www.ietdl.org

Published in IET Control Theory and Applications
Received on 2nd October 2012
Revised on 13th March 2013
Accepted on 17th March 2013
doi: 10.1049/iet-cta.2012.0779

ISSN 1751-8644

Brief Paper

Distributed cooperative control for deployment and task allocation of unmanned aerial vehicle networks

Jinwen Hu[1,2], *Zhao Xu*[3]

[1] *Singapore Institute of Manufacturing Technology, 638075, Singapore*
[2] *School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore*
[3] *Institute of High Performance Computing, 138632, Singapore*
E-mail: jwhu@SIMTech.a-star.edu.sg

Abstract: In this study, the authors consider the deployment of unmanned aerial vehicle networks for task accomplishment within a closed region. Each agent with limited sensing range and communication range needs to take charge of the task accomplishment within a part of the whole region. The main objective is to optimise the deployment of the agents such that the maximum travelling time the agents take to reach a place within the surveillance region is minimised. The deployment issue is formulated as the worst-case disc-covering problem and a distributed cooperative control strategy is designed for agents with limited mobility. It is proven that by the proposed control strategy the network configuration converges to a local optimum configuration. Moreover, a combined optimisation approach is developed to improve the performance by optimising the initial configuration. To guarantee the proper working of the designed control strategy and K-connectivity of the network, a distributed topology control scheme is proposed. Finally, the effectiveness of the proposed control strategy is testified by simulation.

1 Introduction

Unmanned aerial vehicles (UAVs) are increasingly employed in the civil and military applications such as battlefield surveillance, remote sensing and map building, etc [1–4]. Two major categories of control problems for UAV networks are coverage control and topology control. Coverage control mainly aims at the control of coverage area in terms of some sensing criteria, whereas topology control mainly aims at the control of communication topology. In many applications, the two problems are combined. For example, in the collaborative coverage over a closed surveillance region, agents may have to know the information of other agents and adaptively tune their communication topology and geometric formation to achieve the optimal coverage, which requires that the network be connected. Instead of being treated as two separate problems, the topology control and coverage control are usually considered as a combined cooperative control problem and have been studied in many applications.

These topology control and coverage control methods can be roughly classified into two types: centralised and distributed. Centralised algorithms can achieve the optimal performance in most cases [5]. However, they suffer from unnegligible potential unsafeties, for example, the whole network would be down or work improperly if any failure or fault happens to the centralised control agent. Compared with the centralised control algorithms, distributed control algorithms are more robust to accidental failures of agents and breaks of communication links [6]. There have been a lot of recent works addressing the distributed cooperative control. Gradient following by autonomous vehicle systems inspired by bacterial chemotaxis has been explored in [7, 8]. In [6], a swarming control method is proposed, in which individuals balance their own gradient descent with inter-agent attraction and repulsion forces. However, this method requires each agent to know the gradient at its location and know the relative position of each of the other agents. In [9], a virtual leader approach to gradient climbing is taken, where the virtual leader finally can reach the local maximum point instead of the true agent. A coordinated control strategy is developed for a group of autonomous vehicles to descend or climb an environmental gradient using measurements of the environment together with relative position measurements of the nearest neighbours [10]. The gradient at current location of each agent is still assumed to be known in this strategy. In [11], a stable control strategy is proposed for groups of agents to move and reconfigure cooperatively in response to a sensed environment. The underlying coordination framework uses the virtual bodies and artificial potentials mentioned in [9]. This strategy aims to seek out local maxima or minima in the environment filed. However, each agent is required to implement a centralised processing of the measurements from the whole network. In [12], a distributed and fault-tolerant control algorithm is designed for the cooperation and redeployment of mobile sensor networks such that the covered area can be enlarged, which combines the virtual potential method and the Delaunay triangulation. In [13], a theoretical framework is put forward for the design and analysis of distributed flocking algorithms for multi-agent networks. A group of agents can finally achieve an expected formation and reach a given moving rendezvous point by the proposed flocking algorithms. The optimal sensor placement and motion coordination strategies are studied in [14] for target tracking with range sensors in mobile sensor networks (MSNs). In [15], the problem of environmental modelling is addressed using a proportional–integral average consensus estimator to fuse the local data of each individual agent to estimate the environment model. A control law is proposed for mobile agents to move to maximise their sensory information relative to current uncertainties in the model. In [16], a distributed cooperative control method is proposed to let robots reach peaks of an unknown scalar field, which is similar to the energy intensity filed of targets. However, such method relies on the parameterisation of the unknown field and the basis functions for the parameterisation are assumed to be known, which makes it unsuitable for time-variant energy intensity fields of moving targets. A distributed Kriged Kalman filter is developed in [17] to estimate a spatiotemporal field. Gradient control laws are developed to move the mobile agents to critical points of the sensory field. Some other related works are: cooperative task allocation [18–21] and cooperative formation control [22–25].

In this paper, we consider the distributed cooperative control for deployment of UAV networks over a given surveillance region. In our problem, each agent with limited communication range is responsible for answering the demandings within a part of the whole region. Our main objective is to find the optimal deployment of the agents such that the maximum travelling time the agents take to reach a place within the surveillance region is minimised. This is formulated as a worst-case disc-covering problem similar to the one in [26]. However, limited communication and moving capabilities of agents are not considered in [26], which have a great impact on the optimisation performance. The main contribution of this paper is that we design a distributed control strategy for agents with limited mobility and prove that by our control strategy the network converges to stationary configuration. Moreover, a combined optimisation approach is developed by optimising the initial configuration before minimising the travelling time, through which the performance is shown to be improved by simulation. Besides, a distributed topology control scheme is designed for K-connectivity maintenance, which is also the key to guarantee the convergence of network configuration by providing the correct Voronoi partition.

The rest of the paper is organised as follows. In Section 2, we give the basic definitions and assumptions. The cooperative control strategy for optimisation of travelling time is considered in Section 3. A distributed topology control scheme for K-connectivity maintenance is proposed in Section 4. The proposed distributed cooperative control strategy is validated by simulation in Section 5. Section 6 is the conclusion.

2 Basic definitions and assumptions

We assume that all agents move in the same plane parallel to the ground plane and have a limited sensing range R_S as shown in Fig. 1. In this case, we can project the planar coordinates of all agents onto the ground plane, where $\mu_i \in \mathbb{R}^2$ is the projection of agent i, and only need to design a two-dimensional control law u_i for each agent. In the discrete-time case, we can evaluate μ_i at a discrete-time step k, which gives $\mu_{i,k}$, and we keep the same meaning of notations for other variables. In this section, we consider the following first-order dynamic model for each agent such as a quadrotor helicopter

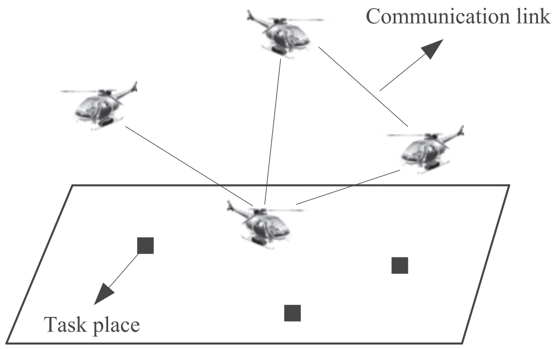

Fig. 1 *Deployment of UAV networks for task allocation*

$$\mu_{i,k+1} = \mu_{i,k} + u_{i,k} \tag{1}$$

Note that not all UAVs can be modelled accurately by (1) and we use it mainly for the ease of discussions.

