A Single- and Multi-Dimensional Cellular Automata Approach to Robot Formation Control

Jerry B. Weinberg, *Member, IEEE*, and Ross Mead

***Abstract*—One of the challenges to the realization of harvesting solar power from space is overcoming the difficulties and high cost for the transportation and construction of a large orbiting solar reflector. One approach that has received attention is using thousands of individual robots moving in formation, each with a piece of the reflector attached, to form the structure. In previous work, we demonstrate an algorithm that treats a group of robots as a 1-dimensional cellular automaton, which is able to establish formations defined by a geometric description that can be expressed as a single mathematical function. In this paper we extend the algorithm for robots to establish formations defined by multiple mathematical functions in order to form grid structures.**

I. Introduction

The Space Frontier Foundation recently sponsored a public online discussion on the feasibility of creating an orbiting reflector for harvesting solar power [http://spacesolarpower.wordpress.com/]. This discussion was inspired by an ongoing study of space-based solar power (SSP) being conducted by the National Security Space Office. One of the major technical issues discussed was the construction of a large orbiting solar reflector. Many satellite concepts have been proposed, including the Suntower and the SolarDisc [1; Fig. 1]. However, none of these monolithic architectures have tackled "the biggest technical challenge: launching the device into orbit" [2]. For a large solar reflector that is potentially numerous kilometers in size, the difficulties and high-cost of transporting the necessary components for its construction and the actual construction of the reflector itself are major obstacles.

One approach articulated in a joint NSF/NASA workshop considered the use of individual robots in formation to shape a solar reflector structure:

> [The reflector is] actually created by having a swarm of coordinated independent semi-intelligent objects [(i.e., robots)] acting in concert. A solar reflector might be created in this way by having thousands of small free-flyers, each with a piece of mirror attached to themselves, fly into and then maintain a parabolic formation. One advantage of this strategy is that if the system is ever damaged, the swarm could reconfigure to eliminate the damaged elements but still maintain whatever level of uniformity might be required. [3]

This poses some interesting questions. For example, once deployed, how does this large collection of robots communicate and coordinate their activities to form an organized parabolic structure resembling a reflector? How do the robots know when and where to move to maintain their position within the formation? And how can one operator or a small group of operators communicate with thousands of robots to effectively change the formation as needed?

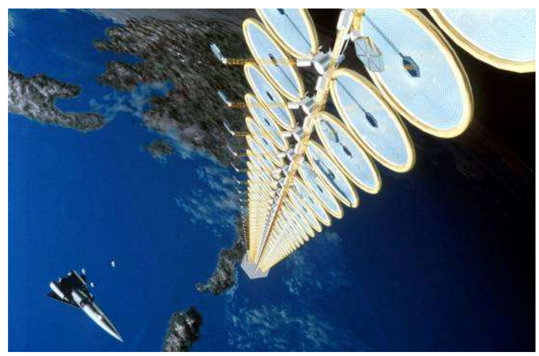

Fig. 1. The proposed Suntower for SSP [1].

In our previous work, robots are treated as cells in a 1-dimensional cellular automaton [4, 5]. Each robot "cell state" consists of its orientation in 2-dimensional space in relation to neighboring robots. Using a reactive control architecture this approach is able to establish and maintain formations defined by a geometric description that can be expressed as a single mathematical function. The viability of this approach was demonstrated in simulation with thousands of robots [4] and on a physical platform with twelve robots [5].

In order to attain parabolic formations necessary for applications such as a solar reflector the algorithm must be generalized to 2- and 3-dimensional cellular automata. In this paper we present a formal description of the algorithm for robots to move in formations defined by a single mathematical function and then show how it can be generalized to formations defined by multiple mathematical functions in order to form grid structures.

II. Background

Seminal work by C. Reynolds [6] on flocking shows how individual agents can exhibit emergent aggregate behavior,

Manuscript received September 14, 2007.

J. B. Weinberg is with the Department of Computer Science, Southern Illinois University Edwardsville, Edwardsville, IL 62025 USA (email: jweinbe@siue.edu).

