10-19-2006

Task-driven multi-formation control for coordinated UAV/UGV ISR missions

Herbert G. Tanner

DEPARTMENT OF
MECHANICAL ENGINEERING

The University of New Mexico

Task-driven multi-formation control for coordinated UAV/UGV ISR missions

Herbert G. Tanner [1]
Department of Mechanical Engineering
The University of New Mexico
Albuquerque, NM 87131
e-mail: `tanner@unm.edu`

UNM Technical Report: ME-TR-06-001

Report Date: October 19, 2006

[1] Supported by Sandia National Laboratories SURP grant # 480480.

Abstract

The report describes the development of a theoretical framework for coordination and control of combined teams of UAVs and UGVs for coordinated ISR missions. We consider the mission as a composition of an ordered sequence of subtasks, each to be performed by a different team. We design continuous cooperative controllers that enable each team to perform a given subtask and we develop a discrete strategy for interleaving the action of teams on different subtasks. The overall multi-agent coordination architecture is captured by a hybrid automaton, stability is studied using Lyapunov tools, and performance is evaluated through numerical simulations.

1 Introduction

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.

The goal of this work develop a mathematical formalism that allows modeling, analysis and design of complex group behaviors, which consist of coordinated sequential composition of elementary group actions and interaction between heterogeneous groups, and apply it to the problem of performing intelligence gathering, enabling persistent surveillance, and continuous reconnaissance by a heterogeneous mix of autonomous UAVs and UGVs.

Our motivation for developing this coordination and control architecture is the following intelligence, surveillance and reconnaissance (ISR) problem: human (HUINT) or imagery (IMINT) intelligence has provided with a cue pertaining to the location of a hostile platform. The information is given in the form of an area, which is likely to contain the target. The problem at hand is to dispatch a combined UAV/UGV task force to search the area quickly and either locate the target or report a false alarm.

The main hardware constraint that the UAV and UGV platforms are subject to, is power storage. Due to size, these vehicles have to operate with limited energy resources, which places a restrictive upper limit on the time the task has to be completed if refueling/recharging is not an option. Besides the fact that the original intelligence information has to be validated quickly, disengagement for renewing power supplies will allow the target to relocate. Thus, the vehicle formation maneuvers and the search strategy have to be coordinated, complete and efficient.

The solution we envision requires close cooperation and coordination between the UAV and UGV formations. It exploits the speed of the aerial vehicles, the maneuverability of the ground vehicles, and the pluralism of the combined air-ground task force in autonomous units. Efficiency is served by using the aerial vehicles for the search, while assigning the ground vehicles to guarding roles. Completeness is ensured by having the ground vehicles deploy in formations that monitor target movements between different regions. Using a "divide and conquer strategy, the original field is decomposed into several cells that are searched sequentially until the target is discovered in one of them.

We approach this problem adopting an approach that involves completing an ordered sequence of subtasks, leading to a definitive answer as to whether the target is in the area, and its general location in case of an affirmative answer. The vehicle teams have delineated roles, according to their mobility and sensing capabilities, and are assigned to different subtasks. The teams coordinate and synchronize in order to ensure that the subtask sequence is of the desired order. The overall control and coordination architecture is captured by a hybrid automaton. Within this theoretical framework, we design the closed loop dynamics of the teams at each mode of operation where a subtask is completed, as well as the switching rules that regulate the succession of the subtasks. Finally we provide bounds on the task completion time, based on the stability properties of the teams' continuous dynamics.

Cooperative control for multi-agent systems has rapidly evolved in the last few years, from reactive approaches [BA98, LA04], to centralized control architectures [OEH02, BLH01, TLK03], then to distributed approaches [JK02, OSM02, LF01, DD03, GP03], and finally to biologically inspired decentralized coordination algorithms [JLM02, CMKB04, OSM04, MBF04, Tan04]. One limitation of distributed architectures such as [OSM02, LF01, DD03] is that there is no general design methodology to specify the states where the system converges to. In addition, coordination problems that are not related to synchronization have not been addressed, mainly because they will inevitably require a fundamentally different approach. Li et al. [JAE06] 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 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 [CP97, ZJBY93], even under uncertainty regarding the position of obstacles and obstructions [GBL02]. 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 [Moo01], 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 [RDM97, BFM$^+$00]. In [RDM97], multiple robots explore the workspace, which is decomposed into cells, while minimizing their position errors. In the multi-agent "frontier-based" exploration strategy of [YL97, Yam98], robots in an effort to expand the map of the known environment. In the approach of [BFM$^+$00] 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 [Moo01], it is actually closer to [YL97] 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. [VSK$^+$02] combine pursuit evasion games with map building by a team of multiple UAVs and UGVs, and implemented sub-optimal solutions, without excessive computational overhead. Our approach in this paper is different, primarily because besides algorithmic completeness, another important consideration in the scenario at hand is energy requirements (as they can be 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 [TJP03].

