graph theory can now be used to analyze the localizability of the abstract maps constructed by continuous observation. In this paper, the method is applied to analyze the locatable conditions of the proposed relative location algorithm and verified by simulation.

To verify the performance and localizability of the cluster location algorithm, we designed a cluster location simulation software, which can generate simulation data

and display the location results. Based on this, we carried out localizability simulation and positioning accuracy simulation under different scenarios.

The simulation results show that compared with the traditional pure INS algorithm and EKF based algorithm, the proposed algorithm has better positioning accuracy in long run and produces the positioning error not diverging over time, which is applicable to the tasks under the condition of satellite rejection. The proposed algorithm cannot be located if the two nodes have the same direction and speed of movement, and the locatable constraints are more relaxed, which can adapt to more cluster motion scenes.

Of course, the proposed algorithm also has some limitations. The time consumption of the algorithm is improved compared with the traditional algorithm, but it can still meet the real-time positioning requirements of larger than 3 Hz; Because the complete distance information between all nodes is required, the number of nodes participating in core cluster location is limited, otherwise it will bring a large communication load. It is necessary to ensure the communication between two nodes, which limits the spatial distribution of nodes.

The future works are summarized as follows:

- (1) This paper initially proposes a hierarchical unmanned cluster positioning architecture design based on the concept of pigeon flock bionics, but only provides specific algorithm implementation and simulation analysis for the core cluster level. Further research should be carried out on the leader node and follower cluster, and more detailed analysis and algorithm implementation should be given. We will carry out the research on the leader node and follower cluster, and give more detailed analysis and algorithm implementation. This includes, but is

Table 4 Comparison of RMSE index of three relative positioning algorithms

Node pair	RMSE / m		
	INS	EKF	Proposed
ε_{12}	130.76	60.63	11.45
ε_{13}	262.30	131.13	8.60
ε_{23}	132.06	67.38	8.96

not limited to, the virtualization method, selection strategy, and absolute positioning method of the leader node, the positioning method and the anchor point selection strategy of the follower cluster.

- (2) The core cluster level currently only analyzes the positioning of a single cluster and needs further study on the positioning and coordinate system merging algorithms between multiple clusters.
- (3) Improve the simulation platform to support a complete cluster positioning architecture. Carry out a cluster positioning simulation experiment under the leader node + core cluster + follower cluster architecture to evaluate its absolute positioning and relative positioning capabilities.
- (4) On the basis of the simulation test, carry out the test based on the unmanned car platform to verify the effectiveness of the cluster positioning algorithm.

Abbreviations

MDS: Multidimensional scaling; MDSMAP: Multidimensional scaling map; PSO: Particle swarm optimization; MOPSO: Multiple objective particle swarm optimization; EKF: Extended Kalman filter; UAV: Unmanned aerial vehicles; GNSS: Global navigation satellite system; RMSE: Root mean square error; IMU: Inertial measurement unit; UWB: Ultra-wide band; NED: North east down; INS: Inertial navigation system.

Acknowledgements

The authors would like to express thanks to Ruixin Xue for the help in the paper's revisions and Huayu Yuan, Wenju Su, Jianmin Zhao for the help in the implementation of experiments.

Author contributions

Conceptualization, ZD and HQ; Methodology, ZD and HQ; Software HQ; Validation, HQ, RW, and EH; Formal analysis, CW; Investigation, HQ, RW; resources, CW; Data curation, RW; Writing—original draft preparation, HQ; writing—review and editing, HQ; visualization, HQ; supervision, ZD; project administration, ZD and HQ; funding acquisition, EH, CW All authors have read and agreed to the published version of the manuscript.

Funding

Science and Technology on Complex System Control and Intelligent Agent Cooperative Laboratory foundation (201101).

Availability of data and materials

The datasets generated and/or analysed during the current study are not publicly available due the foundation requirements but are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹Beijing University of Posts and Telecommunications, Beijing 100876, China.

²Science and Technology on Complex System Control and Intelligent Agent Cooperative Laboratory, Beijing 100074, China.

