Proceedings of the 15th Mediterranean Conference on
Control & Automation, July 27 - 29, 2007, Athens - Greece

Unmanned Ground Vehicle Swarm Formation Control Using Potential Fields

Laura Barnes*, MaryAnne Fields** and Kimon Valavanis*
* University of South Florida/Department of Computer Science and Engineering, Tampa, FL USA
** US Army Research Lab Aberdeen, MD USA

***Abstract*—A novel technique is presented for organizing swarms of robots into formation utilizing artificial potential fields generated from normal and sigmoid functions. These functions construct the surface swarm members travel on, controlling the overall swarm geometry and the individual member spacing. Limiting functions are defined to provide tighter swarm control by modifying and adjusting a set of control variables forcing the swarm to behave according to set constraints, formation and member spacing. The swarm function and limiting functions are combined to control swarm formation, orientation, and swarm movement as a whole. Parameters are chosen based on desired formation as well as user defined constraints. This approach compared to others, is simple, computationally efficient, 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 four and ten particles following circle, ellipse and wedge formations. Experimental results are also included with four unmanned ground vehicles (UGV).**

***Keywords* – swarms, formation control**

I. INTRODUCTION

Artificial potential fields generated from normal and sigmoid functions are derived to construct surfaces swarm members travel on, controlling overall swarm geometry and individual member spacing. Limiting functions are also defined to provide tighter swarm control by modifying/adjusting a set of control variables forcing swarms to behave according to set constraints, formation and member spacing.

An interesting applicable area for the proposed approach is convoy protection where a swarm of robots assigned to protect a convoy of vehicles requires a specific formation around the vehicles. Thus, given a convoy of assets to protect, determining the configuration of the swarm and arranging the support vehicles around the convoy poses a particularly interesting problem.

The proposed solution is based on potential fields generated by sigmoid and normal functions that are used to control the swarm geometry and the inter-member spacing. These functions are used to create the vector fields that control the velocity and heading of robot swarms, with the swarm located on the surface created by the function.

This work is an extension and improvement of previous research reported in [4, 3]. In [4, 3] bivariate normal functions were used to control the swarm geometry, but the number of generated potential fields was large and parameters were chosen heuristically. In this work, the number of generated fields is decreased and the parameter selection method is refined. In [4, 3], the work presented used a set of ten heterogeneous robot models to simulate the approach. The results were similar to that in this paper, but the method has been improved by reducing parameters needed to hold formation as well as some selection guidelines for choosing parameters.

Further, potential fields are not only used in the classical sense for path planning [19], but also to hold robots in a specified formation while in motion. Fields generate desired velocities and headings to maintain a formation and a specified distance from a neighbor. Potential field methods do have the local minima problem [23]; however, in this case, vector fields are dynamically changing at each time step so the chances of hitting local minima are greatly reduced. Even if a local minima is hit, the next time the field changes (at the next time step), the robot escapes this minimum.

The advantage and difference of this approach compared to similar ones is the simplicity of the vector generation. The control parameters are easily modified and adjusted on the fly to change formation and relative distance between swarm members, while other methods are rigid in formation constraints [22]. Since the vector field dynamically changes over time, it can easily adjust to new information gained by the swarm itself or provided from an external controller.

This presented method lends itself to both centralized and decentralized models. To adhere to formation constraints, global knowledge is not a requirement. Depending on the mission, robots may communicate between themselves on an as needed basis. The method is flexible in formation geometry and adaptable to varying team sizes. The only requirement to hold formation constraints is the detection of the center of the convoy by each swarm member.

Existing methods depend on a central controller [39, 29, 26, 24, 11, 5], but dependence on a single entity is prone to failure and it is not ideal for missions where there is a high possibility of failure. Decentralized formation control methods are presented in [6, 13, 14, 21, 36, 37, 38]. The methods in [22, 14] suffer from high computation complexity. Additional approaches include graph theoretic ones [9, 12, 18, 31], neighbor and leader-following methods [7, 11, 20, 39], vision-based methods [8, 28], behavior-based methods [1, 18] and potential

functions [2, 10, 15, 16, 17, 25, 27, 30, 33, 32, 34]. In [35] physics properties of liquids, solids, and gases are used to maintain swarm formation.

The problem formulation is discussed in Section III. Section IV presents the vector generation with simulations and real-time results shown in Section V. Conclusions and future work are presented in Section VI.