Related to pursuit-evasion games are problems of guarding certain regions, in the sense that one needs to confine an evader before being able to intercept it. A class of "guarding" problems related to the scenario at hand is the one referred to as art-gallery problems [She92]. In these problems, the question is where to place a number of guards having omnidirectional vision capabilities, within a workspace partitioned in polygonal cells, so that all points in the workspace are monitored by some guard. For the application we consider here, the assumption on omnidirectional vision is relatively restricting. We will assume guards having limited and directed sensing capabilities. The latter relates the application more to a particular subclass of art-gallery problems, where guards are modelled as searchlights [SSY90]. In this field there is a rotating beam emitted from each guard, scanning the workspace without being able to penetrate workspace boundaries (walls). In case of multiple guards [YSK04, OGB06], one has to devise a scheduling strategy so that the intruder cannot slip into cleared regions, and the whole workspace is covered in finite time. Each guard beam extends to the point where it intersects with the boundary, and thus the range of sensing is unlimited, determined by the environment only. For the type of guards (UGVs) envisioned in this scenario (depicted in Figure 1), existing solutions cannot be applied.

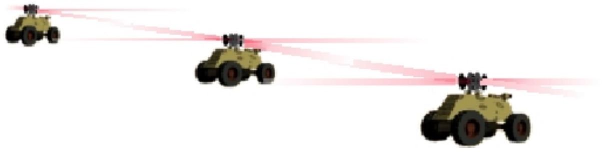

Figure 1: The role envisioned for the UGVs. Their four motion detection sensors can pivot around a vertical axis, so that the sensors can point independently of the vehicle's orientation. When appropriately configured, they can create a "virtual fence," capable of detection of motion to and from an area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

The power of hybrid systems as a mathematical formalism that captures both discrete and continuous interactions between agents in homogeneous groups has been demonstrated early on, through successful approaches to Air Traffic Control management [TPS98] and formation control [RFO01, NAS04, ZKS03]. So far, the hybrid formalism is used to describe interactions between homogeneous agents. In formation control approaches [RFO01, NAS04, ZKS03] hybrid mode transitions capture formation reconfigurations. Similarly, formation topology switching has been described using dynamic graphs [Mes02]. The existence of alternative modeling and analysis methods for formation control indicates that the hybrid formalism has not been exploited to its full extend, whereas in heterogeneous multi-agent systems with complex task-driven control transitions, all other representations seem inadequate. A hybrid model, therefore, emerges as the natural way to capture and analyze the evolution of heterogeneous multi-agent interactions.

2 Formal Problem Statement

For modeling purposes, ground vehicles are assumed to be capable of holonomic, omnidirectional motion and be kinematically equivalent to single integrators. 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

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

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. 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,

$$\ddot{y} = w, \quad y, w \in \mathbb{R}^{2N_a}. \tag{1}$$

The UAVs are assumed to be equipped with cameras, obtaining images of a certain disk-shaped region on the ground. The camera footprint of each UAV is fixed and moves along with the UAV.

If the target is found within one of the footprint regions of the UAVs, or it intersects with one of the rays of the UGV motion detectors, it is considered located and intercepted.

The problem we address in this work is stated as follows:

> Develop control strategies for the deployment and coordination of a mixed UAV/UGV team, so that it can search a rectangular planar region and intercept an arbitrary moving target in finite time.

3 Approach to Solution

3.1 Search Area Partition

The size of the area that can be searched so that completeness is ensured, is determined by the sensing capabilities and the number of the available ground and aerial assets. What is needed is a method of *partitioning* the space into cells so that

1. ground vehicles can detect every transition of the target from one cell to another, and
2. 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. When these UAVs fly in close formation, their footprints cover a strip of certain length. Setting this length equal to the length of the cell edge, enables us to scan one such cell with one UAV formation pass.

The number of cells that can be securely covered depends on the number of available ground vehicles. 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, $\{x_d(i)\}$, $i = 1,\dots,N_g$, 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.

3.2 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, but it places no significant computational burden since it is polynomial time ($O(N^3)$).

3.3 UGV Motion Planning

The ground vehicles navigate to their designated grid points using a (centrally generated) artificial potential field. In earlier work [TK05] we introduced a new navigation function that is appropriate for multi-robot formation control and navigation. In [TK05] 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" [SK02] 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.

The navigation function is constructed according to [TK05], 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

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

where k_g is the function's tuning parameter, and $\beta(\cdot)$ is the product of all collision proximity (obstacle) functions $\beta(\cdot)_{ij}$

$$\beta(x) \triangleq \prod_{(i,j)\in N_g\times N_g,\, i\neq j} (\beta_{ij}(x) - b_0), \tag{3a}$$

which are defined as

$$\begin{aligned} \beta_{ij}(x_i,x_j) &\triangleq 1 - \frac{h(\varepsilon^2 - \|x_i - x_j\|^2)}{h(\varepsilon^2 - \|x_i - x_j\|^2) + h(\|x_i - x_j\|^2)} \\ b_0 &\triangleq 1 - \frac{h(\varepsilon^2 - R^2)}{h(\varepsilon^2 - R^2) + h(R^2)}, \end{aligned} \tag{3b}$$

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

$$h(r) \triangleq \begin{cases} 0, & r \leq 0 \\ \exp(-1/r^2), & r > 0. \end{cases}, \tag{3c}$$

Construction (3) is adapted from [Boo86], 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 $\left\|x_i - x_j\right\|$) that allows us to compute $\prod_{(i,j)\in N_g \times N_g,\, i\neq j} \beta_{ij}$ in (2) using only the (i,j) pairs of UGVs in close proximity to each other; namely (i,j) such that $\left\|x_i - x_j\right\| < \varepsilon$. Moreover, this property will enable us to decentralize navigation in the future according to [TK05].

