Proceedings of the 39th Chinese Control Conference
July 27-29, 2020, Shenyang, China

Distributed anti-flocking method for area coverage of multiple unmanned aerial vehicles

Mengge Zhang[1], Jie Li[1], Xiangke Wang[1]
1. College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, P. R. China
E-mail: xkwang@nudt.edu.com

Abstract: The area coverage problem is an important issue because it can minimize uncertainty in a complex and unknown environment. This paper adopts a distributed anti-flocking algorithm inspired by the social behavior of solitary animals, which enables self-organized collaboration of multiple unmanned aerial vehicles (multi-UAVs) to achieve effective coverage of the mission area. The designed area coverage map represents the historical coverage information of each unmanned aerial vehicle (UAV). According to the rules of collision avoidance, decentering, and selfishness in the distributed anti-flocking algorithm, the UAVs are guided to move towards the direction that maximizing the coverage area and try to reduce the overlap of coverage region as well. Simulations show that the algorithm can achieve approximate full coverage of the task area and has good scalability, adaptability, and robustness.

Key Words: Anti-flocking, Area Coverage, Multiple Unmanned Aerial Vehicles

1 Introduction

Cooperative reconnaissance of multiple unmanned aerial vehicles (multi-UAVs) will be a useful means to obtain battlefield information in the future [1][2]. In complex and unknown dynamic environments, the primary task of multi-UAV cooperative reconnaissance is to cover the designated mission area in the shortest time and in the most economical and effective way. How to effectively control multi-UAVs to cooperatively perform area coverage tasks has become one of the research hotspots in the field of cooperative control in recent years, because of its important application value and practical significance in military and civil aspects [3–5].

The existing research methods of area coverage are mainly divided into two types: centralized and distributed. The centralized approach usually includes area decomposition and multi-UAV formation coverage. The method which is based on area decomposition divides the original area coverage problem into two sub-problems [6–8]. One is the division of the mission area. The other is the sub-area coverage planning. In [7], the coverage area was segmented according to the relative capability and initial position of UAVs. Each UAV applied the scanning method to cover its mission area. Reference [8] adopted an area-based polygon decomposition method and applied it to the task area allocation of multi-UAVs. While the multi-UAV formation coverage method is equal to extend the coverage of a single UAV. Commonly used methods include back-and-forth search [9] and internal spiral search [10]. Reference [11] improved the commonly used back-and-forth search method by adjusting the turning timing and turning position of UAVs. The advantage of the centralized approach is that it can achieve optimal coverage of the mission area. But when UAVs fail or environment changes, the centralized approaches are often limited.

While the distributed approach can deal well with unknown dynamic environments. Based on the distributed model predictive control theory, reference [12] proposed a collaborative area search algorithm, which can deal well with the problem of collaborative search of UAV swarms in unknown environments without prior information. Reference [13] designed a self-organized planning method based on reciprocal principle for the area coverage of UAV swarms. In this method, the coverage problem of UAVs was directly modeled and optimized in the velocity domain, and the optimal coverage velocity region was determined by the reciprocal behavior of neighboring UAVs.

Besides, some swarm intelligence methods that mimic biological behaviors are also gradually applied to regional coverage planning tasks. Reference [14] proposed a rule-based collaborative search algorithm based on the kinetic model-Boids. This algorithm applied centrifugal, disordered, inertial, random, and other rules, which enabled the UAV swarms to be efficiently distributed in the searching area. In [15], an anti-flocking control method that mimicked the social behavior of solitary animals was proposed for the coverage problem of mobile sensor networks. References [16–18] improved the anti-flocking method. By designing a distributed information map, the sensor nodes were guided to move towards the direction that maximizing their coverage area, which improved the coverage capability of the sensor networks. While, in most of the coverage methods [13–18], a simple second-order integrator model is adopted. When applying these methods to UAV swarms, the dynamic constraints of the UAV should be considered.

In this paper, the distributed anti-flocking algorithm is designed for the area coverage problem of multi-UAVs. The main work includes: (1) An area coverage map is designed for each UAV to store its historical coverage information independently. (2) The UAV kinematics are embedded in the design of three rules, i.e., collision avoidance, decentering, and selfishness in the distributed anti-flocking method. (3) A boundary processing mechanism is designed to avoid UAVs flying outside the mission area. MATLAB simulations show that the algorithm can achieve more than 99% coverage of the area.

