A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication

Jakob Fredslund and Maja J. Matarić

***Abstract*—We study the problem of achieving global behavior in a group of distributed robots using only local sensing and minimal communication, in the context of formations. The goal is to have N mobile robots establish and maintain some predetermined geometric shape. We report results from extensive simulation experiments, and 40+ experiments with four physical robots, showing the viability of our approach. The key idea is that each robot keeps a single *friend* at a desired angle θ, using some appropriate sensor. By *panning* the sensor by θ degrees, the goal for all formations becomes simply to center the friend in the sensor's field of view. We also present a general analytical measure for evaluating formations and apply it to the position data from both simulation and physical robot experiments. We used two lasers to track the physical robots to obtain ground truth validation data.**

***Index Terms*—Local sensing, minimal communication, multiple robot coordination, robot formations.**

I. Introduction

In THIS paper, we have studied the problem of achieving global, coordinated behavior in a group of distributed robots using only local sensing and minimal communication. As an instance of this general problem, we have considered *formations*. We study how to achieve global-level formation coordination *without* providing the robots with global knowledge of other robots' positions or headings. We present an algorithm that displays generality in the number of formations it can produce, stability of established formations, robustness to changes in group size, dynamic switching between formations, and obstacle avoidance, and we validate it through extensive simulation and real robot trials. Finally, we evaluate the performance of the algorithm with a formalized formation evaluation measure.

The rest of this paper is organized as follows. Section II gives an overview of related work. Section III presents our algorithm and evaluation measure, and Section IV describes an implementation of the algorithm. In Section V, we present and evaluate our experimental results. Section VI discusses our approach and concludes the paper.

Manuscript received March 17, 2001; revised February 21, 2002. This paper was recommended for publication by Associate Editor E. Pagello and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by DARPA under Grant DABT63-99-1-0015, and by the Office of Naval Research under Grant DURIP N00014-00-1-0638 and Grant N00014-01-1-0354. This paper was presented in part at IEEE LIRA, Banff, AB, Canada, July 29–August 1, 2001, and at IAS7, Marina Del Rey, CA, USA, March 25–27, 2002.

J. Fredslund is with the Department of Computer Science, Aarhus University, DK-8000 Århus C, Denmark.

M. J. Matarić is with the Interaction Laboratory, University of Southern California, Computer Science Department, Los Angeles, CA 90089 USA.

Digital Object Identifier 10.1109/TRA.2002.803458

II. Related Work

A variety of approaches have been proposed to create global behavior in a group of mobile robots. [11] showed how a set of simple behaviors (avoidance, aggregation, and dispersion), based on local sensing only, can be combined so that a global flocking behavior emerges, and demonstrated the behaviors on a group of 13 mobile robots. [16] demonstrated a robot soccer-playing team with a minimalist, behavior-based control system. By combining a few basic behaviors, two different group formations of three robots emerged. In [9], as few as two basic, local behaviors (avoidance and goal seeking) were shown in experiments with five physical robots to be enough to result in a successful collaborative box-pushing behavior. [17] presented work where a group of real robots, with only touch sensors, were able to form a physical chain reaching from a nest to a source of food, and use it to collect and transport food. [2] presented theoretical work where a large set of robots, represented as points in the plane, congregated at a single position. Moving synchronously in discrete time steps, robots iteratively observed neighbors within some visibility range V, and followed simple rules to update their position.

In contrast to all these demonstrations, formations require a more precise and reliable spatial structure. In [14], a group of simulated robots formed approximations to circles and simple polygons, using global knowledge of all robots' positions. Each robot oriented itself to, e.g., the furthest and nearest robot. In [5], a similar setup was presented, but group motion was also considered, e.g., a matrix formation performing a right turn. In [10], a formation is defined by a so-called *virtual structure* (VS). The algorithm assumed that all robots had global knowledge; it iteratively fit the VS to the current robot positions, displaced the VS in some desired direction, and updated the robots' positions.

In [3], three principles of formation control are identified: *unit-center referenced*, *leader referenced*, and *neighbor referenced*. In the first, each robot decides its position relative to the centroid of all robots; in the second, the robot uses the position of a predetermined leader; and in the third, the robot's nearest neighbor is used as a reference point. The work was demonstrated in experiments using physical robots with odometry, GPS, and global broadcast of the robots' coordinates. In [7], each robot was controlled using local information, either by referencing itself to one neighboring robot and maintaining a certain distance and angle to it ($l-\psi$ $control$), or to *two* neighbors and maintaining two fixed distances to those ($l-l$ $control$). The algorithm was demonstrated in simulation. The $l-\psi$ $control$ was adopted in [1] as part of the *Leader-Following* behavior. In an experiment with two physical robots, the follower (using a color camera and color-blob tracking) kept a fixed heading and distance to the leader. In [4], simulated robots had a set of *attachment sites* defined uniformly around the body, allowing the group to "snap" into shape as robots were "pulled" toward each other's sites. However, multiple configurations with the same attachment sites were possible. In [13], simulated robots were represented as points in the plane; spatial patters emerged from using local force laws between neighbors. The multiple-configuration problem (as in [4]) was solved by adding special labels to the particles that others could detect.

There is a spectrum of strategies, ranging from simple, behavior-based, purely local ones out of which global formations emerge, to more involved ones relying, to varying extent, on global knowledge, typically a global coordinate system and/or knowledge of other robots' positions and headings. The former category is characterized by minimalism and robustness but a lack of guarantees that the desired formation will actually emerge; the latter category is characterized by reliability and efficiency but also a need for global knowledge and computation. [11], [16], [9], [17], [14], and [4] all share the feature that a guarantee of the emergence of a desired formation cannot be given (in some cases simply because it is not the goal of the work). We have addressed this problem by assigning a unique "right spot" in the formation to each robot. Thus, when all robots are in place, the desired formation is established. In [3], [5], and [10], global knowledge is assumed: each robot knows the position of all others. In our approach, robots only use local sensing and do not share a common coordinate system. [2] and [13] represent robots as points and show simulated experiments with discrete time steps of synchronous robot movements. Each point reacts to all points within some distance of it. In our algorithm, each robot references itself locally to one neighboring robot and keeps a certain bearing and distance to it, as in [1], [6], and [7], but in contrast to this work, we have demonstrated our algorithm through both simulation and real-world experiments, and with different numbers of robots doing various formations. Bearing this spectrum of movement-coordination strategies in mind, we have in our approach sought simplicity yet reliability through local sensing with minimal communication.

