

Fault-Tolerant Formations of Mobile Robots

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

Abstract—The goal of a robot formation control architecture is to get a number of robots into a specified form. To be effective and practical, the control architecture must be able to transition a group of robots from an initial swarm to a final formation. It must then be able to handle real-world events that could disrupt the formation, thus, requiring formation repair, obstacle avoidance, and changes in the formation. In previous work, we presented a distributed, reactive cellular automata-based formation control architecture capable of controlling any number of robots in formation at once. In this paper, we examine our architecture with respect to necessary characteristics to handle real-world occurrences. To address issues of formation repair and obstacle avoidance, the control architecture is extended by a distributed auctioning method that allows the robot formation to reconfigure autonomously.

I. INTRODUCTION

Proposed applications of large formations of mobile robots are numerous and include search and rescue, battlefield reconnaissance, and exploration and survey. Some proposals call for formations of robotic satellites, specifically, space-based solar power collection [1] and sparse aperture telescopes like NASA's Terrestrial Planet Finder [2] and the ESA's Darwin telescope (Fig. 1) [3]. These applications call for large numbers of homogeneous robotic satellites to establish a tight formation, maneuver and maintain formation, and exhibit fault-tolerance such that the malfunction of one or a small group of units can occur without affecting the function of the system as a whole.

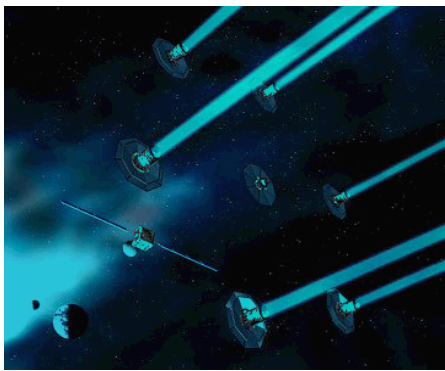


Fig. 1: An illustration of ESA's Darwin. [3]

Manuscript received March 1, 2009.

Ross Mead is with the Computer Science Department, University of Southern California, Los Angeles, CA 90089 USA (e-mail: rossmead@usc.edu).

Rob Long is with the Department of Computer Science, Southern Illinois University Edwardsville, Edwardsville, IL 62025 USA (e-mail: Robert.L.Long@gmail.com).

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

It is useful to draw a distinction between swarms and formations. An example of a *swarm* in nature is a school of fish. The individual members of the school have no fixed relationship to any of the other members. The school undulates and changes size and shape as it maneuvers, with individuals changing position relative to one another [4]. An example of a *formation* in nature is a “V”-shaped flock of migratory birds. Such natural formations differentiate themselves from swarms in that any two individuals maintain a relatively constant relationship to one another.

For robot formations to be effective in the applications noted, the control architecture must be able to transition a swarm into a formation and handle the various difficulties of malfunctions, interactions with the physical environment, and other unexpected events. To determine the effectiveness of a formation control system, Fredslund & Mataric [5] have defined a number of criteria. In this paper, we examine our approach for control against these criteria and propose an extension to alleviate identified limitations.

II. BACKGROUND

As described in [6], the approach taken to robot formation control is to treat robots as cells in a cellular automaton. An individual robot's behavior is reactive with respect to its neighbors, producing order among the entire group. Each unit need only be aware of its relative position and orientation to each neighbor in its neighborhood to calculate and realize a motion path to reach its desired position. Thus, a formation is established and maintained using a distributed reactive control architecture, requiring no global information or coordinate system, and no central leader.

An important aspect of this approach is that the robots are cells in a *robot-space cellular automaton* rather than a *world-space cellular automaton*. The latter assumes that the space in which the robots exist is topologically segmented into a grid of cells. The former views the robots that make up the structure as the cells in the automaton; in this sense, the automaton is an abstraction for the formation (Fig. 2). This approach overcomes many of the limitations of the world-space automaton, eliminating the dependence on the environment and reducing automaton complexity [6].

