In "Robot Teams: From Diversity to Polymorphism", T. Balch and L. E. Parker, eds., A K Peters Ltd, April 2002

# Design and Evaluation of Robust Behavior-Based Controllers for Distributed Multi-Robot Collection Tasks

Dani Goldberg and Maja J Matarić
Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
{dani,mataric}@usc.edu

**Abstract**

In this chapter, we demonstrate the effectiveness of behavior-based control in facilitating the development and evaluation of multi-robot controllers that are: (1) robust to robot failures, and (2) easily modified to facilitate development of the controller variation that sufficiently satisfies the design requirements for the task. Our experimental focus here is *distributed multi-robot collection*, a class of tasks that includes de-mining and toxic waste clean-up. We demonstrate a basic, *homogeneous* multi-robot controller for the collection task, then show how to easily derive two *heterogeneous*, spatio-temporal variations with markedly different performance properties. We evaluate the desirability of these controllers with respect to design requirements involving inter-robot interference, time-to-completion, and energy expenditure. The data for evaluation come from experiments using four physical mobile robots performing the three variations of the collection task.

## 1 Introduction

Designing and implementing robust controllers for multiple interacting mobile robots is considered something of a black art, often involving a great deal of reprogramming and parameter adjustment. It is difficult enough to develop a multi-robot controller that functions only under the ideal conditions of little noise and no robot failures. The fact that such ideal conditions do not often exist, even in a laboratory setting, places certain practical requirements on the multi-robot controller. In particular, the controller must exhibit group-level robustness to noise and robot failures. This is especially important when physical human intervention is difficult (e.g., a toxic waste spill) or impossible (e.g., an extraterrestrial mission).

Additional design requirements for the controller arise from the fundamental, constrained resources of the system, including energy, time, and the number of robots. Untethered mobile robots are generally powered by batteries and can only perform a limited amount of work before needing recharging. Minimizing energy utilization is thus often required in domains, such as space exploration, where recharging is expensive, difficult, or time consuming. In time-critical domains, such as search and rescue, the requirement is for expedient execution of the task. Additionally, regardless of the domain, the fragility of the robots may require the controller to maintain both robot-object and inter-robot collisions at a minimum.

For a given task environment and set of robots, the requirements for the controller may not be independent but instead arise as tradeoffs. For example, minimizing both time and inter-robot collisions may not be possible since faster moving robots are less likely to properly sense each other and thus more likely to collide. Different controller variations may have to be tested and compared in order to select one that sufficiently satisfies the requirements given the tradeoffs among them. This places an additional requirement on the controller, namely that it be easily modifiable. The testing and comparison of the variations could potentially be accomplished analytically if an adequate model of the system were developed (a significant challenge in itself), or in simulation (potentially less difficult). In either case, the desire to be able to easily modify the controller remains. Our assumption in this work is that neither an adequate (i.e., very high fidelity) model nor simulation of the physical multi-robot collection task need exist, and thus we performed all tests directly on physical robots.

The controllers we present in this chapter are designed to address the requirments above. Specifically, they exhibit group-level robustness to robot failures and noise, and are easily modified. Our focus is on the domain of distributed multi-robot collection (foraging) tasks, including toxic waste clean-up and demining. We present a basic *homogeneous* controller for the collection task in which all of the robots have identical behavioral repetoirs and work concurrently. We then derive two *heterogeneous* variations, *pack* and *caste*, which respectively modify the robots' temporal and spatial interactions. Finally, we evaluate and compare the performance of the controllers using three spatio-temporal criteria: inter-robot collisions, distance traveled by each robot, and time-to-completion for the task. The latter two criteria also provide an indication of the energy expenditure of the robots. The data for evaluation come from experiments we conducted using four physical mobile robots performing the three variations of the collection task.

After a review of related work in Section 2, Section 3 describes the structure of the collection task as well as the group of physical mobile robots that performed it. Section 4 then presents the details of the homogeneous controller including the behaviors it contains and how it achieves robustness. Section 5 considers spatio-temporal interactions between robots, especially physical interference, and motivates the two interference-modifying heterogeneous controller versions, pack and caste, presented in Sections 6 and 7. Section 8 presents an analysis of the controllers using data from physical experiments, and provides a comparative evaluation. Finally, conclusions are presented in Section 9.

## 2 Related Work

This section gives a partial review of some of the most related robotics work. The reader is encouraged to see Cao et al. (1997) and Matarić (1995) for a more complete set of references from Artificial Intelligence, Robotics, Distributed AI, and Artificial Life.

Much research has been conducted on the performance and properties of robot collection and foraging tasks. Arkin et al. (1993) demonstrate simulation work studying the issues of density and critical mass in a collection task using fully homogeneous robots. Density and critical mass of robots has significant effect on the manifested physical interference between robots. Arkin and Hobbs (1993) describe the general schema-based control architecture, which bears some fundamental similarities to behavior-based control, which we used, and give the critical mass experiments. Arkin and Ali (1994) present a series of simulation results on related spatial tasks such as foraging, grazing, and herding. This work is similar to the homogeneous controller implementation presented in Section 4.

Matarić (1992b) describes a similar behavior-based approach for minimizing complexity in controlling a collection of robots performing various behaviors including following, aggregation, dispersion, homing, flocking, and foraging (similar to our collection task). The work also includes a simulated dominance hierarchy based on IDs and used to evaluate performance of homogeneous versus ordered aggregation and dispersion behaviors. Our pack controller (Section 6) also utilizes a dominance hierarchy based on robot ID. In other work, we have demonstrated that allowing dynamic reorganization of such a dominance hierarchy can improve group performance (Goldberg & Matarić, 1999b). Fontán and Matarić (1998) also present work on multi-robot collection, but focused on issues of critical mass in territorial task division, corresponding to an extreme case of the caste controller we present in Section 7. Goldsmith et al. (1998) also present a caste-like strategy for multi-robot search, but with each robot able to dynamically switch its team and function. Parker (1992) and Parker (1994) describe multi-robot experiments on foraging R2e robots with *a priori* hard-wired heterogeneous capabilities using the Alliance architecture. Parker (1994) describes a temporal division that sends one robot to survey and measure the environment for toxic spills, then has the rest of the group use its information to clean up the spill. This two group division is similar to our caste controller, though ours uses a spatial division rather than a temporal one.

Tan and Lewis (1996) describe an approach to maintaining a geometric configuration of a robot group using virtual structures, tested on a group ISR R3 mobile platforms, a later generation of our R2e robots. Similar to our homogeneous implementation, this work also exhibits spatial and temporal homogeneity, though the coupling here is tighter. Beckers et al. (1994) describe a group of five robots without external sensing or communication effectively clustering pucks through a careful combination of the mechanical design of the robots' puck scoops and the simple controller that moves them forward and in reverse. This work demonstrates a homogeneous controller performing a task similar to our collection task, but where the goal location is not pre-specified, instead emerging during execution. Holland and Melhuish (2000) present more recent results from an expanded study with essentially the same experimental scenario.