R. Mead is with the Department of Computer Science, Southern Illinois University Edwardsville, Edwardsville, IL 62025 USA (email: qbitai@gmail.com).

similar to flocking birds, based solely on the orientation, direction, and proximity of each agent to its neighbors. Flocking algorithms have been applied to both physical [7] and simulated robots [8]. Distributed control methods based on gravitational physics [9], gas particle interactions [10], and a digital hormone model [11] have generated similar emergent structures. However, these approaches apply to swarms of agents as opposed to formations.

For our purposes, a *swarm* is defined as a massive collection that moves with no internal group organization, much like a school of fish or a flock of birds. While a *formation* is similar, the distinction is made in that it maintains a global structure, much like a flock of geese or a marching band. Robot formations have been applied to applications such as automated traffic cones [12].

Robots capable of attaching to one another have been proposed for formation applications. Tethered formation flight has been implemented by SPHERES satellites [13; Fig. 2]. By keeping each tether taut, these small robotic units are able to preserve their organization. Related work utilizes electromagnets and a reaction wheel to control the position and orientation of a robot relative to another [14], but is currently limited in the number of units that can be controlled.

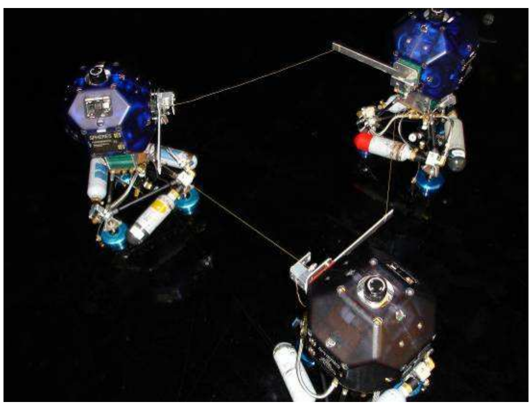

Fig. 2. SPHERES satellites maintain formation by keeping tethers taut [13].

In [15], robots autonomously self-assemble into a single robotic configuration. These "swarm-bots" only interact with and connect to robots in their immediate vicinity and, thus, self-organize without relying on a global coordination system [Fig. 3]. These types of strict connections between robots make for a strong structure, but are rather restricted in their ability to change configurations and replacement of malfunctioning units. The challenges for applying a robot formation approach to an application such as SSP include managing the shape and size of the structure, easily altering the orientation of the structure, and repairing the structure through easy replacement of malfunctioning robots.

Previous work on maintaining formation has been applied on up to a dozen physical robots [5]; however, to better demonstrate group behavior on a larger scale, the number of mobile units must be increased. For a formation of robots numbering in the thousands**,** such as the 33,000 robots that Landis [16] projected would be needed for SSP, direct control through a remote teleoperation interface, in which an operator or a small group of operators must take control of each individual robot, is impractical. Rather, the robot formation will require some level of autonomous control. Computationally, the simplest control strategy is reactive control where, in the case of formations, each robot responds to changes in its local neighborhood. It has the advantage of eliminating the need for each robot to maintain information about the global or overall structure of the entire formation.

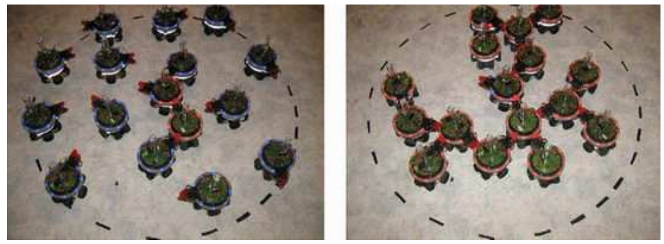

Fig. 3: Independent "swarm-bots" (left) connect with others (right) [15].

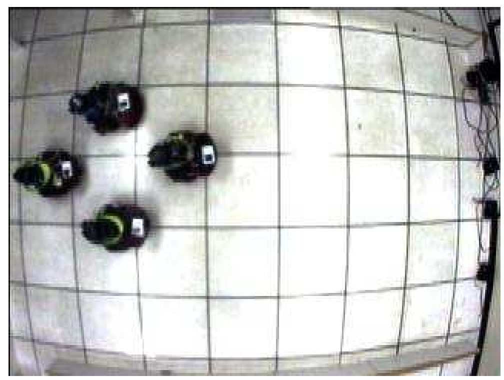

Fig. 4. Four mobile robots traveling in diamond formation [18].