2 Preliminaries

2.1 Problem Description

A fleet of N UAV system is considered, in which the i-th UAV is denoted by $U_i (i = 1, 2, \ldots, n)$. The mission area is an $L * W$ rectangular area, where L and W represent the length and width of the area respectively. Our goal is to achieve effective coverage of the mission area with multi-UAVs in the shortest time and in the most economical way.

It is assumed that all the UAVs are homogenous. The detection sensor is fixedly mounted on the UAV and modeled by a disk with a radius Rs. Suppose the center of the disk coincides with the geometric center of the UAV. UAVs communicate with each other through the wireless communication module, and the communication range is Rc, usually $Rc > 2Rs$. $p_i = \begin{bmatrix} x_i & y_i \end{bmatrix}$ is the position vector of U_i, and N_i represents the set of neighbors within the communication range of U_i, that is, $N_i = \{U_j : \|p_j - p_i\| < Rc, j = 1, 2, \ldots, n, j \neq i\}$.

2.2 UAV Model

To simplify the analysis, assuming that the UAV maintains a constant altitude over the mission area and flies at a constant speed. The two-dimensional unicycle model is adopted as the kinematics model of the UAV. Therefore, the discrete kinematics model of UAV is established as follows:

$$\begin{cases} x_i(t+1) = x_i(t) + vT\cos\varphi_i \\ y_i(t+1) = x_i(t) + vT\sin\varphi_i \\ \varphi_i(t+1) = \varphi_i(t) + Tw_i \end{cases} \quad (1)$$

where, (x_i, y_i) represents the position of U_i in the two-dimensional plane, v is the constant flight speed, T denotes the discrete time step, w_i is the control input of U_i, and it denotes the changing rate of the heading angle of the UAV, $w_{\max}$ is the upper bound of w_i, which is $|w_i| \leq w_{\max}$.

2.3 Area Coverage Map

The mission area is discretized into $M * N$ grids, and an area coverage map is created for each UAV to represent its historical coverage information on every grid.

Let I_i represent the area coverage map of U_i. Each grid in I_i is denoted as I_i^p, where $p = \begin{bmatrix} x & y \end{bmatrix}$ is the center coordinate of the grid. I_i^p stores the time when the grid was last covered by UAVs. Let P be a set of all such p.

First, initialize the area coverage map with $I_i^p = 0$. With the continuous exploration of UAVs, I_i is updated. At time t, if $\|p_i(t) - p\| < Rs$, there has:

$$I_i^p = t \quad (2)$$

When $N_i \neq \emptyset$, U_i can exchange area coverage maps with its neighbors. For $U_j (U_j \in N_i)$, when $I_i^p < I_j^p$, there has:

$$I_i^p = I_j^p \quad (3)$$

Through this updating mechanism, the UAV can track its historical coverage information and obtain the coverage information of its neighbors directly as well as its non-neighbors indirectly.

3 Area Coverage Strategy Based on Distributed Anti-flocking

3.1 Distributed Anti-flocking Algorithm

Inspired by solitary animals, a distributed anti-flocking algorithm was proposed in [15]. Accordingly, three rules are considered for the fleet of UAVs, which are:

1) Collision avoidance: stay away from obstacles and neighbors that within its safety distance.

2) Decentering: keep away from the center of its neighbors.

3) Selfishness: move to the direction which can maximize its gains.

The above three rules can be considered in the area coverage problem. The collision avoidance term can prevent the UAV from colliding with obstacles and neighboring UAVs, thus ensuring the safety of the UAV. The decentering term allows UAVs to be dispersed in the mission area as much as possible, which avoids the overlap of coverage region of each UAV and increases the instantaneous coverage area. While, the selfishness term can make UAV move along the direction that can maximize its gains, thereby maximizing its coverage area.