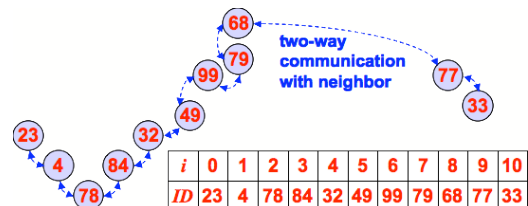


Fig. 2: Robots as cells in a 1-dimensional robot-space cellular automaton.

While the major goal of a formation control method is to attain an overall formation of robots, to be effective, the control architecture must also handle various real-world conditions to maintain or repair the formation. Fredslund & Mataric [5] define a number of criteria for evaluating the performance of the control architecture for robot formations. The first, *generality*, is the ability to define formations to conform to a variety of different shapes. The second is *stability*, or, the ability to maintain a formation over time. The third criterion is *robustness*, or, the flexibility and responsiveness of the formation to changes in group size. The fourth criterion is *dynamic switching capability*, or, the ability to switch from one formation to any other formation. The final criterion is *obstacle avoidance*, which evaluates an ability to deal with both large and small-scale obstacles.

To deal with some of these dimensions, our control architecture needs to be extended in a way that allows robots to dynamically determine which robots in the vicinity are relevant to the given task (we refer to these robots as *neighbors*). Previous work by Lemay *et al* [7] has been done on dynamically establishing relationships between robots attempting to initialize a formation. The method describes a neighbor discovery state, in which each individual builds a visibility table of all the robots within its sensing range. These tables are then cross-populated and shared among the group. Each individual robot then performs a bounded depth-first search looking for the best positioning with itself as the leader of the formation.

In contrast to a method that requires sharing of information among the entire group of robots, we propose a distributed auction-based approach to dynamic neighborhoods. Implementations of auctions for mobile robot coordination take many forms, ranging from the high-level frameworks for communicating the terms of auctions (such as the “contract net protocol” [8]) to the more specific application of auction algorithms to the allocation of tasks to a heterogeneous group of mobile robots [9]. A distributed auction method has the benefit of a decentralized implementation. While the solution is not guaranteed to be optimal, it is uniquely suited to a collection of mobile robots, in that each individual already possesses the knowledge needed to calculate its bid.

III. FORMATION CONTROL ALGORITHM

The inter-agent communication required by the robot-space cellular automaton is limited to a few key parameters. Each of N robots is represented as a cell c_i (where i refers to the i^{th} robot in the cellular automaton), which maintains a neighborhood h_i , a state s_i , a formation definition F , and a state transition function S (each described in this section):

$$c_i = \{h_i, s_i, F, S\} \quad (1)$$

A. Neighborhood

The neighborhood h_i of c_i consists of n neighbors in addition to c_i itself, where $n \leq n_{max}$ (Eq. 2). The entire automaton C can then be expressed as the union of all N neighborhoods (Eq. 3).

$$h_i = \{c_{i-n/2}, \dots, c_{i-1}, c_i, c_{i+1}, \dots, c_{i+n/2}\} \quad (2)$$

$$C = h_1 \cup h_2 \cup \dots \cup h_N \quad (3)$$

B. State

The state s_i of cell c_i is written as:

$$s_i = \{p_i, r_{i,des}, r_{i,act}, \Theta_i, \Gamma_i, t_i\} \quad (4)$$

The variable p_i represents the *formation-relative position*, a calculated value that corresponds to a position in the formation (used only for *calculating* relationships; does not necessarily correspond to a global physical robot location); $r_{i,des}$ and $r_{i,act}$ are sets of calculated desired and actual spatial relationships with all neighboring cells, respectively; Θ_i and Γ_i represent the accumulated rotational and translational error, respectively; and t_i represents the time step [6].

C. Formation Definition

A desired formation F is defined by f , a geometric description (i.e., a set of m mathematical functions); R , the desired separation between neighbors in the formation; Φ , the orientation of each robot relative to the formation (referred to as *formation-relative orientation*); p_{seed} , a formation-relative position that serves as the starting point in the formation from which relationships will propagate [6]:

$$F = \{f, R, \Phi, p_{seed}\} \quad (5)$$

This definition is sent to any single robot, which we refer to as the *seed* cell c_{seed} of the automaton. Note that c_{seed} is not a leader; rather, this action instigates changes in the state of the cell, which, in turn, propagates changes in its neighbors.

