The Aeronautical Journal (2023), **127**, pp. 289–304
doi:10.1017/aer.2022.61

REGULAR PAPER

Neural network-based velocity-controllable UAV flocking

T. He and L. Wang*

Research Institute of Electronic Science and Technology, University of Electronic Science and Technology of China, Chengdu China
*Corresponding author. Email: wang_lei@uestc.edu.cn

Received: 26 February 2022; **Revised:** 27 April 2022; **Accepted:** 29 April 2022

Keywords: Unmanned aerial vehicle (UAV); Flocking; Multi-objective particle swarm optimisation (MOPSO); Neural network (NN); Obstacle avoidance

Abstract
The unmanned aerial vehicle (UAV) flocking among obstacles was transferred to a velocity-controllable UAV flocking problem, which means that multi-UAV gradually form and maintain the α-lattice geometry as they track the desired flocking velocity, and can be applied to tasks such as obstacle avoidance and velocity tracking. Velocity-controllable UAV flocking problem is a multi-objective flocking controller parameters optimisation problem, for which we design flocking velocity and geometry objective function, and solve them using a multi-objective particle swarm optimisation algorithm (MOPSO). On this basis, to address the problem that MOPSO has random results and long computation time, we propose to use a neural network (NN) to approximate the mathematical relationship between the UAV flocking state and the flocking controller parameters. We simulate the flight process of 5 and 49 UAVs performing obstacle avoidance and velocity tracking tasks, respectively. The results show that the proposed UAV flocking controller has better convergence performance, obtains reproducible results, reduces computation time, and can be used for large-scale UAV flocking control.

Nomenclature

N	number of UAVs
P^i	position vector of UAV i (m)
P^i_x	x component of position vector of UAV i (m)
P^i_y	y component of position vector of UAV i (m)
P^{core}	position vector of flocking core (m)
P^{core}_x	x component of position vector of flocking core (m)
P^{core}_y	y component of position vector of flocking core (m)
V^i	velocity vector of UAV i (m/s)
V^i_x	x component of velocity vector of UAV i (m/s)
V^i_y	y component of velocity vector of UAV i (m/s)
V^i_{xy}	horizontal airspeed of UAV i (m/s)
$V^{\min}_{xy}$	lower limits of horizontal airspeed (m/s)
$V^{\max}_{xy}$	upper limits of horizontal airspeed (m/s)
V^l_{xy}	allowable control error of horizontal airspeed (m/s)
V^e	desired velocity vector (m/s)
V^e_{xy}	desired horizontal airspeed (m/s)
efv^i	desired flocking velocity vector of UAV i (m/s)
efv^i_x	x component of desired flocking velocity vector of UAV i (m/s)
efv^i_y	y component of desired flocking velocity vector of UAV i (m/s)
ϕ^i	yaw angle of UAV i (rad)
ϕ^l	allowable control error of yaw angle (rad)

The two authors, Tao He and Lei Wang, are co-first authors.

ϕ^e	desired yaw angle (rad)
δ^i	desired flocking yaw angle of UAV i (rad)
δ^{core}	desired flocking yaw angle of flocking core (rad)
C_δ	parameter that controls the geometry of flocking
$n^{\max}$	maximum lateral overload (g)
g	gravitational acceleration (m/s^2)
u^i	control input vector of UAV i
u^i_x	x component of control input vector of UAV i
u^i_y	y component of control input vector of UAV i
u^i_f	flocking geometry control component of UAV i
u^i_{av}	horizontal airspeed alignment control component of UAV i
u^i_c	collision avoidance control component of UAV i
u^i_{vf}	flocking velocity control component of UAV i
C_f	strength coefficient of the flocking geometry control component
C_{av}	strength coefficient of the horizontal airspeed alignment control component
C_c	strength coefficient of the collision avoidance control component
C_{vf}	strength coefficient of the flocking velocity control component
W^i_j	influence weight of UAV j to UAV i for flocking geometry control
W^i_i	influence weight of UAV i to UAV i for flocking velocity control
$\hat{W}^i$	optimal influence weight vector of UAV i
d^{ij}	horizontal distance between UAV i and j (m)
D_c	maximum horizontal communication distance (m)
D_{l1}	minimum distance between UAVs to avoid collision (m)
D_{l2}	minimum distance allowed between the UAV and obstacle (m)
D_d	desired horizontal distance between UAVs(m)
Q^j	position vector of obstacle j (m)
Q^j_x	x component of position vector of obstacle j (m)
Q^j_y	y component of position vector of obstacle j (m)
R^j_o	radius of the obstacle j (m)
S_o	the set of obstacles within maximum perceived distance ahead of the UAV flocking in the desired velocity direction
$R_{formation}$	radius of the UAV flocking in the desired flocking yaw angle direction (m)
obj^i_1	flocking velocity objective function of UAV i
obj^i_2	flocking geometry objective function of UAV i
$input^i$	input vector i of NN
$label^i$	label vector i of NN

1.0 Introduction

Flocking is a collective behaviour that describes the phenomenon of all individuals being close to each other and moving at a common velocity. This phenomenon is universal in nature, such as jackdaw flocks [1], locust groups [2], fish school [3], honeybee colonies [4], etc. Animals in flock are more likely to avoid predators, which also makes them more conducive to getting more food. Flocking also bring many benefits to artificial agents (e.g. unmanned vehicles, autonomous ships and robots), especially UAVs. A UAV with single payload and limited ability to perform tasks cannot meet the demand. UAV flocking, on the other hand, can enhance system performance and efficiently accomplish more complex and larger missions by cooperating with each other and complementing each other's strengths. Hence, UAV flocking are widely used in both military and civilian fields [5, 6, 7, 8, 9, 10], while attracting the interest of a growing number of researchers.

Obstacle avoidance is a key problem that needs to be solved for safe UAV flocking flight, and numerous studies have been conducted to address this problem and various solutions have been proposed. In Ref. [11], an improved artificial potential field method is recommended, which uses k-means algorithm to divide space potential field into the inter-UAV potential field and the global potential field according to the action range of potential field internal force, thus optimising the calculation of attractive field between UAVs. It also introduces the attraction disturbance component of the target point and proposes a backtracking-filling method to solve the local minimum problem in the artificial potential fields. Dai et al. [12], use the geometric obstacle avoidance control method (GOACM) to plan a safe path, and the adaptive tracking control algorithm guarantees that the trajectory and velocity tracking errors converge to zero. Olfati-Saber [13] presents a flocking algorithm with obstacle avoidance capability by creating a virtual agent on the boundary of each nearby obstacle. In the work most closely related to this research, Qiu and Duan [14] modeled obstacle and collision avoidance as hard constraint, hoping to realise that the UAV moves more distance in the desired velocity direction in the process of changing direction for obstacle avoidance, but the hard constraints did not work in the whole flight process, and the velocity control performance is poor. Therefore, we transform the UAV flocking problem among obstacles into a velocity controllable UAV flocking problem, which means that multi-UAV gradually form and maintain the α-lattice [13] geometry as they track the desired flocking velocity (e.g. optimal velocity for obstacle avoidance in complex environments). It is a multi-objective optimisation problem, for which we design flocking velocity and geometry objective function, and solve them using a multi-objective particle swarm optimisation algorithm.

