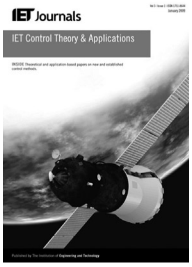
www.ietdl.org

Published in IET Control Theory and Applications
Received on 10th May 2014
Revised on 20th August 2014
Accepted on 25th August 2014
doi: 10.1049/iet-cta.2014.0469

Special Issue on Co-operative Multi-Agent Systems with Engineering Applications



ISSN 1751-8644

Integrated multi-agent system framework: decentralised search, tasking and tracking

Wei Meng[1]*, Zhirong He*[2]*, Rodney Teo*[1]*, Rong Su*[3]*, Lihua Xie*[3]

[1]*Temasek Laboratories, National University of Singapore, Singapore*
[2]*School of Mathematics, Sichuan University, Chengdu, Sichuan, People's Republic of China*
[3]*EXQUISITUS, Centre for E-City, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore*
E-mail: tslmeng@nus.edu.sg

Abstract: This study is concerned with integrated target search, tasking and tracking using multiple fixed-wing unmanned air vehicles (UAVs). The problem is to design control logic and optimise flight paths for UAVs. The fixed-wing UAVs are required to cooperatively search the potential targets in the area of operation (AO) and keep monitoring and tracking the found targets according to a certain predefined minimal revisit time. Each UAV can only communicate with its neighbours and also flight autonomy is designed for individual UAVs. High level control logic based on finite state automaton model, integrating the four modes of operations, that is, take-off mode, fly-to-AO mode, search mode and tracking mode, is developed. For the target search, an improved decentralised coverage based algorithm is designed for fixed-wing UAVs. Task assignment is developed based on contract net protocol. For the target tracking, the optimal paths are derived for the UAVs to continuously monitoring multiple static targets. The developed integrated multi-agent system is evaluated by a hybrid three-dimensional simulation system using a real miniature fixed-wing UAV dynamic model.

1 Introduction

Distributed optimisation and control are hot topics in the field of multi-agent systems [1–4]. Teams of unmanned air vehicles (UAVs) equipped with sensors (such as camera, two-dimensional Lidar, sonar) are currently being developed to facilitate area surveillance, coverage, search and rescue [5–10]. Decentralised control of UAVs makes it possible to autonomous tasking which can reduce the burden of operators [3, 11]. UAVs can cooperate when their common objectives are well defined and each UAV has the information needed to cooperate, even when there are communication difficulties and failures. In this paper, we are interested in multi-UAV target search and tracking problems. As we know, UAV's camera view is often blocked or interfered by tall buildings, trees and so on. The monitoring tasks become quite challenging which require UAVs autonomously and cooperatively search and tracking targets of interest. In addition, task assignment also requires to be carried out in a decentralised way by UAVs themselves. Efficient decentralised deployment strategies need to be designed for UAVs to finish the tasks cooperatively and time efficiently.

Deployment of UAVs for coverage search and tracking has been studied extensively [3, 5, 7, 12–15]. In [3], control and coordination algorithms for groups of vehicles are presented. Distributed coverage control algorithm is developed based on locational optimisation method. In [14], a fixed-wing UAV is used for searching and mapping boundaries of a river. In [5], a vision-based persistent search and track using multiple UAVs is addressed. The main contribution therein is target detection (based on video data) and task assignment methods. In [15], the authors consider the problem of keeping track of all discovered targets and simultaneously search for new targets by controlling the pointing direction of the vision sensor and the motion of the UAV. In this research work, only one UAV is used. Furukawa *et al*. [12] present a coordinated control technique that allows heterogeneous vehicles to autonomously search and track multiple targets using recursive Bayesian filtering. UAVs can switch their operational mode from search to track. However, the proposed algorithm is centralised. In [7], a distributed target search algorithm using multiple UAVs is developed. However, the developed gradient control laws only can be applied to omni-directional vehicles. To the best of our knowledge, there is no much work on integrated design of search, tasking and tracking with multiple fixed-wing UAVs. Moreover, we also consider multi-UAV autonomous and decentralised take-off problem.

The main objective of this paper is to design autonomy for fixed-wing UAVs in cooperative search and track problems. One important characteristic unique to fixed-wing UAVs is that forward motion is required. Therefore stop-and-wait path de-confliction algorithms are not applicable. Most of the existing works on coverage search can only be applied to omni-directional vehicles. However, due to speed and turning rate constraints, the effectiveness and efficiency cannot meet the requirements of practical usage. In [16, 17], the authors have considered optimal multi-agent trajectory optimisation problem for non-holonomic mobile agents. Agent modelling and adaptive methods are developed, respectively, for coverage control problem. In [18], map exploration problem is formulated as a constrained optimisation problem and reinforcement learning method is used to find trajectories for robots that lead to accurate maps.

Our case study in this work consists of a team of fixed-wing UAVs performing search and tracking tasks over a surveillance environment. The boundary of the search region is assumed to be known to all UAVs. We present the overall high-level control logic design, integrating the four modes of operations, that is, take-off mode, fly-to-AO mode, search mode and tracking mode. Each mode is implemented using several module processes running concurrently, supporting communication, coordination. Our aim is to provide decisional architectures for multi-UAV systems which act autonomously with minimal supervision from human beings. In addition, an efficient coverage search algorithm is developed for fixed-wing UAVs taking into account UAV travelling time and multi-UAV de-confliction. Autonomous task assignment is designed based on contract net protocol [19]. An optimal plan planning for continuous multiple target tracking and monitoring is also proposed.

The remainder of this paper is organised as follows. The problem description is given in Section 2. Decentralised search, tasking and tracking algorithms are addressed in Section 3. In Section 4, simulation results are presented. Section 5 concludes the paper.

2 Search and track problem description

In this work, we consider a general search and track scenario. Teams of UAVs are deployed to perform search and track-related tasks. At first, an operator marks out the boundary of an area for search. There exist multiple static targets (may change to be mobile targets) placed in the area of operation (AO).

The operator uploads mission commands to all UAVs while they are waiting for launch. The UAVs acknowledge receipt of commands. The operator launches the UAVs in sequence due to the assumption that we use fixed-wing UAVs in the scenario. They fly into two predefined race courses at separate altitudes, where UAVs wait for others to join the flock. When all UAVs are in the race courses, they fly towards the search area as a flock.