Let $\eta_i^d = l_i^d \begin{bmatrix} \cos\varphi_i^{\mathrm{d}} & \sin\varphi_i^{\mathrm{d}} \end{bmatrix}$ denote the expected velocity direction of U_i. l_i^d only represents the norm of η_i^d. According to the anti-flocking algorithm, η_i^d consists of three terms:

$$\eta_i^d = k_o\eta_i^o + k_c\eta_i^c + k_s\eta_i^s \quad (4)$$

where, η_i^o, η_i^c, and η_i^s are the collision avoidance term, the decentering term, and the selfishness term respectively. k_o, k_c, and k_s are all positive.

The normalized actual velocity direction of U_i is denoted as $\eta_i = \begin{bmatrix} \cos\varphi_i & \sin\varphi_i \end{bmatrix}$. θ_i represents the angle between the desired velocity direction η_i^d and the actual velocity direction η_i, which is:

$$\theta_i = \arccos\left(\frac{\eta_i^d \eta_i}{\|\eta_i^d\| \, \|\eta_i\|}\right) sign\left(\begin{bmatrix} \eta_i^d & 0 \end{bmatrix} \times \begin{bmatrix} \eta_i & 0 \end{bmatrix}\right) \quad (5)$$

Then we can obtain the control input w_i from θ_i, that is:

$$w_i = \begin{cases} \min\{w_{max}, k_w\theta_i\} & \theta_i \geq 0 \\ \max\{-w_{max}, k_w\theta_i\} & \theta_i < 0 \end{cases} \quad (6)$$

k_w is a positive constant. Therefore, the movement of the UAV can be controlled by obtaining its desired velocity direction η_i^d. In the following, we give the realization of the three terms in η_i^d.

First, the collision avoidance term η_i^o is given as:

$$\eta_i^o = \sum_{o=1}^{m} s(\|p_i - p_i^o\|, d_o)\frac{p_i - p_i^o}{\|p_i - p_i^o\|} \quad (7)$$

where, $p_i = \begin{bmatrix} x_i & y_i \end{bmatrix}$ is the position coordinate vector of U_i, $p_i^o = \begin{bmatrix} x_i^o & y_i^o \end{bmatrix}$ is the position of obstacles or neighboring UAVs, m is the total number of them, and d_o represents the safety distance of these UAVs.

$s(z, d)$ is a non-negative potential function. It reaches its maximum as $z \to 0$, smoothly decreases to 0 as $z \to d$, and remains at 0 when $z > d$. There are many functions can be used as $s(z, d)$, such as:

$$s(z, d) = \begin{cases} 1 + \cos\dfrac{\pi(z+d)}{2d} & z \in [0, d] \\ 0 & otherwise \end{cases} \quad (8)$$

Second, the desired velocity direction η_i^c obtained from the decentering term is:

$$\eta_i^c = s(\|p_i - p_i^c\|, d_c)\frac{p_i - p_i^c}{\|p_i - p_i^c\|} \quad (9)$$

while, $p_i^c = \begin{bmatrix} x_i^c & y_i^c \end{bmatrix}$ is the position vector of the centroid of the neighboring UAVs, d_c denotes the distance threshold that determines whether the decentering term works.

Third, let $p_i^s = \begin{bmatrix} x_i^s & y_i^s \end{bmatrix}$ represent the position of the center of the target grid that can maximize the gains. Then, the selfishness term is:

$$\eta_i^s = \frac{p_i^s - p_i}{\|p_i^s - p_i\|} \tag{10}$$

So the expected velocity direction η_i^d of U_i can be rewritten as:

$$\begin{aligned} \eta_i^d = k_o \sum_{o=1}^{m} s(\|p_i - p_i^o\|, d_o) \frac{p_i - p_i^o}{\|p_i - p_i^o\|} \\ + k_c s(\|p_i - p_i^c\|, d_c) \frac{p_i - p_i^c}{\|p_i - p_i^c\|} + k_s \frac{p_i^s - p_i}{\|p_i^s - p_i\|} \end{aligned} \tag{11}$$

3.2 Target Grid Selection

To increase the coverage area of UAVs and reduce areas that have not been covered for a long time, a benefit function is designed to evaluate each grid. And the UAV is guided to move to the grid with maximum benefits.