This approach to autonomous multi-robot formations has been applied to Earth-bound, mobile robots [17, 18]. Inspired by biological or organizational systems, such as geese flying in formation, Fredslund & Mataric [18] assign a particular formation for robots to follow, such as a line, a V-shape, or a diamond [Fig. 4]. Each robot is given and transmits a position in the formation, an identification number, and the color of its corresponding "helmet". Using a combination of laser sensors, color tracking, and robot intercommunication, each robot is able to identify its "friend" (i.e., reference neighbor). A robot leader is designated that transmits the specified formation. When the leader moves, neighboring robots adjust their orientation appropriately. In this way, the robot group is able to maintain a stable formation. However, this will only occur if the formation is in motion; if the formation is not moving, robots are only able to determine their distance to their corresponding friend—true orientation is unknown. Because the robots are forced to move, an inherent restriction is imposed on the system: robots must follow (i.e., stay behind) their friend. Thus, formations are limited to those of non-frontal concavities with respect to the leading robot.

Mead *et al.* [4, 5] treat robots in a formation as cells in a

1-dimensional *robot-space cellular automaton* [Fig. 5]. This differs from Dudenhoeffer & Jones [19] where the robots exist within an environment modeled as a 2-dimensional cellular automaton, referred to as a *world-space cellular automaton*. Working in the robot-space overcomes many of the limitations inherent in a world-space automaton, eliminating the dependence on the environment and reducing the complexity of the automaton. Each robot "cell state" consists of its desired and actual spatial orientation in 2-dimensional space in relation to neighboring robots. The robot's behavior is governed by a set of rules for changing its state with respect to its neighbors. By designating a single robot as a "seed" or "initiator" cell of the formation, human intervention can change its orientation directly. This causes a type of chain reaction in the formation. This is analogous to seeing a crowd in a baseball stadium "doing the wave", where each individual's reaction in the crowd is based solely on the people sitting nearby. The overall formation is defined by a single mathematical function [Fig. 6].

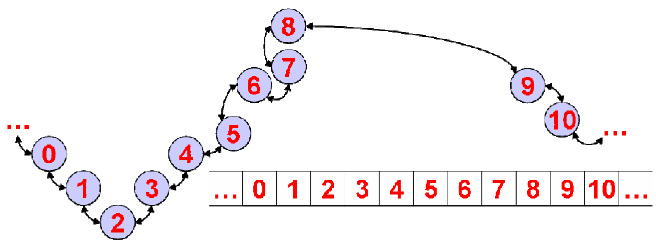

Fig. 5. Robots as cells in a 1-dimensional robot-space cellular automaton.

A physical robot platform was developed to test and evaluate the algorithm [5; Fig. 7]. Each robot is 7 inches in diameter and is crafted from expanded PVC. Two modified servo motors provide differential steering. The algorithm is implemented in Interactive C [www.kipr.org/ic] and runs on an onboard XBCv2 microcontroller [www.botball.org]. Using back-EMF PID motor control, the robot is able to make movements accurate to ±1.62cm. It features a color camera for simultaneous, multiple blob tracking. Rotating the robot provides a 360° view of the environment and neighboring robots, which are identified by color bands. Using an XBee radio communication module [www.maxstream.net], robots share state information with their neighbors [5]. A reliable packet communication library was implemented that provides automatic retries and acknowledgements. Packets can be addressed to a specific robot or broadcast. This allows each robot to communicate locally within its neighborhood.

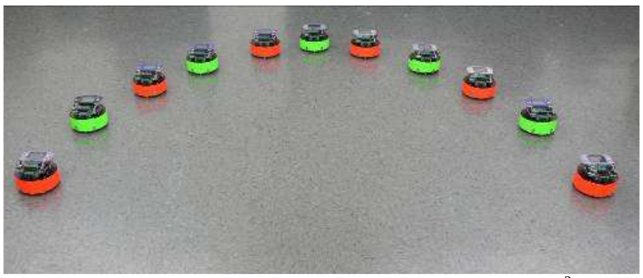

Fig. 6. Robots in a parabolic formation defined by $f(x) = x^2$.