Upon reaching the area, the UAVs collaboratively search for the target in the AO. The UAVs will head for the high uncertainty locations, exchange search results, update the target location probability map to be maintained on each UAV. The UAVs ensure that they do not overlap in their respective search areas to ensure the best search effectiveness. When a target is detected, the founder UAV will assign itself as a manager to assign a target monitoring (tracking) task. Task assignment will be done based on contract net protocol.

UAVs will continue searching the targets if they are not assigned to track the found targets. However, once UAV is awarded to track the targets, it will keep monitoring the targets. After search task completed, UAVs who take the search task will go back to the launch station automatically.

Finally, the operator issues the recovery command. The UAVs acknowledge receipt and return to the recovery area as a flock. Upon reaching, they go into the two race courses awaiting the operator's command to land singly. The whole scenario is depicted in Fig. 1.

The main challenging of this decentralised search and track problem is that in the whole process, after receiving the take-off command, a team of UAVs is required to accomplish the task cooperatively without help from ground control station.

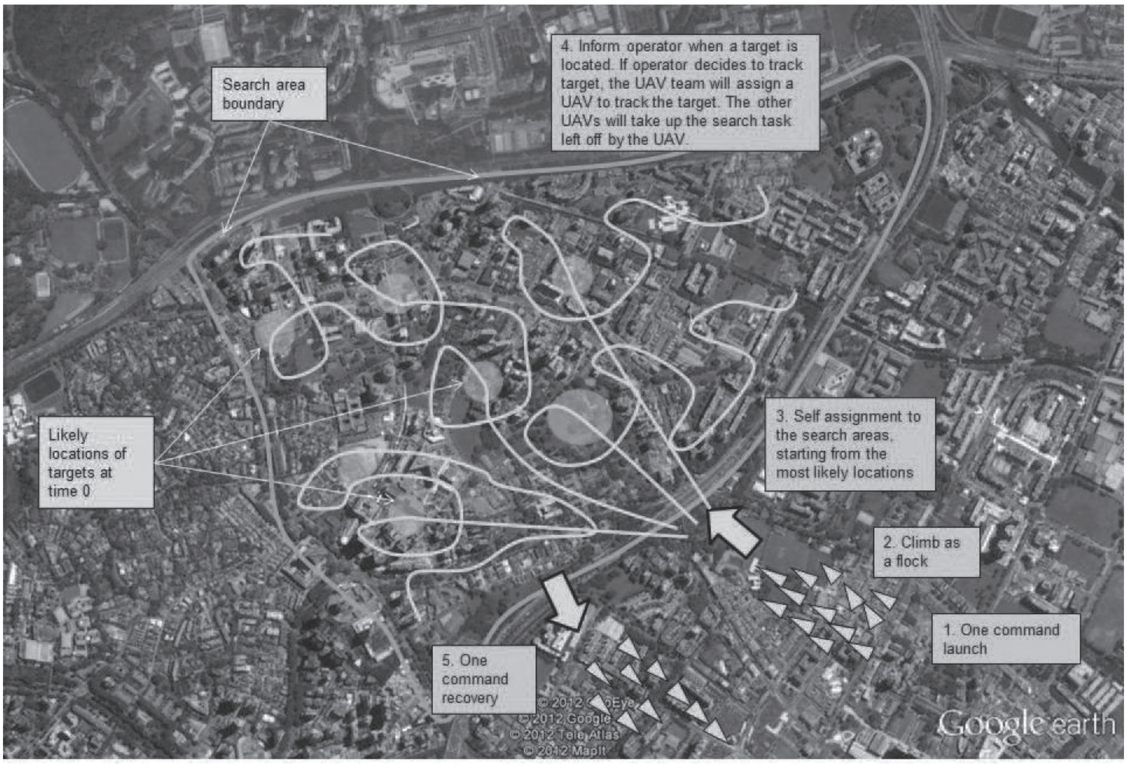


Fig. 1 *Scenario description: operations in search and track mission*

doi: 10.1049/iet-cta.2014.0469

3 Decentralised search, tasking and tracking

3.1 Control logic

In this paper, we first present the high-level control logic design which will serve as the 'brain' of the whole surveillance system. As mentioned before, the main objective of this work is to design autonomy for individual UAVs. As seen from Fig. 2*a*, each UAV contains different modules, including sensors, navigation section, logic, inner and out loop control, communication module. In this paper, we mainly focus on the logic design part.

The UAVs work in one of the four operational modes as shown in Fig. 2*b*, that is, take-off mode, fly-to-AO mode, searching mode and tracking mode. A finite-state automata (FSA) template model of UAV operations has been developed. Upon the FSA template model coordination protocols are synthesised, whose instantiations to each individual UAV serve as local supervisors that can guide each individual UAV to act properly based on other UAVs' status to ensure conformation with pre-specified global requirements.

An FSA template model has a hierarchical structure like a statechart [20], which captures UAV operations with different levels of modelling details. More explicitly, at the top level we shall see a proper sequence of predefined operations treated as macro states in the FSA model. As seen in Fig. 2*b*, four modes run sequentially. At the second level each operation (i.e. a macro state) is refined with roughly three sequential operational phases encoded in proper transitional structures: pre-condition checking, command execution and post-condition checking. Operations in each phase include executing operational functions (e.g. UAV takeoff), collecting observations (e.g. UAV at the right altitude in the air), initiating communication (or negotiation) processes (e.g. have all other UAVs taken off?), updating the UAV states and possibly taking contingent actions upon receiving abnormal observations about the UAV operations (e.g. what to do if one UAV does not respond to inter-UAV communication during the takeoff stage?). Each operation at the second level may be refined further by finer actions at a lower level, for example, the operation of initiating a communication process may require sequential actions of sending a message and waiting for a response, which can be described at the third level of the hierarchy.

As seen from Fig. 2*b*, in each mode, there are several parallel processes run concurrently which can be triggered simultaneously and terminated: (i) coordination process, which serve as a logic controller, which takes other UAVs' communication toke information, detection results and message to generate a concrete flight command; (ii) communication process, which undertakes inter-UAV sensing information exchange and other relevant message exchange; (iii) flight management process which receives flight control commands from coordination process and translate the commands into flight actuation signal for a low-level flight controller; and (iv) target search and identification process, which undertake vision-based target detection process.

In the search and track problem, each mode will be triggered by different events. These events are generated from UAVs' sensing results to the environments. For example, one of the UAVs detects one target on the ground of AO.

Communications among UAVs are realised by using heartbeat-based protocol. UAVs are required to exchange the probability map (updated in search mode) and their state (location and heading angle) with neighbours. Communication failure is also considered in our study.

In the following subsections, UAVs' operation modes will be addressed in detail including mode transfer conditions, multi-UAV decentralised cooperation strategies.

