Swarm Formation Control Utilizing Elliptical Surfaces and Limiting Functions

Laura E. Barnes, *Member, IEEE*, Mary Anne Fields, and Kimon P. Valavanis, *Senior Member, IEEE*

***Abstract*—In this paper, we present a strategy for organizing swarms of unmanned vehicles into a formation by utilizing artificial potential fields that were generated from normal and sigmoid functions. These functions construct the surface on which swarm members travel, controlling the overall swarm geometry and the individual member spacing. Nonlinear limiting functions are defined to provide tighter swarm control by modifying and adjusting a set of control variables that force the swarm to behave according to set constraints, formation, and member spacing. The artificial potential functions and limiting functions are combined to control swarm formation, orientation, and swarm movement as a whole. Parameters are chosen based on desired formation and user-defined constraints. This approach is computationally efficient and scales well to different swarm sizes, to heterogeneous systems, and to both centralized and decentralized swarm models. Simulation results are presented for a swarm of 10 and 40 robots that follow circle, ellipse, and wedge formations. Experimental results are included to demonstrate the applicability of the approach on a swarm of four custom-built unmanned ground vehicles (UGVs).**

***Index Terms*—Formation control, multiagent systems, potential fields, swarms.**

I. Introduction

In RECENT years, multiagent systems have been suggested as a means of accomplishing tasks in the battlefield and other environments. Such applications include finding and neutralizing mines, containing spills, forming perimeters, and providing ad hoc reconfigurable communication networks. However, most of these applications assume that the swarm members begin their activities in the mission space—they do not address how the robots get to the mission space. In some scenarios, groups of unmanned vehicles need to autonomously travel from one region in the mission space to another region while remaining as a cohesive unit. This case is particularly true in battlefield applications—robots and soldiers start in an assembly area and then travel to an area of interest to start the mission.

Manuscript received January 27, 2009. First published May 12, 2009; current version published November 18, 2009. This paper was recommended by Associate Editor E. Santos, Jr.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. In this experiment four UGV vehicles travel in an ellipse formation surrounding the virtual swarm center. A GPS coordinate is used for the swarm center. This coordinate is a function of time which all robots receive periodically. The four UGVs travel surrounding each center point and staying at a minimum specified distance away from one another. This material is 4.70 MB in size.

L. E. Barnes is with the Automation and Robotics Research Institute, University of Texas at Arlington, Arlington, TX 76019 USA.

M. A. Fields is with the Intelligent Control Team, Unmanned Systems Division, Vehicle Technologies Directorate, Army Research Laboratory, Adelphi, MD 20783-1197 USA.

K. P. Valavanis is with the Department of Electrical and Computer Engineering, University of Denver, Denver, CO 80208 USA.

Digital Object Identifier 10.1109/TSMCB.2009.2018139

Different formations, e.g., diamond, wedge, line, and column, allow soldiers to respond to potential threats in a variety of terrains and battlefield conditions while maintaining contact with each other. In future applications, robot teams and mixed robot teams will be expected to move in military-style formations.

Numerous methods have been proposed for formation control. In the virtual-structure approach, the entire formation is treated as a single rigid body, and agents maintain a specific geometry among each other [1]–[4]. In [2], a strategy is presented to arrange a large-scale homogeneous team in a geometric formation by utilizing potential functions with preset attachment sites. However, there is a limited number of formations that can be achieved, and the method is computationally expensive. In [4], a control architecture is presented for formation control that considers leader-following, behavior, and virtual-structure approaches, combining the advantages of each.

In [5], a potential field methodology for formation control is presented. The desired formation pattern is represented in terms of queues and formation vertices. The desired pattern and trajectory for the group of robots is represented by artificial potential trenches. Each robot is attracted to and moves along the bottom of the potential trench, automatically distributing with respect to each other. Queues and formation vertices are defined based on the number of robots and the desired formation. Although this method greatly improves on node to robot formation structures, queues and vertices are still calculated based on the formation and number of robots. In [6] and [7], this work is improved upon by utilizing a limited communication scheme. In [8], a set of artificial points is used as beacons that guide the robots to their goal. This approach uses the geometric relationship between beacon points to move the robots in formation. Other examples of potential field approaches are shown in [8]–[17].

Centralized methods for formation control have successfully been used in [18]–[25]. However, relying on one resource for command of the swarm is prone to failure in dynamically changing and unstructured environments. Thus, a decentralized or hybrid control method is preferred in swarm systems [16], [26]–[28].

Behavior-based approaches are used to control formations for constellations of spacecraft and for multirobot tasks such as box pushing [29], [30]. The approaches are advantageous, because they are decentralized and require less communication, but group behavior cannot explicitly be defined.

Leader–follower strategies have also been proposed for the formation control of multirobot systems [31]–[38]. In these approaches, some agents are designated as leaders and others as followers. In [33] the leader–follower approach is extended by adding a control graph that defines the relative position of

1083-4419/$25.00 © 2009 IEEE

each robot in the formation. By maintaining a relative distance and orientation with respect to the reference robot, the team of robots can exhibit several different formations. In [34], local sensing and minimal communication between agents is used to maintain a predetermined formation. Each robot in the group keeps a single "friend" robot that it knows from an appropriate sensor and maintains a specific angle at all times in relation to this "friend" robot.

There are also a number of graph-based formation control strategies [19], [39]–[42]. Other methods of formation control are presented in [18] and [43]–[52].

The proposed approach is a hybrid approach that can be either completely distributed compared with the approaches in [18]–[25] or a fusion of approaches that contain facets from multiple formation control strategies. Potential fields are the underlying method of control to attract swarm members to a single adaptable virtual structure. The formation can be held in a completely distributed manner by using only local information that sums different weighted vectors for formation keeping and obstacle avoidance. Moreover, a hybrid approach with leaders and followers can also be utilized to create a tighter formation.

The strategy is also platform independent. The proposed method is also applicable to heterogeneous swarms, because the vector generation is independent of the specific robot vehicle platform (compared with the approaches in [8], [17], and [53] that are not scalable in heterogeneity).

The main contribution of this paper is the introduction of limiting functions to hold multirobot systems on an enclosed curvilinear formation structure. In this paper, we generalize our approach to ellipse-based configurations. By adjusting the control parameters on the surface, we can control the shape and extent of the formation, as well as the relative distance between swarm members. These formations can move as a unit, adapt to nonuniform surfaces, and dynamically change.

All swarm members are attracted to a single differentiable surface unlike in approaches where nodes or vertices must be calculated for different formations and swarm members. For the purposes of this work, only a single surface is used, but multiple surfaces could be used to create more complicated formations. This factor makes the approach scalable to large swarm sizes. In addition, very little information transmission is required to hold formation. Swarm members only need to know formation parameters and nearby neighbors for the dispersion aspect of the formation. The proposed method is demonstrated with simulations of 10 and 40 robots. In addition, real-time experiments are run with three and four custom-built unmanned ground vehicles (UGVs). Ellipse and line formations are experimentally demonstrated. Preliminary results of this research are presented in [54] and [55]. Table I shows the nomenclature used in this paper.

TABLE I
NOMENCLATURE

$f(x, y)$	main swarm function
α	controls distance vector field is aloud to exist
γ	control variable in y direction (*YLength / XLength*)
x_c, y_c	swarm surface center in world reference frame
XLength (2A)	major axis of the swarm surface
YLength (2B)	minor axis of the swarm surface
d_x, d_y	velocity vectors in the x and y directions
x_{rot}, y_{rot}	*x* & *y* coordinates in the rotated reference frame
Φ	heading between the swarm formation x-axis and the center ($x_c y_c$)
R^*,R_{in} ,R_{out}	optimal, inner , and outer elliptical rings
ΔR_{in} , ΔR_{out}	distance from the R^* band to the inner and outer boundaries of the formation band
R	Euclidean distance from swarm member to center ($x_c y_c$)
x_{co}, y_{co}	obstacle center location
S_{out}, S_{in}, $N_\perp$	limiting functions away from the center, towards the center, and perpendicular to the center respectively
SGN	multiplier to change direction of perpendicular field about the x-axis
κ	obstacle avoidance tuning variable
α_{out}, α_{in}, $\alpha_\perp$	control variables used in respective limiting functions S_{out}, S_{in}, and $N_\perp$
ΔR_{avoid}	desired distance to maintain from obstacles and/or other swarm members
r_{avoid}	Euclidean distance from swarm member to obstacle
S_{avoid}	limiting function to control obstacle avoidance

II. Problem Formulation

A. Generation of Formation Surface

The main objective of the proposed approach is to attract elements of a swarm into a bounded formation and allow the swarm to stay in that formation as it moves around the mission space. A vector field is used to attract swarm members to an ellipse with desired parameters. The minimum distance between swarm members is controlled using an additional vector field.

At any instant in time, UGVs can be visualized as particles that move in a potential field that was generated from a bivariate normal "hill" that controls the velocity and heading of the swarm members. A bivariate normal function of the form

$$f(x,y) = e^{-\alpha\left((x-x_c)^2+\gamma(y-y_c)^2\right)} \tag{1}$$

produces an oval/ellipsoid-shaped function. Assuming that the current robot location is at (x, y), the center of the function in (1) is represented by (x_c, y_c) with respect to the world reference frame. The control variable γ determines the ratio of the minor axis (y-direction) to the major axis (x-direction), which affects the eccentricity of the swarm. Note that the center (x_c, y_c) and the orientation could be a function of time, allowing the swarm to move along a path.

