Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 03-011

Performance Evaluation of a Multi-Robot Search & Retrieval System: Experiences with MinDART

Paul E. Rybski, Amy Larson, Harini Veeraraghavan, Monica Lapoint, and Maria Gini

February 08, 2003

Performance Evaluation of a Multi-Robot Search & Retrieval System: Experiences with MinDART

Paul E. Rybski, Amy Larson, Harini Veeraraghavan, Monica LaPoint, and Maria Gini
Department of Computer Science and Engineering
University of Minnesota

Abstract

The costs of developing mobile robot teams can be reduced if they are designed to exploit swarm techniques. In this methodology many simple homogeneous units solve complex tasks through emergent behavior. The challenge lies in selecting an appropriate control strategy for the individual units. Complexity in design costs both money and time, therefore a control strategy should be just complex enough to perform the task successfully in a variety of environments, relative to some performance measure. To explore the effects of control strategies and environmental factors on performance, we have conducted two sets of foraging experiments using real robots (the *Minnesota Distributed Autonomous Robotic Team*). The first set of experiments tested the efficacy of localization capabilities, in addition to the effects of team size and target distribution. The second set tested the efficacy of communication. We found that more complex control strategies do not necessarily improve task completion times, however they can reduce variance in performance measures. This can be valuable information for designers who need to assess the potential costs and benefits of increased complexity in design.

1 Introduction

Designing a distributed robotic system using swarm techniques, whereby simple homogeneous units solve a complex task through emergent behavior, is an attractive engineering solution for many reasons [Bonabeau *et al.*, 1999]. Analogous to software design, there are obvious advantages to this modular approach, including reducing a complex task to a simpler, more manageable one. There also arises a natural redundancy in the resultant system, as well as the ability to scale to task with minimal tractability issues [Easton and Martinoli, 2002]. The difficulty lies in determining an appropriate robot control strategy for a given task.

The key in designing such a distributed system is to create an individual that is just smart enough, or just complex enough, to solve the problem. Complexity comes at a price, both in money and time, and by minimizing the complexity, thus the cost, we hope to create a simple, efficient, and tractable system. Of course the question is "How smart, is smart enough?". In a general sense, we believe this is a difficult question to answer, because, not only is the efficacy of a control strategy dependent on the specific task, but also on the environment in which the task is performed.

In exploring the effects of environmental factors and control strategies, we conducted a series of experiments on real robots, the *Minnesota Distributed Autonomous Robot Team* (MinDART), shown in Figure 1. While there are many tasks that could serve as a testbed (e.g. box-pushing, mapping, and distributed sensor array), we chose foraging. In our version of the task, robots locate a target in an enclosed arena, pick up the target, and then drop it off at a designated home base. We used this task in two sets of experiments, whereby we evaluated task performance relative to team size, target distribution, and control strategy. In the first set of experiments, we considered the efficacy of using localization, in addition to the performance effects of team size and target distribution. In the second set, we tested control strategies using communication. In both sets of experiments, it was our hypothesis that the performance would improve (i.e. task completion times would decrease) with the ability to localize or to communicate. We learned that this was not necessarily the case, however, we also learned that we could minimize variance of the task completion times using these more complex control strategies. Our findings can be valuable information for designers, because typically complex control strategies take more time to design and implement, more CPU-time to carry-out (thus more battery power), and often require specialized equipment. These costs may be difficult to justify, unless the task requires more consistency in performance results.

We would like to emphasize that all of our experiments were conducted with real robots. We contend that a rigorous study of system design warrants physical robots, as opposed to simulated, to examine the unforeseen effects of an embodied control strategy. We will further discuss this issue of real versus simulated in the following section.

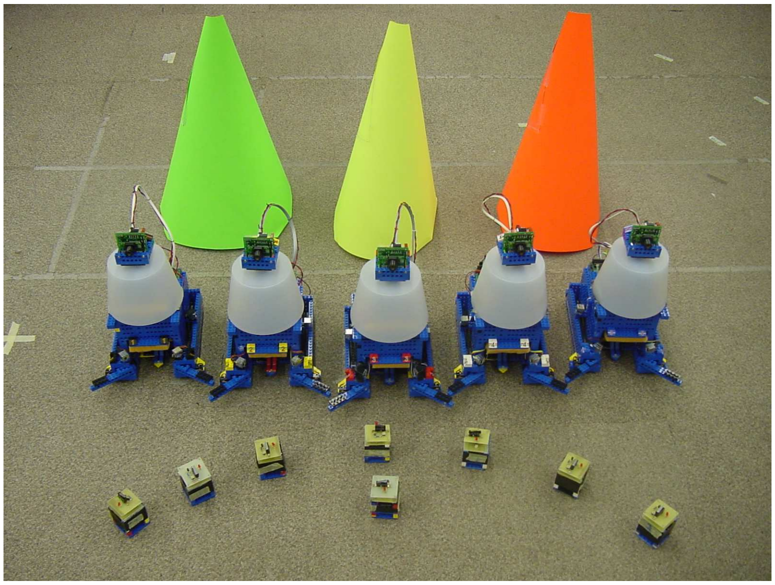


Figure 1: The Minnesota Distributed Autonomous Robotics Team (MinDART) with the infrared targets in front and colored landmarks in back. The MinDART robots searched for the infrared emitting targets in a search and retrieval task. Landmarks were used for homing and localization.

2 Related Work

Most research with multiple robots has focused on various forms of collaborative work as detailed, for instance, in [Arkin and Bekey, 1997, Cao *et al.*, 1997, Dudek *et al.*, 2002]. While collaboration may be essential for some tasks, we are interested in studying tasks that can be done by a single robot, but where using multiple robots can potentially increase performance by decreasing the time to complete the task and/or by increasing the reliability. Sample tasks include mapping a large area [Thrun, 2001], placing a distributed sensor network [Earon *et al.*, 2001], and cleaning up trash [Parker, 1996].

Trash collecting is an example of foraging, which is a widely used testbed application for distributed systems, for example [Carpin *et al.*, 2002, Drogoul and Ferber, 1992, Goldberg and Mataric, 2002] and our previous work [Rybski *et al.*, 1998, Rybski *et al.*, 2002]. In addition to these experimental studies, researchers have developed predictive models of foraging behavior. In [Martinoli *et al.*, 1999], a probabilistic model was developed and verified using simulation and some real robot experiments. Similarly in [Lerman and Galstyan, 2002], a mathematical model was developed and used to study the effects of interference among robots. Optimal foraging theory models are mathematical models used in biology to model foraging behaviors of animals [de Bourcier, 1996, Seth, 2000].

The effect of group size on performance has been well studied. For instance, [Hayes, 2002] presents a quantitative analysis of the tradeoffs between group size and efficiency in collective search tasks. The analysis can be used to predict the optimal number of robots required to complete a task in the most efficient way. The study was done only using simulation.

In [Hayes *et al.*, 2000] simulated and real exploration strategies are compared on an exploration problem where robots start on one side of an arena and have to reach a target area on the other side. The study compares real robots, a sensor based simulation, and a probabilistic simulation model with the objective of assessing the consistency between simulation and real robots. The performance (how long it takes for all the robots to reach the target area) is analyzed for different number of robots and for collaborative robots versus non collaborative robots, where a collaborative robot is one that turns on its own beacon as soon as it sees the beacon in the target area. Collaborative robots perform consistently better than non collaborative robots. However, the performance quickly reaches a plateau which stays constant even when the number of robots increases.