Other work on multi-robot collection is inspired by trail formation in ants (Drogoul & Ferber, 1992). Werger and Matarić (1996) describe a foraging robot chain that is constructed and modified using only contact sensing for communication. Vaughan et al. (2000) present multi-robot ant-like foraging in a simulated environment where efficient foraging trails are dynamically constructed using a mechanism analogous to ant pheromones.

Apart from the collection tasks, behavior-based control has been used in many other applications ranging from multi-robot soccer (Lund & Pagliarini, 2000) and service robotics (Lindström et al., 2000), to control of underwater robots (Rosenblatt et al., 2000) and ape-like robots (Hasegawa et al., 2000). In all of these behavior-based systems, there is some action selection mechanism that produces a coherent, global behavior. The work described in this chapter uses behavior arbitration in which some (possibly small) subset of the behaviors control the motors at any time. Pirjanian (1998) describes a number of action selection mechanisms. Pirjanian et al. (1998) present a voting-based action selection mechanism which is extended to multi-robot coordination in Pirjanian and Matarić (2000).

Unlike the work in this chapter, our other work involving robot collection focuses not on controller design, but rather on endowing the robots with the ability to adapt to their changing environment. The robots employ augmented Markov models (Goldberg & Matarić, 1999a) to detect significant shifts in object densities (Goldberg & Matarić, 2000a), or estimate the state of their environment in order to maximize performance of the collection task (Goldberg & Matarić, 2000b). The following section describes the collection task in detail.

# 3 The Collection Task

The controllers we present implement versions of a multi-robot collection (foraging) task, a prototype for various applications including distributed solutions to de-mining, toxic waste clean-up, and terrain mapping. We present the general structure of the collection task, our multi-robot testbed, and then the controllers.

## 3.1 Task Structure

We define the collection task as a two-step repetitive process in which:

1. $N$ ($N \geq 1$) robots search designated regions of space for certain objects, and
2. once found, these objects are brought to a goal region using some form of navigation.

A region in the task is any contiguous, bounded space (in the case of mobile robots, a planar surface) which the robots are capable of moving across. There are three mutually-exclusive, non-overlapping types of regions:

- search regions, $S$, containing $P$ objects, a fraction of which must be delivered to a goal region;
- goal regions, $G$, where objects are delivered;
- and, optionally, empty regions, $E$, that contain no objects and are not goal regions.

The only restrictions placed on the configuration of the regions for the collection task are that there be at least one search and one goal region, and the union of all the regions be contiguous. Figure 1 gives two examples of possible region configurations for the collection task.

The specific configuration we used is shown in Figure 2. The experiments were performed in an $11 \times 14$ foot rectangular enclosure (the Corrall). The search region, $S$, is approximately 126 square feet and has $P = 27$ small metal cylinders (pucks) evenly distributed throughout. The goal region $G$, also called *Home*, is a ninety degree sector of a circle with a radius of 2 feet, located in one corner of the Corrall. Finally, there is a 25 square foot empty region, $E$, separating the search and goal regions. $E$ is composed of the

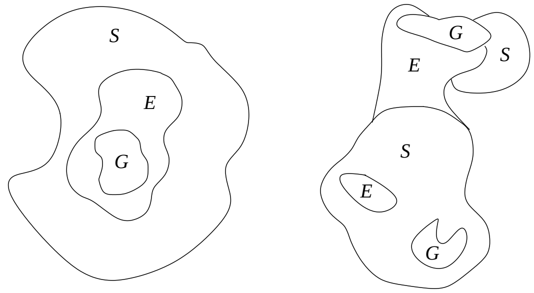


Figure 1: Two example region configurations for the collection task.

Boundary and Buffer zones, whose functions will be described in the next section. $N = 4$ robots are used in the experiments.

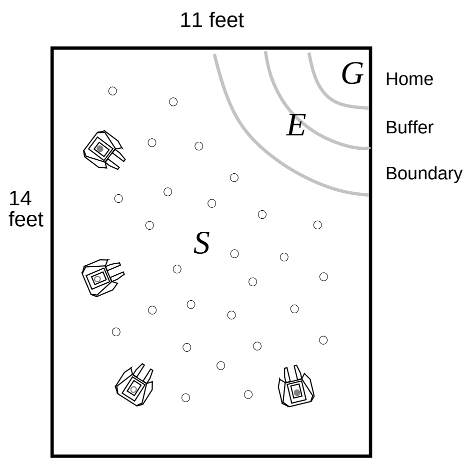


Figure 2: Actual configuration used in the collection task.

## 3.2 The Robots

Four IS Robotics R2e robots were used (Figure 3). Each is a differentially-steered base equipped with two drive motors and a two-fingered gripper. The sensing capabilities of each robot include piezo-electric contact (bump) sensors around the base and in the gripper, five infrared (IR) sensors around the chasis and one on each finger for proximity detection, a color sensor in the gripper, a radio transmitter/receiver for communication and data gathering, and an ultrasound/radio triangulation system for positioning (Figure 4). The robots are programmed in the Behavior Language (Brooks, 1990), a parallel, asynchronous, behavior-based programming language inspired by the Subsumption Architecture (Brooks, 1986). The main computational power on each robot is a single Motorola 68332 16-bit microcontroller running at 16 MHz. Even though computationally impoverished by today's standards, the processing capabilities have proven to be adequate for most tasks we have envisioned, helping to show that robust, effective control need not be computationally expensive. Perhaps the greatest drawback of the 68332 is its lack of floating point computation, which, for example, influences our calculation of heading, described in the following section.

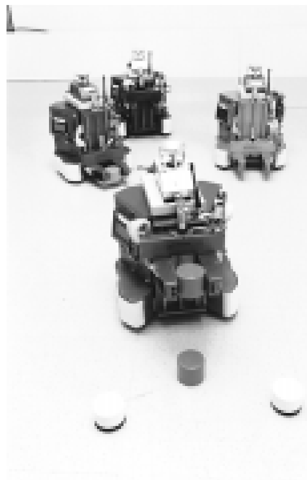

Figure 3: The four R2e robots used in the experiments.