Fig. 7. The robot platform.

In the next section, we formally define this cellular automata-based formation control algorithm. In Section IV, we show how it can be generalized to grid formations.

III. FORMATION CONTROL ALGORITHM

A. Representation and Notation

Each robot is represented as a cell c_i in a 1-dimensional cellular automaton, where i is an index that refers to the robot within the automaton; note that an index is not necessarily the robot's identification number or address that is used for communication—it is simply a reference. Each cell is given a neighborhood, denoted $\{c_{i-1}, c_i, c_{i+1}\}$, where c_{i-1} and c_{i+1} refer to the left and right neighbors of c_i, respectively. We refer to any given neighbor as c_j, where j is the index of the cell representing a neighboring robot.

B. Formation Definition

A desired formation F is defined as a geometric description (in the current implementation, a single mathematical function) $f(x)$. This definition is sent to some robot, designating it as the *seed* cell c_{seed} of the automaton. For purposes of determining relationships, a cell considers itself to be at some function-relative position p_i:

$$p_i \leftarrow \langle x_i, f(x_i) \rangle \quad (1)$$

In the case of c_{seed}, the position p_{seed} is given and serves as a starting point from which the formation and relationships will propagate.

C. Desired Relationships

Relationships are determined by calculating a vector v from p_i to the intersection of $f(v_x)$ and a circle centered at p_i with radius R, where R is the desired distance to maintain between neighbors in the formation:

$$R^2 \leftarrow (v_x - p_{i,x})^2 + (f(v_x) - p_{i,y})^2 \quad (2)$$

$$r_{i\to j,des} \leftarrow \langle v_x, f(v_x) \rangle \quad (3)$$

Solving for the *desired relationship vector* $r_{i\to j,des}$ to some neighbor c_j results in two intersections: one in the positive direction and one in the negative direction. These solutions define right and left relationships $r_{i\to i+1,des}$ and $r_{i\to i-1,des}$, respectively [Fig. 8].

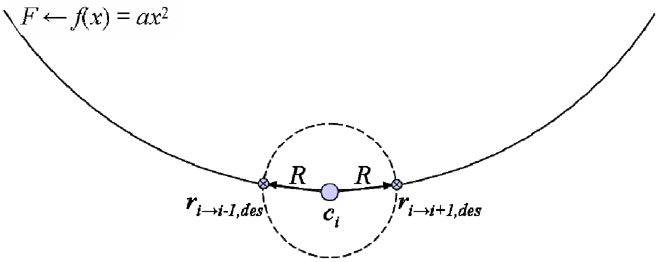

Fig. 8. c_i calculates the desired relationships to its neighbors.

The formation definition and relationship information are communicated locally within the neighborhood. Neighboring robots repeat the process, but consider themselves to be at different function-relative positions as determined by the desired relationship from their neighbor. For a neighbor c_j:

$$p_j \leftarrow p_i + r_{i\to j,des} \tag{4}$$

$$r_{j\to i,des} \leftarrow -r_{i\to j,des} \tag{5}$$

Note that relationships $r_{j\to i,des}$ and $r_{i\to j,des}$ are equal in magnitude, but opposite in direction. This property of the algorithm is what guarantees convergence and stability between two robots attempting to establish and maintain relationships with one another.

Using only sensor readings and local communication, these relationships result in the overall organization of the desired global structure [Fig. 9]. It follows that a movement command sent to a single robot will cause a chain reaction in neighboring robots, which then change states accordingly, resulting in a global transformation. Likewise, to change a formation, a seed cell is chosen and given the new geometric description and the process is repeated.

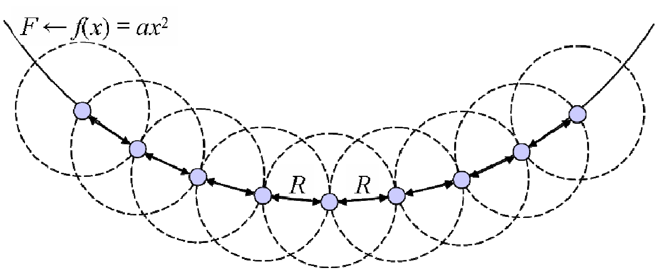

Fig. 9. Calculated relationships generate a parabolic formation.