There have also been a handful of studies to evaluate the efficacy of communication strategies applied to the foraging task. Balch and Arkin [Balch and Arkin, 1994] conducted an extensive investigation into the impact of various communication strategies on three separate tasks, including foraging. Communication strategies were based on indirect communication based on cues from the environment. (This form of communication, called *stygmergy* in the biology literature, is commonly used in robotics, for example in [Arkin, 1992, Beckers *et al.*, 1994, Michaud and Yu, 1999].) Sugawara et al. [Sugawara and Watanabe, 2002, Sugawara *et al.*, 1999] also looked at the effects of indirect communication in regards to collection patterns and team efficiency using both experimental results and a mathematical model. The conclusion of both is that communication can improve performance.

It is reasonable to assume that communication will assist in foraging, since it is a strategy that has evolved in nature. It is widely known that bees "dance" to communicate the direction of pollen sources [Seeley, 1989] and ants communicate the location of prey with pheromone trails [Hölldobler and Wilson, 1978]. To our knowledge, biologically-inspired communication strategies for foraging on small scale robots have yet to provide performance improvements as predicted by the above mentioned work.

In those studies, the majority of the experiments were conducted in simulation and were not fully implemented on real robots. Simulation is important to establish the potential of control and communication strategies, but our results serve as a caution to designers, that simulation may be deceiving. Easton and Martinoli [Easton and Martinoli, 2002] had similar findings in a stick pulling experiment using Khepera robots and its simulation environment Webots. They found that the real robot performance was consistently (and we would say significantly) lower than the predicted performance based on simulation. They attribute this discrepancy to sensor error and robot entanglement. It has often been shown (see, for instance, [Drogoul and Ferber, 1992]) that small changes in the behaviors of an individual robot can greatly modify the global behavior of the group.

We are interested in studying the problem of robot control strategies from a rigorous experimental standpoint. We want to examine the kinds of unforeseen effects (such as robot entanglement or hardware malfunction) that are caused by the implementation of algorithms on real robots. Such details may be overlooked or be impractical to implement in a simulation study. While some of the observed effects may be unique to our hardware, we hope that by analyzing the data from real robots we can uncover cases where the performance deviates from the expected norm.

We've observed this difficulty in earlier research involving reinforcement learning [Hougen *et al.*, 1998] on real robots where we were training a robot to solve a nonlinear control problem. Setting up the experimental environment so that the robot would see a proper set of training examples was much more difficult to do than previous simulation work suggested.

3 Robotic Hardware

Each MinDART robot, as seen close up in Figure 2, is constructed out of LEGO Technic blocks, which are lightweight, easy to work with, and ideal for rapid prototyping. The chassis is a dual-treaded skid-steer design, allowing the robot to turn in place. The gripper is an articulated cargo bay that grasps and transports targets. Bumpers, which are used for obstacle avoidance, are located both just beyond the front of a robot's treads and on the back. Infrared sensors are mounted on each side of the robot and in the front to detect targets. Targets transmit an omnidirectional stream of infrared light (modulated at 40 KHz) that is detectable at a range of approximately 70 cm. A light-bulb beacon serves as communication among robots.

To detect beacon activation and to identify landmarks for homing, a CMUCam [Rowe *et al.*, 2002] is mounted on top of a servo-controlled turret. The CMUCam is a small CMOS-based digital camera attached to a Scenix SX microprocessor which captures frames and performs color segmentation on the image. Blob statistics of 2-3 frames per second are computed and sent to the robot's on-board computer, the Handyboard [Martin, 1998]. The frame rate is a limitation of the slow serial port (9600 bps) and clock speed of the Handyboard, which is a 2MHz MC68HC11-based microcontroller with 32K of RAM. Power is provided to the camera and Handyboard by two 9.6V NiCad battery packs.

In our initial experiments that tested the efficacy of localization, the robots were equipped with cadmium sulfide (CdS) photoresistors to detect the presence of light bulb landmarks, which were used for homing and localizing. Light bulb landmarks complicate localization because they are indistinguishable from one another. Localization time is reduced when using the CMUCam because landmarks can be employed that are uniquely identifiable by color.

4 Robot Software

Developed in the multi-tasking language Interactive-C 3.1 [Wright *et al.*, 1996], the control software consists of a set of parallel sensory-motor behavior processes, similar to the subsumption algorithm [Brooks, 1986]. Each process is responsible for handling one segment of the robot's control code by mapping sensors to actuators. When a process is activated by a sensor (e.g. when collision detection is activated by a depressed bumper), it tries to control the actuators. In order to resolve conflicts between processes running in parallel, each process is given a unique priority and control of the robot goes to the process with the highest priority. Subsets of sensory-motor behaviors constitute a state in our finite state machine controller.