The study of flocking originated with the pioneering work of Reynolds [15] in 1987, who introduced three heuristic rules: aggregation, alignment and separation to realise the computer simulation of collective behaviour. Then, different flocking models [16, 17] and optimisation algorithms [18] have been developed successively. The literature on the design of UAV flocking controller can be divided into three categories: One is to manually complete the design of flocking controller according to the rules, and then optimise the relevant parameters of flocking controller to improve the security of the system and the performance of the algorithm [13, 14, 19, 20]. The second is to directly solve the flocking controller by using the optimal control theory [21, 22], which requires the minimum energy consumption with specific safety and task constraints. Third, a learning-from-data approach is used to design the controller autonomously, avoiding the high a prior knowledge requirements of manual design and also addresses, to some extent, control errors caused by incomplete problem models. In the literature [23, 24, 25, 26], a neural network term is added to the flocking controller to actively compensate for approximation errors in the non-linear dynamics model as well as uncertainties and external disturbances. The literature [27, 28, 29, 30], uses a neural network as a controller to achieve an end-to-end mapping of flocking states to control outputs. In this paper, the design approach for the flocking controller is a combination of the first and third approaches. As we all know, in order to optimise the performance of the hand-designed UAV flocking controller, it is usually necessary to optimise its parameters using intelligent optimisation algorithm. However, there is a randomness in the results calculated by the intelligent optimisation algorithm and the computational complexity is high. If the mathematical relationship between the UAV flocking state and the controller parameters can be found, then the performance of the controller will be improved. Therefore, in this paper, we use the powerful approximation capabilities of NN [31] and take the UAV flocking state as the input of training NN and the optimal flocking controller parameters calculated by MOPSO as the label of training NN. The trained NN successfully approximates the mathematical relationship between the UAV flocking state and the controller parameters.

The rest of the paper is organised as follows: UAV model, flocking model, obstacle avoidance and velocity tracking model are established in Section 2. Problem formulation is introduced in Section 3. Section 4 describes the UAV flocking control based on optimisation. Section 5 shows the simulation results in detail. Finally, the conclusion is presented in Section 6.

2.0 Model

2.1 UAV model

In this paper, N UAVs in a two-dimensional space are taken into consideration. The dynamic model of each UAV is described as follows [14]:

https://doi.org/10.1017/aer.2022.61 Published online by Cambridge University Press

$$\begin{cases} \dot{P}^i_x = V^i_{xy} \cos \phi^i, \\ \dot{P}^i_y = V^i_{xy} \sin \phi^i, \\ \dot{V}^i_{xy} = u^i_x \cos \phi^i + u^i_y \sin \phi^i, \\ \dot{\phi}^i = u^i_y \cos \phi^i - u^i_x \sin \phi^i, \end{cases} \tag{1}$$

where, for UAV i, $P^i = (P^i_x, P^i_y)$ is the position vector, $V^i = (V^i_x, V^i_y)$ is the velocity vector, V^i_{xy} is the horizontal airspeed, ϕ^i represents yaw angle, and $u^i = (u^i_x, u^i_y)$ is the control input vector. In addition, the following constraints must be met:

$$\begin{cases} V^{\min}_{xy} \le V^i_{xy} \le V^{\max}_{xy}, & \\ \left|\dot{\phi}^i\right| \le \frac{n^{\max} g}{V^i_{xy}}, & \\ V^i_{xy} = V^e_{xy}, & \text{if } \left|V^i_{xy} - V^e_{xy}\right| < V^l_{xy}, \\ \phi^i = \phi^e, & \text{if } \left|\phi^i - \phi^e\right| < \phi^l, \end{cases} \tag{2}$$

where, $n^{\max} = 10\text{g}$ is the maximum lateral overload, $g = 10\text{m/s}^2$ is the gravitational acceleration, $V^{\min}_{xy} = 5\text{m/s}$, $V^{\max}_{xy} = 15\text{m/s}$ are the lower and upper limits of horizontal airspeed, respectively, $V^l_{xy} = 0.25\text{m/s}$, $\phi^l = 0.1\text{rad}$ are allowable control error of horizontal airspeed, yaw angle, respectively, and $\phi^e = a\tan\left(\frac{V^e_y}{V^e_x}\right)$, $V^e = (V^e_x, V^e_y) = (10, 0)\text{m/s}$, V^e_{xy} are the desired yaw angle, desired velocity vector, desired horizontal airspeed, respectively.

2.2 Flocking model

To achieve velocity controllable UAV flocking, the following movement rules need to be guaranteed: (1) Each UAV stay close to its neighbours; (2) All UAVs fly at a common velocity; (3) UAVs avoid conflict with each other; (4) The flocking velocity of UAVs is controllable.

Therefore, control input of UAV i consists of four components: u^i_f is flocking geometry control component, which regulates the distance between UAV i and its neighbour. u^i_{av} is horizontal airspeed alignment control component, which regulates the velocity of UAV i to be consistent with its neighbour. u^i_c is collision avoidance control component, which regulates the safe distance among UAVs. u^i_{vf} is flocking velocity control component, which regulates the velocity of UAV i to be consistent with the desired flocking velocity. Details are as follows [14]:

$$u^i_k = u^i_f + u^i_{av} + u^i_c + u^i_{vf} - V^i_k, \quad k = x, y, \tag{3}$$

where, $k = x, y$ denotes the corresponding components in the x, y directions.

The flocking geometry control component u^i_f is as the following equation:

$$u^i_f = C_f \sum_{j\in\left\{d^{ij}\le D_c\right\}} W^i_j (P^j_k - P^i_k) ln\left(\frac{d^{ij}}{D_d}\right) \tag{4}$$

where, $C_f = 0.1$ denotes strength coefficient of the flocking geometry control component, W^i_j denotes influence weight of UAV j to UAV i for flocking geometry control within [0,1], $d^{ij} = \sqrt{(P^i_x - P^j_x)^2 + (P^i_y - P^j_y)^2}$ denotes the horizontal distance between UAV i and j, $D_c = 20\text{m}$ denotes the maximum horizontal communication distance, $D_d = 10\text{m}$ denotes the desired horizontal distance between UAVs.

The horizontal airspeed alignment control component u^i_{av} is as the following equation:

$$u^i_{av} = C_{av} \sum_{j\in\left\{d_{ij}\le D_c\right\}} W^i_j (V^j_k - V^i_k) \tag{5}$$

where, $C_{av} = 0.1$ denote strength coefficient of the horizontal airspeed alignment control component.

https://doi.org/10.1017/aer.2022.61 Published online by Cambridge University Press

The collision avoidance control component u_c^i is as the following equation:

$$u_c^i = C_c \sum_{j \in \{d^{ij} \le D_{l1}\}} \left(\frac{1}{d^{ij}} - \frac{1}{D_{l1}} \right)^2 \frac{P_k^i - P_k^j}{d^{ij}} \tag{6}$$

where, $C_c = 10^5$ denote strength coefficient of the collision avoidance control component, $D_{l1} = 2\text{m}$ denotes the minimum distance between UAVs to avoid collision.

The flocking velocity control component u_{vf}^i is as the following equation:

$$u_{vf}^i = C_{vf} W_i^i efv_k^i \tag{7}$$

where, $C_{vf} = 1$ denotes strength coefficient of the flocking velocity control component, W_i^i denotes influence weight of UAV i to UAV i for flocking velocity control within $[1, 1.1]$, $efv^i = (efv_x^i, efv_y^i) = (V_{xy}^e \cos \delta^i, V_{xy}^e \sin \delta^i)$ denotes the desired flocking velocity vector of UAV i, and δ^i denotes the desired flocking yaw angle of UAV i.

