IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 8, 2020, accepted June 24, 2020, date of publication July 2, 2020, date of current version July 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006479

An Optimal Guidance Strategy for Moving-Target Interception by a Multirotor Unmanned Aerial Vehicle Swarm

XI WANG, GUANZHENG TAN, YUSI DAI, FANLEI LU, AND JIAN ZHAO
School of Automation, Central South University, Changsha 410083, China

Corresponding author: Guanzheng Tan (tgz@csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61403422 and Grant 61973320.

ABSTRACT To solve the problem of intercepting a moving target by a multirotor unmanned aerial vehicle (UAV) swarm, an optimal guidance strategy is proposed. The proposed guidance law is based on the integration of the classic pure pursuit guidance law and Kuhn–Munkres (KM) optimal matching algorithm, and virtual force potential functions are used to avoid collision. The proposed optimal guidance strategy is demonstrated by simulation experiments. The simulation results indicate that with the proposed optimal guidance strategy, a UAV swarm can intercept a moving target while maintaining the predetermined formation, and during the formation flight, the collisions between UAVs or the target can be avoided. Through a comparative experiment, the proposed optimal matching algorithm is proven to significantly reduce the average per-sampling-period total flight distance of all the UAVs and accelerate the interception process, and the formation completion degree is improved.

INDEX TERMS Optimal matching, unmanned aerial vehicles, pursuit algorithms, three-dimensional guidance law, target interception, collision avoidance.

I. INTRODUCTION

Currently, with the advancement of computer control technology, unmanned aerial vehicles (UAVs) have developed greatly. In contrast to the remote manipulation used in the early stages, most UAVs can now fly autonomously. They are exerting an increasingly prominent effect on many areas related to everyday life, such as disaster relief [1], aerial photography, security tracking, goods transportation, and agricultural plant protection.

Due to the limited function of a single UAV, some complex tasks cannot be performed. Therefore, many scholars have conducted research on cooperative tasks of UAVs, and related research has attained great development in recent years. For example, in [2], a UAV swarm is used to build a queuing network, and the length distribution of the shortest paths for cube and sphere formations of UAV swarms is determined. In [3], nPSO is applied to solve the problem of finding the optimal positions of UAVs that form a swarm in the presence of large obstacles such as buildings in an urban environment. In [4], a UAV swarm is used for reconnaissance missions, and distributed PSO is applied to path planning. In [5], the authors propose a new mobility model using the Ant Colony algorithm combined with chaotic dynamics (CACOC) to enhance the coverage in area exploration by a swarm of UAVs. In [6], a UAV swarm is used to track a target. In [7], the optimal deployment and movement of multiple UAVs is studied, and an accurate formula for the total amount of UAV movement that guarantees the best time-averaged performance is determined. In [8], user-centric UAV swarm networks are designed, and UAV swarm diversity is increased. In [9], based on the UAV swarm. a deployment algorithm of flying base stations is proposed.

Sometimes, when unidentified flying objects intrude into a no-fly zone, interception cannot be completed by a single UAV. In this case, an intuitive strategy is to employ a UAV swarm to intercept the intruder, and a guidance strategy is required for the UAV swarm. In [10], a multitarget guidance strategy is proposed, which is used to guide multiple missiles to attack a stationary target simultaneously. In [11], the application scenario of [10] is extended, and constraints on acceleration are added. A three-dimensional multitarget cooperative

The associate editor coordinating the review of this manuscript and approving it for publication was Moayad Aloqaily.

guidance law is developed from the two-dimensional guidance law in [12]. And in [13], [14] the authors use virtual forces to form some specific topologies and on-demand coverage for UAV swarms. In addition, the problem of guiding multiple missiles to attack a target simultaneously is studied in [15] and [16].

In the aforementioned studies, the guidance strategies are mainly used formation flight of missiles or fixed-wing aircraft [17], [18]. The difference between multirotor UAV guidance and fixed-wing aircraft or missile guidance is that a fixed-wing aircraft or missile is subject to nonholonomic constraints, while a multirotor UAV is capable of full degree-of-freedom flight. For our application scenario, it is beneficial to adopt a multirotor UAV swarm to intercept the intruder. For instance, when a hostile unidentified flying object invades our airspace, we can send a UAV swarm to surround or intercept it with a specific formation. Alternatively, if birds approach an airport, a UAV swarm can drive the birds away with a specific formation. The main differences between our research and the containment control problem are in our research UAVs need to maintain the preset formation as much as possible during the flight, and the correspondences between the leaders and followers are optimized.

To solve the collision avoidance problem in a multi-objective environment, many scholars have performed many studies, such as those in [19]–[21]. In these studies, the virtual force-based method is frequently used, the key point of which is to construct a potential function based on distance and position. In brief, the virtual repulsive force increases with decreasing distance between two objects, and this principle can be used in collision avoidance [22]. In [23], an estimation of the control force for network connectivity preservation and collision avoidance is proposed. In [24], [25], an enhanced virtual force algorithm is proposed, which improves the accuracy of the virtual force model. In [26], the repulsive forces between vehicles or objects and the environment can be obtained according to the potential function, and an application of distance-based collision avoidance is implemented. An alternative approach is to use constrained optimization, where collisions are explicitly regarded as spatial constraints [27]. In [28], a virtual force-based collective motion algorithm is proposed for the self-adaptive collective motion of swarm robots, which can avoid collision among members while decreasing the moving distance.

The objective of this paper is to study the problem of intercepting a moving target using a UAV swarm. A key issue of this problem is to design the guidance law [29]. At present, the classic guidance laws primarily fall under the following three categories: pure pursuit, deviated pursuit, and parallel or proportional pursuit [30]. The pure-pursuit guidance law is simple and efficient and only needs to measure the position of the target, so it is widely utilized in related works [31]; the other two methods are more difficult to implement because they require complicated measurement of the target, but multirotor UAVs have lower loads and are unable to carry heavy precision measuring equipment. However, for the scenario of this paper, the pure-pursuit guidance law is not perfectly suitable. Aiming at addressing this problem, an enhanced three-dimensional guidance law for controlling the trajectories of the UAVs will be designed and demonstrated in this paper.

Our main contributions are as follows:

1. Based on the pure-pursuit guidance law, an enhanced guidance law is proposed to guide a UAV swarm to form a specific formation and intercept a moving target. In previous studies, the target is usually assumed to be in a state of uniform rectilinear motion [16], [29], [32]. However, in our study, the target is allowed to be in a state of variable-velocity curvilinear motion. To eliminate the time delay and reduce the measuring error of the observation of the target position, a Kalman filter is used to estimate the position of the target during the interception process. During the interception process, the concept of virtual leader UAV is introduced to maintain the formation, and a formation strategy is designed according to the flight characteristics of multirotor UAVs. In addition, a virtual force-based algorithm is proposed for collision avoidance.

2. Based on graph theory, combined with Kuhn–Munkres (KM) and Hungarian algorithms, a novel algorithm is designed to optimally match UAVs and their flight destinations. Through this algorithm the average per-sampling-period total flight distance of all the UAVs is significantly reduced, the interception process is accelerated, and the formation completion degree is improved, where the "formation completion degree" is used to evaluate the difference between the preset formation and the actual formation. The smaller the difference, the higher the formation completion degree.

II. PROBLEM FORMULATION

In the application scenario we set, we find that compared with a fixed-wing UAV, a multirotor UAV has better low-speed performance, a greater number of degrees of freedom, and better hover ability. Therefore, the following studies are conducted based on the characteristics of a multirotor UAV. Multirotor UAVs will be referred to as UAVs in what follows.

Assumption 1: When an unidentified flying object intrudes into a no-fly zone, a UAV swarm will be sent to intercept it. All UAVs may launch from different initial positions.

Assumption 2: The unidentified flying object is treated as the target, which can be detected by the cameras carried by the UAVs, and the distances between it and UAVs can be measured by the laser distance sensors carried by the UAVs, then its position can be obtained.

Assumption 3: The number of UAVs in the swarm is set according to the task demands, which is denoted as N.

Assumption 4: For each UAV, its spatial position information is obtained via an onboard GPS.

Assumption 5: The sampling periods of sensors for the measurement of the position and orientation of each UAV and the target position are uniformly set to T. In this case, the target position and the motion information of the UAVs will be obtained at the beginning of each sampling period.

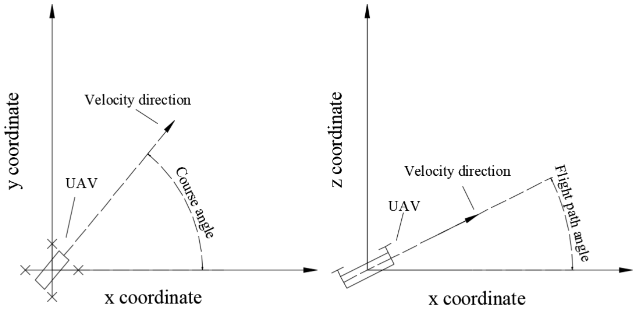


FIGURE 1. Definitions of course angle and flight path angle.

In this paper, the suffix (nT) in a variable means that this variable represents the state at the beginning of the n-th sampling period, where ($n = 1, 2, \ldots$). And the superscript i in a variable means that this variable belongs to the i-th UAV, where ($i = 1, 2, \ldots, N$).

For example, the position of the target at the beginning of the n-th sampling period is denoted as $\boldsymbol{p}_T(nT) \in R^3$, which is measured by the UAVs. And the actual position of the i-th UAV at the beginning of the n-th sampling period is denoted as $\boldsymbol{p}_U^i(nT)$.

Assumption 6: To simplify the motion model and facilitate calculations, the control of the UAVs is implemented perfectly, and the response time is zero, this assumption is widely applied in the related works [29], [33], [34]. And during each sampling period, the UAVs are in a state of uniform rectilinear motion, their average flight speeds along the line of sight are denoted as $\boldsymbol{V}_U$. Whose direction is given by the flight path angle μ and the course angle φ. The definitions of these two angles are shown in Fig.1. And the kinematic model of a UAV is given by Equation (1).

$$\boldsymbol{p}_U^i[(n+1)T] = \boldsymbol{p}_U^i(nT) + [\Delta x^i(nT), \Delta y^i(nT), \Delta z^i(nT)]$$
$$\begin{cases} \Delta x^i(nT) = V_U^i(nT)T \cos \mu^i(nT) \cos \varphi^i(nT) \\ \Delta y^i(nT) = V_U^i(nT)T \cos \mu^i(nT) \sin \varphi^i(nT) \\ \Delta z^i(nT) = V_U^i(nT)T \sin \mu^i(nT) \end{cases} \tag{1}$$

where Δx^i, Δy^i, Δz^i respectively represent the projections of the displacement vector on the three coordinate axes *x, y,* and *z* of a UAV during a sampling period.

