An Adaptive Communication Protocol for Cooperating Mobile Robots

Holly Yanco and **Lynn Andrea Stein***
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
545 Technology Square
Cambridge, MA 02139
email: holly@ai.mit.edu, las@ai.mit.edu

Abstract

We describe mobile robots engaged in a cooperative task that requires communication. The robots are initially given a fixed but uninterpreted vocabulary for communication. In attempting to perform their task, the robots learn a private communication language. Different meanings for vocabulary elements are learned in different runs of the experiment. As circumstances change, the robots adapt their language to allow continued success at their task.

1 Introduction

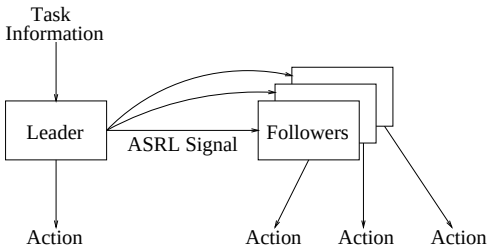
In this paper, we investigate the evolution of simple communication protocols among nonverbal subjects engaged in cooperative tasks. Gregarious animals, small children, and even adult humans lacking common language engage in such activity routinely. Grunts, gestures, and other nonverbal signals take on mutually agreed-upon meanings in the context of cooperative tasks. "Follow me," "Look out!" and "Raise your end of the table higher" can all be conveyed without previously agreed-upon language. Satisfactory completion of cooperative tasks such as table-carrying, hunting, or tribal survival often depends on making effective use of such communications.

This paper describes an example of a cooperative task—coordinated movement—achieved by a troupe of mobile robots. Depending on circumstances, different actions are required. One robot, in the role of the leader, has access to this information and learns to act appropriately. In addition, the leader must communicate to the troupe what actions are required on their parts. The communication language is not fixed at the outset; instead the troupe must learn to associate appropriate actions with the commands issued by the leader. As circumstances evolve, the meanings of the leader's commands may change; the troupe must adjust their actions accordingly. The goal is for the entire troupe to act appropriately and to adapt those actions and the underlying language over time.

When a task requires cooperation, there is often a need for some form of communication between the participating agents. Cooperative work requires communication whenever one agent's actions depend critically on knowledge that is accessible only to another agent. It is often an expedient even when one agent can accomplish the task on its own or when all agents have access to the requisite information. Previous work on cooperative behavior among mobile robots has largely assumed a fixed communication language. (See, for example, [Fukuda and Kawauchi, 1990], [Matsumoto *et al.*, 1990], or [Shin and Epstein, 1985].) However, a language created for the robots may not provide the optimal solution. The language itself may not be natural either to the robots or to the task at hand. In addition, in a changing world, a hard coded language may make it difficult for the agents to adapt to novel situations. Fixed communication languages are less able to handle circumstances in which changing environments dictate changing communications, just as agents that cannot adapt to new environments are at an evolutionary disadvantage relative to those that are able to learn.

The research described in this paper is aimed towards giving autonomous agents the ability to develop their own language. Our initial work was inspired by that of [Shewchuk, 1991]. His Ph.D. thesis addresses the design of appropriate reinforcement learning algorithms to learn languages for internal repre-

*This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124. The first author is supported in part by a scholarship from Digital Equipment Corporation. The second author is supported in part by the Class of 1957 Career Development Chair and in part by a General Electric Junior Faculty Career Award. Support for this research was also provided by the Gordon S. Brown Fund of the Department of Electrical Engineering and Computer Science at MIT.




```
Leader:

   loop:  on valid input signal from environment
             choose an action to perform
             choose a signal to send to follower via
                the radio boards
          wait for reinforcement signal
          on reinforcement signal
             increment variables for action and signal
          goto loop

Follower:

   loop:  on valid input signal from leader
             choose an action to perform
          wait for reinforcement signal
          on reinforcement signal
             increment variables for action
          goto loop
```

Figure 1: Information flow in the coordinated movement task.

sentation as well as for communication. He has implemented a simple simulation of a language learning task similar to the basic experiment we describe below (two robots, two language elements) as a part of his symbolic test suite for reinforcement learning algorithms. Work on the development of communication between groups of autonomous agents has also been done by [MacLennan, 1990] and [Werner and Dyer, 1990]. Their research addresses the problem of language learning with genetic algorithms. Language evolves over many generations of the community. Within an individual agent, however, language is fixed over its lifetime. In all of these cases, implementation is limited to simulation; only the work of Shewchuk addresses the problem of task-based reinforcement (see below).