We apply the benefit function designed in [18] to evaluate the area coverage map I_i, which is:

$$\lambda_i^p(t) = (t - I_i^p) \left[\rho + (1-\rho)\exp(-\alpha \|p_i - p\| - \beta \|p_i^s - p\|)\right] \tag{12}$$

Where, $(t - I_i^p)$ is the time span after the location p has been last visited. ρ, α, and β are all positive constants.

Then, the target grid is given as:

$$p_i^s(t+1) = \arg\max_{p \in \tilde{P}_i} \lambda_i^p(t) \tag{13}$$

and $\tilde{P}_i = \{p : p \in P, \|p_j - p\| \geq \|p_i - p\| > Rs, j \in N_i\}$.

We improve the target grid selection method by adding additional selection mechanism to avoid the oscillation of UAVs when there is more than one grid satisfying formula (13). Suppose there are $F(F > 1)$ that grids, using $\mathrm{p}_i^{s_k}(k = 1, 2, \ldots, F)$ to represent the position of them. And the corresponding expected velocity direction $\eta_i^{s_k}$ can be calculated by formula (10). Then, through formula (5), $\theta_i^{s_k}$, which is the angle between the desired velocity direction $\eta_i^{s_k}$ and the actual velocity direction η_i, can be obtained. The target grid p_i^s is the one with the smallest $|\theta_i^{s_k}|$, which denoted as $p_i^{s_q}$, that is:

$$\begin{cases} p_i^s = p_i^{s_q} \\ q = \arg\min\limits_{k=1,2,\ldots,F} |\theta_i^{s_k}| \end{cases} \tag{14}$$

When there are two grids both satisfying the minimum value, the grid with the desired velocity direction to the right of the actual velocity direction is preferred. This means that when the solution of formula (14) is not unique, choose the one satisfying $\theta_i^{s_q} > 0$.

3.3 Area Boundary Processing

To avoid UAVs flying out of the mission area, the boundary pre-control layer is designed by adding polygonal geofence. As shown in Fig. 1, for a rectangular mission area, the designed geo-fence is the dashed line in the figure, the inner area of the geo-fence is represented by Ω, and the shaded part is the boundary pre-control layer.

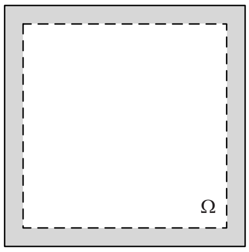


Fig. 1: Mission area and the geo-fence

When located in the boundary pre-control layer, the UAV needs to move away from the boundary as well as move toward the center of the task area, thereby preventing it from leaving the mission area. Let $p_i^b = \begin{bmatrix} x_i^b & y_i^b \end{bmatrix}$ denote the position of the nearest boundary point from U_i and $p^m = \begin{bmatrix} x^m & y^m \end{bmatrix}$ denote the center of the mission area. Then, the desired velocity direction obtained from the boundary pre-control layer is:

$$\eta_i^b = \begin{cases} q_m \dfrac{p^m - p_i}{\|p^m - p_i\|} + q_b \dfrac{p_i - p_i^b}{\|p_i - p_i^b\|} & p_i \notin \Omega \\ 0 & p_i \in \Omega \end{cases} \tag{15}$$

Here, η_i^b is given by the repulsion of the boundary and the attraction of the regional center. q_m and q_b are the weight of the corresponding items.

Therefore, after introducing the boundary processing mechanism, the expected velocity direction of U_i is:

$$\eta_i^d = k_o \eta_i^o + k_c \eta_i^c + k_s \eta_i^s + k_b \eta_i^b \tag{16}$$

Then we can calculate the control input according to formula (5) and (6).

4 Simulation Analysis

In this section, the distributed anti-flocking algorithm proposed in this paper is verified by MATLAB simulations. The covered area is a square area of $2km * 2km$, and the whole region is discretized into grids of $25m * 25m$. UAV flies at a constant speed of $v = 18m/s$. The time step is $T = 0.2s$, the communication range is $Rc = 400m$, and the perception radius is $Rs = 150m$. Other parameters in the anti-flocking algorithm are set as follows: $k_o = 0.8$, $k_c = 0.4$, $k_s = 0.5$, $k_b = 1$, $\rho = 0.2$, $\alpha = 0.04$, $\beta = 0.01$, $q_m = 0.6$, and $q_b = 0.8$.