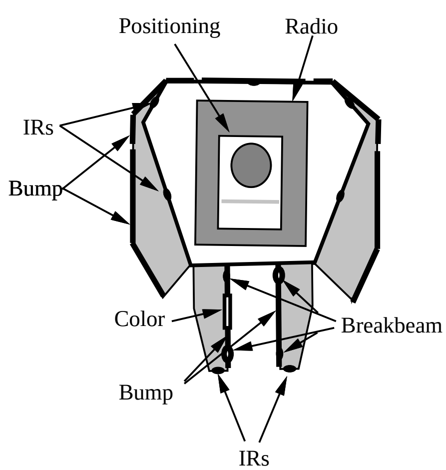


Figure 4: The sensor configuration of an R2e robot.

### 3.3 Behavior-Based Control

The work presented in this chapter is couched in the framework of distributed behavior-based control (Brooks, 1991; Matarić, 1992a). Behavior-based control has proven to be an effective paradigm for developing single-robot and multi-robot controllers (Arkin, 1998). In behavior-based control, the robot controller is organized as a collection of modules, called behaviors, that receive inputs from sensors and/or other behaviors, process the input, and send outputs to actuators and/or other behaviors. Each behavior generally serves some independent function, such as *avoiding* obstacles or *homing* to a goal location. All behaviors in a controller are executed in parallel, simultaneously receiving inputs and producing outputs. An action selection mechanism prevents conflicts when multiple outputs are sent to actuators or other behaviors (Pirjanian, 1998). The controllers presented in this chapter demonstrate the suitability of the behavior-based paradigm for designing robust and modifiable multi-robot controllers.

In the next section, we present our initial, *homogeneous* controller for the collection task, followed later by two heterogeneous variations, *pack* and *caste*.

# 4 The Homogeneous Controller

In this section, we present the first behavior-based controller which implements a homogeneous version of the collection task where the robots' behavioral repetoirs are identical, and the robots act concurrently and independently.

The overall structure of the controller is presented in Figure 5. In the figure, the rounded rectangles represent the robot's sensors, with sensor values being transmitted to behaviors along the dotted lines. The behaviors themselves are drawn as ellipses with text in one of three font styles: italics for behaviors that only receive sensor inputs; bold for behaviors that send actuator outputs; and bold-italics for behaviors that do both. The dashed lines represent commands sent by behaviors to the actuators (rectangles), and the solid lines represent control signals sent between behaviors. These control signals include: inhibition signals that temporarily disable behaviors, or do so permanently until the inhibition is lifted; information about the state of the behaviors; and signals indicating that a behavior should perform a certain action. These control signals establish the hierarchy of actuator commands shown at the right of the diagram. The $\otimes$ represents behavior selection and indicates that only one of relevant actuator command pathways is active at any time. The $\odot$ represents a Subsumption-style priority scheme with the actuator command coming from above taking precedence (Brooks, 1986). The hierarchy of command pathways in the diagram helps illustrate that behavior arbitration is the action selection mechanism for the controller. The next section presents in detail the function of the each behavior in the controller, and the structure of the inter-behavior command pathways. The subsequent section discusses the group-level robustness achieved by this controller.

### 4.1 Behaviors

In order to provide a clear picture of the interaction between behaviors, we describe the individual behaviors of the controller in an order that mirrors the progression of the task as the robot performs it. The following twelve behaviors constitute the collection task:
1) *avoiding*: This behavior avoids any object (including other robots) detected by the IR sensors and deemed to be in the path of, or about to collide with, the robot. If the robot has already collided with an object, as detected by the contact sensors, it steers away from it. This behavior is critical to the safety of the robot and therefore takes precedence over most of the behaviors that control the drive motors (*puck detecting*, *wandering*, *homing*, *reverse homing*).
2) *wandering*: The robot moves forward and, at random intervals, turns left or right through some random arc. Using this behavior, the robot searches a region for pucks.
3) *puck detecting*: If an object is detected by the front IR sensors while *wandering*, this behavior, by lifting the gripper, determines whether the object is short enough to be a puck, or whether it is an obstacle that must be avoided. If it is a puck, the robot carefully approaches the object and attempts to place it between its fingers. Otherwise, the robot performs *avoiding*.

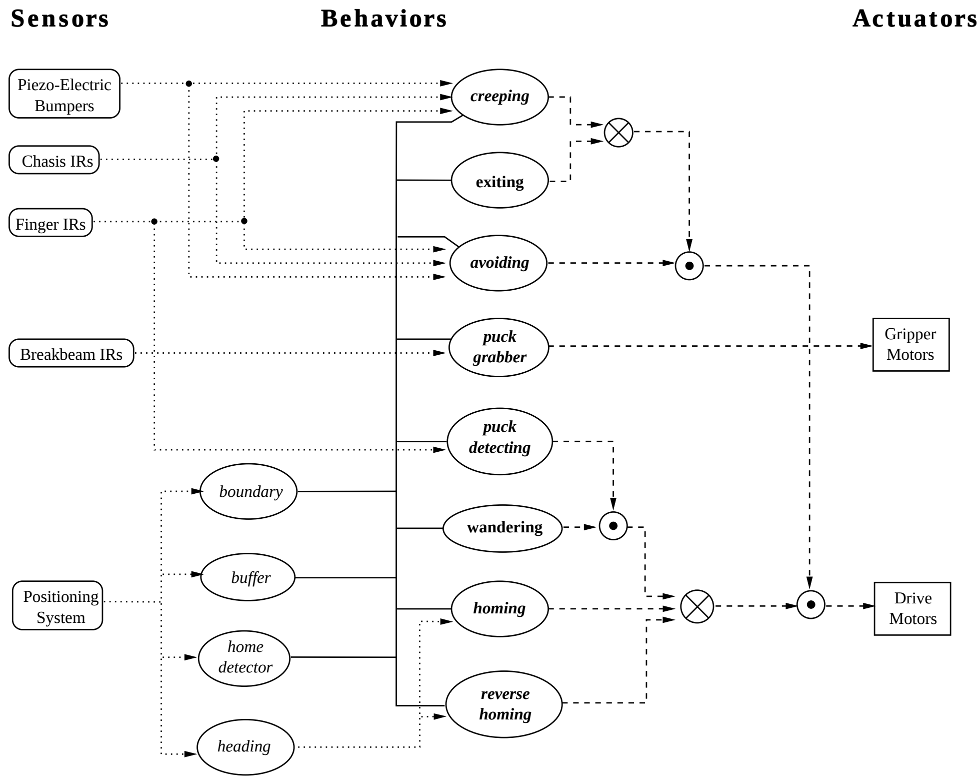


Figure 5: The homogeneous controller for the collection task. Rounded rectangles represent the robot's sensors, ellipses represent behaviors, and rectangles represent actuators. Sensor values are transmitted along dotted lines, actuator commands along dashed lines, and inter-behavior control signals along solid lines. The $\otimes$ represents behavior selection and the $\odot$ represents Subsumption-style precedence.

