Switched UAV-UGV Cooperation Scheme for Target Detection

Herbert G. Tanner

Abstract— We develop a switched cooperative control scheme, to coordinate groups of ground and aerial vehicles for the purpose of locating a moving target in a given area. We do so by stabilizing the ground group into a guarding formation using a navigation function, and then steering the aerial group along a trajectory that uniformly scans the enclosed regions. The novelty of the approach lays in combining decentralized flocking algorithms with navigation functions for obstacle avoidance, convergence to designated position, and direction control.

I. INTRODUCTION

A. Motivation

In the not-so-distant future, intelligence, surveillance and reconnaissance (ISR) missions will be performed by small, agile, autonomous robotic units. Recent operational developments regarding US deployed troops, reinforce the trend to avoid exposing humans to danger when operating in hostile environments. Such ISR missions are envisioned to involve a variety of robotic assets, such as unmanned ground vehicles (UGVs) and combat unmanned aerial vehicles (UAVs), closely cooperating to obtain situational awareness, not only for the remote operator, but for the robotic team performing the task. To successfully complete such a cooperative, multi-agent ISR mission, we need novel methods to network and control the robotic vehicles.

This work is a step in this direction. Previous efforts [24] aimed at coordinating UGV and UAV teams, in continuous cooperative motion that enables reconnaissance along a strip of hostile territory and simultaneous area coverage. In this paper we employ a *switched* cooperative control architecture, to solve a different problem: locate a possibly moving target within an given area. Our ultimate goal is to develop a methodology that can scale nicely with the size of the vehicle groups.

B. Working Assumptions

Ground assets are assumed for modeling purposes to be omni-directional vehicles, kinematically equivalent to a single integrator. If x_i denotes the planar coordinates of UGV i, then $\dot{x}_i = u_i$, where u_i is the control input to UGV i. Stacking the coordinate vectors of all N_g UGVs, we obtain

$$\dot{x} = u, \quad x, u \in \mathbb{R}^{2N_g}.$$

Herbert Tanner is with the Department of Mechanical Engineering, The University of New Mexico, Albuquerque, NM 87131, USA. Email: tanner@unm.edu

The UGVs are assumed to carry four motion detection sensors (could be cameras), with limited range, arranged in a configuration as shown in Figure 1.

Fig. 1. The role envisioned for the UGVs. Their four motion detection sensors can pivot around a vertical axis, so that the sensors can point independently of the vehicle's orientation. When appropriately configured, they can create a "virtual fence," capable of detection of motion to and from an area of interest.

These motion detection sensors are assumed mounted on a rotating base, in a way that the orientation of the sensors can be set independently of that of the vehicle's.

Aerial assets are identical, and are N_a in number. The coordinate vector of UAV i is denoted $y_i \in \mathbb{R}^2$, with the vertical coordinate ignored after assuming that all UAVs are assumed to fly at the same altitude (this ensures consistency in their sensor footprints). have double integrator dynamics,

$$\ddot{y} = w, \quad y, w \in \mathbb{R}^{2N_a}.$$

C. Organization and Overview

Our approach to solving this problem builds on earlier work. In [26] we combined decentralized flocking and steering using navigation functions, by superimposing control terms. Although decentralized, that approach lacked an element of global coordination, since all agents had their own navigation objective. Here, we *integrate* a navigation function inside the flocking algorithm, obtaining significantly more control over the final shape (and position, if necessary) of the group. We extend the navigation function development of [28], relaxing the assumption of point-robots; now the vehicles are assumed to be surrounded by a visco-elastic "shell," that enables them to keep reasonably large distances (Figure 5), without compromising convergence (Figure 6). By merging formation control by means of navigation functions with provably convergent, decentralized velocity synchronization, we are also able to achieve absolute shape control, and to adjust overall motion direction of the flock at will (Figure 7). Our approach here is mainly centralized, however all but the combinatorial part of the solution can also be implemented in a decentralized way. As of this moment, we can not find an efficient way to solve the linear assignment problem without imposing significant computational and communication overhead [1].

In section II we review briefly some related literature to put our approach in perspective. Section III describes

our technical approach in some detail, while numerical simulation results are presented in section IV. Some concluding remarks and future direction highlights are included in section V.