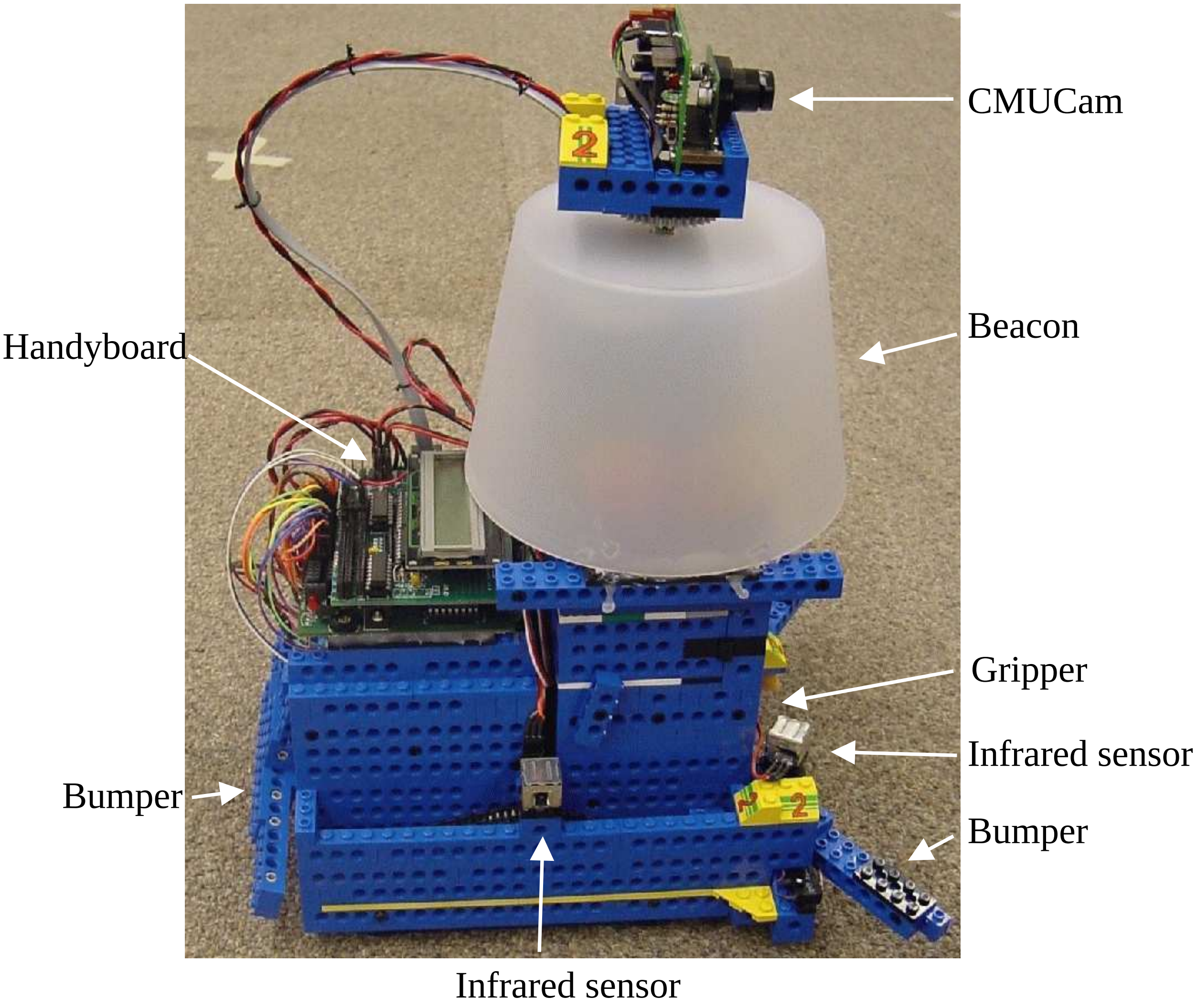


Figure 2: A MinDART robot constructed out of LEGO Technic blocks.

4.1 Finite State Machine Controller

There are three subtasks comprising the search and retrieval task, including find a target, grab a target, and return a target to the home base. These three subtasks correspond to the states of our finite state machine controller as shown in Figure 3. In the initial state, *Find Target*, a robot searches for targets, heads toward a previously seen target (in our localization experiments), or heads toward an activated beacon (in our communication experiments). Once a target is detected with the robot's infrared sensors, the control system switches to the *Grab Target* state which is responsible for maneuvering the robot such that the target fits into the gripper. If the robot successfully grabs the target, the control system switches to *Return Target*, which will return the robot to the drop-off location.

4.2 Behavior Hierarchies

As stated above, each of the states consists of a subset of sensory-motor behaviors executing in parallel, vying for motor control. Behaviors operate independently and are given access to the

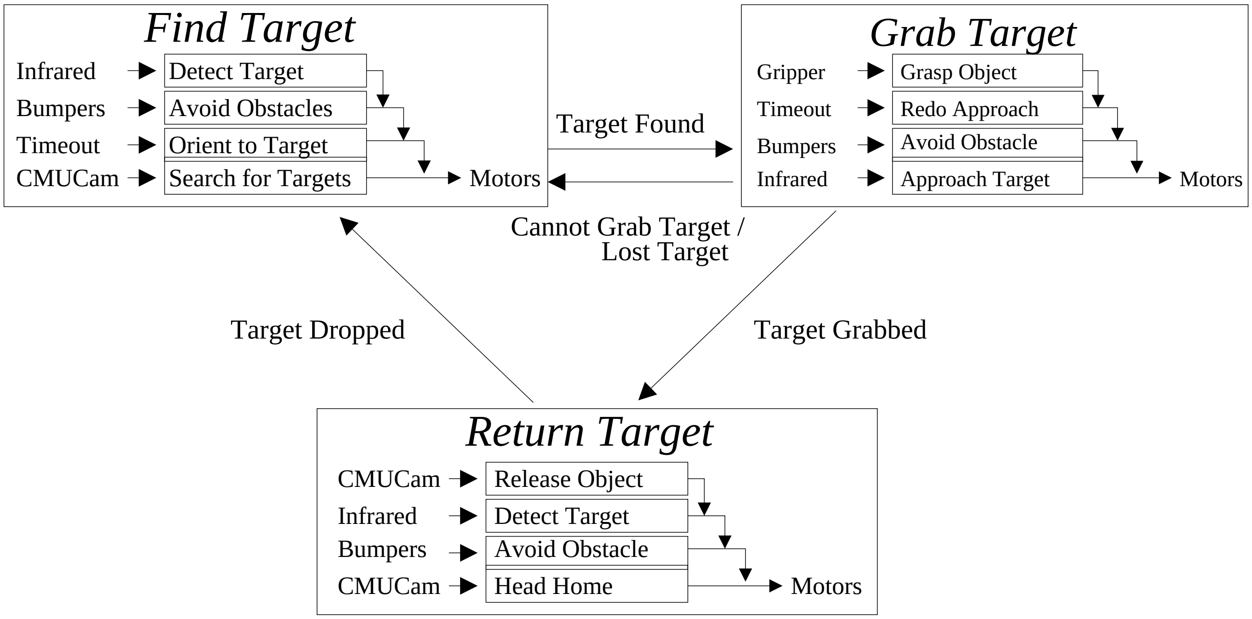


Figure 3: The high-level finite-state machine controller used in our search and retrieval task. Labels on the arrows between the three states indicate state transition events. Within each state, the corresponding sensory-motor behaviors are listed in order of precedence from top to bottom. To the left of each behavior is the sensor that triggers it. The precedence of the behaviors is shown to the right.

motors by an arbiter, based on priority. In Figure 3, the state *Find Target* contains a list of its behaviors in order of priority, from the lowest **Search for Targets** to the highest **Detect Target**. The sensor listed to the left of each behavior triggers its activation. Control of the motors is given to the behavior with the highest priority, which is currently active. The following is a brief description of each of the sensory-motor behaviors of *Find Target*.

Search for Targets drives the robot forward unless the CMUCam detects a beacon. Can optionally change the robot's heading at random intervals.

Orient to Target directs the robot towards position of a previously-seen target (in localization experiments). The robot verifies its heading every 30 seconds. If the target cannot be found at location of sighting, the robot searches for location of the next target on the stack.

Avoid Obstacles directs the robot away from a triggered bumper.

Detect Target monitors infrared sensors for target signals. Detection of target signal triggers state transition to *Grab Target*.

The four behaviors of the *Grab Target* are listed in order of priority in Figure 3. They are as follows.

Approach Target uses infrared sensors to center the robot in front of a target. When the target is aligned, this behavior drives the robot forward. In some communication experiments, the beacon is activated as well.

Avoid Obstacles (different from the **Avoid Obstacles** behavior of *Find Target*) assumes bumpers are activated by collisions with the target and attempts to center the gripper on the "obstacle", as opposed to turning away from it.

Redo Approach moves the robot to approach a target from a different angle after several failed attempts to grasp it.

Grasp Object monitors touch sensor inside the gripper which is indicative of a target within its grasp. When activated, this behavior stops the motors, closes the gripper, and signals the finite state manager to switch the controller to the *Return Target*.

The behaviors of *Return Target* are listed in order of priority in Figure 3. They are as follows.

Head Home drives the robot back to the home base using either landmarks for homing or localization for navigation to a specific (x, y) position.

Avoid Obstacles is identical to **Avoid Obstacles** in *Find Target*.

Detect Target monitors infrared sensors for new targets. In localization experiments, the location is stored for future use. In some communication experiments, the robot stops and activates its beacon for a fixed amount of time in an attempt to recruit other robots to that location.

Release Object activates when target is home, stops the robot, drops the target, and signals finite state manager to switch to *Find Target*.