3.2 Take-off mode

The take-off operation is the phase of operation starting from when the UAVs are on the ground and ending just after they are all released to fly to the AO. This includes the launch, the climb and the flight in the holding pattern. The main objective of take-off mode is to let a team of UAVs autonomously and sequentially fly to several predefined holding heights. In the holding pattern, the fixed-wing UAVs are expected to running in circles, which is different from quadrotors. The main challenging of the take-off mode is that UAVs are required to autonomously climb to certain predefined height without colliding with others. To

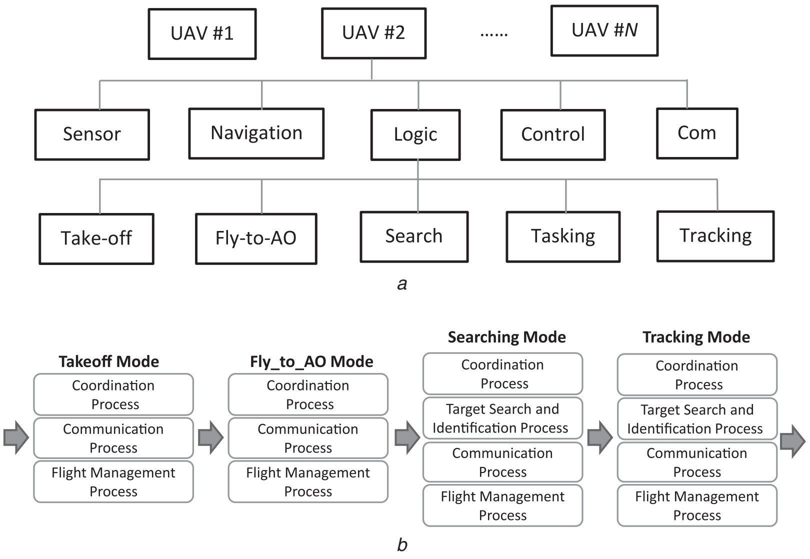


Fig. 2 *UAV control structure and logic*

a UAV control structure

b UAV operational modes

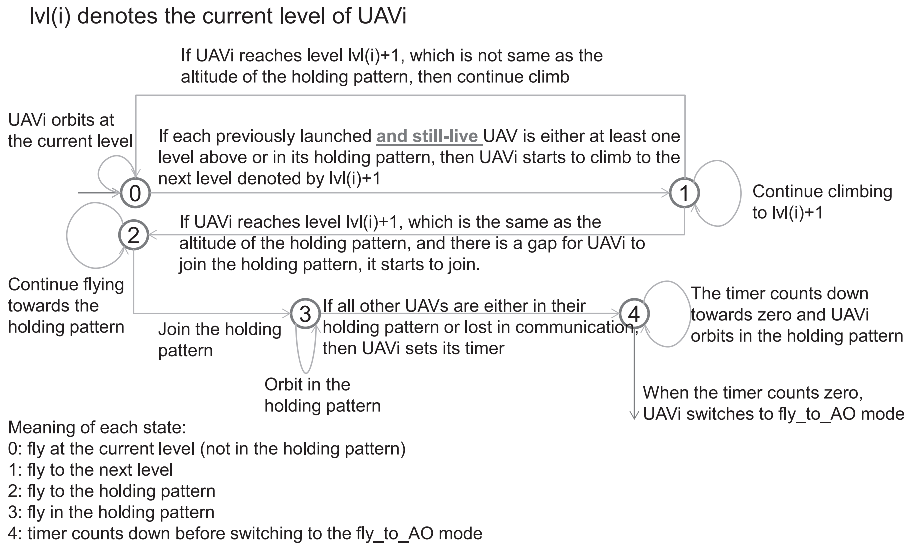


Fig. 3 *Textual explanation of coordination protocol of take-off operation*

reach this goal, UAVs need to keep communication with its neighbours even though communication failure may occur during communication process.

We assume that the takeoff sequence is determined by the operator and reflected in the UAV IDs. An FSA with text explanations is depicted in Fig. 3. In the figure, numbers in the small circles represent UAV states in the take-off mode. 'lvl' denotes the level of UAV's climbing height which is predetermined by the operator.

The coordination process describes that at state 0 UAVi either orbits at its current location [i.e. Orbit(lvl(i))] when it is in the air or stays still when it is on the ground [i.e. onGrd when $\mathrm{lvl}(i) = 0$]. Based on the received information from neighbouring UAVs, if all predecessor UAVs are either above UAVi's target intermediate orbit or have moved to their designated race course, then UAVi will start to fly to the target intermediate orbit, whose altitude is specified by $\mathrm{lvl}(i) + 1$, and progress to state 1. UAVi stays at state 1 while continuing its flight to the orbit at $\mathrm{lvl}(i) + 1$. When the target orbit is reached, UAVi will check whether this orbit is at the same altitude as that of its assigned race course. If it is not, then UAVi will return back to state 0 and continue climbing to the next orbit; otherwise, UAVi will wait for a sufficiently large gap appearing in the race course before it starts to join the race course and moves to state 2. UAVi will stay at state 2 until it successfully joins the race course, then its state is updated to state 3. UAVi will stay in the race course until all other 'live' UAVs are all in their respective race course. When this happens, UAVi moves to state 4, while it starts count down of its timer from value D. This process will allow other UAVs to have time to receive the latest status of UAVi via its heartbeat message. After the countdown is over, UAVi will fly to the AO area. Here, UAVs may leave the race course at different times due to imperfect communication.

Proposition 1: The proposed coordination protocol described in Fig. 3 will ensure (i) mutual exclusion at intermediate climbing zones; (ii) all UAVs reach their designated holding patterns eventually; and (iii) the maximum delay between the first UAV flying to AO and the last UAV flying to AO is bounded if the following condition holds:

- There exists an upper bound d_1 which is the maximum number of consecutive communication message drops that a live UAV will have. After d_1 communication drop outs, it is assumed that a live UAV will definitely communicate successfully immediately after that.

Proof: The proof of this proposition can be found in [21]. □

Based on upper bound d_1 stated in the above proposition (which is expected to be known according to field tests), we can set the timer (UAV states 3 and 4 as shown in Fig. 3) $d_2 \geq d_1$ to ensure that the countdown is long enough for UAVs to receive message from others before it flies away after countdown.

3.3 Fly-to-AO mode

After all the UAVs are in the holding pattern from the take-off mode and timers count down to zero, UAVs switch to fly-to-AO mode. In the fly-to-AO mode, UAVs automatically initiate a waypoint which is near to or on the boundary of the AO. Teams of UAVs will sequentially fly to the AO. Note that UAVs can also fly in a certain formation when they fly to the AO.