II. PREVIOUS WORK

In the field of robotic search and exploration, approaches differ depending on the *a priori* information available for the environment. When the boundary of the environment can be characterized, a robot can follow a variety of pre-specified paths (patterns) to cover the entire space [6], [32], even under uncertainty regarding the position of obstacles and obstructions [10]. When the environment boundaries are not known, exploring the area in minimum time is known to be an NP-complete problem, even for environments with graph structure. One of the most sophisticated approaches to robot exploration is that of [17], where a *single* robot decides the new search directions by weighting the information gain against the cost of moving along each particular direction.

Multi-robot systems have advantages in terms of speeding the search task, which have long been recognized [5], [21]. In [21], multiple robots explore the workspace, which is decomposed into cells, while minimizing their position errors. In the multi-agent "frontier-based" exploration strategy of [30], [31], robots in an effort to expand the map of the known environment. In the approach of [5] a probabilistic occupancy map is iteratively built with the robots deciding a new position by weighting the cost of reaching it against its utility. This method, seemingly along the lines of [17], it is actually closer to [31] in the sense that it uses "frontier cells as new candidate robot positions.

Our problem is not so much an exploration problem, but rather a pursuit-evasion game, that requires coordination between different teams of autonomous systems. Kim et al. [29] combine pursuit evasion games with map building by a team of multiple UAVs and UGVs, and implemented sub-optimal solutions, without excessive computational overhead. The approach here is different, because of the need to focus on algorithmic completeness and energy requirements (quantified by distance traveled). In addition, we seek a solution that can be effectively be implemented in a decentralized fashion, in the spirit of recent nearest-neighbor cooperative control schemes [27].

Cooperative control for multi-agent systems has rapidly evolved in the last few years, from reactive approaches [2], [15], to centralized control architectures [3], [18], [25], then to distributed approaches [8], [9], [13], [14], [19], and finally to biologically inspired decentralized coordination algorithms [7], [11], [16], [20], [26]. Li et al. [12] study the whole process of how teams of robots can fall into formation, during which process robots select their role within a formation by solving an assignment problem, using the Hungarian algorithm.

In our our approach to realizing the scenario outlined in section I, role assignment is part of the overall solution, and we adopt the same computationally efficient method to associate UGVs to desired grid points. Our robotic groups are diverse, and thus we assign roles both at the team and at the group level. The task of the UGV team is to partition the area and guard the partitions, while the UAV team swiftly scans each partition to detect the target. A first technical challenge is to control and coordinate each team toward its objective. A second is to synchronize the two teams to ensure completeness.

III. APPROACH

To coordinate the robots within the same team we build on our previous work on cooperative formation control [26], [28], using distributed synchronization algorithms and navigation functions. Synchronization between teams is achieved through controller switching, which is state dependent and is triggered by conditions that assess the level at which a particular task is completed. The overall coordination scheme can be expressed as a hybrid system, although the discrete dynamics are trivial.

A. Workspace Partitioning

The size of the area that can be searched is determined by the sensing capabilities and the number of the available ground and aerial assets. We *partition* the space into cells so that (i) ground vehicles can detect every transition of the target from one cell to another, and (ii) aerial vehicles have a collective sensor footprint that enables them to sweep the cell with one elementary maneuver.

The size of the cell depends on the number of available aerial vehicles. Setting the lenght of the strip formed by the UAVs flying in close formation equal to the length of the cell edge, enables us to scan one such cell with one UAV pass.

The number of cells that can be searched with confidence depends on the number of available UGVs. Positioning the UGVs along the edges of the cells, so that a target crossing over an edge can be detected by at least one UGV, creates a set of grid points, $\{x_d(i)\}$, $i = 1, \ldots, N_g$, each one representing a desired position for a UGV guarding a crossing (Figure 1). For a (square) area of $k \times k$ cells, where the length of the cell edge can be covered by the sensing range of a UGVs, the number of UGVs (grid points) required is $2ak(k + 1)$. Once the UGVs position themselves on the nodes of this grid (Figure 3), movement between cells can be detected and thus the position of the target within the area can be immediately determined.