D. State Transition

In a cellular automaton, each successive state of a cell is dictated by a state transition function S , which considers the current state of the cell as well as the states of its neighbors. In a robot-space cellular automaton, cells are distributed units and, thus, each cell c_i must maintain the communicated state of its neighbors. Therefore, a robot must transmit its own state information within the neighborhood.

For purposes of determining relationships with neighboring robots, a cell considers itself to be at some formation-relative position p_i (Eq. 6); in the case of c_{seed} , the position p_{seed} is provided as the starting point from which the formation and relationships will propagate. The *desired relationship* $r_{i \rightarrow j, des}$ from c_i to some neighbor c_j is 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 (Eqs. 7 & 8; Fig. 3). Solving for $r_{i \rightarrow j, des}$ results in two intersections: one in the positive direction and one in the negative direction. These solutions define right and left neighbor relationships $r_{i \rightarrow i+1, des}$ and $r_{i \rightarrow i-1, des}$, respectively; these relationship vectors are then rotated by $-\Phi$, so that robot heading is consistent within the formation definition [6].

$$p_i = [x_i \quad f(x_i)]^T \quad (6)$$

$$R^2 = (v_x - p_{i,x})^2 + (f(v_x) - p_{i,y})^2 \quad (7)$$

$$r_{i \rightarrow j, des} = [v_x \quad f(v_x)]^T \quad (8)$$

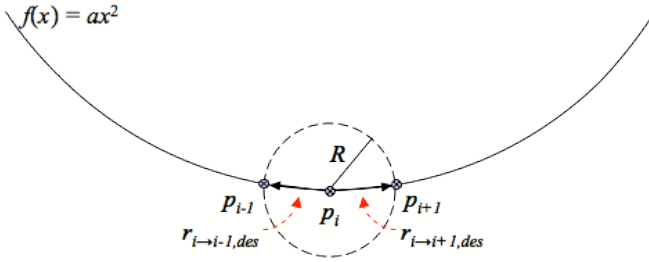


Fig. 3: Calculating desired relationships, r_{i-1} and r_{i+1} , from p_i to neighbors.

The formation definition and relationship information are communicated locally within the neighborhood. Each neighboring robot c_j repeats the process in a distributed, asynchronous fashion, considering itself to be at a different formation-relative position as determined by the propagated desired relationship from its neighbor (Eq. 9). The resulting relationship $r_{j \rightarrow i, des}$ is complementary to $r_{i \rightarrow j, des}$, in that it is equal in magnitude, but opposite in direction (Eq. 10). This property of the algorithm guarantees convergence and stability between two robots attempting to establish and maintain relationships with one another. The calculated desired relationships generate a connected graph that yields the shape of the formation (Fig. 4). Using only sensor readings, the i^{th} robot determines its *actual relationship* $r_{i \rightarrow j, act}$ with its neighbor c_j (described in [10]). Discrepancies between $r_{i \rightarrow j, des}$ and $r_{i \rightarrow j, act}$ are used to calculate corresponding rotational and translational errors, Θ_i and Γ_i , respectively [6]. Correcting for these errors produces robot movements that result in the desired formation [10].

$$p_j = p_i + r_{i \rightarrow j, des} \quad (9)$$

$$r_{j \rightarrow i, des} = -r_{i \rightarrow j, des} \quad (10)$$

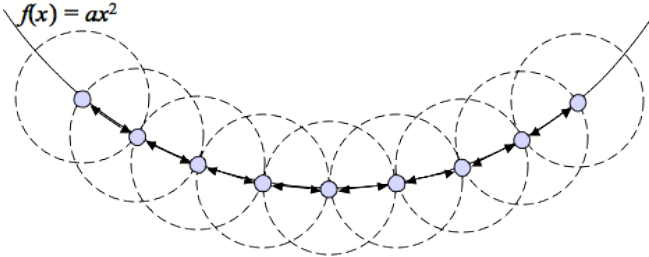


Fig. 4: Calculated relationships generate a parabolic formation.