2.3 Obstacle avoidance and velocity tracking model

Because the magnitude of the desired velocity remains constant throughout the flight, obstacle avoidance and velocity tracking can be achieved by setting an appropriate desired flocking yaw angle δ^i which is calculated as follows:

$$\delta^i = a\tan \left(\frac{P_y^{core} + C_\delta \sin(\delta^{core}) - P_y^i}{P_x^{core} + C_\delta \cos(\delta^{core}) - P_x^i} \right) \tag{8}$$

$$P_k^{core} = \frac{1}{N} \sum_{i=1}^{N} P_k^i, \quad k = x, y, \tag{9}$$

where $P^{core} = (P_x^{core}, P_y^{core})$ is position vector of flocking core calculated by Equation (9), C_δ is the parameter that controls the geometry of flocking, δ^{core} is the desired flocking yaw angle of flocking core, which is directly set as the reference desired flocking yaw angle in the velocity tracking process, but its equation is as follows in the obstacle avoidance process:

$$\delta^{core} = \begin{cases} a\tan \left(\dfrac{P_y^A - P_y^{core}}{P_x^A - P_x^{core}} \right), & \text{if } S_o \neq \emptyset, \\ a\tan \left(\dfrac{V_y^e}{V_x^e} \right), & \text{else,} \end{cases} \tag{10}$$

where S_o is the set of obstacles within maximum perceived distance ahead of the UAV flocking in the desired velocity direction, as shown in Fig. 1. A is the point closest to the flocking core among the two edge points projected in the vertical direction of V^e by the envelope of the obstacle closest to the flocking core in S_o, $D_{l2} = 10\text{m}$ is the minimum distance allowed between the UAV and obstacle, $R_{formation}$ is radius of the UAV flocking in the desired flocking yaw angle direction. Note that although the UAV flocking bypass the current obstacle, it cannot bypass the next obstacle until it passes through the current obstacle. The maximum perceived distance of UAV is 105.

3.0 Problem formulation

According to the above model, the influence weight vector $W^i = (W_1^i, W_2^i, \ldots, W_N^i)$ directly determines the next state of the UAV i. Therefore, the UAV flocking control problem is transformed into solving the

https://doi.org/10.1017/aer.2022.61 Published online by Cambridge University Press

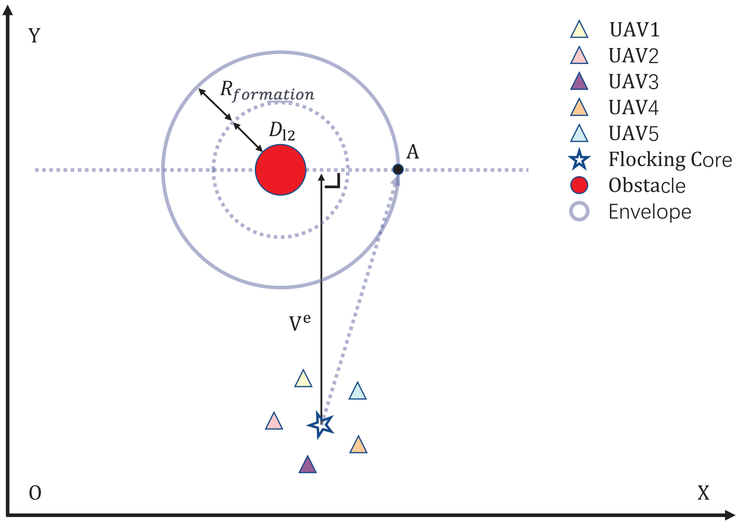

***Figure 1.** Obstacle avoidance model.*

optimal influence weight vector $\hat{W}^i$ of UAV i which satisfies the following objective functions:

$$\begin{cases} \min \quad obj^i = [obj_1^i, obj_2^i], \\ s.t. \quad \text{Equations (1)–(8)}, \end{cases} \tag{11}$$

Flocking velocity objective function obj_1^i indicates how far the actual velocity deviates from the desired flocking velocity. The equation is as follows:

$$obj_1^i = \begin{cases} -\dfrac{(V_x^i, V_y^i).(efv_x^i, efv_y^i)}{2\,\|efv^i\|}, & \text{if} \quad efv^i \neq V^e \text{ or } S_o \neq \emptyset, \\ \sum_{k=x,y} \left| efv_k^i - V_k^i \right|, & \text{else}, \end{cases} \tag{12}$$

Where, $efv^i \neq V^e$ indicates the velocity tracking process, and $S_o \neq \emptyset$ represents obstacle avoidance process. In the process, the velocity of UAV i is not required to be exactly the same as the desired flocking velocity, but is allowed to change slightly to adapt to the process as soon as possible.

Flocking geometry objective function obj_2^i describes the degree of UAV flocking and the alignment performance with neighbours velocities. The equation is as follows:

$$obj_2^i = \sum_{j \in \left\{ d^{ij} \leq D_c \right\}} \left[\left| D_d - d^{ij} \right| + \sum_{k=x,y} \left| V_k^i - V_k^j \right| \right] \tag{13}$$

4.0 UAV Flocking control based on optimisation

4.1 UAV flocking obstacle avoidance and velocity tracking based on MOPSO

UAV flocking obstacle avoidance and velocity tracking based on MOPSO is a special case of UAV flocking control framework (see Algorithm 1). It uses MOPSO [32] to calculate the optimal influence weight vector $\hat{W}^i$, The specific steps of the MOPSO are as follows:

Input: Objective function obj_1^i and obj_2^i, maximum generation M_G, number of particles N_P, and the upper and lower bounds of L_g^h and S_g^h are the same as those of influence weight vector. Where, L_g^h and S_g^h are position vector and velocity vector of the generation g of particle h respectively, $h = 1, 2, \ldots, N_P$, $g = 1, 2, \ldots, M_G$, L_g^h is a feasible influence weight vector.

Algorithm 1 UAV flocking control framework

```
Initialise $t = 0$, number of UAVs $N$, state of UAV $i$ ($P^i$, $V^i = V^e$, $\phi^i = 0$), state of obstacles,
  maximum simulation time $T_{\max}$, sampling time ts;
while $t < T_{\max}$ do
    for: $i = 1$ to $N$ do
        Calculate $\delta^i$ by Equation (8);
        Calculate the optimal influence weight vector $\hat{W}^i$;
        Calculate control input vector $u^i$ by Equations (3)–(7);
        Calculate the next state by Equations (1)–(2);
        $i = i + 1$
    end for
    $t = t + ts$
end while
Output all UAV states
```

Initialisation: Initialise the L_g^h and S_g^h of N_P particles within their upper and lower bounds respectively, current iteration $g = 1$.

Step 1: Calculate the objective function obj_1^i and obj_2^i of all the particles by Equations (1)–(8) and (12)–(13).

Step 2: Rank the particles positions in the set L_g by Pareto sorting scheme, store the positions of particles in the Pareto frontier in the repository REP.

Step 3: $pbest^h = L_1^h$.

Step 4: *gbest* is selected randomly from REP.

Step 5: Update the velocity of each particle by the following equation.

$$S_{g+1}^h = aS_g^h + r_1^T(pbest^h - L_g^h) + r_2^T(gbest - L_g^h) \tag{14}$$

Where, $a = 0.4$ is the inertia weight, r_1 and r_2 are random number vectors uniformly distributed in the interval [0,1].

Step 6: Update the position of each particle by the following equation.

$$L_{g+1}^h = L_g^h + V_{g+1}^h \tag{15}$$

Step 7: Initialises the state of the particle if its position exceeds the upper and lower bounds.

Step 8: Calculate the objective function obj_1^i and obj_2^i of all the particles by Equations (1)–(8) and (12)–(13).