2 The task

The cooperative task that we have chosen is coordinated movement. Groups of animals engage in such activity when hunting, escaping predators, herding, flocking, migrating, or supervising their young. Environmental cues inform the leader (or leaders) of the troupe as to appropriate troupe movements; a successful leader learns to interpret these cues. Further, the leader learns to communicate to the rest of the troupe the tasks that they are to execute. For example, upon hearing predatory noises, a mother may shepherd her young in the opposite direction or encourage them to remain motionless to avoid detection. The followers may be unaware of or unable to interpret the environmental cues.

We have implemented this task with troupes with two and three members and with a variable number of vocabulary elements on robots and in simulation. A troupe of two robots made up of one leader, Bert, and one follower, Ernie, was used for the robot experiments. The simulator was used to gather data for the three agent experiments and for the larger vocabulary experiments with two agents. Because we are particularly interested in the development of language, we assume that the followers do not have access to the task specification (i.e. the environmental cues) and must rely completely on the communication signals emitted by the leader. In future experiments, we expect to allow the follower robot(s) to use some environmental input to modulate the communication signals from the troupe leader.

Since this is a cooperative task, successful performance depends on the actions of the troupe as a whole. Analogously, the mother animal succeeds in hiding her young only if all remain motionless; one disobedient cub can give away the hiding place and cause the whole troupe to be eaten. Environmental reinforcement is therefore positive only if all agents perform the appropriate actions. Since the followers cannot correctly interpret the environmental cues, this performance can be achieved reliably only when the leader and follower robots mutually agree on the development and interpretation of a private communication protocol.

Thus, the learning tasks are as follows:

- For the leader robot, the interpretation of the environmentally supplied signal, the execution of an appropriate action, and the transmission of an appropriate signal to the follower robot.
- For the follower robots, the execution of an appropriate action based on the signal received from the leader robot.

The "appropriateness" of an action is determined by the environmentally supplied signal. The "appropriateness" of the leader robot's signal, however, is constrained not by the environment but by the leader and follower robot's adapted internal state. That is, the signal is appropriate if and only if the follower robot takes the

(environmentally constrained) appropriate action when that signal is received. (The algorithm is summarized in figure 1.)

3 The robots

Bert and Ernie, the two robots used in this research, are Sensor Robots designed by Fred Martin at the Media Laboratory at the Massachusetts Institute of Technology [Martin and Sargent, 1991]. Each robot is approximately $9''l \times 6''w \times 4''h$, with a single circuit board containing most of the computational and sensory resources of the robot. A 6v battery strapped to the underside of the chassis supplies the power for the robot. The robots are shown in figure 2.

The primary computational resource is an onboard Motorola 6811 microprocessor. The programming environment is IC, a multi-tasking interactive C compiler and interpreter developed by Randy Sargent [Sargent and Martin, 1991]. IC allows a Sensor Robot to be addressed through a serial line from a host computer as well as the downloading of programs for autonomous activity. The work described in this paper was implemented with the robots under autonomous control.

Locomotion is controlled by a dual geared-wheel drive stripped from a Radio Shack Red Fox Racer. The direction of the robot is controlled by varying the speeds of the left and right motors (with negative speed moving the motor backwards). The two motorized wheels are at the rear of the robot chassis and a caster is on the front.