4.1 Performance Analysis

The evaluation indexes of this algorithm are cumulative area coverage, instantaneous area coverage, and task time consuming, which are defined as follows:

1) Cumulative area coverage: the fraction of the area covered by at least one UAV at least one time versus the whole area of the task region within the time interval $[0, t)$, denoted as $AC(t)$.

2) Instantaneous area coverage: the fraction of the area covered by UAVs versus the whole area of the task region at the time instant t.

3) Task time consuming: the time taken to complete the area coverage task.

First, the cumulative area coverage is analyzed when the number of UAVs varies. Fig. 2 shows the average results of the cumulative area coverage for 10 simulations. With time passing by, the cumulative area coverage continues to increase and reaches greater than 99%. The more UAVs, the faster the cumulative area coverage increases. However, with the increase in the number of UAVs, the growth rate of gains slows down while the cost increases. Therefore, the number of UAVs should be carefully determined to obtain good returns.

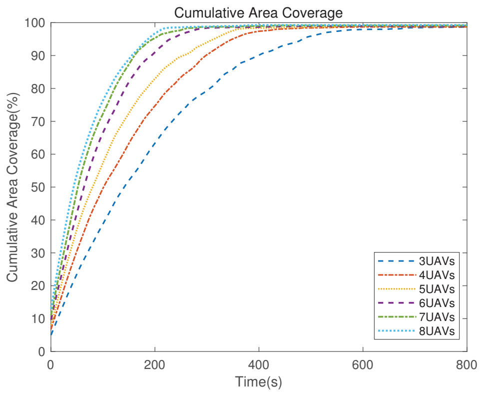


Fig. 2: Average cumulative area coverage for different scales of the UAV swarm

Second, the relationship between the instantaneous area coverage and the number of UAVs is analyzed. As shown in Fig. 3, the instantaneous coverage increases linearly with the number of UAVs, which indicates the effect of the decentering term.

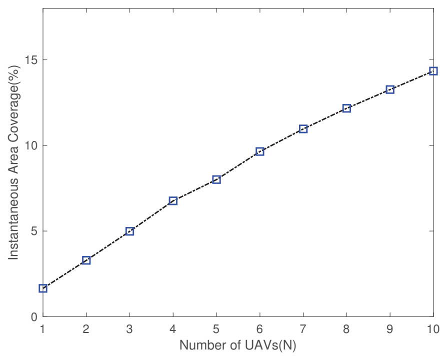


Fig. 3: The relationship between the instantaneous area coverage and the number of UAVs

Third, we analyze the task time consuming with different numbers of UAVs. For different scales of the UAV swarm, we take the average results of 10 simulations. The simulation results are shown in Fig. 4. The more UAVs, the shorter the time required to complete the task. But when the number of UAVs increases to a certain value, the time required to complete the task is not significantly reduced. Increased time spent in collision avoidance and the repeat of the coverage area may be the reasons. Therefore, there is no need of using too many UAVs.

Then, the performance of cumulative area coverage under the control of distributed anti-flocking algorithm and random search algorithm is compared. Here, we use five UAVs to cover the area. For each algorithm, we take the average results of 10 simulations. The initial conditions of the two

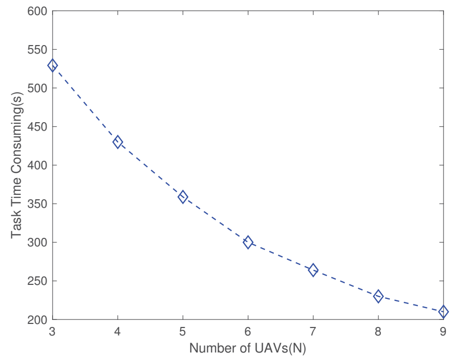


Fig. 4: Average task time consuming

algorithms are set to be the same. Simulation results are shown in Fig. 5. It indicates that the cumulative area coverage of the distributed anti-flocking algorithm proposed in this paper increases faster than random search algorithm.

