2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

An Almost Communication-Less Approach to Task Allocation for Multiple Unmanned Aerial Vehicles

Peng Cheng Vijay Kumar

GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104 USA
{chpeng, kumar}@grasp.upenn.edu

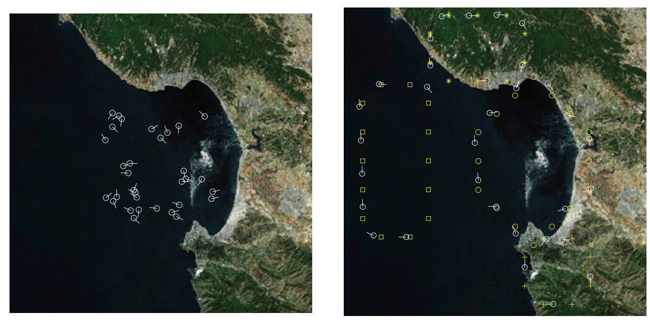
***Abstract*— In this paper, we present a scalable, decentralized task allocation algorithm for a group of unknown number of Unmanned Aerial Vehicles (UAVs), which are equipped with GPS receivers, synchronized clocks and radars with a finite, but known operating distance to identify neighbors. The algorithm assigns subgroups of UAVs, whose initial positions are randomly scattered in a bounded space, to a finite set of independent tasks. The key features of the proposed algorithm are: (1) the algorithm does not require any communication between the UAVs; (2) the task allocation is achieved in finite time. The analysis and results in the simplified 2D simulation environment respectively prove and verify the correctness of the proposed algorithm.**

I. Introduction

UAVs represent the fastest growing market segment of the robotics industry with projections of exceeding $13.5 billion by 2014 [1]. The price/performance ratio for UAVs is falling and networked groups of autonomous UAVs will soon become a reality. There is great potential for the use of such networked groups in many applications including search and rescue operations, scouting and reconnaissance missions for homeland security, and environmental mapping with three-dimensional mobile sensors. In all these applications heterogeneous unmanned vehicles must be able to search for information, localize and identify the source of information, and track the information source in a dynamic setting. One of the critical problems in applications is the *task allocation problem*. Generally, the mission or task is first decomposed into a set of subtasks, which are then allocated by a coordination algorithm to individual robots.

In this paper, we are interested in the following task allocation problem. Suppose a group of unknown number of homogeneous or heterogeneous UAVs (fixed or rotary wings) which are equipped with GPS receiver and synchronized time clocks, are at different points on a known plane at time t_0 (see Fig. 1 (a)). Suppose each UAV can sense other UAVs that are within a distance d_s but cannot communicate with other UAVs. At t_0, all UAVs are triggered (by a broadcast message[1]) to start a common task whose description consists of (a) a set of independent subtasks, *e.g.*, tracking different targets and surveilling different areas of interest (see Fig. 1 (b)); and (b) an allocation function which computes the number of UAVs required for each subtask given the total number of UAVs. The problem is to establish consensus on the total number of vehicles available and which vehicle should take which subtask in finite time.

[1]This broadcast message is the reason that we call our algorithm “almost communication-less”. This is the only required communication, which is used to initiate the proposed algorithm. After this initiation, no more communication is necessary.



(a) (b)

Fig. 1. Initial randomly scattered configurations at t_0 (a) and final target trajectories for four area surveillance subtasks (b) of a group of unknown number of UAVs

Multiple robot task allocation algorithms have been studied extensively. In the following, we will briefly review the literature from the perspective of existence of communication. When communication is available, many algorithms achieve task allocation through passing messages between robots [2] in either a centralized or decentralized way by maximizing a utility function [3], [4], [5], [6]. If communication is not available, task allocation can still be solved implicitly using *emergent* approaches [4] in the context of rendezvous [7], [8], consensus problems [9], [10], or formation control [11], [12], [13], [14], [15]. Control of robots is based on just local information from near neighbors. In the rendezvous problems [7], [8], a group of vehicles need to converge to an unknown point. In consensus problems [10], a group of vehicles need to agree on a specific value through local sensing. In formation control problem, vehicles are controlled according to neighboring information to achieve a given formation. If only local information (no GPS, no common map, trajectories, or targets) is available to each agent, the existence of a solution is guaranteed only under very restrictive assumptions [8], [9], [10] on the connectivity or visibility graph. When some global information, *e.g.*, common map or trajectories, is available, many results exist [16], [12], [13], [14], [15], [7], which use either gradient- or geometric-based methods to achieve the various objectives. However, these methods only guarantee asymptotic convergence and/or are not readily applicable for UAVs, which are subject to nonholonomic kinematic constraints and whose forward velocity must be larger than a positive lower bound (fixed wing UAVs cannot hover at one point or reverse directions).