Communication from human to Bert is through an infra-red remote control transmitter. Bert uses infra-red receivers similar to those found in televisions and VCRs. (While Ernie also has infra-red receivers on board, they are not being used in this work – only Bert hears signals from humans.) The robots communicate between themselves using a pair of radio transmitter and receiver boards similar to those used in garage door openers. (The transmitter and receiver each run off of a separate 9v battery.) Additionally, each robot has a speaker and a 16-character LCD, both used primarily for debugging and monitoring of the robot's activity.

In addition to the infra-red and radio receivers, the sensor robots contain four (front and rear, left and right) bump sensors, left and right shaft encoders, an inclination sensor, photosensitive cells, a microphone, and infra-red emitters. These additional sensory abilities of the robots were not substantively used in the experiments described here.

4 The implementation

In our experiments, the environment is represented by a human "instructor" who issues one of a number of signals to indicate the desired action. Currently, the number of signals is equal to the size of the language. The leader robot performs an action and also signals the follower robot. Upon receipt of the leader's signal, the follower robot selects and performs an action. If both robots have performed correctly, positive reinforcement (+) is issued. If either robot performs incorrectly, negative reinforcement (−) is issued. Based on this environmental feedback, the robots learn to select appropriate actions and communication signals. This algorithm is summarized in figure 1.

Both the action selection and the signal selection are learned using standard reinforcement learning techniques. (See, e.g., [Kaelbling, 1990] or [Sutton, 1992] for overviews of reinforcement learning.) The particular algorithm that we use is adapted from Kaelbling's interval estimation method [1990]. Interval estimation is a relatively simple form of reinforcement: A table of inputs × actions is maintained. Each time an input is received, the expected "best" action is taken and the counter for that input/action pair is incremented. If positive reinforcement is received, a second counter for that input/action pair is also incremented. The "best" action given some input is selected by an optimization function. If no one particular action is the "best", an action is selected randomly. (The algorithm for interval estimation is given in figure 3.)

In our initial experiments, we allow each of the robots two possible actions. At each iteration, each robot chooses either *go straight* or *spin*. Further, the communication protocol contains only two vocabulary elements—

The initial state, s_0, consists of the integer variables x_0, n_0, x_1, and n_1, each initialized to 0.

```
u(s,a,r) = if a = 0 then begin
                x0 := x0 + r
                n0 := n0 + 1
           end else begin
                x1 := x1 + r
                n1 := n1 + 1
           end

e(s) =     if ub(x0,n0) > ub(x1,n1) then
                return 0
           else
                return 1
```

where

$$ub(x,n) = \frac{\frac{x}{n} + \frac{z^2_{\alpha/2}}{2n} + \frac{z_{\alpha/2}}{\sqrt{n}}\sqrt{(\frac{x}{n})(1-\frac{x}{n}) + \frac{z^2_{\alpha/2}}{4n}}}{1 + \frac{z^2_{\alpha/2}}{2n}}$$

and $z_{\alpha/2} > 0$.

Figure 3: Kaelbling's interval estimation algorithm [Kaelbling, 1990, Figure 21].