II. PROBLEM FORMULATION

A. Swarm Surface

At any instant in time, the robots can be visualized as particles moving in a potential field generated from a bivariate normal "hill" that controls the velocity and heading of the swarm members. A bivariate normal function with form given in (1):

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

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) affecting the eccentricity of the swarm. Note that the center *(x_c, y_c) and the eccentricity, γ, could be functions of time allowing the swarm to move along a path.*

The x and y partial derivatives create the velocity vectors that are used to determine the heading and velocity of each member of the swarm as shown in (2):

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

Just as with a single robot, the swarm formation, treated as a single shape, 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 (x_c, y_c) must be found; the rotated coordinates for (x, y) and (x_c, y_c) can be found using (3):

$$\begin{aligned} x_{rot} &= \cos(\phi)(x-x_c) - \sin(\phi)(y-y_c) \\ y_{rot} &= \sin(\phi)(x-x_c) + \cos(\phi)(y-y_c) \end{aligned} \tag{3}$$

The rotated coordinates are then substituted back into find d_x and d_y.

B. Formation Problem

In order to describe the general formation problem, it will be discussed in reference to convoy protection. Suppose that a swarm of robots needs to accompany a convoy of vehicles by surrounding them in a particular formation. A field needs to be designed to attract the 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 away to allow the convoy to move safely.

Suppose the positions of each 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 the center *(x_c, y_c)*. Fig. 1 depicts an ellipse with center *(x_c, y_c)*, semi-major axis *2A*, and semi-minor axis *2B*, surrounding a convoy of vehicles. By attracting swarm members to a specific elliptical ring described by (4):

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

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

The general form of the swarm controller is described by (5):

$$\vec{V}(x,y,t) = \sum_{i=1}^{N} w_i(x,y,t)\vec{V}_i(x,y,t) \tag{5}$$

where *V(x,y,t)* gives the velocity of the swarm at a particular time and place. In general, the field *V(x,y,t)* is the weighted sum of N different vectors, each of which is acting on the swarm.

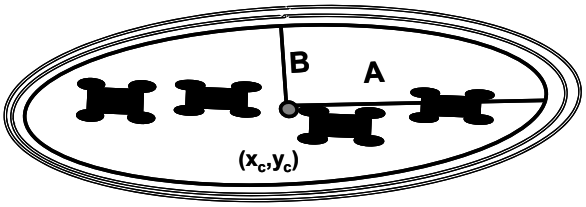

Figure 1. Convoy of vehicles surrounded by ellipses

The challenge is to develop a potential field based controller using a small number of physically relevant weights, w_i, and vectors v_i that attracts particles to a neighborhood of the R* ellipse. This neighborhood is shown in Fig. 2. The variables ΔR_{in} and ΔR_{out} denote the inside and outside boundaries of the R* neighborhood respectively as shown in Fig 3. The desired vector fields will 'trap' the robots in these bands. Typically, this is a very narrow band of allowable space for the robots with a controllable width of: $\Delta R_{in} + \Delta R_{out.}$

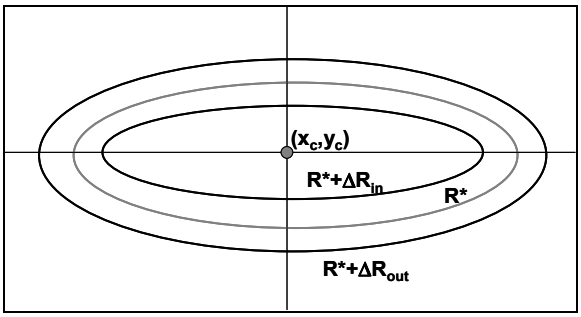

Figure 2: Elliptical attraction band for the swarm robots

We will construct the vector field utilizing the normalized gradient from (2). For every *(x, y)*, let the field vector have the form:

$$V(x,y)=\begin{cases} W(x,y)\dfrac{1}{\mathcal{L}(x,y)}\begin{pmatrix}(x-x_c)\\ \gamma(y-y_c)\end{pmatrix} \text{ for } (x,y)\neq(x_c,y_c) \\ \begin{pmatrix}0\\0\end{pmatrix} \text{ for } (x,y)=(x_c,y_c)\end{cases} \quad (6)$$

where

$$\mathcal{L}(x,y)=\sqrt{(x-x_c)^2+\gamma^2(y-y_c)^2} \quad (7)$$

The vector $\dfrac{1}{\mathcal{L}(x,y)}\begin{pmatrix}(x-x_c)\\ \gamma(y-y_c)\end{pmatrix}$ 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** ellipse, with:

$$R^*=\sqrt{(x-x_c)^2+\gamma^2(y-y_c)^2} \quad (8)$$

move out from the center until they reach the *R** neighborhood. Particles starting outside the ellipse move toward the center until they reach the *R** neighborhood. Eventually all the robots will be trapped within the neighborhood given in equation (8a):

$$R^* - \Delta R_{in} < R < R^* + \Delta R_{out} \quad (8a)$$

.