The topology of the network of all agents at time k is modelled by an undirected graph $\mathcal{G}_k = (\mathcal{E}_k, \mathcal{V})$. $\mathcal{V} = \{1, 2, \ldots, N\}$ is the vertex set and $\mathcal{E}_k = \{\{i,j\} : i,j \in \mathcal{V}; \|\mu_{i,k} - \mu_{j,k}\| \leqslant R_C\}$ is the edge set, where each edge $\{i,j\}$ is an unordered pair of distinct agents and R_C is the communication range of each agent. The graph or the network is connected if for any two vertices i and j there exists a sequence of edges (a path) $\{i, h_1\}, \{h_1, h_2\}, \ldots, \{h_{n-1}, h_n\}, \{h_n, j\}$ in $\mathcal{E}_k$. The network is K-connected ($K < N$) if the removal of any $K-1$ agents does not partition the network with the left agents. Let $\mathcal{N}_{i,k} = \{j \in \mathcal{V} | \{i,j\} \in \mathcal{E}_k\} \cup \{i\}$ denote the set of neighbours of agent i at time k where an agent is assumed to be a neighbour of itself. The degree (number of neighbours) of agent i at time k is denoted as $d_{i,k} = |\mathcal{N}_{i,k}|$.

3 Optimisation of travelling time

3.1 Cooperative control strategy

In this part, we first consider the problem of minimising the largest time length for travelling to an arbitrary task place q in a closed region $\mathbb{O} \in \mathbb{R}^2$ in the ground plane by at least one agent. Assuming that all agents move at the same constant speed to finish a task, we should minimise the largest distance over all agents and all task places. This is formulated as the worst-case disc-covering optimisation (i.e. use a fixed number of discs to cover a close region but minimise the radius of the largest disc) [26]

$$\text{minimise} \quad \mathcal{H}_{\text{dc}}(\mu_1, \ldots, \mu_N) = \max_{q \in \mathbb{O}} \min_{i \in \mathcal{V}} \|q - \mu_i\| \tag{2}$$

where N is the total number of agents and $\|\bullet\|$ denotes the Euclidean norm. Denoting the Voronoi partition of $\mathbb{O}$ by $\{\mathbb{V}_1, \ldots, \mathbb{V}_N\}$, where $\mathbb{V}_i = \{q : \|q - \mu_i\| \leqslant$

$\|q-\mu_j\| \forall j \in \mathcal{V}\}$ is the Voronoi cell generated by the point μ_i, we have $\min_{i\in\mathcal{V}} \|q-\mu_i\| = \|q-\mu_j\| \ \forall q \in \mathbb{V}_j$ and $\mathbb{O} = \bigcup_{j\in\mathcal{V}} \mathbb{V}_j$. Then, we can obtain

$$\begin{aligned}\max_{q\in\mathbb{O}} \min_{i\in\mathcal{V}} \|q-\mu_i\| &= \max_{j\in\mathcal{V}} \max_{q\in\mathbb{V}_j} \min_{i\in\mathcal{V}} \|q-\mu_i\| \\ &= \max_{j\in\mathcal{V}} \max_{q\in\mathbb{V}_j} \|q-\mu_j\| \\ &= \max_{j\in\mathcal{V}} \max_{q\in\partial\mathbb{V}_j} \|q-\mu_j\| \end{aligned} \tag{3}$$

Therefore (2) is equivalent to

$$\text{minimise} \quad \mathcal{H}_{\text{dc}}(\mu_1, \ldots, \mu_N) = \max_{i\in\mathcal{V}} \max_{q\in\partial\mathbb{V}_i} \|q-\mu_i\| \tag{4}$$

Proposition 1 [26]: Let

$$\begin{aligned}\mathcal{H}_{\text{dc}}(\mu_1, \ldots, \mu_N, \mathbb{W}_1, \ldots, \mathbb{W}_N) &= \max_{i\in\mathcal{V}} \max_{q\in\mathbb{W}_i} \|q-\mu_i\| \\ &= \max_{i\in\mathcal{V}} \max_{q\in\partial\mathbb{W}_i} \|q-\mu_i\| \end{aligned} \tag{5}$$

For any configuration $\{\mu_1, \ldots, \mu_N\}$ and any partition $\{\mathbb{W}_1, \ldots, \mathbb{W}_N\}$, we have

$$\begin{aligned}&\mathcal{H}_{\text{dc}}(\mu_1, \ldots, \mu_N, \mathbb{V}_1, \ldots, \mathbb{V}_N) \\ &\quad \leqslant \mathcal{H}_{\text{dc}}(\mu_1, \ldots, \mu_N, \mathbb{W}_1, \ldots, \mathbb{W}_N)\end{aligned} \tag{6}$$

and

$$\begin{aligned}&\mathcal{H}_{\text{dc}}(\text{CC}(\mathbb{W}_1), \ldots, \text{CC}(\mathbb{W}_N), \mathbb{W}_1, \ldots, \mathbb{W}_N) \\ &\quad \leqslant \mathcal{H}_{\text{dc}}(\mu_1, \ldots, \mu_N, \mathbb{W}_1, \ldots, \mathbb{W}_N)\end{aligned} \tag{7}$$

where $\text{CC}(\mathbb{W}_i)$ is the centre of the disc that covers $\mathbb{W}_i$ with the shortest radius, that is, the circumcentres of $\mathbb{W}_i$. Therefore the circumcentres of the Voronoi cells $\{\text{CC}(\mathbb{W}_1), \ldots, \text{CC}(\mathbb{W}_N)\}$ is the local optimum configuration of the agents.

Based on the above conclusion, we can design the control law as

$$u_{i,k} = G_{i,k}(\text{CC}(\mathbb{V}_{i,k}) - \mu_{i,k}) \tag{8}$$

where $0 < G_{i,k} \leqslant 1$ is a gain parameter. That is, let each agent move towards the circumcentre of its Voronoi cell at each time. In [26], only the case $G_{i,k} = 1$ is considered and no proof of the convergence of network configuration is given. However, owing to the system limitations, the condition $\|u_{i,k}\| \geqslant (\text{CC}(\mathbb{V}_{i,k}) - \mu_{i,k})$ may not be satisfied. Hence, we need to show that for any given positive gain $G_{i,k} \leqslant 1$, all agents still can converge to stationary points.

Remark 1: $G_{i,k}$ can be a time-varying parameter updated at each time. For example, if each agent has limited mobility, that is, $\|u_{i,k}\| \leqslant U$, then $G_{i,k}$ can be determined by

$$G_{i,k} = \begin{cases} \dfrac{U}{\|\text{CC}(\mathbb{V}_{i,k}) - \mu_{i,k}\|}, & \text{if} \quad \|\text{CC}(\mathbb{V}_{i,k}) - \mu_{i,k}\| > U \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

Lemma 1: Under the control law (8) with $0 < G_{i,k} \leqslant 1$, $\mathcal{H}_{\text{dc}}(\mu_1, \ldots, \mu_N)$ is non-increasing.