Angles μ and φ are the absolute flight path angle and the absolute course angle of a UAV relative to the ground coordinate system. The variation values of them between the current sampling period and the previous sampling period are considered as the guidance inputs, which are denoted as $\Delta\mu$ and $\Delta\varphi$ as shown in Equation (2). By controlling these two angles and another guidance input $\boldsymbol{V}_U$, the trajectory of this UAV can be determined according to the proposed guidance law.

$$\begin{cases} \Delta\mu^i(nT) = \mu^i(nT) - \mu^i[(n-1)T] \\ \Delta\varphi^i(nT) = \varphi^i(nT) - \varphi^i[(n-1)T] \end{cases} \tag{2}$$

Assumption 7: The flight destinations of the UAVs are assigned by a centralized mechanism. The control center will calculate the optimal matching and the optimal guidance strategy at the beginning of each sampling period, and then send them to the UAVs.

Assumption 8: Under all circumstances, the maximum speed of the target should be smaller than the maximum speed of a UAV. Otherwise, the UAVs will never catch up with the target.

Assumption 9: There are no obstacles in the environment. Collisions between UAVs or between UAVs and the target are considered, and to avoid the collision, a safe distance should be kept. The safe distance will be set according to the experimental environment.

Assumption 10: The communication range and sensing range of each UAV can cover the entire no-fly zone. UAVs can upload the position of the target and themselves to the control center through the mobile wireless sensor network (MWSN). And in this area, the target can be detected by every UAV.

III. THREE-DIMENSIONAL GUIDANCE STRATEGY

The three-dimensional guidance law is used to guide the UAVs to their flight destinations in such a way that the UAV swarm can intercept the target within an appropriate time.

In this paper, we propose a guidance law based on the pure-pursuit guidance law. As shown in Fig. 2, a virtual leader UAV (referred to as virtual leader) is introduced, which is used as the position reference point in the formation of the UAV swarm, and its kinematic model is the same as the real UAV. According to the application scenario, the relative positions between UAVs in the formation and between UAVs and virtual leader are predetermined. So that the flight destinations of UAVs at the beginning of each sampling period can be obtained, and the UAV swarm can fly according to the preset formation except some UAVs cannot arrive at the preset flight destinations on time or collisions have occurred between some UAVs.

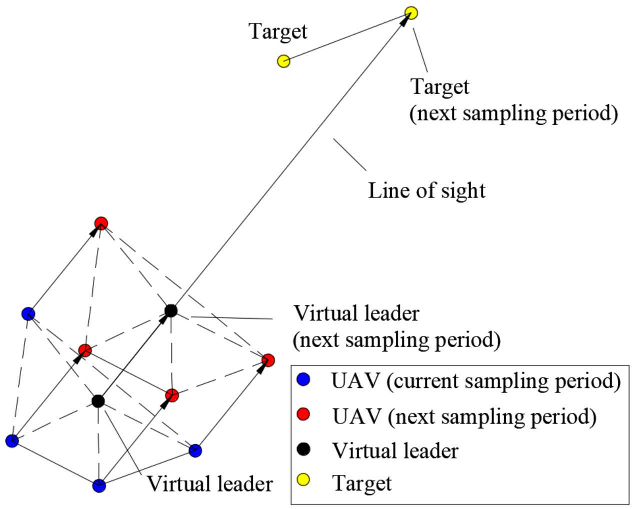


FIGURE 2. Geometric description of the three-dimensional guidance law.

A. TARGET POSITION ESTIMATION

In this paper, a Kalman filter is used to estimate the position of the target and reduce measuring error. Assuming that the noise of measuring data follows the Gaussian distribution, the whole system can be described as a linear Gaussian system. The motion equation of the target is given as follows:

$$\begin{cases} \boldsymbol{p}_T[(n+1)T] = \boldsymbol{p}_T(nT) + \boldsymbol{v}_T(nT)T + \dfrac{1}{2}\boldsymbol{a}_T(nT)T^2 \\ \boldsymbol{v}_T[(n+1)T] = \boldsymbol{v}_T(nT) + \boldsymbol{a}_T(nT)T \end{cases} \tag{3}$$

where vectors $\boldsymbol{p_T} \in R^3$, $\boldsymbol{v_T} \in R^3$, and $\boldsymbol{a_T} \in R^3$ represent the position, velocity, and acceleration of the target, respectively.

Based on these motion equations, five equations for the linear Kalman filter can be obtained, as shown in Equation (4).

$$\begin{cases} \boldsymbol{p}_T^-[(n+1)T] = \boldsymbol{F}\boldsymbol{p}_T(nT) \\ \boldsymbol{\Sigma}^-[(n+1)T] = \boldsymbol{F}\boldsymbol{\Sigma}^-(nT)\boldsymbol{F}^T + \boldsymbol{Q} \\ \boldsymbol{K} = \boldsymbol{\Sigma}^-(nT)\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{\Sigma}^-[(n+1)T]\boldsymbol{H}^T + R_T)^{-1} \\ \boldsymbol{p}_T[(n+1)T] = \boldsymbol{p}_T^-[(n+1)T] + \boldsymbol{K}\{(\boldsymbol{O}_T[(n+1)T] \\ \qquad\qquad - \boldsymbol{H}\boldsymbol{p}_T^-[(n+1)T]\} \\ \boldsymbol{\Sigma}[(n+1)T] = (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{\Sigma}^-[(n+1)T] \end{cases} \tag{4}$$

where $\boldsymbol{p_T^-}[(n+1)T]$ denotes the estimated value of the target's position at the beginning of the next sampling period; $\boldsymbol{F}$ is the state transition matrix and can be expressed as Equation (5), which indicates the relationship between the current state and subsequent states; and $\boldsymbol{\Sigma}^-[(n+1)T]$ denotes the covariance prediction at the beginning of the next sampling period. The symbol "−" in the above variables indicates that the state is estimated based on the previous state. $\boldsymbol{O_T}[(n+1)T]$ represents the position measurement value of the target at the beginning of the next sampling period. $\boldsymbol{Q}$ represents the covariance matrix of the process noise, and the larger its vector magnitude is, the more trusted the position measurement value of the target is, and on the contrary, the estimated value is more trusted. $\boldsymbol{H}$ represents the measuring matrix, and $\boldsymbol{H}=$ [1 0]. R_T denotes the measurement noise covariance matrix. The value of R_T is approximately equal to the variance of the measurement noise. $\boldsymbol{K}$ is the Kalman gain, which is used for moderating the prediction values of $\boldsymbol{p}_T$ and $\boldsymbol{\Sigma}$.

$$\boldsymbol{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{5}$$

In Equation (4), the first two equations are used to estimate the position of the target, and the fourth and fifth equations are used to modify the estimated value based on the position measurement value of the target so that a more accurate target position can be obtained.

The effect of the Kalman filter described in Equation (4) can be shown with the example in Fig. 3, in which the blue points represent the estimated position and the red "+" represents the actual position. The estimated position largely coincides with the actual position; R_T =0.01 (m), and the value of $\boldsymbol{Q}$ is shown in Equation (6).

$$\boldsymbol{Q} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \tag{6}$$

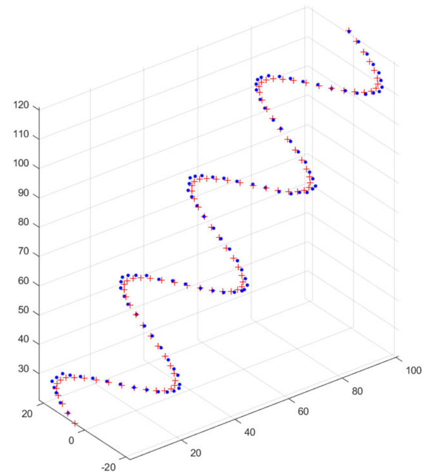


FIGURE 3. Example of a moving target trajectory prediction in a three-dimensional coordinate system.

There are still errors in the estimated position when the velocity varies greatly, but most of the time, the errors are very small. Furthermore, we have designed an experiment to observe the effect of different measurement noise (R_T) on the error of the estimation position. The experimental environment and parameters are the same as in Fig 3. And the average distance between the estimation position and the actual position per second (denoted as d_e) is used to evaluate the effect of the estimation error. The results are shown in Table 1.

TABLE 1. The effect of different R_T on the estimation error.

R_T (m)	0.01	0.1	1
d_e (m)	1.3437	1.9595	4.0766

From Table 1, we can see that the d_e does not grow proportionally with R_T, because according to the characteristic of Kalman filtering, the estimated value will be constantly corrected based on the actual measured value. And, the error of the estimated value will not accumulate, so that the performance of the whole swarm will not have a big effect.

This example proves that the Kalman filter is suitable for our application scenario, and by using it, the position of the target can be estimated with high accuracy.

B. INTERCEPTING PROCESS AND FORMATION STRATEGY

In the problem of intercepting a moving target with a UAV swarm, the UAV swarm is required to maintain a suitable formation during flight. And the formation strategy should be dynamically adjusted according to the distance between the virtual leader and the target. The intercepting process can be divided into the following four phases:

Phase 1: Assembling: The UAVs start from their initial positions and are assembled according to a predetermined formation. The virtual leader is located at the mean position of the UAVs at their initial states and will stay until the UAV swarm completes the formation. Then, phase 1 is over.

Phase 2: Pursuit: After phase 1, if the distance from the target to the virtual leader is relatively long. The UAV swarm will fly towards the target in a formation which is suitable for flight. Starting from this phase, the position of the virtual leader keeps shifting according to the proposed guidance law which is detailed in Section III-C Step 1.

Phase 3: Approach: When the UAV swarm gets closer enough to the target, it will be arranged into a specific formation and get ready to surround the target. And, a larger formation spacing is set to facilitate the surrounding.