Proposition 3.1. *The function defined in* (2) *is a navigation function.*

Proof. We will only sketch the proof, due to lack of space. A detailed proof essentially tracks the steps of the proof in [TK05]. Here we demonstrate that all of the intermediate results leading to the proof in [TK05], hold:

1. the configuration where $\|x - x_d\|$ is a nondegenerate critical point of φ_g; it can be verified that as in [TK05] 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 $\left\|x_i - x_j\right\| \to R$, $\beta \to 0$, and $\nabla\varphi_g \to -\frac{1}{e^{\beta^{1/k_g}}} \frac{\sum_{i=1}^{N_g}(x-x_d)^2}{k_g} \beta^{\frac{1}{k_g}-1} \nabla\beta$ [TK05]. With $\nabla\beta_{ij}(R)$ being equal to

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

   and with the product $\beta_{ij}^{1/k_g - 1} \nabla\beta_{ij}$ at R evaluated at

$$\beta_{ij}^{1/k_g-1}\nabla\beta_{ij} = \frac{-4\varepsilon^2 \left(3R^4 - 3R^2\varepsilon^2 + \varepsilon^4\right)}{\left(e^{R^{-4}} + e^{\left(R^2-\varepsilon^2\right)^{-2}}\right)^2 R^5 \left(R^2-\varepsilon^2\right)^3} \cdot e^{R^{-4} - \left(R^2-\varepsilon^2\right)^{-2}} \left(\frac{e^{\left(R^2-\varepsilon^2\right)^{-2}}}{e^{R^{-4}} + e^{\left(R^2-\varepsilon^2\right)^{-2}}}\right)^{\frac{1}{k_g}+1} > 0,$$

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

3. 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 [TK05].

4. $\nabla^2\beta_{ij}$ evaluated at R has a single root for R in $[0,\varepsilon]$, 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 $\hat{v}^T \nabla^2 \varphi_g \hat{v} < 0$ [TK05]. For example, if one sets $R = \frac{\varepsilon}{2}$, then

$$\nabla^2\beta_{ij} = \frac{-1024\, e^{\frac{128}{9\varepsilon^4}} \left(12544 + 891\,\varepsilon^4 + e^{\frac{128}{9\varepsilon^4}} \left(-12544 + 891\,\varepsilon^4\right)\right)}{729 \left(1 + e^{\frac{128}{9\varepsilon^4}}\right)^3 \varepsilon^{10}},$$

   which is positive and strictly increasing with ε.

5. the degeneracy of critical points in the free space can be established as in [TK05], independently of β_{ij}.

□

Therefore, (2) 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.

3.4 UAV Formation Control

The aerial vehicles are steered into a formation that scans uniformly the cells in the grid defined in section 3.1 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 [TJP03], by replacing the inter-agent potential function with a multi-agent navigation function of the same form as the one constructed in section 3.3.

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 $i \sim j$. To meet these inter-UAV relative position specifications encoded in c_{ij}, the UAVs are assumed to exchange position and velocity information locally. By local information exchange we mean that a UAV does not need to broadcast its information to and receive data from all other UAVs; instead only a certain subset of the team is thought to be a “network neighbor” of each agent. The inter-agent relative position specifications and communication requirements are captured formally in our definition of the formation graph:

Definition 3.2 (Formation graph). *The formation graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{L}\}$, is a directed labeled graph consisting of:*

- *a set of vertices (nodes), $\mathcal{V} = \{1, \ldots, N\}$, indexed by the mobile agents in the group,*
- *a set of edges, $\mathcal{E} \subseteq \{(i,j) \in \mathcal{V} \times \mathcal{V}\}$, containing ordered pairs of nodes that represent relative position objectives, and*
- *a set of labels, $\mathcal{L} = \{c_{ij} \in \mathbb{R}^n, \mid (i,j) \in \mathcal{E}\}$, that set the specifications for inter-agent objectives.*

Formation graph edges imply the existence of a constant bidirectional communication link between agents, because systems that are unaware of each other’s state will not be able to coordinate toward a state-dependent common objective. If $\mathcal{G}$ is (weakly) connected, all agents have a specific desired configuration in the formation. Relative positions remove the ambiguity introduced by desired relative distances. Once the relative positions are set, both the position and orientation of the formation is specified.

Then the goal configuration for the UAV formation is described by the zero level set of the function:

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

Collision proximity is encoded using (3), 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. At the time of writing, potential field forces (generated by $-\nabla\varphi_a$) are computed centrally, but decentralization is a straightforward next step, where only tuning of k_a is required [TK05].

Rimon and Koditschek, in their seminal paper on navigation functions [RK92], 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:

$$w = \underbrace{-\nabla_y \varphi_a + \frac{1}{2}\left(1_{N_a} \otimes \ell(y,t) - \dot{y}\right)}_{\text{guiding vector field}} \underbrace{-(L \otimes I_2)\dot{y}}_{\text{dissipative field}}, \tag{4}$$

where L is the Laplacian of the graph defined by the relative position vector specifications, c_{ij} [TJP03], and $\ell(y,t)$ is a formation control signal, acting as a common virtual leader, 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 a positive semi-definite matrix. 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 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 3.5.