4) *puck grabber*: When a puck enters the fingers and is detected by the breakbeam IR sensors, this behavior grasps it and raises the fingers. Raising the fingers above puck height prevents the robot from unnecessarily avoiding pucks while *homing*, and allows the robot to collect up to about four additional pucks with its base.
5) *homing*: If carrying a puck, the robot moves towards the designated goal location, Home. While *homing*, *avoiding* can take precedence in order to avoid obstacles.
6) *boundary*: This behavior monitors how the robot enters the Boundary region. If the robot enters this region without a puck, it returns it to the search region using *reverse homing*. If carrying a puck, the robot is allowed to enter this region and proceed towards Home (see Figure 2). This behavior prevents the robot from collecting pucks that have already been delivered.
7) *buffer*: This behavior monitors entry into the Buffer region. Entering this region triggers the activation of the *creeping* behavior.
8) *creeping*: This behavior is a refined combination of the *homing* and *avoiding* behaviors designed to carefully bring the robot to the very corner of the Corrall where Home is located and where the pucks must be delivered. Under *creeping*, the robot moves more slowly and uses its IR sensors at a closer range appropriate for working within the corner. The standard versions of *homing* and *avoiding* would conflict in a confined corner situation, since *avoiding* would perceive the goal corner as an obstacle and attempt to move the robot away from it. *Creeping* takes precedence over *avoiding* since it already incorporates a version of this behavior.
9) *home detector*: A monitoring behavior for entry into the Home region. Upon entering this region, *home detector* sends a signal to *puck grabber* to release the puck.
10) *exiting*: Entering the Home region triggers this behavior which moves the robot several inches backwards,

then performs a 180-degree turn in place. This behavior also sends the signal that lowers the gripper. When *exiting* terminates, the robot remains within the Boundary region without a puck. This in turn triggers the *boundary* behavior to begin *reverse homing*.

11) *reverse homing*: Starting from within the Boundary region, this behavior performs the opposite of *homing*; it moves the robot out into the search region. This behavior is essentially identical to *homing* except that the goal location is set to the corner of the Corrall opposite Home. Once the Boundary region has been left, *reverse homing* becomes inactive and the robot once again begins searching for pucks using *wandering*.

12) *heading*: This behavior processes the positioning system data and provides approximate heading values for the *homing* and *reverse homing* behaviors. The positioning system supplies the robot's current $(x, y)$ position at approximately 1–2 Hz. Consecutive position values, $(x_0, y_0)$ and $(x_1, y_1)$, are used in an approximate integer-based calculation of $\arctan(\frac{y_1 - y_0}{x_1 - x_0})$ adjusted for the quadrant of the angle to provide one of sixteen possible sector headings. The accuracy of this heading calculation is usually within one sector of the true heading, but may be far worse when the robot turns in place. Frequent updates of the heading, with little reliance by the other behaviors on any one heading value, help to compensate for the inaccuracies. (An alternative is to use a physical compass for heading data. In our lab, however, the high variance in magnetic fields makes this an inviable option.)

### 4.2 Robustness

In the above described homogeneous controller, group-level robustness is a direct result of the robots behaving identically and independently. No noise-susceptible, or time-critical, radio communication that could be a source of fragility in the system is necessary. Each robot must individually manage the noise and uncertainty associated with its sensors and actuators, and the complexity of a dynamic and basically unknown environment. (Our controller, as is true for most behavior-based controllers, accommodates noise and uncertainty by tightly coupling sensing to action so that no great reliance is placed on any one sensor reading.) The partial, or even complete, failure of any one robot, or a subset of them, does not debilitate the entire group. As long as there is one functioning robot, the task will be accomplished.

As discussed previously, in addition to exhibiting group-level robustness, a multi-robot controller should be easy to modify in order to facilitate the search for a acceptable variation. The desirability of the controller must be measured with respect to any design requirements, such as time-to-completion of the task, energy consumption, or the amount of interference exhibited. Thus, before we present the variations of our homogeneous controller, we discuss the key diagnostic parameters used in evaluation. Our focus here is on inter-robot interference, specifically physical collisions between robots. The goal that motivates the modification of the homogeneous controller is minimization of such interference. The next section provides a discussion of interference and the two spatio-temporal solutions to it which provide the basis for our heterogeneous controller variations.

## 5 Spatio-Temporal Interactions

In this section, we discuss the nature of physical inter-robot interference (i.e., collisions), and how a multi-robot system may be modified to manipulate this interference. Our discussion here provides the motivation for the two controller variations, *pack* and *caste*, presented later.

Multi-robot systems are by definition physically embodied and embedded in dynamic environments. The types of interference they contain can be distinguished about a physical/non-physical dichotomy. Physical interference manifests itself most overwhelmingly in competition for space. Non-physical interference ranges from the sensory (shared radio bandwidth, crossed infrared or ultrasound sensors) to the algorithmic (the goals of one robot undoing the work of another, competing goals, etc.). Here we focus on physical interference and demonstrate that it is an effective tool for system evaluation and design.

We define the *characteristic interference* of a system at a particular point in space to be the sum, over some finite time period, of all measured interference occurring at that location (see Figure 6). The result is a surface that can be used to adjust the controller in order to reduce interference and thus modify the system's overall performance. Robot density is a critical factor in characteristic interference. Single-robot

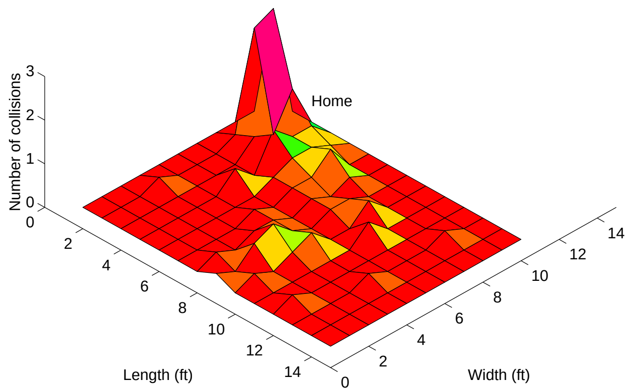


Figure 6: This plot shows the characteristic interference pattern for the homogeneous implementation of the collection task on the four physical robots. The shading corresponds to the height of the peaks, best seen in color.

systems and systems with density so high as to prevent movement produce no characteristic interference. Systems of interest lie in between the two extremes.

A principled multi-step process of controller modification can be implemented by using characteristic interference as a guide indicating where in the robots' physical interaction, and when within the lifetime of the task, behaviors should be switched and the task should be divided to modify overall task interference. Multi-robot interactions we focus on are spatio-temporal in nature and fall into four basic categories. Robots may either be in the *same place* (SP) or in *different places* (DP), both of which can occur at *same time* (ST) or at *different times* (DT), resulting in four forms of interaction: SPST, SPDT, DPST, and DPDT.