We propose a scalable, decentralized algorithm for the task allocation problem considered in the paper. The key idea is to use a local sensing based control law and time synchronization to distribute all vehicles at regular intervals along a common curve in finite time such that consensus can be established on the total number of vehicles and allocation of subtasks to vehicles. The proposed algorithm 1) does not need any communication between neighboring vehicles for the designed control law, 2) converges in finite time, and 3) is applicable for both fixed and rotary wing UAVs which might have a large positive lower bound on the forward velocity.

In the following, we will first formulate the problem mathematically in Section II. The proposed algorithm for a group of homogeneous vehicles is presented in Section III. Sections IV and V respectively provide the results in a simplified 2D simulation environment and conclusion.

II. Problem Formulation

The task allocation problem considered in this paper can include either homogeneous or heterogeneous vehicles. For conciseness, we will first limit ourselves to the case with homogeneous vehicles. Extension to the heterogeneous case will be discussed later.

A. The UAV model

The dynamics of the UAV is simplified as the Dubins car [17]: $\dot{x} = v_f \cos\theta$, $\dot{y} = v_f \sin\theta$, and $\dot{\theta} = w$, in which $v_f \in [v_f^{\min}, v_f^{max}]$ with $v_f^{\max} = k_v v_f^{\min} > 0$ and $k_v > 1$ is the input determining the forward velocity, and $w \in [-w^{\max}, w^{\max}]$ is the input determining the yaw angle turning rate. Note that we model the UAV with a positive lower bound on the forward velocity such that the proposed algorithm will be applicable for the fixed wing UAVs.

B. The sensor models

Each UAV is equipped with two sensors. One is a GPS signal sensor, from which the UAV will know its position and orientation in a common inertial frame. The other one is an omnidirectional sensor, *e.g.*, the radar, which has a limited sensing distance d_s. Once the distance between two UAVs is less than $d_s > 0$ as shown in Fig. 2, both vehicles will know the relative position to each other.

C. Task description

A task description, denoted as T, consists of m subtasks, denoted as $\{T_i\}$, a bounded mission area $S \subset \Re^2$, and a task allocation function $\mathcal{L}$:

$$\begin{aligned} \mathcal{L}: \quad N \times N \quad &\to N \cup \{0\} \\ n \times k \quad &\to l_k, \end{aligned} \tag{1}$$

which shows that $l_k \leq n \in N \cup \{0\}$ vehicles will be assigned to subtask $k \leq m \in N$, where the total number of vehicles is $n \in N$. If $\sum_{k=1}^{m} l_k < n$, some UAVs will not be assigned any subtask (or equivalently assigned the null subtask). If $\sum_{k=1}^{m} l_k > n$, the subtasks must be prioritized and lower priority subtasks may not be assigned to any vehicle.

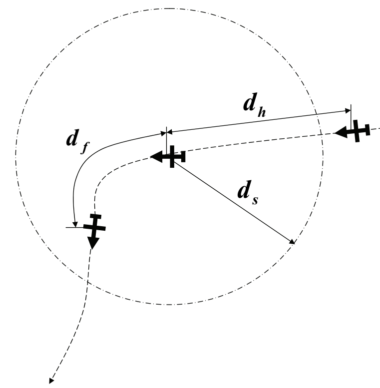


Fig. 2. The limited sensing distance of the UAV

D. Objective

Given an unknown number $n > 0$ of homogeneous UAVs starting with a common task description T from arbitrary initial configurations $q_i(t_0) \in S$ at time t_0, the objective is to design decentralized control laws for these UAVs such that there exists a constant $t_f > t_0$ after which each vehicle will be assigned to a specific subtask and the number of vehicles assigned to subtask T_k is $n_k = \mathcal{L}(n, k)$.

III. Almost Communication-Less Task Allocation (ATA) Algorithm

We will first describe the outline of the proposed ATA algorithm, and then provide details.