The stability properties of the closed loop system (1)-(4) is established by considering the dynamics of relative position $(B^T\otimes I_2)y$ and velocity $(B^T\otimes I_2)\dot{y}$ vectors, where B is the incidence matrix of the formation graph $\mathcal{G}$:

Proposition 3.3. *The dynamics of $(B^T\otimes I_2)y$ induced by (1)-(4):*

$$(B^T\otimes I_2)\ddot{y} = (B^T\otimes I_2)\left(-\nabla\varphi_a + \frac{1}{2}(1\otimes\ell - \dot{y}) - (L\otimes I_2)\dot{y}\right),$$

stabilize asymptotically to the set where velocity vectors are synchronized, $\dot{y}\in\operatorname{span}\{1\}$, and relative positions have assumed their desired values making φ_a zero.

Proof. Note that φ_a depends on differences of the form $y_i - y_j$ [Tan07] and not on any y_i explicitly, and differentiating $V = \varphi_a + \frac{1}{2}\dot{y}^T(L\otimes I_2)\dot{y}$:

$$\begin{aligned}\dot{V} &= \nabla_{(B^T\otimes I_2)}\varphi_a^T(B^T\otimes I_2)\dot{y} + \dot{y}^T(B\otimes I_2)(B^T\otimes I_2)\left(-\nabla\varphi_a + \frac{1}{2}(1\otimes\ell - \dot{y}) - (L\otimes I_2)\dot{y}\right)\\ &= \nabla_{(B^T\otimes I_2)}\varphi_a^T(B^T\otimes I_2)\dot{y} - \dot{y}^T(B\otimes I_2)(B^T\otimes I_2)\nabla_y\varphi_a - \frac{1}{2}\dot{y}^T(L\otimes I_2)\dot{y} - \|(L\otimes I_2)\dot{y}\|^2. \qquad (5)\end{aligned}$$

By chain rule, $\nabla_{(B^T\otimes I_2)y}\varphi_a = \sum_{j\sim i}\nabla_{y_i}\varphi_a\frac{\partial y_i}{\partial(y_i-y_j)} = (B^T\otimes I_2)\nabla_{y_i}\varphi_a$, and therefore (5) is rewritten as

$$\dot{V} = -\frac{1}{2}\left\|(B^T\otimes I_2)\dot{y}\right\|^2 - ((B^T\otimes I_2)\dot{y})^T(L\otimes I_2)(B^T\otimes I_2)\dot{y}, \qquad (6)$$

which is negative semi-definite, and establishes the invariance of the interior of the level sets of V. The invariance principle suggests the existence of an attractive invariant set in the interior of the region where $(B^T\otimes I_2)\dot{y} = 0$, i.e., where velocity vectors are aligned. The dynamics of $(B^T\otimes I_2)y$ in this set reduce to

$$(B^T\otimes I_2)\ddot{y} = -(B^T\otimes I_2)\nabla_y\varphi_a \in \operatorname{span}\{1\},$$

which implies that $\nabla_y\varphi_a = 0$, because the range of B^T is orthogonal to 1. Since φ_a is a navigation function, the only stable configuration where we can have $\nabla_y\varphi_a = 0$ is the one corresponding to the desired values for the relative position vectors. Thus at steady state, we have that velocity vectors being aligned and relative position vectors stabilized at desired values. □

3.5 UAV-UGV Coordination

Aerial and ground vehicle groups are coordinated using state-dependent switches. Initially, 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 the UGV to a cell already covered, and thus escape detection.

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. In the current implementation, the switching signal is generated by monitoring the value of the centralized UGV navigation function; in a decentralized ground navigation setting, vehicles will first reach consensus over their proximity to their goal configurations, and then broadcast an appropriate message to the UAV group.

Once the UGV are in place on the grid, 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, 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). In what follows, we describe the structure of $\ell(y,t)$ and the switching conditions for the case of column scanning; the case of row scanning is dealt similarly.

The controller switching is implemented through $\ell(y,t)$, which is in-fact a piecewise continuous switching signal, and has the following 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 $\ell_{k+1}(y)$ can be given as

$$C_k = \frac{(-1)^{k-1}\sum_{i=1}^{N_a}(0,1)\cdot y}{N_a} - (-1)^{k-1}(\max\{(0,1)\cdot x_d\} + \varepsilon) > 0, \tag{7}$$

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.

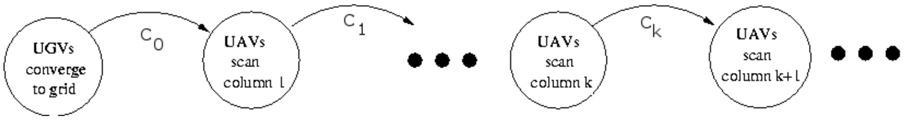

Figure 2: Controller mode transition in the UAV-UGV team. The UAV group scans the first column only after the UGVs are in position ($C_0 : \varphi_g < \varepsilon$, where $\varepsilon > 0$ is the desired accuracy). Condition $C_1 > 0$ refers to the expression of (7) for $k = 1$ being positive. After scanning column k, UAVs switch their controller to scan column $k+1$ as soon as $C_k > 0$ is satisfied.

