

Robust Multi-Robot Formations under Human Supervision and Control

Yehuda Elmaliach and Gal A. Kaminka

Abstract—There is considerable interest in real-world formation-maintenance tasks, where robots move together while maintaining a geometric shape. This interest is motivated by promise of robustly and efficiently moving multiple robots along a path, guided by a human operator. This paper presents a comprehensive set of techniques that fulfill this promise: (i) a novel method for fusing open- and closed- loop controllers, for robust formation-maintenance; (ii) an ecological display, allowing a human operator to monitor and guide robots, while improving their performance and reducing the failure rate; and (iii) a set of methods for interacting with the formation in the case of a disconnect in the formation. We evaluate each of these contributions in extensive experiments, including 25 human operators. We show significant improvements in performance (in terms of movement time), robustness (both in number of failures, as well as failure rate), and consistency between operators.

I. INTRODUCTION

THERE is considerable commercial interest in formation-maintenance tasks, where a team of robots move together while maintaining a geometric shape. This interest is motivated by promise of robustly and efficiently moving multiple robots along a path chosen for them by a human operator. Examples of such applications include cooperative object carrying [40], scouting [5], platooning and efficient convoys [7], groups of unmanned aerial vehicles [38], and spacecraft formation flying [1]. Realizing the potential of formation maintenance in real-world multi-robot applications requires addressing several open challenges.

First, many real-world applications require a human operator to monitor the state of the robots [14]. Previous approaches to monitoring multiple robots use individual robot displays that are independent of each other. For instance, the operator may monitor all robots in parallel, via a split display showing each robot's individual state; or the operator may switch between such displays [2], [14], [41]. However, independent individual displays lead to difficulties in monitoring *coordinated tasks*, requiring tight, continuous coordination between the robots. The problem is that the operator needs to infer the state of the formation based on individual displays, and (as we show) this decreases significantly from the operator's abilities, reducing the team's performance and robustness.

Second, formation maintenance requires a robot to know of the location of at least one other robot—called a *target*¹. There are many methods that rely on *closed-loop control* to maintain the formation by tracking the target(s); these rely on the robots' sensors (typically, vision and range sensors) to

supply continuous, uninterrupted, feedback about the target's location [5], [6], [9], [10], [13], [15]. However, this feedback is easily interrupted in the presence of obstacles, and when traversing rough terrains. While it can be possible to then re-link the formation [20], persistent interruptions slow the group's progress, and add to the load on the operator.

Finally, if and when an interruption occurs, and the formation becomes disconnected, the human operator must be able to effectively interact with the robots when needed (for example, when the formation becomes disconnected). Robots that require the operator's assistance initiate (or are issued) *call-requests*, which are queued for the operator. Traditionally, the operator switches control between robots, and uses single-robot teleoperation with individual robots to resolve the call requests in some (prioritized) sequence (e.g., [2], [12], [14], [28], [34], [37]). This method works well in settings where the task of each robot is independent of its peers, and thus the resolution of a call request is independent of others. However, in multi-robot formations, the robots are tightly-coordinated, and individual call-request handling means that all robots remain idle, while the operator tends to any single operator. This, despite the additional information that the otherwise idle robots may be able to offer the operator, based on their knowledge of the formation (which allows them to estimate the position of any robot).

This paper presents a comprehensive set of techniques for addressing the challenges above, including: (i) an ecological display, allowing a human operator to monitor and guide robots, while improving their performance and reducing the failure rate; (ii) a set of methods for interacting with the formation in the case of a disconnect, where a robot can no longer proceed; and (iii) a novel method for fusing open- and closed- loop controllers, for robust formation-maintenance that minimizes disconnects.

The paper is organized in three parts:

- First, we discuss tools for single-operator monitoring and guidance of multi-robot formations (Section III). This part explores a novel ecological display allowing a single operator to control multiple robots in formation. We report on extensive experiments with 25 human subjects. The results show that the use of the ecological display (i) reduces the number of failures and task completion time in these tasks; (ii) reduces the number of failures per second; and (iii) reduces the variance in controlling robots, thus leading to more consistent performance across operators.
- The next part (Section IV) explore ways to *multiplex* and/or *fuse* closed-loop and open-loop formation control, to limit the number of disconnects that occur in forma-

The MAVERICK Group, Computer Science Department and the Multidisciplinary Brain Research Center, Bar Ilan University, Israel.
E-mail: {elmaley,galk}@cs.biu.ac.il.

¹Typically, additional conditions must hold as well.

tions. The idea is to utilize communications to overcome sensor failures, and thus limit the number of interruptions that require human assistance. Experiments show that the techniques are able to significantly reduce the number of undetected obstacles, and thus increase the robustness of the formation to obstacles, even with limited sensing.

- In the final part of the paper (Section V) we report on first steps towards allowing coordinating robots, in a formation task, to use their knowledge of the coordination to autonomously assist the operator in resolving call requests involving a disconnected robot. Experiments with up to 25 human operators show that this call-request resolution method leads to shorter failure-recovery times.

Before presenting these three parts, we discuss relevant relevant literature and motivation for our work (Section II). Section VI concludes.

II. BACKGROUND AND MOTIVATION

This paper touches on a vast body of literature, as multi-robot formations is a well known canonical task, that has been investigated for many years. We therefore focus our review on the most closely related investigations. We first briefly discuss multi-robot formations, specifically in the context of obstacles that interfere with the formation (Section II-A). We then discuss investigations of human operation and monitoring of multiple robots (Section II-B).

A. Robust Formations

Balch and Arkin [5] examine three fundamental techniques for formation maintenance, in experiments with up to four (4) homogeneous robots:

- 1) *Unit-center-referenced* is a technique where the robots place themselves according to X, Y coordinates, relative to their peers in the formation, and subject to the geometric shape to be maintained. This technique relies on the ability of robots to sense the locations of all others.
- 2) To address this requirement, the *leader-referenced* technique instead allows robots to position themselves relative to the position of only a single robot, which acts as a leader. However, all robots must orient themselves with respect to the same leader.
- 3) Finally, the *neighbor-reference* technique relaxes this requirement further. Here, each robot positions and orients itself with respect to a single robot—called the *target*—but different robots can choose different targets.

It was shown that the last two categories in Balch and Arkin's work (*Leader-Referenced* and *Neighbor-Referenced*) are both related to a general method for formation maintenance, called *Separation-Bearing Control* (SBC) [9], [10], [13], [15]. In SBC, a single robot is chosen as the leader of the formation. Each robot (but the leader) must maintain connectivity—a given distance (separation) and angle (bearing)—with respect to an assigned target. There must be a path of such connected robots from every robot in the team to the leader. It was shown that SBC controllers are sufficient to maintain stable formations.

SBC is arguably the most practical formation-maintenance technique today for real-world settings. This is likely due to its simple requirements of sensing (monitoring) only one other robot, and to the wealth of opportunities it presents for optimizing sensor usage [20], [27] and robot role assignment [24], [26].

There have been several works addressing the robustness of SBC-based formations. Fredslund and Mataric [15] describe an algorithm for generating SBC monitoring rules for robots in a given formation. The robots are assumed to have supporting sensing capabilities, and the position of the leader is given. The monitoring rules are supplemented by communications for robustness against robot death.

Kaminka et al. [20] describe an algorithm that generates SBC monitoring rules based on the sensor configuration of the robots, and dynamically adjusts these rules to overcome sensor failures. They show that this leads to significantly improved robustness, as long as alternatives exist to prevent a robot from becoming completely disconnected. Their approach is susceptible to latencies of the communication protocol used to switch between different monitoring rules.

Our approach relies on fusing SBC control with open-loop, communication-based control of the formation, which relies on the localization of the robots and their ability to accurately estimate their own movements. In this, we complement the techniques outlined above, rather than compete with them.

Most previous work on formation maintenance in the presence of obstacles has assumed that obstacles are detectable in some unspecified fashion. Using the techniques presented below, robots can use their sensors to detect obstacles, to a greater extent than they do when they have to utilize their sensors to maintain the formation. In this, we facilitate the use of techniques which rely the use of obstacle-detection sensors, and that are difficult to use in sensor-impooverished robots that utilize their robots for formation maintenance.

For example, Chen and Li [8] propose a technique where obstacles are recognized by the leader robot, which builds a path for the formation to avoid the obstacles. Thus the leader is responsible for detecting any obstacles. Our approach complements this technique, by allowing other robots to also detect obstacles.

Similarly, Ogren and Leonard [30] describe an approach for allowing a group of robots moving in formation to avoid known obstacles. They show how to calculate a path for each robots that best maintains the formation while avoiding obstacles. Our work is complementary: The multiplexing technique we present is focused on detection of unknown obstacles; but we do not provide a method for calculating obstacle-avoiding paths.

Balch and Hybinette [6] use social potential fields which use attraction and repulsion to position robots within their relative positions in a defined formation. This technique is robust to obstacles in the path of the robots, in the sense that the geometric shape maintained by the formation is dynamically stretched to account for obstacles. However, the techniques assumes that robots know of the positions of obstacles. The technique we present in this paper frees up the robots' sensors for this purpose.

B. Operator Interaction and Monitoring of Formations

The literature on human-robot interaction is vast, but most investigations address a single operator controlling a single robot. Below, we focus on closely related works, which utilize an ecological approach to monitoring the state of robots, as well as methods for interacting (e.g., commanding) the robots. We refer the reader to a comprehensive survey by Goodrich and Schultz [17] for additional related investigations.

Traditionally, human-robot interaction addressed interaction with a single robot. Nevertheless, important lessons can be learned from single-robot displays.

Ricks et al. [31] and Nielsen et al. [29] discuss the ecological approach to displaying information in single-robot navigation tasks. This approach focuses on explicit display of the key constraints of the task [39]. We argue that the *relation tool* we present is indeed such a display for tightly-coordinated tasks, in that it explicitly displays the state of coordination of the robots.

Other ecological displays in single-robot interaction support this approach. Johanson et al. [18] propose a discrete geodesic dome called a Sensory EgoSphere (SES). The SES is a "two dimensional data structure centered on the robot's coordinate frame" that provides the operator with a pointer to an object on a map and the robot's sensor state. The *relation tool* similarly centers the display on the lead robot in the formation, and shows all other robots' in relation to it.

Yanko et al. at [41] describe techniques for making human operators aware of pertinent information regarding the robot and its environment. They tested this technique in a rescue-robot competition. Based on their study, they recommend providing user interfaces that support multiple robots in a single display, minimizing the use of multiple windows. The ecological coordination-monitoring display fills these requirements by giving the operator a single view of the controlled robots and their coordination state.

Possibly as a result, the bulk of existing work on controlling multiple robots puts the operator in a centralized role in attending to robots, and does not often distinguish between different task types on the basis of the coordination involved.

Indeed, many existing approaches implicitly assume that the sub-tasks assigned to different robots are independent of each other (e.g., exploring different sub-areas). In such settings, a centralized control scheme does not interfere with task execution, and the monitoring of each robot can be done individually, i.e., by reverting back to a single-operator/single-robot paradigm.

Adams et al. [2] investigated the use of a three-dimensional GUI that has selectable operation modes to switch control between robots, teleoperate a robot, create a navigation plan for the robot, or replay the last few minutes of the robot's task execution (for diagnosis of failures). Our work contrasts sharply with this approach, as we focus on a display that abstracts away the details of the robots' local surroundings, focusing instead on displaying their relative state, not their absolute state with respect to their environment.

Keskinpala et al. at [22], [23] developed a system for controlling a robot from a PDA (Personal Digital Assistant). PDA platforms are small, light-weight and mobile interaction

devices, but their size and limited resources pose significant challenges the user-interface design. The system Keskinpala et al. present includes three screens: vision-only, sensor-only and vision with sensory overlay. These methods stem from the requirement to provide the minimally necessary data to the operator needs, because of lack of space in the PDA's display. In our work we suffer from a similar problem, because we monitor multiple robots, and duplicating the displays for each quickly exhausts the available screen resources.

An exception to the single-operator/single-robot paradigm, Skubic et al. [34] reports on an investigation of a sketch-based interface for controlling one or more robots through a known map, on which sketches are drawn grouping robots together for group movement. Waypoints are then plotted and the robots navigate to the waypoints. This display does not keep track of coordinated movement. Instead, it shows where each *individual* robot is located, with respect to a global coordinate system, in contrast to our method, which displays information about the relative position of robots. However, similarly to our approach, all robots are commanded together.

Indeed, monitoring the status of robots and their tasks is only one component of the interaction of a human operator with a team of robots. An additional important component is the ability of the operator to issue commands to the robots, or interact with them when they require assistance. Fields [12] discusses unplanned interactions between a human and multiple robots in battlefield settings, where otherwise-autonomous robots send *call requests* to the human operator to ask for assistance. These call requests are queued, and the operator resolves the problems one by one.

