# Evaluation of Control Strategies for Multi-Robot Search and Retrieval

Paul E. Rybski, Amy Larson, Anne Schoolcraft, Sarah Osentoski and Maria Gini
*Department of Computer Science and Engineering, University of Minnesota, U.S.A.*
{rybski,larson,schoolcr,osentosk,gini}@cs.umn.edu

**Abstract.**
We are interested in studying how environmental and control factors affect the performance of a homogeneous multi-robot team doing a search and retrieval task. In particular, we looked at the affects of target distribution (uniform or nonuniform), the number of robots, and search strategies (purposeful or random). In our experiments, purposeful search employs localization so that a robot can determine its position and remember locations of target sightings. During random search, a robot has no knowledge of either its own or a target's location. Regardless of search strategy, the robots perform the task without any explicit communication or knowledge of their teammates. Although their sensor suite is very limited, the robots are still able to complete their task. We analyzed the performance of a series of experiments and we present the results.

## 1 Introduction

Cooperating teams of robots have the potential to outperform a single robot attempting an identical task. Increasing task or environmental knowledge may also improve performance, but increased performance comes at a price. In addition to the monetary concerns of building multiple robots, the complexity of the control strategy and the processing overhead can outweigh the benefits. In this paper, we explore these tradeoffs by comparing single robot to multi-robot team performance, as well as examining the benefits of increased intelligence in the form of environmental knowledge.

We propose a task of search and retrieval whereby robots locate, collect, and return targets to a home base. Robots are homogeneous and perform independently with a localized goal of target retrieval without the aid of communication. The task is a simplified version of minefield clearing where mines are localized using close-proximity sensors such as magnetometers, or of a search-and-rescue task where robots find and retrieve specific targets such as those dropped by air. For performance evaluation relative to strategy, the research questions we address in this paper are: How does the distribution of the targets in the environment affect performance? Is the ability to explicitly localize helpful in solving the task? How does the number of robots operating in the same area affect performance?

## 2 Related Work

Most research with multiple robots has focused on various forms of collaborative work as detailed, for instance, in [2, 6]. While collaboration may be essential, we are interested in studying tasks that can be done by a single robot, but where using multiple robots can potentially increase performance either by decreasing the time to complete the task or by increasing the reliability. Sample tasks include cleaning up trash, mapping a large area, and placing a distributed sensor network. For this type of task, cooperation usually requires communication among the robots [7, 9, 10]. Even simple communication has been shown to substantially increase the performance of robots when foraging, consuming, and grazing [3]. However, direct communication can be replaced by indirect communication via sensing or via the environment [1, 4].

We are interested in studying this problem from a rigorous experimental standpoint. We want to examine the kinds of unforeseen effects 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.

## 3 Robot Hardware

The robots are constructed out of LEGO Technic blocks. LEGOs were used because they 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. Each robot is equipped with an articulated cargo bay that is capable of securely grasping a target. For obstacle avoidance, a set of bumpers are located just beyond the front of the robots' treads as well as on the back. The robots and the targets are shown in Figure 1.