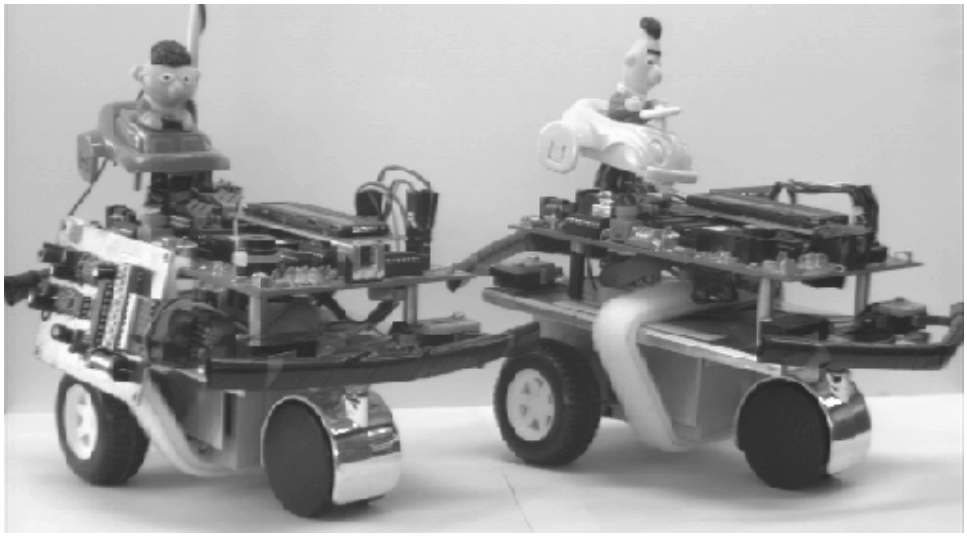


Figure 2: Ernie and Bert

high and low—so that the learning problem remains tractable. The leader robot must thus learn to select one of four possible action/communication pairs; the follower robot must learn to associate each of the vocabulary items with one of its two possible actions. Convergence on the robots is easily verified by testing each environmental input; if all behaviors are as expected, the protocol will not change further without environmental adaptation.

We have also implemented the identical learning algorithms in simulation in C running under UNIX on SUN SPARCstations. The simulation is used primarily for ease of data collection over large numbers of runs and in scaling up the experiments to larger language and troupe sizes. To assess convergence, we wait until all of the instructor's signals (i.e. all distinct environmental cues) have been completed three times consecutively without negative reinforcement. At that point, each input is tested to verify that convergence has truly been reached. (This leads to slightly inflated convergence times in simulation over experiments on the robots.)

In the case of a two-element language, convergence times vary wildly when a true random function is used. Extended series of a single environmental cue cause oscillations in the agents' state variables, leading to protracted convergence times in a substantial fraction of the runs. (The simulation took an average of 12006.25 iterations to converge for a two element language using two agents, with a minimum of 10 iterations and a maximum of 235109 iterations. In 100 tests of this case, the simulation took over 100 iterations to converge in one quarter of the tests.) To avoid this problem, we used a biased random function that prevented extended series of similar environmental inputs. The data given in the results section of this paper was collected using the biased random function. Data was also collected with an unbiased random function; in all but the two-element case, the results are comparable.

In the implementation on our robots, communication noise is dwarfed by human error and by the complexity of the task-based learning problem; we therefore omit it from our simulation.

5 Task-based reinforcement.

Our experiment is unusual in that reinforcement—positive environmental feedback—is received only when both robots succeed in performing the appropriate actions. This sort of *task-based* reinforcement contrasts with the usual individually based reinforcement typical in the literature. Because robots receive reinforcement only when the troupe as a whole performs the task correctly, it is possible for an individual to perform correctly but receive negative feedback. In addition, none of the robots can sense the action taken by the others; thus, the environmental reinforcement cannot be interpreted in that light. This further complicates the reinforcement learning task.

Reinforcement learning algorithms typically generate action policies for individual agents given some environmental constraints. The adaptation that we describe here is no exception. The leader robot, for example, learns policies for what action to take and what signal to send the follower on a given input signal. However, what is reinforced—what the environment rewards—is not the policy of the individual robot but the successful performance of the total cooperative task.

For example, if the environmental constraints make *both spin* the appropriate course of action, the leader robot may spin and send the follower robot the signal that the leader believes to mean *spin*. If the fol-

	Appropriate action	Leader's action	Leader's signal	Follower's action	Reinforcement
1.	↑↑	*spin*	low	*spin*	−
2.	◯◯	*spin*	low	*straight*	−
3.	↑↑	*straight*	high	*spin*	−
4.	◯◯	*straight*	high	*straight*	−
5.	◯◯	*spin*	low	*spin*	+
6.	↑↑	*straight*	high	*spin*	−
7.	◯◯	*spin*	low	*spin*	+
8.	◯◯	*spin*	low	*spin*	+
9.	◯◯	*spin*	low	*spin*	+
10.	↑↑	*spin*	low	*spin*	−
11.	↑↑	*straight*	high	*straight*	+
12.	↑↑	*straight*	high	*straight*	+
13.	◯◯	*spin*	low	*spin*	+