Phase 4: Interception: When the target is intercepted (the distance between the estimated position of the target and the virtual leader is zero), the UAV swarm will completely surround the target and the formation spacing will be reduced appropriately.

In the description hereinbefore, the flight destination of a UAV is the predetermined position of this UAV at the beginning of the next sampling period under ideal conditions. However, due to speed restrictions and collision avoidance, some UAVs may not be able to reach their destinations on time.

The flight destination of the virtual leader is located at the estimated position of the target at the beginning of the next sampling period. And for a real UAV, its flight destination will be calculated based on the virtual leader's position and the relative position of this UAV in the formation.

The position and the flight destination of the virtual leader can be expressed by Equation (7).

$$\begin{aligned} &\boldsymbol{p}_U^{vl} \in R^3 \\ &\boldsymbol{p}_D^{vl}(nT) = \boldsymbol{p}_T^{-}[(n+1)T] \end{aligned} \tag{7}$$

The flight destinations $\boldsymbol{p}_D$ of the real UAVs can be obtained by Equation (8).

$$\boldsymbol{p}_D^i(nT) = \boldsymbol{p}_U^{vl}[(n+1)T] + \boldsymbol{\Delta f}(nT)(i,:), \quad i = 1, 2, \ldots, N. \tag{8}$$

where the formation parameter $\boldsymbol{\Delta f} \in R^{N\times 3}$ is used to determine the position of the UAVs relative to virtual leader, and $\boldsymbol{\Delta f}(nT)(i,:)$ represents all the elements in i-th row of $\boldsymbol{\Delta f}(nT)$.

And $\boldsymbol{\Delta f}$ is designed to be related to the motion direction of the virtual leader. The details of the calculation are as follows: The reference direction of the UAV formation is set to $\boldsymbol{u}_1 = [0, 0, 1]$ which is a unit direction vector. The direction vector of the virtual leader in the motion direction is denoted as $\boldsymbol{u_2}$. By solving the quaternion between $\boldsymbol{u}_1$ and $\boldsymbol{u_2}$, the rotation matrix $\boldsymbol{Ro}$ can be calculated by a transformation from the quaternion (except for Phase 1, because the formation does not need to be rotated in this phase). The equation for the computation of $\boldsymbol{\Delta f}$ is given by Equation (9)

$$\boldsymbol{\Delta f}(nT) = k_s\boldsymbol{A} \times \boldsymbol{Ro}(nT) \tag{9}$$

where parameter k_s is used to adjust the formation spacing of the UAVs, and $\boldsymbol{A}$ is the original designed formation matrix. By substituting $\boldsymbol{Ro}$ into Equation (9), the reference direction of the UAV formation will be turned in the same direction as the line of sight of the virtual leader.

According to the experimental environment, the formation parameters (k_s and $\boldsymbol{A}$) and the entry conditions of each phase are set in section V-A.

The flight destinations will be optimally allocated to the real UAVs. And the optimal algorithm for matching the UAVs and their flight destinations will be proposed in section IV.

C. GUIDANCE LAW

The original pure-pursuit guidance law is used to track a known path. The main operation is to continuously search for the nearest waypoint within the range of the look-ahead distance and set it as the look-ahead point. Then, the pursuer will move towards the look-ahead point until the look-ahead point has reached the end of the path.

In this study, our goal is to intercept the moving target as fast as possible. The UAV swarm does not need to follow the target's trajectory before the target is intercepted. Therefore, the look-ahead point of a UAV can be set at its current flight destination. Both the virtual leader and real UAVs should calculate their trajectories according to the guidance law, but unlike the real UAVs, the flight destination (look-ahead point) of the virtual leader is set at the target. Other differences between virtual leader and real UAV will be explained in detail during the calculation.

During the flight, calculating the appropriate guidance inputs in real-time can make the motion trajectory of each UAV consistent with its expected trajectory. The calculation process of guidance inputs is given as follows:

Step 1: This step is to estimate the position of the i-th UAV and the virtual leader at the beginning of the next sampling period, the estimated position is denoted as $\boldsymbol{p}_U^{i-}[(n+1)T]$, As shown in Fig. 4, according to the pure-pursuit guidance law, $\boldsymbol{p}_U^{i-}[(n+1)T]$ can be obtained by Equation (10).

What is special in this step is that the position of the virtual leader is also obtained by Equation (10), the difference is the superscript, superscript vl represents this variable belongs to the virtual leader, and vl is added at the end

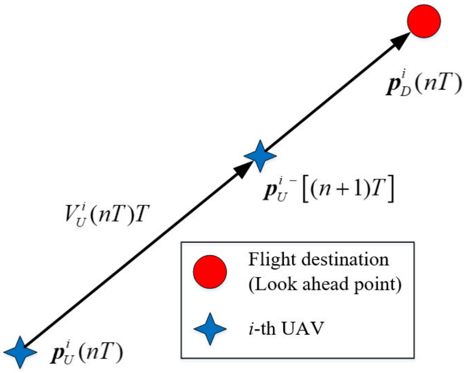


FIGURE 4. Geometric description of the pure-pursuit guidance law.

of i ($i = 1, 2, \ldots, N, vl$).

$$\boldsymbol{p}_U^{i-}[(n+1)T] = \boldsymbol{p}_U^{i}(nT) + \boldsymbol{\lambda}_D^{i}(nT)V_U^{i}(nT)T,$$
$$\boldsymbol{\lambda}_D^{i}(nT) = \frac{\boldsymbol{p}_D^{i}(nT) - \boldsymbol{p}_U^{i}(nT)}{\left|\boldsymbol{p}_D^{i}(nT) - \boldsymbol{p}_U^{i}(nT)\right|},\ i = 1, 2, \ldots, N, vl. \tag{10}$$

where $\boldsymbol{p}_D$ is the position of the flight destinations obtained by (7) and (8), and $\boldsymbol{\lambda}_D^{i}(nT)$ is the unit direction vector which points from $\boldsymbol{p}_U^{i}(nT)$ to $\boldsymbol{p}_D^{i}(nT)$.

The average flight speed $V_U^{i}(nT)$ during the n-th sampling period is designed in Equation (11). The purpose of this setting is to assign a specific value for $V_U^{i}(nT)$ as one of the guidance inputs, and the second equation in Equation (11) can prevent UAVs from going too fast and passing the target.

$$\begin{cases} V_U^{i}(nT) = v_p & \text{if } d_r^{i}(nT) > v_pT \\ V_U^{i}(nT) = d_r^{i}(nT)/T & \text{if } d_r^{i}(nT) \le v_pT \end{cases} \tag{11}$$

where $d_r^{i}(nT)$ represents the distance between $\boldsymbol{p}_U^{i-}[(n+1)T]$ and its corresponding flight destination $\boldsymbol{p}_D^{i}(nT)$, v_p is the predetermined flight speed of the UAV or the virtual leader during a sampling period, and the value of v_p is given in section V. If $\boldsymbol{p}_U^{i-}[(n+1)T]$ is updated, its corresponding $V_U^{i}(nT)$ also needs to be updated.

For the virtual leader, its calculation process stops at this step. It does not need to consider the guidance inputs and collision avoidance because it is virtual, and it will update to the estimated position $\boldsymbol{p}_U^{vl-}[(n+1)T]$ at the beginning of the next sampling period directly.

Step 2: Next, considering the collisions between UAVs or between UAVs and the target, a collision avoidance method is proposed. The UAVs and the target are treated as nodes, their positions are denoted as $\boldsymbol{p}_N$ and calculated as in Equation (12).

$$p_N = [p_U^{1-}[(n+1)T], p_U^{2-}[(n+1)T], \ldots, p_U^{N-}[(n+1)T], p_T[(n+1)T]] \tag{12}$$

For any two of the nodes, if the distance between them is less than the safe distance, a virtual force will be generated between the two nodes. And the displacement caused by the resultant virtual force for a node is denoted as $S_F^{i}(nT)$, which is calculated as in Equation (13). This part is inspired by the part of a coverage maximization algorithm based on molecular force [22], which can make the distance between the nodes approach to the optimal value, the difference is that in our study, the distance between nodes only needs to be greater than the safety distance.

Because the target cannot be controlled, in Equation (13) only the resultant virtual force of each UAV needs to be calculated for controlling the position of the UAVs.

$$S_F^{i}(nT) = \sum_{j=1, j\neq i}^{N+1} \frac{k_f \boldsymbol{Fv}_j^{i}[(n+1)T]}{\left\{\boldsymbol{d}_j^{i}[(n+1)T]\right\}^2},\ i = 1, 2, \ldots, N$$
$$\text{if } \boldsymbol{d}_j^{i}[(n+1)T] \ge \text{safe distance},\ \boldsymbol{Fv}_j^{i}[(n+1)T] = 0. \tag{13}$$

where $\boldsymbol{d}_j^{i}[(n+1)T]$ is the distance between the $\boldsymbol{p}_N^{i}$ and $\boldsymbol{p}_N^{j}$, $\left\{\boldsymbol{d}_j^{i}[(n+1)T]\right\}^2$ is used to amplify the influence of the closer nodes, so that each node will preferentially avoid the closest node during flight, $\boldsymbol{Fv}_j^{i}[(n+1)T]$ represents the vector from the $\boldsymbol{p}_N^{i}$ to $\boldsymbol{p}_N^{j}$, and k is a gain coefficient. The gain k_f is used to adjust the moving distance of the nodes when avoiding collision. In other words, k_f is the step length, if the value of k is too large, the moving distance will be too large, causing waste, but if the value of k_f is too small, the collision avoidance steps may need to run too many times to ensure collision avoidance. After many experiments, we concluded that when $k_f = -5$, the collision avoidance effect in our application scenario is better. Fig. 5 shows the schematic diagram of the virtual forces.