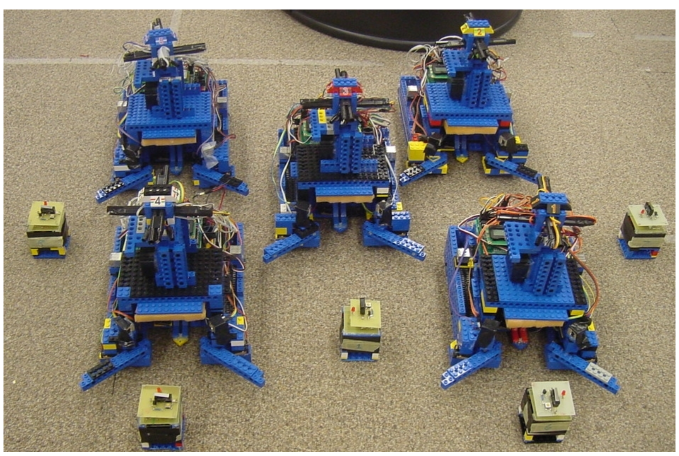


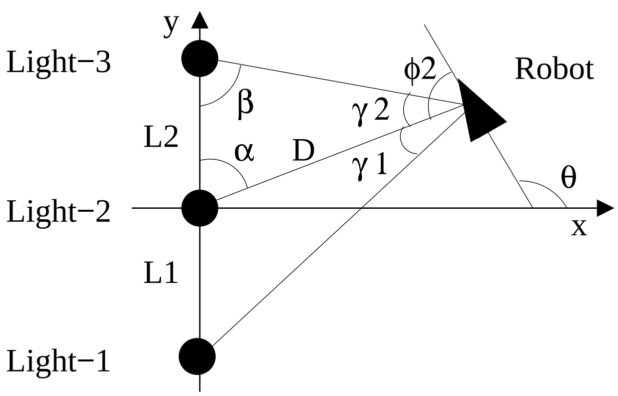
Figure 1: The robots and targets

The targets that the robots attempt to locate transmit an omnidirectional stream of 40 KHz infrared light that is detectable at a range of 70 cm. Two infrared detectors are mounted on each side of the robot and two more are mounted on the front. A turret-mounted set of cadmium-sulfide (CdS) photoresistors is used to track visible-light landmarks. The on-board computer is the Handyboard, an MC68HC11-based microcontroller with 32K of RAM [8]. The software was developed in Interactive-C [11], a subset of C with multitasking capabilities.

## 4 Robot Software

Several parallel sensory-motor behavior processes, similar to the subsumption algorithm[5], are used to control the robot's behavior. Each process is responsible for handling one segment of the robot's control code by mapping sensors to actuators. When the sensor(s) monitored by a process are activated (e.g. when collision detection is activated by a depressed bumper), the process tries to control the actuators. Conflicts between processes running in parallel are resolved with assigned priorities.

Localization and navigation is achieved with three collinear lightbulbs that serve as both home bases and as landmarks. When a robot randomly searches for targets, it navigates by moving away then towards the light at random time intervals. For purposeful search, the robot uses localization to navigate towards a known target location. To establish a target location, the localization routine is invoked when a robot encounters a target while returning another to home base. Once the target is dropped off, the robot navigates back to where the other target was found by invoking the localization routine every 20-30 seconds and using it to correct its heading.




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

Figure 2: Landmark localization used by the robot. The lines connecting the robot and lights represent line of sight. The origin of the coordinate system is positioned at Light-2. To avoid symmetry problems, the robot moves only in the positive X direction. The values of $a, b, c$ & $d$ are computed as shown on the right.

Given some assumptions about the placement of the lights in the environment, the global pose (x,y,$\theta$) of the robot can be determined. The values of $L_1$ and $L_2$ are programmed into the robot *a priori* and are assumed never to change. The robot uses its light-tracking turret to measure the angles to the three lights with respect to its own orientation ($\phi_1$, $\phi_2$, and $\phi_3$)[1], thus $\gamma_1 = (\phi_1 - \phi_2)$ and $\gamma_2 = (\phi_2 - \phi_3)$. The angles $\alpha$ and $\beta$ and the distance to the center landmark $D$ are solved for and from these values, the robot's global pose $(x, y, \theta)$ can be calculated. The robot's orientation $\theta$ is measured with respect to the global $x$ axis. Figure 2 illustrates this analytical solution. The localization method estimates the robot's position to within 25 cm and its orientation to within 5 degrees. However, if it is too close to a light, localization will fail.

## 5 Experimental Description

Many factors determine the effectiveness of a cooperative multi-robotic solution to a search and retrieval task. Three such factors include the physical distribution of the targets, the kinds of search strategies employed by the robots, and the number of robots used. The purpose of this work is to study how the overall performance of a robotic team is affected by altering these factors.

To solve this task, the robots started from a fixed location, searched an area for targets and returned them to one of three drop-off zones. Experiments were run with

[1] For the sake of clarity, only $\phi_2$ is shown in the figure.

one-, two-, and four-robot configurations. The robots were not explicitly aware of each other's presence and simply treated each other as obstacles if they collided. Target locations were either distributed uniformly or nonuniformly (i.e. all placed in one far corner of the arena). Some experiments were run using localization while others were not. Without the ability to localize, a robot's search for targets was random. Figure 3 describes the experimental setup.