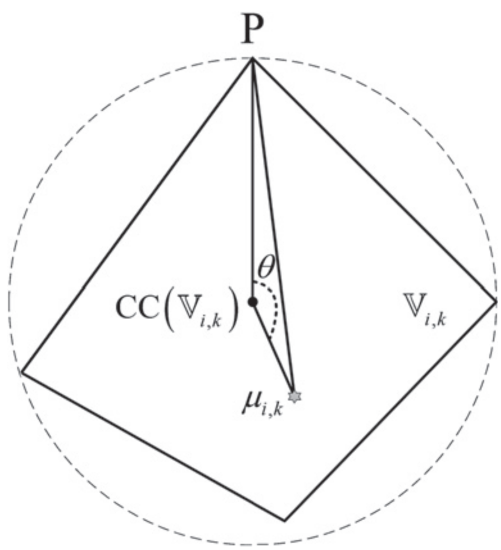

Fig. 2 *Voronoi region and its circumcircle of agent i*

Proof: For an arbitrary agent i, if $\mu_{i,k} \neq \text{CC}(\mathbb{V}_{i,k})$ that at time k, then we can always find a point P on $\partial\mathbb{V}_{i,k}$, such that

$$\|P-\mu_{i,k}\| = \max_{q\in\partial\mathbb{V}_{i,k}} \|q-\mu_{i,k}\| > \|P - \text{CC}(\mathbb{V}_{i,k})\| \tag{10}$$

and the angle θ between the vectors $P - \text{CC}(\mathbb{V}_{i,k})$ and $\mu_{i,k} - \text{CC}(\mathbb{V}_{i,k})$ is no less than $(\pi/2)$. This is done by acknowledging that $\text{CC}(\mathbb{V}_{i,k})$ is the centre of the disc with the smallest radius that can fully cover $\mathbb{V}_{i,k}$ and $\text{CC}(\mathbb{V}_{i,k}) \subset \mathbb{V}_{i,k}$. An example of such geometry is shown in Fig. 2, from which it can be seen that with $0 < G_{i,k} \leqslant 1$ in the control law (8), $\|P-\mu_{i,k+1}\| < \|P-\mu_{i,k}\|$, that is

$$\max_{q\in\partial\mathbb{V}_{i,k}} \|q-\mu_{i,k+1}\| < \max_{q\in\partial\mathbb{V}_{i,k}} \|q-\mu_{i,k}\| \tag{11}$$

Recalling the definition (5), for all agents we have

$$\begin{aligned}&\mathcal{H}_{\text{dc}}(\mu_{1,k+1}, \ldots, \mu_{N,k+1}, \mathbb{V}_{1,k}, \ldots, \mathbb{V}_{N,k}) \\ &\quad \leqslant \mathcal{H}_{\text{dc}}(\mu_{1,k}, \ldots, \mu_{N,k}, \mathbb{V}_{1,k}, \ldots, \mathbb{V}_{N,k})\end{aligned} \tag{12}$$

At time $k+1$, generate the new Voronoi cell $\mathbb{V}_{j,k+1}$ and calculate $\text{CC}(\mathbb{V}_{j,k})$. Further, according to (6), we can obtain

$$\begin{aligned}&\mathcal{H}_{\text{dc}}(\mu_{1,k+1}, \ldots, \mu_{N,k+1}) \\ &\quad = \mathcal{H}_{\text{dc}}(\mu_{1,k+1}, \ldots, \mu_{N,k+1}, \mathbb{V}_{1,k+1}, \ldots, \mathbb{V}_{N,k+1}) \\ &\quad \leqslant \mathcal{H}_{\text{dc}}(\mu_{1,k+1}, \ldots, \mu_{N,k+1}, \mathbb{V}_{1,k}, \ldots, \mathbb{V}_{N,k}) \\ &\quad \leqslant \mathcal{H}_{\text{dc}}(\mu_{1,k}, \ldots, \mu_{N,k}, \mathbb{V}_{1,k}, \ldots, \mathbb{V}_{N,k}) \\ &\quad = \mathcal{H}_{\text{dc}}(\mu_{1,k}, \ldots, \mu_{N,k})\end{aligned} \tag{13}$$

□

To guarantee the convergence of network configuration, the above conditions are not enough, since agents may continue moving after $\mathcal{H}_{\text{dc}}$ stops decreasing. Therefore we need to add on some extra conditions such that all agents stop moving when they detect that $\mathcal{H}_{\text{dc}}$ has stopped decreasing. To obtain the information of $\mathcal{H}_{\text{dc}}$ regarding the whole network, we use the following distributed maximum consensus

doi: 10.1049/iet-cta.2012.0779

protocol for $k > 0$

$$H_{i,k} = \begin{cases} \max_{q \in \partial \mathbb{V}_{i,k-1}} \|q - \mu_{i,k}\|, & \text{if } \mathrm{mod}(k, T) = 0 \\ \max_{j \in \mathcal{N}_{i,k}} H_{j,k-1}, & \text{otherwise} \end{cases} \tag{14}$$

where T is a predefined positive integer and $\mathrm{mod}(k, T)$ is the modulo function that gives the remainder of division of k by T. Specifically, we let $H_{i,0} = \max_{q \in \partial \mathbb{M}_i} \|q - \mu_{i,k}\|$, where $\{\mathbb{M}_1, \ldots, \mathbb{M}_N\}$ is any given initial partition subject to $\mu_{i,k} \in \mathbb{M}_i$ for all i. Thus, at each time k, each agent i communicates not only its current position $\mu_{i,k}$ but also $H_{i,k}$ to its neighbours. Now, we modify the control law (8) as follows (see (15))

where $t_k = k - \mathrm{mod}(k + 1, T)$ and $\delta > 0$ is a predefined threshold.

Theorem 1: Under the control law (15), if the network is connected all the time and $T \geqslant N$, $\{\mu_{1,k}, \ldots, \mu_{N,k}\}$ will converge to a stationary configuration.

Proof: First, if the network is connected all the time and $T \geqslant N$, we can obtain

$$\begin{aligned} H_{i,nT-1} &= \max_{i \in \mathcal{V}} \max_{q \in \partial \mathbb{V}_{i,(n-1)T-1}} \|q - \mu_{i,(n-1)T}\| \\ &= \mathcal{H}_{\mathrm{dc}}(\mu_{1,(n-1)T}, \ldots, \mu_{N,(n-1)T}) \end{aligned} \tag{16}$$

where $n \geqslant 1$ is an integer. This implies that for $k \geqslant T - 1$

$$H_{i,t_k} = \mathcal{H}_{\mathrm{dc}}(\mu_{1,t_k-T+1}, \ldots, \mu_{N,t_k-T+1}) \quad \forall i \in \mathcal{V} \tag{17}$$

From Lemma 1, we know that

$$\begin{aligned} \lim_{k \to +\infty} & \mathcal{H}_{\mathrm{dc}}(\mu_{1,t_k-2T+1}, \ldots, \mu_{N,t_k-2T+1}) \\ & - \mathcal{H}_{\mathrm{dc}}(\mu_{1,t_k-T+1}, \ldots, \mu_{N,t_k-T+1}) = 0 \end{aligned} \tag{18}$$

Hence, there exists a time k^* such that for $k \geqslant k^*$

$$\begin{aligned} H_{i,t_k-T} - H_{i,t_k} = & \mathcal{H}_{\mathrm{dc}}(\mu_{1,t_k-2T+1}, \ldots, \mu_{N,t_k-2T+1}) \\ & - \mathcal{H}_{\mathrm{dc}}(\mu_{1,t_k-T+1}, \ldots, \mu_{N,t_k-T+1}) > \delta \end{aligned} \tag{19}$$

which implies that $u_{i,k} = 0$ $\forall i \in \mathcal{V}$ for $k \geqslant k^*$, that is, the network configuration converges. □

Remark 2: Note that $\mathrm{CC}(\mathbb{V}_{i,k})$ defined in Proposition 1 for a region $\mathbb{V}_{i,k}$ is different from the common definition of the circumcentre of a triangle which can be located out of the triangle. In our case, $\mathrm{CC}(\mathbb{V}_{i,k}) \subset \mathbb{V}_{i,k}$ always holds so that agent i will not run out of $\mathbb{V}_{i,k}$ at time $k+1$ by moving towards $\mathrm{CC}(\mathbb{V}_{i,k})$. However, there might be cases that $\mathrm{CC}(\mathbb{V}_{i,k}) = \mathrm{CC}(\mathbb{V}_{j,k}) \in \partial \mathbb{V}_{i,k}$ (See Fig. 3*a*). If this happens, agents i and j would converge to the same place at time $k+1$ following (8), where they collide with each other. To avoid collision of neighbouring agents, we have to introduce extra control influence which will be discussed in the following part.