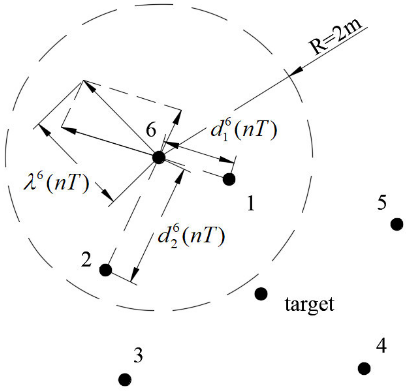


FIGURE 5. **Schematic diagram of virtual force for one of the nodes. The black points represent nodes. There are two nodes within the safe distance of the 6-th node–and $S_F^6(nT)$ is the obtained direction vector of the virtual force. $R = 2$m is the safe distance set here.**

The line of sight of each UAV is designed via Equation (14), which is a unit direction vector and denoted as $\boldsymbol{\lambda}^{i}(nT)$.

$$\boldsymbol{\lambda}^{i}(nT) = \frac{S_F^{i}(nT)}{\left|S_F^{i}(nT)\right|} \tag{14}$$

Step 3: $\boldsymbol{p}_U^{i-}[(n+1)T]$ can be updated by substituting $\boldsymbol{\lambda}^{i}(nT)$ from Equation (14) into Equation (15).

$$\begin{aligned} \boldsymbol{p}_U^{i-}[(n+1)T] &= \boldsymbol{p}_U^{i-}[(n+1)T] + \boldsymbol{\lambda}^{i}(nT)v_pT \\ &= \boldsymbol{p}_U^{i-}[(n+1)T] + \frac{S_F^{i}(nT)}{\left|S_F^{i}(nT)\right|}v_pT \end{aligned} \tag{15}$$

When UAVs are located at $\boldsymbol{p}_U^{-}[(n+1)T]$, if some of the distances between two nodes are still less than the safe distance, then the Steps 2-3 should be repeated from the beginning, and further update the estimated positions $\boldsymbol{p}_U^{-}[(n+1)T]$. Until all the distances between every two nodes are greater than the safe distance, the next step can be performed.

Step 4: $\boldsymbol{p}_U^{i-}[(n+1)T]$ can also be expressed according to the kinematic model of the UAV. By combining them, the value of Δx, Δy, and Δz can be obtained respectively.

$$\begin{aligned}
&\boldsymbol{p}_U^{i-}[(n+1)T] \\
&\quad = \boldsymbol{p}_U^i(nT) + [\Delta x^i(nT), \Delta y^i(nT), \Delta z^i(nT)] \\
&\qquad \times [\Delta x^i(nT), \Delta y^i(nT), \Delta z^i(nT)] \\
&\quad = \boldsymbol{p}_U^{i-}[(n+1)T] - \boldsymbol{p}_U^i(nT) \\
&\qquad \times \begin{cases} \Delta x^i(nT) = V_U^i(nT)T\cos\mu^i(nT)\cos\varphi^i(nT) \\ \Delta y^i(nT) = V_U^i(nT)T\cos\mu^i(nT)\sin\varphi^i(nT) \\ \Delta z^i(nT) = V_U^i(nT)T\sin\mu^i(nT) \end{cases}
\end{aligned} \tag{16}$$

By substituting Δx, Δy, and Δz into the kinematic model of the UAV. The flight path angle $\mu^i(nT)$ and course angle $\varphi^i(nT)$ can be obtained by Equation (17).

$$\begin{aligned}
\mu^i(nT) &= \sin^{-1}\left[\frac{\Delta z^i(nT)}{V_U^i(nT)T}\right] \\
\varphi^i(nT) &= \cos^{-1}\left\{\frac{\Delta x^i(nT)}{V_U^i(nT)T}\Bigg/\sqrt{1-\left[\frac{\Delta z^i(nT)}{V_U^i(nT)T}\right]^2}\right\} \\
&= \sin^{-1}\left\{\frac{\Delta y^i(nT)}{V_U^i(nT)T}\Bigg/\sqrt{1-\left[\frac{\Delta z^i(nT)}{V_U^i(nT)T}\right]^2}\right\}
\end{aligned} \tag{17}$$

Finally, the guidance inputs, i.e. the variation flight path angle $\Delta\mu^i(nT)$ and the variation course angle $\Delta\varphi^i(nT)$, can be obtained by calculating the variation value of μ and φ between the current sampling period and the previous sampling period as shown in Equation (2). Then, each UAV will update their position according to their guidance inputs ($\Delta\mu^i(nT)$, $\Delta\varphi^i(nT)$, and $V_U^i(nT)$) and arrive $\boldsymbol{p}_U^{i-}\ [(n+1)T]$ at the beginning of the next sampling period punctually, and at this moment, the estimated positions will be changed into the actual positions of the UAVs, i.e. $\boldsymbol{p}_U\ [(n+1)T] = \boldsymbol{p}_U^-\ [(n+1)T]$.

IV. MATCHING OPTIMIZATION

During the interception process, the matching between UAVs and their flight destinations needs to be optimized at the beginning of every sample period, otherwise, it may lead to a waste of the flight distance or failure to reach the predetermined flight destination in time, especially when the formation of the UAV swarm changes or the target makes a large angle maneuver.

To solve this problem, an optimal matching algorithm is proposed, which can find the optimal matching between the UAVs and their flight destinations during the flight.

A. FINDING THE OPTIMAL MATCHING

The matching problem between multiple UAVs and multiple flight destinations can be seen as a bipartite graph optimal matching problem with weights. In graph theory, this problem can be described as: how to find a certain perfect matching in a bipartite graph with edge weight and its total weight of the matching edges is maximum. In this paper, an optimal matching algorithm is proposed based on a combination of KM algorithm and Hungarian algorithm, and it is used to solve this problem. An example of this problem is shown in Fig. 6, where $x_\alpha \in \boldsymbol{X}\ (\alpha = 1, 2, \ \ldots, N)\,, y_\beta \in \boldsymbol{Y}\ (\beta = 1, 2, \ldots, N)$, and N is the number of UAVs.

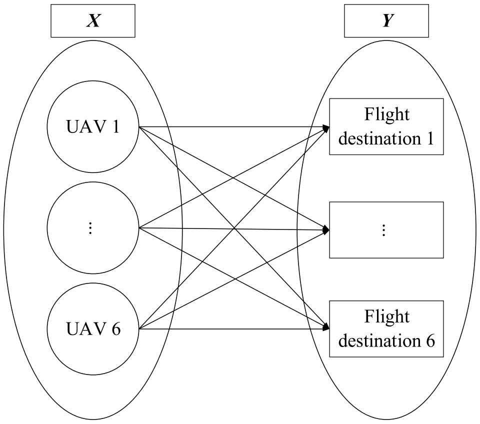


FIGURE 6. Example of a bipartite graph, an element in *X* represents a UAV, an element in *Y* represents a flight destination, and *X* and *Y* have the same number of elements.

Definition 1: A maximum matching is a matching that contains the largest possible number of edges. There may be many maximum matchings.

Definition 2: A perfect matching is a matching that matches all elements of the graph; that is, each element of the graph matches exactly one edge of the matching.

Hungarian algorithm is an optimal algorithm to solve the assignment problem, and the maximum matching of the above bipartite graph can be obtained by this algorithm. In our application scenario, the number of elements in $\boldsymbol{X}$ and $\boldsymbol{Y}$ is the same, and all the elements in $\boldsymbol{X}$ can be matched with all the elements in $\boldsymbol{Y}$. Therefore, the maximum matching is not unique, and these maximum matchings are also perfect matchings. If Hungarian algorithm is used alone, all UAVs can be guaranteed to be assigned to a flight destination, but the optimal matching cannot be obtained. Therefore, KM algorithm must be introduced to solve the optimal matching problem. In KM algorithm, the concept of weight is added. The weight of the connection between the α-th element in $\boldsymbol{X}$ and the β-th element in $\boldsymbol{Y}$ is denoted as ***weight*** (α, β), and the corresponding connected edge is denoted as ***edge*** (α, β). The value of the weight can reflect the priority of the match. The final result generated by the KM algorithm is the optimal matching with the largest total weights. The core of the KM algorithm is that by modifying the top marks of the elements of X and Y, so that the number of feasible edges is continuously increased, and all these top marks are guaranteed to be feasible top marks until a perfect matching is obtained, where the perfect matching includes only feasible edges.

Definition 3: In the bipartite graph, the top mark is a value used to determine whether the elements in $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be matched. The top marks of the elements in $\boldsymbol{X}$ and $\boldsymbol{Y}$ are denoted as $\boldsymbol{lx}(\alpha)$ and $\boldsymbol{ly}(\beta)$, respectively. If $\boldsymbol{lx}(\alpha)+\boldsymbol{ly}(\beta)\geq \boldsymbol{weight}(\alpha,\beta)$ is satisfied for any edge, these top marks are feasible top marks. If $\boldsymbol{lx}(\alpha)+\boldsymbol{ly}(\beta)=\boldsymbol{weight}(\alpha,\beta)$ is satisfied for an edge, this edge is called a feasible edge, which means that element α can be matched with element β at this time.

The process of finding the optimal matching based on the combination of the KM algorithm and the Hungarian algorithm is given as follows:

Step 1: Initialization. Let $\boldsymbol{lx}(\alpha)=\max\,[\boldsymbol{weight}\,(\alpha,\beta)]$, $\boldsymbol{ly}(\beta)=0$, and matrix $\boldsymbol{link}=[1,2,\ldots,N]$ which is a matching sequence of elements in X to elements in Y.

Step 2: Starting from the first element in $\boldsymbol{X}$, the depth-first search (DFS)-based Hungarian algorithm is used to find the maximum matching.

Step 3: If the perfect matching is not found, modify the value of the feasible top marks $\boldsymbol{lx}(\alpha)$ and $\boldsymbol{ly}(\beta)$ to increase the number of feasible edges, the detailed operations can be found in [35].

Step 4 Repeat Step 2 and Step 3 until the optimal perfect matching is obtained. Then update the ***link*** according to the optimal perfect matching. As shown in the example of Fig. 7.

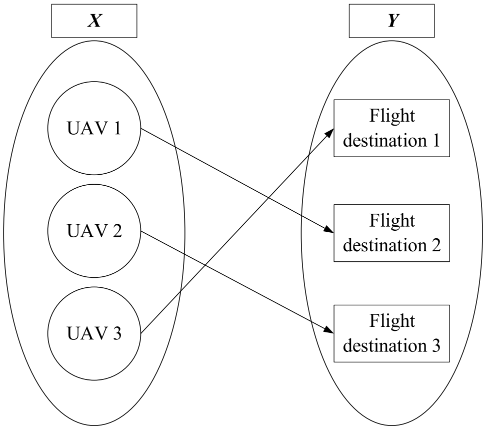


FIGURE 7. An example of an optimal matching result, where *link* = [3, 1, 2], which means that flight destination 1 is allocated to UAV 3, flight destination 2 is allocated to UAV 1, and flight destination 3 is allocated to UAV 2.