Physical interference fits into the SPST category, covering the case when two or more robots try to occupy the same location at the same time. The other three categories are useful for deriving and fine-tuning controllers that modify SPST interactions. For two of these categories, we implemented and tested a corresponding controller. The SPDT category is associated with our *pack* controller, a temporal modification to the homogeneous controller, while DPST is associated with our *caste* controller, a spatial modification of the homogeneous controller scheme. The DPDT category represents the case where there is little possibility of physical interaction. For example, the robots may occupy non-contiguous regions of space, or only one robot at a time may be activated. Since our focus is on controllers for multiple robots interacting to accomplish a task, the DPDT category does not provide an acceptable solution for interference management.

Figure 6 presents the characteristic interference pattern for the homogeneous implementation showing the number of collisions between robots within the Corrall. The data for the plot are an average of the collisions observed over five trials with the completion criterion defined as collecting 14 of the 27 pucks at Home. The figure shows high levels of interference near Home resulting from multiple robots simultaneously attempting to deliver pucks. We thus seek to modify the controller in order to reduce this interference using our two spatio-temporal variations, pack and caste. We also present a more detailed comparative evaluation of interference in the Analysis section.

# 6 The Pack Controller

In the pack controller, as in the homogeneous version, all individuals have identical behaviors and activation conditions. Unlike the homogeneous controller, however, the robots do not act concurrently and independently. Instead, a dominance hierarchy is imposed, based on some functional criterion such as the robots'

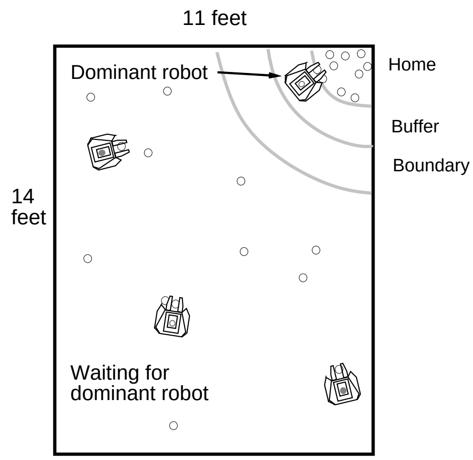


Figure 7: The pack variation of the collection task.

different capabilities, or on an arbitrary assignment scheme such as the robot ID, if the robots are functionally identical (as are ours).

The dominance hierarchy induces a temporal structure on the task by allowing only one of the robots to deliver a puck at any time. All of the robots may search for pucks in parallel, as in the homogeneous implementation, but if two or more robots simultaneously find pucks, the one highest in the hierarchy is allowed to deposit its pucks first. The other robot(s) cannot proceed until the first has finished delivering its pucks and has left the Boundary region (Figure 7). This scheme introduces temporal heterogeneity to the homogeneous version, and thus corresponds to SPDT (or temporal) arbitration of SPST interactions.

The pack strategy requires that some form of dominance hierarchy can be assigned and that dominance rank can be recognized between the robots. In our case, rank was communicated over the radios, but in other implementations it could be based on physical characteristics that can be sensed directly.

## 6.1 The "message passing" Behavior

Figure 8 presents the controller for the pack implementation. This controller is almost identical to the homogeneous controller (Figure 5), except that it includes a high-precedence *message passing* behavior. The function of *message passing* is to send the robot's status, specifically whether it is delivering a puck, to the other robots, and in turn monitor the status of the other robots. When a robot finds a puck, *message passing* places the robot into a wait state with the motors off and enters the following communications routine:

1. Wait two communication cycles (approximately 6 seconds) to accumulate the most current status information from each robot.
2. If after (1) above, no other robot is currently delivering a puck, transmit the desire to do so. Otherwise return to (1).
3. Wait three communication cycles (approximately 10 seconds) for synchronization with the other robots.
4. If after (3) above, no other robot wishes to deliver a puck, or any that do are less dominant, then proceed to deliver the puck and inform the other robots when finished. Otherwise, return to (1).

## 6.2 Robustness

As we have discussed, it is important that multi-robot controllers be robust to noise and robot failures. Similar to the homogeneous controller, the pack controller accommodates robot failures by having each

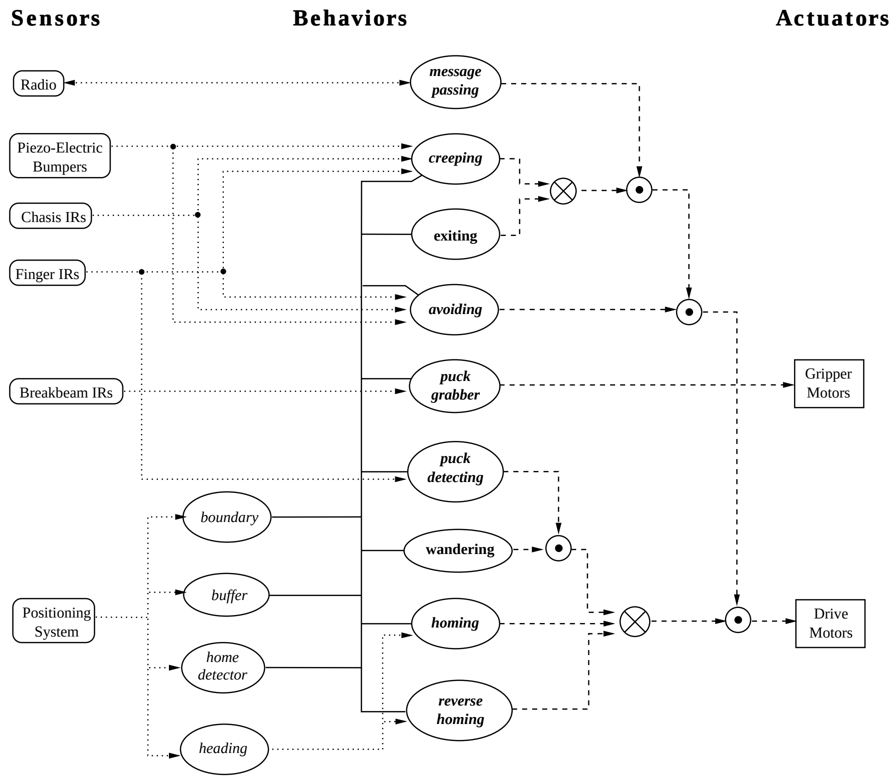


Figure 8: The pack version of the controller for the collection task.

robot able to accomplish the entire task. Unlike the homogeneous controller, the coordinated hierarchy of the pack controller requires special measures by the *message passing* behavior to ensure robustness. If a robot fails while searching for a puck, no special measures are required since no other robot is waiting upon its actions. If, however, the robot fails while delivering a puck, the other robots must be informed so as not to wait indefinitely. The failed robot can send such a message if it is able to detect the failure (a difficult problem in itself). Otherwise, some external agent, such as a human operator, can send the message.