A. Outline of the algorithm

Step 0: Initialize k as 0.

Step 1: At time t_k, all vehicles compute the largest closed admissible curve Q_k with total length L_k in the mission area S. By the "admissible curve", we mean the UAVs are able to track exactly the curve with any forward velocity $v_f \in [v_f^{\min}, v_f^{\max}]$ in the nominal condition. Because all vehicles are homogeneous, they compute the same closed curve (see Fig. 6 a). The reason to choose the largest curve is to reduce the effect of uncertainties in the task allocation based on the curve length, which will be explained with details in Section III-C.

Step 2: At time t_k (assuming the computation time in Step 1 is negligible), all vehicles start to fly to and track the common closed admissible curve L_k with constant forward velocity v_f in the same direction (clockwise or counter clockwise) (see Fig. 6 b). Because the mission area S is bounded, there exists ΔT_t (a positive constant to be determined in Section III-B), after which all vehicles will track the closed curve.

Step 3: At time $t_k + \Delta T_t$, each vehicle will adjust its inputs according to the distance to its adjacent neighbors such that after ΔT_k (a positive constant to be computed in Section III-C) all vehicles will achieve equal distance between each other along the closed curve (see Fig. 7 a) if there are enough vehicles for them to see each other. We call this the *equal distance condition*. At time $t_k + \Delta T_t + \Delta T_k$, each vehicle will compute the total number n of vehicles on the closed curve according to the total curve length L_k and distance d_n between adjacent vehicles (see Fig. 3). According to the task allocation function $\mathcal{L}$, each vehicle will know how many vehicles for each subtask. Finally, based on the current location on the closed curve, vehicle will decide which subtask it should choose at this moment (see details in Section III-C).

Step 4: Before each vehicle flies to its assigned subtask, it will check whether the equal distance condition was established (see details in Section III-D). If the equal distance condition is achieved, then each vehicle will fly to its assigned subtask; otherwise, all vehicles compute the common closed curve Q_{k+1} with total length $L_{k+1} = L_k - d_s$.

Step 5: Set $k \leftarrow k+1$ and t_k be the time after the equal distance condition checking is done, and go back to Step 2.

Steps 0, 1 and 5 are straight forward, we will only explain the other three steps in the following sections.

B. Step 2: fly to and track the common closed admissible curve in finite time

Objective: Given a common closed admissible curve Q_k and n vehicles starting from random configurations $q_i(t_0) \in S$ at time t_0, the objective of this step is to control these vehicles such that after constant ΔT_t time all vehicles will track the closed curve in the same direction with a constant forward velocity. We ignore the collision avoidance here assuming that there is a collision avoidance strategy, *e.g.*, the vehicles will fly on the different altitudes when potential collision might happen.

Algorithm: The following decentralized control law is used for each vehicle:

1) Select a set of way points on the closed curves. Because the vehicle is able to track the curve in the same direction, a unique configuration can be determined for each way point.
2) Choose the way point whose corresponding configuration is closest to the current configuration of the vehicle according to the shortest path between two configurations for the Dubins car.
3) Follow the shortest path with constant forward velocity to the way point on the closed curve. If the vehicle reaches the curve earlier than ΔT_t, then it will track the curve.

Because the initial configurations of all vehicles are assumed to be in the bounded mission area S, the following lemma will ensure that there exists such ΔT_t, after which all vehicles track the common closed curve in the same direction.

Lemma 1: For a bounded space $S \subset [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$, there is an upper bound $\Delta T_t = \frac{4\pi}{w_{\max}} + \frac{\sqrt{(x_{\max}-x_{\min})^2+(y_{\max}-y_{\min})^2}}{v_f}$ on the time for the Dubins car with constant forward velocity v_f and turning rate in $\{-w_{\max}, 0, w_{\max}\}$ to move between any two configurations in S.