III. GENERATION OF VECTOR FIELD

A. *Description of Fields and Limiting Functions*

In order to generate the desired vector fields to hold the robots inside the R* neighborhood, three fields are needed. The gradient vector field, $G^- = -(d_x, d_y)$ points away from the center as shown in Fig. 3b. 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 towards the center as illustrated in Fig. 3a. The vector fields $(d_x, -d_y)$ and $(-d_x, d_y)$ are perpendicular to the gradient; Fig. 3c shows such a perpendicular field.

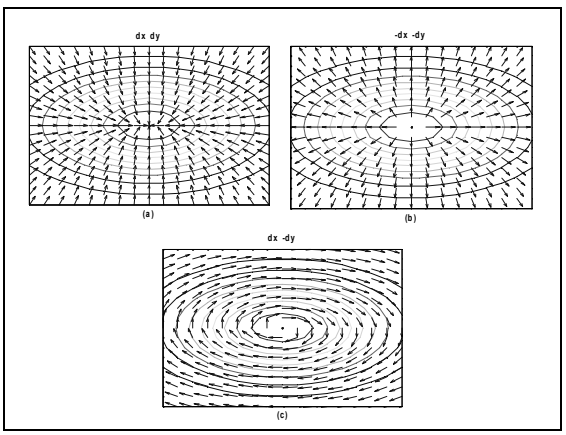

Figure 3. Vector fields directed (a) toward the center (G^+) and (b) away from the center (G^-). (c) perpendicular to the center (perpendicular to gradient).

Tighter swarm control may be accomplished when restricting the influence of the vector fields to a small region of the *x-y* plane by multiplying each of the fields by a 'limiting function'. This limiting function controls how far from the center the vectors in the field 'die out' or become smaller than some number ε.

In order to create the desired field, the G^- and G^+ fields as shown in Fig. 3a and Fig. 3b must be limited to end at the appropriate boundaries. These fields will be limited with *sigmoid* functions. The G^- field should die out at $R^*-\Delta R_{in}$, and the G^+ field should die out at $R^*+\Delta R_{out}$. This creates the field shown in Fig. 4.

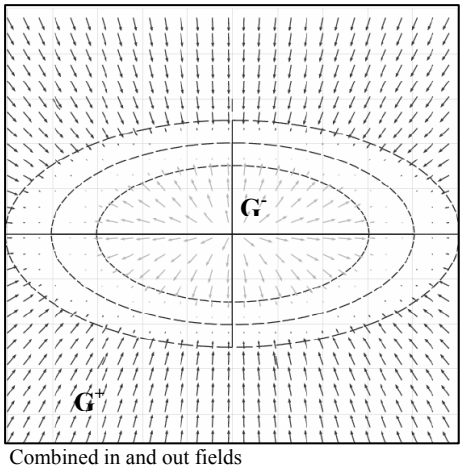

Figure 4. Combined in and out fields

For the weight, $W(x,y) = S_{in}(x,y) - S_{out}(x,y)$ so that we can consider the inside and the outside of the R* ellipse separately.

As a simplification we will change variables x and y to r where

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

So *W(x,y)* becomes $W(r) = S_{in}(r) - S_{out}(r)$. Notice r is never negative.

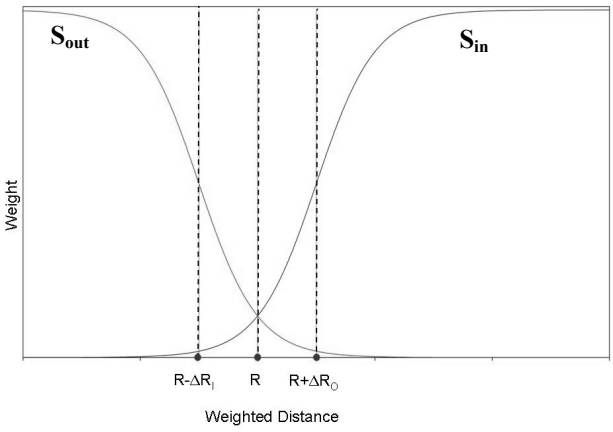

Figure 5. –The weight functions S_{in} and S_{out} as a function of the weighted distance r defined in equation 9.