3.4 Search mode

When UAVs cross the AO boundary, they switch to search mode autonomously. In this work, we study the vision-based target search problem. Assume Q is a convex search region and AO for UAVs. Each UAV is equipped with a downward camera.

Classical target search methods, such as spiral or lawn-mower are of centralised which need an offline planning and not robust to online UAV failures. In addition, for lawn-mower type of search methods, all the UAVs line up before they start to search. However, it is more reasonable if UAVs can enter the AO from anyplace and on anytime through the boundary. The another concern is that the existing methods do not perform well in dynamic environments. Recently, developed coverage search which relies

doi: 10.1049/iet-cta.2014.0469

on locational optimisation techniques can be implemented in a distributed fashion in the sense of the Delaunay graph [3]. The locational optimisation function to serve as a measure of coverage can be written as follows

$$\mathcal{H} = \sum_{i=1}^{N} \int_{V_i} \|q - p_i\|^2 \phi(q)\, \mathrm{d}q \tag{1}$$

where N denotes the number of UAVs; V_i is the Voronoi partition of ith UAV. $q \in Q$ denotes a point or cell in the search region. $\phi(q)$ represents an a priori measure of information on Q. Each Voronoi region has mass M_i and centroid C_i, where $M_i = \int_{V_i} \phi(q)\, \mathrm{d}q$ and $C_i = 1/M_i \int_{V_i} q\phi(q)\, \mathrm{d}q$.

The problem formulation in (1) can be used for static target search in a free space. General used Lloyd-like gradient descent control law for the problem formulation in (1) is applied to omni-directional vehicles. However, fixed-wing UAVs have to follow a forward motion and also have their own dynamic constraints, such as turning rate and speed constraints. Hence a novel and efficient search algorithm needs to be developed.

In this paper, we design an improved coverage search algorithm based on our previous work [7] where decentralised probability map-based search method was proposed. Here we will mainly focus to introduce our new contribution. In this work, UAV flight dynamics and multi-UAV de-confliction have been taken into account.

3.4.1 Basic idea: In our problem setting, UAVs sequentially enter the AO and are required to find targets within a certain time constraint. We divide the AO into cells with certain sizes. Each UAV will take charge of cells that fall into its Voronoi region.

Most of the works on coverage search in the literature only consider the first-order kinematic vehicle model. However, fixed-wing UAVs have their higher-order flight dynamics. In this work, we still adopt the coverage control method to generate the waypoints at each time step for the UAVs. The only difference is that angular information of potential flight paths compared with UAV's heading angle is also considered. The basic strategy is to reformulate $\phi(q)$, which denotes the sampling importance of cell q. In our previous work, $\phi(q)$ represents target existence probability, which is denoted by $\mathrm{Prob}(q)$ in this paper for the convenience.

Now we start to formulate $\phi(q)$. As we know, the fixed-wing UAV has its minimum turn radius as shown in Fig. 4. At each time step, based on UAV's current heading angle, the potential flight path can be generated easily (see flight paths shown in Fig. 4). In terms of travelling time, it is more reasonable to guide the UAVs to fly the paths within the region bounded by two dotted lines (exclude the region falling into circular area). To consider this flight paths concern, we use α_q to represent a heading angle coefficient. Without loss of generality, for a cell q which is in ith UAV's Voronoi region, α_q can be formulated as follows

$$\alpha_q = \begin{cases} a_1 & \text{if } q \text{ is within the region bounded by} \\ & \text{the two dotted lines} \\ a_2 & \text{otherwise} \end{cases} \tag{2}$$

where a_1 and a_2 are positive scalars and $a_1 > a_2$.

Another important factor considered here is multi-UAV de-confliction. Voronoi partition can guarantee that UAVs do not overlap in their respective search area. However, it cannot guarantee that fixed-wing UAVs do not collide with each other. Hence in this work, we take into account de-confliction coefficient β_q to further disperse UAVs to avoid potential collision risks and also improve the cooperation efficiency among UAVs. Without loss of generality, for a cell q which is in ith UAV's Voronoi region, β_q is formulated as follows

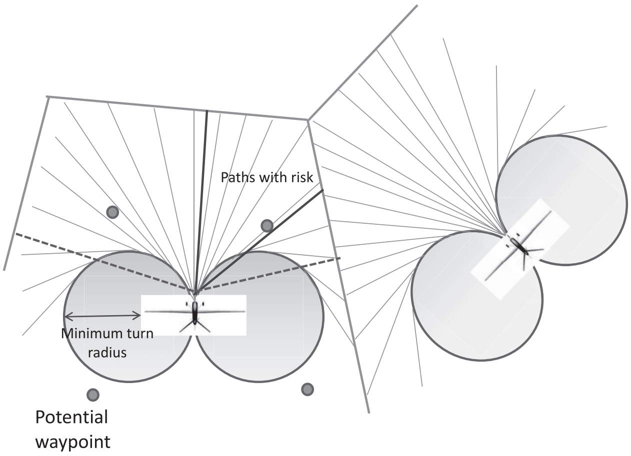


Fig. 4 *Waypoint generation for fixed-wing UAVs*

$$\beta_q = \begin{cases} b_1 & \text{if } q \text{ is within a region which contains} \\ & \text{paths with risk} \\ b_2 & \text{otherwise} \end{cases} \tag{3}$$

where b_1 and b_2 are positive scalars and $b_1 < b_2$.

In the above, we have adopted simple formulations to take into account UAV flight constraints and de-confliction. Then for a cell q which is in ith UAV's Voronoi region, $\phi(q)$ in (1) can be formulated as

$$\phi(q) = \alpha_q \times \beta_q \times \mathrm{Prob}(q) \tag{4}$$

where $p_b \geq \mathrm{Prob}(q) \geq 0$ denotes the target detection uncertainty of cell q. p_b denotes the uncertainty upper bound.

The detailed search algorithm is presented in Algorithm 1.

Remark 1: At each time step, UAVs need to share its position and probability map with its neighbours. The map fusion can use a general consensus protocol [7]. In addition, UAVs need to store the road map of AO.

3.5 Tasking

As mentioned above, task assignment in this work adopts the contract net protocol [19], which is a task-sharing protocol in multi-agent systems, consisting of a collection of nodes or software agents that form the 'contract net'. Each node on the network can, at different times or for different tasks, be a manager or a contractor.