The x and y partial derivatives create the velocity vectors for determining the heading and velocity of each member of the swarm as follows:

$$\begin{aligned} d_x &= -2\alpha f(x,y)(x-x_c) \\ d_y &= -2\alpha\gamma f(x,y)(y-y_c). \end{aligned} \tag{2}$$

Similar to a single UGV, the swarm formation, which is treated as a single structure, has both a local reference and a world reference frame. For the swarm to follow a trajectory in the world reference frame, an axis rotation is required. The heading ϕ between the swarm formation's x-axis and the center

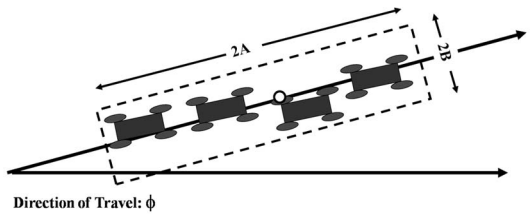

Fig. 1. Convoy description.

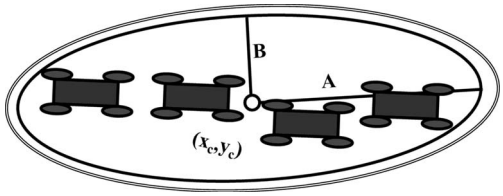

Fig. 2. Convoy of vehicles surrounded by concentric ellipses.

(x_c, y_c) must be found. The rotated coordinates for (x, y) and (x_c, y_c) can be found using

$$x_{\text{rot}} = \cos(\phi)(x - x_c) - \sin(\phi)(y - y_c)$$
$$y_{\text{rot}} = \sin(\phi)(x - x_c) + \cos(\phi)(y - y_c). \quad (3)$$

The rotated coordinates are then substituted to find d_x and d_y.

B. Formation Problem

To describe the general formation problem, it is discussed in reference to convoy protection. Suppose that a swarm of UGVs needs to accompany a convoy of vehicles, surrounding them in a particular formation. In the general case, the convoy can be enclosed in some geometric shape, which is loosely defined by dimensions, direction of travel, and the center of mass, as shown in Fig. 1. The length of the convoy along the axis of travel is 2A. The width of the convoy with respect to the axis of travel is 2B.

A field needs to be designed to attract swarm members to surround the convoy in a designated formation. The swarm members need to be close enough to the convoy to offer protection but far enough to allow the convoy to safely move.

Assume that the positions of each of the convoy vehicles are known and that the centroid of the convoy is (x_c, y_c). It is possible to enclose the convoy within a sequence of concentric ellipses with center (x_c, y_c). Fig. 2 depicts three elliptical rings with center (x_c, y_c), semimajor axis A, and semiminor axis B, surrounding a convoy of vehicles.

By attracting swarm members to the center elliptical ring that is described as the set of points $(x, y) \in \Re^2$ that satisfies

$$R^{*2} = (x - x_c)^2 + \gamma^2(y - y_c)^2 \quad (4)$$

where (x_c, y_c) is the center, and γ is the axis ratio B/A, the swarm can closely be associated with the convoy without endangering the convoy vehicles. For a fixed value of γ, refer to the set of points (x, y) that satisfies (4) as the R^* ellipse.

The general form of the swarm controller is described by

$$V(x, y, t) = \sum_{1}^{N} w_i(x, y, t) V_i(x, y, t) \quad (5)$$

where $V(x, y, t)$ gives the velocity of the swarm at a particular time and place. Each of the vectors $V_i(x, y, t)$ is associated with different fields, and $w_i(x, y, t)$ are weights of the overall contribution of the ith vector. In general, the field $V(x, y, t)$ is the weighted sum of N different vectors, each of which is acting on the swarm. In this case, three different vector fields are utilized: 1) one attracts UGVs to the elliptical band from points outside the elliptical region; 2) one pushes UGVs away from the center toward the desired band; and 3) one controls the movements of the UGVs within the band.

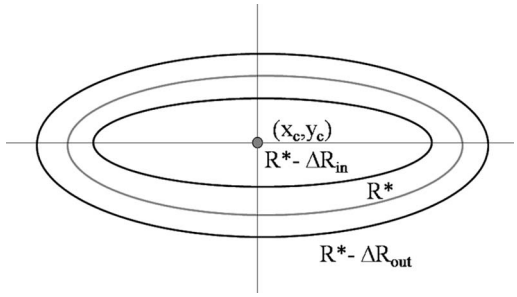

Fig. 3. Elliptical attraction band for the swarm.

The challenge is to create a potential field-based controller by using a small number of physically relevant weights w_i and vectors v_i that attract UGVs (particles) to a neighborhood of the R^* ellipse. This neighborhood is shown in Fig. 3. The parameters R_{in} and R_{out} denote the inside and outside boundaries of the R^* neighborhood, respectively, as also depicted in Fig. 3. The desired vector fields will "trap" the UGVs (particles) in these bands. Typically, this is a very narrow band of allowable space for the UGVs with a controllable width of $\Delta R_{\text{in}} + \Delta R_{\text{out}}$, where

$$R_{\text{in}} = R^* - \Delta R_{\text{in}} \quad (6)$$
$$R_{\text{out}} = R^* + \Delta R_{\text{out}}. \quad (7)$$

The vector field is constructed utilizing the normalized gradient from (2). For every (x, y), let the gradient field vector have the form

$$V_i(x, y) = \begin{cases} W_i(x, y) \frac{1}{L(x,y)} \begin{pmatrix} (x - x_c) \\ \gamma(y - y_c) \end{pmatrix}, & \text{for } (x, y) \neq (x_c, y_c) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, & \text{for } (x, y) = (x_c, y_c) \end{cases} \quad (8)$$

where

$$L(x, y) = \sqrt{(x - x_c)^2 + \gamma^2(y - y_c)^2}. \quad (9)$$

The vector $(1/L(x, y))\binom{(x-x_c)}{\gamma(y-y_c)}$ is a unit vector that provides the direction of the vector at (x, y). The function $w(x, y)$ provides the magnitude of the vector at that point. Notice that, for any (x, y), this vector points *away* from the center of the ellipse.

In the defined vector field, particles starting within the $R^* - \Delta R_{in}$ ellipse with

$$R^* = \sqrt{(x - x_c)^2 + \gamma^2(y - y_c)^2} \quad (10)$$

move out from the center until they reach the R^* neighborhood. UGVs (particles) that start outside the $R^* + \Delta R_{\text{out}}$ ellipse move toward the center until they reach the R^* neighborhood. Eventually, all UGVs will be trapped within the neighborhood, i.e.,

$$(R^* - R_{\text{in}}) \leq r \leq (R^* + R_{\text{out}}). \quad (11)$$

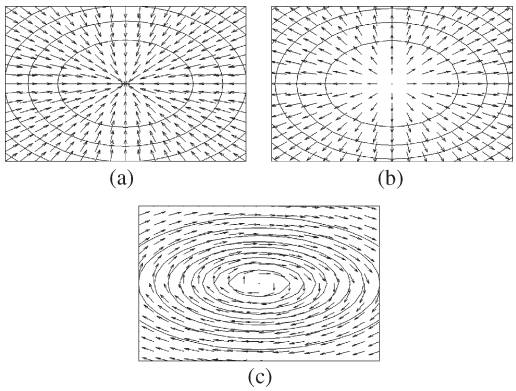

Fig. 4. Vector fields directed (a) toward the center (G^+), (b) away from the center (G^-), and (c) perpendicular to the center.

III. Generation of Vectors and Vector Fields

A. Description of Vector Fields

To generate the desired vector fields to hold UGVs inside the R^* neighborhood, three fields are needed: 1) one attracts swarm members to the elliptical band from points outside the elliptical region; 2) one pushes swarm members away from the center toward the desired band; and 3) one controls the movement of swarm members within the band. The first two fields utilize the gradient vector field that was discussed in the previous section, and $G^- = -(d_x, d_y)$ points away from the center, as shown in Fig. 4(b). Vector calculus dictates that the gradient vector field, $G^+ = (d_x, d_y)$, points in the direction of greatest increase of the function $f(x, y)$, which is toward the center, as illustrated in Fig. 4(a). The third vector field utilizes a vector that is perpendicular to the gradient vector. The vectors $(d_x, -d_y)$ and $(-d_x, d_y)$ are perpendicular to the gradient, and Fig. 4(c) shows such a perpendicular field.

B. Description of Limiting Functions

Tighter swarm control may be accomplished when restricting the influence of the vector fields to a small region of the xy plane by multiplying each of the fields by a limiting function. This limiting function controls the influence of the vector field in various regions of R^2. For example, the limiting function can determine the distance from the center at which the vectors in the field "die out" or become smaller than some number ε.

To create the desired field, the G^- and G^+ fields as shown in Fig. 4(a) and (b) must be limited to end at the appropriate boundaries. These fields will be limited with *sigmoid* functions or S-shaped functions.