Vector fields 'moving away' from the center (the vectors inside of the ellipse) require a limiting function

that approaches zero as the distance from the center is increased; such a limiting function is given in (10):

$$S_{in}(\alpha_{in}, r, R^*_{in}, \Delta R_{in}) = 1 - \frac{1}{1 + e^{\alpha_{in}(r-(R^*-\Delta R_{in})}} \quad (10)$$

Gradient vector fields directed towards the center (those vectors outside of the ellipse) are required to approach zero as the vectors 'move towards' the center; this is achieved using the limiting function in (11):

$$S_{out}(\alpha_{out}, r, R^*, \Delta R_{out}) = 1 - \frac{1}{1 + e^{-\alpha_{out}(r-(R^*+\Delta R_{out})}} \quad (11)$$

Attracting the robot to the R* neighborhood specified in equation (8a) 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 equation (8a) vectors in this field must die off outside this narrow band. A limiting function accomplishing that is given in (12):

$$N_{\perp}(\alpha_{\perp}, r, R^*) = e^{-\alpha_{\perp}(r-R^*)^2} \quad (12)$$

Function $N_{\perp}$ in (12) includes one tuning parameter $\alpha_{\perp}$. The distance at which the fields die out on either side of this perpendicular path is controlled by $\alpha_{\perp}$.

In addition, another multiplier to the perpendicular field must be added so the robots do not circle around the ellipse bands as in Fig. 3c. In order for the perpendicular field to change directions, the field perpendicular to the gradient is multiplied by (13):

$$SGN(\alpha_{\perp}, y_{rot}) = 1 - 2.0\left(\frac{1}{1 + e^{-\alpha_{\perp}(y_{rot})}}\right) \quad (13)$$

Each of the limiting functions in (10) through (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 parameter α_{in}, α_{out}, and $\alpha_{\perp}$ controls the slope of $S_{in}(r)$, $S_{out}(r)$, and $S_{\perp}(r)$ for r in the set $R - \Delta R_{in} < r < R + \Delta R_{out}$.

The value of $S_{in}(R^*)$ can be made arbitrarily small. Let $\varepsilon > 0$ be a small number such that $S_{in}(R^*) = \varepsilon$. We can then determine α_{in}. The same technique is used in the other limiting functions. The resulting equations are shown in (14) to (16).

$$\alpha_{in} = \frac{1}{\Delta R_{in}} \ln\left(\frac{1-\varepsilon}{\varepsilon}\right) \quad (14)$$

$$\alpha_{out} = \frac{1}{\Delta R_{out}} \ln\left(\frac{1-\varepsilon}{\varepsilon}\right) \quad (15)$$

$$\alpha_{\perp} = \frac{1}{((\Delta R_{out} - \Delta R_{in})/2)^2} \ln(1-\varepsilon) \quad (16)$$

The final vector field is depicted in Fig. 6. Functions S_{out}, $N_{\perp}$ and S_{in} 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 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 shown in (17):

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = (S_{in} - S_{out}) \begin{bmatrix} d_x \\ d_y \end{bmatrix} + SGN * N_{\perp} \begin{bmatrix} d_x \\ d_y \end{bmatrix}_{\perp} \quad (17)$$

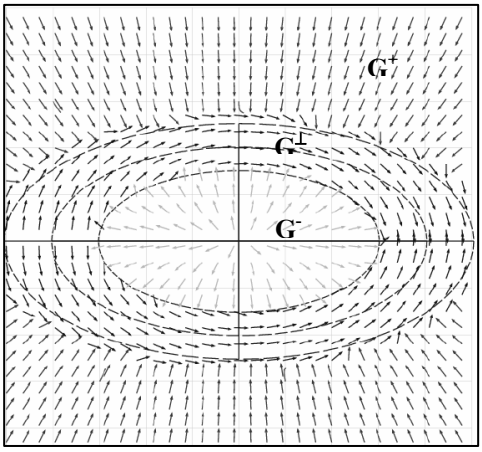

Figure 6. Final vector field

B. *Controlling Swarm Member Spacing within Bands*

Sigmoid functions may be used for obstacle avoidance as well as controlling member spacing by creating vectors moving away from the center of the obstacle's or other swarm member's location (x_{co}, y_{co}). For the purposes of this work, the concern is formation including member spacing, so 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 equations (18) to (20):

$$r_{avoid} = \sqrt{(x - x_{co})^2 + (y - y_{co})^2} \quad (18)$$

$$S_{avoid}(\alpha_{avoid}, r_{avoid}, \Delta R_{avoid}) = 1 - \frac{1}{1 + e^{\alpha_{avoid}(\sqrt{r_{avoid} - \Delta R_{avoid}})}} \quad (19)$$

$$\begin{bmatrix} d_{x\_avoid} \\ d_{y\_avoid} \end{bmatrix} = \begin{bmatrix} S_{avoid}(x - x_{co}) \\ S_{avoid}(y - y_{co}) \end{bmatrix} \quad (20)$$

Notice that r_{avoid} is similar to r except that instead of distance from the center, the distance to the swarm

member is used. The ΔR_{avoid} parameter determines the distance from other members. This is how equal dispersion of members occurs in formation. The S_{out}, $N_{\perp}$ and S_{in} get the robots to the band, but do not control their dispersion.