B. UGV Grid-Point Assignment

Each ground vehicle is assigned to a specific grid point, by means of a graph-matching algorithm that associates a UGV with a grid point in such a way so that the total distance between the initial positions of the UGV and their associated grid points is minimized. The method used for matching UGVs to grid points is the *Hungarian* algorithm. Execution of the matching algorithm needs to be done centrally, hoewever the algorithm terminates in polynomial time ($O(N^3)$).

C. UGV Motion Planning

The ground vehicles navigate to their designated grid points using a (centrally generated) artificial potential field. In earlier work [28] we introduced a new navigation function that is appropriate for multi-robot formation control and navigation. In [28] we made the restrictive assumption that robots can be represented by points. In this paper we lift this assumption, and we allow for disk (or sphere) - like objects; thus, regardless of the actual shape of the vehicle, a spherical "shell" [23] can be assumed to surround it, consisting of a protected zone, that is to remain collision free, and an external sensing zone, in which collision avoidance is initiated. It must be noted that under this framework stationary obstacles can also be considered.

The navigation function is constructed according to [28], and, with appropriate tuning, guarantees that UGVs will reach their desired grid locations, from every initial condition, while avoiding collisions between them. The navigation function that coordinates the UGVs has the form

$$\varphi_g(x) = \frac{\sum_{i=1}^{N_g}(x - x_d)^2}{\exp(\beta(x)^{\frac{1}{k_g}})}, \tag{1}$$

where k_g is the function's tuning parameter, and

$$\beta(x) \triangleq \prod_{(i,j) \in N_g \times N_g, \ i \neq j} (\beta_{ij}(x) - b_0), \tag{2a}$$

is the product of all collision proximity (obstacle) functions defined as

$$\beta_{ij}(x_i, x_j) \triangleq 1 - \frac{h(\varepsilon^2 - \|x_i - x_j\|^2)}{h(\varepsilon^2 - \|x_i - x_j\|^2) + h(\|x_i - x_j\|^2)} \tag{2b}$$

$$b_0 \triangleq 1 - \frac{h(\varepsilon^2 - R^2)}{h(\varepsilon^2 - R^2) + h(R^2)},$$

Parameter R is the radius of the "shell," and ε determines the detection range of the UGV proximity sensors used for collision avoidance. Function h is given as

$$h(r) \triangleq \begin{cases} 0, & r \leq 0 \\ \exp(-1/r^2), & r > 0. \end{cases} \tag{2c}$$

Construction (2) is adapted from [4], where it is identified as a way to construct a smooth (C^∞) function, taking values in $[0, 1]$, and being identically equal to 0, and 1, in disjoint two intervals F and D, respectively. It is exactly this property (namely that β can be made identically one for large $\|x_i - x_j\|$) that allows us to compute $\prod_{(i,j) \in N_g \times N_g, \ i \neq j} \beta_{ij}$ in (1) using only the (i, j) pairs of UGVs in close proximity to each other; namely (i, j) such that $\|x_i - x_j\| < \varepsilon$. Moreover, this property will enable us to decentralize navigation in the future according to [28].

Proposition 1: Function (1) is a navigation function.

Proof: A detailed proof would track the steps of the proof in [28]. Here we demonstrate that all of the intermediate results leading to the proof in [28], hold:

1) the configuration where $\|x - x_d\|$ is a nondegenerate critical point of φ_g; it can be verified that as in [28] that $\nabla^2 \varphi_x(x_d) = \frac{2}{\exp(\beta(x_d)^{1/k_g})} I$.

2) there are no critical points on the configurations where protected zones touch, since when $\|x_i - x_j\| \to R$, $\beta \to 0$, and $\nabla \varphi_g \to -\frac{1}{e^{\beta^{1/k_g}}} \frac{\sum_{i=1}^{N_g}(x - x_d)^2}{k_g} \beta^{\frac{1}{k_g}-1} \nabla \beta$ [28]. With $\nabla \beta_{ij}(R)$ being equal to

$$\frac{-4 \, e^{R^{-4} + (R^2 - \epsilon^2)^{-2}} \epsilon^2 \left(3 R^4 - 3 R^2 \epsilon^2 + \epsilon^4\right)}{\left(e^{R^{-4}} + e^{(R^2 - \epsilon^2)^{-2}}\right)^2 R^5 \left(R^2 - \epsilon^2\right)^3} > 0,$$

we have that $\lim_{\|x_i - x_j\| \to R} \nabla \varphi_g \neq 0$.