In this work, when new target found, task assignment will be trigged autonomously. The founder UAV will assign itself as a manager. All the neighbouring UAVs who receive the ask for proposal message from the manager UAV will act as contractors. They will propose their proposals according to its current task list, distance from UAVs to targets, to calculate the revisit time for the newly found target. The winner UAV will add the new found target into its tracking list. The message format is presented in Table 1 including messages for mangers and contractors.

Algorithm 1: Cooperative target search automata for individual UAV

```
(1) Initialisation: Initialise waypoint (entry area of AO),
detection probability, false alarm;
divide AO into cells and set initial probability of each cell
to 0.5 (the largest uncertainty of target existence).
(2) Sampling step: UAV takes measurement and update its
probability map. Then sharing the probability map with its
neighbours.
If target found, then UAV first checks whether the target is
tracked by other UAVs. If not, then UAVs will assign
itself as the leader, prepare and send out announcement to
ask for task assignment proposal.
UAV (leader) itself also will propose a proposal. Task
assignment is based on contract net protocol. UAVs who
are awarded the task will switch to tracking mode.
(3) Waypoint planning: If no target found, then UAV
keeps itself in search mode.
Waypoint generation follows the Lloyd-like gradient
descent control law and will take into account of
probability map, UAV flight constraints, and also
multi-UAV de-confliction. See our formulation of $\phi(q)$
in (4).
(4) Waypoint-based path planning according to a real
fixed-wing UAV model.
Then go to step (2).
```

Table 1 Contract net protocol message format

addresseeID — senderID — CNET — TARGET — east — north — type
addresseeID — senderID — CNET — PROPOSAL — numTargets — cirTime
addresseeID — senderID — CNET — AWARD — east — north — type

The process of contract net protocol-based task assignment is presented in Fig. 5. For the contractors, they will compute circuit duration when tracking this target. If the UAV is already tracking other targets, it will compute the circuit duration when adding this new target to the targets it is already tracking. When UAVs compute the circuit duration based on minimum turning radius, they will also add travelling time to new target to circuit time. This will then penalise those that are far from the target in the case when the new target is the only target it will track.

Then the contractors will check whether circuit duration is within maximum circuit duration. Send proposal if it is, otherwise, do nothing. Then updates own target list with this new target. This way, when it comes across this same target in the future, it will not treat it as a new target and will not initiate a contract net.

The manager UAV evaluates the proposals and finds first the UAV that tracks the most number of targets within the minimal tracking time interval. If there is a tie, choose the one that tracks the targets with a shorter time interval.

Remark 2: There is one contract net per target. If two targets are found by the manager UAV, it will handle each target by one contract net consecutively.

Our proposed task assignment protocol can handle the case when the number of UAVs is less than targets'. The key point is that travelling time from UAVs to new found target is also considered into UAV' cost. This will enable one UAV to track one cluster of targets. In each cluster, targets are near to each other. In the experimental study, the proposed task assignment protocol will be evaluated.

3.6 Tracking mode

When UAVs are awarded the tracking task, they will switch to tracking mode. In the tracking mode, the optimal path

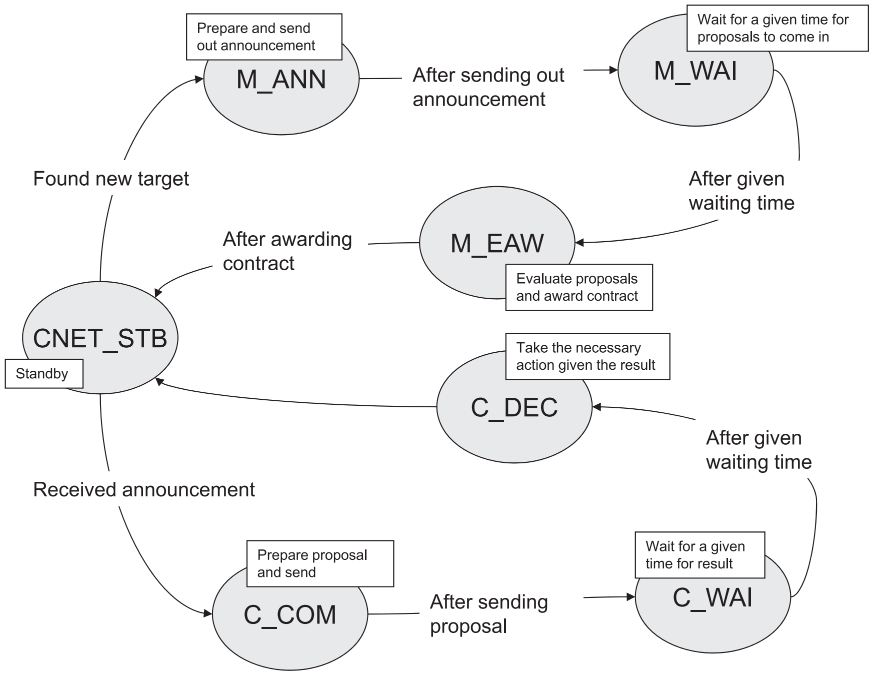
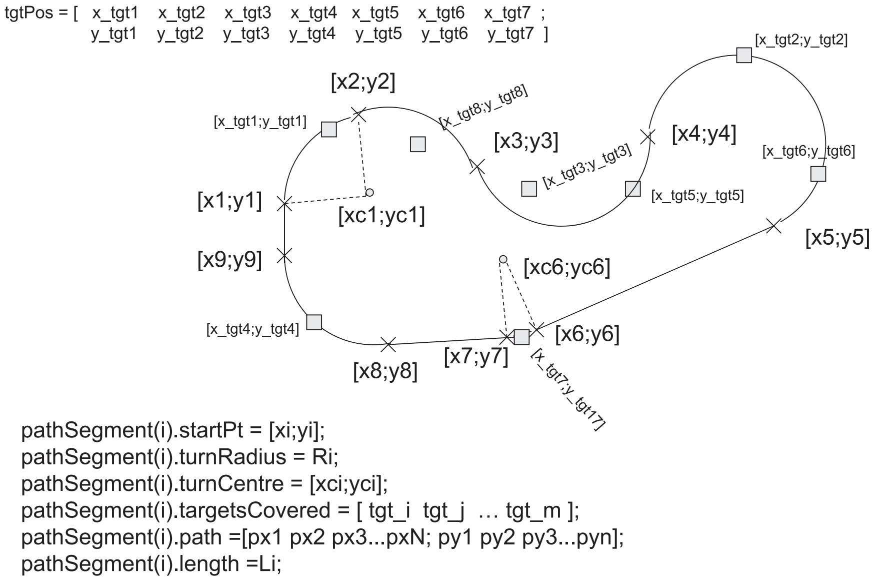