Proof: The minimum turning radius of the Dubins car is $\frac{v_f}{w_{\max}}$. According to [17], the shortest path between two any configurations in S is less than $4\pi v_f / w_{\max} + \sqrt{(x_{\max}-x_{\min})^2 + (y_{\max}-y_{\min})^2}$, and therefore we have the bound on the duration of the flight between any two configurations. ■

C. Step 3: achieve the equal distance condition on a closed curve in finite time and assign subtasks based on the curve length

Objective: Let Q_k be the current admissible closed planar curve with total length L_k. Assume that n vehicles fly along this curve with same direction at random positions on the curve at $t_k + \Delta T_t$. We assume that no two vehicles are at the same position. Let function $\mathcal{D} : \Re^2 \times \Re^2 \rightarrow \Re^+ \cup \{0\}$ return the length of the shorter curve segment between two points on the closed curve. Let $p_i(t)$ denote the location of vehicle i at time t. The objective is to control these vehicles such that after constant duration ΔT_k we have

$$\mathcal{D}(p_i(t), p_k(t)) = \frac{L_Q}{n}, \forall t > \Delta T_k + t_k + \Delta T_t \quad (2)$$

for all adjacent vehicles i and k.

Algorithm: Let d_f and d_h be the curve length from the current vehicle respectively to the vehicle in the front and behind as shown in Fig. 2. If there is no vehicle in the front or behind, then d_f or d_h is set to be infinity. The following control law is proposed.

1) Compute distance d_f and d_h for the current vehicle.
2) If $d_f > d_h$, then the forward velocity of the vehicle is set to be $k_{\text{up}} v_f \leq v_f^{\max}$ for some constant $k_{\text{up}} > 1$; otherwise, the forward velocity is set to be $v_f \geq v_f^{\min}$.
3) After ΔT_k, if a vehicle does not have equal distance d_n to its front and behind neighbors, then it just keeps tracking the closed curve; otherwise, it will make tentative decision about which subtask to take using the following procedure:
   a) The vehicle computes the total number of vehicle as $n = \frac{L_k}{d_n}$.
   b) The vehicle computes how many vehicles are needed for a specific subtask with the given task allocation function $\mathcal{L}$. Assume that subtask T_i needs n_i vehicles.
   c) The vehicle computes the curve length l_c from the current position to the right most intersection of the closed curve with the positive x axis of an inertial frame with origin at the the center of the closed curve (see Fig. 3).
   d) For some $i \in \{1, 2, \cdots, m\}$, if $\sum_{j=1}^{i-1} \frac{n_i}{n} L_k \leq l_c \leq \sum_{j=1}^{i} \frac{n_i}{n} L_k$, then this vehicle is assigned to subtask T_i.

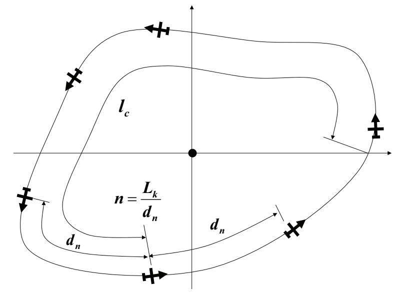


Fig. 3. Tentative task allocation based on the curve length

The following two lemmas will show the correctness of the proposed algorithms in this section. The first lemma shows that there exists constant ΔT_k after which the equal distance condition will be achieved if it is possible. Note that there exist other methods, *e.g.*, [12], [15], to achieve the equal distance condition along a curve. However, these methods normally can only converge to the equal distance condition when time goes to infinity and/or require that the vehicles are

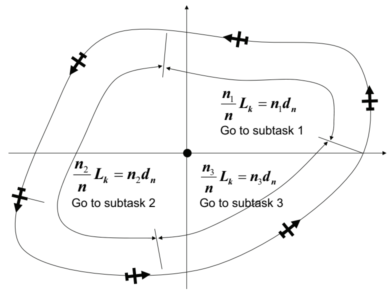


Fig. 4. The equal distance condition ensures the correct task allocation based on the curve length

either considered as holonomic points or are able to move in both forward and backward directions. The second lemma shows that the subtasks will be allocated correctly if the equal distance condition is achieved.

Lemma 2: If $L_k \leq d_s n$, then the vehicles achieve the equal distance condition after executing the above control law for duration $\Delta T_k = \frac{L_k}{(k_{\text{up}}-1)v_f}$.

Proof: Let $\Delta d_{k,j} = \mathcal{D}(p_k, p_j) - \frac{L_k}{n}$,

$$\Delta d^+_{k,j} = \begin{cases} \Delta d_{k,j}, & \Delta d_{k,j} > 0 \\ 0, & \Delta d_{k,j} \leq 0 \end{cases}, \quad (3)$$

and

$$\Delta d^-_{k,j} = \begin{cases} 0, & \Delta d_{k,j} > 0 \\ \Delta d_{k,j}, & \Delta d_{k,j} \leq 0 \end{cases}. \quad (4)$$

It is easy to see that $\sum_{i,j} \Delta d^+_{i,j} + \sum_{i,j} \Delta d^-_{i,j} = 0$. When these vehicles are in the equal distance condition, then $\sum_{i,j} \Delta d^+_{i,j} = \sum_{i,j} \Delta d^-_{i,j} = 0$.