4.3 Localization

The MinDART localization method assumes the existence of three collinear landmarks at known positions. In this work, light bulb landmarks were identified by CdS photoresistors. Subsequently, we have applied the method using colored landmarks and the CMUCam, which significantly reduced processing time of the localization algorithm. The identified landmarks are used to resolve a robot's (x, y, θ) position in a global frame of reference. Figure 4 illustrates the analytical solution to this localization problem.

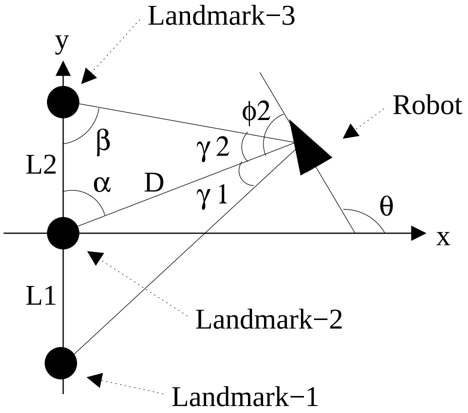
The values of L_1 and L_2 are programmed into the robot *a priori* and are assumed never to change. The robot measures the angles to the three landmarks with respect to its own orientation (ϕ_1, ϕ_2, and ϕ_3)[1], thus $\gamma_1 = (\phi_1 - \phi_2)$ and $\gamma_2 = (\phi_2 - \phi_3)$. The angles α and β and the distance to the center landmark D are solved for and from these values, the robot's global pose (x, y, θ) can be calculated. In our approach, the robot's orientation θ is measured with respect to the global x axis.

This localization method typically estimates the robot's position to within 25 cm and its orientation to within 5 degrees. However, it will fail if it cannot resolve three distinct landmarks, such as if a landmark is occluded.

5 Experimental Description

We conducted two distinct sets of experiments to test the efficacy of localization and communication during a search and retrieval task. Additionally, we tested the impact of environmental factors and

[1] For the sake of clarity, only ϕ_2 is shown in the figure.



$$\text{Let } a = sin(\gamma_1)/L_1$$
$$\text{Let } b = sin(\gamma_2)/L_2$$
$$\text{Let } c = cos(\gamma_1 + \gamma_2)sin(\gamma_1)/L_1$$
$$\text{Let } d = (b - c)/sin(\gamma_1 + \gamma_2)$$
$$D = \sqrt{1/\,(a^2 + d^2)}$$
$$\beta = \pi - (\gamma_1 + \gamma_2) - tan^{-1}(a/d)$$
$$\alpha = \pi - \beta - \gamma_2$$
$$x = D * sin(\alpha)$$
$$y = D * cos(\alpha)$$
$$\theta = \pi + tan^{-1}(y/x) - \phi_2$$

Figure 4: Three-landmark localization used by the robot. The lines connecting the robot to each landmark represent the robot's line of sight. The position of Landmark-2 is the origin of the coordinate system. To avoid problems with symmetry, the robot is only allowed to move in the positive X direction. The values of a, b, c & d are computed through algebraic manipulations involving the Law of Sines and various other trigonometric identities.

team size. In the first set of experiments, the type of search strategy (based on the ability to localize), distribution of targets, and number of robots were varied. In the second experiment, the duration and intent of communication was varied.

5.1 Scalability and localization experiments

In this set of experiments, the robots started from a fixed location, searched an area for targets, and returned targets to one of three drop-off zones. Experiments were run with one-, two-, and four-robot configurations. The robots were not explicitly aware of each other, treating each other as obstacles if they collided. Targets were distributed either uniformly or nonuniformly (i.e. all placed in one far corner of the arena). For each of these configurations, experiments were run with and without localization. Without the ability to localize, a robot's search for targets was random. With localization, the search was purposeful, aimed at a known target location. Figure 5 illustrates the experimental setup.

For each of the experiments, the time that a robot returned a target to a drop-off zone was recorded and averaged over five runs. Table 1 shows the average time in seconds that it took to retrieve all targets, as well as the standard deviation in completion times across all trials. The left half of the table contains results for uniformly distributed targets, while the right half is for a nonuniform distribution. Results across columns differ by team size, and rows differ by use of localization.

Each time a robot localized, it remained stationary for 18 seconds while collecting and processing the landmark data[2]. This 18-second delay had a significant effect on the overall time to complete the task, as reflected in the nonuniform distribution half of the table. There are two major reasons for factoring out localization overhead. First, to find the potential payoff for improvement of the localization technique, and second to determine how much overhead the system can afford while

[2] This slow speed is due to the fact that Interactive-C is an interpreted language, all floating point processing on the MC68HC11 is done in emulation (there is no hardware FPU), and the Handyboard's CPU clockspeed is only 2MHz.

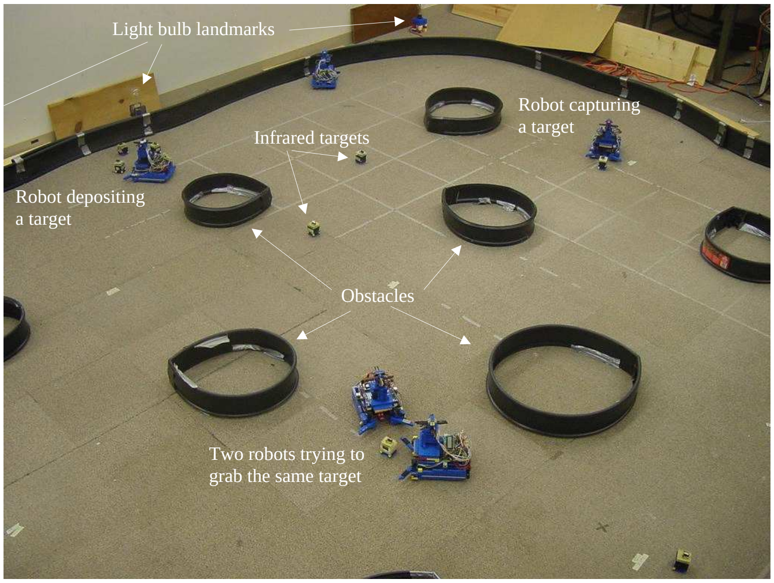


Figure 5: The experimental environment was roughly 5.4 m on a side. All experiments contained nine targets and eight obstacles. Three lightbulbs were placed at known locations and were used to determine position and orientation. Obstacles are relatively low and do not block a robot's view of the landmarks. The obstacles do block a robot's view of the targets.

still improving task performance.

In practice, instantaneous localization would be nearly impossible to achieve, but it is reasonable to assume that the 18 seconds could be significantly reduced. In preliminary testing, we have already reduced localization times to 5 seconds by using the CMUCam. We will discuss this issue further in the Conclusions section.

In looking at how localization affects performance, it can be seen from Table 1 that, as expected, there was no improvement when retrieving uniformly distributed targets. This is because a robot rarely encountered other targets while returning to home base, but even if it did, the target may have already been picked up by the time the robot returned. This, and localization errors that sent the robot in the wrong direction, actually degraded performance when using localization in the uniform distribution experiments.

In the experiments with nonuniformly distributed targets, there was a performance improvement only when factoring out localization overhead. We expected significant improvements in this case, because robots almost always encountered other targets when returning to base, thus a robot could navigate directly from home towards the cache of targets. However, the computational overhead of localization outweighed this benefit of purposeful search.