Fig. 5 *Task assignment based on contract net protocol*



```
tgtPos = [  x_tgt1   x_tgt2   x_tgt3   x_tgt4  x_tgt5   x_tgt6   x_tgt7  ;
            y_tgt1   y_tgt2   y_tgt3   y_tgt4   y_tgt5   y_tgt6   y_tgt7  ]
```

```
pathSegment(i).startPt = [xi;yi];
pathSegment(i).turnRadius = Ri;
pathSegment(i).turnCentre = [xci;yci];
pathSegment(i).targetsCovered = [ tgt_i  tgt_j  … tgt_m ];
pathSegment(i).path =[px1 px2 px3...pxN; py1 py2 py3...pyn];
pathSegment(i).length =Li;
```

Fig. 6 *Tracking path generation*

planning needs to be worked out based on the number of tracked targets, the minimal turn radius R and the field of view (FOV) of the fixed-wing UAVs. In this work, we have derived the optimal path, which is similar to Dubins path [22], for the UAVs to continuously monitor multiple static targets.

When the first target is founded by one UAV, the founder UAV switches to a tracking mode to track this target. For the consecutive founded targets, we use the following method to plan the tracking process.

Basic idea: firstly, work out the path segment based on target positions as shown in Fig. 6. Then generate the tracking paths according to Dubin path method. As shown in the figure, the rectangles represent target positions. UAV needs to monitor these targets. PathSegment shown in the figure gives the output the algorithm, that is, waypoints to travel from one target to another.

Consider M (≥ 2) targets in the group $\mathcal{W}$

$$\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$$

where $\mathbf{w}_j = (x_{\text{tgt}_j}, y_{\text{tgt}_j})$ denotes the coordinate of target $\mathbf{w}_j$. Do the following judgement sequentially according to the id of the targets. Suppose that one UAV locates just above target w_j. If there is other target $\mathbf{w}_{j_i}$ in the field of the UAV's view, delete $\mathbf{w}_{j_i}$ in $\mathcal{W}$. In fact, we do not need specially monitor target $\mathbf{w}_{j_i}$ in the track path. Repeat the process for the target with next id in new $\mathcal{W}$ till the target of the penultimate id if it still remains in group $\mathcal{W}$. For the simplicity of the math symbols, we still denote M as the number of targets.

The way to generate these waypoints is addressed in Algorithm 2.

Note that for any given two targets with certain orientation, respectively, and minimal turn radius of the UAV, the Dubins path (shortest one) is unique. For any generic Dubins path, it contains three part: an arc with radius R, a line segment and another arc with same radius.

In Fig. 6, the symbol '×' represents a certain endpoint (x_i, y_i) of one part of the closed-loop track path. An arc with turn centre (xc_j, yc_j) or a line segment connects two consecutive endpoints. Thus the track path is composed of several arcs and line segments.

Algorithm 2: Target tracking algorithm

---

(1) Find geometric centre $\mathbf{w}_0(x_0, y_0)$ of these M targets $\mathbf{w}_j$ $(j = 1, \ldots, M)$.
(2) Compute the orientation α_j from geometric centre $\mathbf{w}_0(x_0, y_0)$ to $\mathbf{w}_j$, respectively, where each $\alpha_j \in [-(\pi/2), (3\pi/2))$. Sort the targets by $\{\alpha_j\}$, and the result is chosen as the traverse order. For convenience, we still use the notations $\mathbf{w}_j$ and α_j to denote the targets' position and orientation in new order (counter-clockwise).
(3) Define traverse orientation (heading angle)

$$\beta_j = \alpha_j + \frac{\pi}{2} \quad (j = 1, \ldots, M)$$

then each $\beta_j \in [0, 2\pi)$.

Note that for the UAVs in the tracking mode, target tracking list may be dynamically changed, that is, the new founded targets can be added into the list. When one UAV's tracking list is expanded, it will redesign its optimal paths based on the number of targets, the minimal turn radius R and FOV.

4 Implementation results

4.1 Hybrid multi-UAV simulation system

In this subsection, we give a brief introduction to our developed three-dimensional (3D) hybrid multi-UAV simulation system. The main purpose of this simulator is to test UAV control logic before real flight test.

There are two main parts in the developed simulator as shown in Fig. 7*a*: 3D animation and control logic. For the 3D animation part, we use USARSim (unified system for automation and robot simulation), which was initially developed for urban search and rescue simulation [23, 24]. It is a high fidelity simulation tool based on unreal technology [25], unreal development kit (UDK). 3D urban city maps and 3D UAV model have been built and imported to UDK.