Vector fields that "move away" from the center (the vectors inside the ellipse) require a limiting function that approaches zero as the distance from the center increases. Such a limiting function is given by

$$S_{\text{in}}\left(\alpha_{\text{in}}, r, R^*_{\text{in}}, \Delta R_{\text{in}}\right) = 1 - \frac{1}{1 + e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))}}. \quad (12)$$

Gradient vector fields that are directed toward the center (i.e., vectors outside the ellipse) are required to approach zero as the vectors "move toward" the center. This condition is achieved using the limiting function

$$S_{\text{out}}(\alpha_{\text{out}}, r, R^*, \Delta R_{\text{out}}) = 1 - \frac{1}{1 + e^{-\alpha_{\text{out}}(r-(R*+\Delta R_{\text{out}}))}}. \quad (13)$$

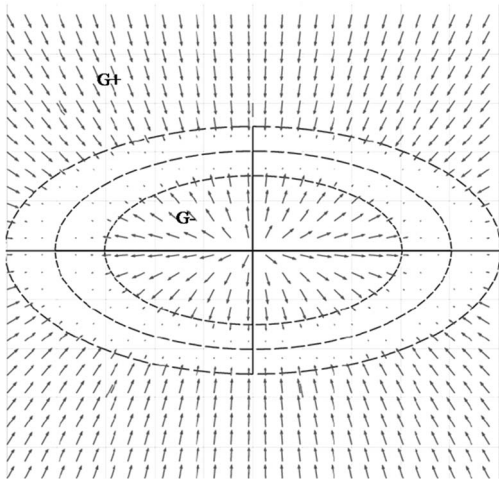

Fig. 5. Combined in (G^+) and out (G^-) fields.

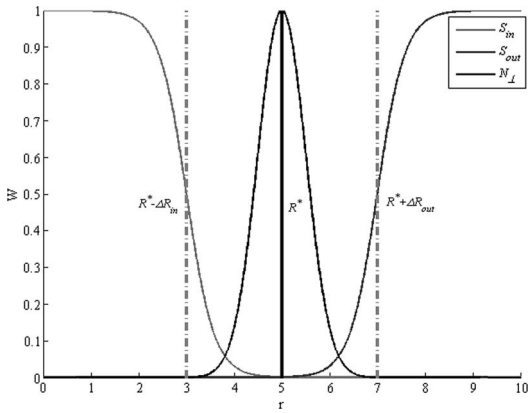

Fig. 6. Weighting functions S_{in}, S_{out}, and $N_\perp$ as a function of the weighted distance r defined in (16).

The G^- field should die out at $R^* - \Delta R_{\text{in}}$, and the G^+ field should die out at $R^* + \Delta R_{\text{out}}$. This case creates the field in Fig. 5.

Although Fig. 5 illustrates a symmetric case where $\Delta R_{\text{out}} = \Delta R_{\text{in}}$, the weight $W(x, y)$ can be written as

$$W(x, y) = S_{\text{in}}(x, y) - S_{\text{out}}(x, y) \quad (14)$$

so that the inside and the outside of the R^* ellipse can separately be considered. As a simplification, a modified distance function r will be used, where

$$r = \sqrt{(x_{\text{rot}})^2 + \gamma^2 (y_{\text{rot}})^2} \quad (15)$$

which can be simplified (using basic trigonometry) to

$$r = \sqrt{(x - x_{\text{c}})^2 + \gamma^2 (y - y_{\text{c}})^2}. \quad (16)$$

Thus, $W(x, y)$ becomes

$$W(r) = S_{\text{in}}(r) - S_{\text{out}}(r). \quad (17)$$

Notice that r is never negative. The plot of the functions S_{in} and S_{out} as a function of r is provided in Fig. 6. S_{out} has its largest influence at points whose distance from the center of the ellipse is small. S_{in} has its greatest influence at points whose distance from the center is large. Neither function has much influence within the R^* band. Fig. 7 shows the symmetric $(\Delta R_{\text{in}} = \Delta R_{\text{out}})$ case for the weighting functions. The

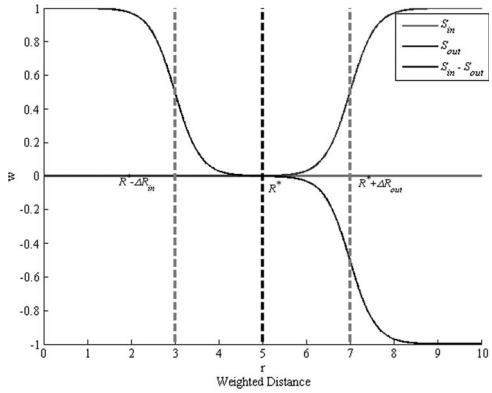

Fig. 7. Weighting function $W(r)$ when $\Delta R_{\text{in}} = \Delta R_{\text{out}}$.

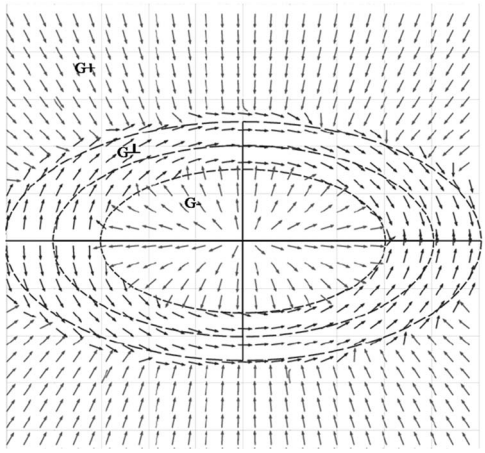

Fig. 8. Vector field with S_{in}, S_{out}, and $N_{\perp}$ limiting functions.

convergence of the S_{in} and S_{out} limiting functions to the R^* band is mathematically shown in the Appendix.

Each of the limiting functions in (12) and (13) contains *tuning parameters* that may be used as *vector field control variables*. These functions include one tuning parameter each, which determines how quickly the function approaches zero. The parameters α_{in} and α_{out} control the slope of $S_{\text{in}}(r)$ and $S_{\text{out}}(r)$ for r in the set $R - \Delta R_{\text{in}} < r < R + \Delta R_{\text{out}}$.

The values of $S_{\text{in}}(R^*)$ and $S_{\text{out}}(R^*)$ can be made arbitrarily small. Let $\varepsilon > 0$ be a small number such that $S_{\text{in}}(R^*) = \varepsilon$ and $S_{\text{out}}(R^*) = \varepsilon$. The values of α_{in} and α_{out} can then be determined. The resulting equations are shown as follows:

$$\alpha_{\text{in}} = \frac{1}{\Delta R_{\text{in}}} \ln\left(\frac{1-\varepsilon}{\varepsilon}\right) \tag{18}$$

$$\alpha_{\text{out}} = \frac{1}{\Delta R_{\text{out}}} \ln\left(\frac{1-\varepsilon}{\varepsilon}\right). \tag{19}$$

Attracting the robot to the R^* neighborhood in (11) is the first step in the construction of the final vector field. Another vector field is needed to control the robots once they are in the elliptical band. In this field, the robots need to move along the ellipse in a field perpendicular to the previously described gradient fields. The influence of these perpendicular fields must be restricted to a narrow band similar to that described by (11). Vectors in this field must die off outside this narrow band. A limiting function that accomplishes this instance (the $N_{\perp}$ limiting function is shown in Fig. 6) is given by

$$N_{\perp}(\alpha_{\perp}, r, R^*) = e^{-\alpha_{\perp}(r-R^*)^2}. \tag{20}$$

In addition, another multiplier to the perpendicular field must be added so that the robots do not circle around the elliptical bands, as shown in Fig. 4(c). For the perpendicular field to change directions, the field that is perpendicular to the gradient is multiplied by a function that changes the direction of the perpendicular field about the x-axis. We have

$$SGN(\alpha_{\perp}, y_{\text{rot}}) = 1 - 2.0\left(\frac{1}{1+e^{-\alpha_{\perp}(y_{\text{rot}})}}\right). \tag{21}$$

Function $N_{\perp}$ in (20) includes one tuning parameter $\alpha_{\perp}$. The parameter $\alpha_{\perp}$ controls the slope of $N_{\perp}(r)$ for r in the set $R - \Delta R_{\text{in}} < r < R + \Delta R_{\text{out}}$. In this case, the parameter will be defined so that the value of $N_{\perp}(R^* + \Delta R_{\text{out}})$ and $N_{\perp}(R^* - \Delta R_{\text{in}})$ can be made arbitrarily small. The resulting formula for $\alpha_{\perp}$ is shown in (22). The same technique is used in the other limiting functions. For the symmetric case ($\Delta R_{\text{in}} = \Delta R_{\text{out}}$), solving for $\alpha_{\perp}$ gives

$$\alpha_{\perp} = \frac{-1}{(\Delta R_{\text{out}})^2} \ln(\varepsilon). \tag{22}$$

The vector field in Fig. 8 is the sum of the three generated vector fields.