T tests were run to determine the significance of the non-localization versus localization trials and the non-localization versus instant localization trials. When comparing the times for completing the entire task, only the two- and four-robot trials with the instant localization and nonuniform target distribution were statistically significant at the 95% confidence interval (one-tailed, two-sample t test, $p = 0.0482$ and $p = 0.0291$ for the two- and four-robot cases, respectively.) All other localization results (instant or otherwise) were not statistically significant from the non-localization cases. However, the variance of the two robot localization case with the nonuniform target distribution was close to being significant at the 95% confidence interval (one-tailed, two-sample f test, $p = 0.0638$).

		uniform			nonuniform		
	# of robots	1	2	4	1	2	4
mean	no localize	934	458	374	1672	1058	587
	localize	1108	478	344	1911	1030	593
	instant localize	932	467	323	1195	***720**	***444**
$\sqrt{\textbf{variance}}$	no localize	179	65	46	263	187	106
	localize	282	101	84	304	80	82
	instant localize	301	98	63	286	84	56

Table 1: The top half of the table (mean) is the average time in seconds for the last target to be dropped at home base. The bottom half ($\sqrt{variance}$) is standard deviation of completion times across all trials. Results across columns differ by team size, and rows differ by use of localization. The left half contains times for uniformly distributed targets, while the right half is for a nonuniform distribution. Star (*) indicates statistically significant difference at the 95% confidence level between *instant localize* and *no localize* results of the same column.

One benefit of localization that we did find is how quickly the robots were able to find a new target once it had dropped one off. Table 2 illustrates the average time. Once again, the localization and instant localization results were compared against the no localization results for statistical significance. For this data, the means of all three of the instant localization with nonuniform target distributions were significant (one-tailed, two-sample t test, $p = 0.0085$, $p = 0.0032$, and $p = 0.0371$ for the one-, two- and four-robot cases.) The variances of the two- and four-robot localization cases were statistically significant (one-tailed, two-sample f test, $p = 0.0006$, and $p = 0.0125$ for the two-, and four-robot cases). The variances of the two- and four-robot instant localization with nonuniform target distributions were also significant (one-tailed, two-sample f test, $p = 0.0004$, and $p = 0.0004$, for the two- and four-robot cases). All other localization results (instant or otherwise) were not statistically significant from the corresponding non-localization results.

		uniform			nonuniform		
	# of robots	1	2	4	1	2	4
mean	no localize	83	57	64	150	181	142
	localize	96	65	79	131	139	152
	instant localize	89	65	72	***78**	***86**	***94**
$\sqrt{\textbf{variance}}$	no localize	65	36	57	104	136	100
	localize	101	55	54	96	***77**	***63**
	instant localize	99	55	51	93	***57**	***49**

Table 2: The top half of the table (mean) is the average time in seconds for a robot to find a target after dropping off another. The bottom half ($\sqrt{\text{variance}}$) is standard deviation of target search times across all trials. Results across columns differ by team size, and rows differ by use of localization. The left half contains times for uniformly distributed targets, while the right half is for a nonuniform distribution. Star (*) indicates statistically significant difference at the 95% confidence level between *instant localize* and *no localize* results of the same column.

To evaluate how team size affects performance, we calculated the speed-up S_n and efficiency

E_n for the two- and four-robot teams, which are defined as follows.

$$S_n = \frac{t_1}{t_n} \quad E_n = \frac{S_1}{n}$$

where n is the number of robots and t_n is the time it takes n robots to retrieve all targets. The system is said to have linear speed-up if $S_n = n$, superlinear if $S_n > n$, and sublinear if $S_n < n$. Results are shown in Table 3. As expected, the efficiency of two-robot teams was better than the efficiency of four-robot teams.

	uniform				nonuniform			
	2 robots		4 robots		2 robots		4 robots	
	S_2	E_2	S_4	E_4	S_2	E_2	S_4	E_4
no localize	2.04	1.02	2.50	0.62	1.58	0.79	2.85	0.71
localize	2.32	1.16	3.22	0.81	1.86	0.92	3.22	0.81
instant localize	2.00	1.00	2.69	0.76	1.66	0.83	2.69	0.67

Table 3: Speed-up and efficiency for two- and four-robot teams based on comparison of task completion times. Efficiency values of 1 indicate linear speed-up, less than 1 indicate sublinear.

5.2 Communication experiments

The experimental setup for the communication experiments was nearly identical to the previous. The main difference was the use of a single colored landmark, which robots identified with the CMUCam. This is in contrast to the three light landmarks of the previous experiments, which were identified with CdS photoresistors. Three landmarks were not needed for these experiments since the robots did not explicitly localize themselves. The targets were distributed in a single non-uniform distribution in the corner of the environment furthest from the drop-off location. All experiments were run with four robots. Robots communicated with their light-bulb beacons, which could be seen by another robot at a maximum range of 2.9 m. Figure 6 shows a view of the experimental setup. Communication varied by intent and duration.

No Communication. This was used as a baseline experiment.

Reflexive Communication. A robot turned its beacon on while trying to pick up a target (i.e. while in the *Grab Target* state). Once the robot grabbed the target, the beacon was deactivated. We consider this a statement of action, not a request for help. When used in an environment with a single clump of targets, other robots would be attracted to a robot which was trying to pick up a target from the group. This strategy would most likely be counterproductive in an environment with a uniform distribution.

Deliberative Communication. A robot turned on its beacon to request assistance in picking up a found target. If a robot encountered a target while on its way to the home base to drop off one that it had previously picked up, the robot would activate its beacon and stay motionless for a fixed amount of time. We tested three fixed durations : 10 s, 20 s, and 30 s.

For each of the experiments, the time that a robot returned a target to the drop-off zone was recorded and averaged over five runs. Each experiment was run until all nine targets were retrieved.