III. Our Algorithm

Our key idea is simple: every robot in the group positions itself relative to a designated neighbor robot, its *friend*. The friend references itself to another friend, and so on. Thus, each robot needs to be able to determine the distance and angle to its friend. We assume each robot has some sort of sensor, referred to as the *friend sensor*, for this purpose. The friend sensor is assumed to be pointing forward with a $\pm 90^\circ$ field of view, and to have limited visibility. Each robot has a unique ID, detectable by other robots through the friend sensor. Each robot *broadcasts* its ID regularly as a heartbeat message. As a result, each robot knows how many robots (N), and which ones, are participating in the formation. The robots are always organized in a *chain of friendships* by the order of their IDs. One robot is the *conductor*, deciding the formation and its heading, and thus, not following any friend.[1] The conductor broadcasts a message indicating the current formation, f. The conductor does *not* broadcast its heading. Any robot can potentially serve the conductor duty; who is conducting depends on N and f.

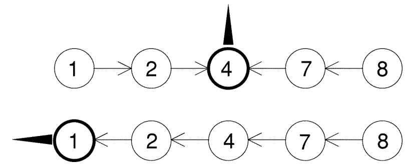


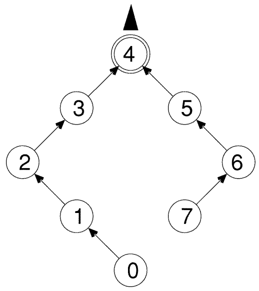
Fig. 1. Chain of friendships: $i \rightarrow j$ means i looks for j (j is i's friend). The black triangle indicates the formation heading. In the centered line formation on top, the robot with the middle ID (4) is the conductor. For the noncentered column formation on the the bottom, the robot with the lowest ID (1) is the conductor.

Our algorithm applies to a particular class of formations. The formation shape has to be folded from the chain of friendships, so it can have, at most, two "loose ends." Further, it cannot be "frontally concave," i.e., make a backward curve with respect to the heading. With only one conductor, some robots would have to look back for their friends, and by the assumption of a forward $\pm 90^\circ$ field of view of the friend sensor, this is not possible, and is disallowed. Consequently, shapes like `J`, `U`, and $\sim$, with upwards heading, are not allowed in our algorithm. With this restriction, the possible shapes are essentially those that can be folded from an open bicycle chain, keeping either the middle or the end of the chain in front. All robots are *a priori* given a comprehensive list of such definitions of shapes that the group is able to establish.

If the middle of the chain is in front, the formation is *centered*; otherwise it is *noncentered*. For centered formations, the robot in the middle of the chain of friendships, and thus, in front of the formation, is the conductor. In contrast, in noncentered formations, the conductor has to be at one end of the chain of friendships to be in front. For centered formations, robots with IDs *less* than the middle ID find a friend with an ID that is immediately *greater* than their own; robots with IDs *greater* than the middle ID find a friend with an ID that is immediately *less* than their own. For noncentered formations, all robots find a friend with an ID that is *less* than their own. See Fig. 1 for a pictorial description.

A switch between two centered or two noncentered formations (i.e., between two formations of the same kind), is a Type A switch, and a switch between a noncentered and a centered formation, or vice versa (i.e., between two formations of different kinds), is Type B. For Type A switches, all robots keep the same friend and the conductor stays the same. For Type B switches, however, a new robot becomes the conductor, and all robots between the old and the new conductor must find a new friend; this is similar to a transition between nonisomorphic control graphs in [7]. Such a change is a special case of our algorithm (see Section V-E). To illustrate, imagine the robots switching between the two formations in Fig. 1. Robot 2 would have to find a new friend; for centered formations, its friend is 4, for noncentered, it is 1.

[1]The term *conductor* is analogous to *leader* in the literature. Each robot follows a friend, so all robots (except the boundary cases) serve as "local leaders," and all are also followers (except the conductor).




```
n = ⌈N/4⌉
if lessThanMe < ⌊N/2⌋ - n
  θ = 45
else if lessThanMe < ⌊N/2⌋
  θ = -45
else if lessThanMe > ⌊N/2⌋ + n
  θ = -45
else if lessThanMe > ⌊N/2⌋
  θ = 45
```

Fig. 2. Calculating the friendship angle in a diamond with eight robots. The diamond is a centered formation, so the robot with the median ID ($lessThanMe = \lfloor N/2 \rfloor$) is the conductor. Arrows indicate angle to friend, filled triangle shows formation heading.

The angle a robot needs to keep to its friend depends on its place in the chain of friendships and on the particular formation. By N, f, and its own ID, each robot can determine this angle *locally*. For all formations, a set of very simple rules, applicable to any group size N, determines for each robot the angle it should keep to its friend (see an example in Fig. 2). To alleviate this calculation, each robot maintains a local value called $lessThanMe$. What matters is really not the robot's ID, but its *rank* in the chain of friendships, which is sorted by IDs. The value $lessThanMe$ is precisely this rank, stating how many robots are currently alive with IDs less than this robot's ID. Thus, the $lessThanMe$s are consecutive. For centered formations with N robots, the conductor is the one with a $lessThanMe$ of $\lfloor N/2 \rfloor$. For noncentered formations, the conductor is the robot with a $lessThanMe$ of 0.

For example, in a diamond, the robots form four perpendicular line segments (see Fig. 2). The low fourth of the robots (0 and 1 in the same figure) form segment I by keeping their respective friends at a 45° angle to the front and left; the second-lowest fourth (2 and 3 in the figure), who are in front of the low fourth, form segment II by keeping their friends 45° to the front and right. The middle robot, 4, is the conductor, and the upper two fourths look left (5 and 6, segment III) and right (7, segment IV) for their friends, respectively.

Since frontally concave formations are not dealt with, a robot will only have to keep its friend in a range of $\theta \in [-90, 90]^{\circ}$ angles with respect to its heading. However, this could still pose a problem; at the extremes, the friend is likely to fall outside the field of view of the friend sensor. Also, for each angle θ, the algorithm for keeping the friend at θ would essentially be different. For these two reasons, we assume that the friend sensor can be *panned*, and use the following approach: instead of keeping its friend at angle θ, the robot pans its friend sensor θ degrees and simply *keeps its friend in the center of the field of view*. Thus, a buffer zone is introduced at the angular extremes of −90° and 90°, and the friend is more likely to stay within view. Moreover, a useful structure immediately emerges, described next.

