

DOI:10.11992/tis.201604006  
网络出版地址: <http://kns.cnki.net/kcms/detail/23.1538.TP.20170630.1420.004.html>.

# A modified consensus algorithm for multi-UAV formations based on pigeon-inspired optimization with a slow diving strategy

ZHANG Tianjie, DUAN Haibin

(Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University (BUAA), Beijing 100191, China)

**Abstract:** This paper considers the formation control problem for a group of unmanned aerial vehicles (UAVs) employing consensus with different optimizers. A group of UAVs can never accomplish difficult tasks without formation because if disordered they do not work any better than a single vehicle, and a single vehicle is limited by its undeveloped intelligence and insufficient load. Among the many formation methods, consensus has attracted much attention because of its effectiveness and simplicity. However, at the beginning of convergence, overshoot and oscillation are universal because of the limitation of communication and a lack of forecasting, which are inborn shortcomings of consensus. It is natural to modify this method with lots of optimizers. In order to reduce overshoot and smooth trajectories, this paper first adopted particle swarm optimization (PSO), then pigeon-inspired optimization (PIO) to modify the consensus. PSO is a very popular optimizer, while PIO is a new method, both work but still retain disadvantages such as residual oscillation. As a result, it was necessary to modify PIO, and a pigeon-inspired optimization with a slow diving strategy (SD-PIO) is proposed. Convergence analysis was performed on the SD-PIO based on the Banach fixed-point theorem and conditions sufficient for stability were achieved. Finally, a series of comparative simulations were conducted to verify the feasibility and effectiveness of the proposed approach.

**Keywords:** unmanned aerial vehicle (UAV); formation; consensus; pigeon-inspired optimization (PIO); Banach fixed-point theorem

中图分类号: TP18; TP273    文献标志码: A    文章编号: 1673-4785(2017)04-0570-12

英文引用格式: ZHANG Tianjie, DUAN Haibin. A modified consensus algorithm for multi-UAV formations based on pigeon-inspired optimization with a slow diving strategy[J]. CAAI transactions on intelligent systems, 2017, 12(4): 570-581.

Formation control of unmanned aerial vehicles (UAVs) can accomplish a task better than a single UAV and has been studied using various methods. UAVs are widely adopted in various situations<sup>[1-2]</sup>. However, the requirements nowadays have already overtaken the current abilities of a single UAV. In order to satisfy the requirements, different plans for the cooperation of many UAVs have been proposed by Chaumette<sup>[3]</sup> and Rasche<sup>[4]</sup>, and the design of a method for multi-UAV formation is important<sup>[5]</sup>. Different methods have been proposed such as the

leader-follower approach<sup>[6]</sup>, artificial potential field<sup>[7]</sup>, and sliding mode control<sup>[8]</sup>.

Among these approaches, consensus is an effective and easy method for distributed control<sup>[9]</sup>. Using several simple consensus equations, UAVs can fly along complex trajectories. Consensus originates from the natural behavior of animals such as fish and birds. A single bird or fish is small and weak, but a population can hold a complicated formation. It is consensus that plays an essential role in this phenomenon. Vicsek<sup>[10]</sup> and Ren<sup>[9,11-12]</sup> have achieved many results in this field. This paper adopts this method to design a controller for a multi-UAV formation.

There are two main parts of the consensus algorithm<sup>[13-14]</sup>. First, the formation is treated as a virtual rigid body, every UAV has a desired position and this acts as a point on the body. Using consensus, the UAVs' desired positions concentrate together and create the desired formation. However, a distance error always exists between the desired position and the real position for each UAV, and the second part of the algorithm deals with this problem by considering this error as a variable in consensus. By devising a useful algorithm, the error closes to zero as time tends to infinity. The real position for each UAV is the sum of the desired position and the error. In this way, UAVs with random initial locations always concentrate and fly along their predesigned routes. These two parts of the algorithm added to the communication networks of the individual vehicles comprise the whole UAV system<sup>[13-14]</sup>. However, consensus can only ensure that all the variables are close to the given state and does not know whether the values of variables at each moment are optimal. It is a good idea to improve consensus with optimal algorithms.

Pigeon-inspired optimization (PIO) is a novel swarm intelligence optimizer proposed by Duan and Qiao in 2014<sup>[15]</sup>. The method imitates pigeons' behavior when seeking a way back to their loft. It is believed that pigeons adopt different tools at different stages when returning home. In the beginning, compass and map factors are adopted<sup>[16]</sup>, each pigeon's velocity is updated by the linear combination of its neighbors' (including itself) best locations and its former velocity. When pigeons approach home, some smart ones with good memories can recognize familiar landmarks which guide them<sup>[17]</sup>, the others just need to follow these leaders. The second part of PIO is designed to concentrate the average location of all the pigeons. It is widely known that each optimizer fits one or several problems, and PIO was first proposed for air robot path planning<sup>[15,18]</sup>, image processing<sup>[17,19]</sup>, and orbit formations<sup>[20]</sup>. When it comes to the optimization of consensus mentioned above, disadvantages appear such as the trajectory closes to the desired trace with oscillation to an unsatisfactory degree<sup>[21]</sup>. This means that PIO needs to be modified and a pigeon-inspired optimization with slow diving strategy (SD-PIO) is

proposed.

The remainder of this paper is organized as follows. In Section 2, details of the application of consensus in a multi-UAV formation are given. The construction of SD-PIO and the convergence analysis is considered in Section 3. In Section 4, the SD-PIO-based consensus algorithm is proposed for a multi-UAV formation, followed by numerical simulation and results analysis in Section 5. Finally, several conclusions concerning the use of optimal algorithms in consensus theory are drawn.

## 1 Controller design based on consensus

### 1.1 Multi-UAV formation controller design

It is supposed that each UAV possesses a moving coordinate system with three axes ( $x$ ,  $y$ , and  $z$ ). The shape of the UAV is ignored and a mass point model is used instead. This is because the time delay of an aircraft control loop is much shorter than that of the navigation and guidance loops. Therefore, when dealing with formation control, the UAV model can be simplified as a point, which means there is no need to consider each UAV's attitude. Under this assumption, all the UAV's  $x$ ,  $y$ , and  $z$  axes are parallel.

Consensus means that multiple UAVs reach an agreement on a common state, e. g., the UAV's position. In detail, every UAV has its initial 3-dimensional position ( $x$ ,  $y$ ,  $z$ ) in an inertial coordinate system, namely the ground coordinate system. The velocity of the original UAV<sub>*i*</sub>'s moving coordinate system converges to the standard state of the virtual leader UAV<sub>10</sub> by consensus. The UAV<sub>10</sub>'s state consists of a time-varying position and predesigned velocity. In this way, all the UAVs' coordinates become one common moving coordinate system. After adding the relative position of each UAV in the moving coordinate system, the desired time-varying position is achieved for each UAV. When viewed as a mass of moving points, all these UAVs constitute a virtual body.

The formation method consists of both single- and double-integrator consensus algorithms. The latter aims to concentrate the UAVs with random initial positions and ensure that the desired positions of all the UAVs converge to points on the predesigned trajectory. The

former ensures the distance error between the desired and the real position converges to 0<sup>[14]</sup>. The real position of each UAV can be easily acquired and the UAV's trajectory is drawn.