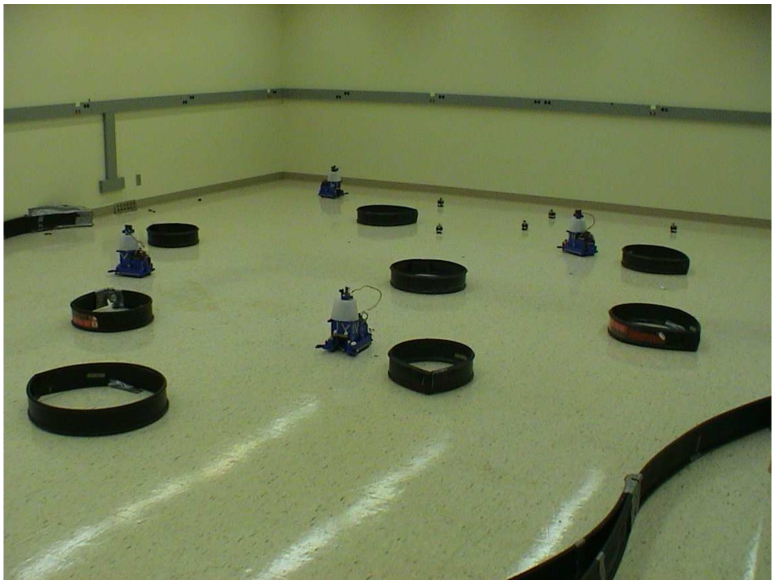


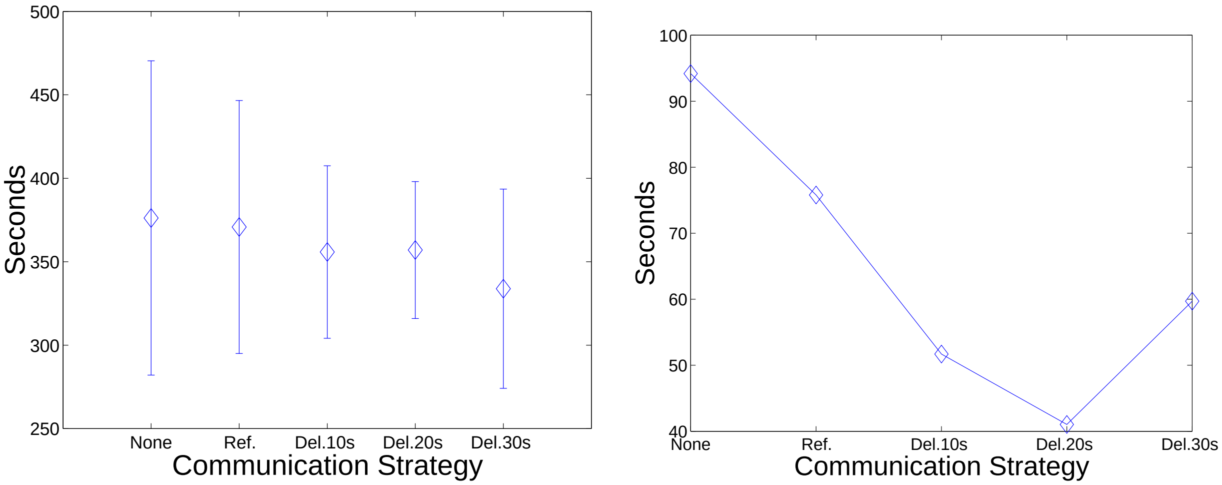
Figure 6: The environment in the communication experiments was approximately 7 m x 8 m. All experiments contained nine targets and eight obstacles. As before, the obstacles were relatively low and did not block a robot's view of the landmarks or of each other. However, they did block a robot's view of the targets.

We compared the times between the dropping off of the first and eighth target, to discount the times in the experiment when communication had little effect. Figure 7 shows the means and standard deviations of these times.

In these experiments, communication was intended as a method of reducing the time that robots would have to search for targets, thus improving overall task performance. Interestingly, there is only a slight trend in the means of the total time to complete the task (see Figure 7) but nothing statistically significant, and no discernible improvements for target search times (see Figure 8). The variance of both have a very obvious trend, as can be seen from the figures. Although f tests show no statistical significance at the 95% confidence interval, the variance of the 20 second communication trials were very close to being significant (one-tailed, two-sampled f test with p=0.0682 and p=0.0511, for time to completion and target search times, respectively).

What is not obvious from these figures is how the duration for the reflexive communication experiments fits into this graph since the communication duration was variable. The average light-on time for the reflexive communication experiments was approximately 16 s with a standard deviation of 11.6 s. However, the distribution of times is not Gaussian. Instead, most of the distribution is clumped down in the region close to 10 s and shorter. As can be seen from the histogram of the on-times, shown in Figure 9, the mode of the distribution is less than 10 s. The decrease in variance from the 10 s communication duration experiments to the 20 s communication duration experiments suggests a correlation between the length of communication duration and the performance variance. As the communication duration of the reflexive communication experiments is less than the 10 s communication experiments, the results match the hypothesis that the variance in the reflexive communication results would be greater.

The lowest variance in drop times occurred when using the 20 seconds deliberative communication strategy. For communication durations less than 20 seconds, the robots often did not have



Means with standard deviation ($\sqrt{\text{variance}}$) errorbars

Standard deviations only

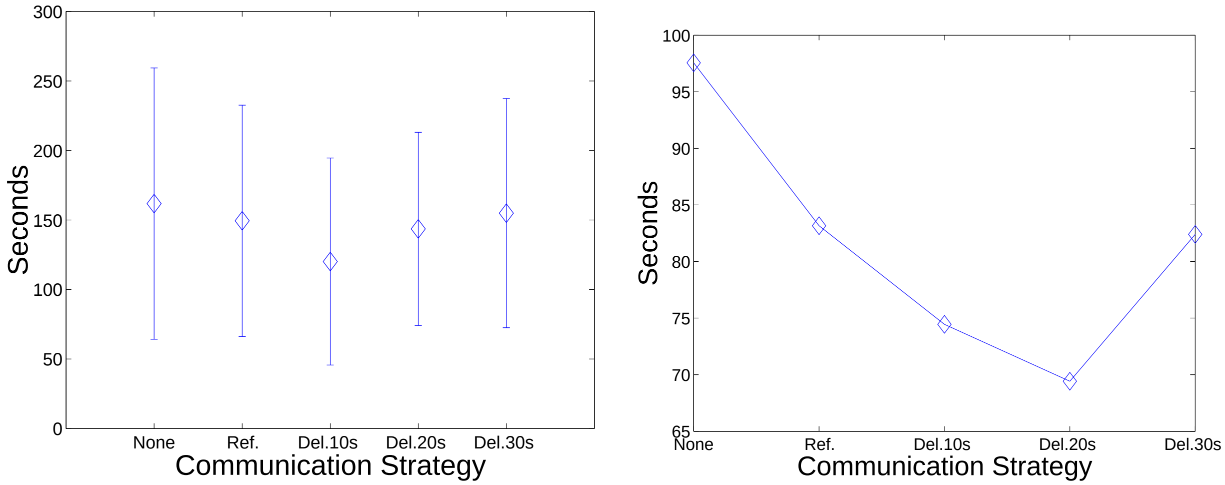
Figure 7: Means and standard deviations of the times to complete the task for each of the communication strategies. The standard deviation graph, on the right, shows a discernable trend. The labels on the x axis stand for the different communication experiments. None=none, Ref.=reflexive, Del.10=10 s deliberative, Del.20=20 s deliberative, Del.30=30 s deliberative.

enough time to reach the target before the beacon turned off, which helps to explain the higher variance in the times. For the 30 second communication, where the searching robots would have ample time to reach the transmitting robot, the variance increases. We conjecture that communication is most advantageous when information regarding available targets is given in such a way as to not interfere with other robots acquiring targets and returning to home base. Interference between robots can occur when more than one robot attempts to retrieve the same target at the same time, as well as when robots get too close to a communicating robot and impede the latter's progress. This happen most often when the recruited robot reaches the communicating robot before the former detects the target that the latter detects. One example of this is if the communicating robot occludes the target from the sensors of the recruited robot. We believe the potential for interference between robots causes the larger variations in drop off times.