First, the control law will decrease $\sum_{i,j} \Delta d^+_{i,j}$ monotonically with at least rate $(k_{\text{up}} - 1)v_f$ because at any moment, there always exists a vehicle which moves with velocity $k_{\text{up}}v_f$ to decrease $\sum_{i,j} \Delta d^+_{i,j}$ if $L_k \leq d_s n$ (see Lemma 5 in appendix). Secondly, $\sum_{i,j} \Delta d^+_{i,j}$ will approach its maximal value $\frac{n-1}{n}L_k$ when all the vehicles are very close to each other (see Lemma 6 in appendix). Therefore, after $\Delta T_k = \frac{L_k}{(k_{\text{up}}-1)v_f} \geq \frac{(n-1)L_k}{n(k_{\text{up}}-1)v_f}$ all vehicles will be in the equal distance condition. ■

Lemma 3: If the equal distance condition is achieved, the proposed curve length based allocation procedure makes the correct task allocation.

Proof: It is easy to see that all vehicles assigned to subtask T_i span a total curve length of $\frac{n_i}{n}L_k = n_i d_n$, which includes n_i vehicles as required by the task allocation function $\mathcal{L}$, if all vehicles are equally distributed as illustrated with three subtasks in Fig. 4. ■

Note that the task allocation is achieved through the curve length. If L_k is small and there are many vehicles on the closed curve, then the difference of the curve length between adjacent vehicles will be small. If the curve distance fluctuation due to uncertainties during the flight is larger than the small equal distance, then task allocation might not be correctly completed. Therefore, it is better to start with the largest possible closed curve to achieve the equal distance condition as stated in Step 1.

D. Step 4: check whether the equal distance condition has been established

Objective: We have shown that Step 3 will achieve correct task allocation if all vehicles achieve the equal distance condition. However, if the equal distance condition is not achieved, the correct task allocation is not guaranteed. In this step, we will show how to detect whether the equal distance condition was established and how to control vehicles if vehicles were not in the equal distance condition.

Algorithm:

1) At time $t_k + \Delta T_t + \Delta T_k$, the vehicle computes a closed curve Q'_k by "growing" the curve Q_k by $0 < \rho \leq d_s$ shown in Fig. 5. Note that Q'_k can be mathematically described as the boundary of the set obtained by taking the Minkowski sum of Q_k with a circle of diameter ρ. The smallest Euclidean distance from any point on one curve to the other curve is always less than d_s such that the vehicle on one curve is able to sense the vehicle on the other curve. Assume that the larger curve has total length L'_k.
2) If a vehicle already decides which subtask to choose in Step 3, then the vehicle flies to and track the larger closed curve Q'_k with velocity v_f, however, in opposite direction to the curve Q_k. Otherwise, the vehicle keeps tracking the curve Q_k. When the vehicle flies to the curve Q'_k, it initializes a counter with value 0. Whenever it senses that a vehicle flies along the curve Q_k through the closest point to the curve Q'_k, the counter will increase by 1.
3) After $\frac{L'_k + 4\pi r_w + d_s}{v_f}$ in which $r_w = \frac{v_f}{w_{\max}}$ is the minimum turning radius, a vehicle checks whether the equal distance condition has been established as follows:
   a) If the vehicle is still on the curve Q_k, then it will know that the equal distance condition has not been established.
   b) If the vehicle is on the curve Q'_k, then it will know that the equal distance condition has been established only if it did not sense any vehicle on the curve Q_k, i.e., its counter is still at 0, (see Fig. 5 b); otherwise, the equal distance condition was not established (see Fig. 5 a).

The following lemma shows the correctness of the proposed equal distance condition checking algorithm.

Lemma 4: The proposed algorithm will establish a consensus on whether the equal distance condition has been established or not after $\frac{L'_k + 4\pi r_w + d_s}{v_f}$.