Step 9: If L_{g+1}^h is better than $pbest^h$, then $pbest^h = L_{g+1}^h$.

Step 10: Rank the particles positions in the set L_{g+1} and REP by Pareto sorting scheme, store the positions of particles in the Pareto frontier in the repository REP.

Step 11: The current iteration $g = g + 1$. If $g \leq M_G$, then return to Step 4. Otherwise, output the optimal influence weight vector which is the position of particle in the REP whose objective function obj_2^i has the smallest value.

4.2 UAV flocking obstacle avoidance and velocity tracking based on NN

UAV flocking obstacle avoidance and velocity tracking based on NN is a special case of UAV flocking control framework(see Algorithm 1). It uses NN to calculate the optimal influence weight vector $\hat{W}^i$, and the NN is generated as follows.

Generate training samples: The UAV flocking velocity tracking algorithm based on MOPSO is repeated $N_{repeat} = 100$ times to obtain the training sample set $\{input^i, label^i\}_{i=1}^{N_{repeat}\times\frac{T_{\max}}{ts}}$. Where, $input^i$ is the input vector i and expressed by Equation (17), $label^i$ is the label vector i and expressed by Equation (18), P_x^i, P_y^i are random numbers within [0,30], [75,150] respectively, but horizontal distance

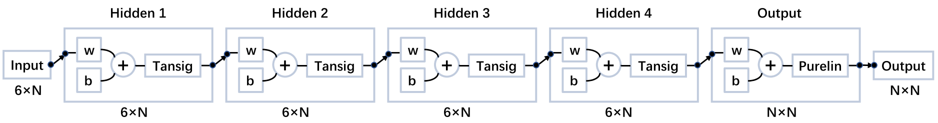

***Figure 2.** Neural network structure.*

between any two UAVs should be less than maximum horizontal communication distance of D_c, number of UAVs $N=5$, maximum simulation time $T_{\max}=50$s, sampling time ts=0.5s, $C_\delta=\infty$, δ^{core} is expressed by Equation (16).

$$\delta^{core}=\begin{cases}\frac{\pi}{3}\sin\left(\frac{2\pi t}{0.8T_{\max}}\right), & \text{if } t\le 0.8T_{\max},\\ a\tan\left(\frac{V_y^e}{V_x^e}\right), & \text{if } t>0.8T_{\max},\end{cases} \tag{16}$$

$$input^i=\Big(\underbrace{P_x^1,P_y^1,V_x^1,V_y^1,efv_x^1,efv_y^1,}_{State\ of\ UAV\ 1}\\ \underbrace{P_x^2,P_y^2,V_x^2,V_y^2,efv_x^2,efv_y^2,}_{State\ of\ UAV\ 2}\\ \ldots,\\ \underbrace{P_x^N,P_y^N,V_x^N,V_y^N,efv_x^N,efv_y^N}_{State\ of\ UAV\ N}\Big) \tag{17}$$

$$label^i=\Big(\underbrace{\hat{W}_1^1,\hat{W}_2^1,\ldots,\hat{W}_N^1}_{Optimal\ influence\ weight\ of\ UAV\ 1},\\ \underbrace{\hat{W}_1^2,\hat{W}_2^2,\ldots,\hat{W}_N^2}_{Optimal\ influence\ weight\ of\ UAV\ 2},\\ \ldots,\\ \underbrace{\hat{W}_1^N,\hat{W}_2^N,\ldots,\hat{W}_N^N}_{Optimal\ influence\ weight\ of\ UAV\ N}\Big) \tag{18}$$

Neural network structure: NN has four hidden layers, each layer has 6×N neurons, and the dimensions of the input and output layers are 6 × N and N × N, respectively. Its structure is shown in Fig. 2. Where, the activation function of hidden layer, output layer are shown in Equations (19), (20), respectively.

$$Tansig(o)=\frac{2}{1+exp(-2o)}-1 \tag{19}$$

$$Purelin(o)=o \tag{20}$$

https://doi.org/10.1017/aer.2022.61 Published online by Cambridge University Press

Training neural network: Each dimension of the input data is normalised to $[-1, 1]$ before training, the loss function is the mean square error function is expressed by Equation (21), the training function is trainscg (scaled conjugate gradient backpropagation), epochs=1,000, and The ratio between training set, validation set and test set is 7:1.5:1.5.

$$mse = \frac{1}{N_d}\sum_{k=1}^{N_d}(label_k - output_k)^2 \tag{21}$$

4.3 Analysis of generalisation capability of UAV flocking control based on NN

We discuss the generalisation capability of UAV flocking control based on NN from the following two aspects:

Position of UAVs: Deviation occurs in the calculation result when UAV position exceeds min-max normalisation range of the NN input vector. This problem can be solved by moving all UAVs in the flocking to min-max normalisation range without changing their relative positions.

Number of UAVs: The input vector of NN only supports the state of k UAVs, and can not be directly applied to larger scale UAV flocking control. This problem can be solved by distributed computing, that is, each UAV only calculates its next state according to the nearest k-1 UAV and its own state.

5.0 Simulation results

In this section, taking obstacle avoidance and velocity tracking as application scenarios, two numerical simulation examples are designed to verify the effectiveness of our proposed algorithm. The first simulation shows UAV flocking obstacle avoidance and velocity tracking, and the second simulation shows generalisation capability of UAV flocking control based on NN. All simulations were conducted on laptops running Windows 10 and MATLAB R2020a with Intel(R) Core(TM) i5-10210U CPU and 16GB memory.

5.1 Simulation of UAV flocking obstacle avoidance and velocity tracking

In the simulation, five UAVs form a flocking, they first spend 50s crossing obstacles, then track the velocity for 50s. Initial parameters of UAVs and obstacles are shown in Table 1, and other initial parameters are as follows: maximum generation MG = 58, number of particles NP = 20, maximum simulation time $T_{\max} = 100$s, sampling time ts=0.5s, $C_\delta = \infty$, and calculate δ^{core} by Equation (22).

$$\delta^{core} = \begin{cases} -\frac{\pi}{3}\sin\left(\frac{2\pi(t-50)}{30}\right), & \text{if } t > 50 \text{ and } t \le 80, \\ a\tan\left(\frac{V_y^e}{V_x^e}\right), & \text{if } t > 80, \end{cases} \tag{22}$$

Figure 3 depicts the simulation results of five UAVs flocking obstacle avoidance and velocity tracking based on NN where, RD_{ij} denotes the relative distance between UAV i and UAV j, D_{l1} is the minimum distance to avoid collision between UAVs. As shown in Fig. 3(a), five UAVs successfully completed obstacle avoidance and velocity tracking tasks and flew out of a smooth trajectory curve. As shown in Fig. 3(b), the difference of horizontal airspeed among UAVs was less than 1.5480m/s, and the horizontal airspeed of the UAV flocking quickly converges to the desired horizontal airspeed after fluctuating within the allowable range. As shown in Fig. 3(c), the UAV flocking has a good velocity tracking effect, and a slight yaw angle difference. As shown in Fig. 3(d), relative distance between UAVs is converges rapidly

https://doi.org/10.1017/aer.2022.61 Published online by Cambridge University Press

Table 1. *Initial parameters of the UAVs and obstacles*

	i	P_x^i(m)	P_y^i(m)	V^i(m/s)
UAV	1	22.0222	107.9526	(10, 0)
	2	25.5856	114.5043	(10, 0)
	3	24.6951	116.9755	(10, 0)
	4	10.5040	113.8653	(10, 0)
	5	8.4631	120.6620	(10, 0)
	j	Q_x^j(m)	Q_y^j(m)	R_o^j(m)
Obstacle	1	100	40	5
	2	230	40	5
	3	165	75	5
	4	295	75	5
	5	100	110	5
	6	230	110	5
	7	165	145	5
	8	295	145	5
	9	100	180	5
	10	230	180	5
	11	165	215	5
	12	295	215	5

and fluctuates less and always greater than the minimum distance to avoid collision between UAVs. As shown in Fig. 3(e) and (f), the geometry and velocity of the UAV flocking converge rapidly and stably.