Table 1: A sample run. The desired behavior is *both spin* on input ◯◯, *both go straight* on input ↑↑. After thirteen iterations, convergence is reached.

lower has not yet learned the communication protocol, it may incorrectly interpret the leader's signal to mean *go straight*. In this case, the leader has performed correctly—both its action and its signal were appropriate—but receives negative reinforcement. Similarly, if the leader issues an inappropriate signal—*go straight*—but the follower interprets that signal to mean *spin*, the environment provides positive reinforcement (for a correctly executed task) in spite of the incorrect internal communication. Further, neither robot is aware of the action taken by the other and so cannot use that knowledge to assess the environmental feedback.

The choice of this form of *task-based reinforcement* is motivated by biological analogy. The world does not generally reinforce inappropriate action, no matter how well meant. Similarly, fortuitously apt actions are rewarded even though the underlying intention is flawed. (This is the root of serendipity and the sentiment underlying the saw, "Necessity is the mother of invention.") Because the robots' communication protocol is private and not interpreted by the external environment (or human "instructor"), it is not relevant to the reinforcement received. If the robots can succeed in taking appropriate actions in spite of miscommunication, they continue to receive positive feedback. In principle and in practice, the task is designed so that only successful learning of the private language allows continued successful execution of the cooperative task.

6 Results

6.1 Developing a Shared Language

Bert and Ernie are able to learn both synchronous action—both performing the same action in the same interval—and divergent action—e.g., leader *spins*, follower *goes straight*. Convergence times typically range from five to twenty iterations. A sample run of the experiment is given in table 1. In this run, the appropriate actions are for both robots to *spin* on input ◯◯ and for both robots to *go straight* on input ↑↑. The robots converge on a mutually agreeable language—a low signal means that the follower should *spin*, while a high signal means to *go straight*—after thirteen iterations.

6.2 Adaptability of language

Once the robots converge on a particular dialect, they continue to receive positive reinforcement as long as the environmental constraints do not change. If circumstances change, however, the robots may find that their previously successful actions no longer earn them positive feedback. For example, after the run in figure 1, we might change the "appropriateness" of the robots' actions by giving positive reinforcement to *leader spin, follower go straight* on ↑↑. Under such circumstances, the robots can adapt their behavior—and, when necessary, their communication protocol—to the changing environment. Convergence times for the new task (i.e., to unlearn portions of the old task and relearn the newly appropriate behavior) range from roughly comparable to the initial learning task to roughly double the time, depending on the difficulty of the new task, the differences between the old and the new, and how firmly the previous behavior is entrenched.

6.3 Scaling Up the Language

The simulator has been used to collect statistics on languages ranging in size from two to twenty elements using troupes of two and three agents. For each language

Size of Language	Number of Iterations to Convergence Average	Minimum	Maximum
2	15.34	10	24
3	110.30	33	501
4	340.38	53	990
5	906.62	255	2472
10	15011.61	2868	51031
20	232267.82	44196	1241767

Table 2: Learning times for a two member troupe. Experiments for each language size were run 100 times.

Size of Language	Number of Iterations to Convergence Average	Minimum	Maximum
2	27.21	10	80
3	327.71	35	1211
4	1530.12	340	6666
5	4415.60	652	17533
10	163530.62	37130	705029

Table 3: Data above is for a three member troupe and was collected over 100 runs for each language size.

size, 100 experiments were run to convergence. Mean, minimum, and maximum convergence times (number of iterations) are given in table 2 and table 3. Time to learn a language grows exponentially in the number of vocabulary elements in the language.