Avoidance of individual robot swarm members including their dispersion is controlled by the range of influence for the avoidance vector field. . The α_{avoid} parameter in (19) 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.

The α_{avoid} is tuned in the same way as the other Sigmoid limiting functions in (10) and (11). The ΔR_{avoid} parameter specifies the minimum distance between robots. The equation is shown in (21):

$$\alpha_{avoid} = \frac{1}{\Delta R_{avoid}} \ln\left(\frac{1-\varepsilon}{\varepsilon}\right) \tag{21}$$

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

The swarm may move from one waypoint to another by moving the center of the swarm (x_c, y_c). The equation to create vector fields to follow a trajectory with member avoidance by summing the vector fields is given in (22):

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = (S_{in}\text{-}S_{out}) \begin{bmatrix} d_x \\ d_y \end{bmatrix} + \sum_{1}^{\#size-1} S_{avoid} \begin{bmatrix} d_{x\_avoid} \\ d_{y\_avoid} \end{bmatrix} + SGN * N_{\perp} \begin{bmatrix} d_x \\ d_y \end{bmatrix} \tag{22}$$

Simulation results demonstrate in Sec. V.

IV. Simulation Results

Simulink has been used to model the robot swarm. Since this is proof of concept, only models of particles are used. Particles follow the vectors perfectly whereas robots are limited in their motion. In previous work, robot models were used. Four and ten particle simulations are run for demonstration of the concept. The swarm formation controller, which is identical for each robot/particle, is programmed in C. Each individual robot's vector generating controller is implemented as a C MEX S-function with the different control parameters fed in as well as a position vector with member locations.

It is important to note, that knowledge of the other member locations are not a necessity except for the dispersion aspect of the swarm. The way the knowledge is shared could be done in numerous ways making a case for both centralized and decentralized robotic systems, but this is not the relevant point of the paper. The robots will hold to the bands of the ellipse regardless of knowledge of each other.

Each of the C MEX S-functions generates the vector fields presented in Section IV at each time step. Formation changes are easily made by updating just a few tuning parameters. Parameters are selected based desired formation and dispersion using methods described in Sec. IV. The center of the swarm is broadcast into the vector generating S-functions. Swarm center, (x_c,y_c), is left fixed for these simulations in order to show the formation shape.

A. Four Robots

In order to demonstrate the capabilities of this method, simulations are run with four robots to make an arc and a circle formation. Table I shows the parameter values used for each formation. Notice that the only factor to change the formation from an arc to a circle is a modification of the ΔR_{avoid} parameter.

TABLE I
Control Variables 4 Robots

Control Variables	Value-Arc	Value-Circle
R^*	100	100
γ	1	1
ΔR_{in}	30	30
ΔR_{out}	20	20
ΔR_{avoid}	50	75
ε	0.001	0.001

Fig. 7 shows the paths from initial positions into formation. Fig. 8 shows the beginning swarm formation to the final swarm formation in the arc. Fig. 9 shows the final swarm formation in circle/square.