Fong et al. [14] propose a *collaborative control* system that allows robots to individually initiate and engage in dialog with the human operator, one robot at a time. This approach requires significant autonomy by the robots, and assumes that their monitoring need not be continuous. The call requests are queued based on priority, and resolved serially.

Myers and Morely [28] discusses an architecture called TIGER that uses a coordinating agent that mediates between the operator and autonomous software agents. This agent centralizes the information from all agents, and can present it to the operator (or provide it to other agents). The agent is also responsible for translating operators instructions to the team. This approach thus assumes that call requests may be resolved autonomously by the robots, given appropriate high-level commands to the team. In contrast to this approach, we believe (as others do [12], [14]) that often, the operator must directly interact with a failing robot or its teammates to resolve a call request. We thus allow the operator to directly interact with any single robot, while others assist.

Rybski et al. [32] describes an architecture for allowing one operator to control multiple (miniature) robots. The main idea is to increase robot autonomy and allow the operator to interrupt the robot behavior with high-level commands. In addition, they supply an interface that supports mission design, and mission execution, where the operator can view mission status and teleoperate the robots. Our interface is fundamentally different from this interface in that we show a state view of all the robots, rather than only an individual

robot's.

ACTRESS (Actor-based Robots and Equipment Synthesis System) [4], [35], [42] is an architecture including an interface for monitoring and controlling multiple robots. The operator may issue commands that affect groups or individual robots; information is presented to the operator based on both explicit requests (from the operator to individual robots), as well as based on gathering of information exchanged by the robots. However, ACTRESS does not focus on visual presentation of the coordination, in contrast to our work. Moreover, ACTRESS does not utilize collaboration between the operator and robots in resolving call requests. The operator may issue commands to robots that assist in such resolution, but the robots are otherwise idle.

In contrast to the above centralized approaches, we believe that in tight-coordination settings, resolving call-requests is in the interest of all robots currently coordinating with the robot requiring assistance—and thus they should actively collaborate with the operator to resolve the call request. Other work has also examined distributed paradigms for human/robot interaction.

Tews et al. [37] describe a scalable client/server architecture that allows multiple robots and humans to queue call requests (*service requests*) for one another. Scerri et al. [33] describe an architecture facilitating teamwork of humans, agents and robots, by providing each member of the team with a proxy and have the proxies act together as a team. Our work differs from both of these investigations in that we do not attempt to put humans and robots on equal ground. Instead, the human initiates and controls the call-request resolution. However, once initiated, the task is carried out by all members of the robotic team and the operator.

Ali [3] compares different classes of human-robot team interaction (*Direct manual control, supervisor control, individual and group control*). The parameters measured are effectiveness (in term of task completion and speed of completion), safety (both for the robots and their environment), and ease of use. While we similarly evaluate different interaction methods, we focus only on the case of one operator and multiple robots. However, within those, we distinguish several different types. Moreover, we provide new distributed resolution types.

III. A SOCIALLY-ATTENTIVE ECOLOGICAL DISPLAY

We first introduce the ecological display approach for human operator monitoring of formations, which focuses on explicitly displaying the state of coordination the team (Section III-A). We then describe the display in detail (Section III-B). We empirically evaluated this approach in extensive systematic experiments with up to 25 human operators. The results show (Section III-C) that the use of the ecological display (i) reduces the number of failures and task completion time in these tasks; (ii) reduces the number of failures per second; and (iii) reduces the variance in controlling robots, thus leading to more consistent performance across operators. To our best knowledge, this is one of the largest studies done with human operators controlling multiple robots.

A. Monitoring the State of Coordination

A key component in real-world applications of multi-robot formation maintenance tasks is allowing the operator to monitor the progress of the team, and the status of the robots. Previous approaches to monitoring multiple robots use individual robot displays that are independent of each other, as discussed above. For instance, the operator may monitor all robots in parallel, via a split display showing each robot's individual state; or the operator may switch between such displays [2], [14], [41].

However, independent individual displays lead to difficulties in monitoring *coordinated tasks*, requiring tight, continuous coordination between the robots; i.e., where robots are highly inter-dependent. Here, the operator must monitor the state of coordination—the relative state of robots—in addition to the state of each robot. Such monitoring is called *socially-attentive* because it focuses on inter-agent relations [21].

Formation-maintenance is an example of such a task, requiring tight continuous coordination between robots. Such a task can be executed by a single operator, by guiding or tele-operating the lead robot, and allowing the others to maintain the formation autonomously. To maintain the formation, the operator must monitor the formation itself—slowing down or speeding up the lead as necessary—in addition to monitoring the movement of the team towards its goal. Such monitoring can be done, in principle, by showing the camera view of each robot. However, it might be much easier to do if the operator has a bird's eye view of the formation, showing the *relative* positions of robots. Unfortunately, a bird's eye view is not always possible, e.g., for lack of a global-view camera.

To address this challenge, we develop a socially-attentive *ecological display* component—called *relation tool*—that explicitly displays the state of coordination in a team, complementing individual display. Ecological interface design emphasizes visual cues that focus on the key constraints in the user's task [39]. For coordinated tasks, these include the coordination constraints in the team [21]. The relation tool allows the operator to visualize the robots' state with respect to each other, and thus visually identify coordination failures. Since the relative state of robots may not be known directly, the relation tool fuses sensor readings from multiple robots, and reconstructs from these the state of coordination between them. In doing this, it must overcome the uncertainty and noise inherent in robot sensor data.

The graphical *socially-attentive* display complements existing displays. It allows the operator to visualize the robots' coordination—their state with respect to each other—and thus visually identify coordination failures before they become catastrophic. By showing the operator an explicit visualization of the coordination state of the team, her cognitive load would be reduced, and her performance would increase.

B. The Relation Tool

Ecological interface design emphasizes explicit visualization of key constraints in the user's task [39]. Socially-attentive monitoring emphasizes that in coordinated tasks, these include the relative state of robots [21]. To show these constraints,

we developed the *relation tool*, a 2D display that shows the relative state of robots by drawing a geometric shape corresponding to their state. Colored dots denote different robots. The positions of the dots denote their states, and thus the shape they make up—their relative positioning—denotes their relative states. In principle, every application requires its own method of projecting robot state onto a 2D plane, and a target shape that defines normative coordination.

The key is that the operator should be able to see, at a glance, whether the shape being maintained corresponds to correct coordinated execution. When the shape deviates from ideal, the operator can easily identify coordination faults within the monitored team, with little or no need for inferring this information from the other displays. This eases the cognitive load on the operator in coordinated tasks.

We investigate the use of the relation tool concretely in two popular formation maintenance tasks (triangle and line). We created human-controlled versions of these tasks, and implemented them using the Tekkotsu software [36] for Sony AIBO robots. Each robot has an on-board video camera and an infra-red distance sensor pointing at the direction of the camera. They transmit their video and sensor readings to the operator's station for monitoring. The operator uses the mouse as a joystick, moving the controlled robot in the direction and speed chosen.

We begin by examining the line formation task, which we refer to here as *cooperative pushing*, as it has two AIBO robots jointly push a light-weight bar across the floor (Figure 1). One robot is teleoperated, while the other pushes the bar while maintaining a straight line with the human-controlled robot. The bar is color-marked, such that a robot can identify its position with respect to the bar. If the mark moves too much to the side, this would indicate a drift, i.e., the robot is either lagging behind or is pushing too quickly ahead. Here, we follow traditional sensor-based formation-maintenance techniques; the robots do not communicate with each other. Section IV examines the use of communications to maintain formations.



Fig. 1. Cooperative pushing (line formation) by AIBO robots.

The coordination between the robots involves a single dimension—the robots are to maintain equal velocities. One

possible visualization of this relationship consists of a horizontal line that connects two dots, representing the robot. The horizontal position of the dots remains fixed, while the Y axis denotes the angle of the color mark within their view.

Figures 2 and 3 show the interfaces when executing this task. Figure 2 shows the split-camera view from the individual robots, as presented to the operator, in a successful case (Figure 2-a), and in a failure case, where the box drifts to one side (Figure 2-b). Figure 3 shows the respective relation tool displays in both cases: The successful push (Figure 3-a) and the failing push (Figure 3-b). As can be seen, it can be difficult to differentiate the split-view displays in cases of success and failure (Fig. 2). However, by showing the relative velocities of the two robots (Figure 3) the failing push is easily detected.

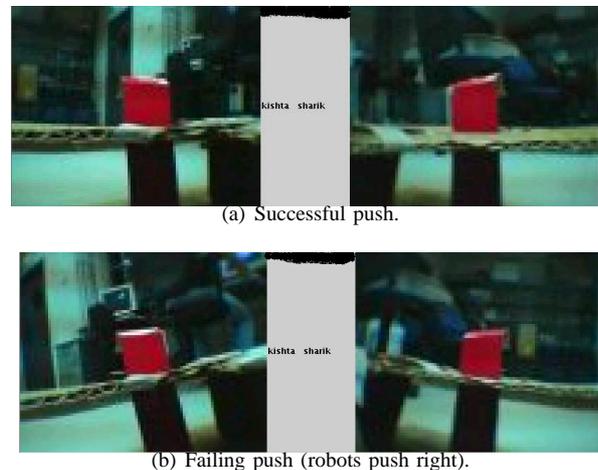


Fig. 2. Cooperative pushing (line formation): Split camera view.

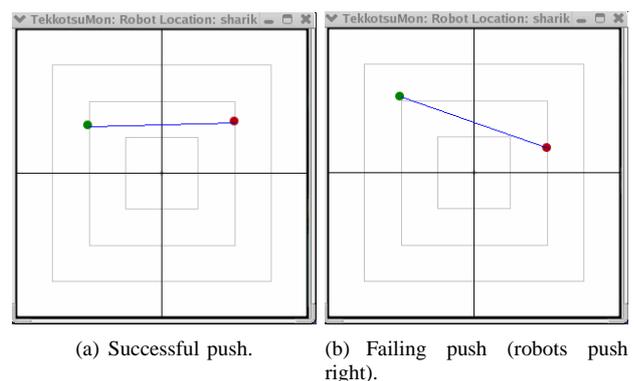


Fig. 3. Cooperative pushing (line formation): Relation tool display.

Of course, providing a visualization of the relative states of robots is trivially done when a global world-view camera exists, or perfect global localization data is available. However, this is not often the case in real-world applications.

Thus a key challenge in developing the relation tool lies in integrating the information needed for the visualization, from the robots themselves. The approach we take analyzes the robots' own sensor readings (including camera positioning,

infra-red range sensor readings, detected objects) to reconstruct the position of the robot with respect to others, from its own perspective. As a side-effect, we expose the relation tool display to the uncertainty and noise inherent in robot perception. This must be countered by noise-filtering processes within the display. In our case, a moving average filter was used on the distance and angle data to create a stable display.

The relation tool may be used to draw the attention of the operator to specific robots that are responsible for any mis-coordination. We use the formation task to demonstrate. Here, the objective is to navigate a triangular formation (three robots), through a short obstacle course. To allow a human operator to control the formation, the lead (front) robot teleoperated by the operator, while the two follower robots maintain fixed angles and distances to this robot using their sensors. Again, the robots do not utilize any communications for maintaining the formation.

Figures 5 and 4 show this task in action. Figure 4 shows an example of perfect formation, while Figure 5 shows a failed formation situation. In both figures, sub-figure (a) shows the actual position of the robots on the ground; (b) shows the split-camera view from each of the individual robots; and (c) shows a screen snapshot of the relation tool.

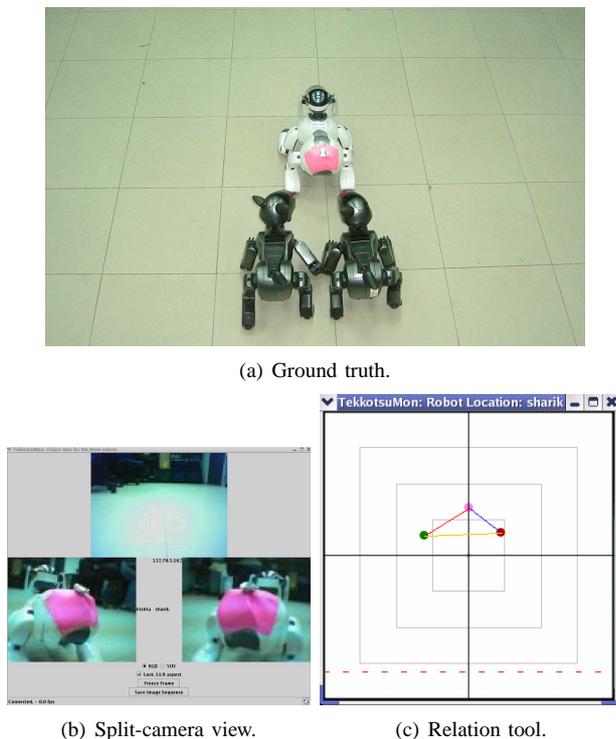


Fig. 4. Successful Triangle Formation.