Received: 2 September 2022 Accepted: 15 November 2022

Published online: 02 January 2023

References

- Arrichiello, F., Antonelli, G., Aguiar, A. P., & Pascoal, A. (2011). Observability metric for the relative localization of AUVs based on range and depth measurements: Theory and experiments. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 25–30 Sept. 2011, San Francisco, CA, USA.
- Borg, I., & Groenen, P. J. (2005). *Modern multidimensional scaling: Theory and applications*. Springer New York, NY.
- Cano, J., & Ny, J. L. (2021). Improving ranging-based location estimation with rigidity-constrained CRLB-based motion planning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 30 May–5 June 2021, Xi'an China.
- Chen, H. B., Wang, D. Q., Yuan, F., & Xu, R. (2013). A MDS-based localization algorithm for underwater wireless sensor network. *2013 OCEANS - San Diego*, 1–5.
- Chen, D., & Gao, G. X. (2019). Probabilistic graphical fusion of LiDAR, GPS, and 3D building maps for urban UAV navigation. *Navigation*, 66(1), 151–168. <https://doi.org/10.1002/navi.298>
- Clinch, K. (2018). Global rigidity and symmetry of direction-length Frameworks. Ph.D. Thesis, Queen Mary University of London, UK.
- Coello, C. A. C., & Lechuga, M. S. (2002). MOPSO: a proposal for multiple objective particle swarm optimization. Proceedings of the 2002 Congress on evolutionary computation. CEC'02 (Cat. No.02TH8600). 12–17 May, 2002, Honolulu, HI, USA
- Conte, G., & Doherty, P. (2008). An integrated UAV Navigation system based on aerial image matching. In *2008 IEEE Aerospace Conference*. 1–8 March 2008, Montana, USA.
- Doriya, R., Mishra, S., & Gupta, S. (2015, 2015/05). A brief survey and analysis of multi-robot communication and coordination In *International conference on computing, communication & automation*, <http://dx.doi.org/https://doi.org/10.1109/CCAA.2015.7148524>
- Eren, T., Goldenberg, O. K., Whiteley, W., Yang, Y. R., Morse, A. S., Anderson, B. D. O., & Belhumeur, P. N. (2004). In *Rigidity, computation, and randomization in network localization*. IEEE INFOCOM. 7–11, March 2004, Hong Kong, China.
- Fallah-Mehdipour, E., Haddad, O. B., & Mariño, M. A. (2010). MOPSO algorithm and its application in multipurpose multireservoir operations. *Journal of Hydroinformatics*, 13(4), 794–811. <https://doi.org/10.2166/hydro.2010.105>
- Fan, Y., Qi, X., Li, B., & Liu, L. (2020). Fast clustering-based multidimensional scaling for mobile networks localisation. *IET Communications*, 14(1), 135–143. <https://doi.org/10.1049/iet-com.2019.0444>
- Frew, E., Dixon, C., Argrow, B., & Brown, T. (2005). Radio source localization by a cooperating UAV team. In *Infotech@Aerospace*. 26–29 September 2005, Arlington, Virginia, USA. <https://arc.aiaa.org/doi/abs/10.2514/6.2005-6903>.
- Garnier, S., Gautrais, J., & Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1), 3–31. <https://doi.org/10.1007/s11721-007-0004-y>
- Gautam, A., & Mohan, S. (2012). A review of research in multi-robot systems. In *2012 IEEE 7th International conference on industrial and information systems (ICIIS)*. 6–9 August 2012, Chennai, India. <https://doi.org/10.1109/ICIIS.2012.6304778>
- Gyagenda, N., Hatilima, J. V., Roth, H., & Zhmud, V. (2022). A review of GNSS-independent UAV navigation techniques. *Robotics and Autonomous Systems*, 152, 104069. <https://doi.org/10.1016/j.robot.2022.104069>
- Hamaoui, M. (2019). Non-iterative MDS method for collaborative network localization with sparse range and pointing measurements. *IEEE Transactions on Signal Processing*, 67(3), 568–578. <https://doi.org/10.1109/TSP.2018.2879623>
- Jackson, B., & Keevash, P. (2011). Necessary conditions for the global rigidity of direction-length frameworks. *Discrete & Computational Geometry*, 46(1), 72–85. <https://doi.org/10.1007/s00454-011-9326-z>
- Kumar, S., Kumar, R., & Rajawat, K. (2016). Cooperative localization of mobile networks via velocity-assisted multidimensional scaling. *IEEE Transactions on Signal Processing*, 64(7), 1744–1758. <https://doi.org/10.1109/TSP.2015.2507548>