To illustrate the superiority of the UAV flocking obstacle avoidance and velocity tracking based on NN, we compare it with the UAV flocking obstacle avoidance and velocity tracking based on the MOPSO. The statistical results of the two algorithms are shown in Table 2. From the data in the table, the NN-based algorithm achieves faster convergence of horizontal airspeed and yaw angle, especially the simulation time has been greatly reduced. The simulation results of the UAV flocking obstacle avoidance and velocity tracking based on the MOPSO is shown in Fig. 4. From the comparison between Figs 3 and 4, it can be seen more intuitively that the experimental results based on NN have smaller fluctuation, faster convergence, and better obstacle avoidance and velocity tracking effects.

5.2 Simulation of generalisation capability of UAV flocking control based on NN

In the simulation, we adopt the neural network for 5 UAVs in the previous experiment to verify its generalization capability in a flocking of 49 UAVs, which first track the velocity for 60s, then spend 60s crossing obstacles, and finally track the velocity for 60s. The initial velocity of the UAV and radius of obstacle is the same as that shown in Table 1, and the positions of UAVs and obstacles have been moved away from where they were during training NN (due to the large amount of data, no specific data is given here). Other initial parameters are as follows: maximum simulation time $T_{\max} = 180$s, sampling time ts=0.5s, calculate C_δ by Equation (23), calculate δ_1^{core} for the first velocity tracking by Equation (24), calculate δ_2^{core} for the second velocity tracking by Equation (25).

$$C_\delta = \begin{cases} 80, & \text{if } S_o \neq \emptyset, \\ \infty, & \text{else,} \end{cases} \tag{23}$$

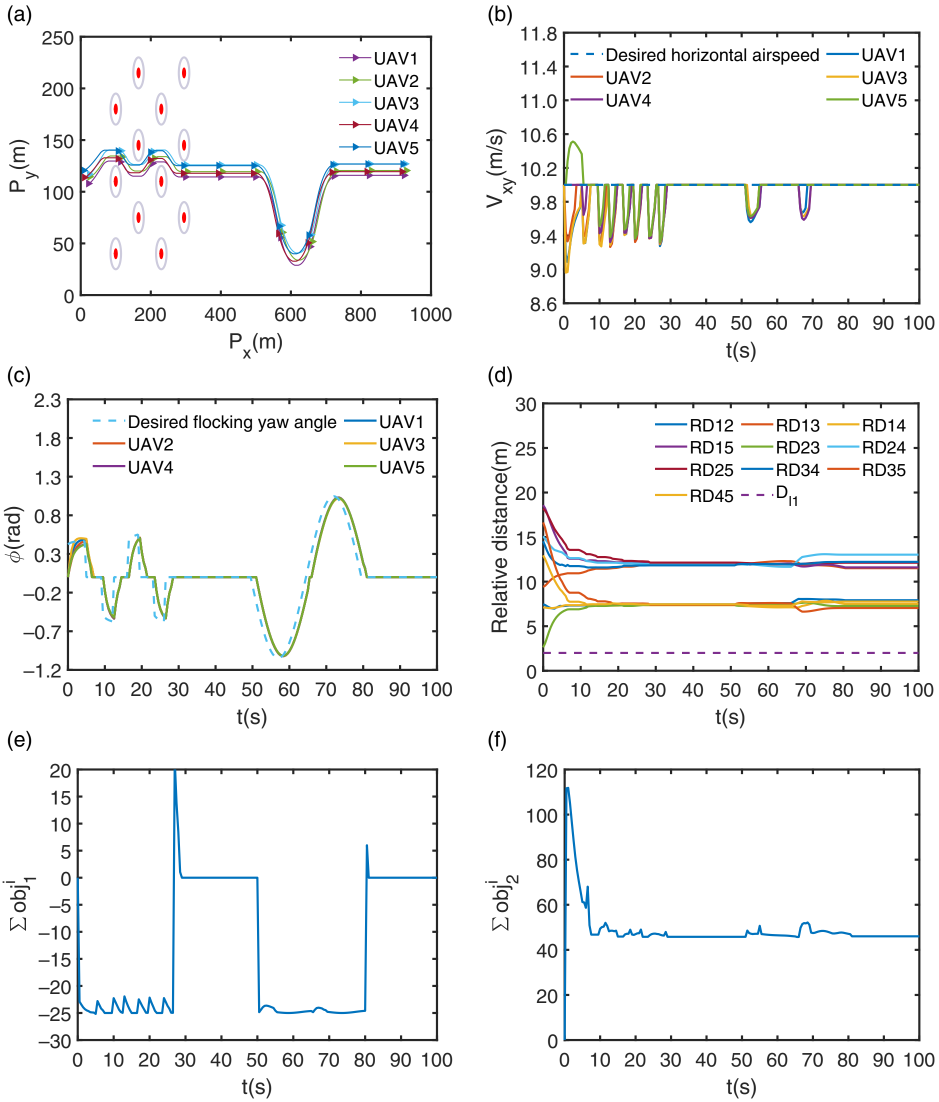

Figure 3. *Simulation results of 5 UAV flocking obstacle avoidance and velocity tracking based on NN: (a) trajectory curves, (b) horizontal airspeed curves, (c) yaw angle curves, (d) relative distance curves, (e) $\sum obj_1^i$ curves, and (f) $\sum obj_2^i$ curves.*

$$\delta_1^{core} = \begin{cases} \frac{\pi}{3}\sin\left(\frac{2\pi t}{40}\right), & \text{if } t \le 40, \\ a\tan\left(\frac{V_y^e}{V_x^e}\right), & \text{if } t > 40 \quad \text{and} \quad t < 50, \end{cases} \tag{24}$$

$$\delta_2^{core} = \begin{cases} -\frac{\pi}{3}\sin\left(\frac{2\pi(t-120)}{40}\right), & \text{if } t > 120 \quad \text{and} \quad t \le 160, \\ a\tan\left(\frac{V_y^e}{V_x^e}\right), & \text{if } t > 160, \end{cases} \tag{25}$$

https://doi.org/10.1017/aer.2022.61 Published online by Cambridge University Press

Table 2. *Statistical results of two algorithms for 5 UAVs*

Algorithm	CTOHA^a(s)	CTOYA^b(s)	TST^c(s)
MOPSO	86.0	66.0	785.4
NN	69.5	7.0	3.8

[a] Convergence time of horizontal airspeed.
[b] Convergence time of yaw angle.
[c] Total simulation time.