Based on N, f, and $lessThanMe$, each robot determines the angle θ to keep relative to its friend and pans its friend sensor to that angle (called *camangle*). This represents the highest level of control abstraction. At the next level, the core of the algorithm works to center the friend in the view of the friend sensor. It does so with no regard to the current *camangle*; this has been

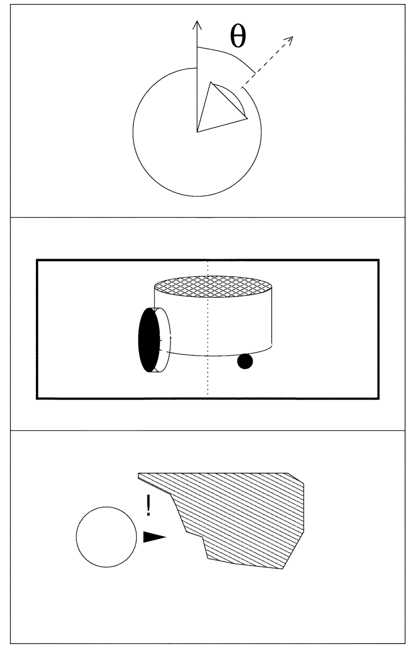


Fig. 3. Three levels of abstraction.

abstracted away at the level above. Finally, at the lowest level, basic collision avoidance is introduced, if necessary, by means of other sensors. This layered structure is depicted in Fig. 3.

There are several nice implications resulting from this approach. First, once the conductor starts moving, the only way for a robot to keep a stable position relative to its friend is by finding the same heading. Thus, the conductor "drags" the whole formation into place just by moving along, much like when one picks up a mobile that has fallen to the floor. No global heading needs to be agreed upon; it emerges automatically. Since any robot can be the conductor, what seems a centralized element really is not: if the conductor fails, another robot takes over the role.

Second, since the algorithm is basically "keep your friend in the center," a switch between two centered or two noncentered formations is easily done by gradually panning the friend sensor to the new appropriate angle; the change in position results automatically. As discussed above, switching between a noncentered and a centered formation is somewhat more involved.

Third, no global coordinate system is used and hence, no communication of coordinates and headings; only minimal heartbeat messages and formation messages are communicated. Yet, a formation is a global phenomenon; at some point in the system, some mechanism has to introduce the globality. Keeping the chain of friendships sorted by the unique IDs is a way of enforcing a global structure on the group, through only local sensing and minimal communication. Once the sorted chain is established, it is easy to keep it sorted. The global structure is what solves the agreement problem of selecting a conductor, and it eliminates ambiguity about who goes where; all robots know their spot, and this guarantees that the right formation is established—and consequently, the friendship angle calculations are very simple. Without IDs and without sorting the robots, it would be nontrivial to acquire the same characteristics, and at least a potentially significant increase in communication would be needed.

Fig. 4. A robot with laser and panned camera, the lens peering out through the hole in the color helmet.

IV. Implementation

In our implementation, the friend sensor is based on the combined data from a laser scanner and a color camera: the camera identifies the friend, the laser gives the distance to it. However, only the camera can pan. Friend identification is done using color-blob detection; each robot wears a unique helmet with two fluorescent color stripes. The stripes are clearly visible to other robots, and by their color and order, the corresponding ID can be inferred (see Fig. 4).

Our controller consists of a module holding state data and three concurrent behaviors; two handle communication and update the state data, and the main behavior controls the wheels. Each robot receives a one-byte formation message from the conductor every 2 s and updates the current-formation variable, f, if necessary (a human supervisor can issue a formation change on the same communication channel). When a robot detects an obstacle in its path (see below), it swerves to avoid it and sends out a *swerve* message with its ID and a value indicating the direction and angle of its turn. Other robots, not necessarily sensing the obstacle, react to this message by making a swerve of solidarity of the same angle and direction, if the sender is swerving toward them.

A robot R moves by setting two parameters: translational and rotational speed ($tspeed$ and $rspeed$). In the main behavior, R cycles through a control loop that reads the sensors, sends out its heartbeat message (2 b/s), and sets $tspeed$ and $rspeed$ to their default values. If R is the conductor, it modifies neither $tspeed$ nor $rspeed$, unless it has just circumnavigated an obstacle or made a swerve of solidarity. In that case, it gradually modifies $rspeed$ so as to return to the heading it had before the interruption. If R is not the conductor, it first locates its friend, F. It looks up F's ID and deduces the color helmet to look for. If other robots are swerving its way to avoid obstacles, R makes a swerve of solidarity. Continuously, R makes small corrections to $rspeed$ and $tspeed$ through a few simple rules so as to center F in the image and keep the right distance with no regard to the current *camangle*. The following is a pseudocode example of such a rule; it uses the variables $distError$ (the distance error: actual distance–desired distance) and $friendOffset$ (F's offset from the center in R's camera image), and some thresholds and constants as follows:

```
if (distError > 250 millimeter)
  rspeed += (friendOffset)*4
if (behind)
  tspeed += distError/3
```

If the first condition is true, R is too far from F, so it turns toward F by making a correction to $rspeed$ proportional to F's offset from the center of the camera image. Then, if R has fallen behind F, it speeds up by making a correction to $tspeed$ proportional to the distance error. The $behind$ Boolean is calculated using the camera pan direction and $friendOffset$; it is true if F is in the left half of the image while R's camera is panned right, or vice versa, false otherwise.

In collision avoidance, the so-called $aheadbuffer$ plays a central role. Using $tspeed$ and $rspeed$, a bounding box for the resulting movement is calculated, and a buffer zone is added around the box: `robot-size` on each side and $aheadbuffer$ on the front end. If any obstacles are found within this buffer, a correction is made to $tspeed$ and $rspeed$ proportional to the proximity of the nearest sensed obstacle.