(iii) there are no critical points near the destination or "shell"-collision configurations, because with β_{ij} being smooth, $\|\nabla \beta\|$ is always bounded, and one can obtain a lower bound for k in the same way as in [28].

3) $\nabla^2 \beta_{ij}$ evaluated at R has a single root for R in $[0, \epsilon]$, and is strictly positive for R smaller than this root. Therefore, given R, (the boundary of the protected zone), one can find an ϵ (a required sensing zone), and a unit vector $\hat{v}$, such that $\hat{v}^T \nabla^2 \varphi_g \hat{v} < 0$ [28]. For example, if one sets $R = \frac{\epsilon}{2}$, then $\nabla^2 \beta_{ij} = $

$$\frac{-1024 \, e^{\frac{128}{9 \, \epsilon^4}} \left(12544 + 891 \, \epsilon^4 + e^{\frac{128}{9 \, \epsilon^4}} \left(-12544 + 891 \, \epsilon^4\right)\right)}{729 \left(1 + e^{\frac{128}{9 \, \epsilon^4}}\right)^3 \epsilon^{10}}, \text{ which}$$

is positive and strictly increasing with ϵ.

4) the degeneracy of critical points in the free space can be established as in [28], independently of β_{ij}. ∎

Therefore (1) defines a navigation function on $\mathbb{R}^{N_g}$, and setting $u_i = -\nabla_{x_i} \varphi_g$ establishes uniform asymptotic stability of x to x_d.

D. UAV Formation Control

The aerial vehicles are steered into a formation that scans uniformly the cells in the grid defined in section III-A along one direction. Formation control is combined with velocity synchronization, so that not only do the UAV group steers itself into a specific shape, but it also moves in unison along a desired direction. To achieve this goal, we adapt the flocking algorithm of [27], by replacing the inter-agent potential function with a multi-agent navigation function of the same form as the one constructed in section III-C.

The shape of the desired formation can be specified in terms of desired relative vectors between UAVs, c_{ij}, with $(i, j) \in N_a \times N_a$. If a desired relative position vector is specified between UAV i and UAV j, we write $i \sim j$. Then the goal configuration for the UAV formation is described by the zero level set of the function

$$\gamma(y) = \sum_{i \sim j} \|y_i - y_j - c_{ij}\|^2 .$$

Collision proximity is encoded using (2), and a navigation function $\varphi_a(y)$ is constructed as

$$\varphi_a(y) = \frac{\gamma(y)}{\exp(\beta(y)^{1/k_a})},$$

where k_a is the associated tuning parameter.

Rimon and Koditschek [22], point out that in a mechanical system

$$M(p)\ddot{p} + f(p, \dot{p}) + g(p) = \tau,$$

with a torque/force input of the form

$$\tau(p, \dot{p}) = -\nabla V(p) + d(p, \dot{p}),$$

where d is a dissipative vector field and $V(p)$ is a navigation function, the critical qualitative behavior of the navigation function's gradient is copied to the mechanical system's trajectories, making it behave as $\dot{p} = -\nabla V(p)$. Our formation stabilization and flocking controller has this exact structure:

$$w = \underbrace{-\nabla_y \varphi_a + \frac{1}{2}\left(\mathbf{1}_{N_a} \otimes \ell(y,t) - \dot{y}\right)}_{\text{guiding vector field}} \underbrace{-(L \otimes I_2)\dot{y}}_{\text{dissipative field}},$$

where L is the Laplacian of the graph defined by the relative position vector specifications, c_{ij} [27], and $\ell(y,t)$ is a formation control signal, that serves to center and steer the UAV formation along a particular direction. Vector field $-(L \otimes I_2)\dot{y}$ is dissipative because L is positive semidefinite. The "guiding vector field" consists of the negated gradient of a multi-agent navigation function, $\nabla_y\varphi_a$, which steers the UAVs into formation, plus a vector with projection along the longitudinal and latitudinal components of $\dot{y}$ is parallel to $\mathbf{1}$. This latter vector drives all the UAVs, to converge uniformly to the the centerline of the set of cells that are to be scanned at time t; the term $\ell(y,t)$ essentially forces synchronization between the UAV and UGV groups, and is defined in section III-E.

