Autonomous Robots, 1, 1–25 (1994)

Communication in Reactive Multiagent Robotic Systems

TUCKER BALCH AND RONALD C. ARKIN

tucker@cc.gatech.edu arkin@cc.gatech.edu

Mobile Robot Laboratory, College of Computing, Georgia Institute of Technology, Atlanta, Georgia 30332-0280 U.S.A.

Received ??. Revised ??.

Abstract. Multiple cooperating robots are able to complete many tasks more quickly and reliably than one robot alone. Communication between the robots can multiply their capabilities and effectiveness, but to what extent? In this research, the importance of communication in robotic societies is investigated through experiments on both simulated and real robots. Performance was measured for three different types of communication for three different tasks. The levels of communication are progressively more complex and potentially more expensive to implement. For some tasks, communication can significantly improve performance, but for others inter-agent communication is apparently unnecessary. In cases where communication helps, the lowest level of communication is almost as effective as the more complex type. The bulk of these results are derived from thousands of simulations run with randomly generated initial conditions. The simulation results help determine appropriate parameters for the reactive control system which was ported for tests on Denning mobile robots.

Keywords: multiagent system, mobile robots, social communication

1. Introduction

Robot system designers must carefully consider each component of their design. The inclusion of sensors, actuators, or additional robots must be justified by contributing to efficient task completion. Components that do not directly contribute add cost without benefit. Communication is another component of multiagent robotic systems that merits careful consideration. The question is not simply whether or not to include inter-robot communication, but what type, speed, complexity and structure. How should these design decisions be made?

As in other disciplines, a formal methodology helps the designer answer these questions. At the Georgia Tech Mobile Robotics Laboratory, such a robot system design methodology has been developed and refined for both single and multiagent robotic systems. These systems are implemented in both simulation and on mobile robots (e.g., [3], [13]). The approach relies on two key points: 1) an objective metric of system performance, and 2) an iterative cycle of simulation and instantiation on real systems. Through simulation, the designer can quickly discover which sensors, actuators, and control parameters are most critical. Parameters are varied as performance is measured and compared to that of other configurations. The goal is to find a system that maximizes (or minimizes) the performance metric. Finally, the configuration is ported to a real robotic system for testing. In this article, the approach is applied to communication in reactive multiagent robotic systems.

To discover how communication impacts multiagent robotic system performance, three societal robot tasks were devised. The performance in simulation of a team of robots is measured for each of these tasks for three different types of communication. The experiments are designed so that performance for each type of communication can be compared across different tasks. In all, a six-dimensional space of task, environment, and control parameters was explored including: task, communication type, number of robots, number of attractors, mass of attractors, and percentage of obstacle coverage. The simulation results were supported by porting the control system to a team of Denning mobile robots.

2. Related Work

2.1. Multiagent Robotic Systems

Multiagent robotic systems constitutes a very active area of research. A large body of literature exists regarding systems ranging in size from two to thousands of robots. Dudek et al [19] provide a taxonomy of these systems classified along the dimensions of group size, reconfigurability, processing ability, and communication range, topology and bandwidth. The research in this article concentrates on relatively small group sizes, typically on the order of two to ten agents. Large scale swarm robotic systems (e.g., [27]) are not considered.

Fukuda was among the first to study multiagent robotic systems in the context of what he refers to as cellular robotics [24]. This pioneering work is mainly concerned with heterogeneous agents. The research reported in this article is for homogeneous societies, where all the agents are functionally equivalent. Recently, researchers at MIT's AI Laboratory [16], [40] have studied aspects of subsumption-based reactive control using robot societies consisting of up to 20 agents. In particular, learning methods have been evaluated (e.g., [45]). Applications of multiagent systems are also being investigated in military environments in both the United States and Europe [43], [34]. Extra-terrestrial planetary exploration has also been proposed as a useful target domain for these societies [41].

Our research focuses on three tasks: foraging, consuming, and grazing. Foraging consists of searching the environment for objects (referred to as attractors) and carrying them back to a central location. Consuming requires the robot to perform work on the attractors in place, rather than carrying them back. Grazing is similar to lawn mowing; the robot or robot team must adequately cover the environment. Of these three, foraging has been the most widely studied to date. Floreano [21] describes nest-based foraging strategies using a neural network architecture. Drogoul and Ferber [20] report results of simulations of foraging robots demonstrating the spontaneous evolution of structure such as chains from extremely simple agents.

A pressing question, and one which the research described in this article addresses, is the role of communication in multiagent robotic systems. Arkin [5] previously reported that successful task-achieving behavior can occur even in the absence of communication between agents. Altenburg and Pavicic created a multi-robot society consisting of a group of small robots conducting a search and retrieve task using either an infra-red or incandescent recruitment signal. The authors reported an approximately 50% improvement in performance for target acquisition using this type of signal. The work as reported in [1] is very preliminary.

Werner and Dyer [52] have studied the evolution of communication in synthetic agents. They have demonstrated that directional mating signals can evolve in these systems given the presence of societal necessity. MacLennan [38] also has studied this problem and has concluded that communication can evolve in a society of simple robotic agents. In his studies, the societies which evolved communication were 84% fitter than those in which communication was suppressed. An order of magnitude better performance was observed when learning was introduced. Franklin and Harmon, in simulation research conducted at ERIM [22], used a rule-based cooperative multiagent system to study the role of communication, cooperation, and inference and how these relationships lead to specialized categories of cooperative systems. Regarding communication, they recognized that information need not be explicitly requested by a receiver in order for it to be potentially useful to the multiagent system as a whole.

Yanco studied communication specifically in the context of robotic systems. In her research [53], a task is defined which requires communication to coordinate two robots, Ernie and Bert. The robots have a limited vocabulary which self-organizes over time to improve the performance of the task, which involves mimicking the behavior of a leader robot. Noreils [44] describes coordinated protocols as a basis for encoding communication signals between robots for navigational tasks. Formal theoretical methods are also being applied in a limited way to this problem. For example, Wang [51] has looked at distributed mutual exclusion techniques for coordinating multi-robot systems.

The research we report in this article is motivated by the desire to create a design methodology for multiagent reactive robotic systems. To effectively design these systems it is important to choose correctly the number of agents and the communication mechanisms of a robot society for a particular task. This goal is decidedly different than the studies reported above.

2.2. Biological Systems

Nature offers a wealth of existing *successful* behaviors which robot designers can often directly apply to their work. Since communication is important in many natural societies it is appropriate to look to them for inspiration. Our strategy for creating multiagent systems has been significantly influenced by biological and ethological studies. In [10], we reported the dimensions by which communication can be described in these systems. A few specific examples of the role of communication in animal societies are reported below.

One of the most commonly studied social biological systems is that of ants. Excellent references on their social organization and communication methods include [29], [25]. Ants typically use chemical communication to convey information between them. Goss et al [26] have studied foraging behavior in ants, creating computer models that are capable of replicating various species' performance for this task. Franks [23] has looked in particular at the behavior of army ants in the context of group retrieval of prey regarding the relationships of mass to objects retrieved and velocity of return.

Tinbergen's influential work on social behavior in animals [50] describes a range of behaviors including: simple social cooperation involving sympathetic induction (doing the same things as others), reciprocal behavior (e.g., feeding activity), and antagonistic behavior; mating behaviors involving persuasion, appeasement, and orientation; family and group life behaviors involving flocking, communal attack (mobs), herding behaviors, and infectious behaviors (alarm, sleep, eating); and fight-related behaviors involving reproductive fighting (spacing rivals), mutual hostility (spacing group individuals), and peck-order (reducing fighting).

An interesting study showing environmental impact on foraging behavior in fish is presented in [18]. The factors considered include food supply, hunger, danger and competition. Mob behavior and communication in the whiptail wallaby [31] also provides an understanding for the emergent organization of multiple agents and the nature of communication that supports this group behavior. Studies in primates have been conducted regarding the organization of colonies [2] relative to their environment. Finally, research in display behavior in animals (e.g., [42]) provides insights in relation to the state-based communication mechanisms described later in this article.

3. Three Tasks for Robotic Societies

The task a robotic system is to perform dictates to some extent the sensors and actuators required. It is not as apparent how the task impacts control system and communication parameters. To investigate this question, three generic multiagent tasks are considered: *Forage*, *Consume*, and *Graze*.

3.1. Forage

The *Forage* task for a robot is to wander about the environment looking for items of interest (attractors). Upon encountering one of these attractors, the robot moves towards it, finally attaching itself. After attachment, the robot returns the object to a specified home base. Many ant species perform the *Forage* task as they gather food. Robots performing this task would potentially be suitable for garbage collection or specimen collection in a hazardous environment.

Figure 1a shows a simulation of two robots foraging for seven attractors and returning them to a home base (the simulation environment is described in Section 6). In the simulation, obstacles are shown as large black circles, attractors are represented as small circles, and the paths of the robots are shown as solid or dashed lines. They leave dashed lines as they wander, and solid lines when they acquire, attach, and return the attractors to home base.

The mass of the attractor item dictates how quickly a robot can carry it. The heavier the attractor, the slower the speed. Several robots cooperating can move the attractor faster, but only up to the maximum speed of an individual robot.

3.2. Consume

Like *Forage*, the *Consume* task involves wandering about the environment to find attractors. Upon encountering an attractor, the robot moves towards it and attaches itself to the object. Unlike the *Forage* task, however, the robot performs work on the object in place after attachment. The time required to do the in-place work is proportional to the mass of the object. It is not necessary for the robot to carry the object back to home base. Applications might include toxic waste cleanup, assembly, or cleaning tasks.

Figure 1b shows a simulation of two robots consuming seven attractors. Note that this task is performed in exactly the same environment as the forage task shown in Figure 1a. The robots leave dashed lines as they wander, and solid lines when they acquire and move to the attractors.

The mass of the attractor item dictates how quickly a robot can consume it. The heavier the attractor, the more time it takes. Several robots cooperating can consume an attractor faster. For this task the rate of consumption is linear with the number of robots and has no ceiling.

3.3. Graze

The *Graze* task differs from *Forage* and *Consume* in that discrete attractors are not involved. Instead, the object is to completely cover, or visit the environment. Some familiar examples are mowing the lawn, sowing seed, and of course, cows grazing. The *Graze* task for a *robot* is to search for an area that has not been grazed, move towards it, then graze over it until the entire environment (or some percentage of it) has been covered. It is assumed that the robot possesses some means to "graze" and that it grazes over a fixed "swath." The size of the task is dictated by the proportion of environment that must be covered before completion. Figure 1c shows a simulation of two robots grazing over 95% of the environment. The robots leave dashed lines as they wander, and solid lines when they graze. Grazing robots might be used to mow, plow or seed fields, vacuum houses [37], or remove scrub in a lumber producing forest.