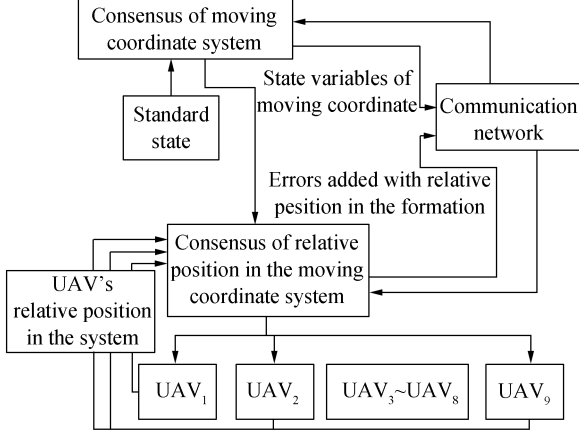


Fig.1 Frame of consensus controller in multi-UAV formation

For the formation problem in a multi-UAV system, the UAVs' states should include both position and velocity, and the control inputs are the accelerations<sup>[14]</sup>. This model is always described by second-order differentiation equations, and as a result, a consensus algorithm for double-integrator dynamics is required. The dynamic math model of the multi-UAV system is given below,

$$\begin{cases} \dot{\xi}_i = \zeta_i \\ \dot{\zeta}_i = \mu_i \\ \xi_i = [x_i \ y_i \ z_i]^T \end{cases} \quad (1)$$

where  $\xi_i, \zeta_i, \mu_i \in \mathbb{R}^3$  represent the position, the velocity and the acceleration of UAV<sub>*i*</sub>, respectively and  $x_i, y_i, z_i$  are the 3-dimensional positions,  $i \in \{1, 2, \dots, 10\}$ . The consensus algorithm can be described as in Ref [14].

$$\begin{aligned} \mu_i = & \frac{1}{k_i} \sum_{j=1}^9 a_{ij} [\dot{\zeta}_j - K_r(\xi_i - \xi_j) - K_v(\zeta_i - \zeta_j)] + \\ & \frac{1}{k_i} a_{i10} [\dot{\zeta}^r - K_r(\xi_i - \xi^r) - K_v(\zeta_i - \zeta^r)] \end{aligned} \quad (2)$$

Where  $a_{ij}$  is element  $(i, j)$  in the adjacency matrix  $A_{6 \times 6} \in \mathbb{R}^{6 \times 6}$ , and  $k_i = \sum_{j=1}^{10} a_{ij}$ ,  $K_r, K_v$  is a  $3 \times 3$  size symmetric positive definite matrix,  $i \in \{1, 2, \dots, 9\}$ , in this work this is replaced by a Pascal matrix for simplicity. The algorithm has been proven effective and can make all the UAVs' both positions and velocities converge to the state of the root. This is because the graph satisfies the sufficient and necessary condition

that the directed communication topology graph must include a spanning tree<sup>[14]</sup>.

To attain the desired position for one single UAV in the moving coordinate system, the error between the desired and the real position should be treated as an extra variable, this can be described by single-integrator dynamics for simplicity. The math model can be described as

$$\dot{er}_i = u_i \quad (3)$$

and the consensus algorithm is in Ref [9].

$$u_i = - \sum_{j=1}^9 a_{ij} (er_i - er_j) \quad (4)$$

$$\tilde{r}_i = rp_i + er_i \quad (5)$$

Where  $\tilde{r}_i$  is the real position of UAV<sub>*i*</sub>, UAV<sub>*i*</sub> in the moving coordinate system,  $er_i$  is the distance error between the real and the desired position of UAV<sub>*i*</sub>,  $rp_i$  which is shown in Fig.2, is the desired position of UAV<sub>*i*</sub> in the moving coordinate system,  $i \in \{1, 2, \dots, 9\}$ . Equations (1~5) above can be transformed into a collective matrix form,

$$\begin{cases} (L \otimes I_3) \ddot{\xi} + (b \otimes I_3) \ddot{\xi}^r = -K_r[(L \otimes I_3)\xi + (b \otimes I_3)\xi^r] - \\ K_v[(L \otimes I_3)\dot{\xi} + (b \otimes I_3)\dot{\xi}^r] \\ \dot{e} = -(L \otimes I_3)e \\ r = \tilde{r} + \xi = rp + er + \xi \end{cases} \quad (6)$$

where  $L_{9 \times 9}$  is the Laplace matrix of the communication topology graph,  $b \in \mathbb{R}^9$  reflects which UAV can directly receive information from UAV<sub>10</sub>,  $\xi^r$  is the desired position and also the position of UAV<sub>10</sub>, which shows the desired path for all the UAVs, finally,  $\otimes$  represents the Kronecker product.

$K_r = \text{diag}(K_{r1}, K_{r2}, \dots, K_{r9})$ ,  $K_v = \text{diag}(K_{v1}, K_{v2}, \dots, K_{v9})$ ,  $er = (er_1^T \ er_2^T \ \dots \ er_9^T)^T$ ,  $\tilde{r} = (\tilde{r}_1^T \ \tilde{r}_2^T \ \dots \ \tilde{r}_9^T)^T$ ,  $r$  is the vector of the real position of each UAV,  $\xi$  is the vector of the position of the origin of each UAV's moving coordinate system.

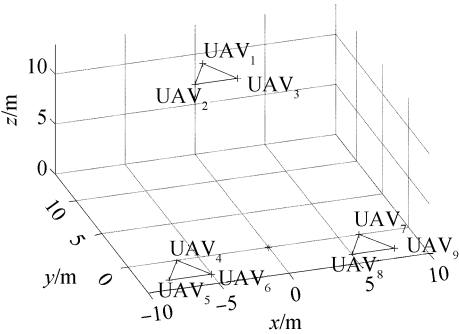
Equation (6) is the core of the multi-UAV controller, and  $r$  provides the necessary positional information for all the UAVs for use in the next step. After acquiring the target instructions, the UAVs fly toward their destinations.

## 1.2 Multi-UAV formation task

The desired formation shape is designed as follows:

**Table 1** Positions of 9 UAVs under the desired formation structure m

| UAV number       | $x$   | $y$   | $z$   |
|------------------|-------|-------|-------|
| UAV <sub>1</sub> | 0     | 12.60 | 10.00 |
| UAV <sub>2</sub> | -1.50 | 10.00 | 10.00 |
| UAV <sub>3</sub> | 1.50  | 10.00 | 10.00 |
| UAV <sub>4</sub> | -6.50 | 0     | 0     |
| UAV <sub>5</sub> | -8.00 | -2.60 | 0     |
| UAV <sub>6</sub> | -5.00 | -2.60 | 0     |
| UAV <sub>7</sub> | 6.50  | 0     | 0     |
| UAV <sub>8</sub> | 5.00  | -2.60 | 0     |
| UAV <sub>9</sub> | 8.00  | -2.60 | 0     |



**Fig.2** Formation structure for UAVs

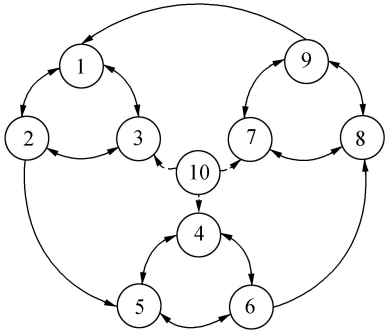
There are nine UAVs flying at two different heights, they are divided into three groups, each group containing three UAVs. In every small group, the UAVs form a triangle and these are the same size, so the UAVs fly symmetrically. The side length of the triangle and the distance between different triangles is much larger than the size of an UAV, so they can fly safely without danger of collision. The desired heights in Table 1 are relative values and the height of the origin of the moving coordinate system is their reference height, so when a UAV flies at zero or even a negative height, it is still acceptable. The blue star in Fig. 2 represents the center of the formation.  $x/m$  means distance along the  $x$ -axis in meters.