D. Actual Relationships

A neighboring robot represented by c_j is identified by either an orange or green color band; the color of each robot is assigned based on its ID: green for even; orange for odd. The alternating of color bands reduces the chances of overlapping color blobs and improves the accuracy of detecting a neighbor. To locate c_j, a robot represented by c_i rotates until the band of the appropriate color is within its view; it then centers on that band. The heading of c_i is always considered to be directed at the x-axis (0°); relative to the robot, left yields positive angles and right yields negative angles. The distance $d_{i\to j}$ between c_i and c_j is determined by recognizing that the perceived vertical displacement Δy between the top and bottom of a color band is proportional to the perceived vertical displacement ΔY at a known physical distance D [Fig. 10]:

$$d_{i\to j} \leftarrow D \times \Delta Y / \Delta y \tag{6}$$

The relative orientation $\alpha_{i\to j}$ from c_i to c_j is simply the angular displacement from the initial location (i.e., prior to the search) of c_i. Thus, the *actual relationship vector* $r_{i\to j,act}$ is written in polar coordinates as:

$$r_{i\to j,act} \leftarrow \langle d_{i\to j}, \alpha_{i\to j}\rangle \tag{7}$$

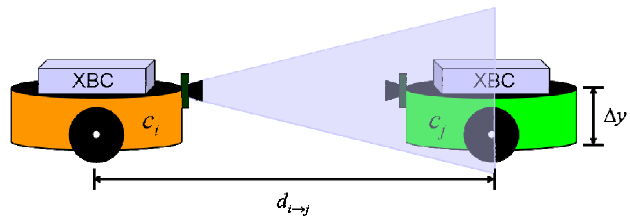

Fig. 10. c_i identifies and determines its distance to c_j.

E. Motion Control

By evaluating the state of the neighborhood, c_i is able to determine the translational error Γ_i and rotational error Θ_i that will define its movement, which is assumed to be along a safe trajectory. To do this, the robot must first seek a point of reference. Recall that the seed cell c_{seed} was given an initial function-relative position p_{seed}, from which the formation and relationships propagate. As the distance from c_{seed} increases, the propagated error accumulates. It follows that the neighbor of c_i whose function-relative position is closest to p_{seed} will have the least amount of propagated error within the system and, thus, will likely be the most reliable robot to reference. We denote this reference neighbor c_k to be the neighbor that yields the minimum distance $\|p_k'\|$ from p_{seed} to p_k:

$$\|p_k'\| \leftarrow \min\{\|p_{i-1} - p_{seed}\|, \|p_{i+1} - p_{seed}\|\} \tag{8}$$

1) Rotational Error Θ_i

The rotational error Θ_i is a difference in robot orientation, which is propagated to each successive cell in the automaton. To determine Θ_i, c_i considers $r_{k\to i,act}$ with respect to itself. Let $\theta_{i\to k}$ represent the relative angle between the headings of c_i and c_k, and let θ_i and θ_k be the angles of the relationship vectors $r_{i\to k,act}$ and $r_{k\to i,act}$, respectively. Then:

$$\theta_{i\to k} \leftarrow \theta_k - \theta_i + 180°; [-180°, 180°] \tag{9}$$

$$\Theta_i \leftarrow \Theta_k + \theta_{i\to k}; [-180°, 180°] \tag{10}$$

Note that if $\theta_{i\to k} = 0°$, both i and k have the same global heading. The same holds true for every robot in the formation if $\Theta_i = 0°$ for all i. This property of the algorithm to yield an emergent global heading is essential for any subsequent teleoperation from an operator. If a translational movement command is given, the common heading of the robots allows for a smooth transition in the same direction.