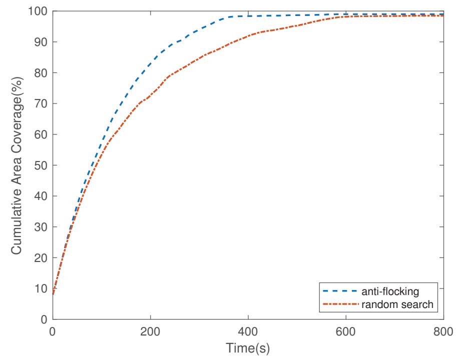


Fig. 5: Average cumulative area coverage for the two different algorithms

4.2 Boundary Processing

To verify the effectiveness of the boundary processing mechanism designed in Section 3.3, the algorithms with and without boundary processing mechanism are compared. The number of UAVs is fixed to 5, and 50 simulations are taken in each case. The initial position and heading of each UAV are given randomly. The statistical results of the average time of UAVs flying out of the area boundary are shown in Table 1. The introduction of the boundary processing mechanism can well prevent UAVs from flying out of the boundary.

Table 1: Average Time of UAVs Flying Out of the Mission Area

Coverage Algorithms	Time (s)
without boundary processing mechanism	77.55
with boundary processing mechanism	12.16

Fig. 6 shows the trajectory of the UAV swarm after the boundary processing mechanism is introduced. The solid circles in the figure represent UAVs, and the dotted circles represent the coverage range of UAVs. The UAVs can complete the area coverage task well with little time out of the mission area.

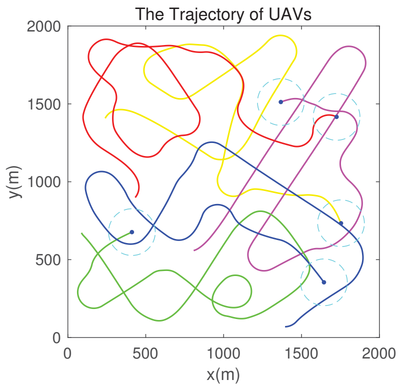


Fig. 6: The trajectory of UAVs after introducing the boundary processing mechanism

4.3 Obstacle Avoidance

In this section, we take several simulations to verify the obstacle avoidance performance of our algorithm. Assuming that there are three static obstacles in the mission area. The two circular obstacles with a radius of $90m$ and $110m$ are located at $(590m, 790m)$ and $(1410m, 610m)$ respectively. And a square obstacle with a side length of $200m$ is located at $(1600m, 1300m)$. As shown in Fig. 7, we use five UAVs to cover the area. The initial position and heading of each UAV are given randomly.

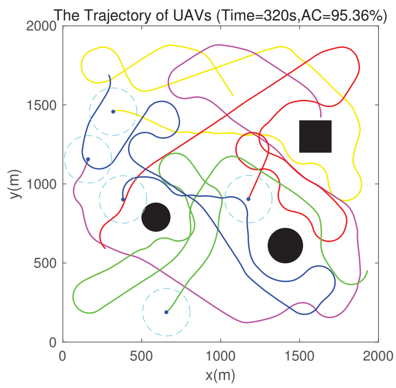


Fig. 7: The trajectory of UAVs when there are obstacles

With the continuous exploration of UAVs, no collision between the UAV and the obstacle is detected. And the cumulative area coverage reaches 95.36% within $320s$. This indicates the effectiveness of the obstacle avoidance performance of our anti-flocking algorithm.

4.4 UAV Failures

One advantage of the distributed algorithm is that it has good adaptability. This section verifies the area coverage performance of our algorithm when some UAVs fail.

As shown in Fig. 8 and 9, five UAVs are used to cover the area. It is assumed that U_1 fails at $60s$ and U_3 fails at $140s$. The simulation results tell us that when some UAVs in the swarm fail, the rest can still complete the coverage task, which showing good adaptability.

5 Conclusion

Based on the distributed anti-flocking algorithm that imitates the social behavior of solitary animals, this paper realizes effective coverage of the mission area with multi-UAVs. This algorithm is a completely distributed intelligent control algorithm with good adaptability, extensibility, and robustness. UAVs make online decisions through local information exchange. The increase or decrease in the number of UAVs does not affect the completion of the area coverage mission, and the algorithm can be applied to the coverage mission of large-scale UAV swarms. MATLAB simulations verify the performance of the algorithm. In our future work, the algorithm proposed in this paper is considered to extend to more complex coverage scenarios, such as areas with no-fly zones and areas with complex terrain.