When designing the communication plan, the UAVs were numbered from 1 to 9. To clarify the plan, some background knowledge on topology and graph theory is required.

A graph with only vertices and arrows can be adopted to show the communication relationship among all the UAVs. Each UAV acts as a vertex, and an arrow from UAV<sub>1</sub> to UAV<sub>2</sub> means that UAV<sub>2</sub> can

receive information from UAV<sub>1</sub>. There is also a virtual leader named UAV<sub>10</sub>, which can emit an arrow to another UAV so that the latter knows the predesigned trajectory.

The communication topology among UAVs is described as a directed graph in Fig. 3. A graph with  $n$  vertices can be denoted by  $G(V,E,A)$ ,  $V = (v_1, v_2, \dots, v_n)$  is the set of vertices of graph  $G$  and  $v_i$  is a vertex.  $E = (e_1, e_2, \dots, e_n)$  is the edge set in which  $e_i$  means an edge.  $A = [a_{ij}]$  demonstrates the adjacency matrix of the graph  $G$ . If there is an edge from  $v_i$  to  $v_j$ , then  $a_{ij} > 0$ , otherwise  $a_{ij} = 0$ . Usually,  $a_{ij} = 1$  if a particular value is not given. The value of  $a_{ij}$  is the weight of communication amount between  $v_i$  and  $v_j$ , which can influence the convergence speed between  $v_i$  and  $v_j$ . In the graph  $G(V,E,A)$ , if there is a set of edges,  $(v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n)$ , then there is a path from  $v_1$  to  $v_n$ . If there is a vertex from which many edges reach all the remaining vertices, then this vertex is called a root and  $G(V,E,A)$  includes a direct spanning tree.



**Fig.3** Communication topology for multiple UAVs

For consensus, the adjacency matrix and the Laplace matrix were acquired. The Laplace matrix  $L = [l_{ij}]$  was the same size as the adjacency matrix. In the matrix,  $l_{ij} = -a_{ij}$ , if  $i \neq j$ , otherwise  $l_{ii} = \sum_{j=1, j \neq i}^n a_{ij}$ .

According to the sufficient and necessary condition of multi-UAV consensus, there is a directed spanning tree with its root UAV<sub>10</sub>, in the communication topology graph, and all the state variables converge to the state of the root as time tends to infinity<sup>[11]</sup>.

## 2 Pigeon-inspired optimization with slow diving strategy

### 2.1 Basic pigeon-inspired optimization algorithm

Qiao<sup>[15]</sup> studied pigeon behavior when seeking

their way home and proposed the two-stage algorithm. The first stage of PIO is the map and compass part.  $\mathbf{X} = [\mathbf{X}_1^T \mathbf{X}_2^T \cdots \mathbf{X}_i^T \cdots \mathbf{X}_n^T]^T$  is the vector of the pigeons' positions,  $i$  is the number of virtual pigeons, and in this paper the UAVs fly in 3-dimensional space,  $\mathbf{X}_i = [x_i \ y_i \ z_i]^T$ .  $\mathbf{V} = [V_1^T V_2^T \cdots V_n^T]^T$  is the vector of the pigeons' velocities,  $\mathbf{V}_i = [v_x \ v_y \ v_z]^T$  is its element,  $i \in \{1, 2, \dots, 9\}$ . Both  $\mathbf{X}$  and  $\mathbf{V}$  are randomly initialized when the program starts. The update formulas are:

$$\begin{cases} v_i(t+1) = v_i(t) \cdot \exp(-Rt) + \text{rand} \cdot (x_{\text{lbst}} - x_i) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad \text{if } t < T_{\text{label}} \quad (7)$$

In the formula above, the best local position  $x_{\text{lbst}}$  is selected from the distance between the desired and present positions. In order to find the best local position, all the distance errors are calculated and ranked, and only the position with the smallest distance is defined as the best local position. For the minimum optimization question, the fitness function is:

$$\text{fitness}(x_i(t)) = |x_i(t) - x_{\text{istandard}}(t)| \quad (8)$$

After calculating the distance between the desired and the present positions of all the pigeons, only half of the pigeons with lower fitness are selected for the iteration. This ensures that the positions of these pigeons converge to the desired position as fast as possible.

In the second part, the landmark operator is employed. Its update formula is:

$$\begin{cases} N_p(t+1) = \frac{N_p(t)}{2} \\ x_{iC}(t+1) = \frac{\sum x_i(t) \cdot f(x_i(t))}{\sum f(x_i(t))} \\ x_i(t+1) = x_i(t) + \text{rand} \cdot (x_{iC}(t+1) - x_i(t)) \end{cases} \quad \text{if } t \geq T_{\text{label}} \quad (9)$$

Apparently,  $x_{iC}$  is the weighted average of all the nearby pigeons in pigeon  $i$ 's communication distance. It is supposed that the map and compass operator has already converged pigeons to the desired trajectory to a great extent. As a result, the landmark operator only needs to speed up this process. The PIO algorithm has already been proven to be robust and reliable for an aircraft path planning problem<sup>[15]</sup>. However, for the optimization of consensus, PIO can improve the

overshoot, but the result is not desirable, and the altitude of a UAV drops to the standard height with continual oscillation. To improve the performance of the algorithm, better methods are required.

## 2.2 Improved method with slow diving strategy

The problem in the former algorithm occurs mainly in the altitude part, the height of the UAV decreases too fast and it dives sharply. To slow the process, the update formula should be changed, not only to consider the best local position but also the slowest diving velocity. In the first part of PIO, the formula is not changed, but the method of acquiring the best local position is modified. The criterion is a linear combination of the distance error between the desired and the real positions and the diving speed. An UAV that dives very slowly is imitated by the others. This is the so-called "slow diving strategy". The update formula is:

$$\begin{cases} v_i(t+1) = v_i(t) \cdot \exp(-Rt) + \\ \quad r \cdot [\varphi \cdot (x_{\text{lbst}} - x_i) + (1 - \varphi) \cdot v_{\text{slow}}] \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad \text{if } t < T_{\text{label}} \quad (10)$$

where  $\varphi$  serves as a regulatory factor in the range of  $(0, 1)$ ; if  $\varphi$  approaches 0, the slowest pigeon will influence the team; if  $\varphi$  approaches 1, best local position will have much more of an effect.  $r \in (0, 1)$  is a random number.