E. Extended Formation Definition

With minor modification, the approach generates lattice formations defined by multiple mathematical functions [11]. The standard implementation of the algorithm produces a sort of “branching structure” due to the use of the p_i term in Equation 7; thus, we omit this term from the equation for purposes of lattices (note that p_i does not go away completely—it is still propagated in Equation 9):

$$R^2 = v_x^2 + f^2(v_x) \quad (11)$$

As an example of a formation that uses this alternative equation (Eq. 11), we consider a hexagonal lattice (i.e., an equilateral triangular lattice) structure by defining a formation by three functions: $f = \{f_1(x), f_2(x), f_3(x)\}$, where $f_1(x) = 0$, $f_2(x) = x\sqrt{3}$, and $f_3(x) = -x\sqrt{3}$. As cells calculate

desired relationships, they begin to develop common neighbors as neighborhoods intersect one another, and an emergent hexagonal lattice structure results (Fig. 5).

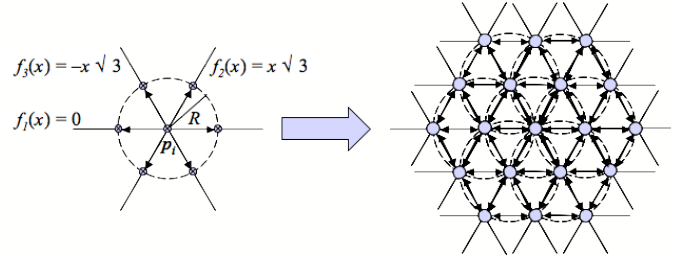


Fig. 5: A formation defined by three functions yields a hexagonal lattice.

IV. EVALUATION

The formation control algorithm was implemented on 20 custom-built robots. These robots are homogeneous—atop a two-wheeled ScooterBot II base sits an XBCv2 microcontroller with a color camera; an XBee wireless communication module with a custom interface to the XBC allows for inter-robot communication. A colored bar-coding system facilitates neighbor identification, with each robot featuring a unique “face”. Once a neighbor c_j is identified, the color camera is used to determine the distance and orientation to that robot and, subsequently, the actual relationship $r_{i \rightarrow j, act}$ [10].

For purposes of comparison and analysis, we base our evaluation of system performance on the criteria discussed in Fredslund & Mataric [5]. We consider the status of the formation at discrete time intervals that correspond to steps within the cellular automaton (i.e., the actual motion during state transitions is not relevant to our task domain).

A. Generality

The *generality* of a system refers to its ability to conform to a variety of different formations [5]. Analysis from various trials and experiments suggests that the algorithm is extremely general and has warranted a classification of the formations that can currently be produced:

- I. *Non-formation*—swarm
- II. *Explicit formation*—strict relationships are provided (like how a marching band is formed)
- III. *Straight line formation*—conform to the function $f(x) = a x$, where a is some constant; relationships calculated using Equation 7 or 11 (p_i -independent)
- IV. *Function-based formation*—defined by a single mathematical function $f(x)$; relationships calculated using Equation 7
- V. *Branching formation*—defined by multiple mathematical functions; relationships calculated using Equation 7
- VI. *Lattice formation*—defined by multiple mathematical functions; relationships calculated using Equation 11

Other potential classifications have also been identified, but have not yet been investigated; these include three-

dimensional and disconnected (disjoint) formations [6]. Formations falling into any of these categories can currently be realized provided that the neighborhood of each cell is given (Fig. 6). In Section V, an auction-based method for dynamically determining neighborhoods is discussed that improves the initialization of such formations.