Remark 3: Theorem 1 only shows the convergence property of the control law (15), and a near local optimum network configuration can finally be achieved. However, such local optimum configuration may be far from what we want (See Fig. 3). In the following part, we will show that a better solution can be obtained by optimising the initial configuration in (2).

Remark 4: The control law $u_{i,k}$ given by (15) relies on the correct geometric information of the Voronoi cell $\mathbb{V}_{i,k}$, which requires that agent i know the position information of all its Voronoi neighbours that contribute to the generation of $\mathbb{V}_{i,k}$ (i.e. the agents in the set $\{j : \mathbb{V}_{j,k} \bigcap \mathbb{V}_{i,k} \neq \emptyset\}$). In a distributed network, each agent i can only communicate with its one-hop neighbours (i.e. the agents in $\mathcal{N}_{i,k}$). If all Voronoi neighbours of agent i are in $\mathcal{N}_{i,k}$ at each time k, an Adjust Sensing Radius algorithm has been proposed in [27] to determine the Voronoi neighbours and generate $\mathbb{V}_{i,k}$. However, there has been no algorithm to guarantee the precondition holds all the time. Therefore we need to design a distributed topology control algorithm to control the network topology adaptively such that all Voronoi neighbours of agent i belong to $\mathcal{N}_{i,k}$ at each time, which will be discussed in the next section.

3.2 Modifications of the control strategy

In this part, we will modify the control law (15) to avoid collision between agents and improve the optimisation performance as pointed out in Remarks 2 and 3.

3.2.1 Collision avoidance: Fig. 3*a* shows an example of two agents configuration in a squared region where collision between the agents happens when i and j both converge to $\mathrm{CC}(\mathbb{V}_{i,k})$. To avoid collision between two agents, the easiest way is to stop them from getting closer than a given threshold distance. For any two neighbouring agents i and j with a user given distance threshold ϵ (i.e. to require the distance between i and j be no less than ϵ), we can define a dead zone related to i and j (see (20))

Thus, the overall dead zone for i is given by

$$\mathrm{D}_{i,k} = \bigcup_{j \in \mathcal{N}_{i,j,k}} D_{i,j,k} \tag{21}$$

Then, we can modify the control law (15) as follows (see (22))

$$u_{i,k} = \begin{cases} 0, & \text{if } H_{i,t_k-T} - H_{i,t_k} \leqslant \delta \text{ and } k \geqslant 2T - 1 \\ G_{i,k}(\mathrm{CC}(\mathbb{V}_{i,k}) - \mu_{i,k}), & \text{otherwise} \end{cases} \tag{15}$$

$$D_{i,j,k} = \left\{ q : \frac{\|\mu_{j,k} - \mu_{i,k}\|}{2} - \frac{(q - \mu_{i,k})^{\mathrm{T}}(\mu_{j,k} - \mu_{i,k})}{\|\mu_{j,k} - \mu_{i,k}\|} < \epsilon, \quad q \epsilon \mathbb{V}_{i,k} \right\} \tag{20}$$

$$u_{i,k} = \begin{cases} 0, & \text{if } H_{i,t_k-T} - H_{i,t_k} \leqslant \delta \text{ and } k \geqslant 2T - 1 \\ \lambda_{i,k} G_{i,k}(\mathrm{CC}(\mathbb{V}_{i,k}) - \mu_{i,k}), & \text{otherwise} \end{cases} \tag{22}$$

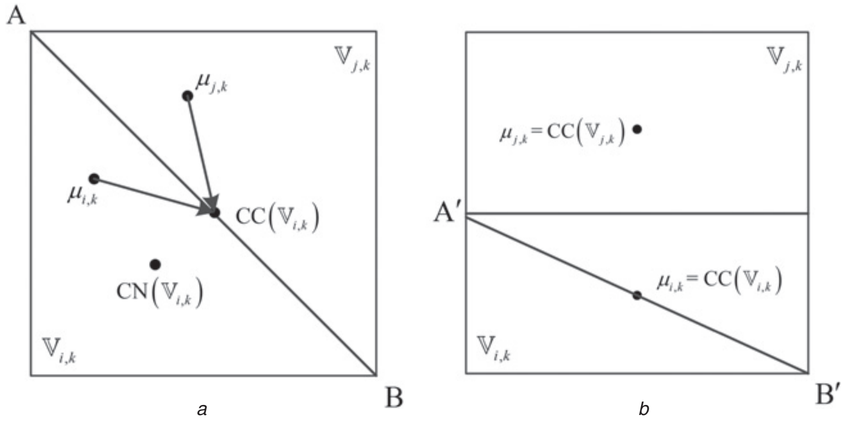

Fig. 3 *Convergence into different configuration*
a Suboptimal configuration: $\mathcal{H}_{\mathrm{dc}} = \frac{1}{2}\|\mathrm{A} - \mathrm{B}\|$
b Optimal configuration: $\mathcal{H}_{\mathrm{dc}} = \frac{1}{2}\|\mathrm{A}' - \mathrm{B}'\|$

The scaling factor $\lambda_{i,k}$ is determined by (see (23))

where $\bar{\mu}_{i,k+1}(\lambda) = \mu_{i,k} + \lambda G_{i,k}(\mathrm{CC}(\mathbb{V}_{i,k}) - \mu_{i,k})$. Through the scaling factor, it can be guaranteed that any two neighbouring agents i and j within the dead zone $D_{i,j,k}$ at time k will not get closer than their current distance at next time step, because $\lambda_{i,k} = 0$ (or $\lambda_{j,k} = 0$ resp.); if i tends to move closer to $\mu_{j,k}$ (or j tends to move closer to $\mu_{i,k}$) which implies $u_{i,k} = 0$ (or $u_{j,k} = 0$). Furthermore, any two neighbouring agents i and j out of $D_{i,j,k}$ at time k will never move into $D_{i,j,k}$ at next time step, because $\mu_{i,k+1} = \bar{\mu}_{i,k+1}(\lambda_{i,k}) \notin D_{i,j,k}$ and $\mu_{j,k+1} = \bar{\mu}_{j,k+1}(\lambda_{j,k}) \notin D_{i,j,k}$ are always satisfied by the selection rule of $\lambda_{i,k}$ and $\lambda_{j,k}$. Therefore the collision between i and j is avoided by preventing them from touching the borders of their own Voronoi cells.