E. UAV-UGV Coordination

Aerial and ground vehicle groups are coordinated using state-dependent switches. The aerial group initiates ingress and formation maneuvering only after the UGVs are in position. Such a waiting period is necessary, because otherwise the target can move undetected from a cell not yet scanned by UGV to a cell already covered.

The "go" signal to the UAV team is generated by the UGV team once the latter finds itself within a small neighborhood of their designated grid points. The switching signal is generated by monitoring the value of the centralized UGV navigation function. Then, the UAVs start maneuvering, coming into the desired formation and aligning their speeds with the centerline of the first row (or column) of the grid that needs to be scanned. When the row (or column) has been scanned, and the UAVs are at a certain distance from the grid and they are heading away from it, a transition in the controller is triggered, changing the equilibrium velocity set and aligning it with the next row (or column) (the case of row scanning can be dealt similarly).

The controller switching is implemented through $\ell(y,t)$, which is infact a piecewise continuous switching signal, and has the form

$$\ell(y,t) \equiv \ell_k(y) = \begin{bmatrix} -\frac{1}{N_a}\sum_{i=1}^{N_a} y - c_k \\ s_k \end{bmatrix} \quad k = 1, 2, \ldots,$$

where c_k is a constant which determines the position of the (current) column centerline, and $s_k \in \{1, -1\}$ sets the direction of scanning (up- or downward). If, for example, UAVs approach the grid from the south (bottom of Figure 7), s_k, can be set as $s_k = (-1)^{k-1}$. In that case, and assuming

that the grid is centered at the origin, the condition for the transition from $\ell_k(y)$ to $\ell_{k+1}(y)$ is

$$C_k = \frac{(-1)^{k-1}\sum_{i=1}^{N_a}(0,1) \cdot y}{N_a} \\ - (-1)^{k-1}(\max\{(0,1) \cdot x_d\} + \varepsilon) > 0,$$

where $\varepsilon > 0$ is a positive parameter that is included to allow the UAVs enough space to maneuver and change their direction before sweeping the next column. The switching scheme is depicted in Figure 2.
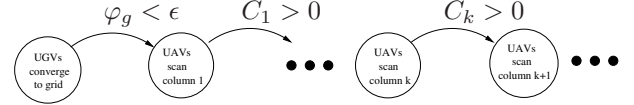

Fig. 2. Controller mode transition in the UAV-UGV team. The UAV group scans the first column only after the UGVs are in position ($\varphi_g < \epsilon$, where $\epsilon > 0$ is the desired accuracy). After scanning column k, UAVs switch their controller to scan column $k + 1$ as soon as $C_k > 0$ is satisfied.

IV. SIMULATION RESULTS

The proposed coordination scheme was tested in simulation, where an area of 3 km^2 needs to be searched by a combined UGV-UAV team, consisted of $N_g = 38$ ground vehicles and $N_a = 3$ aerial vehicles. In this scenario, the UGVs are assumed to have four cameras that enable them to detect the target as it crosses their field of view within a distance of 300 m. Their maximum speed is estimated at 40 km/h. We partition the area into four columns, of 375 m width, which is the width of the combined sensor footprints of the three UAVs once they come into a line formation. Along the edges of these columns, we define grid points separated by the UGV target detection range of 300 m. Since, however, column width is larger than 300 m, we introduce two horizontal rows of grid points, to bound the area from north and south (Figure 3).

Ground vehicles are configured to initiate collision avoidance maneuvers if other vehicles are within a 200 m range (the sensing zone). This is encoded in (2) through a suitable choice of ε. The radius of the protected zone for the UGVs is set at 100 m. Starting from random initial positions within a 4.5 km radius around the center of the search region, they navigate toward their designated grid points following the gradient of (1); a snapshot of one intermediate configuration is shown in Figure 4.