Thus, $aheadbuffer$ induces immediate collision avoidance. In addition, if set to a high value, it allows R to look far ahead for obstacles, resulting in an elegant, smooth avoidance behavior. For robots with no others in front, $aheadbuffer$ is set to a high value proportional to `robot-size` and N; a large formation needs more space to negotiate obstacles than a small one. As a safety measure, another sensor (in our case, sonar) can be used for lowest-level collision avoidance.

Finally, we assume the robots start out in the right order with respect to the chain of friendships, but not necessarily with their respective friends in the field of view. This is reasonable, given that the problem of aggregating robots into such a configuration has already been empirically demonstrated [11], and it is possible through only local interaction to have N robots form a chain [17].

V. Experimental Evaluation

To validate our approach, we performed extensive experiments, both in simulation and with physical robots. Our

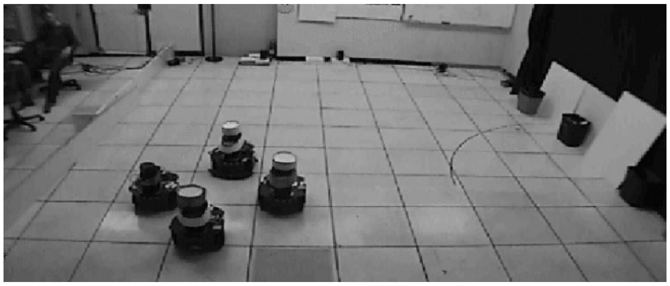


Fig. 5. One author and four robots in our lab arena.

simulation experiments were done using *Player* and *Stage*.[2] *Player* is a server and protocol that connects robots, sensors, and control programs across a network; *Stage* simulates a set of Player devices [8], [15]. In the real robot experiments, we used the ActivMedia Inc. Pioneer2 DX robots with the SICK LMS200 laser, sonars, the Sony PTZ camera, and running *Player* software [8]. To enable the detection of IDs, we constructed the above-mentioned brightly colored helmets and placed them around the camera, atop the laser (see Fig. 4). To obtain ground truth data for evaluating system performance, we set up a laser-based metrology system. Two lasers were placed at the boundaries of the experimental workspace, for tracking the robots, and a reflective marker was placed on top of every robot's helmet. This system (written by Andrew Howard) allowed us to track the individual robots as they moved, and record their positions for analysis. A Pioneer robot is approximately 50 cm wide. Due to the limited space available in our lab's 4 $\times$ 6-meter arena, we had the robots move very slowly in order to allow enough time to get in formation, and for trailing robots to catch up (see Fig. 5).

A. Evaluation Criteria

We developed the following *formation evaluation criteria* as a means of quantitatively judging the notion of being in formation:

Definition 1: Given the positions of N mobile robots, an interrobot distance d_{desired}, a desired heading h, and a connected geometric shape $\mathcal{G}$ completely characterizable by a finite set of line segments and the angles between them, the robots are considered to be in formation $\mathcal{G}$ iff:

1) *uniform dispersion*: $\exists d$, such that $\forall$ pairs of immediate neighbors (R_{i_1}, R_{i_2}) with distance $dist(R_{i_1}, R_{i_2})$, $|d - dist(R_{i_1}, R_{i_2})| < \epsilon_{d_1}$, and $|d - d_{desired}| < \epsilon_{d_1}$;
2) *shape*: $\exists$ a "stretching function" f with $f(\mathcal{G}) = \tilde{\mathcal{G}}$, such that $\forall$ angles $\theta \in \mathcal{G}$, $|f(\theta) - \theta| < \epsilon_a$, and such that $\forall$ robots R_i, with distance $dist(R_i, \tilde{\mathcal{G}})$ to $\tilde{\mathcal{G}}$, $dist(R_i, \tilde{\mathcal{G}}) < \epsilon_{d_2}$;
3) *orientation*: $|f(h) - h| < \epsilon_a$; for small $\epsilon_{d_1}, \epsilon_{d_2}, \epsilon_a > 0$.

Criterion 1 states that the same distance should be kept between all neighboring robots. Criterion 2 states that it should be possible to lay out the desired shape over the positional data and adjust the angles so that all robots are close to this shape. No angle in the original shape must be stretched more than ϵ_a to make the data points fit. Allowing the heading of the shape to be one of its defining angles, Criterion 3 states that the stretching from Criterion 2 must not skew the heading more than ϵ_a.

By using the term "immediate neighbor," *Definition 1* does not demand completeness of formations. For example, six robots can form a diamond. Importantly, the measure above is *global*. For example, N robots are not considered to form a line even if the angular offset between neighboring robots is small, but overall they form, say, an arc. In other words, even if the behavior of a robot is locally meaningful, it may not be so when considered globally [2]. *Definition 1* is easily extended to shapes that do not consist of line segments, such as circles; we omit these special cases for clarity.

B. Experimental Design

Our algorithm applies to a very general set of formations. In the real robot experiments, following [3], we focused on line, column, wedge, and diamond, but in simulation we performed experiments with line, column, wedge, diamond, circle, and miscellaneous asymmetrical shapes. However, to be consistent between simulation and real-world experiments, here we report only results for line, column, wedge, and diamond. Thus, below, any mention of formations refers to those four, unless stated otherwise.

We designed our experiments so as to validate the following properties and capabilities: generality, stability, robustness, switching between formations, and obstacle avoidance. To display stability, the group of robots must be able to *establish* any formation from some arbitrary initial configuration (within the boundaries of the above assumption), and to *maintain* the formation over time, traveling through an environment with no obstacles. To display robustness, the group of robots must be able to adapt any formation in case of one or more robots failing. To display formation switching, the group of robots must be able to perform a switch between any two formations in an environment with no obstacles. To display obstacle avoidance, the group of robots must be able to negotiate different types of obstacles, in any formation, either by maintaining the formation while avoiding the obstacle, or by reforming the formation after passing the obstacle.

In the following sections, we report our experimental results. In *all* experiments, both simulated and real world, we used *Definition 1* to determine whether the robots were in formation. We also used the notion of the distance traveled by the conductor up to the point when the robots established a **f**ormation **t**he **f**irst **t**ime, called ft.