The second part of PIO should be modified in the following way: all the UAVs converge to the center of the flock, which is calculated by a combination of the UAV positions and the slowest diving speed. The update equation can be described as:

$$\begin{cases} N_p(t+1) = \frac{N_p(t)}{2} \\ x_{iC}(t+1) = \frac{\sum x_i(t) \cdot f(x_i(t)) + 10 \cdot x_{\text{slow}} \cdot f(x_{\text{slow}})}{\sum f(x_i(t)) + 10 \cdot x_{\text{slow}} \cdot f(x_{\text{slow}})} \\ x_i(t+1) = x_i(t) + \text{rand} \cdot (x_{iC}(t+1) - x_i(t)) \end{cases} \quad \text{if } t \geq T_{\text{label}} \quad (11)$$

where  $x_{\text{slow}}$  is the position of the UAV with the slowest diving speed, this part represents the slowing diving strategy.

Fitness function Eq. (10) remains. Each UAV has its own fitness value.

### 2.3 Convergence analysis of SD-PIO based on Banach fixed-point theorem

In this section, convergence analysis of SD-PIO is performed and a sufficient condition is proposed for the first part of SD-PIO (SD-PIO-I). The main method adopted to prove the stability of iteration is the Banach fixed-point theorem, as stated below:

A map  $T$  from a normed space  $(V, d(\cdot))$  into itself is a contraction if there exists  $\lambda \in [0, 1)$  such that for all  $x, y \in V, d(T(x), T(y)) \leq \lambda d(x, y)$ .

In the first part of SD-PIO, each pigeon updates its speed and location according to Eq. (10), which can be transformed as

$$\begin{bmatrix} x_i(t+1) \\ v_i(t+1) \end{bmatrix} = \begin{bmatrix} 1-r\varphi & \exp(-Rt) \\ -r\varphi & \exp(-Rt) \end{bmatrix} \begin{bmatrix} x_i(t) \\ v_i(t) \end{bmatrix} + r \begin{bmatrix} \varphi & 1-\varphi \\ \varphi & 1-\varphi \end{bmatrix} \begin{bmatrix} x_{\text{lbest}} \\ v_{\text{slow}} \end{bmatrix} \quad (12)$$

when it considers all the pigeons,  $z(t+1) = Fz(t) + Bu(z(t))$ ,

where  $z(t) = [x_1(t) \cdots x_n(t) \ v_1(t) \cdots v_n(t)]^T$ ,

$$F = \begin{bmatrix} 1-r\varphi & & \exp(-Rt) & & \\ & \ddots & & \ddots & \\ & & 1-r\varphi & & \exp(-Rt) \\ -r\varphi & & \exp(-Rt) & & \\ & \ddots & & \ddots & \\ & & -r\varphi & & \exp(-Rt) \end{bmatrix}$$

$$B = r \begin{bmatrix} \varphi & & 1-\varphi & & \\ & \ddots & & \ddots & \\ & & \varphi & & 1-\varphi \\ \varphi & & 1-\varphi & & \\ & \ddots & & \ddots & \\ & & \varphi & & 1-\varphi \end{bmatrix}$$

$u(z(t)) = [x_{\text{lbest}1}(t) \cdots x_{\text{lbest}n}(t) \ v_{\text{slow}1}(t) \cdots v_{\text{slow}n}(t)]^T$ .

If iteration  $z(t+1) = Fz(t) + Bu(z(t))$  has a fixed-pointed  $z^*$ , as a result,  $z^* = Fz^* + Bu(z^*)$ , then SD-PIO-I will converge to  $z^*$  and each pigeon will reach its final stable location as time tends to infinity.

Denoting  $Z$  as a real number space  $R^n$  and distance  $d(x, y) = \max_{1 \leq i \leq 2n} |x_i - y_i|, x, y \in Z$ ,  $Z$  has been proven as a complete metric space. There is a map  $T: Z \rightarrow Z$ ,  $T(z) = Fz + Bu(z)$ , where  $F \in R_{2n \times 2n}$ ,  $B \in R_{2n \times 2n}$ ,  $u(z) = K_{2n \times 2n} z$ , then  $\forall z_1, z_2 \in Z, T(z_1), T(z_2) \in Z$ , it can be proven that if  $d(T(x), T(y)) \leq \lambda d(x, y)$ ,  $\lambda \in [0, 1)$ , there exists a fixed-point which satisfies the condition  $z^* = T(z^*) = Fz^* + Bu(z^*)$  and the states of SD-PIO-I converge to  $z^*$ . The following proves this conclusion.

Because the best local location and the slowest speed are among all the pigeons' locations and

velocities,  $z(i)$  is the  $i$ th element of  $z(t)$ ,  $i \in \{1, 2, \dots, 2n\}$

$\forall z(i) \in u(z(t)) = [x_{\text{lbest}1}(t) \cdots x_{\text{lbest}n}(t) \ v_{\text{slow}1}(t) \cdots v_{\text{slow}n}(t)]^T$   
 $z(i) \in z(t), \exists K \in N_{2n \times 2n}, \text{s.t. } u(z(t)) = Kz(t)$ .

$$d(T(z_1), T(z_2)) = \max_{1 \leq i \leq 2n} |T(z_1) - T(z_2)| = \max_{1 \leq i \leq 2n} |Fz_1 + Bu(z_1) - Fz_2 - Bu(z_2)| = \|Fz_1 + BK_1z_1 - Fz_2 - BK_2z_2\|_\infty \quad (13)$$

There is

$$L_j = F_{j \cdot} (z_1 - z_2) + B_{j \cdot} [0 \cdots x_{1a} - x_{2b} \ 0 \cdots v_{1a} - v_{2b} \ 0 \cdots 0]^T \quad (14)$$

s.t.  $|L_j| = \|Fz_1 + BK_1z_1 - Fz_2 - BK_2z_2\|_\infty$ , thus the largest element of  $Fz_1 + BK_1z_1 - Fz_2 - BK_2z_2$  is selected and denoted as  $L_j$  supposing it is the  $j$ th element.  $F_{j \cdot}$  and  $B_{j \cdot}$  are the whole  $j$ th row of  $F$  and  $B$ , respectively. If  $j \leq n$ ,  $x_{1a} - x_{2b}$  is the  $j$ th element of  $K_1z_1 - K_2z_2$  and  $v_{1a} - v_{2b}$  is the  $(j+n)$ th one, otherwise  $x_{1a} - x_{2b}$  is the  $(j-n)$ th element and  $v_{1a} - v_{2b}$  is the  $j$ th one.

$$|L_j| = |F_{j \cdot} (z_1 - z_2) + B_{j \cdot} [0 \cdots k_1 \Delta z \ 0 \cdots k_2 \Delta z \ 0 \cdots 0]^T| = |F_{j \cdot} (z_1 - z_2) + B_{j \cdot} [0 \cdots k_1 \cdots k_2 \cdots 0]^T \Delta z| \leq \|F(z_1 - z_2) + BK(z_1 - z_2)\|_\infty \leq \|F + BK\|_\infty \|z_1 - z_2\|_\infty = \|F + BK\|_\infty d(z_1, z_2) \quad (15)$$

Where  $\Delta z = \max_{1 \leq i \leq 2n} |z_1(i) - z_2(i)|$ ,  $|k_1| = |x_{1a} - x_{2b}| / \Delta z$ ,  $|k_2| = |v_{1a} - v_{2b}| / \Delta z, k_1, k_2 \in [-1, 0) \cup (0, 1]$ ,

$$K = \begin{bmatrix} 0 & & 0 \\ & \ddots & k_1 \\ & & \vdots \\ & & k_2 \\ 0 & & 0 \end{bmatrix}, k_1, k_2 \text{ lies in the } j \text{th column},$$

The equation holds if and only if  $x_{1a} - x_{2b} = v_{1a} - v_{2b} = \Delta z$ .