The collision avoidance properties of the navigation function are demonstrated in Figure 5, where the minimum distance between any two UGVs is shown versus time. The final configuration (Figure 6), is typically achieved in less than 250 simulation seconds.

The UGVs end their maneuver when the value of φ_g falls below 10^{-3}, at which point they are positioned as shown in Figure 6. When this occurs, the UAV team starts moving in from the south. The UAVs are initialized with random initial positions (in the $(\pm 3, -1.5 \div -7.5)$ km range) and velocities (in the $(\pm 80, \pm 80)$ km/h range), south of the UGV
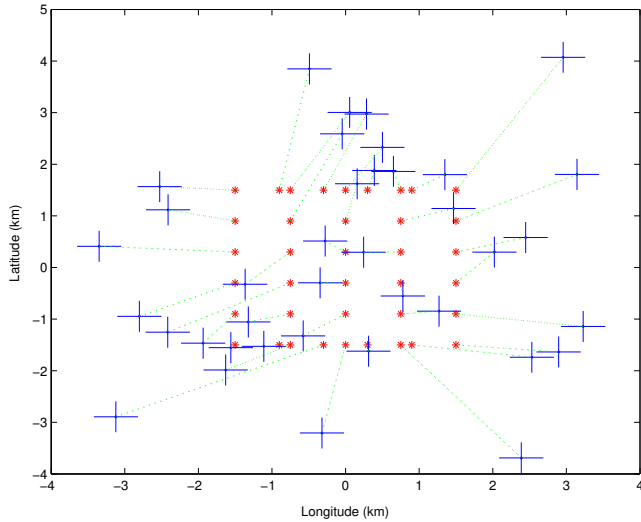
Fig. 3. The initial positions of UGVs (·) and the desired grid points (∗) marking the boundaries of the search area. The cross-marks (+) over the UGV positions mark the directions and range of their target detection systems. The faint (green) dotted lines connecting UGVs with grid points represent goal assignments for each UGV as output of the Hungarian algorithm.
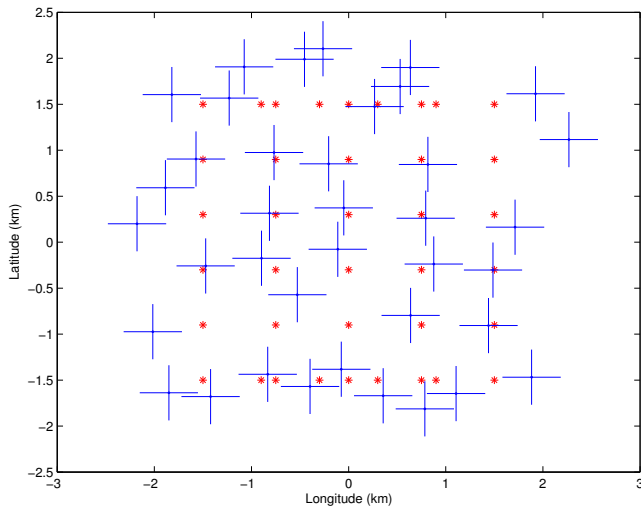

Fig. 4. The UGVs (marked by +) on their way to their designated grid points (marked by ∗). The figure shows an intermediate configuration of the UGV group.

grid formation. The motion paths of the UAVs as they scan the area column by column are shown in Figure 7. Each turn of direction designates a controller switch, and the overall search maneuver is typically completed in approximately two (simulation) minutes.

V. CONCLUSIONS

In this paper we addressed the problem of mobile target detection using a switched cooperative strategy that employed a group of ground vehicles, and a group of aerial vehicles, with delineated roles. We introduced improvements in existing cooperative control algorithms, in terms of collision avoidance and navigation, and demonstrated the efficacy
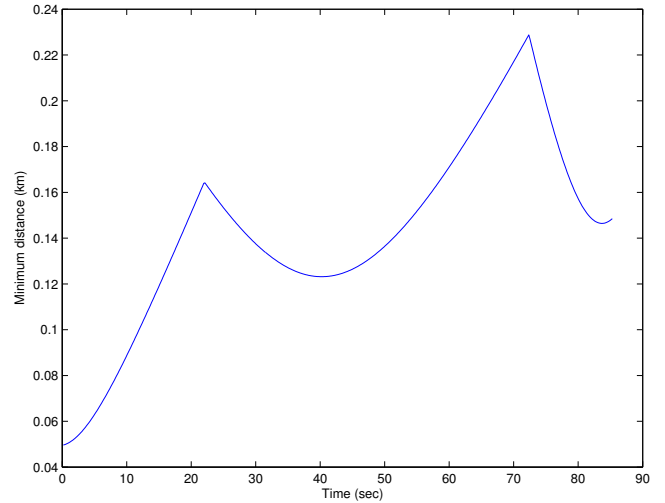