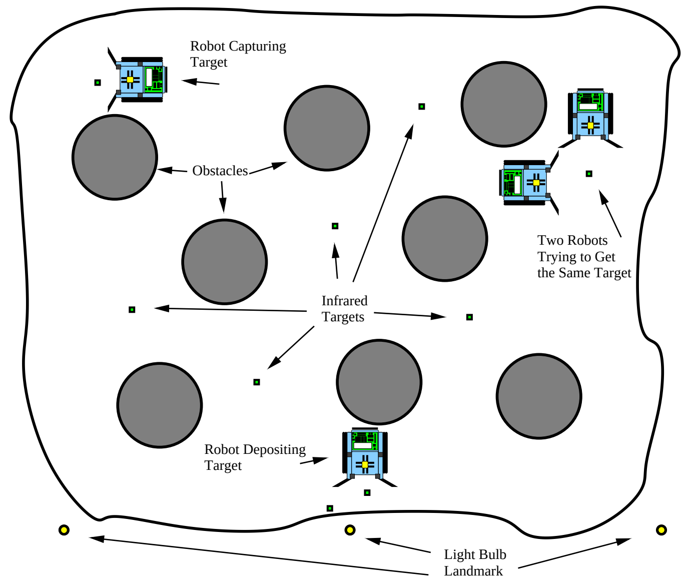



Figure 3: The experimental environment was roughly 5.4 meters on a side. All experiments contained nine targets and eight obstacles. Three lightbulbs were placed at known locations and were used to determine position and orientation. Obstacles are relatively low and do not block the robot's view of the landmarks. *Robots and targets are not shown to scale.*

## 6 Experimental Results

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. Each experiment was run until all nine targets were retrieved. Table 1 shows the average time in seconds that it took to retrieve all targets. Each graph in Figure 4 shows the average time to retrieve each target when using one-, two- and four-robot teams. Results across columns differ by target distribution where left is nonuniform and right is uniform. Rows differ by use of localization. Results were obtained from experiments where no localization was used (top row), where localization was used (middle row), and where localization was used but the processing time was factored out (bottom row), making localization instantaneous.

| | uniform 1 robot | uniform 2 robots | uniform 4 robots | nonuniform 1 robot | nonuniform 2 robots | nonuniform 4 robots |
|---|---|---|---|---|---|---|
| no localize | 934 | 458 | 374 | 1672 | 1058 | 587 |
| localize | 1108 | 478 | 343 | 1911 | 1030 | 593 |
| instant localize | 986 | 478 | 323 | 1328 | ***794** | ***444** |

Table 1: Average time in seconds when the last target was retrieved. Star (*) indicates statistically significant difference at the 95% confidence level between *instant localize* and *no localize* results of the same column.