The Banach fixed-point theorem requires  $\|F + BK\|_\infty \in [0, 1)$

$$\Leftrightarrow \max \{ |1 - \varphi r + \exp(-Rt) + rk_1\varphi + rk_2(1 - \varphi)|, | -\varphi r + \exp(-Rt) + rk_1\varphi + rk_2(1 - \varphi) | \} < 1$$

$$\Leftrightarrow \begin{cases} -1 < -\varphi r + \exp(-Rt) + rk_1\varphi + rk_2(1 - \varphi) < 0 \\ -1 \leq k_1, k_2 \leq 1, k_1 k_2 \neq 0 \end{cases} \quad (16)$$

Equation (16) is a sufficient condition for the convergence of SD-PIO-I, once it is satisfied, we have  $d(T(x), T(y)) \leq \|F + BK\|_\infty d(x, y)$ ,  $\|F + BK\|_\infty \in [0, 1)$ , there exists a fixed-point  $z^*$  satisfying  $z^* = Fz^* + Bu(z^*)$

$$\begin{cases} x^* = v^* + x^* \\ x^* = x^* + \exp(-Rt)v^* + r[\varphi(x_{\text{lbest}}^* - x^*) + (1 - \varphi)v_{\text{slow}}^*] \end{cases} \quad (17)$$

Equation (17)  $\Rightarrow \begin{cases} v^* = 0 \\ x^* = x_{\text{lbest}}^* \end{cases}$ , the states of SD-PIO-I



can only converge to a local optimal solution because each pigeon can only receive its neighbor's information instead of global information.

For the second part of SD-PIO, all the pigeons converge to the flock's center. Their trajectories can only contract and approach each other. In formation processes this part serves only as a transition, and as the traces do not diverge this second part satisfies the demands.

### 3 Multi-UAV formations with a SD-PIO-based consensus algorithm

The first three sections introduced consensus and SD-PIO, now this section uses them to optimize the trajectory of an UAV. When consensus is only applied to the planned routes of all the UAVs, there is unavoidable overshoot because of the distributed control plan itself. Every UAV has its own limited communication relationship, which means it cannot always receive the desired trajectory. For example, UAV<sub>1</sub> can never acquire the planned trajectory directly, it relies on UAV<sub>3</sub> to pass on the information. If UAV<sub>3</sub>'s initial altitude is much too high, it must dive to meet the desired trajectory. Because of inertia and a lack of forecasting, UAV<sub>3</sub> can only produce positive additions to its height when it goes below the desired trajectory and this is the reason for overshoot. However, as for UAV<sub>1</sub>, only when its leader UAV<sub>3</sub> flies below the proposed height does it begin to recognize that it should fly a bit higher, then its velocity is changed by Eq. (7), followed by its position. This process is much slower than a change in the desired route. As a result, the UAVs' trajectories dive sharply and appear to overshoot, then they converge to the desired trajectory. SD-PIO provides a gentle and slowly changing trajectory for each UAV at the start, then, as soon as the UAVs fly down and meet the desired trajectory, consensus starts to operate.

The formula of the combination of these two algorithms can be described as:

$$\begin{cases} x(t+1) = x(t) + T \cdot (v(t+1) + v_2(t+1)) \\ v(t+1) = v(t) + T \cdot a(t+1) \\ a(t+1) = -((L \otimes I_3) \cdot x(t+1) + \gamma \cdot (L \otimes I_3) \cdot v(t+1)) \cdot \\ \left( \frac{1}{\pi} \times \arctan(R \times (t - T_{\text{chg}})) + 0.5 \right) \\ v_2(t+1) = f(x(t), v(t)) \cdot \\ \left( -\frac{1}{\pi} \times \arctan(R \times (t - T_{\text{chg}})) + 0.5 \right) \end{cases} \quad (18)$$

where  $T_{\text{chg}}$  is the moment when consensus replaces SD-PIO,  $\pm \frac{1}{\pi} \times \arctan(R \times (t - T_{\text{chg}})) + 0.5$ , the inverse trigonometric function, is utilized to modify the output of the SD-PIO and consensus, it works as a switch key to smoothly connect the two parts of the trajectory.  $f(x(t), v(t))$  is the output velocity of SD-PIO from Eqs. (7) and (9). The implementation procedure of the SD-PIO-based consensus algorithm is given as follows:

- 1) Input the number of UAVs in the formation task, give the expectation formation and consider the communication restriction.
- 2) Design a suitable communication network topology and acquire its adjacency matrix and the Laplace matrix.
- 3) Randomly initialize the positions and velocities of all the UAVs along with the parameters involved in the equations used in Eq. (18).
- 4) Calculate the acceleration input, consensus, and the velocity input using Eq. (18)
- 5) Update each UAV's position and velocity using Eq. (18)
- 6) Draw the UAV's trajectory according to its position in each iteration.

The process is summarized in the following process flow chart:

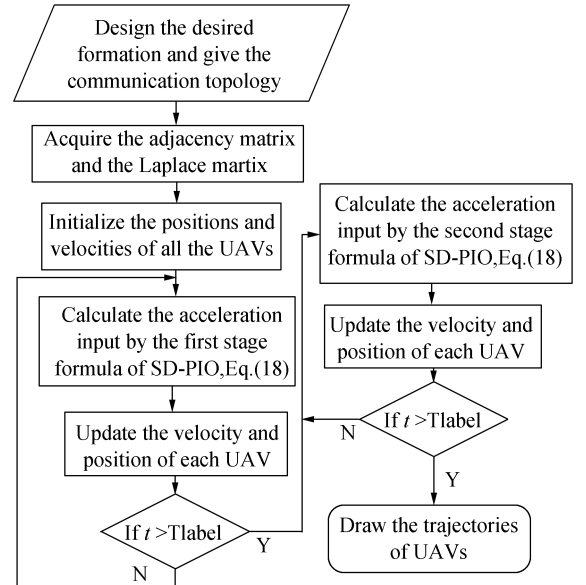


Fig. 4 The process of the SD-PIO based consensus algorithm for a multi-UAV formation

4 Simulation results and analysis

To validate the SD-PIO-based consensus, numerical simulations were conducted using MATLAB 2012a on a PC with an i5-480 m, 2.67 GHz CPU and Windows 10 operating system. SD-PIO was compared with consensus itself and consensus with two other optimizations, particle swarm optimization (PSO) and PIO. The results prove the effectiveness of SD-PIO.

The main parameters in Eqs. (7), (9), and (18) are listed in Table 2 below. The time step for the simulation is  $T=0.05$  s.

Table 2 Simulation parameters

| $\gamma$ | $R$ | $T_{\text{label}}/\text{s}$ | $T_{\text{chg}}/\text{s}$ |
|----------|-----|-----------------------------|---------------------------|
| 5        | 50  | 4                           | 6                         |

Nine UAVs were initialized with random positions in a 3-dimensional environment as shown in Table 3.