C. Stability Results

In simulation, the "establish-formation" and "maintain-formation" components were done simultaneously. The robots started out in the right order next to each other, but with interrobot distance of 70 cm, and with random headings (except for the conductor, whose heading was correct). With four robots, we ran 10 trials for each formation. In every trial, the robots had to establish and maintain the formation, traveling 19.0 m. We then measured the percentage of time-steps the robots were in formation after ft. From the percentages of the 10 trials, we report the weighted average percentage, and the standard deviation σ. Using *Definition 1* (here and in all other

[2]Player and Stage were developed jointly at the USC Robotics Research Labs and HRL Labs and are freely available under the GNU General Public License from http://playerstage.sourceforge.net.

TABLE I
SIMULATION: STABILITY, WEIGHTED AVERAGES OVER 10 RUNS

$N=4$ 19.0 meters	ft (meters)	av.% time in formation after ft (σ)
line	0.7	88.4 (4.9)
column	3.0	100.0 (0.0)
diamond	2.6	100.0 (0.0)
wedge	1.8	92.0 (12.7)

simulation experiments reported in the following sections), we set $\epsilon_{d_1} = \epsilon_{d_2} = 10\%$ of the desired interrobot distance ($d_{\text{desired}} = 60$ cm), and $\epsilon_a = 3.6°$. Results are shown in Table I.

The large standard deviation for the wedge in Table I is due to one run where the percentage of time in formation after ft was only 54.6. If this possible outlier is excluded, the percentage and standard deviation are 95.3 and 3.7, respectively.

Next, we evaluated the stability of the approach with real robots. As our lab space is quite small, there was not enough room for the robots to first establish a formation, and then maintain it for a significant amount of time. Therefore, we had to separate the "establish-formation" and "maintain-formation" portions to validate stability.

With "establish-formation," the same was done as in simulation. In the initial configuration, the robots were placed in the right order, next to each other but with random headings, and they then established the formation within the boundaries of our 4×6-meter lab arena. We verified this for all formations several times. Fig. 6 shows four trials.

For "maintain-formation," the robots were first placed close to the desired formation, thus saving most of the time required for "establish-formation." Then, we again recorded the percentage of time they were in formation after actually establishing it; i.e., the percentage of time in formation after ft. As trials were very time consuming, only five trials with each formation were done. Instead of standard deviation, we simply show the percentages of all five trials, together with their weighted average, in Table II.

In all real robot experiments (here and in the following sections), we used *Definition 1* with $d_{\text{desired}} = 80$ cm, $\epsilon_{d_1} = (0.20 * d_{\text{desired}})$, $\epsilon_{d_2} = (0.08 * d_{\text{desired}})$, and $\epsilon_a = (0.08 * 2\pi)$. In other words, the dispersion of robots was set fairly loosely, allowing the average interrobot distance to differ up to 20% from d_{desired}. Criterion 2 was applied more strictly: all robots had to be, at most, 8% of d_{desired} from the line segments they belonged to.[3] Angles were allowed to deviate up to 28.8°. We used a simple line-fitting algorithm to fit data points to straight lines for evaluation. Tables I and II demonstrate that the column and the diamond are quite stable. The wedge seems to be either very stable or rather unstable, and the line is unstable. These two formations include ± 90 angles which turned out to be problematic; we explain why in Section VI.

Having thus addressed stability, we posit that for robustness, formation switching, and obstacle avoidance, it suffices to show that upon initially establishing the formation, the robots do get back in formation after the occurrence of failure, switch, or obstacle, respectively. In other words, in these three categories of experiments, a trial is a success if the robots establish the formation, overcome some incident, and reestablish an adapted formation within some preset distance.

[3]For the line, we set $\epsilon_{d_2} = (0.30 * d_{\text{desired}})$. Otherwise, at least one robot would often find itself too far from the best-fit line.

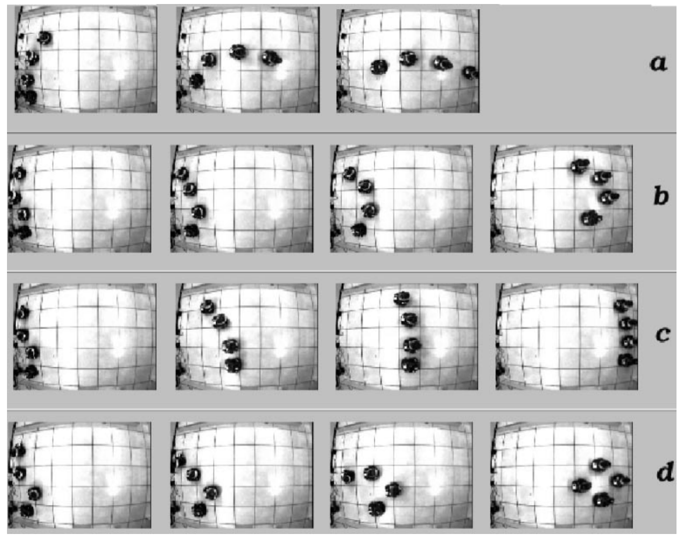


Fig. 6. Sequences of overhead images, order is left to right. Establishing a column (**a**), wedge (**b**), line (**c**), and diamond (**d**).

TABLE II
REAL ROBOTS: STABILITY. VALUES ARE % OF TIME IN FORMATION AFTER ft. FIVE TRIALS AND WEIGHTED AVERAGE

N=4	T1	T2	T3	T4	T5	Av.
line	94.1	46.8	70.4	34.7	64.9	60.9
colm	96.1	100.0	98.9	100.0	100.0	99.2
diam	75.0	99.7	88.6	99.8	100.0	91.9
wedg	100.0	99.9	99.6	50.3	49.8	78.2

D. Robustness Results

The structure of the simulation experiments addressing robustness is as follows. The robots establish the formation, one or two robots are terminated when the conductor has traveled 6.0 m, and the robots then establish an adapted formation. We tested this by showing that the initial formation is established (at ft_1) before the conductor has traveled 6.0 m, and an adapted formation is achieved (at ft_2) before the conductor has traveled 19.0 m. Thus, ft_1 is less than 6.0, and ft_2 is greater than 6.0 and less than 19.0. Table III shows results from two experiments with six robots; 10 trials were done of each.

Once the conductor reached the 6.0 m mark, one or two robots were terminated. For the circle (i.e., hexagon), the conductor itself was terminated. The remaining five robots themselves realized this (having stopped receiving the heartbeat from the conductor), promoted a new conductor, and eventually reached an adapted circle (a pentagon) after an average of 8.9 m. Fig. 7 shows a similar trial where a nonconductor robot was terminated.