3.2.2 Initial optimisation: $\mathcal{H}_{\mathrm{dc}}(\mu_{1,k}, \ldots, \mu_{N,k})$ is generally a multi-centre function and one can only obtain a local optimum solution related to initial configurations as pointed in [26]. An unappropriate initial configuration may lead the agents to a suboptimal solution far away from the optimal solution, an example of which is shown in Fig. 3. Therefore we need to optimise the initial configuration for optimising $\mathcal{H}_{\mathrm{dc}}$. Since $\mathcal{H}_{\mathrm{dc}}$ denotes the maximum value of $\|q - \mu_i\|$, we want to optimise the initial configuration such that $\|q - \mu_i\|$ is minimised in some mean sense. We solve the following optimisation problem to obtain the initial configuration

$$\begin{aligned} \text{minimise} \quad \mathcal{H}_{\mathrm{ic}}(\mu_1, \ldots, \mu_N) &= \frac{1}{N}\int_{\mathbb{O}} \min_{i\in\mathcal{V}} \|q - \mu_i\|^2 dq \\ &= \frac{1}{N}\sum_{i\in\mathcal{V}} \int_{\mathbb{V}_i} \|q - \mu_i\|^2 \, \mathrm{d}q \end{aligned} \tag{24}$$

An iterative control scheme has been proposed in [27] to obtain a local optimum solution

$$\begin{aligned} \mu_{i,k+1} &= \mu_{i,k} + u^*_{i,k} \\ u^*_{i,k} &= L_{i,k}(\mathrm{CN}(\mathbb{V}_{i,k}) - \mu_{i,k}) \end{aligned} \tag{25}$$

where $u^*_{i,k}$ is the control law and $0 < L_{i,k} \leqslant 1$ is the gain parameter. $\mathrm{CN}(\mathbb{V}_{i,k})$ is the centroid of $\mathbb{V}_{i,k}$ defined as

$$\mathrm{CN}(\mathbb{V}_{i,k}) = \frac{\int_{\mathbb{V}_{i,k}} q \, \mathrm{d}q}{\int_{\mathbb{V}_{i,k}} \mathrm{d}q} \tag{26}$$

It is proven in [27] that the network configuration converges to centroidal configuration, that is, $\mu_{i,k}$ converges to $\mathrm{CN}(\mathbb{V}_{i,k})$ $\forall i \in \mathcal{V}$. Thus, before implementing control law (22) to minimise $\mathcal{H}_{\mathrm{dc}}$, we first implement control law (25) to minimise $\mathcal{H}_{\mathrm{ic}}$. It is not necessary to wait until $\mu_{i,k}$ converges to $\mathrm{CN}(\mathbb{V}_{i,k})$ before starting to implement (22) and users can define a duration for the initial optimisation.

4 K-Connectivity maintenance

The key idea of maintaining the network connectivity is to restrict the allowable motion of each agent, that is, to find a set $\chi_{i,k}$ of the control inputs for each agent such that the network is still connected at time $k+1$ by selecting $u_{i,k} \in \chi_{i,k}$ for each agent if the network is connected at time k. In [4], a distributed topology control algorithm has been proposed to maintain the network connectivity. However, it cannot be used for K-connectivity maintenance where $K > 1$. Here, we modify the algorithm so that it can be applied in the K-connectivity maintenance.

Consider two agents i and j at time k that satisfy $\|\mu_{i,k} - \mu_{j,k}\| \leqslant R_{\mathrm{C}}$. In [4], a pairwise connectivity constraint set of agent i with respect to agent j ($i \neq j$) is defined as

$$\Omega_{i,j,k} \triangleq \left\{ \xi \in \mathbb{O} : \left\| \xi - \frac{\mu_{i,k} + \mu_{j,k}}{2} \right\| \leqslant \frac{R_{\mathrm{C}}}{2} \right\}$$

The control input constraint set of agent i is defined as

$$\chi_{i,k} = \left\{ u : \mu_{i,k} + u \in \bigcap_{j \in \overline{\mathcal{N}}_{i,k}} \Omega_{i,j,k} \right\} \tag{27}$$

$$\lambda_{i,k} = \begin{cases} 0, & \text{if } \mu_{i,k}, \, \bar{\mu}_{i,k+1}(1) \in \mathrm{D}_{i,k} \\ \underset{0 \leqslant \lambda \leqslant 1, \bar{\mu}_{i,k+1}(\lambda) \notin \mathrm{D}_{i,k}}{\arg\max} \|\bar{\mu}_{i,k+1}(\lambda) - \mu_{i,k}\|, & \text{otherwise} \end{cases} \tag{23}$$

doi: 10.1049/iet-cta.2012.0779

where

$$\overline{\mathcal{N}}_{i,k} = \left\{ j \in \mathcal{N}_{i,k} \setminus \{i\} : \min_{s \in \mathcal{N}_{i,k} \bigcap \mathcal{N}_{j,k}} s = i \text{ or } j \right\} \tag{28}$$

Remark 5: By the application of $\overline{\mathcal{N}}_{i,k}$, it can be realised that the connection between any two neighbouring agents i and j is surely maintained from time k to $k+1$, unless there exists a third agent l being a neighbour of both i and j and subject to $l < i, j$. The physical meaning of (27) with $\overline{\mathcal{N}}_{i,k}$ is that the maintenance of only one connection path between any two neighbouring agents (through their smallest common neighbour) is set as a constraint for the motion control of the two agents. In this way, the redundant paths or links are not set as constraints and thus may not be maintained. Therefore $\overline{\mathcal{N}}_{i,k}$ only includes the critical neighbours that need to be maintained connected for agent i.

Proposition 2 [4]: If the network is connected at time k, then the network is still connected at time $k+1$ by selecting $u_{i,k} \in \chi_{i,k}$ for each agent.

To maintain the K-connectivity for the whole network, the key is to find out the set of neighbours for each agent i to keep connection with at each time. For a pair of agents $\{i, j\}$, we can define the following set

$$A_{i,j,k} = \left\{ l : \; l \in (\mathcal{N}_{i,k} \bigcap \mathcal{N}_{j,k}) \setminus \{i\} \right\}$$

The elements of $A_{i,j,k}$ can be written in an ordered sequence such that $l_1 < l_2 < \cdots < l_{|A_{i,j,k}|}$. Let $a_{i,j,k} = \min(K, |A_{i,j,k}|)$ and further define the following subset of $A_{i,j,k}$

$$B_{i,j,k} = \{l_1, l_2, \ldots, l_{a_{i,j,k}}\} \subset A_{i,j,k} \tag{29}$$

Then, we can obtain a new subset of neighbours of agent i

$$\widetilde{\mathcal{N}}_{i,k} = \bigcup_{j \in \mathcal{N}_{i,k} \setminus \{i\}} B_{i,j,k} \tag{30}$$

Lemma 2: If the network is connected at time k and $a_{i,j,k} \geqslant K \;\forall i \in \mathcal{V}$ and $j \in \mathcal{N}_{i,k} \setminus \{i\}$, then the network is K-connected at time k and maintains the K-connectivity at time $k+1$ by selecting $u_{i,k} \in \tilde{\chi}_{i,k} \;\forall i \in \mathcal{V}$, where $\tilde{\chi}_{i,k}$ is given by

$$\tilde{\chi}_{i,k} = \left\{ u : \mu_{i,k} + u \in \bigcap_{j \in \widetilde{\mathcal{N}}_{i,k}} \Omega_{i,j,k} \right\} \tag{31}$$

Proof: According to the definition of $a_{i,j,k}$, we have that there exist at least K paths connecting i and j $(j \in \mathcal{N}_{i,k} \setminus \{i\})$. Furthermore, since the network is connected, there exist at least K paths connecting any two agents i and l $(i, l \in \mathcal{V}, i \neq l)$ in the network. Otherwise, since any path from i to $l \in \mathcal{V} \setminus \mathcal{N}_{i,k}$ can be divided into two parts, that is, the part from i to some $j \in \mathcal{N}_{i,k} \setminus \{i\}$ and the part from j to l, we can obtain $a_{i,j,k} < K$. This is contradictory to the former result that $a_{i,j,k} \geqslant K$. Therefore the network is K-connected at time k.