The size of the swath that a robot can graze, and the percentage of the area that the robot must graze over both affect how long it takes to complete the task. Multiple robots can complete the task faster if they avoid traversing already grazed areas and if they can find ungrazed areas quickly.

3.4. Task Parameters

Each of the task definitions include parameters that affect the speed at which a robotic system can carry them out. These are the most important:

- **Number of attractors.** Clearly the number of attractors the robots must collect or consume will affect how long it takes to accomplish the task.
- **Mass of attractors.** In general terms, an attractor's mass can be thought of as a "transportability" factor for the *Forage* task, or a "workability" factor for the *Consume* task.
- **Graze coverage.** For the *Graze* task, the total size of the area and the percentage required to be grazed directly impacts the time to cover it.

Sections 7 and 8 report experimental results on how each of these factors affect performance.

3.5. Complex tasks

For this work, only the three basic tasks and the behaviors necessary for robots to perform them are considered. The results for these tasks are important because more complex tasks are easily described as combinations of simpler ones. Consider a robot removing scrub from a forest, after

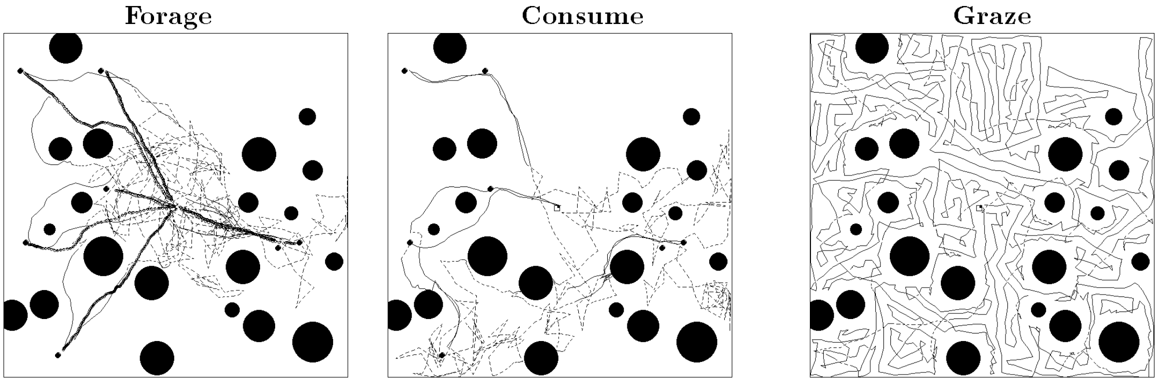

Fig. 1. Simulation of Forage, Consume, and Graze with two robots and seven attractors.

working for a period of time, it must return to a refueling station. The scrub removal portion of the task is analogous to *Graze*, while refueling is similar to *Consume*.

Another complex task, *BoundingOverwatch*, is a movement tactic utilized by Army Scouts. Usually employed by two groups of two ground vehicles, it allows safe penetration into hostile areas. Each group moves forward a short distance, then waits and "covers" the other group as it moves forward. A behavior to perform *BoundingOverwatch* can be built as a more specialized and coordinated *Consume* task. Once appropriate waypoints for each group are selected, virtual attractors can be placed there. The behavior would emerge as each two element group successively moves from attractor to attractor.

Other research in our laboratory is underway which investigates how complex behaviors may be specified as combinations of basic behaviors [36]. The research includes a language which allows individual robots, and societies of robots to be described formally. Formal operators allow basic, or primitive, behaviors to be grouped into more complex assemblages. These assemblages are further combined to form the overall behavior of the robot. The language includes operators that coordinate individual robots into cooperating groups. For clarity, this article will describe the robot behaviors somewhat less formally than in this related work, but the same recursive philosophy applies.

4. Reactive Control

A schema-based reactive control system is used in this research. To provide the reader appropriate background, a brief summary of reactive control is first provided, followed by some of the special characteristics of schema-based systems.

Reactive control is a paradigm which emerged in the mid-1980's as a new approach to controlling robots. It arose in response to perceived problems in earlier research which required heavy reliance on internal world models. Reactive control is characterized by several distinct features:

- Multiple parallel behaviors are constructed in a modular fashion.
- The design of the systems is in a bottom-up manner, incrementally adding more and more competence to the robot.
- Perception and action are tightly coupled.
- Reliance on explicit world models and representational knowledge is avoided during execution.
- They are particularly well-suited for dynamic and unstructured domains.

Brooks' subsumption architecture is a well-known example of this control paradigm [15]. Other representative examples include [3], [30], [46], [39], [49]. These various strategies differ in several significant ways including the organization and decomposition of behaviors and whether arbitration, action-selection, or concurrent processing is used. The interested reader is referred to [8] for a more complete review.

Schema-based reactive control has been widely used with success in our laboratory for both simulation studies and real robot implementations (e.g., [3], [5], [12], [13], [14]). Some features which distinguish schema-based robotic control from other reactive approaches include:

- A dynamic network of processes (schemas) is used rather than a hardwired layered system.
- No arbitration is used, instead behaviors (schemas) execute concurrently.
- Potential field techniques are used to encode the robot's behavioral response.
- Flexibility is introduced by allowing high-level knowledge and planning to select and parameterize the system [6].
- Adaptation and learning are facilitated through this flexibility [17], [47], [48].
- Neuroscientific, psychological, and ethological studies provide motivation for schema use. [7]

In schema-based control, each of the active behaviors (motor schemas) computes its reaction to its perceptual stimuli using a method analogous to potential fields [3]. It must be noted that unlike traditional potential fields [33], [32], the entire field is never computed, only the robot's reaction to its current perception of the world at its present location. All of these independent computations are summed and normalized and then sent to the robot for execution. This perceive-react cycle is repeated as rapidly as possible and is facilitated by the use of action-oriented perception [4] which permits only task-relevant information to be processed on a need-to-know basis. Problems with local minima, maxima, and cyclic behavior which are endemic to many potential fields strategies are handled by several methods including: the injection of noise into the system [3]; resorting to high-level planning [6]; repulsion from previously visited locales [14]; continuous adaptation [17]; and other learning strategies [47], [48]. Schema-based robot control has been demonstrated to provide robust navigation in complex and dynamic worlds. The Appendix contains information on the specific computation of the individual schemas used in this research.

4.1. Baseline Assemblage Parameters

Experimental results were generated for the tasks described in Section 3 by comparing performance of proposed robotic systems to baseline, or control, performance results. The baseline data was computed by first selecting a reasonable set of control parameters, then running a statistically significant number of simulations. Values for these parameters are based on previous research [5]. In this section, the behaviors for executing the three tasks (*Forage*, *Consume*, and *Graze*) and their baseline parameters are described.

At the highest level, the tasks themselves are assemblages which are represented as finite state acceptors (FSAs) consisting of several states. FSAs provide an easy means for both expressing and reasoning about behavioral sets by providing formal semantics [11]. Each state corresponds to a separate assemblage in which a constituent set of motor schemas is instantiated if that particular state is active. *Perceptual Triggers* cause transitions between states. Each active motor schema has a perceptual schema associated with it to provide the information necessary for the robot to interact with its environment.

4.2. Forage

For the *Forage* task, the robots can be in one of three states: *wander*, *acquire*, and *deliver*. All robots begin in the *wander* state. If there are no attractors within the robot's field of view, the robot remains in *wander* until one is encountered. When an attractor is encountered, a transition to the *acquire* state is triggered. While in the *acquire* state, the robot moves towards the attractor and when it is sufficiently close, attaches to it. The last state, *deliver*, is triggered when the robot attaches to the attractor. While in the *deliver* state the robot carries the attractor back to home base. Upon reaching home base, the robot deposits the attractor there and reverts back to the *wander* state. Figure 2 shows the FSA for *Forage*. [1]

For each state, the active schemas and their parameters are:

- *Wander* State
  - **noise**: high gain, moderate persistence to cover a wide area of the environment.
  - **avoid-static-obstacle** for objects: sufficiently high to avoid collisions.

- **avoid-static-obstacle** for robots [2]: moderately high repulsion to force individual robots apart and more efficiently cover the environment.
- **detect-attractor**: perceptual schema that triggers the *acquire* state when the robot senses an attractor.

• *Acquire* State
- **noise**: low gain, to deal with local minima.
- **avoid-static-obstacle** for objects: sufficiently high to avoid collisions.
- **avoid-static-obstacle** for robots: very low gain, to allow robots to converge on the same attractor and thus cooperate, but avoid colliding with one another.
- **move-to-goal**: high gain to move the robot to the detected attractor.
- **detect-attachment**: a perceptual schema that triggers a state transition to *deliver* when the robot is close enough to attach to the attractor.

• **Deliver State**
- **noise**: as in *acquire*, low gain to deal with local minima.
- **avoid-static-obstacle** for objects: as in *acquire*, sufficiently high to avoid collisions.
- **avoid-static-obstacle** for robots: same as in *acquire*.
- **move-to-goal**: high gain, with home base as the target.
- **detect-deposit**: a perceptual schema that triggers a state change when the robot reaches home base.

Specific values used for schema gains and parameters in this study are listed in Table 1 (Sec. 6.3).

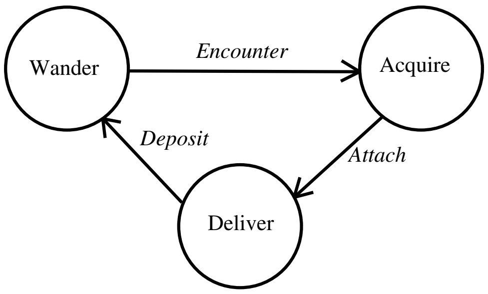

Fig. 2. The Forage FSA

4.3. Consume

The FSA and behaviors for the *Consume* task (Figure 3) are similar to those used in *Forage*. In fact, the schemas and their gains are identical in the *wander* and *acquire* states. The *consume* state, however, is unique to to this behavior. In the *consume* state, only one motor schema, **consume-attractor** is activated. It reduces the mass of the attractor at a fixed rate over time. When the attractor is fully consumed (mass zero) it is deactivated and the robot transitions back to the *wander* state. Table 1 shows the schema parameters for *Consume*.