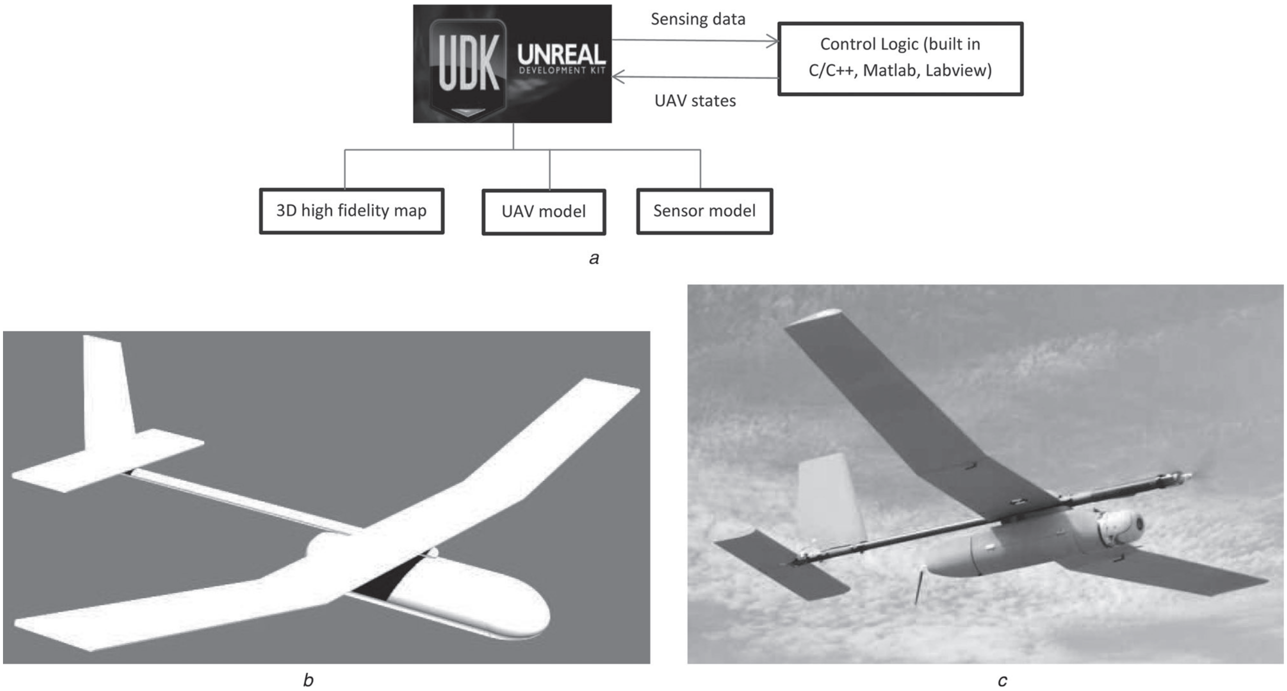


Fig. 7 *3D hybrid multi-UAV simulation system*
a 3D hybrid multi-UAV simulator structure
b 3D fixed-wing physical model
c Real UAV

Table 2 Fixed-wing UAV model

Parameter	Max/Min values
flight path angle	13°/0
bank angle	12°/−12°
air speed	16.4/12.4 m/s
acceleration (forward direction)	3.0/0 m/s^2
turn rate	7.1/0 deg/s

Table 3 Search method comparison

Method	Search time, min
lawn-mower	9
improved coverage search	13
coverage search [7]	22

In addition, each UAV will be mounted a downward camera to detect the target. Fig. 7*b* presents our developed 3D fixed-wing physical model which will be used for our tests. For comparison purpose, the real UAV is shown in Fig. 7*c*.

UAV's control, navigation, logic will be developed in control logic part as shown in Fig. 7*a*. The algorithms can be written in C and/or Matlab and so on.

4.2 3D simulation results

In the simulation and experimental study, we consider a square search region. The size of the map is around 2 km × 2 km. There are total ten static targets placed on the ground. It is assumed that targets form three clusters and in each cluster the targets are near to each other according to Euclidean distance.

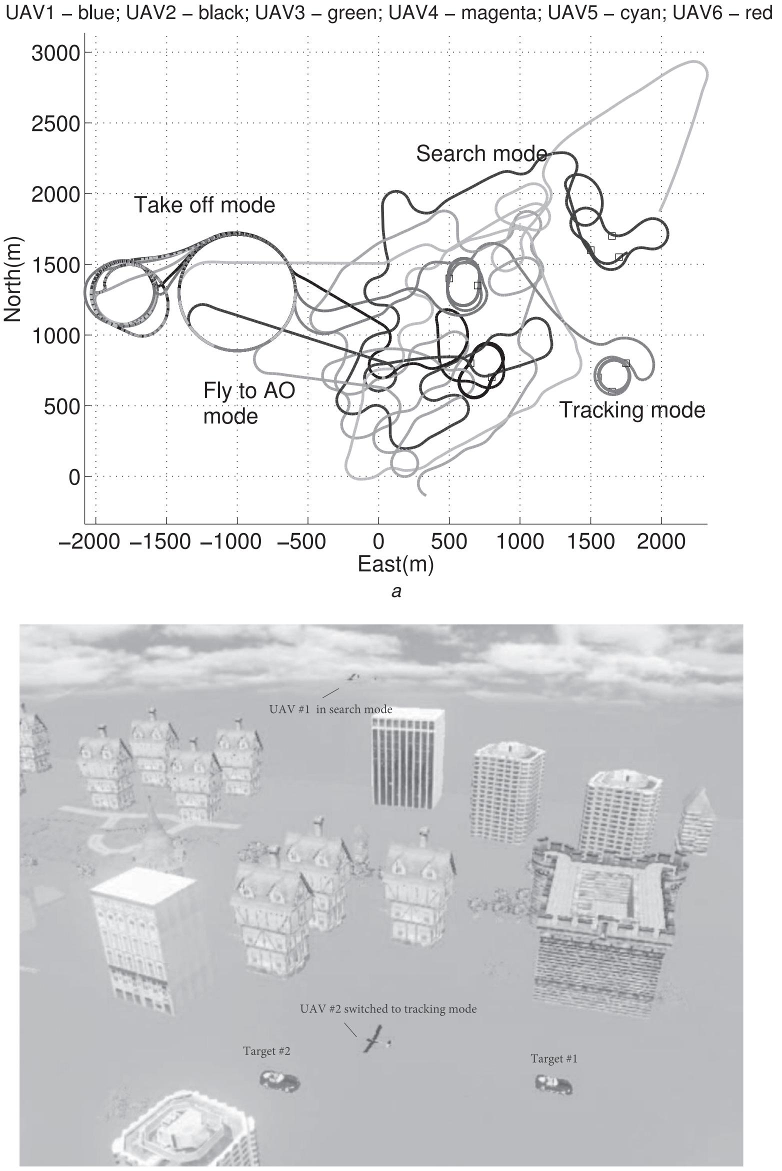
A realistic fixed-wing UAV dynamic model used for our simulation is shown in Table 2. Six UAVs are used and they are expected to find the targets and keep tracking the found targets. The speed of the UAVs is around 14 m/s. The FOV of camera mounted on the UAVs is around 80 m^2 region. The parameters in search mode are set as follows: $a_1 = 20$, $a_2 = 1$, $b_1 = 1$, $b_2 = 10$.

At first, six UAVs are all in take-off mode and then hover in two circular areas autonomously. After all UAVs hover above the ground, they fly sequentially to AO. When UAVs enter the AO, they change to search mode. In the search mode, UAVs cooperatively and autonomously search the moving targets. Once a target found which is not tracked by other UAVs, the founder UAV will initiate the task assignment based on contract net protocol. The awarded UAVs will switch to tracking mode and keep tracking the targets within a certain revisit time. All the control logic will follow the method we have addressed in Section 3.

In the simulations, the effectiveness of search algorithm has been evaluated by comparison with lawn-mower and previously developed coverage search algorithms. The comparison result is listed in Table 3. From the table, we can find that UAVs using Lawn-mower method take a minimal search time. However, it is a centralised method and depends on the offline planning. Our improved search algorithm is better than previous probability map-based coverage search algorithm.