In the diamond case, the six robots established the diamond the first time at 4.0 m on average—an incomplete "six-diamond." At 6.0 m, the two extreme robots were terminated, and with only four robots remaining, a complete four-diamond was

TABLE III
SIMULATION: ROBUSTNESS, AVERAGED OVER 10 RUNS FOR EACH

robot drop-outs (start: $N = 6$)	ft_1 (std.dev.)	ft_2 (std.dev.)
circle, 1 out	3.2 (0.077)	8.9 (0.130)
diamond, 2 out	4.0 (0.205)	8.3 (0.235)

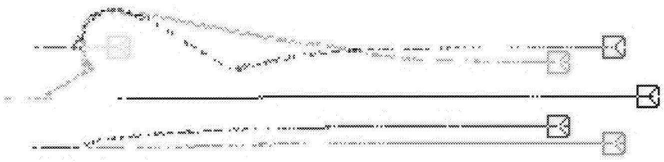


Fig. 7. A trial of 6 robots in a circle (hexagon); one is terminated, and the rest adapt to a 5-circle (pentagon).

formed through restructuring of the remains of the incomplete six-diamond; on average, this adapted formation was established by 8.3 m.

We performed real robot robustness experiments for all formations, both terminating the conductor and other robots. However, because robots tended to occlude each other while negotiating terminated colleagues, it was impossible to get tracking data. Our algorithm proved to be very robust. Due partly to the extremely high reliability of the (minimal) radio communication, and partly to the high recognizability of the color helmets, robots were always able to determine a new (correct) friend and locate it, or to assume the conductor duty. Hence, we believe it to be satisfactory to demonstrate robustness by only showing results from a few sample experiments in simulation and reality.

E. Switching Results

Ideally, many trials of switching between the possible formations should be conducted to fully validate the algorithm. This leads to an intractable number of trials, however, so instead we consider grouped formation switches: a switch between centered formations (Type A), and a switch between a noncentered and a centered formation (Type B). Since the only noncentered formation considered is the column, there is no need to consider switches between noncentered formations.

Type A switches are simple, as all robots keep the same friends, and the conductor stays the same. All that changes is a robot's *camangle* to its friend. As this happens gradually, so does the change in position, and the robot has no difficulty in keeping its friend in the image. Thus, there is no essential difference between switching from, say, the diamond to the line, and switching from the diamond to the wedge, or vice versa. Basically, it is a question of "folding" or "unfolding" the chain of friendships, where the degree of folding depends on the formations involved in the switch.

In our simulation experiments, therefore, we have demonstrated two examples of Type A switches: diamond→wedge and line→diamond, both with four robots (in Fig. 8, we show a diamond→wedge switch with eight robots). We performed 10 (9) trials; results are shown in Table IV. In all trials, the robots established the diamond for the first time at ft_1. Then, after 6.0 m, they were ordered to make the switch, and before 19.0 m, the wedge was established (ft_2). The results for the line→diamond switch are averaged over nine runs only. In one of the 10 runs we performed, the robots failed to form the diamond within the 19.0 m mark, and thus, for this run the ft_2 is void.

Type B switches are more complex, as previously explained. When switching from diamond to column, for example, the new conductor finds itself *behind* the other three robots when the order to switch is received. Therefore, any switch between a centered and a noncentered formation goes through the line as an intermediate step; any folded shape is forced to unfold first so that the new conductor is sure not to have other robots in its path. Table IV shows trials of the switch line→column.

With real-robot experiments, we again encountered the problem of occlusions, and it was not possible to reliably track the robots during a switch. However, again we verified that all Type A switches were performed reliably. Fig. 9 shows overhead camera images of a switch from diamond to line. For Type B switches, we verified that line→column was performed reliably; essentially, the same thing happens as shown in Fig. 6(a).

F. Obstacle Avoidance Results

We tested our algorithm in simulation with various scenarios. All four formations were tested with 10 trials each, in a scenario where a wall (wall$_1$), oriented orthogonally relative to the formation heading, was in the path of the formation. In addition, two more scenarios were tested with 10 trials each: a diamond formation crossing an obstacle field with seven robot-sized obstacles, and a line crossing a long, flat wall (wall$_2$), parallel to the formation heading. Results are shown in Table V. The evaluation goal was for the robots to establish the formation before 6.0 m, and then reestablish the formation within 19.0 m.

In all 10 trials of the diamond in the wall$_1$ scenario, the percentage of time in formation after ft was over 60. In other words, after first establishing the diamond before the obstacle, the robots managed to maintain it while negotiating the wall more than 60% of the time. In four trials, this number was 90% or more. For both sets of line trials, results are for nine runs only; in one trial in each scenario, the robots did not reform the line in time.

With real robot experiments, space was a problem. By the proportionality of $aheadbuffer$ to the number of robots, four robots simply demanded too long a clear line of sight to fit into our lab space. Reducing the $aheadbuffer$ means a deterioration in performance: robots can get closer to obstacles before they initiate their avoidance behavior. With a long $aheadbuffer$, a rigid-body avoidance of the whole formation is possible; with a small $aheadbuffer$, robots tend to get tangled up. In that case, the obstacle avoiding behavior collapses into individual robots avoiding obstacles and each other, which works very reliably, but "formation obstacle avoidance" loses its meaning. Therefore, we instead conducted experiments using only two robots. That meant a comparatively smaller *aheadbuffer*, and hence, it was possible to demonstrate the desired rigid-body avoidance in two scenarios. The robots performed an incomplete "two-diamond," meaning simply that robot 0 kept behind and to the left of robot 1 at an angle of 45°. In what follows, we refer to the conductor as C, and to its follower as R.

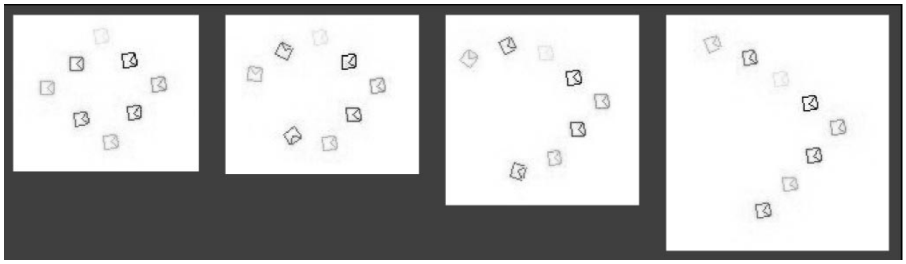


Fig. 8. Example: eight robots switching from a diamond to a wedge.