4.4. Graze

For the *Graze* task, the *wander* and *acquire* states are again similar to those of *Forage* and *Consume*. The primary difference is that **detect-attractor** in the *wander* state is replaced with a similar **detect-ungrazed-area** schema. **Detect-ungrazed-area** has the same fixed sensor range as **detect-attractor**, but it detects ungrazed areas instead of attractors. Each robot starts in the *wander* state and searches for ungrazed areas. Upon encountering one, it transitions to the *acquire* state and moves towards it. When the robot arrives at the graze site, it transitions to the *graze* state. The *graze* state is quite different from the corresponding states in the other FSAs. While in the *graze* state, the robot tends to move along its current heading as it "grazes" over a fixed swath of the environment. As long as there continues to be ungrazed areas directly ahead, the robot remains in the *graze* state. The active schemas for this state are:

• **noise**: low gain, to deal with local minima.

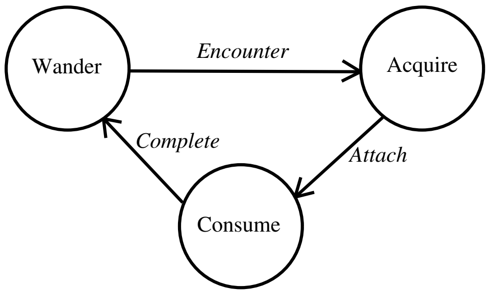

Fig. 3. The Consume FSA

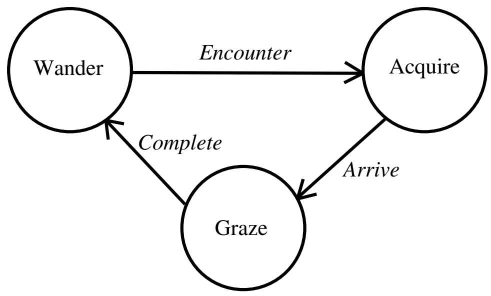

Fig. 4. The Graze FSA

- **avoid-static-obstacle** for objects: high enough to avoid collisions.
- **avoid-static-obstacle** for robots: very low, to allow robots to graze close by, but avoid collisions.
- **probe** - moderate gain, to encourage the robot to keep moving along its current heading towards ungrazed areas.
- **graze** - performs the actual graze operation over a fixed swath.
- **detect-grazed-area** - perceptual schema that triggers a state change once the robot has completely grazed the local area.

For simulation purposes, *Graze* is implemented by maintaining and marking a high resolution grid corresponding to the environment. Initially, the entire grid is marked as ungrazed. As robots graze, they mark visited areas on the grid accordingly.

Gains and parameters for each of the schemas active in the graze state are listed in Table 1.

5. Forms of Inter-agent Communication

Three different types of communication are evaluated in this research. Using a minimalist philosophy, the first type actually involves no direct communication between the agents. The second type allows for the transmission of state information between agents in a manner similar to that found in display behavior in animals [42]. The third type (goal communication) requires the transmitting agent to recognize and broadcast the location of an attractor when one is located within detectable range. Each of these forms of communication is described in more detail below.

5.1. No Communication

For this type of multiagent society no direct communication is allowed. The robots are able to discriminate internally three perceptual classes: other robots, attractors, and obstacles. None of this information, however, is communicated to other agents. Each robot must rely entirely upon its own perception of the world. Arkin has shown in previous work [5] that this basic information is enough to support cooperation in robot retrieval tasks (*Forage*). Cooperation in this context refers to the observed phenomena of recruitment, where multiple agents converge together to work on the same task. The baseline results (Section 7) show that cooperation also emerges in the *Consume* and *Graze* tasks as well.

5.2. State Communication

When state communication is permitted, robots are able to detect the internal state (*wander*, *acquire*, or *deliver*) of other robots. For the results reported in this article, the communication is even simpler than that, where only one bit of data is transmitted: with zero indicative of an agent being in the *wander state* and one indicating that it is in any state other than *wander* (i.e., *acquire*, *deliver*, *consume*, or *graze*). In [9], this type of communication was shown to provide a distinct advantage over no communication for performance of the *Forage* task. Communication is often considered a deliberate act, but state communication is not necessarily "intentional" since information can be relayed by passive observation. The sender does not necessarily explicitly broadcast its state, but allows others to observe it. In nature this type of communication is demonstrated when an animal changes its posture or external appearance, such as a dog raising its hackles or exhibiting flight behavior in response to fear.

To take advantage of state information in reactive control, the behavioral assemblages for each task are modified slightly. ¿From a robot's point of view, the most important states to look for in another robot are those where the other robot has found an attractor or an area to graze; that means that the other robot has found useful work. If the robot goes to the same location, it is likely to find useful work as well, or at least be able to assist cooperatively. The appropriate states are *acquire*, *deliver*, *consume*, or *graze* ; in the *wander* state the robot has not yet found any work to do.

For all three tasks, the behaviors are modified so that a robot will transition to *acquire* if it discovers another robot in *acquire*, *deliver*, *consume*, or *graze*. Since the robot may not yet know the location of the attractor, it follows the other robot instead. Once the attractor is detectable it heads directly for it.

5.3. Goal Communication

Goal communication involves the transmission and reception of specific goal-oriented information. Implementation on mobile robots requires data to be encoded, transmitted, received, and decoded. Goal communication differs from the other two levels in that the sender must deliberately send or broadcast the information. A natural example of this type of communication is found in the behavior of honeybees. When a bee discovers a rich source of nectar, it returns to the hive and communicates the location with a "dance" which encodes the direction and distance from the hive to the source.

For reactive control, goal communication is implemented by modifying the behavioral assemblages in the same manner as described for state communication. However, instead of following the transmitting robot that discovered the attractor, a receiving robot moves directly toward the location of the attractor. The intent is that the agent may now follow a more direct path (beeline) to the attractor.

This very rudimentary form of communication only broadcasts the goal that the transmitting agent is involved with. Another mode of communication, not yet explored, involves the transmission of all detected attractors independent of whether the transmitting agent is already acquiring or delivering one. This would present more options for the receiving agent, perhaps choosing to move to the closest attractor independent of whether or not the transmitting agent would benefit from its help. This additional form of communication is left for future work.

5.4. **Explicit versus Implicit Communication**

The implementation of goal and state communication requires explicit signaling and reception of the communicated information. State communication can be implemented simply by mounting a binary signal atop the robot which is either on or off depending on the robot's internal state. This communication, although trivial, is explicit as it requires the deliberate act of invoking the signal.

Information pertinent to cooperation might be gathered by other means as well. The internal state of a robot could be inferred by observing its movement (e.g., recognizing a robot in the *wander* state due to apparent random movements), thereby placing a larger perceptual burden on the receiving agent. Robots can also communicate through their environment. In the graze task, robots leave evidence of their passage since the places they visit are modified. This fact is observable by the other robots. These types of communication are referred to as *implicit* since they do not require a deliberate act of transmission.

Implicit communication was found to be an important mode of cooperation in simulations of the graze task. Since this communication emerges from the interaction of the agent and the environment, it cannot be "turned off." Thus comparative analyses of performance with and without implicit communication are not meaningful.

6. Simulation Environment

The simulation environment should provide an accurate estimate of robot performance in the real world. Simulation is important because it offers a means to test many robot system configurations quickly. To be useful, the simulation must report performance in terms of the prescribed performance metric and realistically emulate the environment and the robot's interaction with it. Furthermore, the simulation must allow hardware, control, and environmental variables to be readily manipulated.

The test environment for this research is written in C using the X Windows graphics package. The simulator has been a useful tool for other research in the Mobile Robot Lab at Georgia Tech, including [9], [17], [47], [37], [14] among others. Results generated in this simulation environment have routinely been demonstrated on actual mobile robots (e.g., [3], [12], [13], [14]). Except for minor changes [3], the present simulator is the same one used in these earlier projects. The simulator may be run in a visual mode, or in a text-only mode. The visual mode is used primarily for debugging and qualitative accessments. The text-only mode is used for multiple runs to gather extensive statistical data.

Each robot is an identical holonomic vehicle which is controlled by one of the task assemblages described above. Each agent's current state, however, is dependent solely on its own perception. The robotic agents execute their tasks in a 64 x 64 unit environment. The units are dimensionless, but for convenience of comparison to real robot implementations they represent one foot. Time is measured in steps. Each step is one iteration of the program that calculates the robots' next positions. The robots are able to sense their location in the environment, and detect obstacles, attractors and other robots within a fixed radius field of view. They are able to grasp and carry attractors, consume attractors, or graze as the task dictates. The simulation automatically enforces the limits and rules set forth in the task specifications, as well as sensor/actuator limits. The robots are allowed to move without restriction within the 64 x 64 environment, but they may not move outside of it.

6.1. The Performance Metric

What is "performance"? Since one goal of this research is to report the impact of communication on robotic societies, performance must be objectively measurable. Selection of a performance metric is important because these metrics are often in competition - i.e., cost versus reliability. Some potential metrics for multiagent robotic systems are:

- **Cost -** Build a system to accomplish the task for the minimum cost. This may be appropriate for many industrial tasks. Use of this metric will tend to reduce the cost of the system and minimize the number of robots used.
- **Time -** Build a system to accomplish the task in minimum time. This metric will lead to a solution calling for the maximum number of robots that can operate without interference.
- **Energy -** Complete the task using the smallest amount of energy. This is appropriate in situations where energy stores are limited, e.g., space or undersea applications.
- **Reliability/Survivability -** Build a system that will have the greatest probability to complete the task even at the expense of time or cost. This may be useful for certain strategic military applications.

The task metric can also be a numeric combination of several measurements. Whatever the metric is, it must be measurable, especially in simulation. For this research, time to complete the task was chosen as the primary performance metric. It is easily and accurately measurable and conforms to what is frequently thought of as performance. No claim is made however that this is the "best" metric; robot path length or energy consumption may be equally useful. In the simulation studies described herein, performance is measured by counting how many iterations the simulation program executes before the task is completed.

There are a few initial conditions for some tasks that prevent the robots from completing it. For example, if an attractor was somehow placed within a circle of obstacles, the robots would never be able to reach it. Such a scenario is not solvable by any robot system without the capacity to move the obstacles. Other scenarios, however, may ultimately be solvable, but may potentially defeat the purely reactive strategies presented here. To provide for these situations, the simulation is allowed to continue for 8000 steps before failure is declared. Since most runs complete in less than 2000 steps, it is highly likely that the system will *never* complete the task if it does not do so before failure is declared. The objective is to evaluate the impact communication makes on performance, so it is not important to know why the system failed, just to measure how it improves with communication. In cases of failure, the run is recorded as having taken 8000 steps. This approach reports optimistic performance since the run might never have completed (infinite steps). But, to show improvement over a failure case, the system must actually complete the task *and* in less than 8000 steps.