The figures contrast the information presented to the operator with the relation tool and using existing approaches. Unlike the cooperative pushing task, the split-camera view (sub-figure (b) in Figures 5 and 4) does indeed provide indication of whether the formation is maintained. However, it is difficult to see from the split camera view to what degree the formation is maintained (i.e., the magnitude of the failure), and which

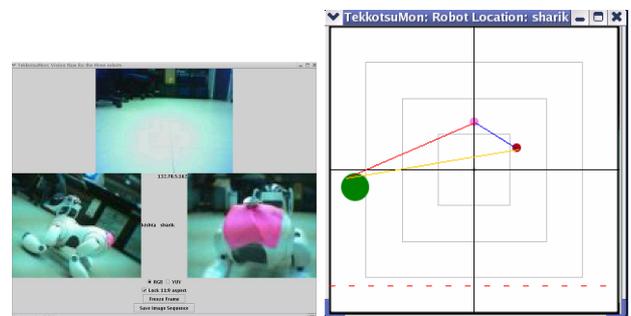
robots are responsible for it (i.e., the location of the failure).

In contrast, the relation tool makes it easy, at a glance, to see not only whether the formation is maintained, but also the magnitude and location of any failures. We chose polar coordinates to describe the formation. The X axis denotes the angle to the leader, while the Y axis denotes the distance to the leader. The position of the leader is always fixed. We connected the points (that represent the robots) with lines to create a shape easily recognizable by the operator.

The choice of the polar coordinates separates distance and angle for the operator. By glancing at the shape, one can fairly quickly determine whether the formation is breaking because a robot is lagging behind (distance too great), or its angle with respect to the leader is too sharp (e.g., because of a sharp turn).



(a) Ground truth.



(b) Split-camera view.

(c) Relation tool view.

Fig. 5. Failing Triangle Formation.

Indeed, to further assist the operator in localizing coordination problems, the display uses additional mechanisms to draw the attention of the operator where its most needed. One such fault-feedback mechanism uses the size of the dots, representing robot positions, to draw the operator's attention to failing robots. We use three sizes: regular, medium, and large. Regular size is used when the associated robot lies fulfills the constraints of the formation. Medium size is used when the robot begins to report intermittent failures in following the lead, as these are indicative of an impending formation failure. The large size is used when the formation is essentially broken, e.g., when the robot in question completely lost track of the lead robot, and is unable to proceed.

Another fault-feedback mechanism is the dashed line drawn across the bottom of the display. This line signifies the maximum distance sensed by the robots' sensors, and thus the

position in which they are likely to lose track of the leader. The operator may use this line to estimate how far it can let the robots stray away from the leader, while not getting into catastrophic failures.

C. Experimental Evaluation of the Relation Tool

We evaluate the effectiveness of the relation tool in the triangle formation maintenance and the cooperative pushing (line formation) tasks. Our goal is to explore the generality of the method. In the triangle formation, the operator leads the robots in a triangular formation towards a target destination, while avoiding obstacles. If the operator causes the lead robot to turn too sharply, or move too quickly, the formation may break, as the SBC controller in the follower robots will lose sight of the leader. However, the operator seeks to minimize the time it takes to reach the destination. In the line formation (box pushing), the operator controls the velocity of one of the robots, while the other is pushing autonomously. The operator must be careful not to push too quickly for the other robot, nor to lag behind.

We believe that the relation tool should be used to complement, rather than replace, existing display (which focus on individual robot state). We thus conducted experiments contrasting different combinations (see below) of the socially-attentive display with individual robot display. We ran multiple experiments with novice operators, age 20–30.

19 operators were tested in the pushing task (18 males/one female, 18 students—including the female—of which 15 are in computer science). 25 operators were tested in the formation task (23 male/two female, 22 students—including the two females—of which 19 are in computer science). The students in both groups were either graduate students or undergraduates in their final year. None had previous experience controlling multiple robots of any kind.

Each operator tried all combinations available in the task she operated (a within-subjects design). However, to avoid ordering effects, the order in which each operator tried each combination was randomized (in both sets of experiments). In no setting were the operators able to see the robots while operating them. In all cases, operators were given an approximate 25-minute training session in operating a single and multiple robots (including the formation and pushing tasks), until they reported they felt comfortable controlling the robots. Overall, the results below represent almost 100 hours of human operation.

1) Cooperative Pushing (Line Formation) Experiments:

The first experiment examined the use of the relation tool in the cooperative pushing task. We contrasted three interfaces: a split-camera view only (representing existing approaches), a combination split-camera and relation tool, and the relation-tool alone. We remind the reader that all 19 human operators were tested on all three interfaces, randomizing the order of their introduction to the different interfaces to prevent biasing the results due to human learning. Their performance was measured as the average absolute angle deviation from the imaginary horizontal line connecting the robots when they maintain ideal relative velocity. This angle was sampled at 20Hz during task execution.

Figure 6 shows the results of this experiment—the average absolute angle error—averaged across all operators. Clearly, both combinations that use the relation tool are significantly superior to the interface relying on camera alone. Moreover, the surprising result here is that the relation tool by itself is sufficient (in fact, even slightly better than its combination with the split view). This is due to this task being essentially a pure-coordination task: The operator does not need to worry about where the pushed object is going, as long as the relative velocity of the robots is 0 (i.e., their velocities are equal). Thus even a socially-attentive display by itself is sufficient. On the other hand, the non-social split-camera view (by itself), is difficult to use for coordination. A one-tailed t-test (assuming unequal variance) shows that the difference between using the tool by itself, and using the split-camera view, is statistically significant (we use a 0.05 significance level). The probability of the null hypothesis is $p < 0.014$ when looking at the difference in the number of failures.

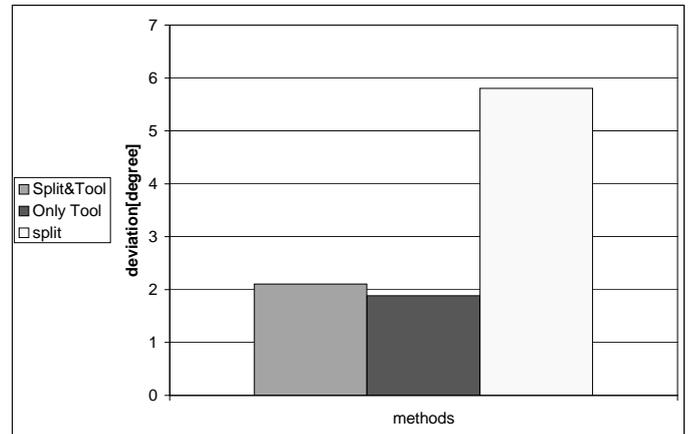


Fig. 6. Line formation: Total number of failures.

We also examine the angle error results with versus task completion time. Figure 7 shows these results as average absolute angle error versus time to complete a 180cm walk. The Y axis is the average angle error and the X axis is the time to complete the task. The fastest time for completing the 180cm walk was by the method that does not use our socially-attentive display (the *split view*). However, we can see that this method also had the maximum average angle error. We believe that this is the result of the operator, having no idea of the relative state of the two robots, just pushed the teleoperated robot as quickly as possible, finishing the course as quickly as possible (but poorly).

These results should only be interpreted qualitatively. The exact distance traveled by robots in each trial is technically difficult to measure precisely (as the robots' own odometry is inherently inaccurate). We thus allowed the operator to always proceed for the estimated distance (180cm), and measured the time it took. Because of the distance is estimated in this trial, the timing may not be accurate.

Finally, we examine the the standard deviation of the distribution of number of failures in these experiments. Lower standard deviation indicates more consistent performance of the different subjects, i.e., less differences between the results

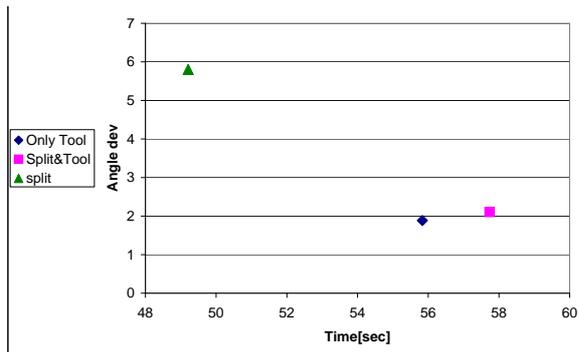


Fig. 7. Line formation: Failures vs. task completion time.

Interface	Std. Deviation
Split-view	7.11
Split+Tool	0.93
Tool alone	0.79

TABLE I

LINE FORMATION: STANDARD DEVIATION OF THE NUMBER OF FAILURES.

of different operators. This, in turn, typically indicates more intuitive and more generally-applicable interfaces. The standard deviation results appear in Table I. They indicate that interfaces using the relation tool unequivocally lead to more consistent performance among operators.

2) *Coordinated Movement Evaluation*: In the triangle formation task, we compare three interfaces. The first presented the operator with the split-view video streams from all robots (e.g., Figure 4-c). The second combined the this split-view with the socially-attentive display previously described. The final interface consisted of a single camera (the lead robot's) and the socially-attentive display. Each of the interfaces was tried with three different obstacle courses, varying in difficulty (a total of 9 different configurations). The *simple* course consisted of an open space with no obstacles at all (Figure 8-a). The *medium* course consisted of a single obstacle that had to be by-passed (Figure 8-b). In the *difficult* course, the operator was to lead the robots between the two obstacles (Figure 8-c). To verify the relative difficulty of the path, we sampled 7 of the experiments for the number of times a robot hit an obstacle: The simple course had no such hits (as there are no obstacles). The medium course had only a single hit in all experiments. The difficult course had 2–3 hits per method.

Again, all 25 operators tried all nine different settings, in randomized order (to prevent learning effects). For each of the trials, we recorded the number of non-catastrophic formation failures, and time to complete the task. Non-catastrophic formation failures were measured as the number of times a follower robot has temporarily lost track of the lead. These are indicative of the quality of the operator's control. Too many of them result in permanent tracking failures, which lead to total breakdown of the formation. When such failures occurred, the operator would have to teleoperate the straying robot until the formation was re-established.

Figures 9,10,11 show the results of these experiments in terms of the average number of non-catastrophic failures

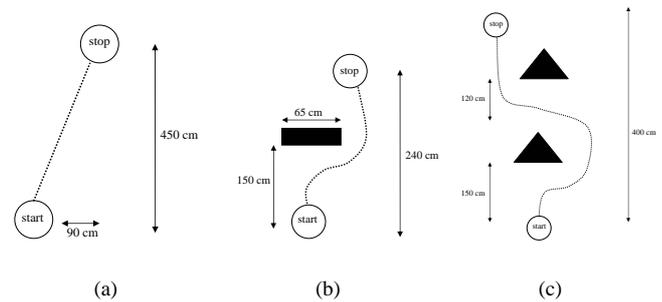


Fig. 8. Triangle formation obstacle courses.

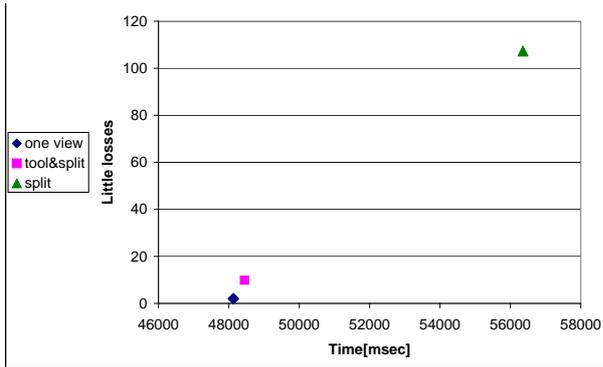
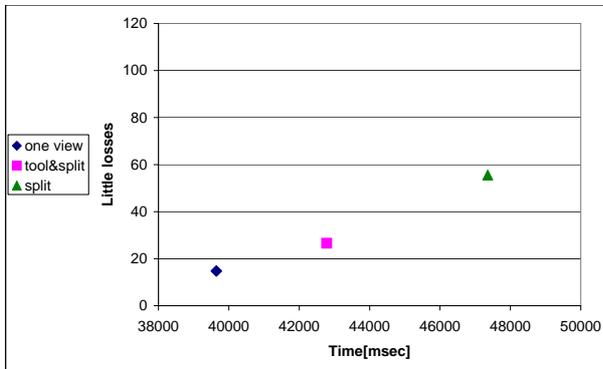
per operator, versus the average task completion time. The horizontal axis shows the time (in milliseconds). The range of the horizontal axis in these figures is fixed at 12 seconds, though the offset is different, as the more difficult courses took longer. The vertical axis shows the average number of non-catastrophic failures that took place during each trial.