On the other hand, since $\overline{\mathcal{N}}_{i,k} \subset \widetilde{\mathcal{N}}_{i,k}$, following Proposition 2, by selecting $u_{i,k} \in \tilde{\chi}_{i,k}$ for each agent i, we can guarantee that the network connectivity is maintained at time $k+1$ and $a_{i,j,k+1} \geqslant K \;\forall i \in \mathcal{V}$ and $j \in \mathcal{N}_{i,k} \setminus \{i\}$, which implies that the network is K-connected at time $k+1$. □

Theorem 2: If the network is all-to-all connected at time $k = 0$, that is, $d_{i,0} = N \;\; \forall i \in \mathcal{V}$, then the network is K-connected for $k > 0$ by selecting $u_{i,k} \in \tilde{\chi}_{i,k} \;\forall i \in \mathcal{V}$.

Proof: All-to-all connectivity at time $k = 0$ implies that the network is connected at the time and $a_{i,j,0} \geqslant K$. Based on the conclusion of Lemma 2, we can obtain the network that is K-connected for all $k > 0$. □

To solve the problem mentioned in Remark 4, we need to further modify $\tilde{\chi}_{i,k}$ to maintain the connection between Voronoi neighbours. The set of Voronoi neighbours of agent i at time k is defined as

$$V_{i,k} = \{j : \mathbb{V}_{j,k} \bigcap \mathbb{V}_{i,k} \neq \emptyset, i \neq j\} \tag{32}$$

Our next goal is to make all agents in $V_{i,k}$ K-connected with agent i at each time k.

Theorem 3: If the network is all-to-all connected at time $k = 0$, that is, $d_{i,0} = N \;\; \forall i \in \mathcal{V}$, then $V_{i,k} \subset \mathcal{N}_{i,k}$ and the network is K-connected for $k > 0$ by selecting $u_{i,k} \in \hat{\chi}_{i,k}$ $\forall i \in \mathcal{V}$, where $\hat{\chi}_{i,k}$ is given by

$$\hat{\chi}_{i,k} = \left\{ u : \mu_{i,k} + u \in \bigcap_{j \in \widetilde{\mathcal{N}}_{i,k} \bigcup (\mathcal{N}_{i,k} \bigcap V_{i,k})} \Omega_{i,j,k} \right\} \tag{33}$$

Proof: First, we show that $V_{i,k} \subset \mathcal{N}_{i,k}$ implies that $V_{i,k+1} \subset \mathcal{N}_{i,k+1}$. If $V_{i,k} \subset \mathcal{N}_{i,k}$, we have $\mathcal{N}_{i,k} \bigcap V_{i,k} = V_{i,k}$. From (33) we can obtain $V_{i,k} \subset \mathcal{N}_{i,k+1}$. On the other hand, according to the relation between Voronoi diagram and Delaunay triangulation that every three vertices of a Delaunay triangle are Voronoi neighbours [28], we obtain that the edges of all Delaunay triangles are no greater than the communication range R_C. Meanwhile, within each Delaunay triangle there exists no other agent. Therefore if an agent $j \notin V_{i,k}$ becomes a Voronoi neighbour of agent i at time $k+1$, that is, $j \in V_{i,k+1}$, j must be within a triangle formed by i and other two agents within $V_{i,k}$, which implies that $\|\mu_{i,k+1} - \mu_{j,k+1}\| \leqslant R_C$, that is, $j \in \mathcal{N}_{i,k+1}$. From this, we conclude that $V_{i,k} \subset \mathcal{N}_{i,k}$ implies that $V_{i,k+1} \subset \mathcal{N}_{i,k+1}$. Further, with the condition that $V_{i,0} \subset \mathcal{N}_{i,0}$, we can obtain $V_{i,k} \subset \mathcal{N}_{i,k} \;\forall k > 0$ by deduction.

Next, we show the K-connectivity maintenance by selecting $u_{i,k} \in \hat{\chi}_{i,k} \;\forall i \in \mathcal{V}$. Since $\widetilde{\mathcal{N}}_{i,k} \subset \widetilde{\mathcal{N}}_{i,k} \bigcup (\mathcal{N}_{i,k} \bigcap V_{i,k})$ implies $\tilde{\chi}_{i,k} \supset \hat{\chi}_{i,k}$, the K-connectivity must be maintained following Theorem 2. □

Following the conclusion of Theorem 3, we need to incorporate the control input constraint $u_{i,k} \in \hat{\chi}_{i,k} \;\forall i \in \mathcal{V}$ for the sake of K-connectivity maintenance under specified initial conditions. One way of confining $u_{i,k}$ within $\hat{\chi}_{i,k}$ while avoiding collisions between neighbours is to modify the scaling factor $\lambda_{i,k}$ defined in (23) of the control law (22) as follows (see (34))

$$\lambda_{i,k} = \begin{cases} 0, & \text{if } \mu_{i,k}, \; \bar{\mu}_{i,k+1}(1) \in \mathrm{D}_{i,k} \\ \arg \max\limits_{0 \leqslant \lambda \leqslant 1, \; \bar{\mu}_{i,k+1}(\lambda) \in \hat{\chi}_{i,k} \setminus \mathrm{D}_{i,k}} \|\bar{\mu}_{i,k+1}(\lambda) - \mu_{i,k}\|, & \text{otherwise} \end{cases} \tag{34}$$

Algorithm 1

Start with $H_{i,0} = \max_{q \in \partial \mathbb{M}_i} \|q - \mu_{i,0}\|$, $\alpha_{i,0} = H_{i,0}$, $\beta_{i,0} = 1$ (α_i and β_i are auxiliary variables) and $k = 0$ for all i:

Communicate with neighbours to get $\mu_{j,k}$ and $H_{j,k-1}$ for all $j \in \mathcal{N}_{i,k}$ (no need to include $H_{j,k}$ in the message when $\mathrm{mod}\,(k, T) = 0$);
Compute $\mathbb{V}_{i,k}$, and $H_{i,k}$ by (14);
If $\mathrm{mod}\,(k+1, T) = 0$ and $k \geqslant 2T - 1$
    $\beta_{i,k+1} = \alpha_{i,k} - H_{i,k} - \delta$;
Else
    $\beta_{i,k+1} = \beta_{i,k}$; (i.e., no need to update β_i)
End
If $\mathrm{mod}\,(k+1, T) = 0$
    $\alpha_{i,k+1} = H_{i,k}$;
Else
    $\alpha_{i,k+1} = \alpha_{i,k}$; (i.e., no need to update α_i)
End
If $\beta_{i,k+1} \leqslant 0$
    Set $u_{i,k} = 0$;
Else
    Compute $\mathrm{CC}\,(\mathbb{V}_{i,k})$, $G_{i,k}$ by (9), $\mathrm{D}_{i,k}$ by (21) and $\hat{\chi}_{i,k}$ by (33) respectively;
    Compute $\lambda_{i,k}$ by (34);
    Set $u_{i,k} = \lambda_{i,k} G_{i,k} (\mathrm{CC}\,(\mathbb{V}_{i,k}) - \mu_{i,k})$;
End
Move to new position $\mu_{i,k+1} = \mu_{i,k} + u_{i,k}$;
$k \leftarrow k + 1$.