B. WEIGHT CALCULATION

Because the distances between the UAVs and their flight destinations are different, and the shorter the distance is, the faster a UAV can reach its flight destination. In order to make all the UAVs reach their destinations faster, the distances are set as the weights for the multitarget matching. If the α-th UAV is flying towards the β-th flight destination during the n-th sampling period, the weight between the α-th UAV and the β-th flight destination is denoted as $\boldsymbol{weight_n}(\alpha,\beta)$.

$$\boldsymbol{weight}_{\boldsymbol{n}}\,(\alpha,\beta)=\mathrm{norm}[\boldsymbol{p}_D^{\beta}(nT)-\boldsymbol{p}_U^{\alpha}(nT)] \tag{18}$$

where function norm returns the vector magnitude.

For the UAV swarm, our goal is to minimize the total distance, which is the sum of the distances between all UAVs and their destination during a sampling period. However, KM algorithm is used to find the perfect matching with the maximum total weight. Therefore, it is necessary to take its negative value for the weights. The optimization goal is given in Equation (19).

$$\max\sum_{i=1}^{N}-\boldsymbol{weight}_n[\boldsymbol{link}(i),i] \tag{19}$$

After the matching optimization, the correspondence between flight destinations and UAVs can be described by Equation (20), and in order to make the superscript of a UAV and its corresponding flight destination the same, $\boldsymbol{p}_U^i(nT)$ is updated by Equation (21).

$$\boldsymbol{p}_U^{\boldsymbol{link}(i)}(nT)\rightarrow\boldsymbol{p}_D^i(nT) \tag{20}$$

$$\boldsymbol{p}_U^i(nT)=\boldsymbol{p}_U^{\boldsymbol{link}(i)}(nT) \tag{21}$$

C. STEPS OF THE INTERCEPTION STRATEGY

The steps of the target interception strategy are listed as follows:

Step 1: Initialize the positions of the UAVs.

Step 2: Estimate the position of the target at the beginning of the next sampling period according to its current and historical positions.

Step 3: Update the position of the virtual leader based on the proposed guidance law. Then, according to the distance between the virtual leader and the target, the formation and flight destinations of the UAV swarm can be obtained.

Step 4: Calculate the distances of each UAV to each flight destination; then, all of the weights can be obtained according to Equation (18).

Step 5: Use the proposed optimal matching algorithm to obtain the optimal matching results of the UAVs and their flight destinations, then update the ***link***.

Step 6: Update the estimated positions of UAVs based on the proposed guidance law and ***link***, when the collision avoidance is considered.

Step 7: Calculate the guidance inputs according to the estimated positions of the UAVs.

Step 8: UAVs fly to the estimated positions according to the guidance inputs, then the estimated positions have become the actual positions.

Step 9: Run Steps 2 to 8 iteratively, and continuously update the estimated position of the target and the positions of the virtual leader and UAVs until the task is completed. The judgment condition for task completion is set according to the specific situation of the application scenario, for example, the invading unidentified flying object has been driven away from our airspace.

The flowchart of this strategy is shown in Fig. 8.

V. SIMULATION EXPERIMENTS

A. EXPERIMENTAL SETUP

To verify the validity of the proposed interception strategy and the optimal matching algorithm, in this section, the

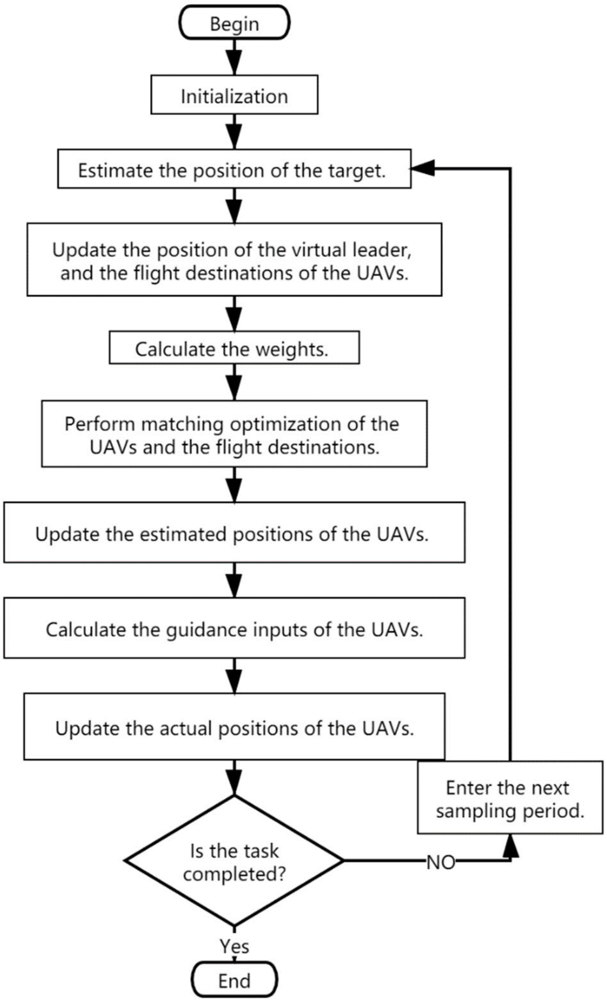

Begin
Initialization
Estimate the position of the target.
Update the position of the virtual leader, and the flight destinations of the UAVs.
Calculate the weights.
Perform matching optimization of the UAVs and the flight destinations.
Update the estimated positions of the UAVs.
Calculate the guidance inputs of the UAVs.
Update the actual positions of the UAVs.
Is the task completed?
NO
Enter the next sampling period.
Yes
End

FIGURE 8. The flowchart of the proposed interception strategy.

simulation software MATLAB 2018b is used to simulate the process of interception.

The experimental environment is set as follows: The experiments are performed in a three-dimensional space. The position measurement value of the target is designed as:

$$O_T(nT) = [nT + \text{randn}\sqrt{R_T}, \\ 20\sin(0.25nT) + \text{randn}\sqrt{R_T}, \\ nT + 20 + \text{randn}\sqrt{R_T}] \tag{22}$$

which is measured by UAVs. And the function "randn" returns a random scalar drawn from the standard normal distribution. In this way, the measurement noise follows the Gaussian distribution and the variance of the measurement noise is equal to R_T.

When the Kalman filter is used to estimate the position of the target, the relevant parameters are consistent with Equation (4) in section III-B, then $\boldsymbol{p}_T^-[(n+1)T]$ can be obtained. For the convenience of calculation, the sampling period T is set to 1 s.

In this application, six UAVs are assigned to intercept the target, so $N = 6$. In the initial state, the UAVs are randomly distributed in three-dimensional space.

The maximum speed and the average speed of the target are about 6 m/s and 3.8 m/s respectively. Therefore, for the virtual leader, the value of v_p is set to 6 m/s during every sampling period. And for each real UAV, the value of v_p should be properly increased so that the UAVs can catch up with the virtual leader, and in this case, v_p.is set to 7.5 m/s.

The initial positions and velocities of the UAVs at the beginning of the first sampling period are generated randomly, which are listed in Table 2. From $n = 2$ to 30 s, the positions of the UAVs and the target will be updated for 29 times.

TABLE 2. The initial position and velocity of the UAVs.

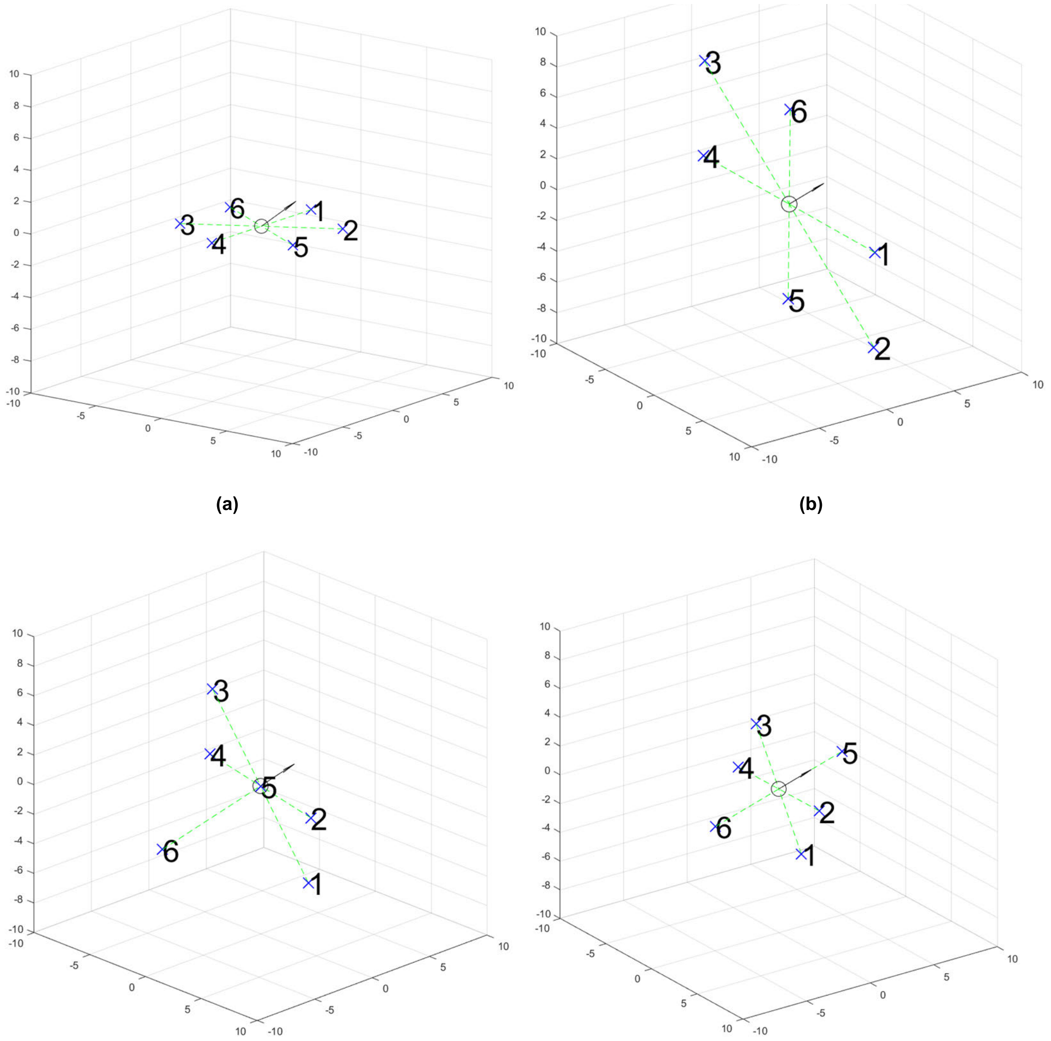
UAVs	Positions (m)	Velocities (m/s)
1	$\boldsymbol{p}_U^1(1T) = [-55.32,\ -19.37,\ 2.54]$	$\boldsymbol{v}_U^1(1T) = [-0.471,\ -0.230,\ -0.844]$
2	$\boldsymbol{p}_U^2(1T) = [-48.92,\ 31.76,\ 3.97]$	$\boldsymbol{v}_U^2(1T) = [-0.195,\ -0.226,\ -0.171]$
3	$\boldsymbol{p}_U^3(1T) = [-35.57,\ -12.14,\ 4.06]$	$\boldsymbol{v}_U^3(1T) = [-0.228,\ -0.436,\ -0.311]$
4	$\boldsymbol{p}_U^4(1T) = [-46.72,\ -14.93,\ 4.70]$	$\boldsymbol{v}_U^4(1T) = [-0.923,\ -0.430,\ -0.185]$
5	$\boldsymbol{p}_U^5(1T) = [-12.41,\ 5.02,\ 3.11]$	$\boldsymbol{v}_U^5(1T) = [-0.905,\ -0.980,\ -0.439]$
6	$\boldsymbol{p}_U^6(1T) = [-41.30,\ -29.23,\ 1.51]$	$\boldsymbol{v}_U^6(1T) = [-0.111,\ -0.258,\ -0.409]$