6.4 Variation over Dialects

In order to achieve convergence on any of these tasks, a particular language must be agreed upon. However, the language varies from run to run. For example, in the run in figure 1, the robots agree that a low signal means that the follower should *spin*. In another run of the same experiment (◯◯ means *both spin*, ↑↑ means *both go straight*), a high signal may be interpreted as *spin* and a low signal as *go straight*. Such dialect differences depend partly on the random selection of vocabulary elements; however, it is critical to the success of the task that the leader and the followers all agree on the same dialect.

Using the simulator, we have counted the numbers of times that particular dialects appear over 100 runs of a particular experiment. In table 4, data is given for the two element language using both two and three agents; table 5 shows data for the three elements language using two and three agents. These percentages are variable and we expect that the agents' selection between dialects would be evenly distributed in a large number of runs.

6.5 Increasing Troupe Size

When the troupe size is increased to three, the amount of time necessary to converge upon a language increases

Dialect	Two robots	Three agents
low = spin high = straight	45	52
low = straight high = spin	55	48

Table 4: The agents can agree on differing dialects in each run. Totals for each dialect are based on 100 runs of the two element language experiment.

Dialect	Two robots	Three agents
low = right med = straight high = left	19	14
low = right med = left high = straight	14	20
low = straight med = right high = left	16	15
low = straight med = left high = right	9	14
low = left med = right high = straight	13	17
low = left med = straight high = right	29	20

Table 5: In 100 runs, agents develop all of the six possible dialects in a three element language. The number of times each dialect was developed is given.

due to the additional learning that is required. (See table 3.) For the twenty element language using three agents, the agents required 12,105,480 iterations to agree upon a dialect. (The experiment was only run once since it required more than 24 hours of processing time on a SUN Sparc2.)

The learning time might be reduced by having a subset of the agents (in this case, two of the agents) agree on a dialect, then have these robots bootstrap the other members of the troupe. This is similar to the way humans learn language; every person who speaks a particular language does not need to relearn the language when other people are learning it. This learning method will be explored in future work.

7 Scaling up

As the number of possible actions and signals increase, the time necessary for convergence increases exponentially. This is due to the reinforcement learning algo-

rithm. For every signal, a table needs to maintained that contains each of the actions that could be performed. In the experiments, we are usually looking for a one–to–one correspondence between signals and actions; however, this fact is not exploited to speed up the learning process since it would make the algorithm too specific.

If we rely on learning a new signal for every action, the learning task quickly becomes intractable. Currently, we are exploring alternate directions for language development.

The language that we have described here maps single vocabulary items onto single actions. Human language gains power by virtue of its compositional nature. That is, because vocabulary elements can be combined into sentences, a vocabulary of fixed size can generate an exponential number of utterances. Further, the interpretation of the sentence depends on the interpretation of the vocabulary elements in isolation, allowing a word learned in one context to be correctly understood in a novel utterance. The next phase of our experimentation will address the task of learning a compositional language.

For example, the robots might have a language with one set of signals for direction of motion and another set for speed. The leader's communications might then be utterances such as "go left slowly" or "spin quickly." Because the learning problem is per word rather than per utterance, the complete language can be approximated by a sublinear number of vocabulary elements. Even allowing for the exponential blowup in space and time of current reinforcement learning methods, the problem remains in the realm of computational feasibility.

8 Discussion

Implications of task-based reinforcement. Task-based reinforcement poses particular challenges for reinforcement learning algorithms. The robots are learning individual action policy but receiving reinforcement based on the global performance of the cooperative task. As a result, task-based reinforcement behaves somewhat like noise in the reinforcement signal. Because most reinforcement learning algorithms are designed to function in the presence of some amount of noise, they are adequate for this situation. However, task-based reinforcement is not random noise, and some algorithms will be better suited to the job than others. Experiments such as ours provide a useful testbed for learning algorithms. [Shewchuk, 1991] describes efforts to design reinforcement learning algorithms more suited to this sort of problem.