Fig. 4 *Computation of control law*

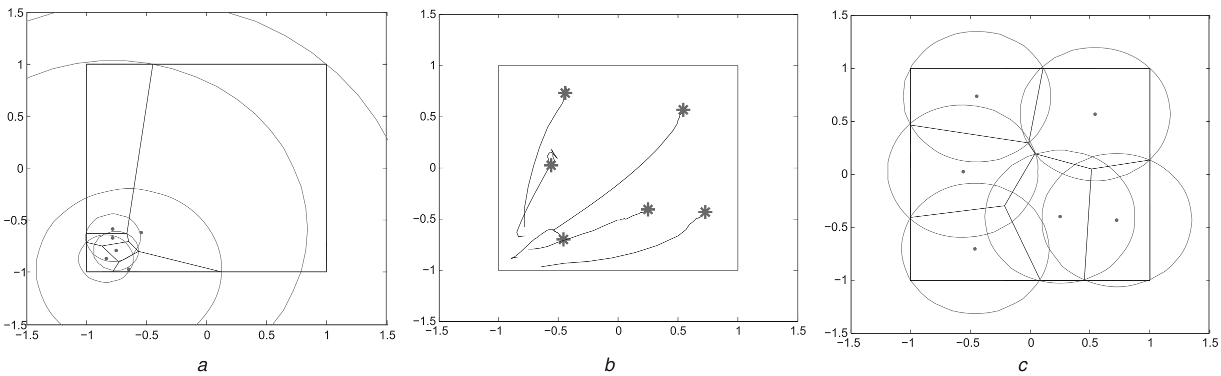

Fig. 5 *Initial and final configurations and moving trajectories of agents (the smallest discs centred at the agents covering the corresponding Voronoi cells are enclosed by red circles and converged positions of agents are denoted by blue stars)*

a Initial configuration
b Moving trajectories
c Final configuration

The algorithm to compute the control law at each time k based on the above discussions is shown by Algorithm 1 (see Fig. 4), where all the variables are iteratively updated.

Remark 6: The significance of Theorem 3 is that it makes the Voronoi cell calculated by agent i in a distributed manner be the same with $\mathbb{V}_{i,k}$ which is calculated by knowing the positions of all agents. That is to say the control law (22) is distributed since the calculation of $u_{i,k}$ only requires the information from the neighbours of agent i which includes $H_{j,k}$ and $\mu_{j,k}$ ($j \in \mathcal{N}_{i,k}$), where $\mathbb{V}_{i,k}$ can be calculated from $\mu_{j,k}$.

Remark 7: Note that the assumption of all-to-all connectivity is required at the initial step in Theorems 2 and 3, which is usually easy to satisfy in that the initial condition can be controlled by users. For example, the agents can be deployed within a small neighbourhood at the beginning. Since network deployment and task allocation are distributed (i.e. only one-hop communication can be used) and require that each agent know the positions of all agents that share Voronoi partition borders with it at each time in order to calculate the Vronoi cells, any two agents sharing a Voronoi partition border must be maintained as one-hop neighbours. To make this requirement satisfied at each time, we need to set all-to-all connectivity at the beginning. The condition of only K-connectivity is generally not enough to make sure such requirement is satisfied, because two agents sharing a Voronoi partition border may not be one-hop neighbours although they are K-connected.

IET Control Theory Appl., 2013, Vol. 7, Iss. 11, pp. 1574–1582
doi: 10.1049/iet-cta.2012.0779

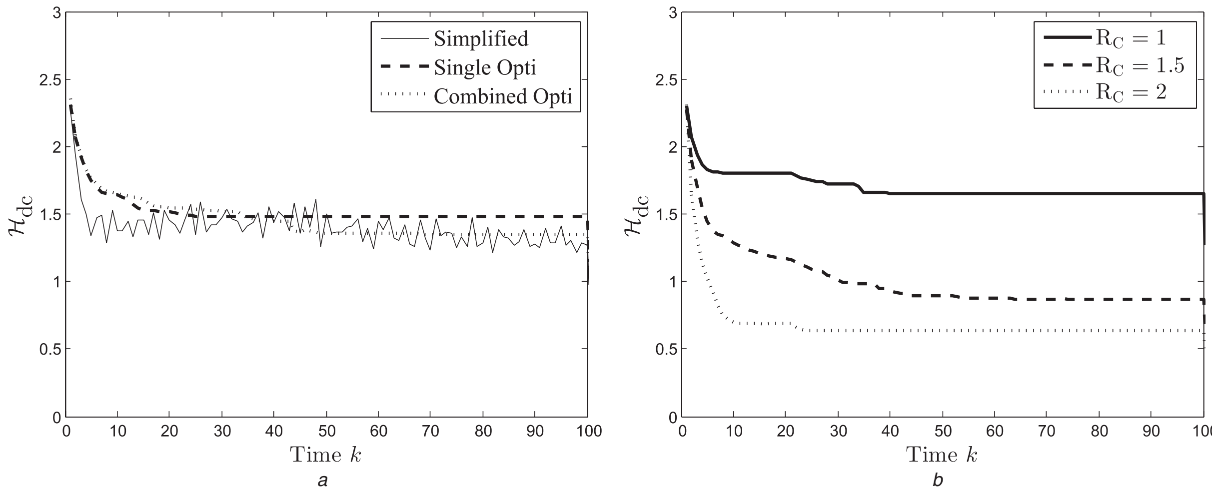

Fig. 6 *Simulation results*

a Results of Scenario I
b Results of Scenario II

5 Simulation

In this section, we test the proposed distributed control strategies by simulation. We deploy six agents in a squared region $[-1, 1]^2$, which are initially placed within region $[-1, -0.5]^2$, so that they are all-to-all connected at time $k = 0$. The Voronoi partition is used as the initial partition. Other related parameters are, respectively, set as $T = N$, $\delta = 0.01$ and $\epsilon = 0.1$. The control law is computed by Algorithm 1 (see Fig. 4). In Scenario I, we compare the performance of three different control strategies: the 'simplified strategy' given in [26], that is, (8) without considering the limited communication and moving capabilities of agents, the proposed 'single optimisation strategy' (22), and the modified 'combined optimisation strategy' by implementing (25) in the first 20 steps followed by the implementation of (22). The communication range is fixed at $R_C = 1$. In Scenario II, we test the influence of the topology control defined by (33) and compare the results with different communication ranges by the 'combined optimisation strategy'. In both of the two scenarios, we set $K = 1$, that is, only requiring one-connectivity. For each simulation, we run 100 Monte Carlo simulations to obtain the averaged results.