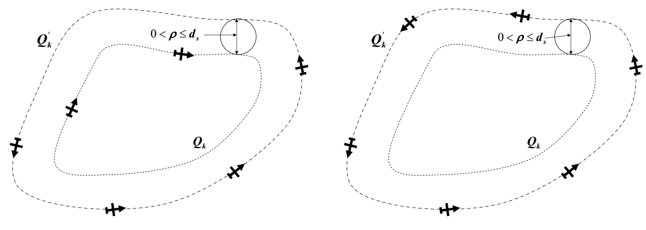
Proof: If the equal distance condition has been established in Step 3, then all vehicles will fly to the curve Q'_k and sense nothing on the curve Q_k after $\frac{L'_k + 4\pi r_w + d_s}{v_f}$ as shown in Fig. 5 (b). Otherwise, the vehicle will know that because it either is still at the curve Q_k or senses vehicles on the curve Q_k from the curve Q'_k as shown in Fig. 5 (a). ■

E. Convergence of the proposed algorithm

The following theorem shows that the proposed algorithm will always converge when there are enough vehicles.

Theorem 1: When the number of UAVs in the bounded region is at least

$$n_{\min} = \lceil \frac{\pi r_w}{d_s} \rceil, \lceil \cdot \rceil \text{ denotes the ceiling function} \quad (5)$$



(a) (b)

Fig. 5. Illustration of the equal distance condition checking

the proposed algorithm will make correct task allocation in finite time.

Proof: First, it is shown Lemma 3 that if the equal distance condition is established, then the correct task allocation will be achieved. Second, the total length of the closed curve for achieving the equal distance condition decreases by d_s in each iteration. Because there are at least $n_{\min} = \lceil \frac{\pi r_w}{d_s} \rceil > 0$ UAVs and the space is bounded, within $\lceil \frac{L_0}{d_s} \rceil$ iterations, there will exist a closed curve such that the equal distance condition is achieved and the subtasks are correctly allocated. Finally, each iteration will take at most $\Delta T_t + \Delta T_0 + \frac{L_0' + 4\pi r_w + d_s}{v_f}$ time for each iteration. Therefore, the algorithm will converge in

$$\lceil \frac{L_0}{d_s} \rceil (\Delta T_t + \Delta T_0 + \frac{L_0' + 4\pi r_w + d_s}{v_f}). \tag{6}$$

■

F. Scalable computational time analysis

In this section, we will analyze the computation time for each step of the proposed algorithm and show that the algorithm is scalable with respect to the number of vehicles, n.

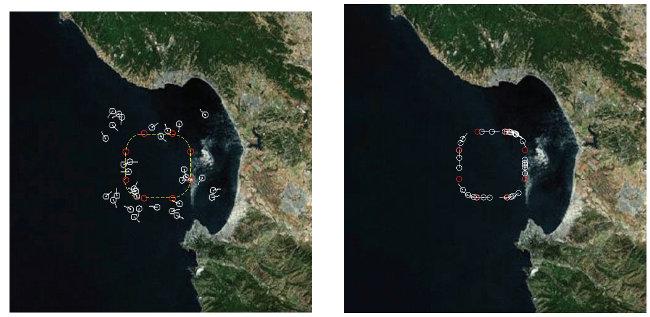
In Step 1, each vehicle just needs to compute the closed curve according to the common bounded space information. This computation is independent of n. In Step 2, the computation of the shortest path for the Dubins car is also independent of n. In Steps 3, the computation is only related to the curve length to the neighboring vehicle along the closed curve, and therefore, is independent of n. In Step 4, the computation is scalable because at most one constant time (independent of n) operation is necessary at each moment.

G. Extension to a group of heterogeneous vehicles

Assume that we have a group of heterogeneous vehicles. We can easily setup a protocol in the broadcasted task description such that different types of vehicles will fly at different altitudes such that the above proposed algorithm is applicable for each group of the same type of vehicles.