References

[1] DOD, Unmanned Systems Integrated Roadmap. Department of Defense, 168(2013).

[2] Chen J, Zha W, Zhang J, et al. Cooperative Area Reconnaissance for Multi-UAV in Dynamic Environment[C], *Control Conference. IEEE*,2013.

[3] Gustavo A, Guilherme P, Luciano P, et al. Multi-UAV Routing for Area Coverage and Remote Sensing with Minimum Time[J]. *Sensors*, 2015, 15(11):27783-27803.

[4] Darrah M, Wilhelm J, Munasinghe T, et al. A Flexible Genetic Algorithm System for Multi-UAV Surveillance: Algorithm and Flight Testing[J]. *Unmanned Systems*, 2015, 03(01):49-62.

[5] Papatheodorou S, Tzes A, Stergiopoulos Y . Collaborative visual area coverage[J]. *Robotics and Autonomous Systems*, 2017, 92:126-138.

[6] Balampanis F, Maza I, Ollero A. Area decomposition, partition and coverage with multiple remotely piloted aircraft systems operating in coastal regions[C]// *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2016.

[7] Maza I, Ollero A. Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms[M]// Distributed Autonomous Robotic Systems 6. 2007.

[8] Hert S, Lumelsky V. Polygon area decomposition for multiple-robot workspace division[J].*International Journal of Computational Geometry & Applications*, 1998, 8(4):437-466.

[9] Andersen H L. Path Planning for Search and Rescue Mission using, Multicopters[J]. Institutt for Teknisk Kybernetikk, 2014.

[10] Gabriely Y, Rimon E. Spiral-STC: an on-line coverage algorithm of grid environments by a mobile robot[C]// IEEE International Conference on Robotics and Automation(ICRA). IEEE, 2002:954-960.

[11] Qing-Po Wu, Shao-Lei Z, Gao-Yang Y, et al. Improvement of Multi-UAV Cooperative Coverage Searching Method[J]. *Electronics Optics & Control*, 2016.

[12] Trodden P, Richards A G. Multi-Vehicle Cooperative Search using Distributed Model Predictive Control[J]. aiaa journal, 2013.

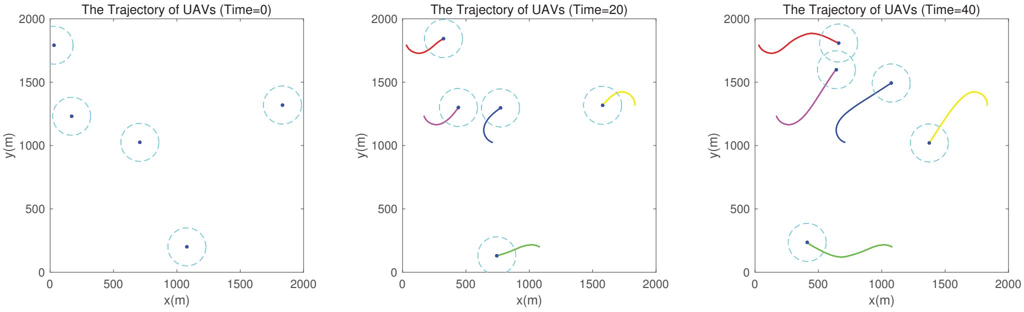
[13] Chen Wei, Hu Yilong, Distributed planning method based on reciprocal principle for self-organized area coverage of UAV[J], *Journal of Computer Applications*,2019,39(z1):46-49.

[14] Zhang Jiangdong, The Regional Collaboration Search Algorithm of UAV Swarm Based on Rules[J],*Computer Science and Application*, 2019, 9(11), 2028-2036.