First, an example of the convergence of network configuration is shown in Fig. 5, where the trajectories of all agents and the smallest discs covering the corresponding Voronoi cells in the initial and final stages are shown. Fig. 6*a* shows the results of $\mathcal{H}_{\rm dc}$ in Scenario I. It can be seen that the 'simplified strategy' cannot guarantee the convergence with limited communication and moving capabilities. While the convergence of network configuration is guaranteed by both of our proposed control strategies, the 'combined optimisation strategy' performs better than the 'single optimisation strategy' by providing a smaller $\mathcal{H}_{\rm dc}$. The influence of the topology control on the optimisation performance is shown in Fig. 6*b*. We can see that a smaller communication range may result in a larger $\mathcal{H}_{\rm dc}$. This is because of that the topology control confines movements of agents, so that they may not reach the local optimum positions in order to maintain network connectivity. According to (34), the control law is affected by topology control through the scaling factor $\lambda_{i,k}$, where a smaller $\lambda_{i,k}$ implies a smaller value of $\|u_{i,k}\|$. When R_C is smaller, so is the control input constraint set $\hat{\chi}_{i,k}$, that is, the constraint on $u_{i,k}$ is more restrictive, which usually implies a smaller $\lambda_{i,k}$ and thus a smaller $\|u_{i,k}\|$. In the meantime, it is illustrated by Fig. 6*b* that the proposed strategy is adapted to the communication capability of agents and can guarantee convergence in either case.

6 Conclusions

In this paper, the distributed cooperative control of UAV networks was addressed for deployment and task allocation. Each agent takes charge of the task accomplishment within a part of the whole region. The control strategy was designed to minimise the maximum travelling time the agents take to reach a place within the surveillance region, which is constrained by limited communication range and mobility for each agent. It has been proven that by our control strategy, the network configuration converges to the local optimum configuration. Collision avoidance was considered and a combined optimisation approach was developed to improve the optimisation performance. Moreover, a K-connectivity maintenance algorithm is proposed to guarantee the convergence of network configuration. Simulation validates the effectiveness of the designed control strategy. In our future work, research will be focused on generalising the proposed control strategy to more complex UAV models and finding new methods to accelerate the convergence speed.

7 References

1 Polycarpou, M., Yang, Y., Passino, K.: 'A cooperative search framework for distributed agents'. Proc. IEEE Int. Symp. on Intelligent Control, 2001, pp. 1–6
2 Yang, Y., Minai, A., Polycarpou, M.: 'Decentralized cooperative search by networked UAVs in an uncertain environment'. Proc. American Control Conf., vol. 6, 2004, pp. 5558–5563
3 Stone, L.: 'Theory of optimal search' (Academic Press, New York, 1975)
4 Hu, J., Xie, L., Lum, K.-Y., Xu, J.: 'Multiagent information fusion and cooperative control in target search', *IEEE Trans. Control Syst. Technol.*, 2012, **l**, (99), p. 1
5 How, J., King, E., Kuwata, Y.: 'Flight demonstrations of cooperative control for UAV teams'. Proc. AIAA Third 'Unmanned Unlimited' Technical Conf., Workshop and Exhibit, vol. 2012, 2004, p. 9
6 Gazi, V., Passino, K.: 'Stability analysis of social foraging swarms: combined effects of attractant/repellent profiles'. Proc. IEEE Conf. on Decision and Control, vol. 3, 2002, pp. 2848–2853

7 Burian, E., Yoerger, D., Bradley, A., Singh, H.: 'Gradient search with autonomous underwater vehicles using scalar measurements'. Proc. Symp. Autonomous Underwater Vehicle Technology, 1996, pp. 86–98

8 Hoskins, D.: 'Least action approach to collective behavior'. Proc. Society of Photo-Optical Instrumentation Engineers Conf., vol. 2593, 1995, pp. 108–120

9 Ogren, P., Fiorelli, E., Leonard, N.: 'Formations with a mission: stable coordination of vehicle group maneuvers'. Proc. 15th Int. Symp. Mathematical Theory of Networks and Systems, vol. 170, 2002

10 Bachmayer, R., Leonard, N.: 'Vehicle networks for gradient descent in a sampled environment (i)'. Proc. IEEE Conf. Decision and Control, vol. 1, 2001, pp. 112–117

11 Ogren, P., Fiorelli, E., Leonard, N.: 'Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment', *IEEE Trans. Autom. Control*, 2004, **49**, (8), pp. 1292–1302

12 Tan, J.: 'A scalable graph model and coordination algorithms for multirobot systems'. Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, 2005, pp. 1529–1534

13 Olfati-Saber, R.: 'Flocking for multi-agent dynamic systems: algorithms and theory', *IEEE Trans. Autom. Control*, 2006, **51**, (3), pp. 401–420

14 Martínez, S., Bullo, F.: 'Optimal sensor placement and motion coordination for target tracking', *Automatica*, 2006, **42**, (4), pp. 661–668

15 Lynch, K., Schwartz, I., Yang, P., Freeman, R.: 'Decentralized environmental modeling by mobile sensor networks', *IEEE Trans. Robot.*, 2008, **24**, (3), pp. 710–724

16 Choi, J., Oh, S., Horowitz, R.: 'Distributed learning and cooperative control for multi-agent systems', *Automatica*, 2009, **45**, (12), pp. 2802–2814

17 Cortés, J.: 'Distributed Kriged Kalman filter for spatial estimation', *IEEE Trans. Autom. Control*, 2009, **54**, (12), pp. 2816–2827

18 Nygard, K., Chandler, P., Pachter, M.: 'Dynamic network flow optimization models for air vehicle resource allocation'. Proc. American Control Conf., vol. 3, 2001, pp. 1853–1858

19 Jin, Y., Minai, A., Polycarpou, M.: 'Cooperative real-time search and task allocation in UAV teams'. Proc. IEEE Conf. on Decision and Control, vol. 1, 2003, pp. 7–12

20 Beard, R., McLain, T., Goodrich, M., Anderson, E.: 'Coordinated target assignment and intercept for unmanned air vehicles', *IEEE Trans. Robot. Autom.*, 2002, **18**, (6), pp. 911–922

21 Ben-Asher, Y., Feldman, S., Gurfil, P., Feldman, M.: 'Distributed decision and control for cooperative UAVs using ad hoc communication', *IEEE Trans. Control Syst. Technol.*, 2008, **16**, (3), pp. 511–516

22 Tanner, H., Jadbabaie, A., Pappas, G.: 'Stable flocking of mobile agents, part I: fixed topology'. Proc. IEEE Conf. on Decision and Control, vol. 2, 2003, pp. 2010–2015

23 Tanner, H., Jadbabaie, A., Pappas, G.: 'Stable flocking of mobile agents part II: Dynamic topology'. Proc. IEEE Conf. on Decision and Control, vol. 2, 2003, pp. 2016–2021

24 Wang, L., Xiao, F.: 'Finite-time consensus problems for networks of dynamic agents', *IEEE Trans. Autom. Control*, 2010, **55**, (4), pp. 950–955

25 Keviczky, T., Borrelli, F., Fregene, K., Godbole, D., Balas G.: 'Decentralized receding horizon control and coordination of autonomous vehicle formations', *IEEE Trans. Control Syst. Technol.*, 2008, **16**, (1), pp. 19–33

26 Bullo, F., Cortés, J., Martinez, S.: 'Distributed control of robotic networks' (Princeton University Press, 2009)

27 Cortes, J., Martinez, S., Karatas, T., Bullo, F.: 'Coverage control for mobile sensing networks', *IEEE Trans. Robot. Autom.*, 2004, **20**, (2), pp. 243–255

28 Fortune, S.: 'Voronoi diagrams and delaunay triangulations', *Comput. Euclidean Geom.*, 1992, **1**, (193–233), p. 2

IET Control Theory Appl., 2013, Vol. 7, Iss. 11, pp. 1574–1582
doi: 10.1049/iet-cta.2012.0779