We use a somewhat different approach in our experiments. Whenever a robot fails, it is shut down and restarted by a human operator. (In hazardous conditions, it could be possible to restart the robots remotely.) During this restart period, the other robots receive no communications from the failed robot. The robots consider such periods of protracted radio silence as an indication of the failure of the robot, and once again enter into the communications routine above. Once the failed robot has restarted and begins communicating, it is seamlessly incorporated back into the hierarchy. Since the communications routine only uses relative dominance to decide which robot should deliver a puck, it easily accommodates the attrition or addition of robots.

Another advantage of our communications routine above is that the use of "radio silence" failure detection helps provide group-level robustness to radio noise. As noise levels increase, communication between the robots becomes increasingly difficult. This may lead to protracted periods of radio silence that are incorrectly interpreted as robot failures. In such a situation, two or more robots may deliver pucks at the same time. The degradation of the hierarchy, however, is what prevents the failure of the entire group. Even if the radio system were to fail completely, the task would still be accomplished because every robot would consider every other robot as having failed. Thus, the pack controller would degenerate into the homogeneous controller. We posit that such graceful degradation in group structure, without jeopardizing the entire task, is an important property of controllers for unknown and dynamic environments.

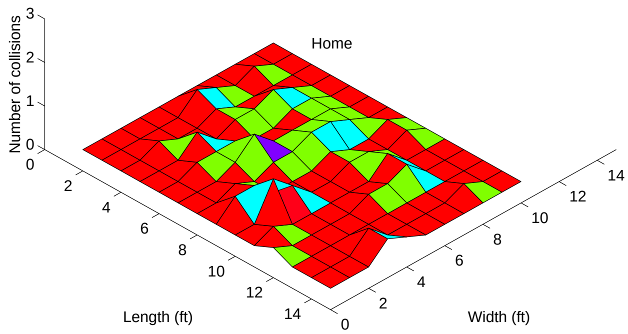


Figure 9: This plot shows the characteristic interference pattern for the pack implementation of the collection task on the four physical robots. The shading corresponds to the height of the peaks.

### 6.3 Interference

Figure 9 shows the characteristic interference pattern for the pack controller, averaged over 5 trials. The completion criterion was identical to the homogeneous case: delivering 14 of the 27 pucks to Home. As is clear from a comparison to the characteristic interference of the homogeneous controller (Figure 6), the pack controller has reduced interference near Home, as desired.

Not only is the pack controller successful in reducing interference, it is also attractive in its ease of implementation. The pack variation is simply the homogeneous controller with the addition of the dominance hierarchy induced by the *message passing* behavior. Such ease of implementation supports our requirement that controllers be easy to modify.

## 7 The Caste Controller

In a caste controller, the group of robots differentiates into two or more sub-groups (castes), each of which acts concurrently and independently, but occupies different regions of the task space. The goal is to manipulate interference by an appropriate division of the task space, and assignment of the castes to the sub-regions. This spatial separation of castes limits physical interactions to the territorial boundaries. The caste scheme introduces spatial heterogeneity and thus corresponds to DPST arbitration of SPST interference.

Unlike the homogeneous and pack strategies, the sub-groups of robots in the caste strategy may have different behavioral repetoirs. Thus, in addition to spatial heterogeneity, a caste controller may also exhibit behavioral heterogeneity. Indeed, that is the case with the caste implementation we present in this section. It consists of two sub-groups: the “Search Caste,” comprised of three robots which find pucks and bring them near Home, and the “Goal Caste,” comprised of one robot which brings the pucks the rest of the way to Home (Figure 10). Each of the two castes has a different controller.

### 7.1 The Search Caste

In our implementation, three of the four R2e robots, comprising the “Search Caste,” have behavior sets similar to the homogeneous implementation. Each robot searches the region $S$ for pucks, but delivers them to the line separating the Boundary and Buffer zones, rather than all the way to Home. Figure 11 presents the controller for the Search Caste. It is identical to the homogeneous controller (Figure 5), except that it lacks the *creeping* behavior. This more refined combination of *homing* and *avoiding*, designed to bring the

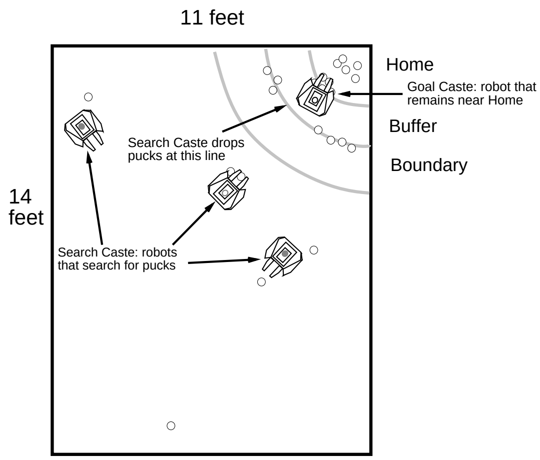


Figure 10: The caste variation of the collection task.

robots to the corner of the Corrall, is no longer necessary since pucks are not brought to the corner. The *buffer* behavior is also removed from the controller because it is not needed to activate *creeping*.

## 7.2 The Goal Caste

The "Goal Caste" consists of one robot that remains in the Home and Buffer regions with the task of transporting to Home the pucks dropped by the Search Caste at the Boundary/Buffer line. The controller for the Goal Caste is presented in Figure 13. The *sweeping* behavior moves the robot away from Home and performs an arc at the Boundary/Buffer line to "scoop up" any pucks left there (Figure 12). The *creeping* behavior then carefully moves the robot to Home, where it performs *exiting* to back up and deliver the pucks. The robot then turns in place 180 degrees to once again begin *sweeping*. During the execution of the controller, the gripper remains lifted allowing the concave front region of the robot's base to scoop up multiple pucks.

## 7.3 Robustness and Interference

The controller for the Search Caste shares many of the characteristics of the homogeneous controller. It achieves group-level robustness by maintaining a behaviorally identical group with no reliance on explicit communication. Thus, neither high levels of noise nor the failure of a robot debilitates the entire caste. The Search Caste controller also provides a good example of the ease with which the homogeneous controller can be modified.