2) Translational Error Γ_i

The appropriate translational movement for each robot in the automaton is determined by an accumulation of error with both x- and y-components. This error is determined by the difference in desired and actual relationships of reference cell c_k. One major consideration is that many of the robots are, themselves, correcting for translational and rotational errors while they are being referenced by other robots. Changes in the orientation of c_k can cause rather entropic behavior in c_i, which depends on it for motion control. To alleviate this problem, $r_{k \to i,des}$ must be rotated, accounting for the propagated rotational error Θ_k within the automaton. Let $r_{k \to i,des}'$ denote $r_{k \to i,des}$ rotated by an angle $-\Theta_k$. We express γ_i as the translational error of c_i with respect to c_k:

$$\gamma_i \leftarrow \Gamma_k + r_{k \to i,des}' - r_{k \to i,act}, \tag{11}$$

Recall that $\theta_{i \to k}$ relates the headings of both of these robots, providing a conversion between the relative coordinate systems of c_i and c_k. Thus, rotating γ_i by $-\theta_{i \to k}$ yields the translational error Γ_i.

IV. Extension for Grid Formations

Extending our previous work [4, 5] to be applied to SSP, the global structure of the solar reflector array could be viewed as a 2-dimensional lattice of robots. In [4, 5, 18], the formations were setup so that each robot only needed to identify two neighbors. In the reflector formation, the definition of a neighborhood becomes more complex. There are various neighborhoods defined for cellular automata, such as the von Neumann Neighborhood (four surrounding cells), Moore Neighborhood (eight surrounding cells), or the Extended Moore Neighborhood (includes cells adjoining the immediate eight surrounding cells) [20]. Landis [16] provides insight into potential physical configurations of robots for SSP, which is considered in determining the appropriate number of neighbors [Fig. 11].

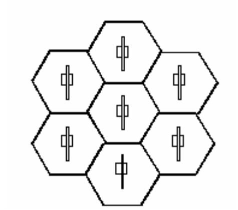

Fig. 11. Arrangement of robotic units to form a tight lattice structure [16].

A. Formations Defined by Multiple Functions

Our previous work [4] identified the potential for different categorizations of formations, including those that are defined by multiple functions. We extend our initial formation definition F to include more than one function $\{f_1(x), f_2(x), \ldots, f_n(x)\}$, where n is the number of functions. Each function is considered individually for purposes of calculating desired relationships and, thus, generates its own 1-dimensional neighborhood as before; this results in n neighborhoods. We write the neighborhood for a function $f_m(x)$ to be $^m\{c_{i-1}, c_i, c_{i+1}\}$, where 1 <= m <= n. The combined neighborhood can then be represented as $\{^1\{c_{i-1}, c_i, c_{i+1}\}, {}^2\{c_{i-1}, c_i, c_{i+1}\}, \ldots, {}^n\{c_{i-1}, c_i, c_{i+1}\}\}$. The location for any cell c_i is unique within the automaton as a whole and is a common intersection of each neighborhood. Therefore, we can rewrite the neighborhood of a cell c_i to be $\{^nc_{i-1}, \ldots, {}^2c_{i-1}, {}^1c_{i-1}, c_i, {}^1c_{i+1}, {}^2c_{i+1}, \ldots, {}^nc_{i+1}\}$. We refer to some neighbor ^mc_j, where m refers to the function upon which j is an index to a neighboring cell. This distinction is especially important when determining the aforementioned reference neighbor c_k; we now determine the identity of this neighbor by extending Equation (8) to consider all neighbors within a neighborhood defined by multiple functions.

As an example and for the purposes of SSP, we aim to generate a hexagonal lattice structure [16]. We define a formation F by three functions: $f_1(x) = 0$, $f_2(x) = x\sqrt{3}$, and $f_3(x) = -x\sqrt{3}$. Solving for Equation (3) for each of these functions yields a total of six desired relationships [Fig. 12].

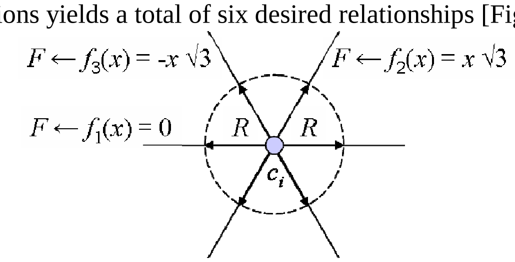

Fig. 12. A formation defined by three functions yields six relationships.