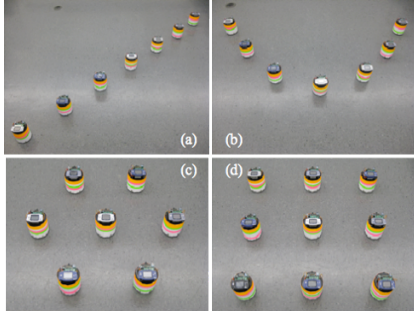


Fig. 6: Robots exhibiting generality, conforming to a variety of formations: (a) a line, (b) a parabola, (c) a hexagonal lattice, (d) a square lattice.

B. Stability

A system’s ability to maintain formation (once established) dictates its *stability* [5]. To test the control algorithm against this principle, we manipulated one or many robots (either via remote control or by hand) in a variety of different formations, changing both its position and orientation (e.g., Fig. 7).

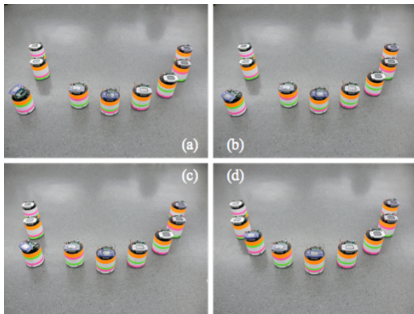


Fig. 7: An example of robots exhibiting stability; a “lost” robot as it finds its way to the correct position. Note that its neighbors maintain formation.

Following a search for its neighbors, the “lost” robot would recognize that it was out of position and then take a movement action to correct for this error. More significant, however, were the behaviors of its neighbors—their respective positions and orientations were maintained. This is a result of each neighborhood communicating the relative accumulated translational and rotational error of the system. Neighbors would detect discrepancies in their respective desired and actual relationships with the “lost” robot, but would also recognize that it was, indeed, the “lost” robot that was in error. Similar reactive formation control systems would propagate this error physically, responding to the displacement of a single robot with an action that would correct for the relative error, but, consequently, cause a chain reaction of movements, thus, displacing the structure as a whole [5]. Our work distinguishes itself in that this detected error is propagated through communication. The

algorithm exhibits strong stability in that any number of “lost” robots will reestablish their relationships in the formation while robots in their correct formation-relative position will maintain their pose.

C. Robustness

Evaluating *robustness* considers the ability of a system to respond to changes in group size [5]. Relying only on local information within an established neighborhood, it can be shown that the algorithm is independent of the number of robots; however, in the current implementation, the number of robots must be given at startup so that neighborhood of each robot can be initialized (i.e., all neighborhoods are specified manually). While this number can, indeed, be any value that corresponds to the true number of robots in the system, once set, this implementation is limited to *only* this number. This is not acceptable for most real-world applications. It is quite common for robots to be reassigned to different tasks or exhibit failure; likewise, as numbers begin to dwindle or the task changes, it is possible that other robot units join the ranks to increase these numbers. A robust formation control system must be tolerant of this. In Section V, we propose a method to address dynamic neighborhoods, thus, overcoming groups of static size.

D. Dynamic Switching Capability

We have already demonstrated that the algorithm is general in establishing a wide variety of formations. We now discuss its *dynamic switching capabilities*—the ability of the system to respond to an operator’s commands for changes in formation organization [5]. To manipulate the formation, a human operator need only send one of a variety of commands to any single robot. This propagates changes in the automaton, causing a chain reaction in neighbors, which then change states accordingly, resulting in a global transformation of the formation. Commands include formation translation, rotation, scaling, resizing, and change [6]. The ease with which an operator can manipulate formations with potentially large numbers of robots is a hallmark of this system. The current implementation, however, presents a limitation in formation transitions that require a change in the size of the neighborhood—recall that neighborhoods are currently specified manually (i.e., they are static) and, thus, cannot accommodate such scenarios. This limitation is clear when an operator commands a change from, say, a function-based formation (e.g., a parabola; two neighbors) to a lattice formation (e.g., a hexagon; six neighbors). Section V describes a method for dynamic neighborhoods to allow such a transition.

E. Obstacle Avoidance

The ability of a system to deal with both large and small-scale obstructions is *obstacle avoidance* [5]. This is yet to be addressed in the current implementation. In Section V, we discuss the implications of dynamic neighborhoods on group obstacle avoidance, as relationships between neighbors might be required to change to navigate an obstruction.