4 Numerical Simulations

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. 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 (3) 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, they navigate toward their designated grid points following the gradient of (2); 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. At no point does this distance fall below 0.14 km,

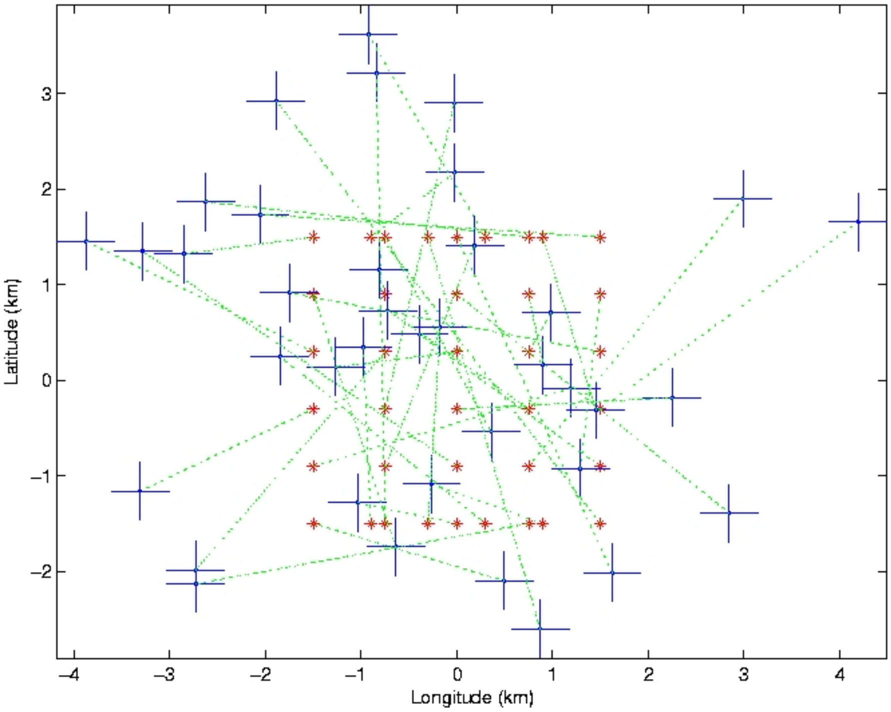

Figure 3: The initial positions of UGVs $(\cdot)$ and the desired grid points $(*)$ marking the boundaries of the search area. The cross-marks $(+)$ over the UGV positions mark the directions and range of their target detection systems. The faint (green) dotted lines connecting UGVs with grid points represent goal assignments for each UGV as output of the Hungarian algorithm.

which is the value at initialization. Note that in the final configuration (Figure 6), the "shells" partially overlap (but not the protected regions); this suggests that the collision avoidance does not inhibit convergence.

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 and velocities, south of the UGV 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 sharpness of the turn as shown in Figure 7 is due to the fact that our simulation is a discrete time implementation, with no bounds on the magnitude of the control signals.

5 Estimation of Subtask Completion Times

5.1 Convergence of UGVs to grid points

Control inputs u_i are selected as $u_i = -\nabla_{x_i}\varphi_g(x)$, where x is the stack vector of all x_i and φ_g is a navigation function [RK92] constructed in [Tan07], satisfying $\varphi_g > 0$ anywhere except for the goal configuration for the UGVs, $x_d \equiv 0$, in which $\varphi_g(0) = 0$. This choice of inputs guarantees uniform asymptotic convergence to x_d except for a set of initial conditions M of measure zero. This set contains configurations where $\nabla\varphi_g$ vanishes. By continuity, however, it is possible to find $\underline{\varphi_g} \triangleq \min_{x:\|\nabla\varphi_g\|\leq\delta}\varphi_g(x)$, for some small $\delta > 0$, and define the set $S = \{x \mid \varphi_g < \underline{\varphi_g}$. Restricting the set of initial conditions for the UGVs on $\mathrm{R}^{2N} \setminus \{S \cup M\}$, we have that $\dot{\varphi}_g = -\|\nabla\varphi_g\|^2 \leq -\delta^2$. The Comparison Principle allows us to determine a upper bound on the time needed for