Functions S_{in}, S_{out}, and $N_{\perp}$ impose additional restrictions and constraints on top of and in addition to the initial swarm function $f(x, y)$. These limiting functions provide a much tighter level of control by limiting and restricting where the vector fields begin and end. The limiting functions, along with vector fields that were created by the bivariate normal function, may be summed to create swarm movement in formation as a group. When combined, these equations form the velocity and direction of the swarm movement with respect to the center of the swarm as follows:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = (S_{\text{in}} - S_{\text{out}}) \begin{bmatrix} d_x \\ d_y \end{bmatrix} + SGN * N_{\perp} \begin{bmatrix} d_x \\ d_y \end{bmatrix}_{\perp}. \tag{23}$$

C. Obstacle Avoidance and Swarm-Member Dispersion

Vector fields that were weighted with sigmoid functions may be used for obstacle avoidance and for controlling member spacing by creating vectors that move away from the center of the obstacle's or other swarm-member location $(x_{\text{co}}, y_{\text{co}})$. For the purposes of this work, the concern is formation that includes member spacing. In describing the formation control methodology, it is assumed that the only obstacles are other members of the swarm. The same form of limiting function as S_{in} may be used. Obstacle avoidance between members is accomplished using

$$r_{\text{avoid}} = \sqrt{(x - x_{\text{co}})^2 + (y - y_{\text{co}})^2} \tag{24}$$

$$S_{\text{avoid}}(\alpha_{\text{avoid}}, r_{\text{avoid}}, \Delta R_{\text{avoid}}) = \kappa - \frac{\kappa}{1 + e^{\alpha_{\text{avoid}}(\sqrt{r_{\text{avoid}} - \Delta R_{\text{avoid}}})}} \tag{25}$$

$$\begin{bmatrix} d_{x\_\text{avoid}} \\ d_{y\_\text{avoid}} \end{bmatrix} = \begin{bmatrix} S_{\text{avoid}}(x - x_{\text{co}}) \\ S_{\text{avoid}}(y - y_{\text{co}}) \end{bmatrix}. \tag{26}$$

The weight function that was generated by a single obstacle is a sigmoid with maximum value κ as shown in (25). Tuning the parameters κ and α_{avoid} ensures that the collision avoidance field dominates the vector field near an obstacle. Notice that r_{avoid} is similar to r from (16), except that, instead of the distance from the center, the distance to the swarm member is used. The ΔR_{avoid} parameter denotes the minimum distance from other members. This parameter determines the dispersion of swarm members in formation. S_{out} and S_{in} get swarm members to the band but do not control their dispersion.

Avoidance of individual swarm members, including their dispersion, is controlled by the range of influence for the avoidance vector field. The α_{avoid} parameter in (25) controls how quickly vector fields die out near obstacles. As α_{avoid} decreases, the influence range of the avoidance vector field increases. By controlling the α_{avoid} parameter, different types of formations can be made within the ellipse bands. Selection of the R_{in}, R_{out}, and ΔR_{avoid} parameters is discussed in the next section.

The α_{avoid} parameter is solved for in the same way as the other sigmoid limiting functions in (12) and (13). The ΔR_{avoid} parameter specifies the minimum distance between swarm members. Solving for $S_{\text{avoid}}(\Delta R_{\text{avoid}}) = \varepsilon$ gives

$$\alpha_{\text{avoid}} = \frac{1}{\Delta R_{\text{avoid}}} \ln\left(\frac{\kappa - \varepsilon}{\varepsilon}\right). \tag{27}$$

Obstacle avoidance is guaranteed with repulsive potential fields. Choosing α_{avoid} to be close to 0, the limiting functions S_{out} and S_{in} will arbitrarily be small near obstacles so that the vectors v_x and v_y are dominated by the obstacle avoidance term. However, there is no guarantee that all zero-length vectors are eliminated in the potential field associated with static and dynamic obstacles, but at every time step, the vector field changes so these zero-length vectors are temporary.

The S_{avoid} function is used to prevent swarm members from colliding with each other. A combination of all of the aforementioned fields creates a static formation for swarm members, and shifting the center of the ellipse as a function of time creates an overall movement of the swarm as a whole.

The swarm may move from one waypoint to another by moving the center of the ellipse $(x_{\text{c}}, y_{\text{c}})$. The general equation for creating the vector to follow a trajectory with member avoidance by summing the vector fields is given by

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = (S_{\text{in}} - S_{\text{out}}) \begin{bmatrix} d_x \\ d_y \end{bmatrix} + \sum_{1}^{\#\text{size}-1} S_{\text{avoid}} \begin{bmatrix} d_{x\_\text{avoid}} \\ d_{y\_\text{avoid}} \end{bmatrix} + SGN * N_{\perp} \begin{bmatrix} d_x \\ d_y \end{bmatrix}. \tag{28}$$

The computational requirements for an individual swarm member are very low, i.e., $O(n)$. The computational complexity of the vector generation depends on the number of obstacles, because this aspect is the only factor in the equations with the potential to continuously be growing. The complexity will grow in denser environments. This complexity is due to the fact that, at each time step, an avoidance vector for n obstacles and/or robots within a certain range must be generated. Assuming that only nearby swarm members are utilized for the dispersion aspect of the swarm, the computation complexity is very low with large swarm sizes. Note that swarm members do not compute the entire field. They compute a single vector from that field, which depends on the center of the ellipse $(x_{\text{c}}, y_{\text{c}})$, the four vectors (i.e., in, out, perpendicular, and avoidance), and their corresponding weights.

IV. Parameter Selection

To select control parameters, some logic and basic mathematics must be used. The formation must be feasible, given the swarm characteristics. These swarm characteristics might include the number of team members, the desired length of the minor and major axes, and the average or maximum length of the robots. The precision at which these R-values are chosen will determine how tight and accurate the formation will be. The first necessary step is to determine which formation is desired. Note that there is some allowable margin of error when selecting parameters, because the swarm members can lie in the area described by (11). Parameter selection guidelines for different formations are discussed in the following sections.

A. Ellipse Formation

If the desired formation is an ellipse, the vectors are generated as described in (28). The only necessary requirement is that the chosen major and minor axes fit the swarm characteristics. To create a circle formation, an equal minor and major axis must be chosen. In addition, with ellipse and circle formations, the ΔR_{avoid} parameter must allow for equal dispersion along the ellipsoid perimeter to actually create an ellipse or circle figure with the swarm. If not, then the robots will tend toward the front or back of the formation shape. A general estimate of the perimeter of the ellipse can be utilized as a guideline for choosing ΔR_{avoid}. Given R^*, γ, and N, denoting the number swarm members, the ellipse perimeter can be used as the upper bound in estimating the ΔR_{avoid} parameter. ΔR_{avoid} should adhere to the following equation, at the very least, to achieve equal dispersion:

$$\Delta R_{\text{avoid}} \leq \pi \sqrt{2(R^{*2} + \gamma R^{*2}) - (R^{*2} - \gamma R^*)^2/2}/N. \tag{29}$$

In addition, it is also necessary to ensure that ΔR_{avoid} is chosen to be large enough to avoid other swarm members. This factor is highly dependent on the obstacle avoidance sensor that was used. If the size of the swarm is small compared to the circumference of the elliptical ring, the swarm members may not uniformly be distributed if an optimal dispersion factor ΔR_{avoid} is not chosen.

B. Arc/Wedge Formation

If the desired formation is an arc or wedge, the vectors are generated as in (28). The parameters are chosen in the same way as the ellipse formation, but to force the swarm members to the front of the formation, it is necessary to choose R^* and ΔR_{avoid} so that approximately half of the perimeter is empty.

C. Line Formation

The line formation can be done with two different methods. The first method utilizes a "skinny" ellipse, and the second way utilizes a leader–follower approach.

Fig. 9. Skinny ellipse with swarm members trapped inside.

The line formation still uses the ellipse as the basis but with a slight modification. The S_{in} and $N_{\perp}$ limiting functions are removed to trap swarm members, i.e., UGVs, inside a narrow or skinny ellipse, as shown in Fig. 9 and described by

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = (-S_{\text{out}}) \begin{bmatrix} d_x \\ d_y \end{bmatrix} + \sum_{1}^{\#\text{size}-1} S_{\text{avoid}} \begin{bmatrix} d_{x\_\text{avoid}} \\ d_{y\_\text{avoid}} \end{bmatrix}. \quad (30)$$

The length or major axis of the ellipse needs to be long enough to hold all the swarm members, and the width or minor axis needs to be wide enough for the swarm member. If the γ parameter is chosen to be small, the swarm members will have a zigzag pattern as they continually overshoot the desired path. If the γ parameter is chosen to be large, the swarm members will have an offset line pattern. A tighter line formation can also be achieved by combining this method with a hierarchical leader–follower approach, in which each swarm member takes the role of a leader, with the exception of the member with the tail position in the line.

V. Simulation Results

Simulink has been used to model a UGV swarm of ten robots. The swarm formation controller, which is identical for each robot, is programmed in C. Each individual robot's vector generating controller is implemented as a C MEX S-function with the control parameters and a position vector with nearby member locations as inputs. In addition to the Matlab simulations, Unreal Tournament has been used to simulate a mission of convoy protection with a swarm of 40 robots.