Table 3 Initial positions of 9 UAVs m

| UAV number       | $x$    | $y$    | $z$    |
|------------------|--------|--------|--------|
| UAV <sub>1</sub> | 20.804 | 30.971 | 30.477 |
| UAV <sub>2</sub> | 12.685 | 37.633 | 31.252 |
| UAV <sub>3</sub> | 22.260 | 38.622 | 24.661 |
| UAV <sub>4</sub> | 0.318  | 27.929 | 9.703  |
| UAV <sub>5</sub> | 21.859 | 15.846 | 6.076  |
| UAV <sub>6</sub> | 16.942 | 13.999 | 6.644  |
| UAV <sub>7</sub> | 19.293 | 20.723 | 17.662 |
| UAV <sub>8</sub> | 24.806 | 23.061 | 9.317  |
| UAV <sub>9</sub> | 37.298 | 12.688 | 7.861  |

These heights, the numbers on the z-axis, are relative positions corresponding to the height of origin of the common moving coordinate system in Fig. 2. Each UAV had the same initial position in the four simulations, i.e., consensus, PSO-based consensus, PIO-based consensus, and SD-PIO-based consensus.

As known from Sections 2 and 3, the performance of consensus cannot satisfy the requirements, which is clearly displayed in Fig. 5. Because of communication restrictions and lack of forecasting, all the UAVs tend to dive sharply and climb quickly. This causes great overload at the turning point, which may damage the UAV's structure and result in difficulties for flight control. Their routes seem to tie a knot when viewed

from this perspective, however, it is important to clarify the fact that the trajectories do not cross and the UAVs do not crash in the sky.

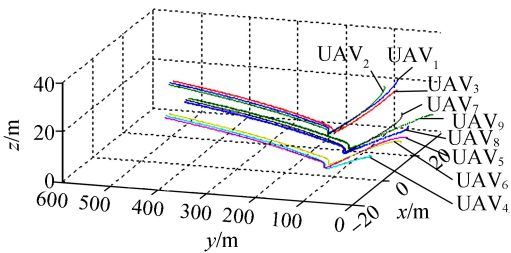


Fig.5 Trajectories of 9 UAVs by consensus

To demonstrate the advantage of SD-PIO-based consensus, PSO- and PIO-based consensus algorithms were adopted to compare their performances with that of the modified algorithm. In Fig. 6, most trajectories are apparently better than those in Fig. 5, as the trajectories are gentle at the beginning and turn without diving sharply. UAV<sub>3</sub>, UAV<sub>5</sub>, and UAV<sub>9</sub> serve as a set of convincing simulations that support the conclusions above. In Figs. 7 and 8, PIO and SD-PIO work and further improve the performance. In Fig. 7, the overshoot almost disappears, but there is still fluctuation in the trajectories from this perspective, especially for UAV<sub>3</sub>, UAV<sub>4</sub>, and UAV<sub>5</sub>. In Fig. 8, the trajectories become much smoother and the overshoot can be ignored compared with Figs. 5 to 7. This is reflected in the lowest parts of the trajectories of UAV<sub>3</sub>, UAV<sub>4</sub>, and UAV<sub>5</sub>.

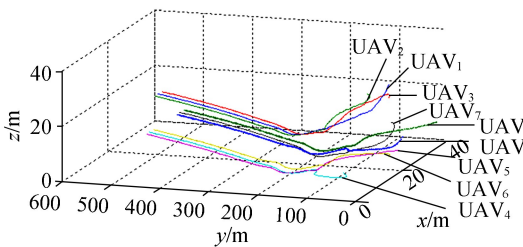


Fig. 6 Trajectories of 9 UAVs by PSO-based consensus

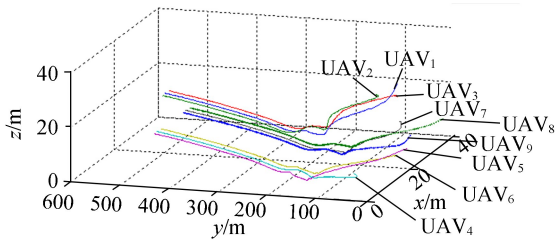
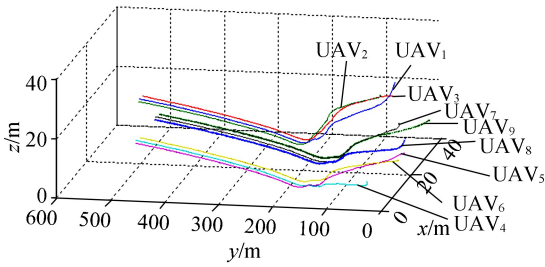


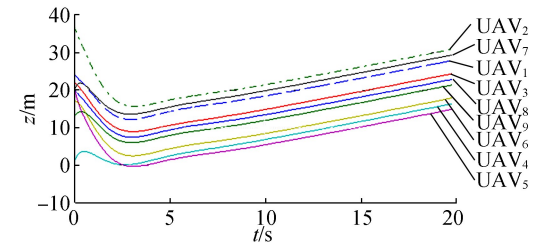
Fig. 7 Trajectories of 9 UAVs by PIO-based consensus



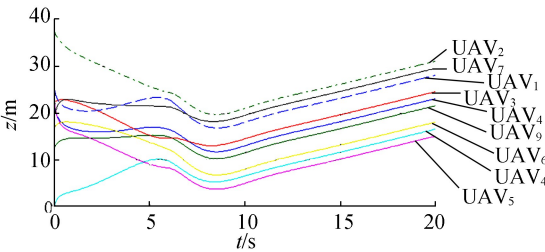


**Fig. 8** Trajectories of 9 UAVs by SD-PIO-based consensus

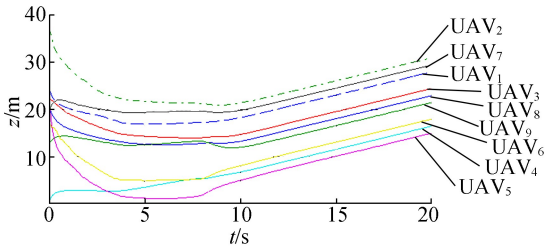
The effect of the SD-PIO-based consensus can be analyzed according to the  $x$  and  $z$  trajectories, but the  $y$  trajectory by consensus is smooth enough and does not need improvement. In Figs. 9 and 10, oscillation is more obvious than the counterparts in Figs. 11 and 12. The undesirable trajectories in Fig.11 partly explain the shortcomings of PIO.



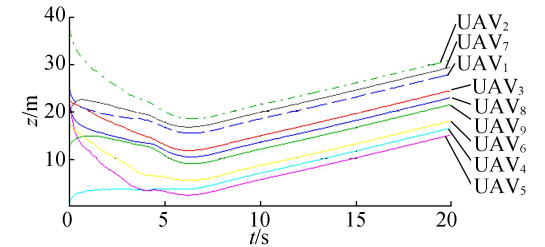
**Fig. 9**  $x$  trajectory of 9 UAVs by consensus



**Fig. 10**  $x$  trajectories of 9 UAVs by PSO-based consensus



**Fig. 11**  $x$  trajectories of 9 UAVs by PIO-based consensus



**Fig. 12**  $x$  trajectories of 9 UAVs by SD-PIO-based consensus

The performance of the  $x$  trajectories can be evaluated using change of direction as the criterion. After starting, if a UAV flies along a smooth trajectory towards the desired one without turning, it must have chosen the best and straightest trajectory. An undesirable trajectory leads the UAV into making frequent turns, which costs too much energy and increases the probability of collision.