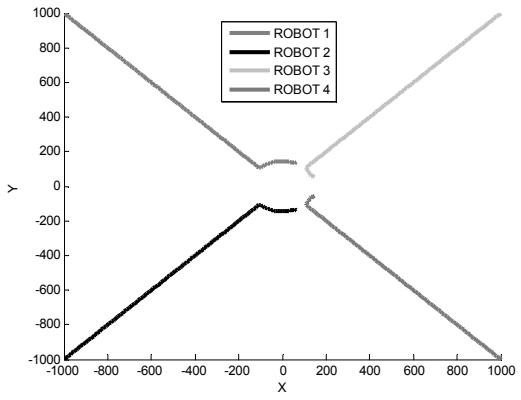

Figure 7. Robot paths from final position into arc formation

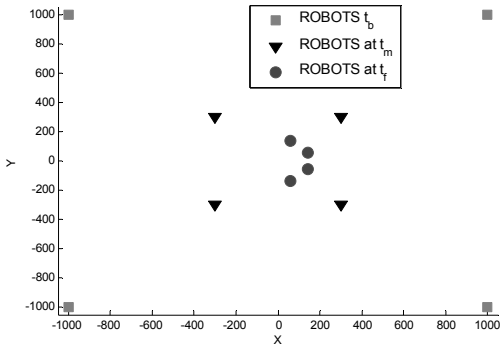

Figure 8. Robot formation from beginning to final in arc formation

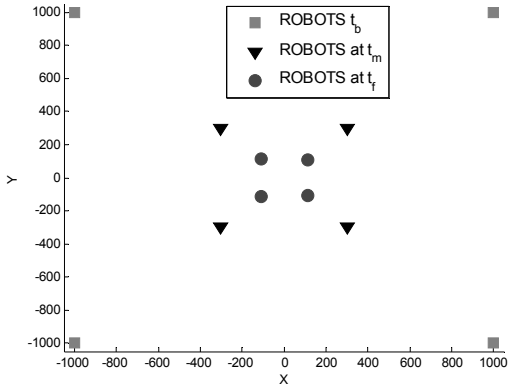

Figure 9. Robot formation from beginning to final-square/circle formation

B. Ten Robots

Simulations with ten particles in circle and ellipse formation are also run. Table II shows the parameter values used for each formation. Notice that the only factor to change the formation from a circle to an ellipse is a modification of the γ parameter to make the *y-axis* 'skinnier'.

TABLE II
Control Variables 10 Robots

Control Variables	Value-Circle	Value-Ellipse
R^*	200	200
γ	1	0.5
ΔR_{in}	60	30
ΔR_{out}	40	20
ΔR_{avoid}	75	75
ε	0.001	0.001

Fig. 10 shows the robots in circle formation. In addition, the ellipse bands are plotted. The central band is R^*. The outer band is $R^* - \Delta R_{in}$ and the inner band is $R^* + \Delta R_{out}$. The robots stay within the acceptable bands and disperse themselves. Fig. 11 is similar to Fig. 10 except it is in ellipse formation. The robots are on the band in order to adhere to the ΔR_{avoid} parameter which remained the same between the circle and the ellipse although the area for formation decreased.

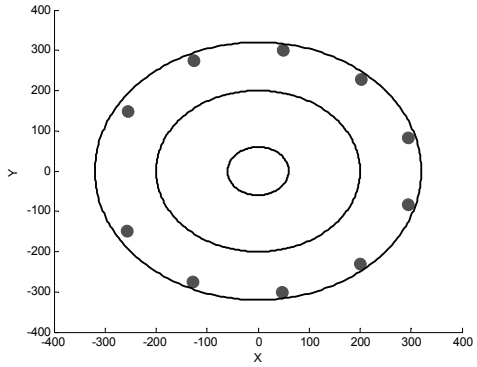

Figure 10. Robot formation Circle

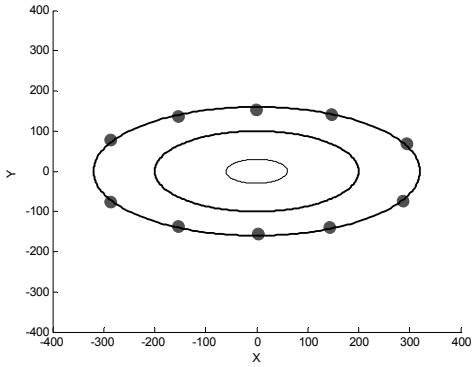

Figure 1. Robot formation ellipse

V. EXPERIMENTS

Initial experiments were performed using Traxxas Emaxx, RC-cars equipped with a custom built computer system shown in Fig. 12. The Emaxx vehicles are Ackerman steered and each is equipped with an inertial measurement unit (IMU) and global positioning system (GPS).

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

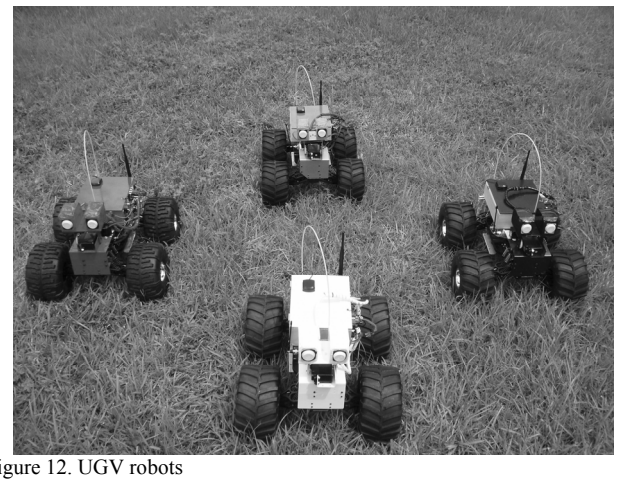

Figure 12. UGV robots

A simple broadcast communication model programmed in C is used for information relay and exchange. Each robot shares its position with the other robots for obstacle avoidance purposes only since the robots currently have no other sensing capabilities to avoid collision. The robots were all programmed in C.

Two different sets of experiments were performed. In one experiment, four UGV vehicles traveled in an ellipse formation surrounding the center that was broadcast to them. The second set of experiments begins the same as the first but integrates a single robot failure with dynamic formation change.