To analyze this, we need to take a closer look at how the robots perform and the times necessary to do certain tasks. The CMUCam turrets can rotate 360 ° and survey the robot's surroundings in 5 seconds. However, as the size of the beacon in the camera's image plane directly affects whether it will be detected, the further the robot is away from the beacon, the less likely it will detect it. Thus, it may take several rotations of the turret before a beacon is identified. Once the beacon is found, the robot rotates its body to face the beacon so that it can drive toward it. The robots can rotate 180 ° in 5 seconds and can translate at a maximum of 0.17 m/s. Using these ranges and approximating probabilities for each of the values, we calculated the mean interference time for each of the deliberative experiments as

$$E(x) = \sum_{i=0}^{5s} \sum_{j=0}^{5s} \sum_{k=0}^{2.9m} p(i,j,k) * (C - B_i - O_j - D_k/.17m)$$

where $p(i,j,k)$ is the probability of $(i \wedge j \wedge k)$, C is communication duration, B is time to find a beacon, O is time to rotate the robot's body to aim at the beacon, and D is distance between



Means with standard deviation ($\sqrt{\text{variance}}$) errorbars

Standard deviations only

Figure 8: Means and standard deviations of the times the robots took to retrieve a new target after dropping one off (i.e. target search time) for each of the communication strategies. Again, we see a discernable trend in the standard deviation, as shown in the graph on the right. The labels on the x axis stand for the different communication experiments. None=none, Ref.=reflexive, Del.10=10 s deliberative, Del.20=20 s deliberative, Del.30=30 s deliberative.

the robots. Probabilities were uniformly distributed across O and D, but were varied across B. The greater the distance between robots, the higher the probability of needing multiple passes to find the beacon. From these, an estimate of the amount of time that a robot will interfere with another can be calculated. These results and the mean travel time, or the average time the robot had to travel towards the communicating robot (once it has oriented its body properly), are shown in Table 4.

	Communication Duration		
	10 seconds	20 seconds	30 seconds
Mean Interference Time	0.9971	5.8964	13.8634
Mean Travel Time	2.8500	11.7006	21.6406

Table 4: Interference time is the time in seconds that the recruited robot will interfere with the communicating robot. Travel time is the time in seconds that the recruited robot has to drive towards the communicating robot, once it has found the beacon and properly oriented itself. Interference times are overestimated because we did not consider travel time for obstacles.

The mean interference times are overestimates because we did not take into account the time to do obstacle avoidance. The targets are situated between several obstacles, thus a recruited robot will probably have to navigate an obstacle to get to the beacon. Additionally, the position of the remaining targets affects the probability of interference occurrences. If a target is detected before the robot reaches the beacon, target acquisition takes precedence.

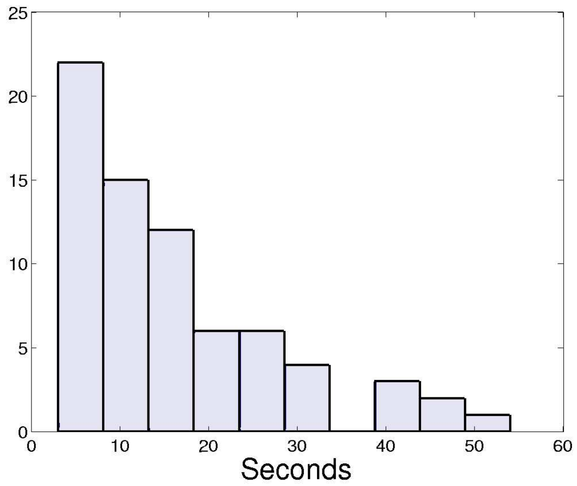


Figure 9: Histogram of the beacon-on times observed in the reflexive communication experiments. It shows that, while on times were variable, the majority were in the 5 to 10s range. This gives a sense of comparison to the deliberative communication experiments where on times were fixed.

6 Conclusions

We have analyzed how the performance of a robotic team is affected by environmental factors, the number of robots, and the search strategy employed by these robots. We expected that localization would greatly assist the robots in the non-uniformly distributed environment but not in the uniformly-distributed environment. What we found was that localization did not improve performance, and in some cases, actually degraded performance. This was due to the time that it took to localize. By using the CMUCams to localize on three unique and collinear landmarks, we can drop our localization time down to 5 s. While this is an improvement over 18 s, the overhead of repeated invocations of the localization routine still noticeably reduced performance as compared to instant localization. If the time to localize was completely discounted (instant localization), the robots were much faster at finding their way back to a new target once one had been dropped off.

Another hypothesis we had was that adding more robots would greatly increase the performance of the team, but continually increasing the number of robots wouldn't be as beneficial. This was proven true in that four robots generally didn't improve the performance over two robots as much as two robots did over one. Additionally, we observed significant interference between the robots when they tried to obtain targets in the non-uniformly distributed environment, which added further evidence to this claim. However, when analyzing the variances of the task completion times for these experiments, in nearly all cases, the variances can be seen to decrease as the number of robots increased. Thus, while the average runtime may not improve as much when adding more robots, the performance of the team will become more consistent. These results show that some knowledge about the structure of the environment is very important when choosing a search strategy for a team of robots.

The communication experiments were designed to test the robot's abilities to lead each other to a single clump of targets. One hypothesis was that once one robot located the group of targets, it would lead the others to that group and thus the overall time to pick up the targets would decrease. Interestingly enough, there was no statistically significant difference in the time necessary to complete the entire task between any of the communication experiments and the non-communication experiments. Instead, a trend towards minimization of the variance in the performance was found. Robots that communicated typically had reduced variance in the time to complete the task as well as in the time to retrieve a new target after dropping one off. While these variances were not statistically significant at the 95% confidence interval, the 20 s communication experiments were very nearly so as compared to the other values.

Another hypothesis for the communication experiments was that the duration of the communication would affect the team's overall performance and that some "critical" duration exists which would maximize the performance. If performance is measured in performance reliability (lack of variance), then performance was the best in the 20 s communication experiments. Increasing or decreasing the communication time increased the variance and thus decreased the reliability.

Other researchers have found that critical values of communication duration have been shown to improve the performance of robotic systems [Sugawara *et al.*, 1999]. In these experiments, the duration of the communication was an exponent of a power law that reflected how the task completion time was related in the number of robots. Our results show that communication duration can be used to describe performance as a measure of reliability as well.

We found that reflexive communication turned out to be too variable to perform as well as the other deliberative strategies. This suggests that communication that relies on the interactions of the robots with their environment would have to be carefully engineered to work properly. Additionally, we hypothesize that such communication may work better when used with larger swarms, but more experimentation is necessary to determine whether this is the case.

With that said, this hand-waiving of an implementation detail raises an important issue. These implementation details are precisely why we think real robots are necessary for this type of analysis. It is too easy to discount or underestimate the effects of even simple implementations on real hardware. It may be that a perfect implementation or perfect knowledge about the environment (or even almost perfect) can improve performance, but we all know, nothing is perfect in real hardware. In [Balch and Arkin, 1994], communication was shown to improve performance, but nearly all of the experiments were done in simulation where the effects of specific actions on the performance of the system (such as cooperative carrying or consuming of a resource) can be abstracted away. The details involved in physically implementing a system which can carry heavy objects or can consume liquid from a spill may affect the performance of the team in ways that those results did not illustrate. Considerable engineering effort may be necessary before the robots would be able to effectively achieve their tasks at the rates reported in this work.

As a point of comparison, the MinDART require the use of the behaviors in the *Grab Target* state to grasp a target. The time that it takes the robots to accomplish this is therefore heavily dependent on the interactions between the robots and their environment. To better quantify this, the times that the robots turned their beacons on in the reflexive communication experiments, as illustrated in Figure 9, were also the times that the robots spent in the *Grab Target* state. As can be seen, these times were quite variable which illustrates the complexity that can arise from a simple operation implemented on robots operating in the real world.