The formation parameters of each phase in our experiments are shown in Table 3. The distance between the virtual leader and the target is denoted as $\boldsymbol{d}_{lt}$, and the formation will be adjusted according to its changes. The formations of the UAV swarm of the four phases are shown in Fig. 9.

B. RESULTS ANALYSIS OF THE COMPARATIVE EXPERIMENTS

Experiment 1 includes the complete proposed interception strategy. To evaluate the performance of the proposed optimal matching algorithm, we introduced a control experiment (Experiment 2) for comparison. In Experiment 2, the optimal matching algorithm is not adopted, and the UAVs fly according to the initial matching sequence, which is set to ***link***= [1, 2, 3, 4, 5, 6]. The rest of the settings of Experiment 2 remain consistent with those of Experiment 1.

The experimental results are shown in Fig. 10 and Fig. 11, from which we can see the relationships among the UAVs, target, and flight destinations. The blue "×" symbols represent the positions of the UAVs, and the blue lines are their trajectories, and the blue numbers are the sequence number of the UAVs. The green dash lines connect the UAVs and the virtual leader at the beginning of the same sampling period, which is used to show the formation of the UAV swarm dynamically. The red "+" symbols represent the flight destinations. The black lines with a black "∗" show the trajectory of the target. The black "o" indicates the position of the virtual leader. The magenta dash line represents the line of sight of the virtual leader, which is from the virtual leader to the estimated position of the target at the beginning of



(a)

(b)

(c)

(d)

FIGURE 9. The formation topology of the four phases, where (a), (b), (c) and (d) correspond to phases 1 to 4, respectively, the blue “x” near the numbers represent UAVs, and the black “o” represents the virtual leader. The black arrow is the line of sight of the virtual leader.

TABLE 3. The formation parameters of each phase.

Phases	Assembling	Pursuit	Approach	Interception
$d_{lt}(nT)$ (m)	-	$d_{lt}(nT)$>500	0< $d_{lt}(nT)$<500	$d_{lt}(nT)$=0
k_s	1	2	7.5	5
$\boldsymbol{A}$	$\begin{bmatrix} 0 & 5 & 0 \\ \sqrt{18.75} & 2.5 & 0 \\ -\sqrt{18.75} & -2.5 & 0 \\ 0 & -5 & 0 \\ \sqrt{18.75} & -2.5 & 0 \\ \sqrt{18.75} & 2.5 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 5 & 0 \\ \sqrt{18.75} & 2.5 & 0 \\ -\sqrt{18.75} & -2.5 & 0 \\ 0 & -5 & 0 \\ \sqrt{18.75} & -2.5 & 0 \\ \sqrt{18.75} & 2.5 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$

the next sampling period. As shown in Fig. 10 and Fig. 11, in the beginning, six UAVs start from different positions and are assembled to a specific formation. Then, the UAV swarm gradually approaches the target using the proposed guidance law. As the distance between the virtual leader and target decreases, the formation is also adjusted according to

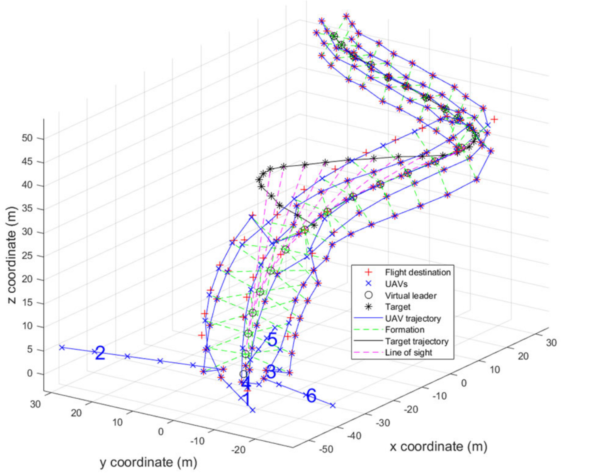


FIGURE 10. Trajectories when the UAVs are commanded to intercept the target with the proposed optimal matching algorithm (Experiment 1).

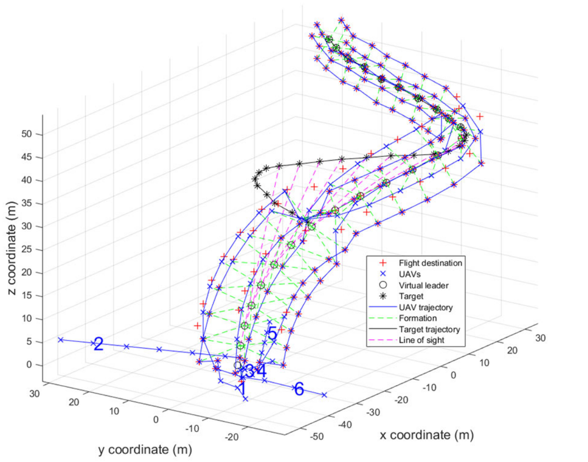


FIGURE 11. Trajectories when the UAVs are commanded to intercept the target without the proposed optimal matching algorithm (Experiment 2).

the proposed formation strategy. Finally, the virtual leader coincides with the estimated position of the target. During this process, when the red "+" and blue "×" coincide with each other, it indicates that the UAVs have reached the predetermined positions according to the proposed guidance law. The two experiments show that with the proposed guidance law, the UAV swarm can successfully intercept the target and maintain a formation flying around the target.

The relative distance between the virtual leader and the target is shown in Fig. 12, which reflects the process of interception. The rising part of the curve in the figure indicates that the UAV swarm is assembling, and as a result, the distance from the target increases. From phase 2, the UAV swarm begins to pursue, and the relative distance gradually decreases until the virtual leader catches up to the target, i.e. the relative distance is close zero.

Contributions of the proposed optimal matching algorithm consist of two parts. First, the UAVs do not need to maintain a specific formation when flying, and their goal is just to reach their flight destinations as soon as possible, such as phase 1. With the help of the proposed optimal matching algorithm, the total distance of all the UAVs from their initial positions to their flight destinations has been reduced from 144.19 m (Experiment 2) to 105.97 m (Experiment 1) in phase 1, and as a result, the flight time is also reduced. And in Fig. 12, we can see that the UAV swarm completed phase 1 in 6 s in Experiment 1, and in 7s in Experiment 2. Meanwhile, the average per-sampling-period total flight distance of all the UAVs (denoted as d_a) has been reduced from 24.03 m (Experiment 2) to 21.19 m (Experiment 1). As shown in Table 4.

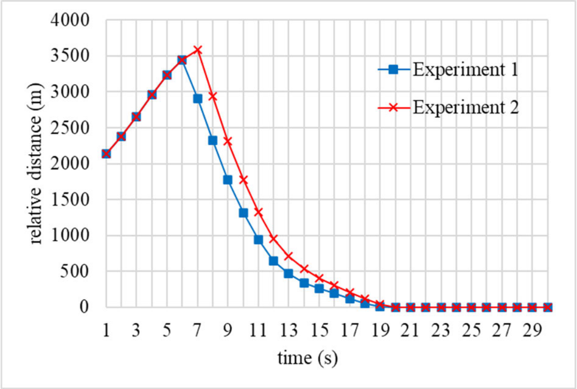


FIGURE 12. Relative distance between the virtual leader and target. The horizontal axis shows the time, and the vertical axis shows the relative distance.

TABLE 4. Contributions of the proposed optimal matching algorithm: phase 1.

	Total flight distance (m)	Duration time (s)	d_a (m)
Experiment 1	105.97	1-6	21.19
Experiment 2	144.19	1-7	24.03

In the second case, the UAVs need to maintain a specific formation while flying, the goal is to get the UAVs as close to their flight destinations as possible at the beginning of the next sampling period, such as phases 2, 3, and 4. Before the demonstration of the advantages, several definitions should be given first.