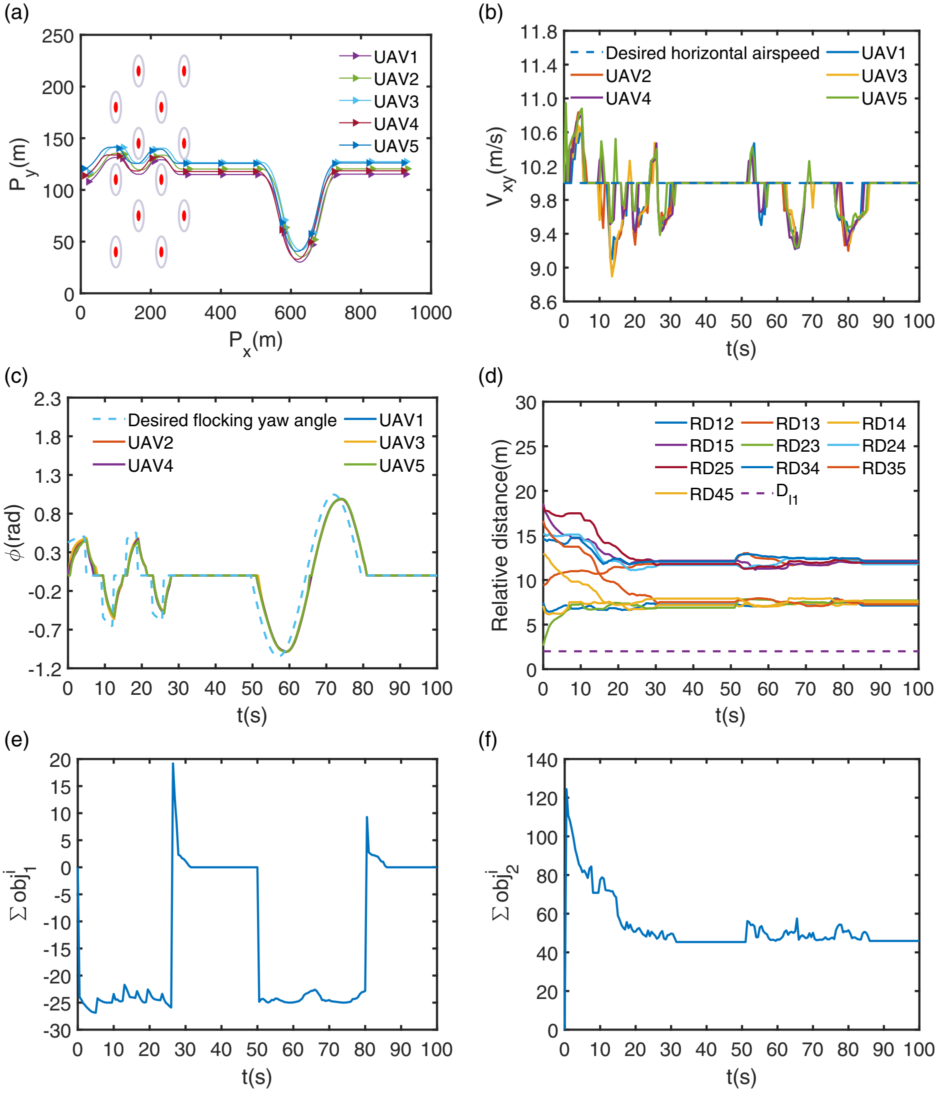

Figure 4. *Simulation results of 5 UAV flocking obstacle avoidance and velocity tracking based on MOPSO: (a) trajectory curves, (b) horizontal airspeed curves, (c) yaw angle curves, (d) relative distance curves, (e) $\sum obj_1^i$ curves, and (f) $\sum obj_2^i$ curves.*

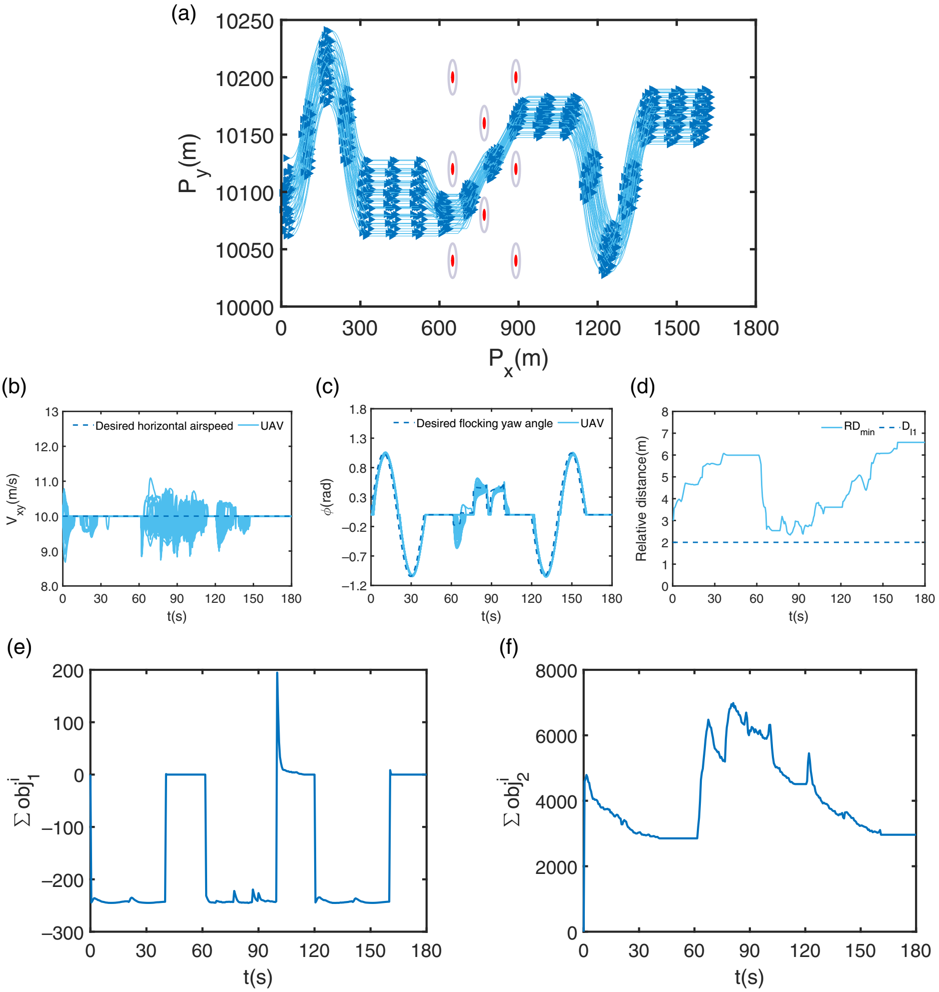

***Figure 5.** Simulation results of 49 UAV flocking obstacle avoidance and velocity tracking based on NN: (a) trajectory curves, (b) horizontal airspeed curves, (c) yaw angle curves, (d) relative distance curves, (e) $\sum obj_1^i$ curves, and (f) $\sum obj_2^i$ curves.*

Figure 5 depicts the simulation results of 49 UAV flocking obstacle avoidance and velocity tracking based on NN. where, $RD_{\min}$ is the minimum relative distance between UAVs. As shown in Fig. 5(a), the UAV flocking can safely pass through the obstacle area through geometric transformation, and the UAV position exceeds min-max normalisation range of the NN input vector, but it has no adverse impact on flight. As shown in Fig. 5(b), the horizontal airspeed of the UAV flocking quickly converges to the desired horizontal airspeed after fluctuating within the allowable range. As shown in Fig. 5(c), the UAV flocking has a good velocity tracking effect, and the yaw angle quickly converges to the desired yaw angle after fluctuations. As shown in Fig. 5(d), during the whole flight, the minimum relative distance between UAVs is always greater than the minimum distance to avoid collision between UAVs. As shown in Fig. 5(e) and (f), the geometry and velocity of the UAV flocking converge rapidly and stably. To sum up, UAV flocking controller based on NN has good generalisation capability and can control the flocking geometry to a certain extent.

https://doi.org/10.1017/aer.2022.61 Published online by Cambridge University Press

Table 3. Statistical results of two algorithms for 49 UAVs

Algorithm	CTOHA[a](s)	CTOYA[b](s)	TST[c](s)
MOPSO	162.5	161.0	25, 847.2
NN	147.5	161.0	317.99

[a] Convergence time of horizontal airspeed.
[b] Convergence time of yaw angle.
[c] Total simulation time.