6.2. *Environmental Factors*

As much as can be known about the target system's operating environment should be incorporated into the design process for the control system. If these factors are known *a priori*, they can be included in the simulation. Important environmental factors include:

- **Mobility factors:** Is the terrain mountainous or flat? What percent of the environment is served by roadways?
- **Obstacle coverage:** What percent of the environment is cluttered with obstacles?
- **Metric** *a priori* **knowledge:** Does the robot have a good map of the area or is it completely unknown?
- **Static or dynamic:** Is the environment filled with moving objects, thus reducing the utility of maps, or is the environment a static one?

For this study, a static flat environment with randomly scattered obstacles is assumed. No *a priori* knowledge of the obstacles' location is available. Obstacle coverage is varied from 5% to 20% of the total area, with 15% as a baseline.

6.3. **Motor and Sensor Constraints**

As a step in the robot system design methodology, realistic bounds on the expected motor and sensor capabilities of robots are set. These bounds help reduce the search space for an optimum solution. The affect of communication on performance is the main thrust of this research, so fixed values representing the expected performance of the robots were used. If the goal were to determine optimal sensor or motor requirements, those parameters could be varied as well. Table 1 shows the experimental motor and sensor values used in the simulations.

7. Baseline Results and Analysis Tools

To build a baseline database of performance measurements, a configuration of environment, control, and task parameters was selected empirically (Table 1). The baseline database serves as a control for comparison in the evaluation of the communication experiments described below. The database is generated by running the simulation using the baseline configuration parameters for each of the three tasks: *Forage*, *Consume*, and *Graze*. For each task, the number of robots and the number of attractor objects (or percentage of graze coverage) is varied. For each combination of robots and attractors, a measure of performance is taken by timing runs on 30 different randomly generated scenarios. Overall performance is the average of those 30 runs. For each run, the simulation records the number of steps taken, and whether or not the run timed-out (failed).

The baseline performance measurements were made with no communication allowed between the robots. This control is then compared with the performance in each of the three tasks when state or goal communication is allowed (Section 7). ¿From these comparisons, one can see quantitatively how these modes of communication impact performance.

7.1. *Basic Performance*

Performance data is visualized as a 3-dimensional surface with the X axis reflecting the number of robots and the Y axis indicating the number of attractors or percent coverage[4] (see Figure 6). The Z, or height, axis shows the average time to complete the task for that combination of robots and attractors (smaller numbers are indicative of better performance).

The plots for all three tasks share a similar shape. Notice that the back left corner is the highest point on the three surfaces. This is expected since that location represents the case where one robot by itself must complete the most work (seven attractors for forage and consume, 95% coverage for graze). Similarly, the right front is the lowest point, since the largest number of robots (five) complete the least amount of work (one attractor). It is also apparent for all three tasks that performance initially improves sharply as more robots are added, but then tapers off. In some cases, performance does not improve much

Table 1. Experimental Parameter Values. Unless noted otherwise, the values are the same for all three tasks.

Factor	**Baseline**	**Experimental Range**
Task Factors		
Number of attractors	-	1 to 7
Mass of attractors	5 avg	1 to 8
Graze Coverage	95%	13% to 95%
Environmental Factors		
Obstacle Coverage	15%	10% to 25%
Obstacle radius	-	1.0 to 4.0
Number of Robots	-	1 to 5
Sensor and Motor Constraints		
Maximum Velocity	2 ft/step	fixed
Attractor Sensor Range	20 ft	fixed
Obstacle Sensor Range	20 ft	fixed
Communication Range	100 ft	fixed
Communication Type	No	No,State,Goal
Graze Swath	2 ft	fixed
Consume Rate	0.01 units/step	fixed
Control Parameters		
Obstacle Sphere of Influence	5 ft	fixed
Obstacle Repulsion Gain	1.0	fixed
Robot Repulsion Sphere	20 ft	fixed
Robot Repulsion Gain (*wander*)	0.5	fixed
Robot Repulsion Gain (*acquire*)	0.1	fixed
Robot Repulsion Gain (*deliver, graze*)	0.1	fixed
Move-to-Goal Gain (*acquire*)	1.0	fixed
Move-to-Goal Gain (*deliver*)	1.0	fixed
Probe Gain (*graze*)	1.0	fixed

at all with more than 4 robots. This is important if robots are expensive.

To illustrate, suppose a robotic system for the *Forage* task should be both fast *and* inexpensive.

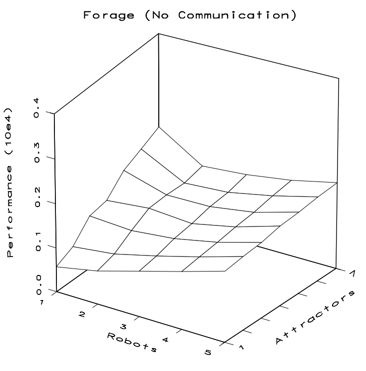

Fig. 5. Optimizing in *Forage* for time and cost. Performance here is defined as time to complete the task plus the number of robots times 300 (no communication).

Performance is then a combination of the time to complete the task and the cost of the system. Ultimately, the designer must balance the importance of cost versus speed of completion, but one approach is to amortize the cost of the robotic system over its expected lifetime. Thus the cost of one run is the overall cost divided by the expected number of runs. For this example, suppose the amortized cost of each robot per run is valued the same as 300 time steps. Then if N is the number of robots, and T is the time to complete the task, the overall performance is:

$$P = N * 300 + T \tag{1}$$

Using timing measurements taken for *Forage* and adding in amortized cost, a three dimensional surface is generated for the new performance metric (Figure 5). A system with two robots is generally best for three or more attractors. If the environment is expected to contain only one or two at-

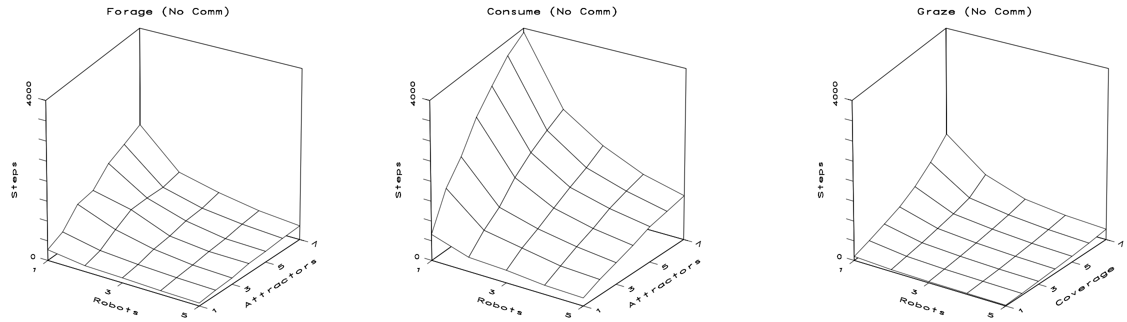

Fig. 6. Time to complete the *Forage*, *Consume*, and *Graze* tasks for one to five robots and one to seven attractors with no communication.

tractors, one robot is the best choice. Even though more robots may be faster, the overall goals of the designer may call for fewer.

7.2. Speedup

Another effective tool is *speedup* measurement. A plot of speedup reveals how much more efficient several robots are than just one in completing a task. If $P[i,j]$ is the performance for i robots and j attractors, the speedup at that point is:

$$S[i,j] = \frac{\frac{P[1,j]}{i}}{P[i,j]} \tag{2}$$

So, if two robots complete the task exactly twice as fast as one robot, speedup is 1.0 (higher numbers are better). Mataric introduced a similar metric of robot performance in [40]. Anywhere speedup is equal to 1.0, the performance is said to be *linear*. *Superlinear* performance is greater than 1.0, and *sublinear* is less than 1.0. Realize, however, that in some cases more robots will be faster for actual task completion time, but still offer sublinear speedup.

Figure 7 shows speedup plots for *Forage*, *Consume*, and *Graze* without communication. Note that speedup for all tasks is generally higher for larger numbers of attractors. Researchers in other branches of computer science have found that randomized search tasks are often completed in superlinear time on parallel systems [28]. Since the *wander* behavior used in all three tasks essentially solves a randomized search task, it is not surprising that performance is superlinear when this behavior is heavily utilized, as is the case when there are large numbers of attractors.

Surprisingly, speedup in the *Consume* task is sublinear at all but one point (Figure 7b). The behavior in the *consume* state can at most offer linear speedup (the limit is set by the specification of the task). So an environment with massive attractors will force the speedup to be limited near 1.0.

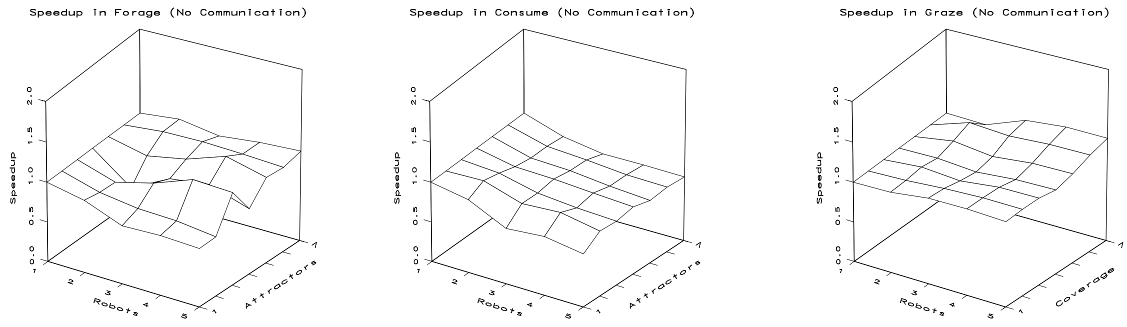

Fig. 7. Speedup in the *Forage*, *Consume*, and *Graze* tasks for one to five robots and one to seven attractors (no communication).

Table 2. Summary of speedup data for three tasks.

Task	Avg. Speedup	Best	Worst
Forage	0.93	1.15	0.64
Consume	0.82	1.01	0.65
Consume(low mass)	0.89	1.26	0.66
Graze	1.07	1.21	0.97

This hypothesis was tested by reducing the average mass of the attractors, then rerunning the simulations. In the baseline runs, attractor mass varies from 2.0 to 8.0 units, but for these experimental runs, mass was reduced to 1.0 to 4.0 units. Reducing attractor mass allows the robots to spend more time wandering (a superlinear task) instead of consuming (at most linear). The speedup for *Consume* with lower mass attractors is shown in Figure 7. At every point on the surface, speedup is better for low mass attractors than for high mass. In fact, in many cases speedup is superlinear.