Complete knowledge of the other member's location is not a necessity, but to get uniform dispersion about the ellipse, the location of neighboring robots is desired. In these simulations, only swarm-member locations within a specified radius are utilized. Knowledge could be shared in multiple ways, making a case for both centralized and decentralized robotic systems, but this case is not explored in this paper. Swarm members will converge to the elliptical band if reasonable parameters are selected.

Each of the vector generation functions generates the vector fields in Section IV at each time step. Formation changes are easily made by updating just a few tuning parameters. Parameters are selected based on the desired formation and dispersion using the methods in Section IV. The swarm center is broadcast into the vector that generates S-functions. The swarm center $(x_{\text{c}}, y_{\text{c}})$ is left fixed for some of the simulations with ten robots to show the exact formation shape. In the simulation with 40 robots, a moving convoy is utilized as the determining factor of the swarm center and the ellipsoid parameters.

A. Formations With Static Center

To demonstrate the capabilities of this method, simulations are run with ten robots to create ellipse, circle, and arc formations. In these simulations, the robots are initialized to a random position around the center $(x_{\text{c}}, y_{\text{c}})$. The robots eventually surround the center in an ellipse formation with the parameters in Table II. Notice that the γ parameter makes the y-axis "skinnier."

TABLE II
Parameters for Different Formations

Formation	R*	κ	γ	ΔR_{in}	ΔR_{out}	ΔR_{avoid}	ε
Ellipse	24	1	0.5	2	2	4.4	0.001
Circle	12	1	1	2	2	3	0.001
Arc	100	1	1.0	2	2	6	0.001

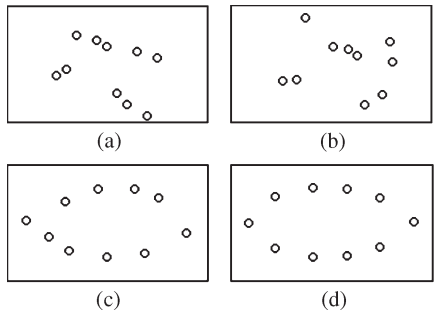

Fig. 10. Snapshot of a swarm of ten robots that enter an ellipse formation. (a) $t = 0$ s. (b) $t = 15$ s. (c) $t = 30$ s. (d) $t = 45$ s.

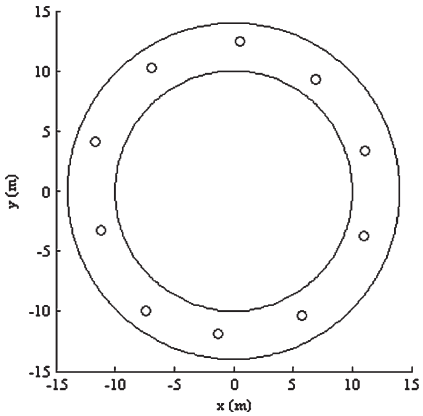

Fig. 11. Final positions of a swarm of ten robots that enter a circle formation.

The $N_{\perp}$ field is utilized until the robots are equally dispersed, and then, it is turned off to leave the robots in a static formation. Snapshots of the robots that enter this formation are shown in Fig. 10.

Fig. 11 depicts the final positions of ten robots on the circular ring described by the parameters in Table II. Fig. 12 shows snapshots of an arc formation for different time slices by utilizing the parameters in Table II. We want the robots to be forced to half the ellipse; thus, the $N_{\perp}$ and SGN fields are utilized until the robots are the desired distance from their neighboring swarm members.

B. Moving Formation With the Influx and Outflux of Swarm Members

To show how this formation control method can easily handle the automatic influx and outflux of team members, a simulation is run, in which members are added and taken away at predetermined times. The swarm center $(x_{\text{c}}, y_{\text{c}})$ changes as a function of time for this experiment, guiding the swarm and controlling the navigation trajectory. At time t_3, five agents are added to a swarm of five. The formation parameters are dynamically changed based on these new vehicles. The formation

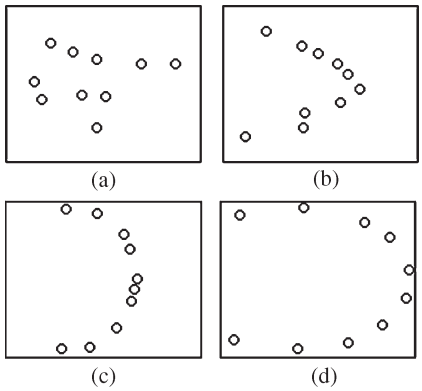

Fig. 12. Snapshot of a swarm of ten robots that enter an ellipse formation. (a) $t = 0$ s. (b) $t = 15$ s. (c) $t = 30$ s. (d) $t = 45$ s.

TABLE III
PARAMETERS FOR INFLUX AND OUTFLUX

Formation	R*	κ	γ	ΔR_{in}	ΔR_{out}	ΔR_{avoid}	ε
Before In-Flux	12	1	0.5	2	2	4.4	0.001
After In-Flux	24	1	0.5	2	2	4.4	0.001

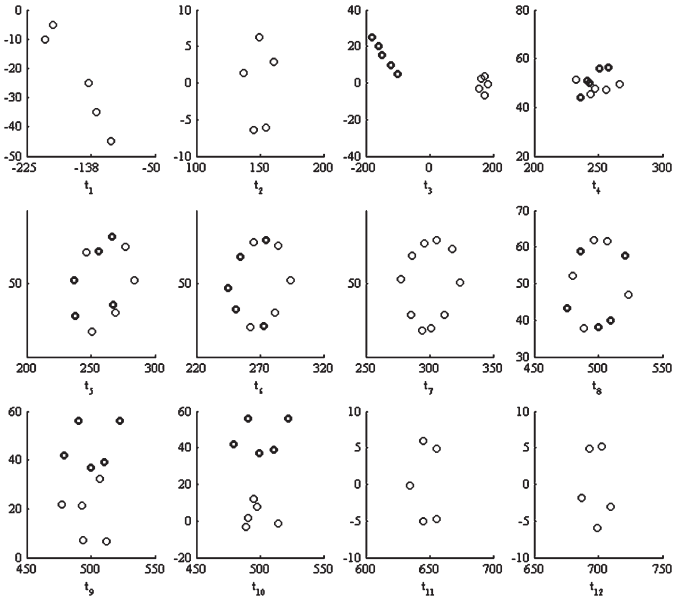

Fig. 13. Snapshot of a swarm of ten robots that enter an ellipse formation with the sudden influx of five team members at t_3 and failure at t_8. The darkened circles are the added and failed members.

parameters that were chosen are shown in Table III for before and after the adding of swarm members. To show the reverse concept, the same five robots are removed from the swarm at t_8. When the other swarm members realize that the failed members have not moved or they have stopped communicating for a predefined number of seconds, then they dynamically change their formation back to the original parameters before the influx. Fig. 13 depicts the influx and outflux of swarm members at different time steps.

C. Application to Convoy Protection

Unreal Tournament is utilized to simulate the real-world problem of convoy protection. In this simulation, a convoy of three vehicles is given a set of waypoints on a road, and a swarm of 40 robots is utilized to surround this convoy as it travels.

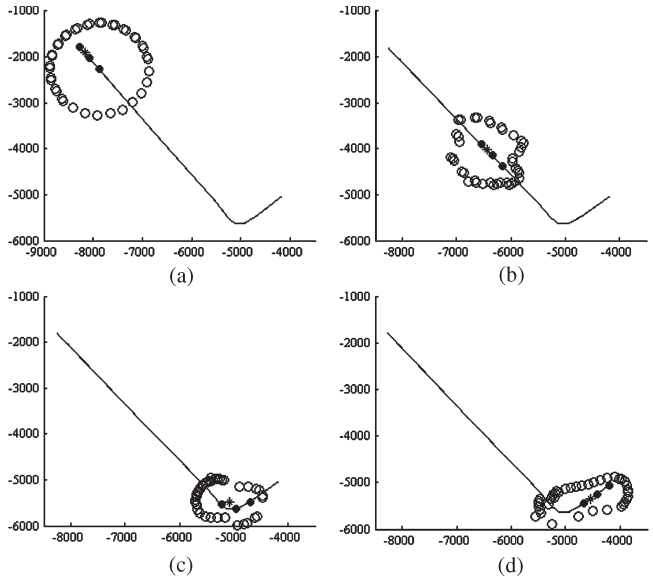

Fig. 14. Snapshot of a swarm of 40 robots that travel and surround a convoy of vehicles in formation. (a) $t_1 = 1$. (b) $t_2 = 135$. (c) $t_3 = 225$. (d) $t_4 = 260$.

The formation dynamically changes as the convoy travels along the road.

The shape of the elliptical formation is determined by the information that was provided by the convoy of vehicles that travel on the road. They send the swarm parameters that describe an ellipse that encloses the convoy—the parameters are the center of the ellipse, the orientation, and the length of the major and minor axes. As the convoy turns around the corner, the convoy trucks bunch up, causing the ellipse to become circular. In turn, the swarm redistributes as the elliptical ring becomes circular. This instance illustrates that the proposed approach can easily adapt to different circumstances.

Fig. 14 shows the swarm formation around the convoy at different time slices. The line is the convoy's path of travel, and the darkened circles represent the convoy vehicles. Note that the formation widens and narrows when necessary. This condition can be noted when the convoy goes around the turn in the road. Fig. 15 shows that the parameter values change over time.