As before, the definition and relationships are propagated to neighboring cells, which repeat the process. Using the standard implementation of the algorithm, the resulting formation would be a sort of "six-armed structure". This is the product of the p_i term in Equation (3), which considers the cell c_i to be at position p_i on some function. While this may be desirable for some applications, it deviates from our goal of a lattice. By removing the p_i term from Equation (2), all cells will be considered at the origin when calculating their desired relationships. Note that the p_i term does not leave the algorithm completely—it is still propagated within the automaton for purposes of determining the reference neighbor. This seemingly minor change produces significant results: cells begin to have common neighbors as neighborhoods intersect one another; an emergent hexagonal lattice structure results [Fig. 13].

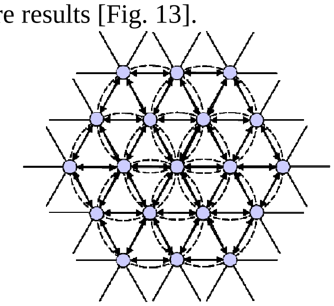

Fig. 13. A three-function formation definition generates a lattice structure.

B. Dynamic Neighborhoods

We have shown that, as the formation changes, the neighborhood may change as well. In particular, consider a formation change in which the number of defining functions changes. The current work on mobile robot formations requires that individual units have some sense of where they belong in the formation, most notably who their neighbors are supposed to be. One of the goals of this project is to generate this information dynamically as the swarm becomes a formation and as the formation adjusts its pattern. If the robots are not initially put in a formation, then a neighborhood must be established dynamically. This can be done by implementing a market-based auctioning method where a robot is chosen to be a neighbor based on its distance to the desired location in the formation description. This provides robustness in a system of imperfect and generic robots. If a robot exhibits failure and is removed from the formation, neighboring robots could dynamically reposition, autonomously, so that the global integrity of the reflector is maintained. Periodically, new robots could be deployed that would add themselves to the edges of the formation for replenishment as needed.

C. Neighbor Identification

For a robot to determine its state, it must be able to identify its neighbors. This is easier said than done, as each unit looks identical. The problem is analogous to being in a room of people that look and dress exactly alike; if one person speaks (i.e., sends a message) without further sensory input, such as auditory localization, it would be impossible to determine who was speaking. We alleviate this problem by utilizing a colored bar-coding system [Fig. 14]. Each robot features a three-color column; the unique vertical location of the ID color bar (in relation to the top and bottom color bars) is proportional to the identification number of the robot.

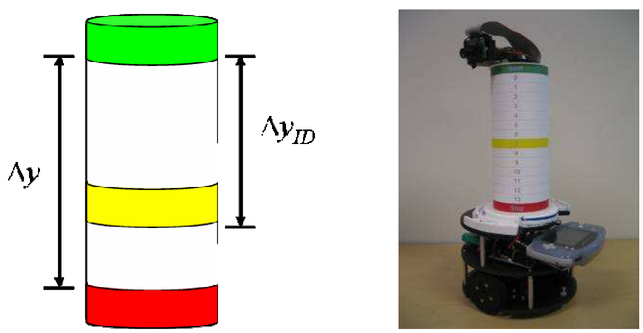

Fig. 14. Colored barcode design (left) prototyped on a robot (right).

V. Future Work

A series of experiments will be conducted and evaluated based on the criteria discussed in Fredslund & Mataric [18] for reasons of comparison and analysis of the algorithm. These include *generality* (can the swarm conform to a variety of formations, including convex and concave forms?), *stability* (how well is the formation maintained when in motion?), *robustness* (can the formation respond to changes in group size?), and *dynamic switching capability* (how well can the formation respond to an operator's command for changes?).

Acknowledgments

We would like to thank Jeff Croxell for his hard work in designing and manufacturing the PCB interface board between the XBee radio communication module and the XBC [5]. We would also like to thank the Undergraduate Research Academy of SIUE for its support of this project.

References

[1] J.C. Mankins, "A Fresh Look at Space Solar Power: New Architectures, Concepts and Technologies". IAF-97-R.2.03, 38th International Astronautical Federation., 1997.