The results show that in all course difficulty settings—simple, medium and difficult—the use of the relation tool is preferable to using only individual displays. This lends support to the hypothesis that socially-attentive ecological displays (explicitly displaying coordination state) can significantly improve monitoring of robots in coordinated tasks.

In particular, both course completion time and the number of failures during execution were generally reduced using the socially-attentive display. In the simple- and medium-difficulty courses, the best monitoring approach was single camera and the socially-attentive display. It was significantly better than the split camera interface, at a 0.05 significance level. In the easy course, a one-tailed t-test (assuming unequal variances—see below) shows a significant difference these method, both in the number of failures (the probability of the null hypothesis being $p < 0.011$), and in the time ($p < 0.015$). Similarly, in the medium course, there are significant differences between these two methods, both in the number of failures ($p < 0.04$) and in task completion time ($p < 0.02$).

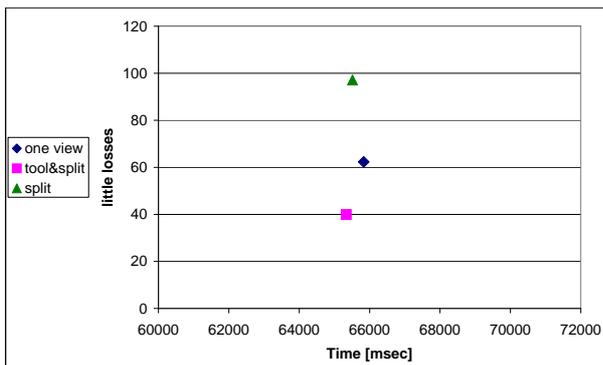
However, in the difficult course the best monitoring approach used both the split-view and the relation tool, in spite of the additional information displayed to the operators. The difference between this approach and the split view interface was not significant in time ($p = 0.48$), but was significantly different in the number of failures ($p < 0.014$). The difference in the number of failures between the split view interface and the interface using single camera and relation tool was only moderate ($p \approx 0.15$). We believe that this is due to the operator using the split-camera view to look at obstacles that have been bypassed by the lead (see [29], [31] for an ecological interface approach to this problem). Such obstacles were not much of a problem in the other, easier, courses. We leave further investigation of this to future work.

While the results show significant improvements in task completion time and number of failures, a question may be raised as to whether a socially-attentive ecological display qualitatively changes the way the operator interacts with the team. For instance, the experiment results above could also be

Fig. 9. Triangle formation failures in *simple* course.Fig. 10. Triangle formation failures in *medium* course.

indicative of the team going slower or faster, but maintaining the same number of failures per second—thus indicating that the drop in failures is due to the team moving faster, rather than to a qualitative change in operator control.

Additional results show that rather, the use of the relation tool leads to qualitative differences in the way the operator controls the robot team. Figure 12 shows the average number of failures per second, in the different courses. Clearly, the easy course is indeed easier than the medium-difficulty course, which is easier than the difficult course. However, what we see in the results is that the use of the socially-attentive display leads to a significant reduction not just in the time and total

Fig. 11. Triangle formation failures in *difficult* course.

number of failures (as evident from the previous figures), but also reduces the *failure rate*.

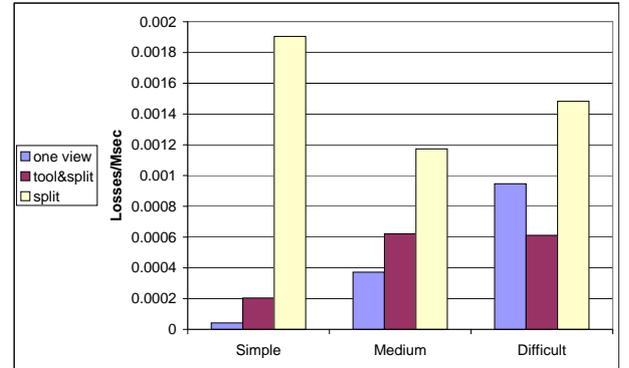


Fig. 12. Triangle formation failures per millisecond.

Additional evidence for this qualitative improvement in operator control is found when we examine the standard deviation values for the number of failures and task completion times of different operators. Table II displays the standard deviation of the number of failures, for the different courses. Table III displays the standard deviation values of the task completion time, for the different courses.

These results show that operators are more consistent in their performance when using the relation-tool, than when using the split view by itself. The standard deviation values for the methods using the relation tool are generally much smaller than for the split camera display. This indicates more consistent values, i.e., less variance between operators in terms of ability to control the robots. In the difficult path, the single camera view with the relation tool has a large standard deviation (though smaller than the one for the split camera view by itself, when looking at the number of failures), but the relation tool with the split camera view has smaller standard deviation.

Course	Split View	Split View and Relation Tool	Single View and Relation Tool
Easy	319.45	32.65	6.59
Medium	141.85	51.30	50.56
Difficult	144.97	66.93	138.65

TABLE II
TRIANGLE FORMATION: STANDARD DEVIATION IN NUMBER OF FAILURES.

Course	Split View	Split View and Relation Tool (<i>tool+split</i>)	Single View and Relation Tool (<i>one view</i>)
Easy	25.53	12.01	9.17
Medium	22.37	14.68	10.68
Difficult	15.88	13.58	19.22

TABLE III
TRIANGLE FORMATION: STANDARD DEVIATION IN TASK COMPLETION TIMES (MEASURED IN SECONDS).

D. Monitoring Formations: Summary

This section took a step towards allowing a single human operator to effectively monitor a team of robots that are tightly coordinated. The experiments we have conducted show that existing techniques do not adequately address this challenge. Their inability to explicitly display the coordination state of the team cognitively burdens the operator and reduces from her effectiveness at controlling the robots.

The socially-attentive *relation tool* display is an ecological interface display addressing this challenge. It has three principal advantages over previous work:

- First, it significantly reduces the amount of inference needed by the operator to infer the relative state of robots—and thus the state of coordination between them.
- Second, its dimensions can be used to directly provide the operator with information about failures, e.g., as in the formation case.
- Third, it can easily complement other types of displays useful for the task, such those that show the heading or distance left to the destination, power, etc.

The experiments on real robots show that the relation tool significantly reduces the total number of failures, and task completion time in two tight-coordination tasks. Furthermore, we have shown that the use of the relation tool leads to qualitative change in the capabilities of the operator: Not only do failures and completion time decrease, but the failure rate (failures per second) improves significantly as well. In addition, methods utilizing the relation tool lead to more consistent operator performance.

IV. MAINTAINING ROBUST FORMATIONS

The operator of a formation is inherently limited by the capabilities of the robots to sense their surroundings, and provide information about potential failures. A challenge arises when robots do not have sufficient sensors to both track their peers and their environment at the same time. This could be, for example, if the limited sensors are kept busy providing input to the closed-loop controller that is used to maintain the formation. This section addresses this challenge.

We first introduce differentiate sensor-based closed-loop formation maintenance from communication-based open-loop formation maintenance (Section IV-A). We then (Section IV-B) discuss the two key methods used to combine open- and closed-loop maintenance (*multiplexing* and *fusing* controllers). Finally, we report on experiments evaluating the different methods (Section IV-C).

A. Open-Loop and Closed-Loop Formation Maintenance

In the *sensor-based* formation-maintenance algorithm, each follower but the leader is to maintain a specific distance and angle to another robot (called the *anchor*). This is called Separation-Bearing formation-maintenance control, and is proven to be stable [13]. A problem arises when the robot's sensors are limited and the robot also needs to detect obstacles: If the follower does not scan for obstacles, it may fail to discover them. And if it scans for obstacles, it may lose sight of its anchor, and thus lose its place in the formation.

For instance, on the Sony AIBO ERS robots, the sensors used for formation maintenance are on a single pan-tilt component (the head). The robot cannot follow a leader (at some fixed angle) and simultaneously scan for obstacles.

One obvious alternative is to utilize open-loop control in formation maintenance, to free up robots' sensors for other uses such as obstacle avoidance. While operating in *communication-based* open-loop control, the leader of the formation broadcasts its movement vector (velocity and heading changes). Based on this communication, and their predefined ideal positions in the formation, all other robots calculate their own relative movements, without relying on sensors. However, this relies on odometry reading in both the leader and the follower. In principle, translating the movements of the leader into each follower's actions, via communications, is sufficient. In practice, accumulating odometry errors prohibit this technique from being used exclusively.

This is indeed an open-loop controller for the formation: Messages cannot in practice be sent continuously, and thus a projection is made as to the anticipated position of the leader (and by implication, the follower), using *affine transformations* [43]. Once the anticipated position is known, the follower can set it as a goal position, and use simple motion planning to generate a movement vector of its own. This movement vector is maintained until a new broadcast from the leader initiates this calculation once again.

The translation of target position to movement vector has two factors. The first is handled by the affine transformation. The second requires additional corrective actions by the follower robots. We describe these factors below.

The first factor is the effect of the leader's heading on the path chosen by its follower. Figure 13-b,c show cases where the position of the leader is identical, but its heading is different. As a result, the target location for the follower, and the path to it (both indicated by the arrow in the figures) are radically different. This also implies that the affine transformations are sensitive to errors in their inputs, as even small deviations in the heading may result in large difference in the computed movement vectors.

The second factor in correctly computing the movement vector is tied to the difference in the body orientations of the leader and follower robots, after the latter reach their target positions. Ideally, the orientation of the leader and followers should be equal at that point. However, depending on the path taken by the follower, the orientation of its body might be different from that of the leader (see Figure 14).

To maintain the orientation error in the followers as small as possible, we recommend explicitly tracking the difference in orientation between the leader and the follower, and correcting it in each time step. However, this approach can result in jerky movement on the part of the robots, when they attempt to correct a large error within a single time-step. To address this, the controller should limit itself to corrections that are only of a limited range, and instead apply them over multiple time-steps, if necessary.

The advantage of the communication-based controller is that it can free up some of the robot's sensors. Instead, the follower robot maintains the formation only by communica-



(a) Ideal positions of the robots.



(b) Leader changed heading.



(c) Leader did not change heading.

Fig. 13. A triangle formation of three Sony AIBO robots. Figure (a) shows the ideal poses of the robots. Figures (b) and (c) illustrate the sensitivity to heading; the leader is in the same x, y location in both figures, but its heading is different, implying a radically different target position for the right follower robot.



(a) Follower orientation maintained at end of path.



(b) Follower orientation not maintained at end of path.

Fig. 14. In (a) and (b) the x, y location of the follower is the same, as the target position. However, the path taken by the right follower to the target greatly affects the final orientation of its body with respect to that of the leader.

tion. The disadvantage of this technique is that it requires perfect odometry, a requirement that cannot be fulfilled in realistic settings. If the anticipated position of the leader and the follower are computed based on imperfect, noisy odometry, the errors quickly accumulate. Moreover, as we have seen, slight differences in values of the heading can imply radically different movement vectors.

B. Combining Controllers

To allow limited-sensor robots to maintain formation while still recognizing obstacles, we propose to combine the two controllers described above, in settings where the robots' sensors are limited, but communication between robots is possible. In such settings we propose to combine two formation controllers types: A closed-loop formation-maintenance algorithm using sensors, and an open-loop algorithm using internal navigation (odometry) and communications.

We compare between two combination approaches: *multiplexing* the controllers (using one at a time), and *fusing* them (using both in parallel). The idea in combining the

controllers is to offset their disadvantages, and gain from their complementary advantages.

The *multiplexing* technique works as follows. Each follower robot relies on the sensor-based algorithm until it arrives to its predefined position in the formation. We therefore explore ways to *multiplex* and/or *fuse* closed-loop and open-loop formation control. Multiplexing is done in time, giving the alternative methods different periods of time in which they control each robot. Switching between the different methods utilized the following principle: Each robot relies on visual tracking (closed-loop control) until it is within tolerance levels of its position in the formation. When this occurs, the robot switches to *communication-based* open-loop control, and uses its sensors to scan for obstacles, while communicating with the leader. To verify its position and inhibit accumulating errors, the robot switches back to visual tracking after a fixed period of time. We also explore combining controllers by fusion, by merging the output commands of each open-loop and closed-loop controllers on (within some tolerance radius, to allow for uncertainty in sensing). When this occurs, the robot switches to the communication-based formation-maintenance behavior. Now, the robot's sensors are free and the robot can search for obstacles. The follower robot moves in this mode for a fixed period of time (which we vary in the experiments, see Section IV-C). It then switches back to the sensor-based algorithm, and the cycle repeats.