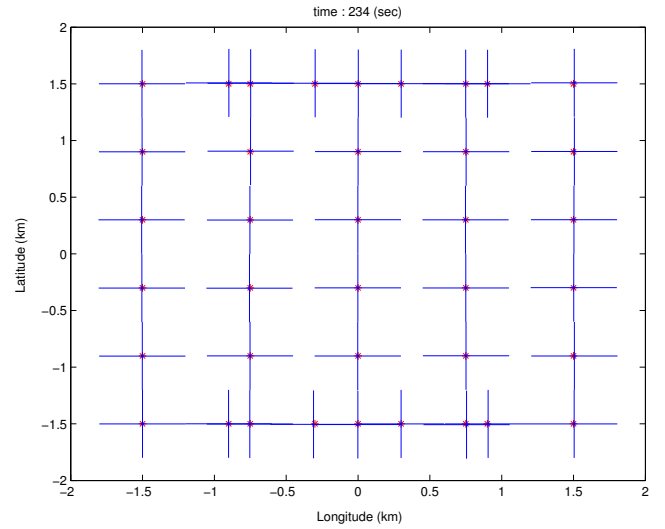Fig. 5. The evolution of the minimum distance between UGVs over time.

Fig. 6. The UGVs in their final configuration, stabilized around their designated goal points. Note how the lines marking their target detection ranges mark the boundaries of the search area, as well as the boundaries of each column that will be subsequently scanned by the UAV team.

of the approach in numerical simulations. We developed the control architecture with an eye on decentralization; currently, the components of our control strategy that are centralized are goal position assignment, and navigation for the UGV team.

REFERENCES

[1] Mehdi Alighanbari and Jonathan P. How. Decentralized task assignment for unmanned aerial vehicles. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 5668–5673, Seville, Spain, 2005.

[2] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926 – 939, 1998.

[3] R. W. Beard, J. Lawton, and F. Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9:777–790, 2001.

[4] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 2nd edition, 1986.
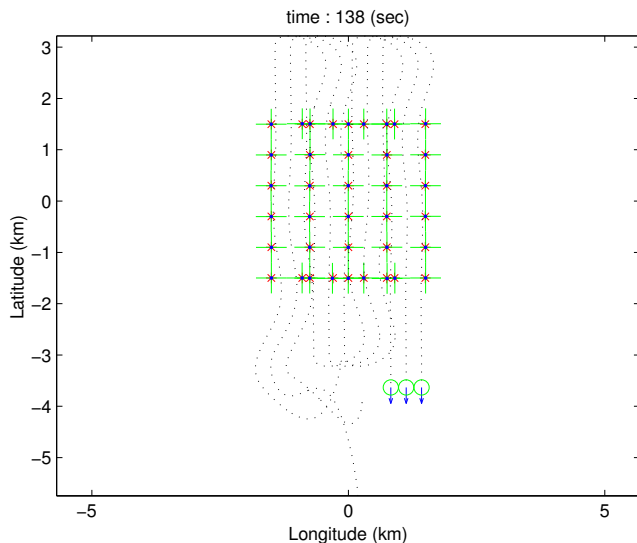
time : 138 (sec)

Fig. 7. Paths of the UAVs as they scan the search region, column by column. They approach from bottom-left corner of the figure and depart the scene from the bottom-right corner. As soon as they complete scanning one column, they reverse direction to scan the next.

[5] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1340–1345, San Francisco, 2000.

[6] Howie Choset and Philippe Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Proceedings of the International Conference on Field and Service Robotics*, Camberra Australia, 1997.