The  $x$  trajectories in Figs. 9 and 10 belong to the undesirable type. In Fig. 9, the trajectories of UAV<sub>2</sub>, UAV<sub>4</sub>, and UAV<sub>7</sub> curve during flight and each of them turns twice or even three times. UAV<sub>2</sub> flies in the direction in which  $x$  becomes larger, however at nearly  $t = 1$  s,  $x$  begins to decrease, then at nearly  $t = 3.5$  s,  $x$  increases again and meets the desired value at  $t = 5$  s. This conflict process can be explained by its topology. In Fig. 3, UAV<sub>2</sub> is deeply influenced by UAV<sub>1</sub> and UAV<sub>3</sub>. Unfortunately, all its neighbors start at positions further along the  $x$ -axis, which means that UAV<sub>2</sub> has to catch up with its neighbor by consensus without reaching its desired positions. Only when its leader realizes they have deviated from the right position, do they begin to correct it. This is a result of a lack of forecasting. However, if UAV<sub>2</sub> can directly receive information from UAV<sub>10</sub>, perhaps it can perform better. This is the limit of communication. These two shortcomings illustrate the necessity for improvement.

In Fig. 10, the situation is not any better. UAV<sub>3</sub>, UAV<sub>4</sub>, UAV<sub>5</sub>, and UAV<sub>8</sub> have the same problem. For example, UAV<sub>8</sub> starts at  $x = 24.806$  m at  $t = 0$ , then  $x = 21.5$  m at nearly  $t = 2$  s, then  $x = 25.1$  m at nearly  $t = 6$  s and decreases to  $x = 18.3$  m at  $t = 8$  s; the real trajectory goes up and down frequently. This implies PSO cannot satisfy the demand of performance improvement.

In Figs. 11 and 12, fluctuation weakens and the trajectories become much smoother. Most UAVs can fly directly towards their destinations at  $t = 7$  s. In Fig. 11, the  $x$  position of UAV<sub>5</sub> decreases quickly from the beginning to  $t = 5$  s, this fast slide slip spoils the performance as it causes a sliding movement of a flying UAV and makes it harder to control.

In Fig. 12, most trajectories are pleasant and

suitable as commands for multi-UAV formations.

Performance of the  $z$  trajectories can be evaluated by the following criterion; diving slowly and never flying up and down. In Fig. 13, all the trajectories quickly go down before  $t = 4$  s, this means that the UAVs dive sharply in the beginning, as shown in Fig. 5. The aim of this work is to overcome this shortcoming.

The trajectories in Fig. 14 clarify the advantage of PSO-based consensus because the modified algorithm can effectively reduce oscillation. UAV<sub>2</sub> and UAV<sub>8</sub> clarify UAV<sub>8</sub> this view as they both slowly decrease their height following the desired trajectory instead of diving down swiftly.

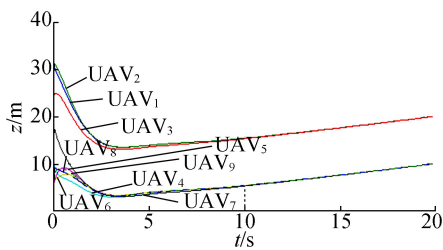


Fig. 13 Z trajectories of 9 UAVs by consensus

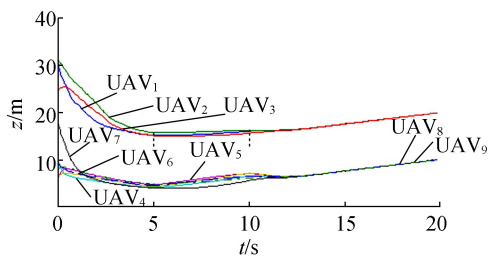


Fig. 14 Z trajectories of 9 UAVs by PSO-based consensus

However, reduplicative climb and dive still exist, which may induce oscillation. The dangerous trajectories of UAV<sub>3</sub> and UAV<sub>5</sub> at UAV<sub>5</sub>  $t \in [0, 10]$  s in Fig.14 reveal that they are not the desired ones that descend slowly at the beginning, then converge to the desired trajectory. This is why there is great demand for another optimizer.

In Fig. 15, PIO is used to apply a gliding procedure. Before 4s, the map and compass operator takes effect and all the UAVs fly towards the best local position. In the time range, 4 to 6 s, the landmark operator drives the UAVs nearby to the same height. When all of them fly to the standard height, the PIO part stops working and consensus continues concentrating the UAVs and guiding them to fly along

the designed trajectory.

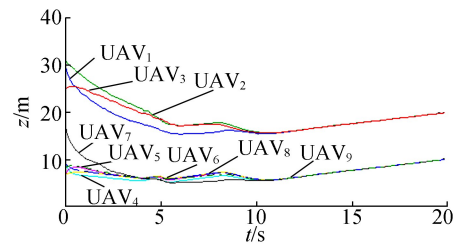


Fig. 15 Z trajectories of 9 UAVs by PIO-based consensus

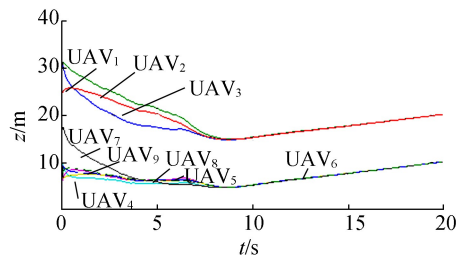
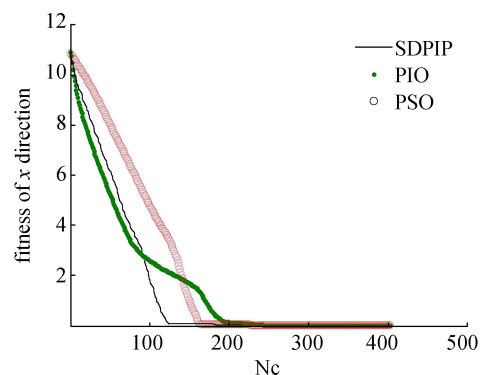


Fig. 16 Z trajectories of 9 UAVs by SD-PIO-based consensus

The SD-PIO-based consensus algorithm also follows the steps above, but the difference occurs in the first 8 s as the UAVs descend more slowly and smoothly. UAV<sub>1</sub>, UAV<sub>2</sub> and UAV<sub>3</sub> can support this opinion, as they at the same respective height, but at  $t = 5$  s, UAV<sub>2</sub> and UAV<sub>3</sub> by SD-PIO are above 20 m while their counterparts are clearly lower. At  $t = 7$  s, the height of UAV<sub>1</sub> by SD-PIO is 16.04 m and the outcome by PIO is only 15.47 m.

From the fitness function Eq. (8), the closer the UAV <sub>$i$</sub>  approaches its desired location, the smaller the fitness value. In Fig. 17, as iteration Nc increases, average fitness values decrease and finally approach zero, and different optimizers possess different convergence rates. In Fig. 17 (a), PIO shows the fastest speed in the beginning, SD-PIO is a bit slower and PSO shows the weakest performance of the three. However, after  $Nc = 80$ , PIO becomes slow and is finally overtaken by SD-PIO at  $Nc = 93$ . The poor efficiency of PIO is due to the inadequate local optimum.