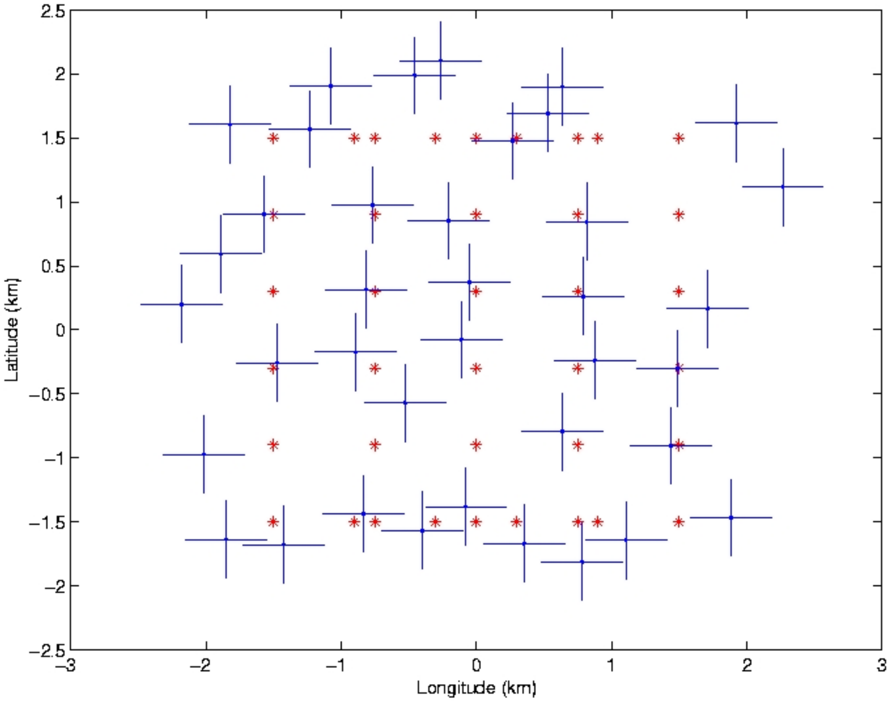

Figure 4: The UGVs (marked by $+$) on their way to their designated grid points (marked by $*$). The figure shows an intermediate configuration of the UGV group.

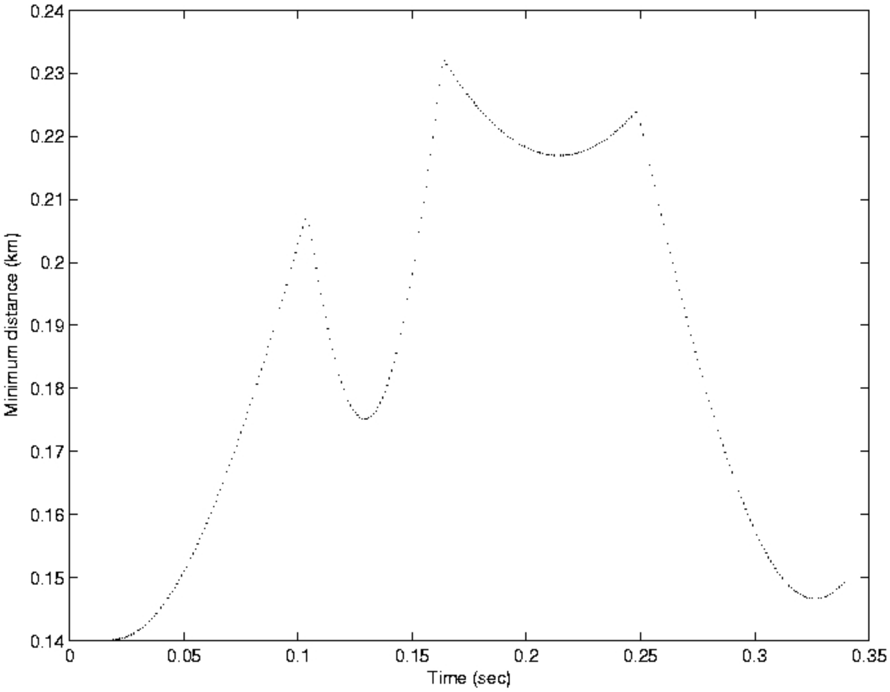

Figure 5: The evolution of the minimum distance between UGVs over time.

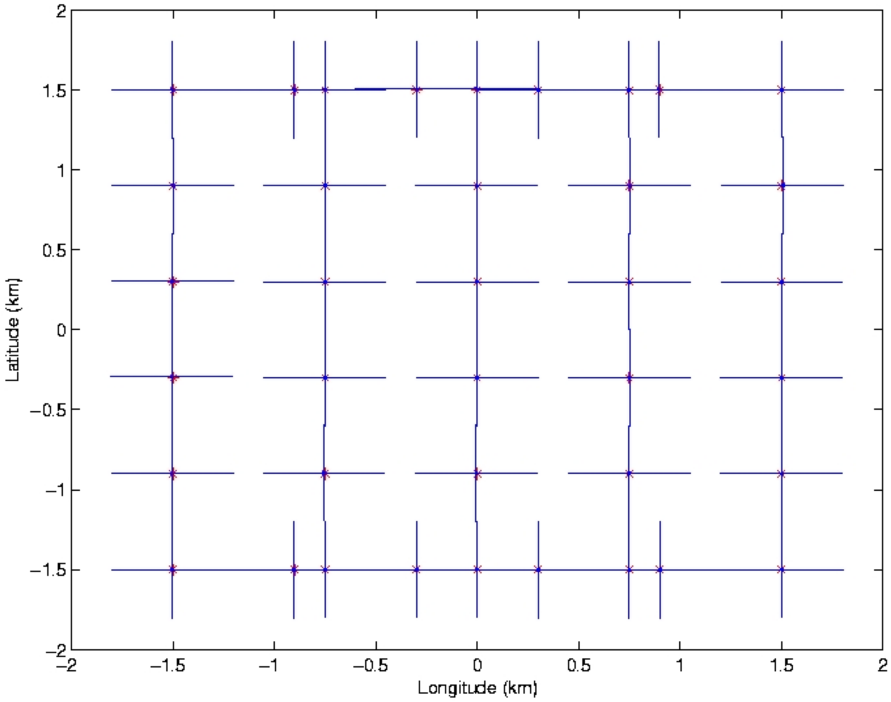

Figure 6: The UGVs in their final configuration, stabilized around their designated goal points. Note how the lines marking their target detection ranges mark the boundaries of the search area, as well as the boundaries of each column that will be subsequently scanned by the UAV team.