Our proposed decentralised search, tasking and tracking method has been extensively tested using our hybrid simulator using the fixed-wing model stated in Table 2. Implementation results are shown in Fig. 8. In Fig. 8*a*, UAVs' whole surveillance procedure is presented. Four flight modes of UAVs are clearly shown in the figures, for example, take-off mode, fly-to-AO mode, search mode and tracking mode. Also from Fig. 8*a*, we can clearly see the mode transmissions.

doi: 10.1049/iet-cta.2014.0469



a

b

Fig. 8 *Implementation results*
a Mode explanation
b Snapshot of 3D view in unreal simulator

Fig. 8*b* shows a snapshot of 3D animation part in UDK. In the figure, we can see UAV1 is still in the search mode and UAV2 is on its way to monitoring two targets simultaneously.

5 Conclusion

In this paper, we have considered decentralised control of multi-UAV for autonomous take-off, search and tracking. By considering the UAV flight dynamics and multi-UAV de-confliction, an improved coverage search algorithm was designed specially for fixed-wing UAVs. To keep monitoring the founded targets, optimal path planning for multiple target tracking was proposed. Through the implementation tests, we can see that the proposed decentralised UAV control algorithm is efficient which can handle autonomous UAV search, tasking and tracking.

6 Acknowledgments

The authors would like to thank Mr Pradeep Yadav from IIT, India for implementing the algorithms. The preliminary results of this work were presented in [26].

7 References

1 Chen, Y., Lu, J., Yu, X., Hill, D.: ‘Multi-agent systems with dynamical topologies: consensus and applications’, *IEEE Circuits Syst. Mag.*, 2013, **13**, (3), 21–34
2 Chen, Y., Yu, W., Li, F., Feng, S.: ‘Synchronization of complex networks with impulsive control and disconnected topology’, *IEEE Trans. Circuits Syst. II, Express Briefs*, 2013, **60**, (5), 292–296
3 Cortes, J., Martinez, S., Karatas, T., Bullo, F.: ‘Coverage control for mobile sensing networks’, *IEEE Trans. Robot. Autom.*, **20**, (2), 243–255, 2004
4 Meng, W., He, Z., Teo, R. *et al.*: ‘Decentralized control of multi-UAVs for target search, tasking and tracking’. IFAC, 2014
5 Bethke, B.M.: ‘Persistent vision-based search and track using multiple UAVs’. Department of Aeronautics and Astronautics, Master’s

thesis, Massachusetts Institute of Technology, Cambridge MA, June 2007

6 Chen, Y., Lu, J.: ‘Finite time synchronization of complex dynamical networks’, *J. Syst. Sci. Complex*, 2009, **29**, (10), 1419–1430

7 Hu, J., Xie, L., Lum, K.Y., Xu, J.: ‘Multiagent information fusion and cooperative control in target search’, *IEEE Trans. Control Syst. Technol.*, 2013, **21**, (4), 1223–1235

8 Hu, J., Xu, J., Xie, L.: ‘Cooperative search and exploration in robotic networks’, *Unmanned Syst.*, 2013, **1**, (01), 121–142

9 Hu, J., Xu, Z.: ‘Distributed cooperative control for deployment and task allocation of unmanned aerial vehicle networks’, *IET Control Theory Appl.*, 2013, **7**, (11), 1574–1582

10 Meng, W., Xiao, W., Xie, L.: ‘A projection based fully distributed approach for source localization in wireless sensor networks’, *Ad hoc Sensor Wirel. Netw.*, 2013, **18**, (1–2), 131–158

11 Li, Z., Liu, X., Fu, M., Xie, L.: ‘Global h/spl infin/consensus of multi-agent systems with Lipschitz non-linear dynamics’, *IET Control Theory Appl.*, 2012, **6**, (13), 2041–2048

12 Furukawa, T., Bourgault, F., Lavis, B., Durrant-Whyte, H.: ‘Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets. Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006, ICRA 2006, 2006, pp. 2521–2526

13 Meng, W., Xie, L., Xiao, W.: ‘Optimality analysis of sensor-source geometries in heterogeneous sensor networks’, *IEEE Trans. Wirel. Commun.*, 2013, **12**, (4), 1958–1967

14 Rathinam, S., Almeida, P., Kim, Z. *et al.*: ‘Autonomous searching and tracking of a river using an UAV’. American Control Conf., 2007, ACC’07, 2007, pp. 359–364

15 Skoglar, P., Orguner, U., Törnqvist, D., Gustafsson, F.: ‘Road target search and tracking with gimballed vision sensor on an unmanned aerial vehicle’, *Remote Sens.*, 2012, **4**, (7), 2076–2111

16 Kwok, A., Martìnez, S.: ‘Unicycle coverage control via hybrid modeling’, *IEEE Trans. Autom. Control*, 2010, **55**, (2), 528–532

17 Luna, J., Fierro, R., Abdallah, C., Wood, J.: ‘An adaptive coverage control for deployment of nonholonomic mobile sensor networks over time-varying sensory functions’, *Asian J. Control*, 2013, **15**, (4), 988–1000

18 Kollar, T., Roy, N.: ‘Trajectory optimization using reinforcement learning for map exploration’, *Int. J. Robot. Res.*, **27**, (2), 175–196, 2008

19 Smith, R.G.: ‘The contract net protocol: high-level communication and control in a distributed problem solver’, *IEEE Trans. Comput.*, 1980, **100**, (12), 1104–1113

20 Harel, D.: ‘Statecharts: a visual formalism for complex systems’, *Sci. Comput. Program.*, 1987, **8**, (3), 231–274

21 Su, R., Teo, R., Meng, W. *et al.* ‘Finite-state automata models and software realizations of multi-UAV Operations’. Technical Report, National University of Singapore, 2013

22 Dubins, L.E.: ‘On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents’, *Am. J. Math.*, 1957, **79**, (3), pp. 497–516

23 Carpin, S., Lewis, M., Wang, J., Balakirsky, S., Scrapper, C.: ‘USARSim: a robot simulator for research and education’. IEEE Int. Conf. Robotics and Automation, 2007, pp. 1400–1405

24 Xu, J., Xie, L., Toh, T., Toh, Y.: ‘HNMSim: a 3D multi-purpose hybrid networked multi-agent simulator’. Chinese Control Conf. (CCC), 2012, pp. 5915–5920

25 Unreal technology www.unreal.com, www.udk.com, www.unrealengine.com

26 Meng, W., He, Z., Teo, R., Xie, L.: ‘Decentralized search, tasking and tracking using multiple fixed-wing miniature UAVs’. ICCA 2014 invited paper

doi: 10.1049/iet-cta.2014.0469