[2] T. Clynes, "The Energy Fix: 10 Steps to End America's Fossil Fuel Addiction". *Popular Science*, July 2006: pp. 47-60.

[3] G. Bekey, I. Bekey, D. Criswell, G. Friedman, D. Greenwood, D. Miller, & P. Will, "Final Report of the NSF-NASA Workshop on Autonomous Construction and Manufacturing for Space Electrical Power Systems". Pp. 4-7 April 2000, Arlington, Virginia.

[4] R. Mead & J.B. Weinberg, "Algorithms for Control and Interaction of Large Formations of Robots". In Proceedings of AAAI-06, pp. 1891-1892. Boston, Massachusetts, 2006.

[5] R. Mead, J.B. Weinberg, & J.R. Croxell, "An Implementation of Robot Formations using Local Interactions". In Proceedings of AAAI-07, pp. 1989-1990. Vancouver, British Columbia, Canada, 2007.

[6] C.W. Reynolds,. "Flocks, Herds, and Schools: A Distributed Behavioral Model", in Computer Graphics. SIGGRAPH '87 Conference Proceedings, 21(4), 1987, pp. 25-34.

[7] I.D. Kelly & D.A. Keating, D.A., "Flocking by the Fusion of Sonar and Active Infrared Sensors on Physical Autonomous Mobile Robots". The 3rd International Conference on Mechatronics and Machine Vision in Practice, 1996, pp. 1-4. Guimaraes, Portugal.

[8] K. Ando, I. Suzuki, & M. Yamashita, "Formation and Agreement Problems for Synchronous Mobile Robots with Limited Visibility". Proceedings of the IEEE International Symposium on Intelligent Control, Monterey, California, 1995, pp. 453-460.

[9] W. M. Spears, D.F. Spears, J. C. Hamann, & R. Hill,. "Distributed, Physics-Based Control of Swarms of Vehicles". Autonomous Robots, (17), 2004, pp. 137-162.

[10] J. Cheng, W. Cheng, & R. Nagpal, "Robust and Self-repairing Formation Control for Swarms of Mobile Agents". In Proceedings of AAAI-05, 2005, pp. 59-64. Pittsburgh, Pennsylvania.

[11] W. Shen, P. Will, A. Galstyan, & C. Chuong "Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms". Autonomous Robots, 17, 2004, pp. 93-105.

[12] S.M. Farritor & S. Goddard, "Intelligent Highway Safety Markers". IEEE Intelligent Systems, 19(6), 2004, pp. 8-11.

[13] A. Saenz-Otero & D.W. Miller, (2005). "SPHERES: A Platform for Formation-flight Research". UV/Optical/IR Space Telescopes: Innovative Technologies and Concepts II Conference, San Diego, California, 2005.

[14] E.M.C. Kong, D.W. Kwon, S.A. Schweighart, L.M. Elias, R.J. Sedwick, D.W. Miller, "Electromagnetic Formation Flight for Multisatellite Arrays". AIAA Journal of Spacecraft and Rockets, 41(4), 2004, pp. 659-666.

[15] R. Groß, M. Bonani, F. Mondada& M. Dorigo, "Autonomous Self-Assembly in Swarm-Bots". IEEE Transactions on Robotics, 22(6), 2006.

[16] G. Landis, "Reinventing the Solar Power Satellite". The 53rd International Astronautical Congress, Houston, Texas, 2004.

[17] T. Balch & R. Arkin, "Behavior-based Formation Control for Multi-Robot Teams". IEEE Transactions on Robotics and Automation, 14(6), 1998, pp. 926-939.

[18] J. Fredslund & M. Mataric, "Robots in Formation Using Local Information". The 7th International Conference on Intelligent Autonomous Systems, Marina del Rey, California, 2002.

[19] D.D. Dudenhoeffer & M.P. Jones, M.P., "A Formation Behavior for Large-Scale Micro-Robot Force Deployment". Proceedings of the 2000 Winter Simulation Conference, 2000, pp. 972-982.

[20] P. Sarkar, "A Brief History of Cellular Automata". ACM Computing Surveys, 32(1), 2000, pp. 80-107.