The distances between $p^i_D[(n-1)T]$ and $p^i_U(nT)$ is denoted as $d^i_{DU}(nT)$, and the total value $d_s(nT)$ can be obtained by sum up the distance of all UAVs in each sampling period, as shown in Equation (23), which is used to evaluate the formation completion degree at the beginning of each sampling period. The results are illustrated in Fig. 13.

$$\begin{aligned} d^i_{DU}(nT) &= \text{norm}\{p^i_D[(n-1)T] - p^i_U(nT)\} \\ d_s(nT) &= \sum_{i=1}^{N} d^i_{DU}(nT) \end{aligned} \tag{23}$$

The smaller the value of d_s is, the smaller the difference between the actual positions of the UAVs at the beginning

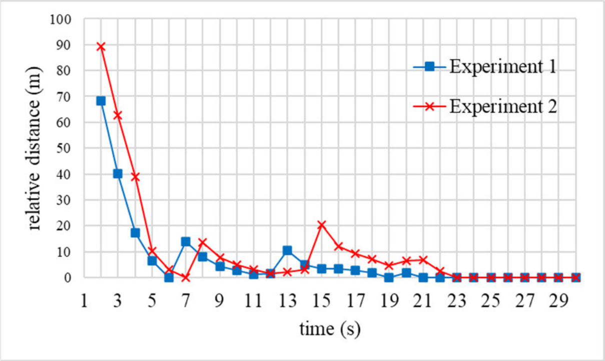


FIGURE 13. Total relative distance between UAVs and their corresponding flight destinations. The horizontal axis shows the time, and the vertical axis shows the relative distance.

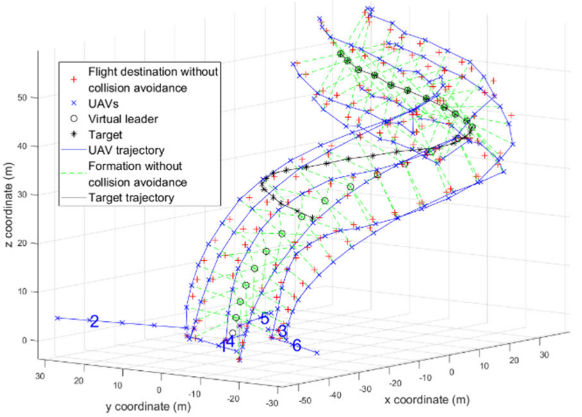


FIGURE 14. The trajectories of the UAVs during collision avoidance.

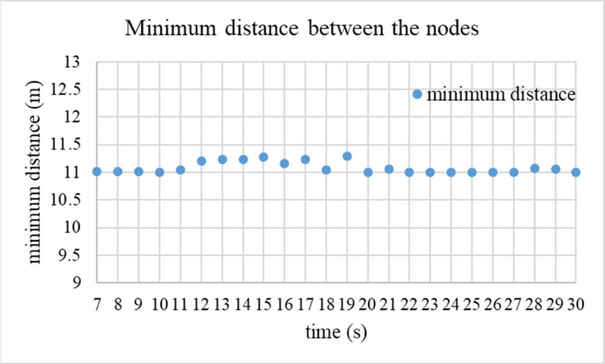


FIGURE 15. Minimum distance between any two nodes per each sampling period.

of the current sampling period and the predetermined flight destinations at the beginning of the previous sampling period is, and the higher the formation completion degree is.

From Fig. 13 we can see that, there are three crests in both two experiments. Because for the UAV swarm, it will take some time to change from the current formation to a new formation, the value of d_s will temporarily rise when the formation changes. In other words, these crests are the beginning points of phases 2, 3, and 4 respectively. For Experiment 1, the beginning points of phases 2, 3, and 4 are at 7 s, 13 s, and 20 s, and for Experiment 2, the beginning points are 8s, 15s, and 21s.

Most of the time, the value of d_s in Experiment 1 is less than that in Experiment 2. And if we sum up the values of d_s in all the sampling periods, this value of Experiment 1 is 192.71 m, which is also significantly smaller than that of Experiment 2 (309.37 m). Besides, for a single UAV, if $d^i_{DU}(nT) = 0$, it means that this UAV has reached its flight destination on time. This situation occurred 131 of 174 times in Experiment 1, but only 117 of 174 times in Experiment 2. Similarly, the average per-sampling-period total flight distance of all the UAVs (d_a) is also reduced. For Experiment 1, this value is 33.33 m, but for Experiment 2 this value is 34.40 m. As shown in Table 5.

TABLE 5. Contributions of the proposed optimal matching algorithm: phases 2-4.

	Sum of d_s (m)	Duration time (s)	d_a (m)	Reach on time
Experiment 1	192.71	7-12; 13-19; 20-30	33.33	131 of 174 times
Experiment 2	309.37	8-14; 15-20; 21-30	34.40	117 of 174 times

Throughout Experiment 1, the ***link*** of the UAV swarm has been changed for three times at 2 s, 13 s, and 21 s. Among them, 2 s and 13 s correspond to the beginning point of phases 1 and 3. The reason for the ***link*** changed at 2 s and 13 s is that at these two time points the formation of UAV swarm has changed significantly. And the change at 21 s is because the target has made a larger angle maneuver at this moment. According to statistics, the angle between the velocity vectors at 20 s and 21 s is about 47°, which is the second highest value during the whole experiment. The maximum angle appears at 7s, but at that time point UAV swarm is far away from the target, and the ***link*** has no need to change. In practical applications, the number of the formation changed or the times that the target made a large angle maneuver will be far more than those in this experiment. We can infer that the contributions of the proposed matching optimal algorithm will increase with the formation change and task processing.

In addition, during Experiment 1, the situation that the UAVs or target entered the safe distance did not occur. And during Experiment 2, this situation occurred for three times when we were estimating the positions of the UAVs (section III-C Step 1). But with the help of the proposed collision avoidance method, this situation did not actually happen.

In conclusion, the advantages of the proposed optimal matching algorithm are as follows. 1) When the UAVs do not need to fly in formation, their total flight distances will be reduced, so that UAVs can reach their predetermined flight destinations faster. 2) When the UAVs need to fly in formation, their actual positions will be closer to the predetermined formation form, so that the formation completion

TABLE 6. Table of notation.

Notation	Description
N	number of UAVs
T	sampling period
suffix n	belong to n-th sampling period (n=1, 2, …)
superscript i	belong to i-th UAV (i=1, 2, …, N)
$\boldsymbol{p}_T$	position of target
$\boldsymbol{p}_U$	position of UAVs
V_U	average flight speed along the line of sight of UAVs
μ	flight path angle
φ	course angle
Δx, Δy, Δz	projections of the displacement vector
$\Delta\mu$ and $\Delta\varphi$	variation values of them between adjacent sampling periods.
$\boldsymbol{v}_T$	velocity of target
$\boldsymbol{a}_T$	acceleration of target
superscript $^-$	estimated value
$\boldsymbol{F}$	state transition matrix
$\boldsymbol{\Sigma}^-$	covariance prediction
$\boldsymbol{O}_T$	position measurement value of target
$\boldsymbol{Q}$	covariance matrix of process noise
$\boldsymbol{H}$	measuring matrix
R_T	variance of the measurement noise
$\boldsymbol{K}$	Kalman gain
d_e	average distance between the estimation position and the actual position per second
$\boldsymbol{p}_D$	flight destination
$\boldsymbol{\Delta f}$	formation parameter
superscript vl	belong to virtual leader
$\boldsymbol{u}_1$	reference direction of formation
$\boldsymbol{u}_2$	direction vector of virtual leader in motion direction
$\boldsymbol{Ro}$	rotation matrix
k_s	parameter used for adjusting the formation spacing
$\boldsymbol{A}$	formation matrix
$\boldsymbol{\lambda}_D$	unit direction vector points from $\boldsymbol{p}_U$ to $\boldsymbol{p}_D$
d_r	distance between $\boldsymbol{p}_U^-$ and $\boldsymbol{p}_D$
v_p	predetermined flight speed of UAV or virtual leader during a sampling period
$\boldsymbol{p}_N$	position of nodes
S_F	resultant virtual force
$\boldsymbol{d}_j^i$	distance between the $\boldsymbol{p}_N^i$ and $\boldsymbol{p}_N^j$
$\boldsymbol{F}v_j^i$	vector from the $\boldsymbol{p}_N^i$ to $\boldsymbol{p}_N^j$
k_f	parameter used for adjusting the nodes' moving distance when avoiding collision
$\boldsymbol{\lambda}$	unit direction vector of line of sight of UAV
$\boldsymbol{d}_{lt}$	distance between the virtual leader and the target
d_a	average per-sampling-period total flight distance of all the UAVs
d_{DU}	distances between p_D and p_U
d_s	total value of d_{DU}

degree is higher, and the average per-sampling-period total flight distance of the UAVs is also reduced. 3) To a certain extent, the probability that the UAVs or the target enter the safe distance can be reduced.

C. EFFECT OF COLLISION AVOIDANCE

To verify the effectiveness of the collision avoidance method, a simulation experiment is performed to observe the process of collision avoidance. Most of the experimental settings are the same as the above experiments. But for the ease of observation, this experiment only contains phases 1 and 2, and the formations are the same as the formation (b) in Fig. 9. Similarly, the UAVs and the target are treated as nodes, and to increase the probability of collision avoidance, from phase 2, the safety distance is set longer than the minimum distance between two nodes in the normal formation flight. In norm formation (b), the minimum distance between two nodes is 10 m, and the safe distance is set to 11 m. The results are shown in Fig. 14, and the meaning of most symbols in this figure are the same as in Fig. 10 and Fig. 11, with the only two differences that in Fig. 14, the red "+" represents the flight destinations according to the normal formation (b), and the green dash line together forms the formation without considering the collision avoidance.

From Fig. 14, we can see that, to maintain the safe distance the formation spacing of the blue "×" is expanded, so that the blue "×" are all on the outside of the normal formation. And according to our statistics, after the collision avoidance operation, all the distances between two nodes are greater than the safety distance (11m), the minimum distances between any two nodes per each sampling period are shown in Fig.15.

In summary, the collision avoidance method can effectively prevent the UAVs and the target from entering the safe distance of each other.