In the *fusing* technique, the robots multiplex between the open- and closed- loop controllers (otherwise, they cannot hope to detect obstacles). However, during the time when both sensor-based and communication-based controllers are active, the output commands of the controllers are fused: The average of the two controllers is taken as the output.

There are competing goals in using the open-loop controller, with both combination techniques. On one hand, the more the robots rely on open-loop control, the more they can scan for obstacles, and provide improved performance. On the other hand, the longer they remain in open-loop control, the more errors in position are accumulated (in relative positions of the robots, with respect to their teammates), and thus the formation degrades.

Thus the timeout period, which limits the amount of time robots remain under communication-based open-loop control must be determined. We take an empirical approach to determining this value. We note that it might be possible to set theoretical bounds on this value, depending on expected obstacle density. We leave this direction of research to future work.

C. Combined-Control Experiments

To evaluate the contribution of these approaches, we compare the *multiplexing* and *fusing* methods with their closed-loop and open-loop components, by themselves. The experiments are carried out using physical (Sony AIBO) and simulated robots.

We carried out two separate repeated-trials experiments. The evaluation has two facets. First, in Section IV-C1, we evaluate the impact of the combination techniques on the ability of

sensor-limited AIBO robots to detect obstacles, the motivation for the techniques. Second, in Section IV-C2, we evaluate the hypothesized costs of combination, i.e., the hypothesized decrease in precision.

1) *Detecting Obstacles*: This section report on experiments carried out with physical Sony AIBO robots moving in formation. The goal of the experiment is to evaluate to what degree does controller combinations (e.g., multiplexing) allow robots to detect obstacles that may be otherwise be undetectable.

Here, three Sony AIBO ERS-7 robots were arranged in a triangular formation (Fig. 13-a). While operating in sensor-based separation-bearing control mode, the two followers in the rear monitor the leader using their head-mounted camera and infra-red range sensors. The robots utilize the color patch on the rear of the leader for identification, and maintain the distance and angle to it [20]. Otherwise (when using communications) they scan for obstacles and maintain the formation by communication.

The leader actively scans for obstacles. On detection, it finds a path around them that considers its own physical body, rather than the entire team (as proposed in [8]). Such a path cannot be considered safe for the followers, and indeed we intentionally place obstacles such that such a path would put them in the way of the followers. This is done so as to examine the followers' ability to detect obstacles.

We use three different obstacle courses for this experiment (see Fig. 15). In the *Right* obstacle course the robots walk in a straight line; the right follower robot needs to recognize the obstacle blocking its path. The *Left* obstacle course poses the same challenge to the left follower. Finally, in the *Diagonal* course the right follower needs to recognize the right obstacle and the left follower needs to recognize the left obstacle (the leader will try to pass between the obstacles).

In each of the obstacle courses, the formation was run five times, in both the visual sensing control mode, and the *multiplexing* mode, for a total of 30 runs (10 in each course). We did not experiment with the open-loop control in these experiments; as it essentially frees up all the robots sensors to focus only on the task of detecting obstacles, it serves as a theoretical upper limit. We therefore assumed that with pure communication-based control, all obstacles are detected. We note also that there are no separate results for the fusion method, because it is identical to the multiplexing method in terms of time available for detecting obstacles (since then only one controller is generating output). The distinction between them is explored in Section IV-C2.

Fig. 16 shows the result of the comparison between the multiplexing technique and the sensor-based formation maintenance. The *X* axis shows the obstacle course. The *Y* axis shows the fraction of the undetected obstacles over all trials, thus a lower value indicates improved performance. We can see that the multiplexing technique performs better than the sensor-based algorithm used earlier, though statistical testing shows that the difference is only moderately significant (one-tailed t-test, $p = 0.07$).

A one-tailed t-test significance test of the experiments with the robots (above) showed that multiplexing was only moderately significantly better ($p = 0.07$). We believe this is

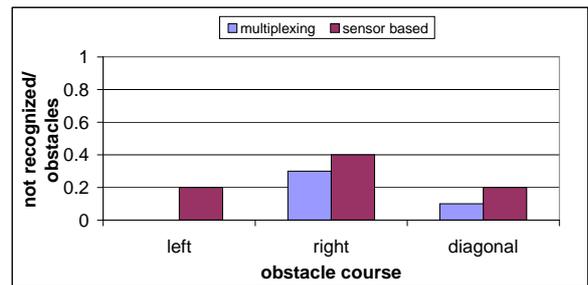


Fig. 16. Fraction of undetected obstacles over multiple runs of each technique, in different obstacle courses.

due to the relatively small number of experiments. We thus ran additional experiments with *simulated* AIBO robots, using the player/stage environment [16], where many more trials could be run. Figure 17 describes the obstacle course used in the simulated environment. Each of the techniques (multiplexing, sensor-based) was run 25 times.

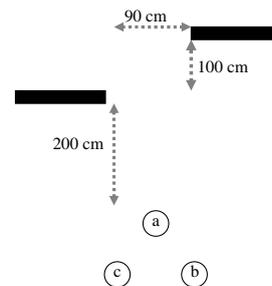


Fig. 17. Obstacle course in the simulation experiments. The leader robot moves in straight line, but its followers must detect the obstacles on its left and right.

Figure 18 shows the fraction of unrecognized obstacles (*Y* axis) in this experiment, for each of the techniques, over 25 runs. The *Y* axis shows the fraction. We again see that the multiplexing approach significantly decreases the fraction of undetected obstacles. The results are significant at a level of $p = 0.00000000164$ (one-tailed t-test).

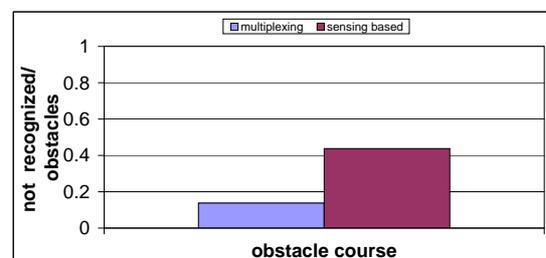


Fig. 18. Fraction of undetected obstacles over 25 runs for each technique.

Thus both in simulation and in experiments in the real world, we see that the multiplexing approach decreases the number of undetected obstacles, though it does not perform as the theoretical best (i.e., with perfect open-loop control and perfect knowledge of obstacles). This happens because the multiplexing technique, while giving more opportunity

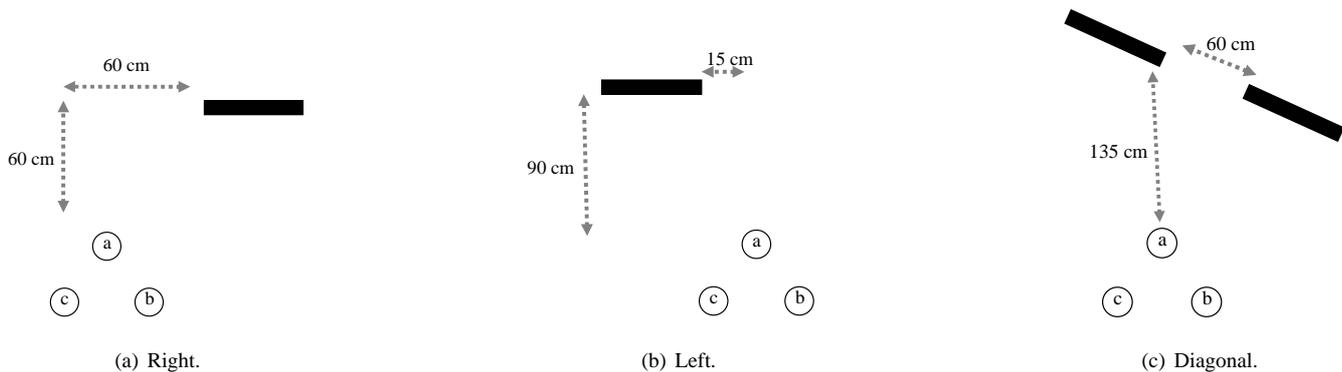


Fig. 15. Three obstacle courses used in experiments with the AIBO robots.

to the followers to detect obstacles, occasionally switches back to sensor-based closed-loop control, for correcting the accumulating odometry errors. In such cases, the follower robots cannot use its sensors to detect obstacles.

2) *Formation Precision*: An hypothesis underlying the combination approach is that the gains it offers (as the previous section demonstrates) will come at a price of decreased precision. The reliance on open-loop control, even if only for limited periods of time, should in principle cause some degradation in the ability of robots to position themselves in the formation. It might therefore be hypothesized that selecting fusion as the combination method may lead to improved results.

This section examines this hypothesis. We compare the quality of the formation maintenance with different formation techniques, under varying conditions of noise in movement. The quality of the formation maintenance is measured as the average absolute deviation of the follower robots from their ideal location in the formation. Our expectation is that combination would fare worse than its constituent techniques, especially with increased noise.

We compare the multiplexing and fusing combination techniques, presented earlier, to their two constituent techniques: Open-loop formation control (*communication-based maintenance*), and closed-loop formation control (*visual formation maintenance*). For the combination technique, we use a timeout of 8 seconds for the period in which the robot scans for obstacles, relying only on open-loop control. The timeout was determined empirically, but experimenting with different timeout values.

Precise positioning in formations is relatively easy when the formation moves in a straight line. It becomes more difficult to achieve in realistic settings, when formation (and robots) have to turn. We thus examine the precision resulting from each formation control technique, when the angle of the leader's turn is varied. In the following experiments, the leader robot moves in a straight line for 20 seconds and then turns in place and proceeds. We control the leader's turn angle (0, 15, 30, 90 degrees), and measure the resulting position errors in the followers once the turn is complete.

Given that we wanted to control the amount of uncertainty in the movements of the robots, we chose to run these

experiments in simulation. We used a Gaussian to model the noise in the movements of the robots, at several qualitative levels of 0%, 20% and 40%. The percentages signify the uncertainty in terms of standard deviation, i.e., a level of 20% Gaussian noise means that the standard deviation of target value X will be 20% of X .

Figures 19, 20 and 21 show the results of these experiments, for noise levels 0%, 20% and 40%, respectively. In these figures, the X axis shows the sharpness of the leader's turn in degrees. The Y axis represents the average absolute deviation (error) of the follower robots from their ideal position in the formation. The line marked *visual* shows the results of the closed-loop sensor-based visual maintenance. The line marked *communication* corresponds to the open-loop communication-based maintenance. The lines marked *multiplexing* and *fusing* correspond to the multiplexing and fusing (the combination approaches). Each one of the points is an average over 40 data points (20 runs, two follower robots).

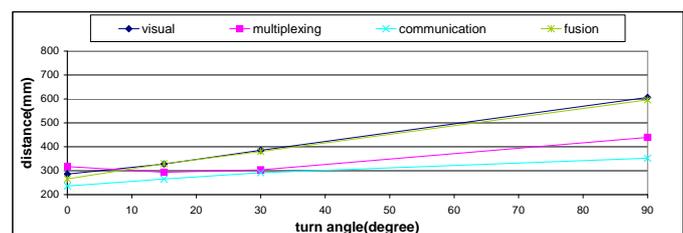


Fig. 19. Deviation from the ideal position in formation vs. the turn angle, with no uncertainty in movement/odometry.

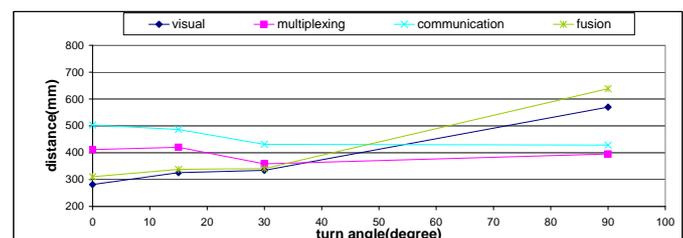


Fig. 20. Deviation from the ideal position in formation vs. the turn angle, with uncertainty levels set at 20%.

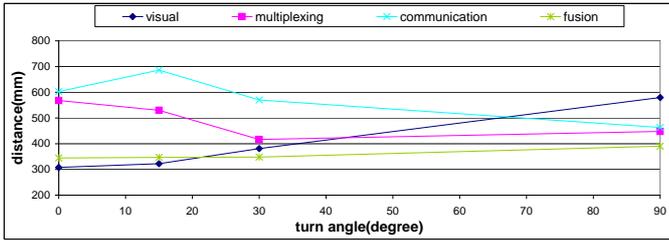


Fig. 21. Deviation from the ideal position in formation vs. the turn angle, with uncertainty levels set at 40%.