φ_g to drop below a certain threshold, say ε:

$$T_g = \frac{\varphi_g - \varepsilon}{\delta}. \tag{8}$$

Using this time estimate, the UAV team can initiate its maneuver into the search area without the need for explicit communication with the UGV team. The state-based transition conditions of (7) and Figure 2 can then be replaced by timed transition conditions.

5.2 Convergence of UAVs to formation

From (6), $\dot{V} \leq -\frac{1}{2}\left\|(B^T \otimes I_2)\dot{y}\right\|^2$, it follows using the comparison principle that

$$V(t) \leq 2\left(t + 2\left\|(B^T \otimes I_2)\dot{y}(0)\right\|^{-1}\right)^{-1}.$$

If the left hand side is set equal to a small constant that expresses the desired accuracy for formation stabilization, ε_a, then the time required for the UAVs to fall into formation can be estimated as:

$$T_a = \frac{2}{\varepsilon_a} - 2\left\|(B^T \otimes I_2)\dot{y}(0)\right\|^{-1}. \tag{9}$$

The terminal velocity of the formation (from which a time required to scan a column could be estimated) is difficult to obtain from the Lyapunov analysis, because the control terms $-\nabla\varphi_a$ and $-(L \otimes I_2)\dot{y}$ do not change the energy of the system, but the term $\frac{1}{2}(1 \otimes \ell - \dot{y})$ is damping-like and it does. Thus, no "energy-conservation" argument can be applied.

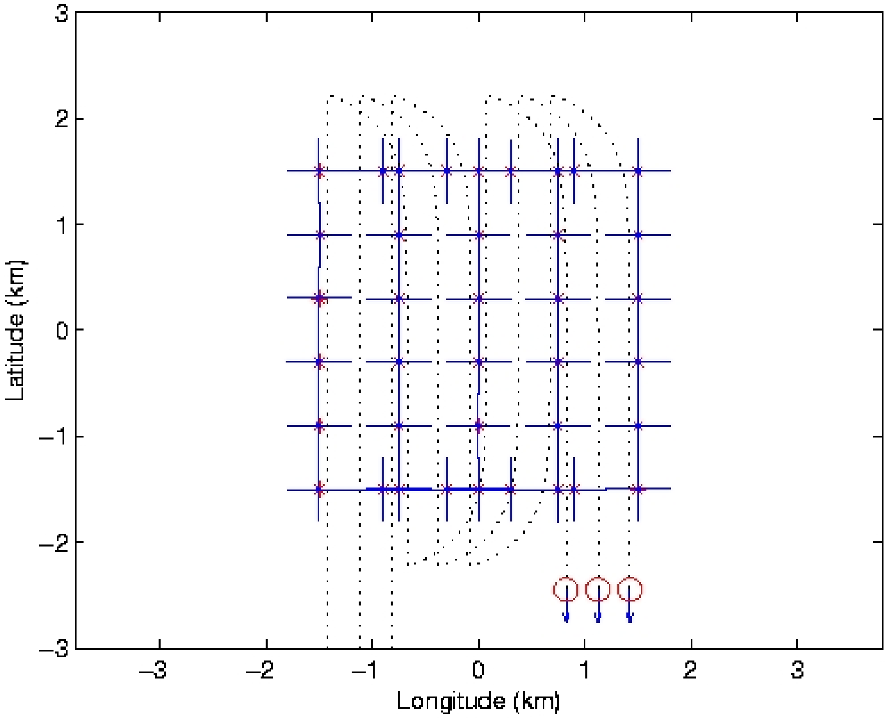

Figure 7: Paths of the UAVs as they scan the search region, column by column. They approach from bottom-left corner of the figure and depart the scene from the bottom-right corner. As soon as they complete scanning one column, they reverse direction to scan the next.

However, after time T_{a_1} UAV velocities have synchronized and stabilized to some value $\dot{\bar{y}}$. At that time, it is possible to estimate the time required to scan column k of length s_k as

$$T_{a_k} = s_k \cdot \dot{\bar{y}}^{-1}.$$

5.3 Using the Task Completion Times for Coordination

Having obtained task completion times we can implement time-based controller switching instead of state-dependent transition rules, reducing the communication overhead. Using these time abstractions, we can implement a new switching regime, under which units know how long each controller is supposed to stay active, and does not need to communicate state information to determine whether a switching condition (hybrid guard) is activated.

The time estimate of (9) can also be used to determine a region for the initial positions for the UAVs: the UAV that is closest to the search area grid should be positioned at a distance

$$d_0 = \max_{j\in\{1,\ldots,N_a\}} \|\{\| \dot{y}(0)\} \cdot T_{a_1},$$

so that the UAV formation is stabilized *before* reaching the search area.