VI. CONCLUSION

In this paper, a novel optimal guidance strategy is proposed for a UAV swarm to intercept a moving target, which includes a three-dimensional guidance law for UAVs and an optimal matching algorithm between UAVs and their flight destinations. It distinguishes from previous studies in that the target is allowed to be in a state of variable-velocity curvilinear motion instead of uniform rectilinear motion only; in addition, collision avoidance is considered. The proposed guidance law is based on the pure-pursuit guidance law and can guarantee that each UAV reaches its flight destination correctly. The formation strategy is designed based on the flight characteristics of multirotor UAVs, which allows the UAV swarm to intercept the moving target with a reasonable formation. From the simulation experiment results, we can conclude that with the proposed guidance law, the UAV swarm can successfully intercept the target while the predetermined formations are maintained, and the collisions are avoided. And with the help of the proposed optimal matching algorithm, the average per-sampling-period total flight distance of all the UAVs is reduced, the interception process is accelerated, and furthermore, the degree of completion of the formation is improved. All the above theories are demonstrated by simulation experiments. However, there is still a shortcoming which is that the gain coefficient k needs to be manually adjusted to ensure the effectiveness of the collision avoidance. Our future research is to design a dynamic scheme so that the gain coefficient k can be automatically adjusted appropriately.

REFERENCES

[1] M. Aljehani and M. Inoue, "Performance evaluation of multi-UAV system in post-disaster application: Validated by HITL simulator," *IEEE Access*, vol. 7, pp. 64386–64400, 2019.

[2] R. Kirichek, A. Paramonov, and A. Koucheryavy, "Swarm of public unmanned aerial vehicles as a queuing network," in *Proc. Int. Conf. Distrib. Comput. Commun. Netw.* Cham, Switzerland: Springer, Oct. 2015, pp. 111–120.

[3] S. Spanogianopoulos, Q. Zhang, and S. Spurgeon, "Fast formation of swarm of UAVs in congested urban environment," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8031–8036, Jul. 2017.

[4] Y. Wang, P. Bai, X. Liang, W. Wang, J. Zhang, and Q. Fu, "Reconnaissance mission conducted by UAV swarms based on distributed PSO path planning algorithms," *IEEE Access*, vol. 7, pp. 105086–105099, 2019.

[5] M. Rosalie, J. E. Dentler, G. Danoy, P. Bouvry, S. Kannan, M. A. Olivares-Mendez, and H. Voos, "Area exploration with a swarm of UAVs combining deterministic chaotic ant colony mobility with position MPC," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2017, pp. 1392–1397.

[6] M. R. Brust, M. Zurad, L. Hentges, L. Gomes, G. Danoy, and P. Bouvry, "Target tracking optimization of UAV swarms based on dual-pheromone clustering," in *Proc. 3rd IEEE Int. Conf. Cybern. (CYBCONF)*, Jun. 2017, pp. 1–8.

[7] E. Koyuncu, M. Shabanighazikelayeh, and H. Seferoglu, "Deployment and trajectory optimization of UAVs: A quantization theory approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8531–8546, Dec. 2018.

[8] W. Huang, J. Peng, and H. Zhang, "User-centric intelligent UAV swarm networks: Performance analysis and design insight," *IEEE Access*, vol. 7, pp. 181469–181478, 2019.

[9] H. Wang, H. Zhao, W. Wu, J. Xiong, D. Ma, and J. Wei, "Deployment algorithms of flying base stations: 5G and beyond with UAVs," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10009–10027, Dec. 2019.

[10] Z. Shiyu and Z. Rui, "Cooperative guidance for multimissile salvo attack," *Chin. J. Aeronaut.*, vol. 21, no. 6, pp. 533–539, Dec. 2008.

[11] D. Hou, X. Sun, Q. Wang, and C. Dong, "Finite-time cooperative guidance laws for multiple missiles with acceleration saturation constraints," *IET Control Theory Appl.*, vol. 9, no. 10, pp. 1525–1535, Jun. 2015.

[12] J. Zhao, R. Zhou, and Z. Dong, "Three-dimensional cooperative guidance laws against stationary and maneuvering targets," *Chin. J. Aeronaut.*, vol. 28, no. 4, pp. 1104–1120, Aug. 2015.

[13] H. Zhao, J. Wei, S. Huang, L. Zhou, and Q. Tang, "Regular topology formation based on artificial forces for distributed mobile robotic networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 10, pp. 2415–2429, Oct. 2019.

[14] H. Zhao, H. Wang, W. Wu, and J. Wei, "Deployment algorithms for UAV airborne networks toward on-demand coverage," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2015–2031, Sep. 2018.

[15] X. Wei, Y. Wang, S. Dong, and L. Liu, "A three-dimensional cooperative guidance law of multimissile system," *Int. J. Aerosp. Eng.*, vol. 2015, pp. 1–8, Oct. 2015.

[16] I.-S. Jeon, J.-I. Lee, and M.-J. Tahk, "Homing guidance law for cooperative attack of multiple missiles," *J. Guid., Control, Dyn.*, vol. 33, no. 1, pp. 275–280, Jan. 2010.

[17] J. Zhang, B. Xiao, M. Lv, and Q. Zhang, "Design and flight-stability analysis of a closed fixed-wing unmanned aerial vehicle formation controller," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 233, no. 8, pp. 1045–1054, Sep. 2019.

[18] J. Zhang, J. Yan, M. Lv, X. Kong, and P. Zhang, "UAV formation flight cooperative tracking controller design," in *Proc. 15th Int. Conf. Control, Automat., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 856–861.

[19] K. D. Do, "Synchronization motion tracking control of multiple underactuated ships with collision avoidance," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 2976–2989, May 2016.

[20] H. A. Poonawala, A. C. Satici, H. Eckert, and M. W. Spong, "Collision-free formation control with decentralized connectivity preservation for nonholonomic-wheeled mobile robots," *IEEE Trans. Control Netw. Syst.*, vol. 2, no. 2, pp. 122–130, Jun. 2015.

[21] H. Rezaee and F. Abdollahi, "A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 347–354, Jan. 2014.

[22] D. V. Dimarogonas, "Sufficient conditions for decentralized potential functions based controllers using canonical vector fields," *IEEE Trans. Autom. Control*, vol. 57, no. 10, pp. 2621–2626, Oct. 2012.

[23] Z. Chen, M. C. Fan, and H. T. Zhang, "How much control is enough for network connectivity preservation and collision avoidance?" *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1647–1656, Aug. 2015.

[24] X. Wang, G. Tan, X. Liu, and Z. Zhao, "A molecular force-based deployment algorithm for flight coverage maximization of multi-rotor UAV," *J. Intell. Robot. Syst.*, vol. 95, nos. 3–4, pp. 1063–1078, Sep. 2019.

[25] X. Wang, G.-Z. Tan, F.-L. Lu, J. Zhao, and Y.-S. Dai, "A molecular force field-based optimal deployment algorithm for UAV swarm coverage maximization in mobile wireless sensor network," *Processes*, vol. 8, no. 3, p. 369, 2020.

[26] R. Conti, E. Meli, A. Ridolfi, and B. Allotta, "An innovative decentralized strategy for I-AUVs cooperative manipulation tasks," *Robot. Auton. Syst.*, vol. 72, pp. 261–276, Oct. 2015.

[27] K. Kang and J. V. R. Prasad, "Development and flight test evaluations of an autonomous obstacle avoidance system for a rotary-wing UAV," *Unmanned Syst.*, vol. 1, no. 1, pp. 3–19, Jul. 2013.

[28] H. Zhao, H. Liu, Y.-W. Leung, and X. Chu, "Self-adaptive collective motion of swarm robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1533–1545, Oct. 2018.

[29] B. Zhu, A. H. B. Zaini, and L. Xie, "Distributed guidance for interception by using multiple rotary-wing unmanned aerial vehicles," *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5648–5656, Jul. 2017.

[30] R. Yanushevsky, *Guidance of Unmanned Aerial Vehicles*. Boca Raton, FL, USA: CRC Press, 2011.

[31] S. Ghosh, D. Ghose, and S. Raha, "Capturability of augmented pure proportional navigation guidance against time-varying target maneuvers," *J. Guid., Control, Dyn.*, vol. 37, no. 5, pp. 1446–1461, Sep. 2014.

[32] E. Garcia, D. W. Casbeer, and M. Pachter, "Cooperative strategies for optimal aircraft defense from an attacking missile," *J. Guid., Control, Dyn.*, vol. 38, no. 8, pp. 1510–1520, Aug. 2015.

[33] Z. Zhu, D. Xu, J. Liu, and Y. Xia, "Missile guidance law based on extended state observer," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5882–5891, Dec. 2013.

[34] G. Weiss and I. Rusnak, "All-aspect three-dimensional guidance law based on feedback linearization," *J. Guid., Control, Dyn.*, vol. 38, no. 12, pp. 2421–2428, Dec. 2015.

[35] R. Chai, J. Lin, M. Chen, and Q. Chen, "Task execution cost minimization-based joint computation offloading and resource allocation for cellular D2D MEC systems," *IEEE Syst. J.*, vol. 13, no. 4, pp. 4110–4121, Dec. 2019.

XI WANG received the B.E. degree in electrical engineering and the M.A.Eng. degree in control science and engineering from the Hunan University of Science and Technology, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree with the School of Automation, Central South University, China. His research areas are unmanned aerial vehicles, and swarm intelligence and its related applications.

GUANZHENG TAN received the B.S. degree from the Department of Aeronautical Engineering, Nanjing Aeronautical Institute, Nanjing, China, in 1983, the M.Sc. degree from the Department of Automatic Control, National University of Defense Technology, Changsha, China, in 1988, and the Ph.D. degree from the Department of Mechanical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, in 1992.

From October 1, 2004 to September 30, 2005, he was a Visiting Professor with the School of Computer Science, University of Birmingham, U.K. He is currently a Professor with the School of Automation, Central South University, Changsha. His main research areas are artificial intelligence and robotics.

YUSI DAI was born in Changsha, China, in 1996. She received the B.E. degree from the Department of Automation, School of Electrical Engineering and Information, Sichuan University, Chengdu, China, in 2018. She is currently pursuing the master's degree with the Department of Control Science and Engineering, School of Automation, Central South University, Changsha. Her main research interests include artificial intelligence and computer vision.

FANLEI LU received the B.E. degree from the Department of Detection Guidance and Control Technology, Central South University, Changsha, China, in 2016, where he is currently pursuing the master's degree with the School of Automation. His main research areas are cooperative control and the path planning of multiple UAVs.

JIAN ZHAO received the B.E. degree from the Department of Electrical Engineering, Chang'an University, Xi'an, China, in 2017. He is currently pursuing the master's degree with the School of Automation, Central South University, Changsha, China. His main research area is artificial intelligence.