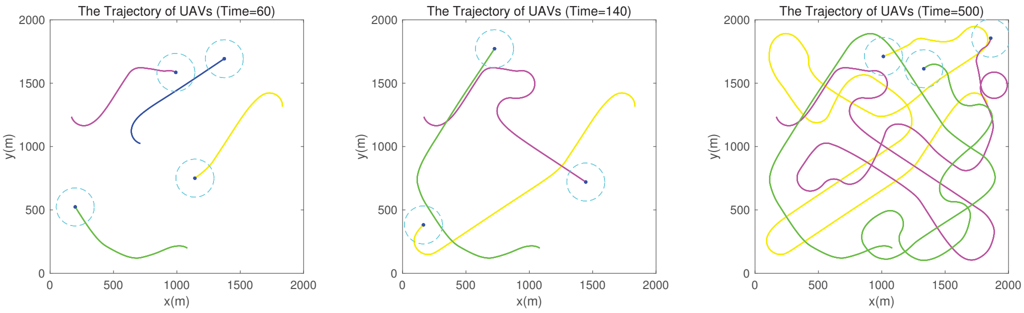
[15] Miao, Yun-Qian, Khamis, Alaa, Kamel, and Mohamed S., Applying anti-flocking model in mobile surveillance systems, 1-6, 2010

[16] Ganganath N, Cheng C T, Tse C K. Distributed Anti-Flocking Control for Mobile Surveillance Systems[C]// *Circuits and Systems (ISCAS), 2015 International Symposium on. IEEE*, 2015.

[17] Ganganath N, Cheng C T, Tse C K. Distributed Anti-Flocking

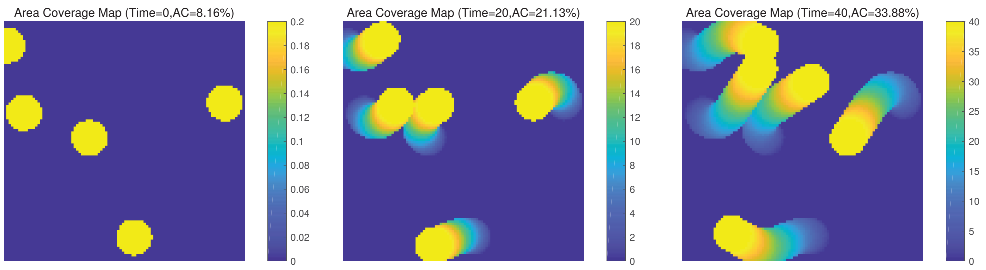


(a) Time = $0s$ (b) Time = $20s$ (c) Time = $40s$

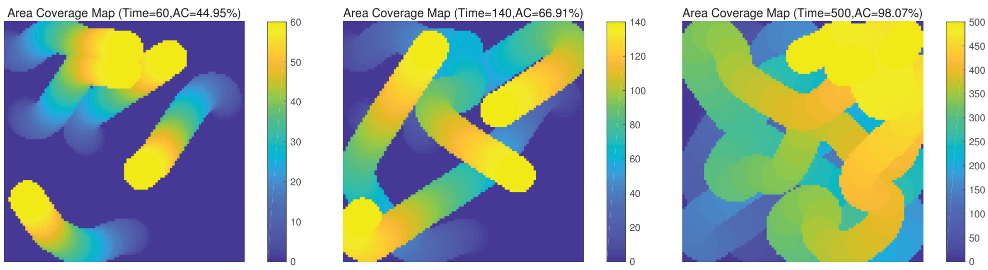


(d) Time = $60s$ (e) Time = $140s$ (f) Time = $500s$

Fig. 8: The trajectory of UAVs, while UAV1 fails at time=$60s$ and UAV3 fails at time=$140s$.



(a) Time = $0s$ (b) Time = $20s$ (c) Time = $40s$



(d) Time = $60s$ (e) Time = $140s$ (f) Time = $500s$

Fig. 9: The area coverage maps of UAVs, while UAV1 fails at time=$60s$ and UAV3 fails at time=$140s$.

Control for Mobile Surveillance Systems[C]// *Circuits and Systems (ISCAS), 2015 International Symposium on. IEEE*, 2015.

[18] Ganganath N, Cheng C T, Tse C. Distributed Anti-Flocking Algorithms for Dynamic Coverage of Mobile Sensor Networks[J]. *IEEE Transactions on Industrial Informatics*, 2016:1-1.