VI. FIELD EXPERIMENTS

Experiments have been performed using Traxxas Emaxx RC-cars that were equipped with a custom-built computer system as shown in Fig. 16. The Emaxx vehicles are Ackerman steered, and each vehicle is equipped with an inertial measurement unit and global positioning system (GPS).

The UGVs are controlled via two servo commands: 1) one command for controlling the speed and 2) the other command for controlling the heading. Although each vehicle appears identical, each speed controller is slightly different and was manually tuned. The vector generation code is identical on every robot. The generated vector from (22) is translated into two robot servo commands.

One simple broadcast communication model that was programmed in C is used for information relay and exchange. Each robot shares its position with the other robots only for obstacle avoidance, because the robots currently have no other purely reactive sensing capabilities to avoid collision. The robots were all programmed in C.

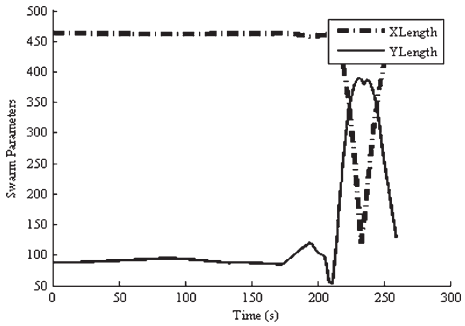

Fig. 15. Swarm-formation parameters change as the convoy travels.

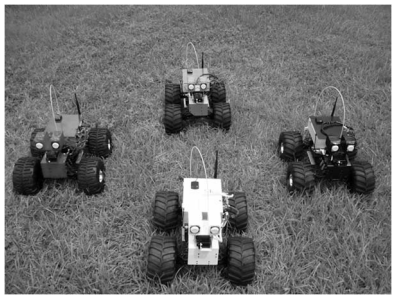

Fig. 16. UGV robots.

Three different sets of field experiments have been performed. The test field is approximately 70 m in width and 100 m in length. Experiments were performed with three and four UGVs. The first experiment demonstrates four UGVs that follow an ellipse formation. The second experiment demonstrates a line formation that utilizes the leader–follower approach and three UGVs. At each time step, the robots compute their vectors based on the current position, the current center, and other swarm-member locations. Based on the output of the vector fields, a desired speed and a desired heading are computed. For all experiments, with the time being set in seconds, coordinates are in Universal Trans Mercators, and distance measurements are set in meters.

In all of the experiments, the swarm center $(x_{\rm c}, y_{\rm c})$ controls the swarm's navigation trajectory. Note that, at any instant in time, the geometric center of the swarm may not coincide with the center of the ellipse. Possible causes for this disparity are explained as follows: 1) the original position of the swarm members and the speed of travel may cause the swarm to lag behind; 2) a significant change in the trajectory's direction may also cause swarm members to be out of place; and 3) obstacles, particularly large static obstacles, will also cause deviations from the desired formation. All of these issues can cause temporary differences between the center of the swarm and the navigation trajectory.

A. Ellipse Formation

In Experiment 1, four UGV vehicles traveled in an ellipse formation that surrounds the swarm center. A GPS coordinate is used for the swarm center $(x_{\rm c}, y_{\rm c})$. This coordinate is a function of time, which all robots periodically receive. The four UGVs travel, surrounding each center point and staying at a minimum specified distance from one another. Table IV shows the control parameters that were used for this experiment. The units for R^*, $\Delta R_{\rm in}$, $\Delta R_{\rm out}$, and $\Delta R_{\rm avoid}$ are all set in meters.

TABLE IV
PARAMETERS FOR FIELD EXPERIMENTS

Formation	R^*	κ	γ	ΔR_{in}	ΔR_{out}	ΔR_{avoid}	ε
Ellipse	7	1	1	4	4	5	0.001
Line	3	1	1	1	1	5	0.001

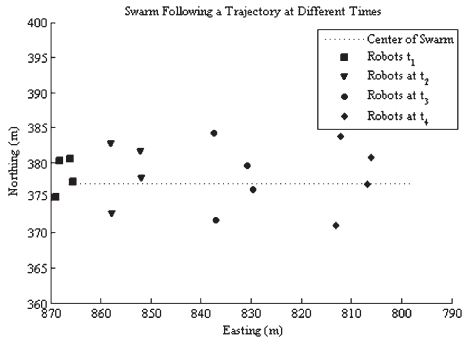

Fig. 17. Ellipse robot formation at t_1, t_2, t_3, and t_4.

Fig. 17 shows the swarm formation at four different times during the trajectory traversal. The swarm members were started at random places at the beginning of the mission and moved into formation over time. The robots continued to hold formation throughout the traversal.

In this experiment four UGV vehicles travel in an ellipse formation surrounding the virtual swarm center. A GPS coordinate is used for the swarm center. This coordinate is a function of time which all robots receive periodically. The four UGVs travel surrounding each center point and staying at a minimum specified distance away from one another. A video that demonstrates the first experiment can be found at http://ieeexplore.ieee.org.

B. Line Formation

In this experiment, three UGV vehicles travel in a line formation. The leader–follower method is used. One of these UGVs, i.e., the alpha robot, follows a formation that utilizes a virtual formation center $(x_{\rm c}, y_{\rm c})$. The alpha robot is at the top of the line hierarchy. The next UGV, i.e., beta robot 1, follows the alpha UGV that stays at a minimum specified distance away. The next UGV, i.e., beta robot 2, follows beta robot 1. Table IV shows the control parameters that were used for the alpha robot in this experiment. Fig. 18 shows each swarm member's distance from the other swarm members over time. The robots are evenly distributed approximately 10 m apart in a line formation.

Fig. 19 shows the robot paths and the centers with respect to time. On the legend, Robot 1 is the path of the alpha robot, Robot 2 is the path of beta robot 1, and Robot 3 is the path of beta robot 2.

C. Observations

The results demonstrate that the proposed method can successfully be utilized to control robot swarm formation. The potential functions, together with the limiting functions, present an elegant way of controlling a swarm, but there are potential failure conditions that are important to mention.

There are two potential shortcomings with this method: 1) the ability of small teams to maintain formations and 2) the ability of the swarm to maintain formation as it moves. This method

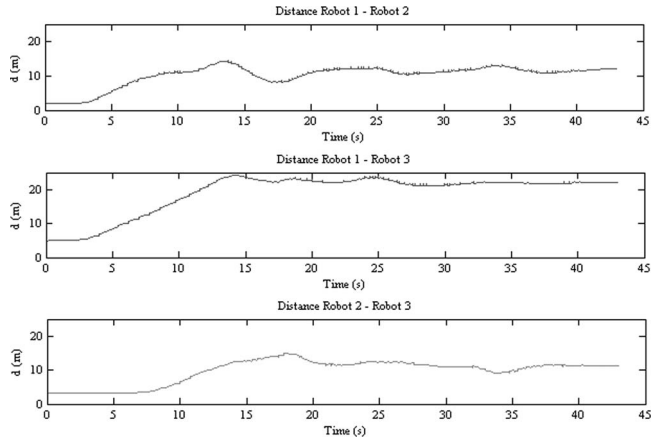

Fig. 18. Distance between swarm members.

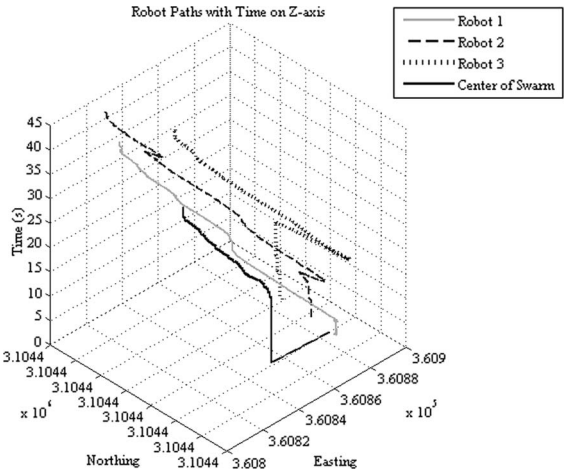

Fig. 19. Robot paths with respect to center (x_c, y_c) with time on the z-axis.

attracts swarm members to a formation that was described by a contour ring around the underlying surface. By assuming that swarm members can sense nearby swarm members, the team can maintain a minimum distance between each other without requiring global location knowledge of all the other swarm members. If the area that was covered by the formation is large compared to the area that was covered by the swarm, the swarm members need to know the location of each member to maintain a maximum distance between neighboring swarm members. Without this knowledge, formations will develop holes as members are lost. Note that this case is not a problem for large teams/swarms—the vector field itself will fill in the holes as members are lost.

In a dynamic environment, maintaining the formation depends on the ability of the individual robots to achieve the appropriate speed. Speed is a problem—robots must move fast enough to keep up with the formation. This condition is particularly true for swarm members that get separated from the main body as they encounter obstacles or other delays. As the formation moves through the environment, the control is smoothest if the direction that the formation moves is known to the team members in advance. If this information is not known, the control will be relatively smooth for team members that are "pulled" by the underlying surface, because the vector that attracts the team to the formation is close to the actual formation's direction. On the other hand, for team members being "pushed" by the underlying surface, the formation vector pulls the team members from the formation, leading to instability. This instability can be overcome by supplying the formation vector to each team member or by developing formations that use only a portion of the contour. In our newest work, we design surfaces to "pull" the desired formations through the environment.