Speedup in the *Graze* task is superlinear at all but three points on the surface (Figure 7). In the very worst case, speedup dips to 0.97. Situations requiring a high percentage of graze coverage result in the best speedup; the peak is 1.21 for five robots and 95% coverage. In cases where high graze coverage is required, robots spend more time in *wander* as they look for the last bit of area to graze. Again, since *wander* is a superlinear time task, the best speedups should be expected for those regimes.

Speedup results are summarized in Table 2.

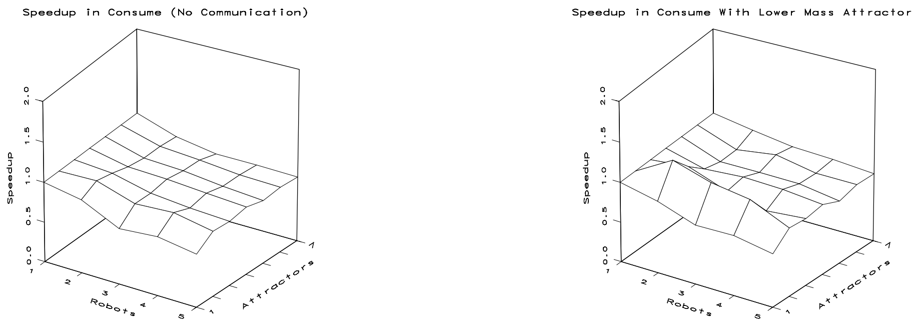

Fig. 7. Side by side comparison of speedup in the *Consume* task (without communication). Performance with attractors of average mass 5.0 (left) and 2.5 (right).

7.3. Timeouts

A *timeout* occurs when a simulation run exceeds a time limit (for these experiments, the limit is 8000 steps). A timeout mechanism is necessary to avoid lockups in infinite loops in the event the society is unable to complete the task for that particular world. Frequency of timeouts for each combination of robots and attractors is measured and plotted in Figure 8. The frequency of timeouts serves primarily as a measure of data quality. In situations where timeout frequency is higher, the experimenter cannot know for sure how long the runs would have taken if they were allowed to complete. Some runs may have completed while others may have run indefinitely. When there are relatively few timeouts, the performance is known with greater certainty. As would be expected, most timeouts occur when fewer robots must solve a task with more attractors or a higher graze coverage requirement.

7.4. Summary of Baseline Results

Baseline results serve as a control for experimental comparison in assessing the impact of other communication modes on performance. It is important to derive and understand fully these basic results before testing more complex robot configurations. Important results for the baseline configuration are:

- For a given number of attractors, more robots complete a task faster than fewer robots.
- For a given number of robots, it takes longer to complete a task with more attractors.

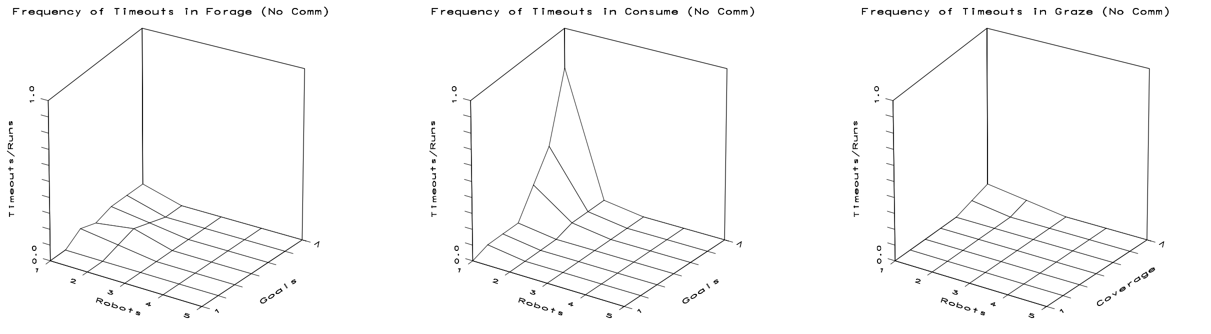

Fig. 8. Frequency of timeouts (percent) in the *Forage*, *Consume*, and *Graze* tasks for one to five robots and one to seven attractors (no communication).

- Some performance metrics may result in a system that is optimized with lower numbers of robots than for other metrics.
- Speedup is greater in scenarios where larger numbers of attractors are present.
- Speedup in the *Consume* task is mostly sublinear, but can be superlinear for lower mass attractors.
- Speedup in the *Graze* task is mostly superlinear.
- Timeouts occur more often for low numbers of robots and high numbers of attractors.

8. Results with Communication

8.1. Communication in the Forage Task

Figure 9 shows a typical simulation run of two robots foraging for seven attractors with no, state, and goal communication. Inspecting the images from left to right reveals an apparent improvement in the "orderliness" of the robots' paths. The quantitative experimental results summarized in Table 3 confirm these qualitative impressions.

Figure 6a shows a typical performance plot for *Forage*, in this case for no communication (better performance is lower). Each data point represents 30 different simulation runs. The plots for no, state, and goal communication are quite similar in contour but there is improvement in performance evidenced by lower surfaces as the communication becomes more complex. The statistical analysis in Table 3 summarizes these observations.

To quantify the difference between performance with and without communication, a performance ratio plot is computed (Fig. 10). At each point, the performance with communication is divided by the performance without communication. Results greater than 1.0 imply improved performance. For instance, a value of 1.1 indicates 10% improvement. For all the cases tested, State communication improved performance in the *Forage* task an average of 16%. On the average, goal communication is 3% better than state communication in the *Forage* task.

8.2. Communication in the Consume Task

The impact of communication on performance of the *Consume* task is similar to that in *Forage*. Figure 11 shows a typical simulation of two robots consuming seven attractors with no, state, and goal communication. A surprising result is that the simulation with goal communication actually takes longer than the one with state communication. This slight increase in run time with goal versus state communication is typical for this task.

A representative example of the basic performance data for simulations of the *Consume* task is plotted in Figure 6b. Again, the contours for all three forms of communication are quite similar. A comparative analysis reveals that on the average, state communication offers a 10% performance advantage over no communication. Goal communication is 4% worse on the average than state communication. Goal communication, however, is still 6% better than no communication at all. Table 3 summarizes these results.

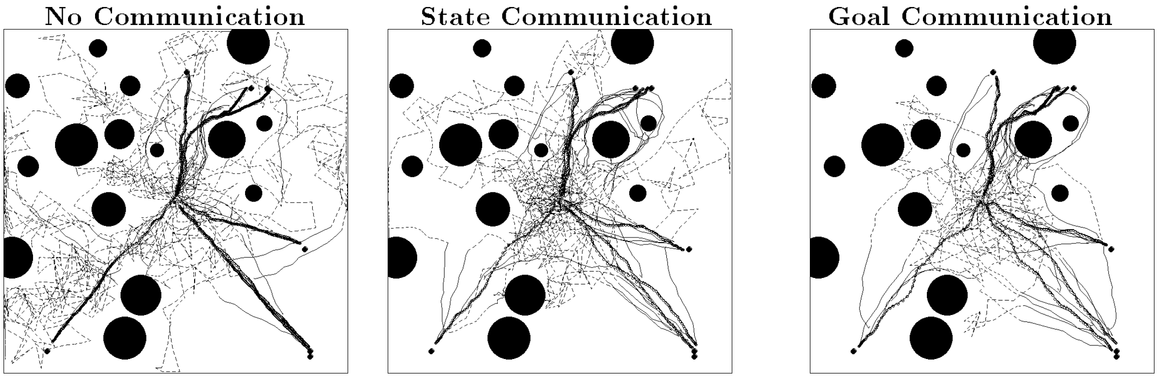

Fig. 9. Typical run for *Forage* task No (left), State (center), and Goal (right) Communication. The simulations required 5145, 4470 and 3495 steps, respectively, to complete.

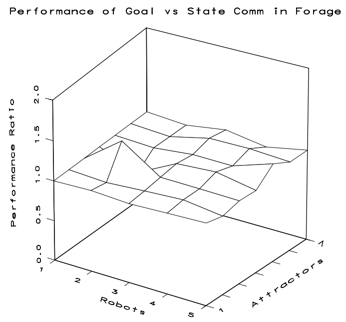

Fig. 10. Performance ratio plot for the *Forage* task for Goal versus State communication.

Recall that speedup in the *Consume* task is linked to attractor mass (Section 4). Attractor mass may also impact the benefit of communication. Analysis of the data from runs with low mass attractors reveals that goal communication performance is almost indistinguishable from that of state communication (1% worse). Future research may determine if this result is just an anomaly or if environmental and task parameters might shift this trend.

8.3. Communication in the Graze Task

The surprising result from Graze task simulations is that communication hardly helps at all. Plots of basic performance data for each of the different levels of communication are not shown because they are visually identical (see Figure 6c for the case with no communication). On average, state communication is only 1% better than no communication. Performance with goal communication is virtually indistinguishable from that with state communication (0% difference). Table 3 summarizes these results.

As robots graze they inevitably leave a record of their passage: the graze swath. This physical change in the environment is actually a form of implicit communication. The robots leave marks that advise others where work has or has not been completed. This result is important because it implies that for tasks where such implicit communication is available, explicit communication is unnecessary.

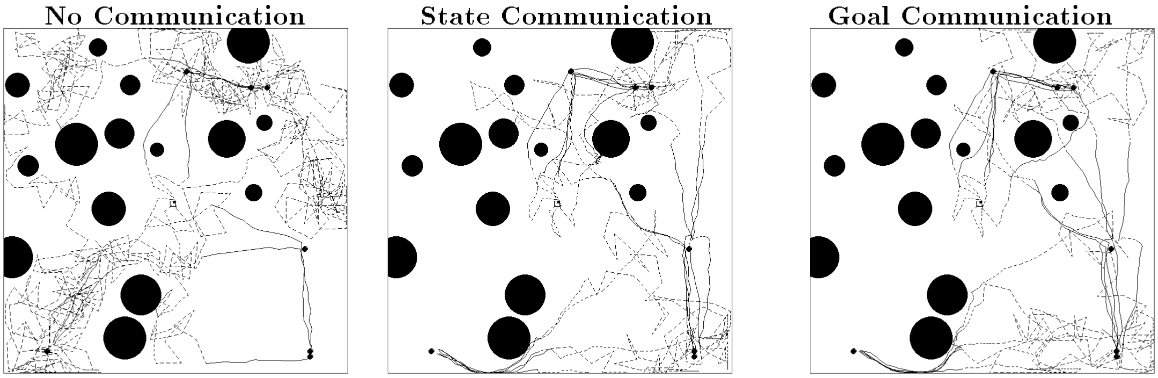

Fig. 11. The *Consume* task with No, State, and Goal Communication. The simulations required 4200, 3340 and 3355 steps, respectively, to complete.