7 Acknowledgments

This work supported in part by the Doctoral Dissertation and by the Louise T. Dosdall Fellowships of the University of Minnesota and by NSF under grant EIA-0224363. Our thanks to Daniel Boley for his insights into the localization algorithm and in computing a closed-form solution. We would like to thank the following students for their contributions to various aspects of the project: Heather Metcalf and Devon Skyllingstad for their help in constructing the electronics for the robot's beacons; Anne Schoolcraft, Michael Lindahl, and Sarah Osentoski for their help in running experiments, constructing targets and helping to implement the localization algorithm; and Joseph Djugash, Ashutosh Jaiswal, Esra Kadioglu, Elaine B. Rybski, Sirintorn Clay, Adam Thorne, David Littau, and Lorry Strother for their help in the construction of the initial robots and collecting data.

References

[Arkin and Bekey, 1997] Ronald C. Arkin and George A. Bekey, editors. *Robot Colonies.* Kluwer, 1997.

[Arkin, 1992] R. C. Arkin. Cooperation without communication: Multi-agent schema based robot navigation. *Journal of Robotic Systems*, 9(3):351–364, April 1992.

[Balch and Arkin, 1994] T. Balch and R.C. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52, 1994.

[Beckers *et al.*, 1994] R. Beckers, O. E. Holland, and J. L. Deneubourg. From local actions to global tasks: Stigmergy in collective robotics. In *Artificial Life IV*, pages 181–189. MIT Press, 1994.

[Bonabeau *et al.*, 1999] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, 1999.

[Brooks, 1986] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, March 1986.

[Cao *et al.*, 1997] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng. Cooperative mobile robotics: antecedents and directions. *Autonomous Robots*, 4(1):7–27, 1997.

[Carpin *et al.*, 2002] S. Carpin, C. Ferrari, and E. Pagello. Mapfocus: A way to reconcile reactivity and deliberation in multirobot systems. *Robotics and Autonomous Systems*, 41:245–255, 2002.

[de Bourcier, 1996] P. de Bourcier. Using A-life to study bee life: the economics of central place foraging. In *From Animals to Animats. Proc. Int'l Conf. on Simulation of Adaptive Behavior*, 1996.

[Drogoul and Ferber, 1992] A. Drogoul and J. Ferber. From tom thumb to the dockers: Some experiments with foraging robots. In *From Animals to Animats. Proc. Int'l Conf. on Simulation of Adaptive Behavior*, pages 451–459, 1992.

[Dudek *et al.*, 2002] Gregory Dudek, Michael Jenkin, and Evangelos Milios. *A taxonomy for multi-agent robotics*, chapter 1. A K Peters Ltd, 2002.

[Earon *et al.*, 2001] E.J.P. Earon, T. D. Barfoot, and G.M.T. D'Eleuterio. Development of a multi-agent robotic system with application to space exploration. In *Advanced Intelligent Mechatronics*, July 2001.

[Easton and Martinoli, 2002] K. Easton and A. Martinoli. Efficiency and optimization of explicit and implicit communication schemes in collaborative robotics experiments. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 2795–2800, 2002.

[Goldberg and Mataric, 2002] D. Goldberg and M. J Mataric. Maximizing reward in a non-stationary mobile robot environment. *Autonomous Agents and Multi-Agent Systems*, 2002.

[Hayes *et al.*, 2000] A. T. Hayes, A. Martinoli, and R. M. Goodman. Comparing distributed exploration strategies with simulated and real robots. In L. E. Parker, G. Bekey, and J. Bahren, editors, *Proc. Int'l Symp. on Distributed Autonomous Robotic Systems"*, pages 261–270, 2000.

[Hayes, 2002] Adam T. Hayes. How many robots? group size and efficiency in collective search tasks. In *Proc. Int'l Symp. on Distributed Autonomous Robotic Systems"*, pages 289–298. Springer Verlag, 2002.

[Hölldobler and Wilson, 1978] B. Hölldobler and E.O. Wilson. The multiple recruitment systems of the african weaver ant *oecophylla longinoda* (latreille). *Behavioral Ecology and Sociobiology*, 3:19–60, 1978.

[Hougen *et al.*, 1998] Dean F. Hougen, Paul E. Rybski, and Maria Gini. Repeatability of real world training experiments: A case study. *Autonomous Robots*, 6(2):281–292, 1998.

[Lerman and Galstyan, 2002] Kristina Lerman and Aram Galstyan. Mathematical model of foraging in a group of robots: effect of interference. *Autonomous Robots*, 13:127–141, 2002.

[Martin, 1998] Fred G. Martin. *The Handy Board Technical Reference.* MIT Media Laboratory, Cambridge, MA, 1998.

[Martinoli *et al.*, 1999] A. Martinoli, A.J. Ijspeert, and F. Mondada. Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots. *Robotics and Autonomous Systems*, 29:51–63, 1999.

[Michaud and Yu, 1999] Francois Michaud and Minh Tuan Yu. Managing robot autonomy and interactivity using motives and visual communication. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 160–167, Seattle, WA, USA, 1999. ACM Press.

[Parker, 1996] Lynne E. Parker. On the design of behavior-based multi-robot teams. *Journal of Advanced Robotics*, 10(6), 1996.

[Rowe *et al.*, 2002] Anthony Rowe, Charles Rosenberg, and Illah Nourbakhsh. A low cost embedded color vision system. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2002.

[Rybski *et al.*, 1998] Paul E. Rybski, Amy Larson, Michael Lindahl, and Maria Gini. Performance evaluation of multiple robots in a search and retrieval task. In *Workshop on Artificial Intelligence and Manufacturing: State of the Art and State of Practice*, pages 153–160, Albuquerque, NM, August 1998. AAAI Press.

[Rybski *et al.*, 2002] Paul E. Rybski, Amy Larson, Anne Schoolcraft, Sarah Osentoski, and Maria Gini. Evaluation of control strategies for multi-robot search and retrieval. In Maria Gini, editor, *Proceedings of The 7th International Conference on Intelligent Autonomous Systems (IAS-7)*, Marina Del Rey, CA, March 2002.

[Seeley, 1989] T.D. Seeley. The honeybee colony as a superorganism. *American Scientist*, 77:546–553, 1989.

[Seth, 2000] A. Seth. Unorthodox optimal foraging theory. In *From Animals to Animats. Proc. Int'l Conf. on Simulation of Adaptive Behavior*, 2000.

[Sugawara and Watanabe, 2002] Ken Sugawara and Toshinori Watanabe. Swarming robots – foraging behavior of simple multi-robot systems. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2002.

[Sugawara *et al.*, 1999] Ken Sugawara, Ikuo Yoshihara, and Kenichi Abe. A scaling law between the number of multirobots and their task performance. In *Artificial Life and Robotics*, volume 3, pages 122–126, 1999.

[Thrun, 2001] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *Int'l Journal of Robotics Research*, 20(5):335–363, 2001.

[Wright *et al.*, 1996] Anne Wright, Randy Sargent, and Carl Witty. *Interactive C User's Guide.* Newton Research Labs, Cambridge, MA, 1996.