The results in Figure 19 show that the worst results are achieved by the visual formation maintenance (closed-loop control, by itself), and by the fusing technique. As the sharpness of the turn increases, use of these techniques lead to increasing errors in the positioning of the follower robots. In contrast, the multiplexing and communication-based maintenance are quite similar in most cases and they have the best results. This happens since in a world where there are fewer odometry errors, a technique that is based on mathematical calculations can calculate the exact location where the follower robot should be and with accurate odometry (i.e. lack of noises) can lead the follower robot to its ideal position in the formation.

However, as odometry noise levels increase, we can see that visual formation maintenance achieves good performance, except for the sharpest (90-degree) turn. Similarly, the fusing technique improves as well, and achieves good performances even in sharp turns. Indeed, the gap between visual maintenance and communication-based maintenance increases (see Figures 20, 21).

Thus one conclusion of these experiments is that the two constituent controllers work well, but not for the same settings. In sharp turns, open-loop control is best (even at higher noise settings). But for robustness to noise, closed-loop control is preferable.

We remind the reader that our hypothesis was that the combination variants would result in decreased precision compared to their constituents. The intuition was that as the combination methods gain the ability to detect obstacles, they sacrifice precision.

The results show that instead, the multiplexing technique emerges as a good controller when the odometry noise level decreases, and is robust in sharp turns as well (a benefit compared to the communication-based controller). It is indeed never the best performer, but it is also never the worst. In fact, this technique seems to be robust both to the noise settings (like its visual maintenance constituent) and to the turn angle sharpness (like the communication-based maintenance constituent). These results thus provide evidence that for robustness, multiplexing controllers (alternating between them) may be a good strategy.

We additionally see that the fusing technique performs well and is robust when the odometry noise level increases. Thus if the robots can recognize its odometry noise level in the environment, it can switch from fusing to multiplexing, and

vice versa (depending the noise) and behave ideally. This second level of multiplexing, however, is beyond the scope of this paper.

D. Robust Formations: Conclusions

We introduce here a combination approach (involving either multiplexing or fusing of controllers) to formation maintenance. The approach combines two different formation maintenance controllers: One open-loop and one closed-loop. Our technique helps to maintain a formation and detect obstacles when the robots' sensors are limited, and therefore cannot easily detect obstacles and track their peers at the same time. In experiments with real and simulated Sony AIBO robots, we have found that the combination approach decreases the number of undetected obstacles (compared to the closed-loop visual formation maintenance controller), and maintains the precision of the formation more robustly than either of its constituent controllers by itself. We also conclude that the level of odometry errors influences the best performer between the multiplexing and fusing methods. In particular, we find that the multiplexing technique is better when the odometry error decreases, and that the fusing is preferable otherwise. Thus, we propose to switch between those two methods when the robots knows their odometry error level (e.g., use multiplexing in flat surfaces, and fusing in rocky terrains. Both techniques allow the robots to use their sensors to detect obstacles, something not possible with their constituent methods.

V. INTERACTING WITH A DISCONNECTED FORMATION

Despite all improvements to the operator interface (Section III) and the formation maintenance method (Section IV), in real-world applications there will be times in which a robot fails to keep track of its teammates, and will become stuck in place, while its peers continue. Such cases require resolution by the operator. The interaction of a single human operator with multiple robots poses significant challenges. Quite literally, an operator has only two hands with which it needs to interact with possibly more than a two robots.

Section V-A introduces the challenges involved. Section V-B presents the distributed call-request approach, which utilizes the organizational knowledge of the robots to resolve call-requests in tightly-coordinated tasks. Section V-C presents the results of extensive experiments evaluating this and other approaches.

A. Call-Requests: Introduction

Robots that require the operator's assistance initiate or are issued *call-requests*, which are queued for the operator. Traditionally, the operator switches control between robots, and uses single-robot teleoperation with individual robots to resolve the call requests in some (prioritized) sequence (e.g., [2], [14]). This method works well in settings where the task of each robot is independent of its peers, and thus the resolution of a call request is independent of others. Here, the operator is used as a *centralized* resource by the robots.

Unfortunately, centralized methods face difficulties in *coordinated tasks*—tasks that require tight, continuous, coordination between the robots, i.e., robot teams where robots are highly inter-dependent. First, due to the coordinated nature of the task, robots depend on each other’s execution of sub-tasks; thus a single point of failure (e.g., a stuck robot) will quickly lead to multiple call requests. Second, when the operator switches control to a robot, the other robots must wait for the resolution of the call-request, because their own decision-making depends on the results of the operator’s intervention. As a result, robots wait idly while the call request is resolved. While monitoring and diagnosis techniques can help localize call-requests to the relevant robot [11], [19], minimizing the duration of call-request resolution remains a key challenge.

Operating a team of coordinated robots raises the opportunity for novel resolution methods, in which the responsibility for the resolution of the call request is *distributed*. Rather than having the operator centrally take all actions to resolve a failure, the otherwise-idle robot teammates can offer assistance, e.g., in providing useful information or in carrying out sub-tasks associated with the resolution process.

For example, in a formation-maintenance task, suppose one of the robots gets stuck, and is unable to move. A call request is issued to the operator, which must identify the failure and attempt to resolve it in some fashion. Previous approaches would have the operator attempt to teleoperate the robot in an attempt to dislodge it, while the other robots are idle [2], [14].

However, the operator could take advantage of the other robots to resolve the failure. First, the other robots could be used to provide video imagery of the stuck robot from various angles. Second, the robots may assist the operator to determine the location of the robots—since they can calculate its expected position with respect to their own position—based on its position within the formation.

This section reports on first steps towards allowing coordinating robots, in a spatial task, to use their knowledge of the coordination to autonomously assist the operator. We examine several variations of a distributed control methodology in which functioning members of the team, rather than switching to an idle mode of operation, actively seek to assist the operator in determining the failure. The key idea is that the responsibility for resolving the call-request is distributed among the team-members *in addition to* the operator.

Moreover, a distributed call-resolution necessarily requires the operator to switch from one robot to the next *while they are moving*. For instance, if the operator is moving one robot, while another requires control, switching time becomes important. Thus the control software on every robot must support quick suspension and resumption of operations, so that switching occurs as quickly as possible. We propose a simple method to enable such quick control-switching.

B. Distributed Call-Request Resolution in a Coordination Task

As previously discussed, centralized resolution of call requests, by the operator, may work well when robots’ tasks are independent of each other. However, in coordinated tasks,

many robots may have to stop their task execution until a call request is resolved, because their own task execution depends on that of the robot that requires the resolution. In such cases, it is critical to minimize the time it takes to resolve a call request.

We thus focus on a distributed control approach, whereby the robots who depend on the resolution of the call-request take active steps to resolve it, in collaboration with the operator. This approach takes advantage of the robot teamwork, by turning the resolution of the call-request into a distributed collaborative task for all involved. Moreover, the active robots (that do not require assistance) are involved in a coordinated effort with the robot requiring assistance, and thus may be in a better position to assist it.

The key idea behind this approach is that call-request resolution is best viewed as an instance of cooperative problem-solving. During task execution, robots collaborate to achieve the operator goal. If task execution is halted due to a failure, a new collaboration problem instance is generated (resolving the call-request), which should then be addressed by the team-members that are affected by the failure, since they have knowledge which they can bring to bear on the problem.

Concretely, we investigate distributed resolution in repairing broken formations of Sony AIBO 4-legged robots. Formation-maintenance tasks require tight, continuous coordination between robots [20]. When a robot fails and is unable to move, the formation cannot proceed until the failure is resolved in some fashion: Either the robot becomes unstuck, or it is declared dead and the formation proceeds without it. A stuck robot often cannot report on why it is stuck, due to sensory range limitations. For instance, in the AIBO robots, the camera (mounted in the head) cannot pan and tilt to cover the rear legs. Thus if one of them is caught by something, the robots own sensors cannot identify it. The robot must then issue a call-request for assistance. The operator, in turn, must use one of the other robots to locate the stuck robot and get video imagery of its state. This act of locating the other robot and getting sufficiently close to it is a key factor in the resolution of the call request in this case.

We construct two variations of distributed call-request resolution. In the first (*semi-distributed*), the robots assist the operator by autonomously beginning to search for the failing robot as soon as the call request is received. The operator views a split-screen view of their video imagery, and as soon as it identifies the stuck robot in one of the displays, can switch control to the robot associated with the display. Once a robot is taken over by the operator, the others become idle. The operator may still switch control to these other robots, but they no longer work in an autonomous fashion.

The fully-distributed scheme exploits all robots through the call resolution process. The operator may teleoperate any robot at any time, and may switch between controlled robots as needed. When not operator-controlled, the robots first head towards the expected position of their stuck peer. This position is estimated based on their knowledge of the formation (organizational knowledge), under the assumption that the robot became stuck in its previous location within the formation. If they fail to find it there, they begin a spiral search

pattern. The robots that maintain the formation have improved chances to localize themselves (and their stuck peer) with respect to the formation, than an operator which takes control of a robot in the formation, without the situational awareness of the robots. On the other hand, the operator has superior inference and vision, and may be able to better identify the stuck robot in the video imagery.

The distributed approach requires the operator to be able to switch control between robots, and for the robots to be active when the switch occurs. Traditionally, when the operator wants to switch control from robot one robot to the next, she would need to turn on (manually) the first robot's autonomous behavior (for working simultaneously, to achieve the common goal). Then she would need to turn off (manually) the second robot's autonomous controller, and take control of the robot.

When robots do not operate in parallel to the operator, as in previous methods, the effects of switching time on performance is negligible. But when using distributed resolution, switching becomes critical, as the robots that is taken control of is continuing to move and turn even while the operator is switching control over it. Any delay here may cause the robot to move away from where the operator intended to go, thus causing cascading failures.

To allow quick suspension and resumption of autonomous control, each robot maintains a suspend flag which causes the motors to ignore (temporarily, if the flag is on) the controller's commands. When an operator uses the interface to take over control of a robot A , giving up control of robot B , A 's associated suspend flag is turned on, and B 's flag is turned off, giving B 's autonomous controller access to the motors again.

C. Experimental Evaluation of Call-Request Resolution Methods

We empirically evaluated the methods discussed in this paper in extensive experiments, with up to 25 human operators. The first set of experiments focused on comparing the distributed resolution methods presented, with alternative, traditional, methods. We simulated failure cases in a triangular formation (three robots). In each case, we disable one of the robots to simulate a catastrophic failure, not letting it move or communicate. Different call-resolution methods were then used to begin the search process. The search stops when any robot is within a predetermined distance of its failing teammate.

A potential advantage of the distributed and autonomous schemes is that they can utilize the robots' own knowledge of the coordination to locate the stuck robot. In particular, because the robots have moved in formation prior to the call-request, they may have an easier time guessing their peer's location than the operator (who needs to orient herself in space via the teleoperated camera).

We therefore examine three scenarios, in which we varied the accuracy of this knowledge. In all failures cases, the right follower robot was disabled, and color marked to allow its detection by the other robots (called *active robots*) and the operator. We varied the position of the disabled robot (Figure

22): The *easy* setup placed the disabled robot at approximately where it would be had it just stopped in its tracks prior to the team getting notification of the call request, i.e., a bit farther behind its location within the formation (Figure 22-a). The *medium* setup placed the robot behind the left follower robot (22-b). The *difficult* setup placed the robot to the left of the left follower robot, and behind it, i.e., completely out of place compared to the formation (22-c). The locations progress from a location easily predictable by the robots, to a location unpredictable to them.

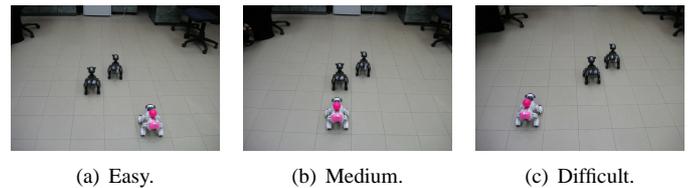


Fig. 22. AIBO robots in initial places for the three experimental setups.

We contrast the distributed and semi-distributed call-resolution methods with two traditional resolution schemes. The first, *teleoperated* scheme corresponds to the centralized control used in previous approaches (e.g., [2], [14]). In this scheme, the operator would switch control from one active robot to the next, as deemed necessary, and manually teleoperated controlled robots (one at a time) until the disabled robot was found. When one robot is controlled, the others remain idle. Another previously-investigated approach is the fully *autonomous* scheme, that lets the active robots (but not the operator) search for the failing robot. This scheme corresponds roughly to the method described in [28], where the robots receive general instructions (here, "search!") by the operator, but are left to translate and follow these commands autonomously, without direct manipulation.