- Liu, Y. (2015). *Optimized Algorithm of Multi-AUV Cooperative Navigation System and Design of Its Formation Configuration*. Ph.D. Thesis, Harbin Engineering University, China.
- Liu, Q., Liu, R., Wang, Z., & Thompson, J. S. (2021). UAV swarm-enabled localization in isolated region: a rigidity-constrained deployment perspective. *IEEE Wireless Communications Letters*, 10(9), 2032–2036. <https://doi.org/10.1109/LWC.2021.3091215>
- Luo, Q., & Duan, H. (2017). Distributed UAV flocking control based on homing pigeon hierarchical strategies. *Aerospace Science and Technology*, 70, 257–264. <https://doi.org/10.1016/j.ast.2017.08.010>
- Marini, F., & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 149, 153–165. <https://doi.org/10.1016/j.chemolab.2015.08.020>
- Martinelli, A., Pont, F., & Siegwart, R. (2005). Multi-Robot Localization Using Relative Observations. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. 18–22 April 2005, Barcelona, Spain.
- Masini, R. (2016). *Multirobot Localization Using Heuristically Tuned Extended Kalman Filter*. Masters' thesis, Université d'Ottawa/University of Ottawa, Canada.
- Moore, D., Leonard, J., Rus, D., & Teller, S. (2004). Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 3–5 November 2004, Baltimore, MD, USA.
- Mostaghim, S., & Teich, J. (2003). Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*. SIS'03 (Cat. No.03EX706), 26–26 April 2003, Indianapolis, IN, USA.
- Niu, D., Guan, B., Zhou, W., & Yu, D. (2010). 3D localization method based on MDS-RSSI in wireless sensor network. In *2010 IEEE International Conference on Intelligent Computing and Intelligent Systems*. 22–24 Oct. 2010, Guilin, China.
- Panzieri, S., Pascucci, F., & Setola, R. (2006). Multirobot Localisation Using Interlaced Extended Kalman Filter. In *2006 IEEE/RSJ International conference on intelligent robots and systems*. 9–15 October 2006, Beijing, China. <https://doi.org/10.1109/IROS.2006.282065>
- Reyes-Sierra, M., & Coello, C. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3), 287–308.
- Shamaei, K., & Kassas, Z. (2019). *Sub-Meter Accurate UAV Navigation and Cycle Slip Detection with LTE Carrier Phase Measurements*. In *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*. 16–20 September 2019, Miami, Florida, USA. <https://doi.org/10.33012/2019.17051>
- Shang, Y., Ruml, W., Zhang, Y., & Fromherz, M. P. (2003). Localization from mere connectivity. *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. 1–3 June 2003, Annapolis, Maryland, USA.
- Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE International conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No.98TH8360)*, 4–9 May 1998, Anchorage, AK, USA.
- Strohmeier, M., Walter, T., Rothe, J., & Montenegro, S. (2018). Ultra-wideband based pose estimation for small unmanned aerial vehicles. *IEEE Access*, 6, 57526–57535. <https://doi.org/10.1109/ACCESS.2018.2873571>
- Tahir, A., Böling, J., Haghighat, M.-H., Toivonen, H. T., & Plosila, J. (2019). Swarms of unmanned aerial vehicles—A survey. *Journal of Industrial Information Integration*, 16, 100106. <https://doi.org/10.1016/j.jii.2019.100106>
- Wei, Y., & Li, Q. (2004). Survey on particle swarm optimization algorithm. *Engineering Science*, 6(5), 87–94.
- Whiteley, W. (1996). Some matroids from discrete applied geometry. *Contemporary Mathematics*, 197, 171–312.
- Whitney, H. (1992). On the Abstract Properties of Linear Dependence. In J. Eells & D. Toledo (Eds.), *Hassler Whitney Collected Papers* (pp. 147–171). Birkhäuser Boston. https://doi.org/10.1007/978-1-4612-2972-8_10
- Wu, C.-F., Cheng, J., Guo, X.-Y., Xu, C., Hu, E.-W., & Qi, H. (2022). Development and Prospect of Aircraft Clusters Cooperative Positioning and Navigation Countermeasures Technology. *Yuhang Xuebao/Journal of Astronautics*, 43(2), 131–142. <https://doi.org/10.3873/j.issn.1000-1328.2022.02.001>
- Xiao, F., Sha, C., Chen, L., Sun, L., & Wang, R. (2015). Noise-tolerant localization from incomplete range measurements for wireless sensor networks. In *2015 IEEE Conference on computer communications (INFOCOM)*. 26 April–1 May 2015, Hong Kong, China.
- Yang, Z., Wu, C., Chen, T., Zhao, Y., Gong, W., & Liu, Y. (2013). Detecting outlier measurements based on graph rigidity for wireless sensor network localization. *IEEE Transactions on Vehicular Technology*, 62(1), 374–383. <https://doi.org/10.1109/TVT.2012.2220790>
- Yi, S., & Ruml, W. (2004). Improved MDS-based localization. In *IEEE INFOCOM*, 7–11 March 2004, Hong Kong, China.
- Yoo, C.-S., & Ahn, I.-K. (2003). Low cost GPS/INS sensor fusion system for UAV navigation. *Digital Avionics Systems Conference, 2003. DASC '03. The 22nd. IEEE*. <https://doi.org/10.1109/DASC.2003.1245891>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)