V. DYNAMIC NEIGHBORHOODS

Autonomous initialization and switching of formations, as well as changes to group size, all require dynamically tasking robots to fill a position in the formation. We propose a distributed auction method to accomplish this, and discuss its application to formation repair and obstacle avoidance.

A. Phase Transitioning

Transitioning from a swarm to a formation will be referred to as a *phase transition*, since it has many parallels to the state phase transitions observed in matter (Spears *et al* [12] also use this term in a similar context). For example, it is useful to think of a group of mobile robots with no particular programming for interaction with each other, aside from collision avoidance, as similar to matter in a gaseous state. That is, the “volume” (i.e., number of robots) and shape that the group takes on is determined by its surroundings. Robots in a swarm can be thought of as similar to matter in a liquid phase—the swarm is free to assume the shape of the boundaries of its surroundings, but will maintain a fixed volume. Robots in formation are similar to matter in a solid state, assuming both a fixed volume and a fixed shape. As seen in Figures 6 & 8, a formation can resemble a crystalline structure quite closely, with an internal configuration situated in a repeating lattice.

As in our previous example, we consider a hexagonal lattice formation; however, the phase transition method is generalizable to any formation definition. It is assumed that each robot has the ability to detect the distance and orientation of all other robots within its sensor range. In the current implementation this is accomplished using vision processing with a color camera [10]. Trilaterative localization is another example of such range and orientation sensing [13]. As in the current implementation, it is expected that there will be a number of robots within communication range. If no robots are in communication range, individual robots proceed to a given destination and can then initiate the phase transition.

B. Communication Model

A topic-based publish/subscribe model [14] is used to facilitate communication among a group of cells in the automaton and among robots wishing to join the formation. The publish/subscribe paradigm allows a distributed group of agents to send (i.e., publish) and receive (i.e., subscribe) various types of information regarding select topics of interest. This allows state information to be passed among neighbors, while still allowing broadcast messages to nearby agents. A topic is addressed with a unique identifier, which, for our purposes, is a formation-relative position p_j . This allows subscribers to discriminate between relevant and irrelevant published information. The content of a topic is the state of the cell c_j associated with p_j . We write the topic identified by this formation-relative position as:

$$T(p_j) = \{p_j, s_j\} \quad (12)$$

C. Auction Model

A cell c_i that requests information on $T(p_j)$ encounters one of three response conditions: (1) $\{p_j, s_j\}$, a response directly from c_j containing its current state information; (2) $\{p_j, \emptyset\}$, a response from the publisher regarding the topic, but containing no state information because c_j does not exist yet; and (3) $\{\emptyset, \emptyset\}$, no response regarding the topic, indicating that no other cell has expressed interest in $T(p_j)$ and c_j does not exist yet. In the first condition, the formation-relative position p_j is occupied by c_j . In the latter two conditions, p_j is unfilled, necessitating the auction process (Fig. 8).

In the case of Condition (3), c_i will announce an auction for p_j —denoted $A(p_j)$ —if and only if $\|p_i - p_{seed}\| < \|p_j - p_{seed}\|$ (for densest packing) and $\Gamma_i \approx 0$ (for stability). This cell, referred to henceforth as an auctioneer a , becomes the sole publisher of information on the topic $T(p_j)$ and will respond to any subsequent inquiries from subscribers to topic with $\{p_j, \emptyset\}$, thus, fulfilling Condition (2).