Task	Average Improvement	Best	Worst
Forage			
State vs No Communication	16%	66%	-5%
Goal vs No Communication	19%	59%	-7%
Goal vs State Communication	3%	34%	-19%
Consume			
State vs No Communication	10%	46%	-9%
Goal vs No Communication	6%	44%	-16%
Goal vs State Communication	-4%	5%	-30%
Goal vs State (low mass attractors)	-1%	23%	-19%
Graze			
State vs No Communication	1%	19%	0%
Goal vs No Communication	1%	19%	0%
Goal vs State Communication	0%	0%	0%

Table 3. Summary of performance ratios for no, state and goal communication.

8.4. Summary of Results with Communication

The performance improvements each type of communication offers for each task are summarized in Table 3. Several important conclusions may be drawn:

- Communication improves performance significantly in tasks with little implicit communication (*Forage* and *Consume*).
- Communication appears unnecessary in tasks for which implicit communication exists (*Graze*).
- More complex communication strategies (Goal) offer little benefit over basic (State) communication for these tasks (i.e., display behavior is a rich communication method).

9. Results on Mobile Robots

The ultimate goal of this research is a working multiagent robotic system; simulation serves only as a development tool. To demonstrate the simulation results, and to move towards a completely functional society, the behaviors for *Forage*, *Consume*, and *Graze* must be instantiated on mobile robots. The target system is a group of three Denning mobile robots, George, Ren, and Stimpy. They each have three-wheeled kinematically holonomic suspensions and a ring of 24 ultrasonic range sensors. George, is a DRV-1; Ren and Stimpy are MRV-2s. Initial results were obtained by porting tasks to Driver, a menu-driven motor schema-based reactive control system written in C.

9.1. Omnidirectional Sensing

The behaviors used here can benefit from an omnidirectional sensor which enables robots to discriminate between other robots, attractors and obstacles. Two omnidirectional sensor systems have been evaluated for multiagent robot implementation at Georgia Tech: a conic mirror camera system, and a laser barcode reading system.

The conic mirror camera system was originally developed by MacKenzie for the Georgia Tech Unmanned Aerial Vehicle Team [35]. It is intended for localization of an unmanned robotic helicopter. Bright lights located about the environment serve as reference points for triangulation. The chief advantage is speed; image processing demands are comparatively low.

Conic vision can be readily adapted for mobile robot systems as follows. Marker lights are affixed to attractors and other robots. For differentiation, robots have two lights arranged vertically, while attractors only have one. A typical configuration is shown in Figure 12 (the system can be seen atop the robot Ren in Figure 11). Figure 13 is an actual image from the system. An attractor and robot are visible as short bright blobs on the left side of the image. Azimuth to an attractor may be computed trigonometrically. Since the robots are equipped with two lights having a known separation, both azimuth and range are

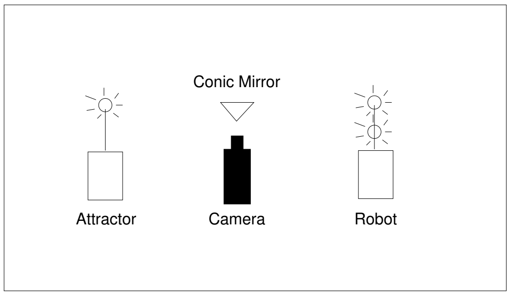

Fig. 12. Typical scene for conic vision processing.

Fig. 13. Image from the conic vision system. The bright blobs identify an attractor and another robot in the environment.

computable. Figure 14 shows how the separation between the blobs resulting from the two lights on a robot grows as it approaches.

The laser barcode reader is manufactured by Denning Mobile Robots. An upward-firing laser is positioned in the center of the device. A rotating mirror inclined at 45 degrees allows it to scan 360 degrees. The sensor is able to detect and accurately report the azimuth to targets coded with retro-reflective tape up to 60 feet away. The barcode reader is shown on Ren in Figure 15.

The barcode reader is supported by the Driver reactive control system, and has been utilized for both localization and attractor detection tasks. The system is well suited for multiagent robotic research. Presently only one barcode reader is available at Georgia Tech, so multi-robot deployment is impossible. The conic vision system is not yet supported by Driver.

Other omnidirectional sensing capabilities for George, Ren and Stimpy should be available shortly as our laboratory acquires new sensors such as IR or laser range scanners.

9.2. Forage

The *Forage* task described in Section 4 was ported and tested on Ren and Stimpy. Most of the required schemas had already been coded in Driver, but the lack of an existing omnidirectional sensor system for attractor and robot detection complicated matters. The problem was circumvented by simulating the sensor within an embedded perceptual schema utilizing shaft encoder data. Spatial locations of attractors and moving robots are maintained in continuously updated shared files. Fidelity is maintained by coding the perceptual schema so that it does not "reveal" the location of attractors or other robots until they are within sensor range.

A test with one robot, Ren, is depicted in Figure 11. The sequence shown was first videotaped and then images were captured for print from that tape later. Telemetry from the run is shown at the right of Figure 11. Initially, Ren is set up at home base. Two attractors are available for collection, marked by circles on the floor in the foreground and background. Another inactive robot, Stimpy is just off to the left. Even though Stimpy was not involved in the task directly, the **avoid-static-obstacle** schema for robot to robot repulsion was active on Ren.

Except for sensor range, parameters were set as in the baseline simulations (Table 1). Since the test area is rather small, attractor sensor range was reduced from 20 to 10 feet. This value, nonetheless, prevented Ren from immediately sensing both attractors at home base.

Fig. 14. In this image the other robot has been drawn closer. Note the increased separation between the two blobs.

Fig. 11. A Denning robot, Ren, demonstrates the *Forage* task (left). Ren tags an attractor (center). Telemetry from the Single-Agent Forage demonstration is shown at right. Home base is in the center. Two attractors are located on the left and right. The inactive robot Stimpy is in the lower left.

The foreground attractor is eight feet from home base, so at the start of the test it is immediately visible. Ren moves towards it and "tags" it in Figure 11. The notional attractor is carried back to home base. Ren transitions to *wander*. Note that the other robot, Stimpy, is located between Ren and the remaining attractor. Since *wander* includes a strong robot to robot repulsion, Ren continues to search away from the attractor. The assumption is that Stimpy would search the rest of the space, but since Stimpy is inactive and there is no communication present, the attractor might take an inordinate amount of time to be discovered. A human steps in to help. The human is able to herd Ren towards the attractor by placing his hands near the ultrasonic sensors. Once Ren gets within 10 feet of the attractor, it transitions to *acquire* (the human leaves), and then the robot tags it. Finally, the attractor is deposited at home base.

A two robot run of the *Forage* task is shown in Figure 15. Again, the parameters are those from the baseline simulation runs, except for the attractor sensor range which was set at 10 feet. The minimum range a robot could approach an obstacle was set at two feet. There are three attractors (boxes) and one obstacle (chair) in the environment. Both robots were initialized at home base. This run was made without communication. At the beginning of the run (Fig. 15), the robots enter the *wander* state, and are repulsed by each other. They immediately detect separate attractors. After tagging their respective attractors, the robots deliver them to home base. Again the robots cycle to *wander*. Only one attractor remains (in the foreground). The attractor is within Ren's sensor range, but outside Stimpy's, so Ren approaches it alone. As Ren returns the attractor to home base, it carries it within Stimpy's sensor range. Stimpy responds by approaching Ren and helping to deliver the attractor. A (hand-drawn) reconstruction of this run is shown in Figure 16.

9.3. Communication modes and *Consume*

All three levels of communication for the *Consume* task have been implemented and tested on Ren and Stimpy. A scenario for the two robots with one attractor was used in testing the *Consume* behavior (Figure 17). Although the scenario is simple it serves to illustrate the advantages of and the qualitative differences between the three levels communication described in Section 5. Runs on mobile robots are directly compared with simulations of the same scenario in Figure 17.

In the test scenario, two robots and one attractor are arranged so that one robot is immediately within sensor range of the attractor, while the other is just outside sensor range. In the simulations, the attractor is 20 feet from the lower robot. If no communication is allowed, one robot should initially move towards the attractor. The other robot should move away, due to inter-robot repulsion. If communication is allowed, both robots should initially move towards the attractor since at least one of them senses it.

Fig. 15. Two Denning robots, Ren and Stimpy, demonstrate the Forage task (upper left). Ren tags an attractor (upper right). Stimpy "tags" an attractor (lower left). Ren and Stimpy deliver the attractors to home base (lower right).

These predictions are borne out in the simulations shown in the top row of Figure 17. The simulations were run in the environment described in Section 6 using the baseline control parameters (Table 1). In the case of No Communication, Robot 1 immediately moves to the attractor and begins consuming it (top left). Robot 2 moves away, and continues to search for attractors in the *wander* state. Eventually it too falls within sensor range of the attractor, moves towards it, and helps consume it. In the case of State Communication (top center), Robot 1 again initially moves towards the attractor. Robot 2 begins to follow it (dotted line), then transitions to the *acquire* state (solid line) when it comes within sensor range of the attractor. Finally, in the case of Goal Communication (top right), both robots immediately move to the attractor and consume it. A qualitative difference between State and Goal Communication is visible in the paths Robot 2 takes to the attractor in Figure 17 (top row). With State Communication, Robot 2, initially outside sensor range of the attractor, makes a curved path to the attractor since it can only follow Robot 1 initially (top center). When Goal Communication is allowed, however, Robot 2 can proceed directly to the attractor (top right).

Now compare the simulations (top row) with runs on the robots Ren and Stimpy (bottom row). Since the sensor range of the robots is set at 10 feet, the scenario was altered for runs on mobile robots so that the attractor is only 10 feet away from the lower robot. The telemetry is shown at half the scale of the simulated runs to account for the smaller scale of the scenario.