In the both sets of experiments, the robots receive different centers, (x_c, y_c), which are broadcast at different time steps and written into shared memory. At each time step, the robots compute their vectors based on current, position and the current center. Based on the output of the vector fields, (dx, dy), a desired speed and a desired heading are computed.

In the first experiment, no failures occur and the four UGVs travel surrounding each center point and staying a specified distance away from one another. Table III, column one, shows the control parameters used for the first set of experiments. The units for R*, ΔR_{in}, ΔR_{out}, and ΔR_{avoid} are all in meters. The UGVs all remain within the acceptable range $R - \Delta R_{in} < r < R + \Delta R_{out}$. A video demonstrating the first experiment can be found at http://www.csee.usf.edu/USL/Videos/4bot-clip1.wmv.

TABLE III
Control Variables UGVs

Control Variables	4-UGVs	3-UGVs
R^*	6	4
γ	1	1
ΔR_{in}	1.8	1.2
ΔR_{out}	9.6	6.4
ΔR_{avoid}	5	5
ε	0.001	0.001

In the second set of experiments, a UGV failure is integrated into the experiments. The UGV does not actually fail, but the communication server dies, and the robot is taken out of autonomous mode. When the other UGVs recognize that the communication link is broken, they dynamically change there parameters from those in column one of Table III to those in column two. This dynamic formation change results in a circle with a smaller radius. The UGVs seamlessly make the transition and continue after the failure. A video demonstrating the this experiment can be found at http://www.csee.usf.edu/USL/Videos/4-robots-failure.wmv.

These experiments demonstrate that the UGVs are capable of following vector field and successfully maintaining formation. The method is also able to smoothly recover from failures and dynamically change formation. No specific commands were given to a particular UGV. All swarm members ran identical code and had an identical mission. Presently, we are designing metrics to measure the how well the UGVs adhere to the formation.

VI. Conclusions

This paper has demonstrated that potential functions together with limiting functions can be successfully utilized to control robot swarm formation.

The presented method supports scalability, different swarm sizes, multiple formations, heterogeneous swarm member teams, centralized and decentralized formation control; since parameters are tuned off-line and used on line, the method is computationally inexpensive.

In the future, the work will be expanded to include more formations. In addition, a tuner for the control variables for the swarm functions via fuzzy logic is also being developed. This is necessary because the more fields that are added, the more necessary it is to modify parameters.

The approach will also be expanded to unmanned aerial vehicles (UAVs) by making the formation function a trivariate normal.

Acknowledgment

This research was supported in part by an appointment to Student Research Participation Program at U.S. Army Research Laboratory administered by the Oak Ridge Institute for Science and Education through interagency agreement between the U.S. Department of Energy and US ARL. This work was also supported partially by two grants ARO W911NF-06-1-0069 and SPAWAR N00039-06-C-0062.

References

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

[2] T. Balch and M. Hybinette, Social potentials for scalable multi-robot formations, *Proceedings of IEEE International Conference on Robotics and Automation*, 2000.

[3] L. Barnes, W. Alvis, M. Fields, K. Valavanis and W. Moreno, Heterogeneous Swarm Formation Control Using Bivariate Normal Functions to Generate Potential Fields, *Workshop on Distributed Intelligent Systems*, 2006.

[4] L. Barnes, W. Alvis, M. Fields, K. Valavanis and W. Moreno, Swarm Formation Control with Potential Fields Formed by Bivariate Normal Functions, *Mediterranean Conference on Control and Automation*, 2006.

[5] Z. Cao, M. Tan, S. Wang, Y. Fan and M. Zhang, The optimization research of formation control for multiple mobile robots, *Proceedings of the World Congress on Intelligent Control and Automation*, 2003.

[6] L. Chaimowicz, N. Michael and V. Kumar, Controlling swarms of robots using interpolated implicit functions, *Proceedings of International Conference on Robotics and Automation*, 2005.

[7] H. Chia-Hung and A. Liu, *Multiple teams for mobile robot formation control*, *Proceedings of IEEE International Symposium on Intelligent Control*, 2004.

[8] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowsky, J. Spletzer and C. Taylor, A vision-based formation control framework, *IEEE Transactions on Robotics and Automation*, 18 (2002), pp. 813--825.

[9] J. P. Desai, Modeling multiple teams of mobile robots: a graph theoretic approach, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2001.

[10] D. D. Dudenhoeffer and M. P. Jones, A formation behavior for large-scale micro-robot force deployment, *Simulation Conference Proceedings*, 2000.

[11] M. Egerstedt and X. Hu, Formation constrained multi-agent control, IEEE Transactions on Robotics and Automation, 17 (2001).