A robot that receives an auction message for p_j must determine whether or not it is advantageous for it to fill the position. This robot will bid if and only if (1) it has no neighborhood (i.e., it is not yet part of the automaton) or (2) its neighborhood is not full (i.e., $n < n_{max}$) and $\|p_j - p_{seed}\|$ is less than the magnitude of the difference between its current formation-relative position and p_{seed} . This robot, referred to henceforth as a bidder b , will announce a bid to fill p_j . This bid, denoted $B(p_j)$, consists of a weighted sum composed of the distance d from b to p_j (weighted by an *energy cost modifier* E) and the number of existing neighbors n in b 's neighborhood (weighted by a *relation cost modifier* X):

$$B(p_j) = E d + X n \quad (13)$$

The distance factor d impacts the efficiency of moving into the position. The auction $A(p_j)$ includes the relationship (i.e., range and bearing) of p_j relative to a . This information, coupled with the relationship from b to a , is used to determine the actual relationship of b to p_j . The number of existing neighbors n favors moving robots that are not already in the automaton or those that have fewer existing neighbors; however, it is sometimes advantageous for a robot to sever existing relationships to move into a new position (as will be seen in Formation Repair). The energy cost constant E and the relation cost constant X can be modified to minimize the time taken for phase transition or to minimize energy usage. These constants are currently tuned by empirical trials on the simulator; a more principled approach is being considered for future implementations.

After a period of time, the auctioneer a determines the winning bidder of $A(p_j)$ based on the minimum $B(p_j)$. The auctioneer passes publishing rights to this bidder, which is now identified as c_j . This cell, now associated with the position p_j , becomes the sole publisher on the $T(p_j)$, and will only ever respond to queries about the topic with $\{p_j, s_j\}$, thus, addressing Condition (1) above.

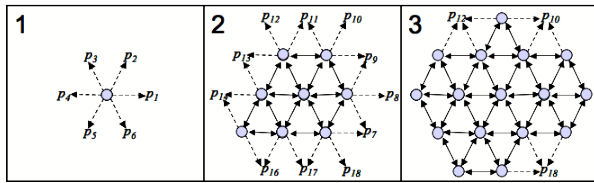


Fig. 8: Distributed auctioning of formation-relative positions.

D. Formation Repair

In the event of robot failure, it may be desirable for robots with established positions in the formation to break free and rejoin the formation in a new formation-relative position. As an example, we consider a scenario in which a cell has been disabled (indicated with an “X” in Fig. 9-1). The formation repair process is initiated upon the confirmation of the loss of communication with the robot. Neighbors auction the position just as in the initial formation process (Fig. 9-2).

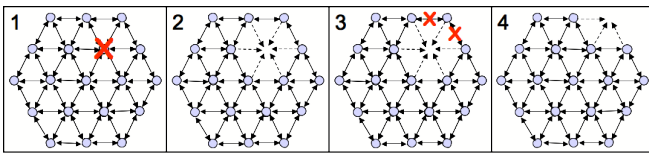


Fig. 9: The formation repair process: (1) a robot malfunctions and is lost, (2) the position is auctioned, (3) a cell wins the auction and breaks its current relations, (4) the winner takes the position.

As before, a bid for a position considers the distance to travel, as well as the number of neighbor relations that would be broken. In the case of the latter, if the winner of an auction must leave an already existing neighborhood, it notifies its neighbors that it will be vacating its current formation-relative position by unsubscribing to its topic. This robot then proceeds to its new position, and begins the auctioning process for its neighbors (Fig. 9-4). Assuming that there is no single robot free of the formation, this same process repeats for each vacant position. The conditions for determining when to bid—specifically, the second condition—guarantees that open positions closest to the seed are filled before farther positions (i.e., densest packing) [6].

Note that, in real-world scenarios, a disabled robot will not simply disappear (as in Figure 9); rather, it will become an obstacle and, thus, a method for its avoidance is necessary.

E. Group Obstacle Avoidance

The phase transition metaphor suggests a method for avoiding obstacles. As robots on the perimeter of the formation encounter an obstacle, they may loosen their “bonds” (relationships; inspired by molecular bonds) to their neighbors (much like solid ice melting into liquid water). For a widely spaced formation encountering a relatively small obstacle—perhaps something no more than twice as big as the robot itself—the neighbor bonds may be loosened, but not broken, allowing the robots in the path of the obstacle to shuffle out of the way. For tighter formations or larger obstacles, it may be necessary to “melt” local portions of the formation, breaking or temporarily ignoring relations with neighboring robots in order to flow out of the way of the obstacle. Once the robot determines itself to have passed the influence of the obstacle, it either reestablishes its

temporarily ignored relationships or bids on open positions in the formation. Robots that have “melted” revert to Reynolds-like rules of swarming [4]. If the entire formation must be melted in order to facilitate obstacle avoidance, then the phase transition method may be employed once again to precipitate the formation from the swarm of robots.