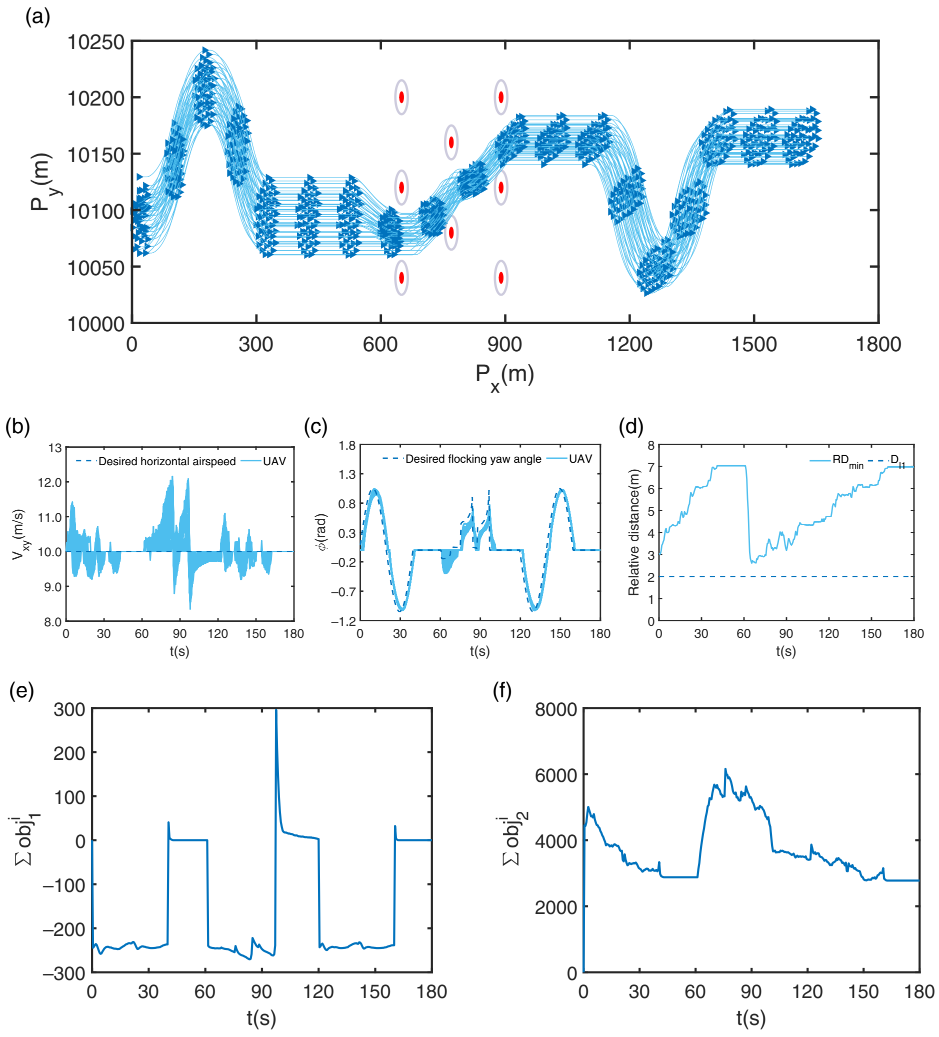

***Figure 6.** Simulation results of 49 UAV flocking obstacle avoidance and velocity tracking based on MOPSO: (a) trajectory curves, (b) horizontal airspeed curves, (c) yaw angle curves, (d) relative distance curves, (e) $\sum obj_1^i$ curves, and (f) $\sum obj_2^i$ curves.*

To illustrate the superiority of the UAV flocking obstacle avoidance and velocity tracking based on NN, we compare it with the UAV flocking obstacle avoidance and velocity tracking based on the MOPSO. The statistical results of the two algorithms are shown in Table 3. From the data in the table, the NN-based algorithm achieves faster convergence of horizontal airspeed, especially the simulation time has been greatly reduced. The simulation results of the 49 UAV flocking obstacle avoidance and velocity tracking based on the MOPSO is shown in Fig. 6. From the comparison between Figs 5(a) and 6(a), it can be seen that in the NN-based experimental results the UAV is further away from the obstacle and safer. The comparison of Figs 5(b) and 6(b) shows that the horizontal airspeed deviates from the desired value in a shorter time, smaller amplitude, more regular, and achieves convergence earlier in the NN-based experimental results. Contrast Fig. 5(c) with Fig. 6(c) indicates that the NN-based method has a fast response to changes in yaw angle. In conclusion, the experimental results based on NN have smaller fluctuation, faster convergence, and better obstacle avoidance and velocity tracking effects.

6.0 Conclusions

In this paper, we propose a method to design UAV flocking controller based on NN. The designed controller has the following advantages: (i) It has good generalisation capability and can be directly used in UAV flocking control with larger scale and wider flight range. (ii) The UAV flocking convergence and velocity controllability can be achieved simultaneously in the environment with and without obstacles. (iii) The trained NN successfully approximates the mathematical relationship between the UAV flocking state and the flocking controller parameters. It not only improves the quality of solution and obtains repeatable results, but also reduces computation time.

Nevertheless, there are still some drawbacks in this paper: (i) The structure of NN is determined by simple algorithm without optimisation. (ii) The quality of training samples is not carefully evaluated. In the future, we will continue to deepen the research on UAV flocking technology, and ultimately verify the feasibility of the algorithm in the actual flight environment.

Acknowledgements. This research was funded by the Sichuan Science and Technology Program (No. 2020YFG0059, 2019YJ0195) and the Fundamental Research Funds for the Central Universities (No. ZYGX2019J128).

References

[1] Ling, H., Mclvor, G.E., Westley, J., van der Vaart, K., Vaughan, R.T., Thornton, A. and Ouellette, N.T. Behavioural plasticity and the transition to order in jackdaw flocks, *Nat. Commun.*, 2019, **10**, (1), pp 5174. https://doi.org/10.1038/s41467-019-13281-4

[2] Dkhili, J., Berger, U., Idrissi Hassani, L.M., Ghaout, S., Peters, R. and Piou, C. Self-organized spatial structures of locust groups emerging from local interaction, *Ecol. Modell.*, 2017, **361**, pp 26–40. https://doi.org/10.1016/j.ecolmodel.2017.07.020

[3] Becco, C., Vandewalle, N., Delcourt, J. and Poncin, P. Experimental evidences of a structural and dynamical transition in fish school, *Phys. A Stat. Mech. its Appl.*, 2006, **367**, pp 487–493. https://doi.org/10.1016/j.physa.2005.11.041

[4] Arvin, F., Samsudin, K., Ramli, A.R. and Bekravi, M. Imitation of honeybee aggregation with collective behavior of swarm robots, *Int. J. Comput. Intell. Syst.*, 2011, **4**, pp 739–748. https://doi.org/10.1080/18756891.2011.9727825

[5] McGuire, K.N., De Wagter, C., Tuyls, K., Kappen, H.J. and de Croon, G.C.H.E. Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment, *Sci. Robot.*, 2019, **4**, (35), pp eaaw9710. https://doi.org/10.1126/scirobotics.aaw9710