[12] R. Fierro and A. K. Das, A modular architecture for formation control, *International Workshop on Robot Motion and Control*, 2002.

[13] J. Fredslund and M. J. Mataric, A general algorithm for robot formations using local sensing and minimal communication, IEEE Transactions on Robotics and Automation, 18 (2002).

[14] K. Fujibayashi, S. Murata, K. Sugawara and M. Yamamura, Self-organizing formation algorithm for active elements, *Proceedings*

of IEEE International Conference on Robotics and Automation, 2000.

[15] V. Gazi, Swarm Aggregations Using Artificial Potentials and Sliding Mode Control, *IEEE Transactions on Robotics and Automation*, 21 (2005).

[16] S. S. Ge and C. H. Fua, Queues and artificial potential trenches for multirobot formations, *IEEE Transactions on Robotics and Automation*, 21 (2005).

[17] S. S. Ge, C. H. Fua and W. M. Liew, Swarm formations using the general formation potential function, *Proceeding of IEEE Conference on Robotics, Automation, and Mechatronics*, 2004.

[18] M. C. d. Gennaro, L. Iannelli and F. Vasca, Formation control and collision avoidance in mobile agent systems, *Proceedings of IEEE International Symposium on Intelligent Control*, 2005.

[19] Y. K. Hwang and N. Ahuja, A potential field approach to path planning, IEEE Transactions on Robotics and Automation (1992).

[20] 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 (2003).

[21] W. Kang and N. Xi, Formation control of multiple autonomous vehicles, *Proceedings of the IEEE International Conference on Control Applications*, 1999.

[22] W. Kang, N. Xi, Y. Zhao, J. Tan and Y. Wang, Formation control of multiple autonomous vehicles: Theory and experimentation, *Proceedings of IFAC 15th Triennial World Congress*, 2002.

[23] Y. Koren and J. Borenstein, Potential field methods and their inherent limitations for mobile robot navigation, *Proceedings of IEEE International Conference on Robotics and Automation*, 1991.

[24] W. Kowalczyk, Target assignment strategy for scattered robots building formation, *Proceedings of International Workshop on Robot Motion and Control*, 2002.

[25] W. Kowalczyk and K. Kozlowski, Artificial potential based control for a large scale formation of mobile robots, *Proceeding of Robot Motion and Control*, 2004.

[26] V. Kumar and L. Chaimowicz, Coordination among UAVs and Swarm Robots, *International Symposium on Distributed Autonomous Robotic Systems*, 2004.

[27] N. Leonard and E. Fiorelli, Leaders, artificial potentials and coordinated control of groups, *Conference on Decision and Control*, 2001.

[28] P. Renaud, E. Cervera and P. Martinet, Towards a reliable vision-based mobile robot formation control, *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2004.

[29] R. O. Saber, W. B. Dunbar and R. M. Murray, Cooperative control of multi-vehicle systems using cost graphs and optimization, *Proceedings of the American Control Conference*, 2003.

[30] R. O. Saber and R. Murray, Distributed cooperative control of multiple vehicle formations using structural potential fields, *IFAC World Congress*, 2002.

[31] R. O. Saber and R. Murray, Graph rigidity and distributed formation stabilization of multi-vehicle systems, *Proceedings of Conference on Decision and Control*, 2002.

[32] F. E. Schneider and D. Wildermuth, Experimental Comparison of a Directed and a Non-Directed Potential Field Approach to Formation Navigation, *Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2005.

[33] F. E. Schneider and D. Wildermuth, A potential field based approach to multi robot formation navigation, *Proceedings of International Conference on Robotics, Intelligent Systems and Signal Processing*, 2003.

[34] P. Song and V. Kumar, A potential field based approach to multi-robot manipulation, *Proceedings of IEEE International Conference on Robotics and Automation*, 2002.

[35] W. M. Spears, D. F. Spears, W. K. R. Heil and S. Hettiarachchi, An Overview of Physicomimetics, *SAB International Workshop*, 2004.

[36] K. Sugihara and I. Suzuki, Distributed algorithms for formation of geometric patterns with many mobile robots, *Robot Systems*, 13 (1996).

[37] H. Yamaguchi, Adaptive formation control for distributed autonomous mobile robot groups, *Proceedings of IEEE International Conference on Robotics and Automation*, 1997.

[38] H. Yamaguchi, T. Arai and G. Beni, A distributed control scheme for multiple robotic vehicles to make group formations, *Robotics and Autonomous Systems*, 36 (2001), pp. 125-147.

[39] S. Zilinski, T. Koo and S. Sastry, Optimization-based formation reconfiguration planning for autonomous vehicles, *Proceedings of IEEE International Conference on Robotics and Automation*, 2003.