We studied 25 human operators with each of the failure scenarios, each with all methods (22 male, 2 female; 22 of these—including the two females—were graduate or undergraduate students). All operators were novices; none had previous experience controlling multiple robots. The ordering of the scenarios was randomized between operators to prevent ordering effects.

We distinguished two phases: The first phase of the resolution involved recognition of the disabled failure from any distance. The second phase involved its localization by another robot reaching within 35 centimeters of it. Each scenario began with the simulated disabling of the robot (and issuing of the call request), and ended with its localization by at least one robot—teleoperated or autonomous.

For each of the failure scenarios and for each method, we measure the duration of the two phases. This is an objective performance measure because the initial locations of the robots are fixed, the searching speed is constant for non-teleoperated robots, and the termination condition for the search are fixed (robots within specific distance of the failing robot). Thus other than the typical robot sensor uncertainty, performance variance is introduced solely by operator intervention. The first measured duration is that of the time that it took the

operator to recognize the disabled robot in any one of the cameras (the operator uses the split-view interface in this task), i.e., the duration of the first phase. In all but the teleoperated scheme, the operator is completely passive during this interval. We then measure the time that it takes for an active robot—autonomous or teleoperated—to reach the disabled robot, i.e., the duration of the second phase. Since the motivation behind the distributed control scheme is to reduce the time spent awaiting resolution, we prefer shorter overall durations.

We begin by examining the bottom line—the total time it takes to identify the location of the disabled robot. Figure 23 shows the average total duration for the 25 operators. The vertical axis measures the time in seconds, while the horizontal axis shows the three experiment setups. In each, four bars are shown corresponding to the different resolution schemes (left-to-right: Autonomous, semi-distributed, distributed, and teleoperated).

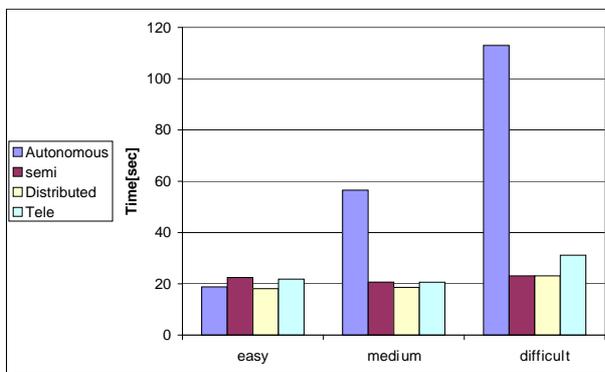


Fig. 23. Total Time to Resolution (in seconds).

The results show that in all *easy*, *medium* and *difficult* locations, the distributed approach is preferable to the both centralized teleoperation approaches, and the fully autonomous approach. Full distributed search does better than the semi-distributed approach in all locations, and better or same than the autonomous approach or same. Overall, the distributed collaboration between the operator and active robots in the distributed approach proves to be a powerful technique for significantly reducing the time to complete the task of locating the disabled robot.

The results have been tested using a one-tailed t-test assuming unequal variances. In the easy setup, the distributed scheme is not significantly different than the autonomous scheme, and only moderately different ($p < 0.12$) than the semi-distributed and teleoperated schemes. However, as we move to the medium and difficult setups, the situation changes. The total time for the distributed scheme is significantly lower than the total time for the autonomous scheme in the latter setups ($p < 0.00004$ and $p < 10^{-12}$, resp.). The distributed scheme does better than the teleoperated scheme in the difficult setup ($p < 0.02$), and is moderately better in the medium setup ($p < 0.13$).

The figure also carries other lessons. First, the ability of the robots to use organizational knowledge of the formation can be very useful in reducing the resolution time, and thus

in assisting the operator. When the stuck robot was located approximately where it was predicted to be in terms of its position in the formations, the robots were able to quickly locate it, in fact beating the operator in terms of total time (see more on this below). However, the distributed scheme was superior even in these cases, because even in where the robots were not as successful, the operator (working in collaboration with the robots) was able to compensate. This is particularly evident as the difficulty of the different setups increased, and the location of the stuck robot was unpredictable to the robots.

To better understand these results, we should consider separately the results for the first phase of the search (when an remote identification of the stuck robot was made by the operator), from the second phase, in which an active robot was to approach the stuck robot to localize it. Figure 24 shows the results of the different control schemes for the first phase, averaged across operators. The figure measures the average time (in seconds) it took the operator to recognize the disabled robot from afar, in the split-view camera display. In the autonomous approach, the operator did not intervene in the operation of the robots, only indicated that the stuck robot was recognized. In the teleoperated scheme, the operator manually turned a robot around until a heading to the remote robot was recognized.

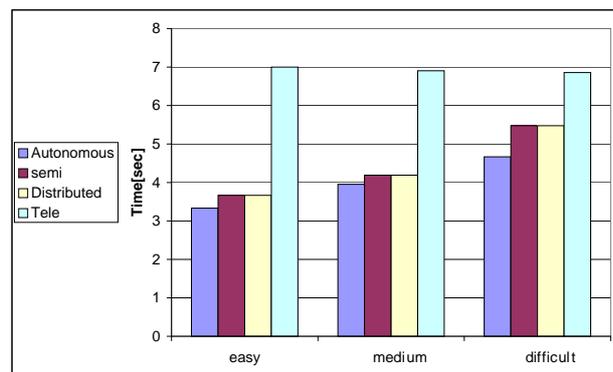


Fig. 24. Phase 1 Time until initial (remote) identification (in seconds).

Clearly, all approaches in which robots attempt to orient themselves towards the predicted location of the disabled are superior to a teleoperated (centralized) approach. Note that in all approaches, the operator recognizes the robot from afar. The active robots do not necessarily recognize the other robot from afar, and as we will see below, may end up searching for it in the wrong location. This significantly shorter initial recognition is a beneficial side-effect of the distributed approaches. However, the initial benefits of the robots to orient themselves towards the stuck robot is lost in more difficult settings.

Figure 24 also shows an important property of the usefulness of human operators: Human ability to recognize the robot from afar is virtually identical in all three difficulty settings. Thus humans bring to bear consistent robust (if slow) capabilities. These can be useful in real applications, where the stuck robot may be partially hidden behind obstacles or otherwise not visible at all to the robots.

An examination of the second phase of the search (once an approximate heading towards the stuck robot is determined) is also telling with respect to this issue. Figure 25 shows the results for this phase, where the task is to arrive within the proximity of the disabled robot. Despite its poor performance in phase 1, the teleoperated approach does quite well in phase 2. This is easily explained—here the disabled robot is already recognized, and the teleoperated approach simply allows the operator to now drive the teleoperated robot as quickly as possible, outrunning automatic approaches that move in constant (and typically conservative) speed. Thus again, the operator brings to bear capabilities that cannot be duplicated by the robots.

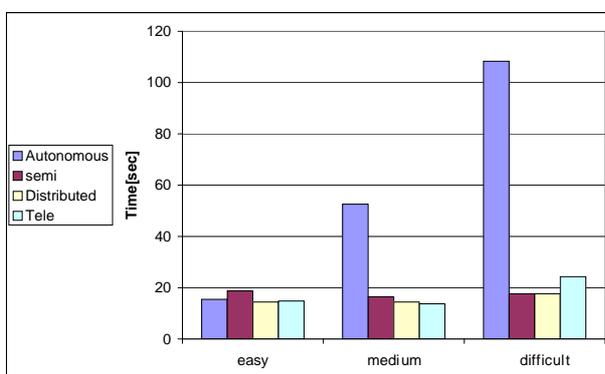


Fig. 25. Phase 2 *From initial identification to localization of the stuck robot* (in seconds).

However, the best performances was by the distributed approach, because it essentially turns this phase into a race between a teleoperated robot and an autonomous robot, as to who gets to the disabled robot first. Moreover, unlike the semi-distributed approach, where there's an overhead of a few seconds while the operator takes over control (see the results for the easy/medium location), here the transition from phase 1 to phase 2 is fairly smooth, because one active robot continues to search even while the operator is taking over control of the other. Thus there is here a composition between the *Autonomous* approach and the *Teleoperated* approach.

Indeed, contrasting the results of the *Autonomous* and *Distributed* approaches is telling. As we move from the easy location to medium to difficult, the gap between the methods is grows in favor to the *Distributed* approach. That happens as a result of the inability of the *Autonomous* approach, to locate the stuck robot in unpredictable places. The collaboration between the human operator and the robot team is superior to either, alone.

An final lesson is revealed by examination of the standard deviation of the results for total task-completion time. Table IV shows the standard deviation for the different approaches, in the three experiment setups. Each row corresponds to a different method, and each column to different setup. We can see that in the easy setup, the autonomous, semi-distributed, and distributed schemes all have essentially the same standard deviation, indicating similar performance. However, the standard deviation for the autonomous scheme in the medium setup is much higher than for the other approaches. In the

	Easy	Medium	Difficult
Autonomous	11.21	34.64	23.82
Semi-Dist.	11.30	5.07	7.78
Distributed	11.29	5.16	7.90
Teleoperated	7.68	5.96	15.87

TABLE IV
STANDARD DEVIATION OF CALL-RESOLUTION TIMES (IN SECONDS).

hard setup, both the autonomous and teleoperated approaches have greater standard deviation in performance than the two distributed schemes. This shows an additional benefit of the distributed methods: A more consistent performance of operators in the distributed and semi-distributed cases.

We now turn to empirically evaluate the importance of the switching latency in the distributed resolution methods. We remind the reader that we proposed a simple mechanism (the suspend/resume flag) that enables quick switching. Otherwise, the operator would need to manually perform several preparatory actions (like, manually turning off a controller) in order to teleoperate a robot.

To test the effects of the control-switch methods, we added two control schemes: *SwitchSemi* and *SwitchDistributed*. This two control schemes are similar to the *semi-distributed* and *distributed* control schemes correspondingly except the transfer from the part of identifies the stuck robot in one of the displays to teleoperate one robot. If the operator in the *SwitchSemi-distributed* and *switchDistributed* approaches want to teleoperate a robot she needs to turn off it search behavior manually. While in the *semi-distributed* and *distributed* approaches its automatically happens and the search behavior is paused. The *semi-distributed* and *distributed* is use our quick-pause technique and the *SwitchSemi-distributed* and *switchDistributed* use the old manually technique.

We tested 21 human operators with each of the three failure scenarios (as described in the previous section), and compare there results with 9 operators using the manual-switching techniques. The results are shown in figure 26 (left-to-right: *SwitchSemi-distributed*, *switch-distributed*, *semi-distributed* and *distributed*). The figure shows the total time to complete the action (First and second phase). We compare each method to its corresponding method (i.e. the manual *SwitchSemi* with the quick *Semi-Distributed*, the manual *SwitchDistributed* with the quick *Distributed*).

We can clearly see that the manual switching methods are far worse than the quick-switching methods. We also see that this effect is of more importance with the more difficult failure cases. This supports our hypothesis that fast operator control-switching is critical to the distributed approaches. Indeed, when we compare the total resolution times of the distributed methods, using the manual switching method, to the autonomous and teleoperated approaches, we find that the advantage of the distributed methods disappears in many cases.

A closer look at the results shows that the differences become apparent in the second phase (Figure 27). The problem arises during the control switching from the first to the second phase. The switching here includes two main steps. First, the time it takes to switch from autonomous behavior to

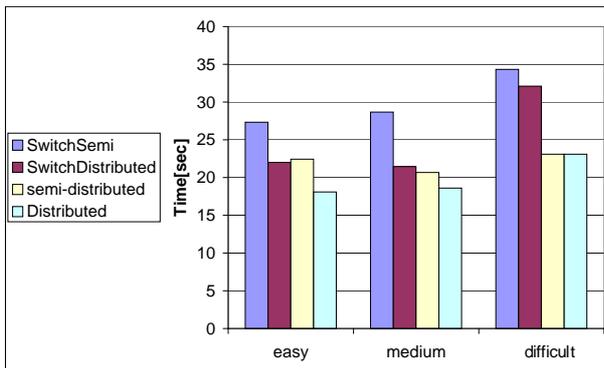


Fig. 26. Total time to resolution, with different switching methods.

teleoperated control must be considered. A secondary issue arises due to this time loss. As the robots act autonomously until the operator assumes control, robots can often take a number of steps after their initial recognition. This results in the robots moving away from the location of their first recognition. We refer to this cascading failure as a *secondary failure*.