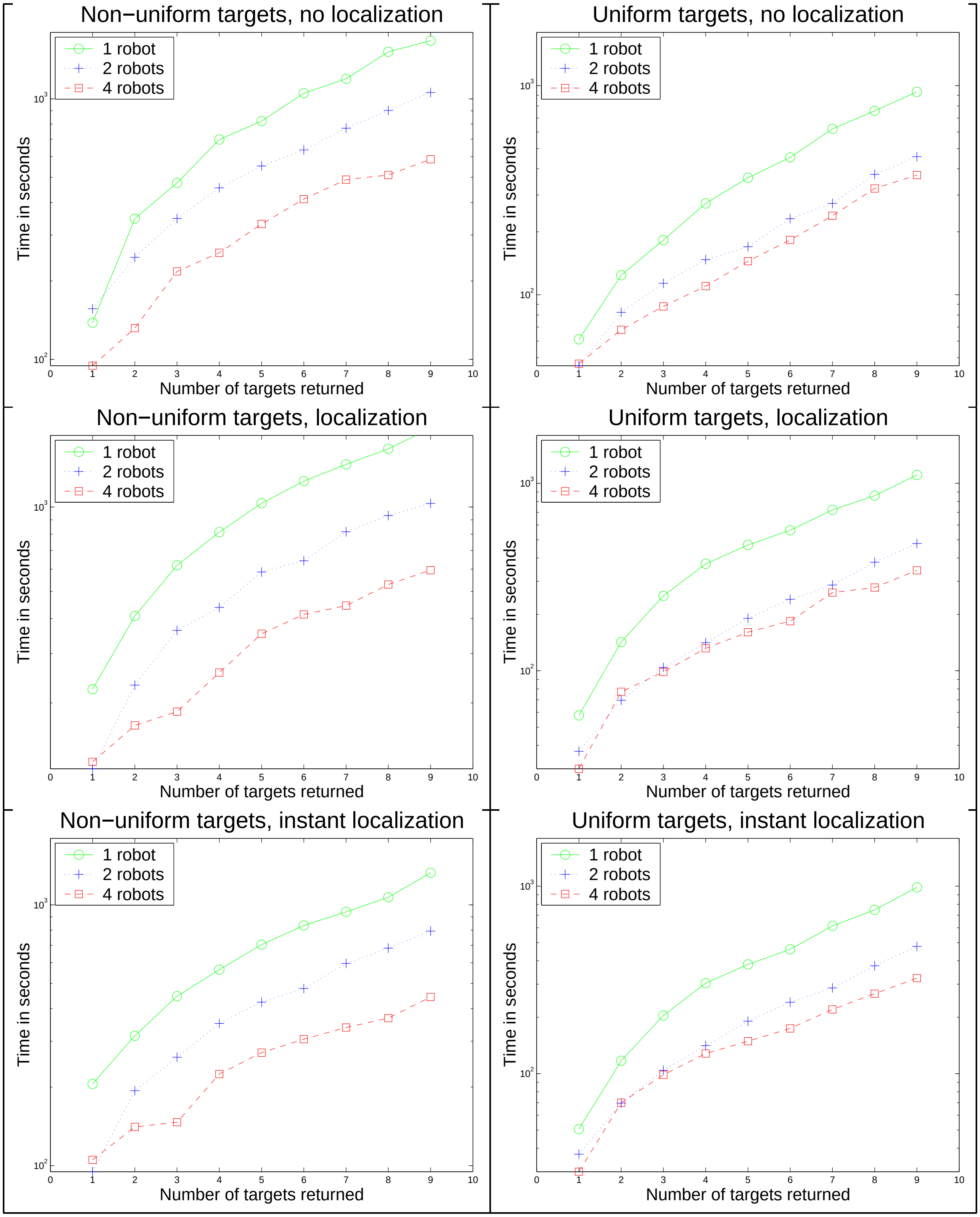



Figure 4: Experimental results for the search and retrieval task. Each chart shows the average time that it took to find each target for one, two and four robots. Experimental variables are the distribution of targets in the environment (from uniform to nonuniform), and purposeful versus random search. The time axis is shown in a base 10 logarithmic scale.

Each time a robot localizes, it must remain stationary for 18 seconds while it collects and processes the landmark data[2]. This 18-second delay had a significant effect on the overall time to complete the task, as reflected in the table. There are two reasons for factoring out localization overhead. 1) It indicates potential payoff for improvement of the localization technique, and 2) it can help determine how much overhead the system can afford while still improving task performance. In practice, instantaneous localization would be difficult to achieve but it is reasonable to assume that the 18 seconds could be significantly reduced. Varying the time it takes to localize can provide a maximum time that localization can take while still improving performance.

In looking at how localization affects performance, it can be seen from inspection of Figure 4 and Table 1 that there was no improvement when retrieving uniformly distributed targets. During the experiments, when returning a target to base, robots rarely encountered other targets, thus the ability to localize (i.e. storing the location of a found target) offered no advantage. In the experiments with nonuniformly distributed targets, there was a qualified performance improvement. Robots almost always encountered other targets when returning to base, thus a robot with localization capability could navigate directly towards the cache of targets rather than wander randomly. The computational overhead of our implementation of localization outweighed this benefit of purposeful search, but by factoring the overhead out (as described above), we can see how performance can improve.

$T$ tests were run to determine the significance of the non-localization versus localization trials and the non-localization versus instant localization trials. 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.