(a) X fitness value

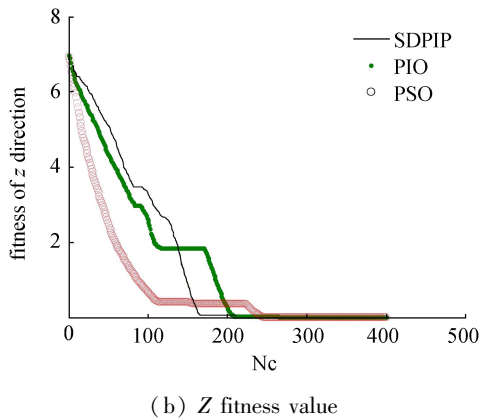


Fig.17 Fitness value in SD-PIO-based consensus

In Fig. 17 (b), all the fitness values show a tendency to decrease and SD-PIO is the slowest at the beginning. This problem is attributed to the effect of the slow diving strategy itself.

In order to avoid diving sharply, each UAV learns from its neighbor with the smallest velocity on the  $z$ -axis in Eq. (11), so the distance error between the desired position and the real one decreases at a slower rate. The UAVs performance and the efficiency of the algorithm have to be a compromise. But SD-PIO successfully eliminates the wrong local optimum and reaches a small value at  $N_c = 167$ . This clearly reflects the advantage of our improvement as PSO and PIO are trapped at  $N_c = 118$ . The two platforms in Fig. 17(b) demonstrate the local optimum and the algorithms temporarily stop evolution. However, all three optimizers manage to find the global minimum after  $N_c = 270$ .

## 5 Conclusions and future works

In this paper, consensus for multi-UAV formations is improved with a modified SD-PIO formula. PIO and PSO are adopted to optimize consensus for better performance of the initial flight trajectory. The method effectively reduces oscillation and smooths the curves. Consensus takes into effect subsequent responses and ensures that all the UAVs converge to the standard path. The combination of the two algorithms is proved to be useful for the multi-UAV formation problem.

Our future work will focus on the optimization of consensus. The method introduced is perhaps not the best choice and there is still room to explore better optimizations. However, the outstanding advantage of consensus is its distributed features, as most optimization methods demand global information. How to combine them while retaining their individual advantages remains a challenge.

## References:

- [1] RYAN A, ZENNARO M, HOWELL A, et al. An overview of emerging results in cooperative UAV control [C]//The proceedings of 43rd IEEE Conference on Decision and Control. atlantis, Paradise Island, Bahamas, 2004: 602–607
- [2] ZHA W, CHEN J, PENG Z. Dynamic multi-team antagonistic games model with incomplete information and its application to multi-UAV[J]. IEEE/CAA Journal of automatica sinica, 2015, 2(1): 74–84
- [3] CHAUMETTE S, LAPLACE R, MAZEL C. CARUS, an operational retasking application for a swarm of autonomous UAVs: first return on experience[C]//The proceedings of 2011 Military Communications Conference Track 5 Communications and Network Systems. Baltimore, Maryland, USA, 2011: 2003–2010.
- [4] RASCHE C, STERN C, KLEINJOHANN L, et al. A distributed multi-UAV path planning approach for 3D environments[C]//The proceedings of the 5th International Conference on Automation, Robotics and Applications. Wellington, New Zealand, 2011: 7–12.
- [5] GIULIETTI F, INNOCENTI M, NAPOLITANO M, et al. Dynamic and control issues of formation flight [J]. Aerospace science and technology, 2005, 9(1): 65–71.
- [6] CRUZ J. Leader-follower strategies for multilevel systems [J]. IEEE transactions on automatic control, 1978, 23(2): 244–255.
- [7] PAUL T, KROGSTAD T, GRAVDAHL J. Modelling of UAV formation flight using 3D potential fields[J]. Simulation modelling practice and theory, 2008, 16(9): 1240–1254.
- [8] RAO V, SINHA N. A sliding mode controller for aircraft simulated entry into spin [J]. Aerospace science and technology, 2013, 8(1): 154–163.
- [9] REN W, BEARD R, ATKINS E. Information consensus in multivehicle cooperative control: collective group behavior through local interaction[J]. IEEE control systems, 2007, 27(2): 71–82.
- [10] VICSEK T. Universal patterns of collective motion from minimal models of flocking[C]//The proceedings of the 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems. Venice, Italy, 2008: 3–11.
- [11] REN W, BEARD R W, MCLAIN T W. Coordination variables and consensus building in multiple vehicle systems [J]. Lecture notes in control and information sciences, 2005, 309: 439–442.
- [12] REN W, BEARD R W. Consensus seeking in multiagent systems under dynamically changing interaction topologies [J]. IEEE transactions on automatic control, 2005, 50(5): 655–661.
- [13] SORESENSEN N, REN W. A unified formation control scheme with a single or multiple leaders [C]//The proceedings of American Control Conference. New York City, USA, 2007: 5412–5418.
- [14] REN W, SORESENSEN N. Distributed coordination architecture

- for multi-robot formation control [ J ]. Robotics and autonomous systems, 2008, 56(4): 324–333.
- [ 15 ] DUAN H, QIAO P. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning [ J ]. International journal of intelligence computation and cybernetics, 2014, 7(1): 24–37.
- [ 16 ] GUILFORD T, ROBERTS S, BIRO D, et al. Positional entropy during pigeon homing II: navigational interpretation of Bayesian latent state models [ J ]. Journal of theoretical biology, 2004, 227(1): 25–38.
- [ 17 ] LI C, DUAN H. Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function [ J ]. Aerospace science and technology, 2014, 39: 352–360.
- [ 18 ] ZHANG B, DUAN H. Three-dimensional path planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment [ J ]. IEEE/ACM transactions on computation and biology bioinformaion, 2017, 14(1): 97–107
- [ 19 ] DUAN H, WANG X. Echo state networks with orthogonal pigeon-inspired optimization for image restoration [ J ]. IEEE transactions on neural networks and learning Systems, 2016, 27(11): 2413–2425.
- [ 20 ] ZHANG S, DUAN H. Gaussian Pigeon-inspired optimization approach to orbital spacecraft formation reconfiguration [ J ]. Chinese journal of aeronautics, 2015, 28(1): 200–205.
- [ 21 ] ZHAO J, ZHOU R. Pigeon-inspired optimization applied to constrained gliding trajectories [ J ]. Nonlinear dynamics, 2015, 82(4): 1781–1795.

作者简介:



ZHANG Tianjie was born in 1992. He received his Bachelor's degree from BUAA in 2015 and now he is a graduate student in BUAA. His main research focuses on aircraft mission planning.



DUAN Haibin was born in Shandong, China, in 1976. He received the Ph.D. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2005. He is currently a Full Professor with the School of Automation Science and Electrical Engineering, Beihang University (formerly Beijing University of Aeronautics and Astronautics), Beijing, China, where he is the Head of the Bio-inspired Autonomous Flight Systems Research Group. He has authored or coauthored more than 70 publications. His research interests are multiple unmanned-aerial-vehicle autonomous formation control and biological computer vision.