[6] Schranz, M., Umlauft, M., Sende, M. and Elmenreich, W. Swarm robotic behaviors and current applications, *Front. Robot. AI*, 2020, 7. https://doi.org/10.3389/frobt.2020.00036

[7] Poljak, M. and Återbenc, A. Use of drones in clinical microbiology and infectious diseases: current status, challenges and barriers, *Clin. Microbiol. Infect.*, 2020, **26**, (4), pp 425–430. https://doi.org/10.1016/j.cmi.2019.09.014

[8] Park, S. and Choi, Y. Applications of unmanned aerial vehicles in mining from exploration to reclamation: a review, *Minerals*, 2020, **10**, pp 663. https://doi.org/10.3390/min10080663

[9] Pham, H.X., La, H.M., Feil-Seifer, D. and Deans, M.C. A distributed control framework of multiple unmanned aerial vehicles for dynamic wildfire tracking, *IEEE Trans. Syst. Man Cybern. Syst.*, 2020, **50**, (4), pp 1537–1548. https://doi.org/10.1109/TSMC.2018.2815988

[10] Márquez-Vega, L.A., Aguilera-Ruiz, M. and Torres-Treviño, L.M. Multi-objective optimisation of a quadrotor flock performing target zone search, *Swarm Evol. Comput.*, 2021, **60**, pp 100733. https://doi.org/10.1016/j.swevo.2020.100733

[11] Wu, E., Sun, Y., Huang, J., Zhang, C. and Li, Z. Multi UAV cluster control method based on virtual core in improved artificial potential field, *IEEE Access*, 2020, **8**, pp 131647–131661. https://doi.org/10.1109/ACCESS.2020.3009972

[12] Dai, Y., Kim,Y.G., Wee, S.G., Lee, D.H. and Lee, S.G. A switching formation strategy for obstacle avoidance of a multi-robot system based on robot priority model, *ISA Trans.*, 2020, **56**, pp 123–134. https://doi.org/10.1016/j.isatra.2014.10.008

[13] Olfati-Saber, R. Flocking for multi-agent dynamic systems: algorithms and theory, *IEEE Trans. Automat. Contr.*, 2006, **51**, (3), pp 401–420. https://doi.org/10.1109/TAC.2005.864190

https://doi.org/10.1017/aer.2022.61 Published online by Cambridge University Press

[14] Qiu, H. and Duan, H. A multi-objective pigeon-inspired optimisation approach to UAV distributed flocking among obstacles, *Inf. Sci.*, 2020, **509**, pp 515–529. https://doi.org/10.1016/j.ins.2018.06.061

[15] Reynolds, C.W. Flocks, herds, and schools: a distributed behavioral model, Proceedings of SIGGRAPH'87, 1987, vol. **21**, pp 25–34. https://doi.org/10.1145/280811.281008

[16] Fine, B.T. and Shell, D.A. Unifying microscopic flocking motion models for virtual, robotic, and biological flock members, *Auton. Robots*, 2013, **35**, (2–3), pp 195–219. https://doi.org/10.1007/s10514-013-9338-z

[17] Zhu, B., Xie, L., Han, D., Meng, X. and Teo, R. A survey on recent progress in control of swarm systems, *Sci. China Inf. Sci.*, 2017, **60**, (7), pp 1–24. https://doi.org/10.1007/s11432-016-9088-2

[18] Beaver, L.E. and Malikopoulos, A.A. An overview on optimal flocking, *Annu. Rev. Control*, 2021, **51**, pp 88–99. https://doi.org/10.1016/j.arcontrol.2021.03.004

[19] Vásárhelyi, G., Virágh, C., Somorjai, G., Nepusz, T., Eiben, A.E. and Vicsek, T. Optimized flocking of autonomous drones in confined environments, *Sci. Robot.*, 2018, **3**, (20), pp 1–14. https://doi.org/10.1126/scirobotics.aat3536

[20] Zhao, W., Chu, H., Zhang, M., Sun, T. and Guo, L. Flocking control of fixed-wing UAVs with cooperative obstacle avoidance capability, *IEEE Access*, 2019, **7**, pp 17798–17808. https://doi.org/10.1109/ACCESS.2019.2895643

[21] Beaver, L.E. and Malikopoulos, A.A. Beyond Reynolds: a constraint-driven approach to cluster flocking, 2020 59th IEEE Conference on Decision and Control (CDC), 2020, pp 208–213. https://doi.org/10.1109/CDC42340.2020.9304333

[22] Beaver, L.E., Kroninger, C. and Malikopoulos, A.A. An optimal control approach to flocking, 2020 American Control Conference (*ACC*), 2020, pp 683–688. https://doi.org/10.23919/ACC45564.2020.9147311

[23] Dierks, T., Brenner, B. and Jagannathan, S. Neural network-based optimal control of mobile robot formations with reduced information exchange, *IEEE Trans. Control Syst. Technol.,* 2013, **21**, (4), pp 1407–1415. https://doi.org/10.1109/TCST.2012.2200484

[24] Yu, J., Ji, J., Miao, Z. and Zhou, J. Neural network-based region reaching formation control for multi-robot systems in obstacle environment, *Neurocomputing*, 2019, **333**, pp 11–21. https://doi.org/10.1016/j.neucom.2018.12.051

[25] Liu, Y., Huang, P., Zhang, F. and Zhao, Y. Distributed formation control using artificial potentials and neural network for constrained multiagent systems, *IEEE Trans. Control Syst. Technol.,* 2020, **28**, (2), pp 697–704. https://doi.org/10.1109/TCST.2018.2884226

[26] Yang, S., Bai, W., Li, T., Shi, Q., Yang, Y., Wu, Y. and Chen, C.L.P. Neural-network-based formation control with collision, obstacle avoidance and connectivity maintenance for a class of second-order nonlinear multi-agent systems, *Neurocomputing*, 2021, **439**, pp 243–255. https://doi.org/10.1016/j.neucom.2020.12.106

[27] Nguyen, T.T., Nguyen, N.D. and Nahavandi, S. Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications, *IEEE Trans. Cybern.,* 2020, **50**, (9), pp 3826–3839. https://doi.org/10.1109/TCYB.2020.2977374

[28] Xu, D. and Chen, G. Autonomous and cooperative control of UAV cluster with multi-agent reinforcement learning, *Aeronaut. J.*, 2022, pp 1–20. https://doi.org/10.1017/aer.2021.112

[29] Hu, T.K., Gama, F., Chen, T., et al. Vgai: end-to-end learning of vision-based decentralized controllers for robot swarms, *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp 4900–4904. https://doi.org/10.1109/ICASSP39728.2021.9414219

[30] Schilling, F., Lecoeur, J., Schiano, F., et al. Learning vision-based flight in drone swarms by imitation, *IEEE Robot. Autom. Lett.*, 2019, **4**, (4), pp 4523–4530. https://doi.org/10.1109/LRA.2019.2935377

[31] Hornik, K. Approximation capabilities of multilayer feedforward networks, *Neural Networks*, 1991, **4**, (2), pp 251–257. https://doi.org/10.1017/S0001924000025173

[32] Coello Coello, C.A. and Lechuga, M.S. MOPSO: a proposal for multiple objective particle swarm optimisation, Proceedings of 2002 Congress on Evolutionary Computation CEC 2002, 2002, vol. **2**, pp 1051–1056. https://doi.org/10.1016/0893-6080(91)90009-T

Cite this article: He T. and Wang L. (2023). Neural network-based velocity-controllable UAV flocking. *The Aeronautical Journal*, **127**, 289–304. https://doi.org/10.1017/aer.2022.61