Additional analysis was conducted on the target search time of each robot. Table 2 illustrates the average time that it took each robot to grab a new target after returning a captured one to the base. Once again, the localization and instant localization results were compared against the no localization results for statistical significance. For this data, all three of the 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.) All other localization results (instant or otherwise) were not statistically significant from the corresponding non-localization results.

| | uniform 1 robot | uniform 2 robots | uniform 4 robots | nonuniform 1 robot | nonuniform 2 robots | nonuniform 4 robots |
|---|---|---|---|---|---|---|
| no localize | 83 | 57 | 64 | 150 | 181 | 142 |
| localize | 96 | 65 | 79 | 131 | 143 | 152 |
| instant localize | 89 | 65 | 72 | ***88** | ***100** | ***94** |

Table 2: Average times in seconds for a robot to grab a new target right after a captured one has been dropped off. These values are calculated by the number of targets actually returned during the run. The differences between the starred instant localize results and the no localize results in the same column are statistically significant at the 95% confidence level.

It should be noted that the use of localization in an environment with uniformly

[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.

distributed targets actually degraded performance. This is attributed both to localization errors that would cause the robot to head off in the wrong direction, and in the multi-robot case, to navigating towards a target that was no longer present because another robot had picked it up.

To evaluate how team size affects performance, we calculated linear speed-up for each of the two- and four-robot teams. Results are shown in Table 3. We defined speed-up $S_n$ as:

$$S_n = \frac{t_1/n}{t_n}$$

where $n$ is the number of robots and $t_n$ is the time it takes $n$ robots to retrieve all targets. This is similar to a speed-up measure found in [3]. The system is said to have linear speed-up if $S_n = 1$, superlinear if $S_n > 1$, and sublinear if $S_n < 1$. As was expected, speed-up of two-robot teams was larger than the speed-up of four-robot teams.

| | uniform 2 robots | uniform 4 robots | nonuniform 2 robots | nonuniform 4 robots |
|---|---|---|---|---|
| no localize | 1.02 | 0.62 | 0.79 | 0.71 |
| localize | 1.16 | 0.81 | 0.92 | 0.81 |
| instant localize | 1.03 | 0.76 | 0.83 | 0.75 |

Table 3: Linear speed-up for two- and four-robot teams. Values of 1 indicate linear speed-up, less than 1 indicate sublinear.

## 7 Conclusions and Future Work

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 nonuniformly distributed environment and not so much in the uniformly-distributed environment. This turned out to not necessarily be the case due to the somewhat lengthy overhead (18 seconds) involved in localizing. We did note that if we discounted the time to localize, the robots were much faster at finding their way back to a new target once one had been dropped off. More work needs to go into the search strategies to reduce the overhead. 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 shown true in our speed-up analysis. These results show that knowledge about the structure of the environment is very important when choosing a search strategy for a team of robots.

Future work will include optimizing the localization system so that it runs faster and is more accurate. This will help increase the amount of time that robots contribute to the completion of the task. Another variation to consider is a maze-like environment where the targets would be enclosed inside of small alcoves. In this case, explicit localization is expected to be extremely important. Path planning may also prove to be beneficial, if not essential, in this kind of environment. Finally, the effects of communication between the robots will be explored. Experiments will be run to analyze different kinds of communication systems and determine how much information should be shared between the robots so they can complete their task.

## 8 Acknowledgments

The work of Amy Larson was supported by the Doctoral Dissertation and Louise T. Dosdall Fellowships of the University of Minnesota. Sarah Osentoski, from the University of Nebraska, Lincoln, was supported by the Computing Research Association (CRA) Distributed Mentor Project. Additional partial support was provided by the Defense Advanced Research Projects Agency, Microsystems Technology Office, (Distributed Robotics) ARPA Order No. G155, Program Code No. 8H20, issued by DARPA/CMD under Contract #MDA972-98-C-0008.

We would like to thank Daniel Boley for his insights into the localization algorithm and the help that he gave us in computing a closed-form solution. We would also like to thank Joseph Djugash, Ashutosh Jaiswal, Esra Kadioglu, Elaine B. Rybski, Lorry Strother, and Harini Veeraraghavan for their help in collecting data.

## References

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

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

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

[4] 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.

[5] 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.

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

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

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

[9] Maja J Mataric. Using communication to reduce locality in distributed multi-agent learning. In *Proc. Nat'l Conf. on Artificial Intelligence*, July 1997.

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

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