One of the keys to robustness in the caste controller is the asynchronicity of interaction between the two castes. The Search Caste must deliver pucks to the Boundary/Buffer line, but the Goal Caste is not dependent upon them arriving at a particular time or in a particular order that may be difficult to ensure in such a complex, stochastic system.

Though not implemented in our caste controller, additional robustness could be added by using a variation of the pack communication protocol to transmit the number of active members of each caste. If one caste were to lose too many individuals, members of the other castes could replace them. For example, if the one robot of the Goal Caste were to fail, a member of the Search Caste could substitute. This scheme, while improving robustness, would require each robot to possess all of the individual caste controllers and be able to switch between them as necessary. Such caste switching would be most robust if duplication of the state of the failed robot were not necessary, as would be the case with our controller.

Figure 14 shows the average characteristic interference over five trials for the caste implementation. The

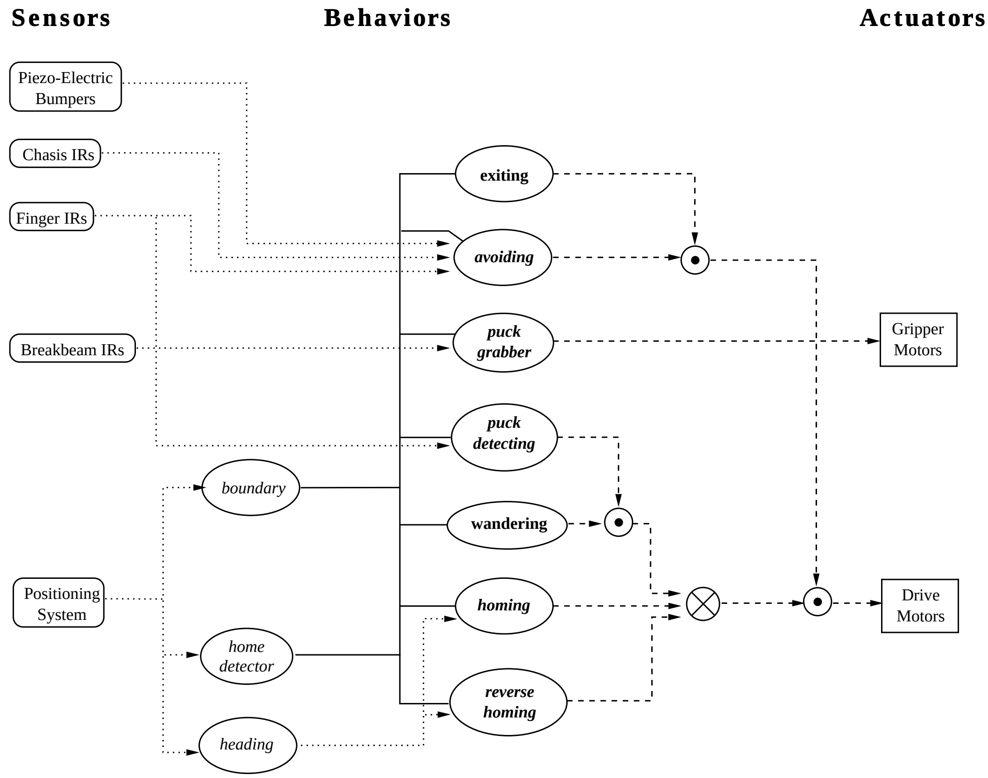


Figure 11: The controller for the Search Caste, the three robot subgroup that searches for pucks.

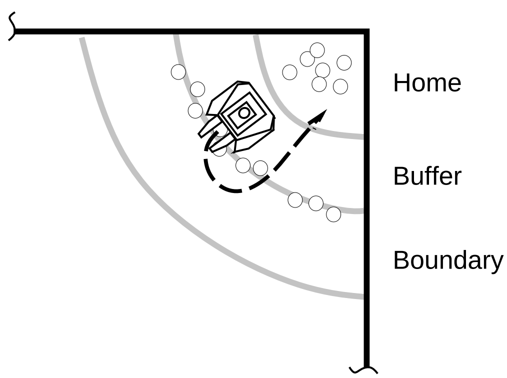


Figure 12: The *sweeping* behavior of the controller for the Goal Caste.

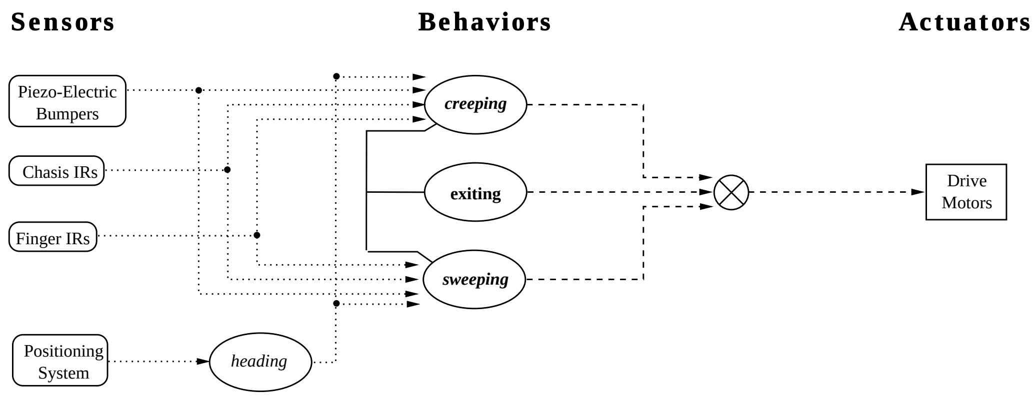


Figure 13: The controller for the Goal Caste, the one robot subgroup that brings pucks from the Boundary/Buffer line to Home.

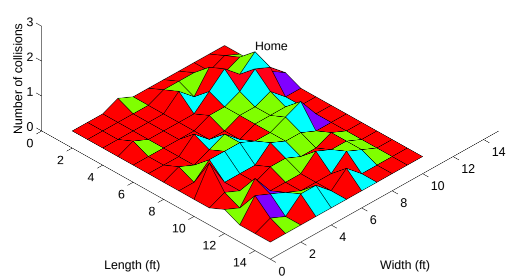


Figure 14: This plot shows the characteristic interference pattern for the caste implementation of the collection task on the four physical robots. The shading corresponds to the height of the peaks.

| Controller | Time (sec) | Avoiding (sec) |
| --- | --- | --- |
| Homogeneous | 549 | 143 |
| Caste | 1447 | 442 |
| Pack | 1081 | 229 |

Table 1: Average time of task completion and average time spent in the *avoiding* behavior for each controller.

completion criterion was the same as for the homogeneous and pack controllers: 14 of the 27 pucks collected. It is clear from a comparison to the characteristic interference of the homogeneous controller (Figure 6) that interference near Home is reduced, as was desired. The overall level of physical interference throughout the Corrall, however, is higher with the caste controller.