TABLE IV
SIMULATION: SWITCHING, AVERAGED OVER 10 (9) RUNS FOR EACH

$N = 4$	ft_1 (std.dev.)	ft_2 (std.dev.)
(A) diamond→wedge	3.6 (0.144)	8.3 (0.089)
(A) line→diamond	1.1 (1.373)	9.7 (0.193)
(B) line→column	4.0 (1.747)	9.2 (0.372)

In the O_1 experiment of Table VI, the robots had to negotiate a wall that was only in R's path. By the combined workings of the long $aheadbuffer$ and the *swerve messages* sent out by R, C made swerves of solidarity in time to let R keep its position. Hence, the robots maintained the formation while still avoiding the wall, as seen by the high percentage values of experiment O_1 in Table VI.

In O_2, the wall was in both of the robots' paths, but this time the wall had a passage in the middle, not wide enough to allow the robots to pass through it while keeping in formation. As seen by the three high % values in trials T1–3 in Table VI, R lost its position for only a short time, while going through the wall, before regaining it on the other side. In the T5 trial, R did not follow C through the passage in the wall, but instead went around it on the left, completely losing sight of C before getting back in position. The T4 trial (with a % of only 21.0) also demonstrated a recovery; i.e., in both T4 and T5 the robots actually got back into formation.

VI. DISCUSSION AND CONCLUSIONS

Our robots use only local sensing and minimal communication to maintain the global goal: formation f with N robots. Our key concept is that each robot follows a designated "friend" robot at the appropriate angle and distance, using a panning sensor that can provide the angle and distance information of the friend. By panning the sensor appropriately, the algorithm simply keeps the friend centered in the sensor's view. This also enables easy switching between formations. Unique IDs and a protocol for minimalist radio communication provide robustness to dropouts and help negotiate obstacles. A conductor that leads the way solves the problem of determining the friend's heading; by the nature of the algorithm, the only stable configuration is when all robots eventually have the same heading as the conductor (in [12], a clever way of displaying a robot's heading to its peers is devised). Any robot can assume the conductor role. While smooth turns are not a problem, this self-organization means that our algorithm cannot deal with sharp turns. One could imagine a future improvement as follows: before a 90° turn, the group forms a circle. Next, all robots individually make a 90° turn on the spot and circulate existing IDs among themselves so that the conductor is again in front with respect to the new heading. Finally, the original formation is reformed.

We implemented the algorithm by equipping each robot with a laser and a panning camera to measure distance and angle to its friend; we then validated the implementation through extensive simulation trials and 40+ experiments with physical robots.[4] Each robot broadcasts a heartbeat message every second, the conductor broadcasts the formation number every other second, and a robot avoiding an obstacle sends out a swerve message to warn others if it turns their way. All messages are one or two bytes. Ideally, only the nearby robots should make a swerve of solidarity, but calculating who is "near by" from group size, own ID, ID of robot avoiding an obstacle, and current formation, has not yet been implemented. Our general, quantitative evaluation criteria proved useful in analyzing positional data; a formal measure like the one we propose lends itself to an easier way of comparing different trials or even different strategies.

Our method proved highly successful for certain formations (column, diamond, and, to some extent, wedge), while the line proved more challenging. The reason is that keeping an angle of $\pm 90°$ to a friend proved to be the most difficult, while an angle of 0° was the most easy. This suggests a way of measuring the difficulty of formations, namely the sum of all friendship angles.

For four robots, the order of difficulty is thus: line (270), wedge (180), diamond (135), column (0)—given the $\pm 90°$ field of view of the friend sensor, 270 is the maximum difficulty with four robots. This matches our empirical results (see Tables I and II). The problem with keeping an angle of 90° to a friend is two-fold. The friend may fall off the edge of the laser's range, thus forcing the distance calculation to rely on the size of a color blob. Since this is obviously a very unreliable measure, varying with angle and from robot to robot, the resulting distance measurement will rarely be correct. The other problem is that it is very hard for a robot with a $\pm 90°$ camera pan to detect whether it is ahead of or behind its friend if their headings are slightly different. In the first case it should slow down, in the other it should speed up.

By a "better safe than sorry" principle, a robot that is close in its relative position to its friend only makes very small rotational corrections. Therefore, small oscillations vanish, in the sense that they are not passed on to follower robots. If the $\pm 90°$ angle problem could be effectively reduced, we would feel confident in claiming that our algorithm would scale to large numbers

[4]Formation videos and additional images can be found at http://robotics.usc.edu/~agents/projects/formations.html.

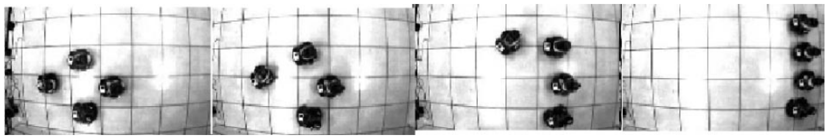


Fig. 9. Sequence of overhead images of a switch from diamond to line, order is left to right.

TABLE V
SIMULATION: OBSTACLE AVOIDANCE, AV. OVER 10 (9) RUNS

$N = 4$	ft_1 (std.dev.)	ft_2 (std.dev.)
line, wall$_1$	1.2 (0.900)	10.5 (1.081)
column, wall$_1$	3.8 (0.018)	13.1 (1.464)
diamond, wall$_1$	2.8 (0.070)	10.7 (1.274)
wedge, wall$_1$	2.1 (0.737)	11.8 (1.614)
diamond,obstacles	2.7 (0.143)	11.5 (0.925)
line, wall$_2$	0.7 (0.936)	14.7 (0.361)

TABLE VI
REAL ROBOTS: OBSTACLE AVOIDANCE. NUMBERS ARE % OF TIME IN FORMATION AFTER ft. FIVE TRIALS AND WEIGHTED AVERAGE

N=2	T1	T2	T3	T4	T5	Av.
O$_1$	100.0	97.9	99.8	97.8	93.9	98.0
O$_2$	81.8	82.0	89.4	21.0	26.7	61.6

of robots. We have already done successful simulation trials of circle and diamond formations with more than 10 robots.