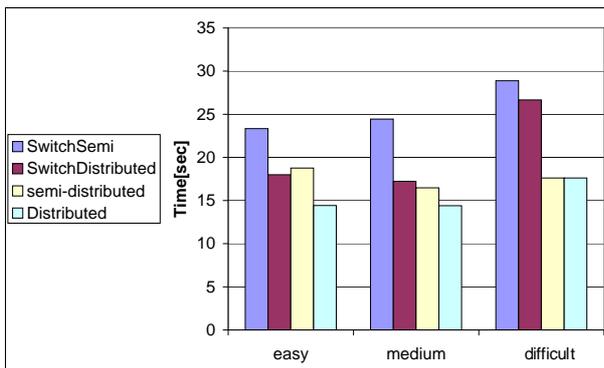


Fig. 27. A comparison of phase-2 durations for different switching methods.

We hypothesize that these secondary failures exist, i.e., that even if the switching duration itself constituted fixed latency, then the effects of this latency would cause additional failures, and thus would vary the actual latency observed. In order to support this claim, we studied the ratio between the *distributed* and *SwitchDistributed* techniques.

Table V shows the difference and ratio between the *distributed* and *SwitchDistributed* methods at easy, medium and hard courses. The results of this table attempt to isolate the delay due to the switching control between the *distributed* and *SwitchDistributed* methods. The tables demonstrates that the difference and ratio between these methods varies. Under the assumption that normal switching behaviors require a fixed amount of time, this result implies that another factor exists that accounts for this variable length. We believe that this factor is the secondary failures caused by the switching latency.

In particular, we believe these secondary failures resulted from the robot's movement from the moment the operator recognized the stuck robot in its video stream, and until the

	Easy	Medium	hard
difference	3.905	2.84	9.015
ratio	1.21	1.15	1.39

TABLE V
DIFFERENCE AND RATIO BETWEEN *distributed* AND *SwitchDistributed* METHODS.

operator was able to control the robot.

The results have been tested using a one-tailed t-test assuming unequal variances. The *SwitchDistributed* scheme was not statistically significantly different for all courses (easy: $\rho < 0.2$, medium: $\rho < 0.3$, hard: $\rho < 0.12$) in respect to the *distributed*. The *SwitchSemi* scheme is statistically significantly different than the *distributed* in all courses (easy: $\rho < 0.012$, medium: $\rho < 0.006$, hard: $\rho < 0.05$).

D. Interacting with a Formation: Summary

This paper explores novel first steps towards distributed call-request resolution schemes, in which the operator and robots collaborate to resolve failures. This scheme is particularly suited to situations where robots are tightly coordinated, and thus are able to use their knowledge of the coordination to effectively assist the operator. The technique builds on a key idea, that the resolution of failures in cooperative tasks should be viewed as a cooperative task in itself. Previous techniques (teleoperation of one robot at a time, autonomous operation of the robots) were meant for tasks that do not require tight coordination between the robots.

We empirically evaluate the distributed resolution methods (and contrast them with previous approaches) in extensive experiments with up to 25 human operators, operating a team of 3 Sony AIBO robots that are moving in formation. The experiments evaluate several concrete call-request scenarios, in which a stuck robot must be located by the operator. The results show that distributed call-request resolution leads to shorter failure-recovery times. Moreover, the results show that a key factor in the success of the distributed method lies in the robots' use of organizational knowledge (i.e., their knowledge of the coordination). However, even in cases where this organizational knowledge fails, the operator is able to compensate. Thus the use of our distributed approach is always better than either the operator or the robots resolving the call request by themselves. We also report on the empirical results of using quick-switch methods (automatic suspension of autonomous activities by a robot, upon the operator switching control to it). A final promising result is that the distributed methods lead to improved operator consistency, reducing the variance in performance between operators.

VI. SUMMARY

This paper tackles key challenges in making formation-maintenance a reality in real-world applications of multi-robot teams. It provides a comprehensive set of techniques that address robustness concerns, both from the perspective of a human operator of the formation, as well as from the point

of view of maintaining greater autonomy by the robot team. Specifically, three sets of techniques are presented:

- 1) First, the paper presented a novel ecological coordination display, which we show improves the performance of human operators guiding a formation through obstacle courses. Performance improves in the time it takes to navigate the chosen path, the number of failures and the failure rate, and the consistency of the operator's success.
- 2) Then, the paper presented an approach for combining sensor-based closed-loop formation-maintenance, and communication-based open-loop formation maintenance, by either fusion or multiplexing in time. The experiments show that this allows robots to better utilize their sensors for detecting and avoiding obstacles, while still maintaining their positions in the formation.
- 3) Finally, the paper presented a novel distributed call-request approach to handling call-requests (operator intervention requests) in tightly-coordinated tasks. The key to this approach is that robots actively collaborate with the operator to resolve conflicts, in particular utilizing their knowledge of the coordination to assist the operator in locating robots that require assistance.

Extensive experiments using real and simulated robots, with up to 25 human operators, show significant improvements over existing methods, in all three contribution areas. The techniques clearly provide a path towards real-world applications of multi-robot formations. Videos showing actual runs in which these techniques were used, as well as videos of related techniques, are available at <http://www.cs.biu.ac.il/~maverick/Movies/> [25].

ACKNOWLEDGMENTS

We thank Avi Rosenfeld for useful comments, and Ruti Glick for help in organizing the experiments. Special thanks to K. Ushi. This work was supported in part by ISF Grant #1357/07.

REFERENCES

- [1] J. Adams, A. Robertson, K. Zimmerman, and J. How. Technologies for spacecraft formation flying. In *Proceedings of ION-GPS-96*, pages 1321–1330, 1996.
- [2] J. A. Adams. *Human Management of a Hierarchical System for the Control of Multiple Mobile Robots*. PhD thesis, University of Pennsylvania, 1995.
- [3] K. S. Ali. *Multiagent Telerobotics: Matching systems to tasks*. PhD thesis, Georgia Institute of Technology, 1999.
- [4] H. Asama, A. Matsumoto, and Y. Ishida. Design of an autonomous and distributed robot system: ACTRESS. In *Proceedings of the 1989 IEEE/RSJ International workshop on intelligent robots and system*, pages 283–290, 1989.
- [5] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.
- [6] T. Balch and M. Hybinette. Social potentials for scalable multirobot formations. In *Proceedings of IEEE International Conference on robotics and automation (ICRA-00)*, 2000.
- [7] A. Broggi, M. Bertozzi, A. Fascioli, C. G. L. Bianco, and A. Piazzini. Visual perception of obstacles and vehicles for platooning. *IEEE Transactions on Intelligent Transportation Systems*, 1(3):164–176, 2000.
- [8] X. Chen and Y. Li. Smooth formation navigation of multiple mobile robots for avoiding moving obstacles. *International Journal of Control, Automation, and Systems*, 4(4):466–479, August 2006.
- [9] J. P. Desai. A graph theoretic approach for modeling mobile robot team formations. *Journal of Robotic Systems*, 19(11):511–525, 2002.
- [10] J. P. Desai, J. P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.
- [11] R. J. Doyle. Determining the loci of anomalies using minimal causal models. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1821–1827, Montreal, Quebec, Canada, 1995.
- [12] M. Fields. Modeling the human/robot interaction in OneSAF. In *Proceedings of the 23rd Army Science Conference*, 2002. (Poster).
- [13] R. Fierro, A. K. Das, V. Kumar, and J. P. Ostrowski. Hybrid control of formations of robots. *IEEE International Conference on Robotics and Automation*, 2001.
- [14] T. Fong, C. Thorpe, and C. Baur. Multi-robot remote driving with collaborative control. *IEEE Transactions on Industrial Electronics*, 50(4):699–704, August 2003.
- [15] J. Fredslund and M. J. Mataric. A general algorithm for robot formations using local sensing and minimal communications. *IEEE Transactions on Robotics and Automation*, 18(5):837–846, 2002.
- [16] B. P. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the International Conference on Advanced Robotics*, pages 317–323, Coimbra, Portugal, Jul 2003.
- [17] M. A. Goodrich and A. C. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.
- [18] C. A. Johnson, J. A. Adams, and K. Kawamura. Evaluation of an enhanced human-robot interface. In *Proceedings of the 2003 IEEE International Conference on Systems, Man, and Cybernetics*, Washington, DC, 2003.
- [19] M. Kalech, G. A. Kaminka, A. Meisels, and Y. Elmaliach. Diagnosis of multi-robot coordination failures using distributed CSP algorithms. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, 2006.
- [20] G. A. Kaminka, R. Glick, and V. Sadov. Using sensor morphology for multi-robot formations. *IEEE Transactions on Robotics*, 2008. To appear.
- [21] G. A. Kaminka and M. Tambe. Robust multi-agent teams via socially-attentive monitoring. *Journal of Artificial Intelligence Research*, 12:105–147, 2000.
- [22] H. K. Keskinpala and J. A. Adams. Usability analysis of a PDA-based interface for a mobile robot. *Human-Computer Interaction*, 2004.
- [23] H. K. Keskinpala, J. A. Adams, and K. Kawamura. PDA-based human-robotic interface. In *Proceedings of the 2003 IEEE International Conference on Systems, Man, and Cybernetics*, Washington, DC, 2003.
- [24] M. Lemay, F. Michaud, D. L  tourneau, and J.-M. Valin. Autonomous initialization of robot formations. *IEEE International Conference on Robotics and Automation*, pages 3018–3023, 2004.
- [25] The MAVERICK Group movies page, Computer Science department, Bar Ilan University. <http://www.cs.biu.ac.il/~maverick/Movies/>, Last checked: Feb 24, 2008.
- [26] F. Michaud, D. L  tourneau, M. Gilbert, and J.-M. Valin. Dynamic robot formations using directional visual perception. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.
- [27] A. I. Mourikis and S. I. Roumeliotis. Optimal sensor scheduling for resource constrained localization of mobile robot formations. *IEEE Transactions on Robotics*, 22(5):917–931, October 2006.
- [28] K. L. Myers and D. N. Morely. Human directability of agents. In *Proceedings of the First International Conference on Knowledge Capture, K-CAP 2001*, Canada, 2001.
- [29] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics and Automation*, 23(5):927–941, 2007.
- [30] P. Ogren and N. E. Leonard. Obstacle avoidance in formation. In *Proceedings of the IEEE Int. Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, 2003.
- [31] R. W. Ricks, C. W. Nielsen, and M. A. Goodrich. Ecological displays for robot interaction: A new perspective. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-04)*, 2004.
- [32] P. E. Rybski, I. Burt, T. Dahlin, M. Gini, D. F. Hougen, D. G. Krantz, F. Nageotte, N. Papanikolopoulos, and S. A. Stoeter. System architecture for versatile autonomous and teleoperated control of multiple miniature robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2001.

- [33] P. Scerri, L. Johnson, D. Pynadath, P. Rosenbloom, M. Si, N. Schurr, and M. Tambe. A prototype infrastructure for distributed robot, agent, person teams. In *AAMAS-03*, 2003.
- [34] M. Skubic, D. Anderson, S. Blisard, D. Perzanowski, and A. Schultz. Using a qualitative sketch to control a team of robots. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-06)*, 2006.
- [35] T. Suzuki, K. Yokota, H. Asama, H. Kaetsu, and I. Endo. Cooperation between the human operator and the multi-agent robotic system: Evaluation of agent monitoring methods for the human interface system. In *Proceedings of the 1995 IEEE/RSJ International conference on intelligent robots and systems*, pages 206–211, 1995.
- [36] The Tekkotsu Homepage. www.tekkotsu.org, 2002.
- [37] A. D. Tews, M. J. Mataric, and G. S. Sukhatme. A scalable approach to human-robot interaction. In *ICRA-03*, 2003.
- [38] S. Venkataramanan and A. Dogan. Nonlinear control for reconfiguration of UAV formation. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2003.
- [39] K. A. Vicente. Ecological interface design: progress and challenges. *Human Factors*, 44(1):62–78, 2002.
- [40] Z. Wang, Y. Hirata, and K. Kosuge. Control a rigid caging formation for cooperative object transportation by multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-04)*, pages 1580–1585, 2004.
- [41] H. A. Yanco, J. L. Drury, and J. Scholtz. Beyond usability evaluation: Analysis of human-robot interaction at a major robotics competition. *Journal of Human-Computer Interaction*, 19(1 and 2):117 – 149, 2004.
- [42] K. Yokota, T. Suzuki, H. Asama, A. Masumoto, and I. Endo. A human interface system for the multi-agent robotic system. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA-94)*, pages 1039–1044, 1994.
- [43] D. Zwillinger, editor. *CRC Standard Mathematical Tables and Formulae*, chapter 4.3, pages 312–314. CRC press, 30th edition, 1995.