Complex tasks. The selected task—coordinated movement—is one in which reinforcement is received at every iteration. In more complex tasks, reinforcement may be received only after completion of a sequence of actions. This *delayed reinforcement* complicates the learning problem and necessitates the use of more sophisticated learning techniques, such as Sutton's temporal differencing (TD) methods [1988]. By replacing the interval estimation algorithm that we have used with an appropriate variant on TD, it should be possible to extend our adaptive communication protocol to tasks in which the cooperative task requires several sequential steps. This will of course slow down the learning phase considerably.

Taking the human out of the loop. One shortcoming of the current work is that a human "instructor" is currently needed to observe the robot's behavior and provide positive or negative reinforcement. In a more natural task, the environment itself should be able to provide that feedback. [Maes and Brooks, 1990] describe such an experiment with an individual legged robot that self-reinforces to learn a balanced gait. Because our robots are somewhat fragile and because their effective ability is largely limited to wheeled locomotion, we have not yet attempted such autonomic reinforcement. The design of a cooperative task that does not pose undue hazard to the hardware but allows for repeated experimentation and miscommunication remains an open challenge.

Acknowledgements

The work described in this paper was inspired by a simulation experiment described by John Shewchuk. The hardware and software that we used were originally developed for the 6.270 course and 6.915 Robot Design Seminar at MIT. The course staff and students of that class spent innumerable hours working on what ultimately became the substrate for our research. We are particularly indebted to Fred Martin, who masterminded the Sensor Robots; to Randy Sargent, who designed and implemented the IC programming language; to Philip Alvelda, who built the prototype radio system; to Tim Tang, who helped to build and debug Bert and Ernie, and who implemented their radio communications; to Anne Wright, who implemented the IR communications; and to Scott Willcox, who helped debug the radio boards in a pinch. Last, but certainly not least, Ian Horswill, Jeanne Speckman, Nancy Pollard, Mark Torrance, and Rod Brooks listened to our ideas and offered support and encouragement for our work. We thank them.

References

[Fukuda and Kawauchi, 1990] T. Fukuda and Y. Kawauchi. Communication and distributed intelligence for cellular robotic system CEBOT. In *1990 Japan-USA Symposium on Flexible Automation*, pages 1085–1092, July 1990.

[Kaelbling, 1990] Leslie Pack Kaelbling. Learning in embedded systems. Technical Report TR–90–04, Teleos Research, Palo Alto, California, June 1990.

[MacLennan, 1990] Bruce MacLennan. Evolution of communication in a population of simple machines. Technical Report CS–90–99, University of Tennessee, Knoxville, Tennessee, January 1990.

[Maes and Brooks, 1990] Pattie Maes and Rodney A. Brooks. Learning to coordinate behaviors. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, pages 796–802, Boston, Massachusetts, July 1990. MIT Press.

[Martin and Sargent, 1991] Fred Martin and Randy Sargent. The MIT sensor robot: User's guide and technical reference. October 1991.

[Matsumoto *et al.*, 1990] A. Matsumoto, H. Asama, and Y. Ishida. Communication in the autonomous and decentralized robot system ACTRESS. In *Proceedings of the IEEE International Workshop on Intelligent Robots and Systems*, pages 835–840, Tsuchura, Japan, July 1990.

[Sargent and Martin, 1991] Randy Sargent and Fred Martin. IC: Multi-tasking interactive C for the 6811. IC Version 2.5, October 1991.

[Shewchuk, 1991] John P. Shewchuk. Ph.D. thesis proposal. Department of Computer Science, Brown University, Providence, Rhode Island, 1991.

[Shin and Epstein, 1985] K. Shin and M. Epstein. Communication primitives for a distributed multi-robot system. In *Proceedings of the IEEE Robotics and Automation Conference*, pages 910–917, 1985.

[Sutton, 1988] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[Sutton, 1992] Richard S. Sutton. Special issue on reinforcement learning. *Machine Learning*, 8(3–4), May 1992.

[Werner and Dyer, 1990] Gregory M. Werner and Michael G. Dyer. Evolution of communication in artificial organisms. Technical Report UCLA–AI–90–06, University of California, Los Angles, California, November 1990.