In conclusion, we found that our simple scheme of minimal communication (heartbeat, swerve, and formation messages) was quite effective and robust. Our work shows that no global positioning system is needed for this task. A global map and global knowledge of all robots' positions would enable reliable performance, but we demonstrate that without such overhead and with only local control and minimal communication, a group of robots can still display global, coordinated behavior in the form of stable, robust, switchable formations.

ACKNOWLEDGMENT

The research reported here was conducted at the Interaction Lab, part of the USC Robotics Research Lab (http://robotics.usc.edu). The authors are grateful to R. Vaughan, A. Howard, B. Gerkey, E. Østergaard, B. Jung, and G. Sukhatme for valuable help, and to the three anonymous reviewers for constructive criticism.

REFERENCES

[1] R. Alur, A. Das, J. Esposito, R. Fierro, G. Grudic, Y. Hur, V. Kumar, I. Lee, J. Ostrowski, G. Pappas, B. Southall, J. Spletzer, and C. J. Taylor, "A framework and architecture for multirobot coordination," in *Proc. Int. Symp. Experimental Robotics (ISER 2000)*, Maui, HI, Dec. 10–13, 2000, pp. 289–299.

[2] H. Ando, I. Suzuki, and M. Yamashita, "Formation and agreement problems for synchronous mobile robots with limited visibility," in *Proc. IEEE Int. Symp. Intelligent Control*, Monterey, CA, Aug. 1995, pp. 453–460.

[3] T. Balch and R. C. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 926–939, Dec. 1998.

[4] T. Balch and M. Hybinette, "Social potentials for scalable multirobot formations," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA-00)*, vol. 1, San Francisco, CA, Apr. 2000, pp. 73–80.

[5] Q. Chen and J. Y. S. Luh, "Coordination and control of a group of small mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA-1994)*, San Diego, CA, 1994, pp. 2315–2320.

[6] J. P. Desai, V. Kumar, and J. P. Ostrowski, "Control of changes in formation for a team of mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA-99)*, Detroit, MI, May 1999, pp. 1556–1561.

[7] J. P. Desai, J. P. Ostrowski, and V. Kumar, "A theoretical framework for modeling and controlling formations of mobile robots," IEEE Trans. Robot. Automat., submitted for publication.

[8] B. P. Gerkey, R. T. Vaughan, K. Støy, A. Howard, G. S. Sukhatme, and M. J. Matarić, "Most valuable player: A robot device server for distributed control," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2001)*, Wailea, HI, Oct. 29–Nov. 3, 2001, pp. 1226–1231.

[9] C. R. Kube and H. Zhang, "Collective robotic intelligence," in *Proc. 2nd Int. Conf. Simulation of Adaptive Behavior*, Dec. 7–11, 1992, pp. 460–468.

[10] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Auton. Robots 4*, pp. 387–403, 1997.

[11] M. J. Matarić, "Designing and understanding adaptive group behavior," in *Adaptive Behavior*. Cambridge, MA: MIT, Dec. 1995, vol. 4, pp. 1:50–1:81.

[12] I. M. Rekleitis, G. Dudek, and E. E. Milios, "On multiagent exploration," in *Proc. Vision Interface*, Vancouver, BC, Canada, 1998, pp. 455–461.

[13] W. M. Spears and D. F. Gordon, "Using artificial physics to control agents," in *Proc. IEEE Int. Conf. Information Intelligence and Systems*, Rockville, MD, Oct. 31–Nov. 3, 1999.

[14] K. Sugihara and I. Suzuki, "Distributed algorithms for formation of geometric patterns with many mobile robots," *J. Robot. Syst.*, vol. 13, no. 3, pp. 127–139, 1996.

[15] R. T. Vaughan, "Stage: A multiple robot simulator," Institute for Robotics and Intelligent Systems, Univ. Southern CA, Tech. Rep. IRIS-00-393, 2000.

[16] B. B. Werger, "Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams," *Artif. Intell.*, vol. 110, pp. 293–320, 1999.

[17] B. B. Werger and M. J. Matarić, "Robotic 'food' chains: Externalization of state and program for minimal-agent foraging," in *From Animals to Animats 4: Proc. 4th Int. Conf. Simulation of Adaptive Behavior*, Maes, Matarić, Meyer, Pollack, and Wilson, Eds., Cape Cod, MA, Sept. 1996, pp. 625–634.

Jakob Fredslund received the Ph.D. degree in 2002, and the M.S. degree in 1998, both in computer science, from University of Aarhus, Aarhus, Denmark.

He is an Assistant Professor in the Bioinformatics Research Centre at the University of Aarhus. Besides multiple robot coordination, he has worked with the humanoid LEGO robot Feelix and with other LEGO robots as part of the University of Aarhus LEGO-Lab.

Maja J. Matarić received the Ph.D. degree in computer science and artificial intelligence from Massachusetts Institute of Technology (MIT), Cambridge, in 1994, the M.S. degree in computer science from MIT in 1990, and the B.S. degree in computer science from the University of Kansas, Lawrence, in 1987.

She is an Associate Professor in the Computer Science Department and the Neuroscience Program at the University of Southern California (USC), Los Angeles, the Director of the USC Robotics Research Lab and the Interaction Lab, and an Associate Director of Institute for Robotics and Intelligent Systems (IRIS). She has published over 30 journal articles, 13 book chapters, 66 conference papers, and 20 workshop papers, and has two books in the works with MIT Press. She has worked with NASA's Jet Propulsion Lab, the Free University of Brussels AI Lab, LEGO Cambridge Research Labs, GTE Research Labs, the Swedish Institute of Computer Science, and ATR Human Information Processing Laboratories. Her research is in the areas of control and learning in behavior-based multirobot systems and humanoids, and skill learning by imitation.

Dr. Matarić is a recipient of the NSF Career Award, the IEEE Robotics and Automation Society Early Career Award, the MIT TR100 Innovation Award, and the USC School of Engineering Junior Research Award. She is on the editorial board of three journals: the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, the *International Journal of Autonomous Agents and Multi-Agent Systems*, and *Adaptive Behavior*. She is on the AAAI Executive Council.