VII. Conclusion and Future Work

In this paper, a methodology for attracting members of a swarm or other multiagent system into a formation has been presented. Potential functions, together with limiting functions, can successfully be utilized to control robot swarm formation, obstacle avoidance, and the overall swarm movement. The presented method has supported scalability, different swarm sizes, multiple formations, heterogeneous swarm-member teams, and centralized and decentralized formation control.

The advantage of this approach, compared with other approaches, is the simplicity of the functions and the ease of formation changes. There is no expensive tuning involved, and little information is required for each robot to adhere to formation constraints.

Our future work includes creating formations with shaper corners by changing the underlying surface that generated the vector field. The method will also be extended to include more complex shapes by combining multiple surfaces. In addition, a fuzzy logic tuner for the control variables for the swarm functions will be developed. The approach will also be expanded to the 3-D case for unmanned aerial systems.

Appendix
Mathematical Proof for the Convergence of S_{in} and S_{out} Limiting Functions

This proof is a simple stability analysis for the proposed system to show that, regardless of the initial location of the robots, they eventually reach the desired elliptical band. Note that this proof only shows the convergence of swarm robots to elliptical bands and not the proof of distribution. Methods in Section IV describe how parameters are chosen to acquire desired distribution. Recalling (17), $W(r)$ can be written as

$$W(r) = \frac{e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))}}{1+e^{\alpha_{\text{in}}(r-(R*-\Delta R_{\text{in}}))}} - \frac{1}{1+e^{\alpha_{\text{out}}(r-(R^*+\Delta R_{\text{out}}))}}. \tag{31}$$

$S_{\text{in}}(R^*)=\varepsilon$ and $S_{\text{out}}(R^*)=\varepsilon$; thus, $W(R^*)=0$. Furthermore, it can be shown that $W(r)>0$ for $r<R^*$ and $W(r)<0$ for $r>R^*$. Equation (32) holds, shown at the bottom of the page.

$$W(r) = \frac{e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))}\left(1+e^{\alpha_{\text{out}}(r-(R^*+\Delta R_{\text{out}}))}\right) - \left(1+e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))}\right)}{\left(1+e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))}\right)\left(1+e^{\alpha_{\text{out}}(r-(R^*+\Delta R_{\text{out}}))}\right)} \tag{32}$$

The denominator is always positive; thus, the sign is controlled by the numerator. Simplifying the numerator results in

$$= e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))}\left(1+e^{\alpha_{\text{out}}(r-(R^*+\Delta R_{\text{out}}))}\right) - \left(1+e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))}\right) \quad (33)$$

$$= e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))} + e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))}e^{\alpha_{\text{out}}(r-(R^*+\Delta R_{\text{out}}))} - 1 - e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))} \quad (34)$$

$$= e^{\alpha_{\text{in}}(r-(R*-\Delta R_{\text{in}}))}e^{\alpha_{\text{out}}(r-(R^*+\Delta R_{\text{out}}))} - 1 \quad (35)$$

$$= e^{\alpha_{\text{in}}(r-(R^*-\Delta R_{\text{in}}))+\alpha_{\text{out}}(r-(R^*+\Delta R_{\text{out}}))} - 1. \quad (36)$$

$e^0 = 1$; thus, the simplified numerator [and $W(R)$] will be positive if the argument in the exponent is positive. Similarly, $W(R)$ will be negative if the argument in the exponent is negative. Substituting from (18) and (19) into the expression of the exponent, we have

$$= \alpha_{\text{in}}\left(r-(R^*-\Delta R_{\text{in}})\right) + \alpha_{\text{out}}\left(r-(R^*+\Delta R_{\text{out}})\right) \quad (37)$$

$$= \frac{1}{\Delta R_{\text{in}}}\ln\left(\frac{\varepsilon}{(1-\varepsilon)}\right)\left(r-(R^*-\Delta R_{\text{in}})\right) + \frac{1}{\Delta R_{\text{out}}}\ln\left(\frac{\varepsilon}{1-\varepsilon}\right)\left(r-(R^*+\Delta R_{\text{out}})\right) \quad (38)$$

$$= \ln\left(\frac{\varepsilon}{(1-\varepsilon)}\right)\left(\left(\frac{1}{\Delta R_{\text{in}}}+\frac{1}{\Delta R_{\text{out}}}\right)(r-R^*) + \frac{\Delta R_{\text{in}}}{\Delta R_{\text{in}}} - \frac{\Delta R_{\text{out}}}{\Delta R_{\text{out}}}\right) \quad (39)$$

$$= \ln\left(\frac{\varepsilon}{(1-\varepsilon)}\right)\left(\frac{1}{\Delta R_{\text{in}}}+\frac{1}{\Delta R_{\text{out}}}\right)(r-R^*). \quad (40)$$

The expression in (40) is positive for $r > R^*$ and negative for $r < R^*$.

REFERENCES

[1] W. Ren and R. W. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *J. Guid. Control Dyn.*, vol. 27, no. 1, pp. 73–82, Jan./Feb. 2004.

[2] K.-H. Tan and M. A. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1996, pp. 132–139.

[3] M. A. Lewis and K. H. Tan, "High precision formation control of mobile robots using virtual structures," *Auton. Robots*, vol. 4, no. 4, pp. 387–403, Oct. 1997.

[4] A. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Trans. Control Syst. Technol.*, vol. 9, no. 6, pp. 777–790, Nov. 2001.

[5] S. S. Ge and C. H. Fua, "Queues and artificial potential trenches for multirobot formations," *IEEE Trans. Robot. Autom.*, vol. 21, no. 4, pp. 646–656, Aug. 2005.

[6] C.-H. Fua, S. S. Ge, K. D. Do, and K. W. Lim, "Multirobot formations based on the queue-formation scheme with limited communications," in *IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2385–2390.

[7] C.-H. Fua, S. S. Ge, K. D. Do, and K. W. Lim, "Multirobot formations based on the queue-formation scheme with limited communication," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1160–1169, Dec. 2007.

[8] T. Balch and M. Hybinette, "Social potentials for scalable multirobot formations," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2000, pp. 73–80.

[9] G. H. Elkaim and R. J. Kelbley, "A lightweight formation control methodology for a swarm of nonholonomic vehicles," in *IEEE Aerospace Conf.*, 2006, pp. 1–8.

[10] V. Gazi, "Swarm aggregations using artificial potentials and sliding-mode control," *IEEE Trans. Robot. Autom.*, vol. 21, no. 6, pp. 1208–1214, Dec. 2005.

[11] V. Gazi, B. Fidan, Y. S. Hanay, and M. Ý. Köksal, "Aggregation, foraging, and formation control of swarms with nonholonomic agents using potential functions and sliding mode techniques," *Turk. J. Elect. Eng. Comput. Sci.*, vol. 15, no. 2, pp. 149–168, Jul. 2007.

[12] S. S. Ge, C.-H. Fua, and W.-M. Liew, "Swarm formations using the general formation potential function," in *IEEE Conf. Robot., Autom. Mechatronics*, 2004, pp. 655–660.

[13] D. H. Kim, H. O. Wang, Y. Guohua, and S. Seiichi, "Decentralized control of autonomous swarm systems using artificial potential functions: Analytical design guidelines," in *43rd IEEE Conf. Decision Control*, 2004, pp. 159–164.

[14] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *15th IFAC World Congr.*, Jul. 2002, pp. 1–7.

[15] J. Yao, R. Ordonez, and V. Gazi, "Swarm tracking using artificial potentials and sliding mode control," in *45th IEEE Conf. Decision Control*, 2006, pp. 4670–4675.

[16] L. Chaimowicz, N. Michael, and V. Kumar, "Controlling swarms of robots using interpolated implicit functions," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 2487–2492.

[17] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. IEEE Conf. Decision Control*, 2001, pp. 2968–2973.

[18] S. Zilinski, T. Koo, and S. Sastry, "Optimization-based formation reconfiguration planning for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 3758–3763.

[19] R. O. Saber, W. B. Dunbar, and R. M. Murray, "Cooperative control of multivehicle systems using cost graphs and optimization," in *Proc. Amer. Control Conf.*, 2003, pp. 2217–2222.

[20] V. Kumar and L. Chaimowicz, "Coordination among UAVs and swarm robots," in *Int. Symp. Distrib. Auton. Robot. Syst.*, 2004, pp. 243–252.

[21] W. Kowalczyk, "Target assignment strategy for scattered robots building formation," in *Proc. 3rd Int. Workshop Robot Motion Control*, 2002, pp. 181–185.

[22] M. Egerstedt and X. Hu, "Formation constrained multiagent control," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 947–951, Dec. 2001.

[23] Z. Cao, M. Tan, S. Wang, Y. Fan, and M. Zhang, "The optimization research of formation control for multiple mobile robots," in *Proc. World Congr. Intell. Control Autom.*, 2003, pp. 1270–1274.