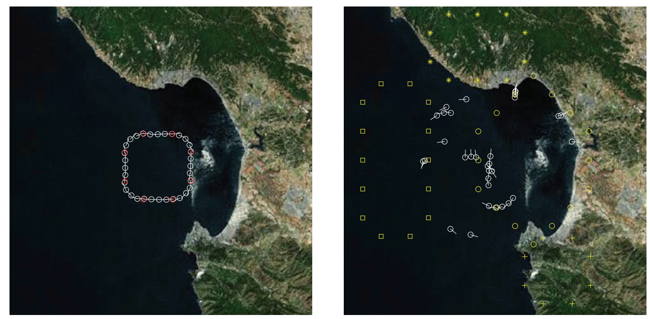
IV. Simulation Results

We use the proposed algorithm to solve the task allocation problem in a surveillance scenario. As shown in Fig. 1 (a), there is a group of unknown number of UAVs scattered around the Monterey Bay area. At time t_0, a broadcast information is sent to these vehicles to fly to and track four trajectories respectively to provide surveillance along the coast line and the entrance to the bay area (Fig. 1 (b)). Assume that the total number of the vehicle is n, and a task allocation function that assigns $\frac{3}{8}n$ vehicles to the middle right closed curve, $\frac{1}{4}n$ vehicles to both the top and middle left curves, and $\frac{1}{8}n$ vehicles to the bottom curve. Clearly, the task allocation function must be designed to ensure that n_k is a natural number.



(a) (b)

Fig. 6. The snapshots of the executing the proposed algorithms



(a) (b)

Fig. 7. The snapshots of the executing the proposed algorithms

The snapshots of the execution of the proposed algorithms are shown in Figs. 6 and 7. Figure 6 (a) shows the common closed curve (the dashed line) and its way points, and Fig. 6 (b) shows that all vehicles track the common closed curve. Figure 7 (a) shows that the equal distance condition has been established on the closed curve, and Fig. 7 (b) shows that subtasks are allocated and vehicles fly to their final target trajectories. Figure 1 (b) shows the final results of the task allocation, in which the top curve and middle left both have 8 vehicles, the bottom one has 4, and the middle right has 12 vehicles. This allocation of the vehicles satisfies the requirement in the given task description.

V. Conclusion

In this paper, we propose an almost communication-less decentralized algorithm to achieve task allocation for a group of unknown number of vehicles, which are initially scattered randomly in a bounded planar space, just through local sensing and time synchronization. The proposed algorithm converges in finite time, and is scalable with respect to the total number of vehicles. It is applicable to UAVs, especially for small-scale fixed wing UAVs which are subject to nonholonomic constraints and have a positive lower bound on the forward velocity.

Appendix

Lemma 5: The equal distance control law in Section III-C will cause $\Delta D = \sum_{i,j} \Delta d_{i,j}^{+}$ to decrease monotonically

	behind	front
Case 1	v_f	$k_{\text{up}} v_f$
Case 2	$k_{\text{up}} v_f$	v_f
Case 3	v_f	v_f
Case 4	$k_{\text{up}} v_f$	$k_{\text{up}} v_f$

TABLE I

THE SPEED OF ADJACENT VEHICLES OF A VEHICLE

	d_f	d_h
α	$> d_n$	$> d_n$
β	$> d_n$	$\leq d_n$
γ	$\leq d_n$	$> d_n$
ξ	$\leq d_n$	$\leq d_n$

TABLE II

THE DISTANCE OF ADJACENT VEHICLES TO A VEHICLE

with at least rate $(k_{\text{up}} - 1)v_f$ if $d_s > \frac{L_k}{n}$.

Proof: We will first study the effect of each vehicle, and then the effect of all vehicles on ΔD.

1) The effect of a single vehicle on ΔD The speed of vehicles behind and in the front of a vehicle can be either of four types in Table I.

The distance to vehicles in the front and behind a vehicle can be either of the four types in Table II.

If a vehicle has velocity $k_{\text{up}} v_f$, then it will change ΔD with the rate in Table III.

If a vehicle has velocity v_f, then it will change ΔD with the rate in Table IV.

2) The effect of all vehicles on ΔD

If we list the distance to adjacent vehicles for each vehicle along the closed curve according to the types in Table II, there are only two patterns before the equal distance condition is established: 1) $\beta\{\alpha\}^*\gamma$ with speed pattern $k_{\text{up}}\{k_{\text{up}}, v_f\}^* v_f$ and 2) $\gamma\{\xi\}^*\beta$ with speed pattern $v_f\{k_{\text{up}}, v_f\}^* k_{\text{up}}$, in which k_{up} denotes speed $k_{\text{up}} v_f$ for

(d_f, d_h)	Case 1	Case 2	Case 3	Case 4
α	$+v_d$	$-v_d$	0	0
β	0	$-v_d$	$-v_d$	0
γ	$-$	$-$	$-$	$-$
ξ	0	0	0	0

TABLE III

THE EFFECT OF A VEHICLE WITH VELOCITY $k_{\text{up}} v_f$ ON ΔD, IN WHICH $v_d = (k_{\text{up}} - 1)v_f$, AND "$-$" MEANS THAT THIS SITUATION IS IMPOSSIBLE.