[7] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[8] R. DAndrea and G. E. Dullerud. Distributed control of spatially interconnected systems. *IEEE Transactions on Automatic Control*, 48(9):1478– 1495, 2003.

[9] V. Gazi and K.M. Passino. Stability analysis of swarms. *IEEE Transactions on Automatic Control*, 48(4):692–696, 2003.

[10] Hector H. Gonzalez-Banos and Jean-Claude Latombe. Navigation strategies for exploring indoor environments. *Journal of Robotic Systems*, 21(10-11):829–848, 2002.

[11] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2002.

[12] M. Ji, S. Azuma, and M. Egerstedt. Role-assignment in multi-agent coordination. *International Journal of Assistive Robotics and Mechatronics*, 7(1):32–40, 2006.

[13] E. Justh and P. Krishnaprasad. A simple control law for uav formation flying. Technical Report 2002-38, Institute for Systems Research, University of Maryland, 2002.

[14] N. E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the IEEE International Conference on Decision and Control*, pages 2968–2973, 2001.

[15] Damian M. Lyons and Ronald C. Arkin. Towards performance guarantees for emergent behavior. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4153–4158, 2004.

[16] J.A. Marshall, M.E Broucke, and B.A. Francis. Formations of vehicles in cyclic pursuit. *IEEE Transactions on Automatic Control*, 49(11):1963– 1974, 2004.

[17] Stewart John Moorehead. Autonomous surface exploration for mobile robots. Technical Report CMU-RI-TR-01-30, The Robotics Institute, Carnegie Melon University, 2001.

[18] Petter Ogren, Magnus Egerstedt, and Xiaoming Hu. A control lyapunov function approach to multiagent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, 2002.

[19] R. Olfati-Saber and R. M. Murray. Distributed structural stabilization and tracking for formations of dynamic multi-agents. In *Proceedings of the IEEE Conference on Decision and Control*, pages 209–215, Las Vegas, NV, 2002.

[20] R. Olfati-Saber and R.M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520 – 1533, 2004.

[21] IoannisM. Rekleitis, Gregory Dudek, and Evangelos Milios. Multi-robot exploration of an unknown environment efficiently reducing the odometry error. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1340–1345, Nagoya, Japan, 1997.

[22] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992.

[23] Peng Song and Vijay Kumar. A potential field based approach to multi-robot manipulation. In *Proceedings of the International Conference on Robotics and Automation*, pages 1217–1222, 2002.

[24] H. G. Tanner and D. K. Christodoulakis. Cooperation between aerial and ground vehicle groups for reconnaissance missions. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, 2006. (to appear).

[25] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos. Nonholonomic navigation and control of multiple mobile manipulators. *IEEE Transactions on Robotics and Automation*, 19:53–64, 2003.

[26] Herbert G. Tanner. Flocking with obstacle avoidance in switching networks of interconnected vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3006–3011, New Orleans LA, 2004.

[27] Herbert G. Tanner, Ali Jadbabaie, and George J. Pappas. Stable flocking of mobile agents, part i: Fixed topology. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2010–2015, Maui, Hawaii, 2003.

[28] Herbert G. Tanner and Amit Kumar. Formation stabilization of multiple agents using decentralized navigation functions. In S. Thrun, G. Sukhatme, S. Schaal, and O. Brock, editors, *Robotics: Science and Systems I*, pages 49–56. MIT Press, 2005.

[29] Rene Vidal, Omid Shakernia, H. Jin Kim, David Hyunchul Shim, and Shankar Sastry. Probabilistic pursuitevasion games: Theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18(5):662–669, 2002.

[30] Brian Yamauchi. Continuous localization using evidence grids. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2833–2839, Leuven, Belgium, 1998.

[31] Brian Yamauchi and Pat Langley. Place recognition in dynamic environments. *Journal of Robotic Systems*, 14(2):107–120, 1997.

[32] A. Zelinsky, R.A. Jarvis, J.C. Byrne, and S. Yuta. Planning paths of complete coverage of an unstructured environment by a mobile robot. In *Proceedings of the International Conference on Advanced Robotics*, pages 533–538, 1993.