Qualitatively, performance for mobile robots with No Communication is quite similar to simulated performance (Figure 17 bottom left). Initially, Ren does not sense the attractor and explores the left side of the laboratory instead. But eventually, it comes within sensor range and moves to the attractor. When State Communication is allowed Ren follows Stimpy to the attractor, making a curved path (bottom center). Fi-

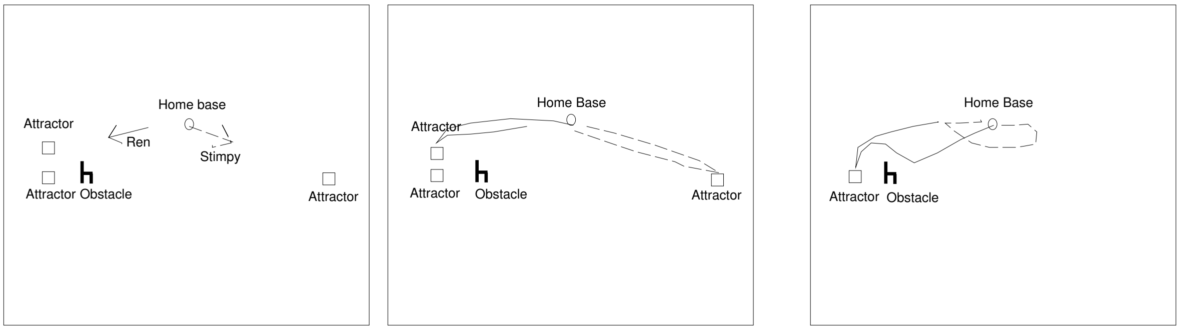

Fig. 16. A reconstruction (from above) of the Forage demonstration.

nally, when Goal Communication is allowed, Ren travels directly to the attractor (bottom right).

The path of the lower robot for the cases of State and Goal Communication is somewhat different in simulation than on mobile robots. On mobile robots, the lower robot curves away from the upper robot much more than in simulation. This is a result of two factors. First, the scale of the telemetry re-creations are half that of the simulations. Thus, the effects of inter-robot repulsion are visually exaggerated. Second, the perceptual process for obstacle detection (a ring of ultrasonic sensors) is not sophisticated enough to ignore robots: robots are detected as robots and as obstacles. The repulsion between them is further exaggerated. This problem will be resolved as better omnidirectional sensors and perceptual processes are incorporated into our research.

10. Summary and Conclusions

The impact of communication on performance in reactive multiagent robotic systems has been investigated through extensive simulation studies. Performance results for three generic tasks illustrate how task and environment can affect communication payoffs. Initial results from testing on mobile robots are shown to support the simulation studies.

The principal results for these tasks are:

- Communication improves performance significantly in tasks with little environmental communication.
- Communication is not essential in tasks which include implicit communication.
- More complex communication strategies offer little or no benefit over low-level communication.

More detailed conclusions appear in Sections 7.4 and 8.4 of this article.

Future work involves three major research thrusts. The first is concerned with societal performance in fault-tolerant multiagent robotic systems; where unreliable communication may be present and the robotic agents have the potential for failure. The second research thrust involves integrating humans more effectively with the control of a society through teleoperation. The last area includes developing novel methods for formalizing and expressing multiagent robotic systems with the goals of producing tools which will facilitate their use and to establish formally provable properties (i.e., necessary and sufficient conditions) regarding their specifications.

Appendix : Motor Schema Formulae

This appendix contains the methods by which each of the individual primitive schemas used in this research compute their component vectors. The results of all active schemas are summed and normalized prior to transmission to the robot for execution.

- **Move-to-goal**: Attract to goal with variable gain. Set high when heading for a goal.

  $V_{magnitude}$ = adjustable gain value

  $V_{direction}$ = in direction towards perceived goal

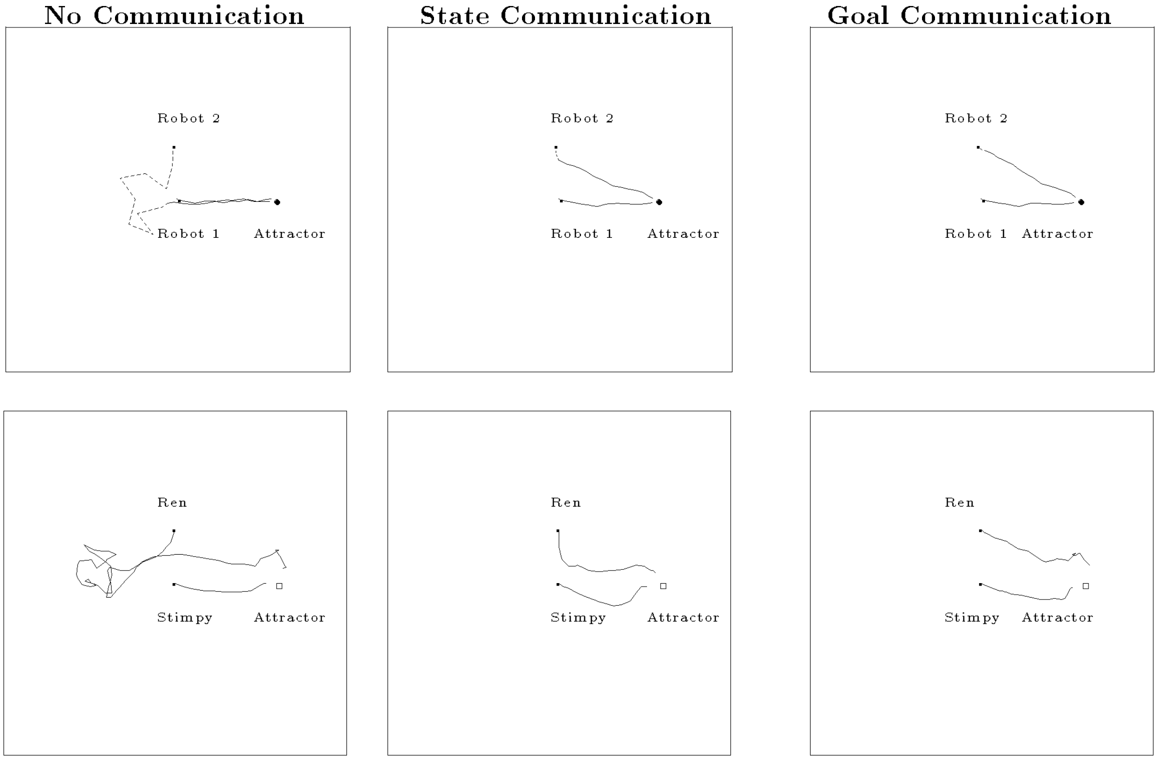

Fig. 17. Comparison of simulated *Consume* task runs (top row) with runs on mobile robots (bottom row).

- **Avoid-static-obstacle**: Repel from object with variable gain and sphere of influence. Used for collision avoidance.
  $O_{magnitude} =$
  $$0 \quad for \quad d > S$$
  $$\frac{S-d}{S-R} * G for \quad R < d \leq S$$
  $$\infty \quad for \quad d \leq R$$
  where:
  S = Adjustable Sphere of Influence (radial extent of force from the center of the obstacle)
  R = Radius of obstacle
  G = Adjustable Gain
  d = Distance of robot to center of obstacle
  $O_{direction}$ = along a line from robot to center of obstacle moving away from obstacle

- **Noise**: Random wander with variable gain and persistence. Used to overcome local maxima, minima, cycles, and for exploration.
  $N_{magnitude}$ = Adjustable gain value
  $N_{direction}$ = Random direction that persists for $N_{persistence}$ steps
  ($N_{persistence}$ is adjustable)

- **Probe**: Used in *Graze* for favoring continued motion in the current directional heading.
  $V_{magnitude}$ = adjustable gain value or 0 if no ungrazed areas detected
  $V_{direction}$ = Straight ahead along an extrapolated path from the current location only if grazed area ahead. Direction not important if no ungrazed area ahead as gain is 0.

Acknowledgements

This research has been supported by the National Science Foundation under grant #IRI-9100149 and ONR/ARPA under grant #N00014-94-1-0215. The Mobile Robot Laboratory is supported by additional grants from NSF and the Westinghouse-Savannah River Technology Center. The authors would like to thank Doug MacKenzie and Russ Clark for their development of software that has been useful for this project.

Notes

1. This task was described earlier in [9]. The "forage" state mentioned there corresponds to the "wander" state here.
2. **Avoid-static-obstacle** is also used for non-threatening moving objects. Other schemas such as **escape** and **dodge** can be used for non-cooperative moving objects when appropriate.
3. The differences are in three areas: 1) How test scenarios are generated, 2) What happens when robots fail to complete the task, and 3) Restrictions on robot movement. In the new simulator, obstacles are not allowed to overlap one another. Previously, this was allowed, resulting in a less accurate accounting of obstacle coverage. For this research robots are initially placed in the center of the environment, at home base. In earlier research they were placed randomly about the environment. The authors believe a single starting location for all robots is more likely in real world implementations. The previous simulator allowed runs not to exceed a maximum of 2000 steps. If a run exceeded this time limit it was halted and discarded. Here the limit is raised to 8000 steps and runs that timeout are counted as taking 8000 time steps. In the new simulator, robots are not allowed to move outside the visual boundaries of the environment, as was previously the case.
4. For *Graze*, the percent of area to be grazed is varied in increments of 13.57%. This allows the difficulty to be varied in seven discrete steps from 13.57% to 95%. Results can be directly compared to *Forage* and *Consume* tasks with one to seven attractors.