(d_f, d_h)	Case 1	Case 2	Case 3	Case 4
$(> d_s, > d_s)$	$-$	$-v_d$	0	0
α	$+v_d$	$-v_d$	0	0
β	$-$	$-$	$-$	$-$
γ	0	$-v_d$	0	$-v_d$
ξ	0	0	0	0

TABLE IV

THE EFFECT OF A VEHICLE WITH VELOCITY v_f ON ΔD, IN WHICH $d_s > d_n = \frac{L_k}{n}$

simple notation and A^* means to generate a sequence with a non negative integer length using elements from set A.

It can be checked with Tables III and IV, the total effect of middle symbols, ξ^* and α^*, will not change ΔD, and the total effect of symbols β and γ will decrease ΔD with at least $(k_{\text{up}} - 1)v_f$. ■

Lemma 6: For a closed curve with length L_k and n vehicles, $\sum_{i,j} \Delta d^+_{i,j}$ will have maximal value $\frac{n-1}{n} L_k$.

Proof: According to (3),

$$\begin{aligned} \sum_{i,j} \Delta d^+_{i,j} &= \sum_{m,n} \left(\mathcal{D}(p_m, p_n) - \frac{L_k}{n}\right) \\ &= \sum_{m,n} \mathcal{D}(p_m, p_n) - \sum_{m,n} \frac{L_k}{n} \\ &\leq L_k - \frac{L_k}{n}, \end{aligned} \quad (7)$$

for all adjacent vehicles m, n whose $\mathcal{D}(p_m, p_n)$ is larger than $\frac{L_k}{n}$ and the equality is true when all vehicles are at the same location. ■

REFERENCES

[1] F. International, "The market for uav reconnaissance systems," http://www.forecastinternational.com/press/release.cfm?article=80, 2006.

[2] L. Parker, "Alliance: An architecture for faulttolerant multi-robot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 220–240, 1998.

[3] M. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, pp. 1257–1270, 2006.

[4] B. Gerkey and M. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 23, pp. 939–954, 2004.

[5] P. Modi, W. Shen, M. Tambe, and M. Yokoo, "Adopt: Asynchronous distributed constraint optimization with quality guarantees," *Artificial Intelligence Journal*, vol. 161, pp. 149–180, 2005.

[6] M. Zavlanos and G. Pappas, "Dynamic assignment in distributed motion planning with limited information," in *Proceedings of the 2007 American Control Conference*, 2007.

[7] J. Cortés, S. Martinez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, pp. 1289–1298, 2006.

[8] Z. Lin, B. Francis, and M. Maggiore, "Getting mobile antonomous robots to rendezvous," in *Workshop on control of uncertain systems: modelling, approximation, and design*, Apr. 2006.

[9] A. Jadbabaie, J. Lin, and A. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[10] W. Ren, R. Beard, and E. Atkins, "A survey of consensus problems in multi-agent coordination," in *American control conference*, 2005.

[11] E. W. Justh and P. S. Krishnaprasad, "Equilibria and steering laws for planar formations," *Systems & control letters*, vol. 52, pp. 25–38, 2004.

[12] Y. Cao and R. Fierro, "Dynamic boundary tracking using dynamic sensor nets," in *Proceedings of IEEE Conference on Decision & Control*, 2006.

[13] M. Hsieh, S. Loizou, and V. Kumar, "Stabilization of multiple robots on stable orbits via local sensing," in *Proceedings IEEE International Conference on Robotics & Automation*, 2007.

[14] M. Hsieh and V. Kumar, "Pattern generation with multiple robots," in *Proceedings IEEE International Conference on Robotics & Automation*, 2006.

[15] F. Zhang and N. Leonard, "Coordinated patterns of unit speed particles on a closed curve," *Systems and Control Letters*, 2006, submitted.

[16] E. W. Justh and P. S. Krishnaprasad, "Steering laws and continuum models for planar formations," in *Proceedings of IEEE conference on decision and control*, 2003.

[17] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.