VI. CONCLUSIONS AND FUTURE WORK

An analysis of our cellular automata-based robot control architecture was presented with respect to characteristics necessary for practical applications. The distributed, reactive nature of the architecture provides generality, stability, robustness, and controllability. An extension to the architecture provides a method for robots to form dynamic neighborhoods. It was shown that this extension could manage initial formation creation, formation repair, and obstacle avoidance. This approach is currently being implemented in the simulator described in [6] and will be ported to the robots described in [10] for further analysis.

REFERENCES

- [1] 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,” Arlington, VA, 4-7 April 2000.
- [2] M. Aung, A. Ahmed, M. Wette, D. Scharf, J. Tien, G. Purcell, M. Regehr, B. Landin, “An overview of formation flying technology development for the terrestrial planet finder mission,” in *Proc. of IEEE Aerospace Conference*, vol. 4, 13 March, 2004, pp. 2667-2679.
- [3] ESA Darwin flotilla, Illustration, 2002, retrieved from http://www.esa.int/esaSC/120382_index_0_m.html on 9-9-08.
- [4] C. W. Reynolds, “Flocks, herds, and schools: a distributed behavioral model,” in *Computer Graphics, SIGGRAPH '87 Conference Proceedings*, 1987, pp. 25-34.
- [5] J. Fredslund, M. J. Mataric, “Robots in formation using local information,” in *7th International Conference on Intelligent Autonomous Systems*, Marina Del Rey, California, 2002, pp. 100-107.
- [6] R. Mead, “Cellular automata for control and interactions of robots in large formations,” M.S. Thesis, Dept. Computer Science, Southern Illinois University at Edwardsville, Edwardsville, Illinois, 2008.
- [7] M. Lemay, F. Michaud, D. Letourneau, J. Valin, “Autonomous initialization of robot formations,” in *Proceedings of ICRA '04, 2004 IEEE International Conference on Robotics and Automation*, vol. 3, 26 Apr-1 May 2004, pp. 3018-3023.
- [8] R. Smith, “The contract net protocol: high level communication and control in a distributed problem solver,” in *IEEE Transactions on Computers*, vol. C-29, no. 12, 1980, pp. 1104-1113.
- [9] B. P. Gerkey, M. J. Mataric, “SOLD!: auction methods for multirobot coordination,” in *IEEE Transactions on Robotics and Automation*, vol. 18, issue 5, Oct. 2002, pp. 758-768.
- [10] R. Mead, J. B. Weinberg, and J. R. Croxell, “A demonstration of a robot formation control algorithm and platform,” in *Proceedings of the Robot Competition and Exhibition of the 22nd National Conference on Artificial Intelligence*, Vancouver, BC, July 2007, pp. 23-27.
- [11] R. Mead, J. B. Weinberg, “A distributed control algorithm for robots in grid formations,” to appear in the *Technical Report of The 2008 Association for the Advancement of Artificial Intelligence (AAAI-08) Workshop on Mobile Robotics*, Chicago, IL, July 2008.
- [12] W. Spears, D. Spears, J. C. Hamann, R. Heil, “Distributed, physics-based control of swarms of vehicles,” *Autonomous Robots*, vol. 17, 2004, pp. 137-162.
- [13] R. Heil, “A trilaterative localization system for small mobile robots in swarms,” M.S. Thesis, Dept. Electrical and Computer Engineering, University of Wyoming, Laramie, Wyoming, August 2004.
- [14] P. Eugster, P. Felber, R. Guerraoui, A. Kermerrec, “The Many Faces of Publish/Subscribe,” *ACM Computing Surveys (CSUR)*, vol. 35, issue 2, 2003, pp. 114-131.