References

1. Altenburg, K. and Pavicic, M., 1993. Initial Results of the Use of Inter-Robot Communication for a Multiple, Mobile Robotic System, *Working Notes of the Workshop on Dynamically Interacting Robots at IJCAI-93*, pp. 95-100.
2. Altmann, S. 1974. Baboons, Space, Time, and Energy. *American Zoology*, 14:221-248.
3. Arkin, R.C., 1989. Motor Schema Based Mobile Robot Navigation, *International Journal of Robotics Research*, vol 8(4), pp. 92-112.
4. Arkin, R.C., 1990. The Impact of Cybernetics on the Design of a Mobile Robot System: A Case Study. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 20, No. 6, pp. 1245-1257.
5. Arkin, R.C., 1992. Cooperation without Communication: Multi-agent Schema Based Robot Navigation, *Journal of Robotic Systems*, Vol. 9(3), pp. 351-364.
6. Arkin, R.C., 1992. Integrating Behavioral, Perceptual, and World Knowledge in Reactive Navigation. *Designing Autonomous Agents*, ed. P. Maes, Bradford-MIT Press, pp. 105-122.
7. Arkin, R.C., 1992. Modeling Neural Function at the Schema Level: Implications and Results for Robotic Control. *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, ed. R. Beer, R. Ritzmann, and T. McKenna, Academic Press, pp. 383-410.
8. Arkin, R.C., 1993. Survivable Robotic Systems: Reactive and Homeostatic Control. *Robotics and Remote Systems for Hazardous Environments*, ed. M. Jamshidi and P. Eicker, Prentice-Hall, pp. 135-154.
9. Arkin, R.C., Balch, T., Nitz, E., 1993. Communication of Behavioral State in Multi-agent Retrieval Tasks, *Proc. 1993 IEEE International Conference on Robotics and Automation*, Atlanta, GA, vol. 1, p. 678.
10. Arkin, R.C. and Hobbs, J.D., 1992, Dimensions of Communication and Social Organization in Multi-Agent Robotic Systems, *From animals to animats 2: Proc. 2nd International Conference on Simulation of Adaptive Behavior*, Honolulu, HI, Dec. 1992, MIT Press, pp. 486-493.
11. Arkin, R.C. and MacKenzie, D., 1994. "Temporal Coordination of Perceptual Algorithms for Mobile Robot Navigation", to appear in *IEEE Transactions on Robotics and Automation*.
12. Arkin, R.C., Murphy, R.R., Pearson, M. and Vaughn, D., 1989. Mobile Robot Docking Operations in a Manufacturing Environment: Progress in Visual Perceptual Strategies, *Proc. IEEE International Workshop on Intelligent Robots and Systems '89*, Tsukuba, Japan, pp. 147-154.
13. Arkin, R.C., *et. al.* , 1993. Buzz: An Instantiation of a Schema-Based Reactive Robotic System, *Proc. International Conference on Intelligent Autonomous Systems: IAS-3*, Pittsburgh, PA., pp. 418-427.
14. Balch, T., Arkin, R.C., 1993. Avoiding the Past: A Simple but Effective Strategy for Reactive Navigation, *Proc. 1993 IEEE International Conference on Robotics and Automation*, Atlanta, GA, vol. 1, pp. 678-685.
15. Brooks, R.A., 1986. A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, RA-2, No. 1, p. 14, 1986.
16. Brooks, R., Maes, P., Mataric, M., and More, G., 1990. Lunar Base Construction Robots, *IEEE International Workshop on Intelligent Robots and Systems (IROS '90)*, Tsuchiura, Japan. pp. 389-392.
17. Clark, R.J., Arkin, R.C., and Ram, A., 1992. Learning Momentum: On-line Performance Enhancement for Reactive Systems. *Proc. 1992 IEEE International Conference on Robotics and Automation*, Nice, France, May 1992, pp. 111-116.
18. Croy, M. and Hughes, R., 1991. Effects of Food Supply, Hunger, Danger, and Competition on choice of Foraging Location by the fifteen-spined stickleback. *Animal Behavior*, 1991, Vol. 42, pp. 131-139.
19. Dudek, G., Jenkin, M., Milios, E., and Wilkes, D., 1993. A Taxonomy for Swarm Robots. *Proc. 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Yokohama, Japan, pp. 441-447.
20. Drogoul, A. and Ferber, J., 1992. From Tom Thumb to the Dockers: Some Experiments with Foraging

Robots. *From Animals to Animats: Proc. 2nd International Conference on the Simulation of Adaptive Behavior*, MIT Press/Bradford Books, Honolulu, HI, pp. 451-459.

21. Floreano, D., 1993. Emergence of Nest-based Foraging Strategies in Ecosystems of Neural Networks. *From Animals to Animats: Proc. 2nd International Conference on the Simulation of Adaptive Behavior*, MIT Press/Bradford Books, Honolulu, HI, pp. 410-416.
22. Franklin, R.F. and Harmon, L.A., 1987. Elements of Cooperative Behavior. Internal Research and Development Final Report 655404-1-F, Environmental Research Institute of Michigan (ERIM), Ann Arbor, MI.
23. Franks, N., 1986. Teams in Social Insects: Group Retrieval of pre by army ants. *Behav. Ecol. Sociobiol.*, 18:425-429.
24. Fukuda, T., Nakagawa, S., Kawauchi, Y., and Buss, M., 1989. Structure Decision for Self Organising Robots Based on Cell Structures - CEBOT. *IEEE International Conference on Robotics and Automation*, Scottsdale Arizona, pp. 695-700.
25. Goetsch, W., 1957. *The Ants*. University of Michigan Press.
26. Goss, S., Beckers, R., Deneubourg, J., Aron, S, and Pasteels, J., 1990. How Trail Laying and Trail Following can Solve Foraging Problems for Ant Colonies. *Behavioral Mechanisms of Food Selection*, ed. R.N. Hughes, Nato ASI Series, Vol. G. 20, Springer-Verlag, Berlin, pp. 661-678.
27. Hackwood, S. and Beni, S., 1992. Self-organization of Sensors for Swarm Intelligence, *1992 IEEE International Conference on Robotics and Automation*, Nice, pp. 819-829.
28. Helmbold, D. and McDowell, C., 1990. Modelling Speedup(n) Greater than n, *IEEE Transactions on Parallel and Distributed Systems*, vol 1(2), pp. 250-256.
29. Holldobler, B., Wilson, E., 1990. *The Ants*, Belknap Press, Cambridge Mass.
30. Kaelbling, L. and Rosechein, S., 1990. "Action and Planning in Embedded Agents", in *Designing Autonomous Agents*, Maes, P. (ed), MIT Press, pp. 35-48.
31. Kaufmann, J. 1974. Social Ethology of the Whiptail Wallaby, Macropus Parryi, in Northeastern New South Wales. *Animal Behavior*. 22:281-369.
32. Khatib, O., 1985. "Real-time Obstacle Avoidance for Manipulators and Mobile Robots", *Proc. IEEE Int. Conf. Robotics and Automation*, St. Louis, p. 500.
33. Krogh, B., 1984. A Generalized Potential Field Approach to Obstacle Avoidance Control, *SME - RI Technical Paper* MS84-484.
34. Lee, J., Huber, M., Durfee, E., and Kenny, P. 1994. UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications. To appear *AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space (CIRFFSS '94)*.
35. MacKenzie, D. and Arkin, R.C., 1992. Autonomous Helicopter Position Determination Using an On-Board Integrated Vision System, *Proc. SME Applied Vision Conference '92*, Atlanta, GA.
36. MacKenzie, D. and Arkin, R.C., 1993. Formal Specification for Behavior-Based Mobile Robots, *Proc. SPIE Conference on Mobile Robots VIII*, Boston, MA, pp. 94-104.
37. MacKenzie, D.C. and Balch. T.R., 1993. Making a Clean Sweep: Behavior Based Vacuuming. *Working Notes of 1993 AAAI Fall Symposium: Instantiating Real-World Agents*. Raleigh, N.C.
38. MacLennan, B., 1991. Synthetic Ethology: An Approach to the Study of Communication. In *Artificial Life II*, SFI Studies in the Sciences of Complexity, vol. XI, ed. Farmer et al, Addison-Wesley.
39. Maes, P., 1991. Situated Agents can have Goals, in *Designing Autonomous Agents*, Maes, P. (ed), MIT Press, pp. 49-70.
40. Mataric, M., 1992. Minimizing Complexity in Controlling a Mobile Robot Population. *Proc. IEEE International Conference on Robotics and Automation*, Nice, FR, May 1992.
41. Miller, D., 1990. Multiple Behavior-Controlled Micro-Robots for Planetary Surface Missions. *Proc. 1990 IEEE International Conference on Systems, Man, and Cybernetics*, Los Angeles, CA, November 1990, pp. 289-292.
42. Moynihan, M. 1970. Control, Suppression, Decay, Disappearance and Replacement of Displays. *Journal of Theoretical Biology*, 29:85-112.
43. Noreils, F.R., 1992. "Battlefield Strategies and Coordination between Mobile Robots", *Proc. 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1777-1784.
44. Noreils, F. 1993. Coordinated Protocols: An Approach to Formalize Coordination between Mobile Robots. *Proc. 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 717-725.
45. Parker, L., 1993. Adaptive Action Selection for Cooperative Agent Teams. *From Animals to Animats: Proc. 2nd International Conference on the Simulation of Adaptive Behavior*, MIT Press/Bradford Books, Honolulu, HI, pp. 442-450.
46. Payton, D.W., 1991. Internalized Plans: A Representation for Action Resources, in *Designing Autonomous Agents*, Maes, P. (ed), MIT Press, pp. 89-103.
47. Pearce, M., Arkin, R.C., and Ram, A., 1992. The Learning of Reactive Control Parameters through Genetic Algorithms, *Proc. 1992 International Conference on Intelligent Robotics and Systems (IROS)*, Raleigh, N.C., pp. 130-137.
48. Ram, A., Arkin, R.C., Moorman, K., and Clark, R., 1992. Case-based Reactive Navigation: A case-based method for on-line selection and adaptation of reactive control parameters in autonomous robotic systems, Technical Report GIT-CC-92/57, College of Computing, Georgia Tech.

49. Slack, M.G., 1990. Situationally Driven Local Navigation for Mobile Robots, *JPL Publication 90-17*, Jet Propulsion Laboratory, Pasadena, CA.
50. Tinbergen, 1966. *Social Behavior in Animals*, Methuen & Co., London.
51. Wang, J., 1992. "Distributed Mutual Exclusion based on Dynamic Costs", *Proc. IEEE International Symposium on Intelligent Control*, Glasgow, UK, pp. 109-15.
52. Werner, G. and Dyer, M., 1990. Evolution of Communication in Artificial Organisms. *Technical Report UCLA-AI-90-06*, AI Laboratory, University of California, Los Angeles.
53. Yanco, H. and Stein, L. 1993. An Adaptive Communication Protocol for Cooperating Mobile Robots. *From Animals to Animats: Proc. 2nd International Conference on the Simulation of Adaptive Behavior*, MIT Press/Bradford Books, Honolulu, HI, pp. 478-485.

Tucker Balch was born in Miami, Florida in 1962. He received the B.S. Degree from Georgia Tech in 1984 and the M.S. Degree from U.C. Davis in 1988. He is currently pursuing a Ph.D. in Autonomous Robotics at Georgia Tech. From 1984 to 1988 he supported research at the Lawrence Livermore National Laboratory as a computer scientist. He entered the Air Force as a Pilot Candidate in 1988 and completed fighter training in 1991. He now flies F-15 Eagles at the Georgia Air National Guard. His research interests include integration of deliberative planning and reactive control, communication in multi-robot societies, and parallel algorithms for robot navigation. He is a Student Member of the IEEE and ACM.

Ronald C. Arkin is Associate Professor and Director of the Mobile Robot Laboratory in the College of Computing at the Georgia Institute of Technology. Dr. Arkin's research interests include reactive control and action-oriented perception for the navigation of mobile robots, robot survivability, multi-agent robotic systems, learning in autonomous systems, and cognitive models of perception. He has over 80 technical publications in these areas. Funding sources have included the National Science Foundation, ARPA, Savannah River Technology Center, and the Office of Naval Research. Dr. Arkin is an Associate Editor for IEEE Expert and the Journal of Environmentally Conscious Manufacturing, and a Member of the Editorial Boards of Autonomous Robots and the Journal of Applied Intelligence. He is a Senior Member of the IEEE, and a member of AAAI and ACM.