[24] C. Belta and V. Kumar, "Trajectory design for formations of robots by kinetic energy shaping," in *IEEE Int. Conf. Robot. Autom.*, 2002, pp. 2593–2598.

[25] T. J. Koo and S. M. Shahruz, "Formation of a group of unmanned aerial vehicles (UAVs)," in *Proc. Amer. Control Conf.*, 2001, pp. 69–74.

[26] K. Fujibayashi, S. Murata, K. Sugawara, and M. Yamamura, "Self-organizing formation algorithm for active elements," in *Proc. IEEE Symp. Reliable Distrib. Syst.*, 2002, pp. 416–421.

[27] A. Jadbabaie, L. Jie, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988–1001, Jun. 2003.

[28] M. A. Hsieh, S. Loizou, and V. Kumar, "Stabilization of multiple robots on stable orbits via local sensing," in *IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2312–2317.

[29] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. Robot. Autom.*, vol. 19, no. 6, pp. 933–941, Dec. 2003.

[30] S. Monteiro and E. Bicho, "A dynamical systems approach to behavior-based formation control," in *IEEE Int. Conf. Robot. Autom.*, 2002, pp. 2606–2611.

[31] G. L. Mariottini, F. Morbidi, D. Prattichizzo, G. J. Pappas, and K. Daniilidis, "Leader−follower formations: Uncalibrated vision-based localization and control," in *IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2403–2408.

[32] J. Shao, G. Xie, and L. Wang, "Leader-following formation control of multiple mobile vehicles," *IET Control Theory Appl.*, vol. 1, no. 2, pp. 545–552, Mar. 2007.

[33] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 905–908, Dec. 2001.

[34] J. Fredslund and M. J. Mataric, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 837–846, Oct. 2002.

[35] M. Sisto and D. Gu, "A fuzzy leader−follower approach to formation control of multiple mobile robots," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 2515–2520.

[36] Y. Li and X. Chen, "Leader-formation navigation with sensor constraints," in *IEEE Int. Conf. Inf. Acquisition*, 2005, pp. 1–6.

[37] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "A geometric characterization of leader−follower formation control," in *IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2397–2402.

[38] H. Takahashi, H. Nishi, and K. Ohnishi, "Autonomous decentralized control for formation of multiple mobile robots considering ability of

robot," *IEEE Trans. Ind. Electron.*, vol. 51, no. 6, pp. 1272–1279, Dec. 2004.
[39] J. P. Desai, "A graph theoretic approach for modeling mobile robot team formations," *J. Robot. Syst.*, vol. 19, no. 11, pp. 511–525, Nov. 2002.
[40] R. Fierro and A. K. Das, "A modular architecture for formation control," in *Proc. 3rd Int. Workshop Robot Motion Control*, 2002, pp. 285–290.
[41] R. Olfati-Saber and R. M. Murray, "Graph rigidity and distributed formation stabilization of multivehicle systems," in *Proc. 41st IEEE Conf. Decision Control*, 2002, pp. 2965–2971.
[42] Y. Li and X. Chen, "Stability on multirobot formation with dynamic interaction topologies," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 394–399.
[43] X. Chen and Y. Li, "Stability on adaptive nN formation control with variant formation patterns and interaction topologies," *Int. J. Adv. Robot. Syst.*, vol. 5, no. 1, pp. 69–82, 2008.
[44] M. Lindhe, P. Ogren, and K. H. Johansson, "Flocking with obstacle avoidance: A new distributed coordination algorithm based on Voronoi partitions," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 1785–1790.
[45] K. Sugihara and I. Suzuki, "Distributed algorithms for formation of geometric patterns with many mobile robots," *J. Robot Syst.*, vol. 13, no. 3, pp. 127–139, Mar. 1996.
[46] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Trans. Robot. Autom.*, vol. 18, no. 5, pp. 813–825, Oct. 2002.
[47] W. B. Dunbar and R. M. Murray, "Model predictive control of coordinated multivehicle formations," in *Proc. 41st IEEE Conf. Decision Control*, 2002, pp. 4631–4636.
[48] F. Kobayashi, N. Tomita, and F. Kojima, "Reformation of mobile robots using genetic algorithm and reinforcement learning," in *SICE Annu. Conf.*, 2003, pp. 2902–2907.
[49] K. Hirota, T. Kuwabara, K. Ishida, A. Miyanohara, H. Ohdachi, T. Ohsawa, W. Takeuchi, N. Yubazaki, and M. Ohtani, "Robots moving in formation by using neural network and radial basis functions," in *IEEE Int. Joint Conf. Fuzzy Syst. Fuzzy Eng. Symp.*, 1995, pp. 91–94.
[50] F. Michaud, D. Letourneau, M. Guilbert, and J. M. Valin, "Dynamic robot formations using directional visual perception," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2002, pp. 2740–2745.
[51] R. Vidal, O. Shakernia, and S. Sastry, "Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation," in *IEEE Int. Conf. Robot. Autom.*, 2003, pp. 584–589.
[52] S. Spry and J. K. Hedrick, "Formation control using generalized coordinates," in *43rd IEEE Conf. Decision Control*, 2004, pp. 2441–2446.
[53] C.-Y. Mai and F.-L. Lian, "Analysis of formation control and networking pattern in multirobot systems: A hexagonal formation example," *Int. J. Syst., Control Commun.*, vol. 1, no. 1, pp. 98–123, Jul. 2008.
[54] L. Barnes, W. Alvis, M. Fields, K. Valavanis, and W. Moreno, "Heterogeneous swarm formation control using bivariate normal functions to generate potential fields," in *IEEE Workshop Distrib. Intell. Syst.: Collective Intell. Appl.*, 2006, pp. 85–94.
[55] L. Barnes, M. A. Fields, and K. Valavanis, "Unmanned ground vehicle swarm formation control using potential fields," in *15th Mediterranean Conf. Control Autom.*, 2007, pp. 1–8.

Laura E. Barnes (M'03) received the B.S. degree in computer science from Texas Tech University, Lubbock, in 2003 and the M.S. and Ph.D. degrees in computer science and engineering from the University of South Florida, Tampa, in 2007 and 2008, respectively.

She was a Research Assistant with the Center for Robot-Assisted Search and Rescue from 2003 to 2005 and with the Unmanned Systems Laboratory, University of South Florida, from 2005 to 2008. In addition, she held a fellowship from the Army Research Laboratory, where she spent two summers with the Vehicles Technology Directorate. In June 2008, she joined the Automation and Robotics Research Institute, University of Texas, Arlington, as a Faculty Associate Researcher. Her research interests include swarms, unmanned aerial vehicles, and distributed/intelligent systems.

Mary Anne Fields received the Ph.D. degree in mathematics from Clemson University, Clemson, SC, in 1989.

In 1989, she joined the U.S. Army Research Laboratory, Adelphi, MD, where she is currently the Team Leader for the Intelligent Control Team, Unmanned Systems Division, Vehicles Technologies Directorate (she has recently transferred from the Weapons and Materials Directorate). Her research interests include planning for uncertain environments, tactical behavior algorithms for teams of robots, and the use of simulation tools and virtual environments for both the development and testing of robotic systems.

Kimon P. Valavanis (SM'91) received the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, in 1986.

He was a Faculty Member with the Northeastern University, Boston, MA, where he was the Analog Devices Career Development Chair for Assistant Professors from 1986 to 1990. From 1991 to 1999, he was with the University of Louisiana, Lafayette, where he was the Regents Professor in Manufacturing from 1995 to 1999 and the A-CM Associate Director for Research. From 1999 to 2003, he was with the Technical University of Crete, Chania, Greece, as the Director of the Graduate Program and the Chair of the University Industrial Advisory Board. From 2003 to August 2008, he was a Professor with the Department of Computer Science and Engineering, University of South Florida, Tampa, where he also served as the Deputy Director with the Center for Robot-Assisted Search and Rescue until the summer of 2005. In 2006, he established the Unmanned Systems Laboratory, College of Engineering, in which he served as the Director. He was also the Managing Director of the National Institute for Applied Computational Intelligence and a Faculty Associate with the Center for Urban Transportation Research. He is currently with the Department of Electrical and Computer Engineering, University of Denver, Denver, CO. He is also a Guest Professor with the Faculty of Electrical Engineering and Computing, Department of Telecommunications, University of Zagreb, Zagreb, Croatia. He has published more than 250 book chapters, technical journal/transactions, and conference proceedings, and is a coauthor of three books. His research interests include intelligent systems, robotics, and manufacturing, with emphasis, during the last few years, on control and coordination of unmanned systems.

Dr. Valavanis has been the Editor-in-Chief of the IEEE Robotics Automation magazine since 1996. He was a General Cochair (with F. Lewis) of the 11th Mediterranean Conference on Control and Automation in 2003 and the Program Chair of the 2004 IEEE International Conference on Robotics and Automation. He was also the General Chair of the 2007 Mediterranean Conference on Control and Automation. In 1998, he was elected the Vice President of Administration of the IEEE Mediterranean Control Association. He is a former Fulbright Scholar (Senior Lecturing/Research Award) and has been a Distinguished Speaker for the IEEE Robotics and Automation Society (since 2003).