Assuming that UAV initial conditions satisfy the condition above, a possible time-based switching scheme can be constructed as follows:

- Time $[0, T_g]$: UGVs move into grid positions.

- Time $(T_g, T_g + 2T_a + T_{a_1}]$: UAVs operate under controller (4) for $\ell(y,t) = \ell_1(y)$.
- Time $(T_g + 2(k-1)T_a + \sum_{m=1}^{k-1} T_{a_m}, T_g + 2kT_a + \sum_{m=1}^{k} T_{a_m}]$: UAVs operate under controller (4) for $\ell(y,t) = \ell_k(y)$.

Therefore, an upper bound on the time to complete the search task in an area decomposed into s columns, T_{total}, can be obtained directly as

$$T_{\text{total}} = T_g + 2sT_a + \sum_{m=1}^{s} T_{a_m}.$$

6 Concluding Remarks

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 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. However, there are recent results on decentralized role assignment [AH05], allowing distributed algorithm execution. In addition, we have shown in earlier work that such provably convergent group navigation and formation control can be decentralized [TK05]. Future work is therefore directed toward incorporating these developments, and producing a fully decentralized and scalable cooperative search and detection strategy.

Acknowledgments

This work was made possible through SURP grant # 480480 and the support of the Intelligent Systems and Robotics Center of the Sandia National Laboratories. The author thanks Dimitris Christodoulakis for his help in literature review, problem formulation and UAV control design, and Sirisha Kota for assistance in simulation studies and visualization of simulation data. The idea for using Boothbys example for navigation came out of discussions with Grigoris Lionis. The implementation of the Hungarian algorithm was by Niclas Borlin.

References

[AH05] 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.

[BA98] T. Balch and R.C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926 – 939, 1998.

[BFM$^+$00] 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.

[BLH01] 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.

[Boo86] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 2nd edition, 1986.

[CMKB04] 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.

[CP97] 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.

[DD03] R. DAndrea and G. E. Dullerud. Distributed control of spatially interconnected systems. *IEEE Transactions on Automatic Control*, 48(9):1478– 1495, 2003.

[GBL02] Hector H. Gonzalez-Banos and Jean-Claude Latombe. Navigation strategies for exploring indoor environments. *Journal of Robotic Systems*, 21(10-11):829–848, 2002.

[GP03] V. Gazi and K.M. Passino. Stability analysis of swarms. *IEEE Transactions on Automatic Control*, 48(4):692–696, 2003.

[JAE06] 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.

[JK02] 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.

[JLM02] 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.

[LA04] 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.

[LF01] 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.

[MBF04] 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.

[Mes02] Mehran Mesbahi. On a dynamic extension of the theory of graphs. In *In Proceedings of the American Control Conference*, pages 1234–1239, 2002.

[Moo01] Stewart John Moorehead. Autonomous surface exploration for mobile robots. Technical Report CMU-RI-TR-01-30, The Robotics Institute, Carnegie Melon University, 2001.

[NAS04] David J. Naffin, Mehmet Akar, and Gaurav S. Sukhatme. Lateral and longitudinal stability for decentralized formation control. In *In Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems*, pages 421–430, 2004.

[OEH02] 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.

[OGB06] K. J. Obermeyer, A. Ganguli, and F. Bullo. Asynchronous distributed searchlight scheduling. In *Proceedings of the 46th IEEE Conference on Decision and Control*, San Diego, CA, 2006. (submitted).

[OSM02] 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.

[OSM04] 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.

[RDM97] 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.

[RFO01] V. Kumar R. Fierro, A.K. Das and J.P. Ostrowski. Hybrid control of formations of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 157–162, 2001.

[RK92] E. Rimon and D. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992.

[She92] Thomas C. Shermer. Recent results in art galleries. *Proceedings of the IEEE*, 80(9):1384–1399, 1992.

[SK02] 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.

[SSY90] K. Sugihara, I. Suzuki, and M. Yamashita. The searchlight scheduling problem. *SIAM Journal on Computing*, 19(6):1024–1040, 1990.

[Tan04] 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.

[Tan07] Herbert Tanner. Switched uav-ugv cooperation scheme for target detection. In *In Proceedings of the IEEE Conference on Robotics and Automation*, 2007. (submitted).

[TJP03] 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.

[TK05] 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.

[TLK03] 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.

[TPS98] Claire Tomlin, George J. Pappas, and Shankar Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998.

[VSK$^+$02] 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.

[Yam98] Brian Yamauchi. Continuous localization using evidence grids. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2833–2839, Leuven, Belgium, 1998.

[YL97] Brian Yamauchi and Pat Langley. Place recognition in dynamic environments. *Journal of Robotic Systems*, 14(2):107–120, 1997.

[YSK04] M. Yamashita, I. Suzuki, and T. Kameda. Searching a polygonal region by a group of stationary k-searchers. *Information Processing Letters*, 92(1):1–8, 2004.

[ZJBY93] 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.

[ZKS03] S. Zelinski, T.J Koo, and S. Sastry. Hybrid system design for formations of autonomous vehicles. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 1–6, 2003.