The following section provides a more detailed quantitative evaluation and comparison of the controllers in terms of interference, as well as time-to-completion and the distance traveled by each robot.

# 8 Analysis

In order to better evaluate the desirability of and tradeoffs between the three controllers — one homogeneous and two heterogeneous — we performed five experimental trials for each of the controllers, gathering both spatial and temporal data. Initial conditions for all trial were as nearly identical as possible in order to minimize free variables, and the completion criterion was 14 out of 27 pucks collected. For each trial, we gathered data on the time-to-completion of the task, and the location and number of collisions between robots, showing the characteristic interference. We calculated the average total number of collisions for each experiment, for relative comparison of the different schemes. Using the positioning system, we also recorded each robot's location at a approximately 0.3 Hz in order to examine the distance traveled and path taken by each. Finally, we monitored the activity of the internal behaviors of the robots. The *avoiding* behavior was of particular interest since it is one directly invoked by physical interference. We hypothesized that time spent avoiding should be correlated with the total amount of interference in each of the implementations, and would thus serve as an alternate measure of interference. As shown below, this hypothesis was validated (see Table 2).

All of the data presented in this section have been analyzed with one or more statistical tests. We have performed hypothesis tests using Student's t statistic, 1-factor analysis of variance (ANOVA), and 2-factor ANOVA, in order to verify that the differences between the results of the implementations were in fact statistically significant. In all cases, these differences were significant with p-values $< 0.05$.

Our discussion in this section is based on the assumption that the task environment is fixed. Another effective method for altering the spatio-temporal properties considered below is modification of the environment, if such is possible. We could, for example, move Home to the middle of the workspace, thus manipulating the properties like interference and time-to-completion.

The majority of this section eschews a quantitative evaluation of heterogeneity, focusing instead on the performance data mentioned above. This eschewal is justified in the concluding paragraphs where we discuss the poorly understood relationship between heterogeneity and performance in multi-robot groups.

## 8.1 Interference, Avoiding, and Time

One factor that impacts the total amount of interference observed for each implementation is the time-to-completion of the collection task. One would expect that for any given implementation, the longer the trial continues, the more interference or collisions there would be. One would also expect the total amount of time spent in the *avoiding* behavior to be positively correlated with the time of completion. In Table 1 we see that this is indeed the case. The homogeneous implementation has the shortest time-to-completion and the least amount of time spent avoiding; the pack implementation has the next larger times; and the caste implementation has the largest times over all.

| Controller | Interference (collisions) | Avoiding/Time |
|---|---|---|
| Homogeneous | 16.4 | 0.27 |
| Caste | 20 | 0.32 |
| Pack | 12.6 | 0.22 |

Table 2: Average amount of interference and average fraction of time spent in the *avoiding* behavior.

| Controller | Interference/Time (collisions/sec) |
|---|---|
| Homogeneous | 0.030 |
| Caste | 0.014 |
| Pack | 0.012 |

Table 3: Average amount of interference per unit time for each controller.

In their current form, the values for time-to-completion and time-spent-avoiding do not provide much useful information about the amount of interference in each controller. We can observe, however, that the amount of time spent in the *avoiding* behavior is composed of the time spent avoiding other robots (before, during, and after collisions) and the time spent avoiding everything else. Since the environment (discounting the robots) is identical in every trial, we can assume that the amount of avoidance per unit time attributable to non-robot objects is constant between the implementations. This assumption suggests that any differences in the amount of avoidance per unit time between the implementations are primarily due to the avoidance of the other robots, possibly during collisions.

Thus we would expect to see a correlation between the average amount of interference observed in each implementation and the ratio of time spent avoiding to total time. In Table 2 we observe that such a correlation does exist and it is quite large at $\rho = 0.995$. This indicates an important link between these two values and suggests that the ratio of avoiding and total time is a good estimate of relative average interference levels.

Another potentially useful statistic is the amount of interference per unit time. As shown in Table 3, the pack implementation has the most desirable ratio while the homogeneous implementation has the least.

## 8.2 Distance Traveled

As mentioned previously, the energy expended by the robots in completing the task may be a concern if recharging is time-consuming or difficult. Time-to-completion provides one approximation of energy expenditure, but it can be inaccurate, especially with a controller such as our pack version where robots can idle for long periods of time. A better approximation is the amount of work accomplished by the robots during the task. Work ($W$), force ($F$), and displacement ($d$) are related through the elementary physics equation

$$W = F \cdot d \cdot \cos\theta$$

or

$$\frac{W}{F \cdot \cos\theta} = d,$$

where $\theta$ is the angle between the force and displacement vectors. Since the robots are mechanically identical, we can consider $F \cdot \cos\theta$ to be constant among them. This allows us to compare the work done by the robots solely in terms of $d$, the distance traveled. Because the robots are identical, $d$ also provides a reasonable, relative indication of the energy expended in performing the work. Finally, it provides a measure of efficiency: the less work required to accomplish the task, the more efficient the controller.

Table 4 presents the average distance traveled by each robot, and the total over all robots, for each of the three controllers. The values were calculated from the robot position data gathered during the experiments.

| Controller | Robot0 | Robot1 | Robot2 | Robot3 | Total (ft) |
|---|---|---|---|---|---|
| Homogeneous | 123 | 120 | 113 | 119 | 475 |
| Caste | 353 | 370 | 385 | 119 | 1227 |
| Pack | 112 | 145 | 188 | 178 | 623 |

Table 4: Average distance (in feet) traveled by the robots for each controller.

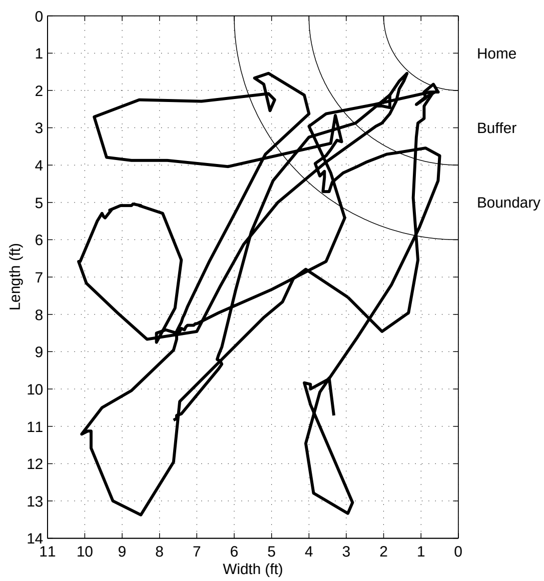


Figure 15: A typical path taken by one physical robot in the homogeneous controller.

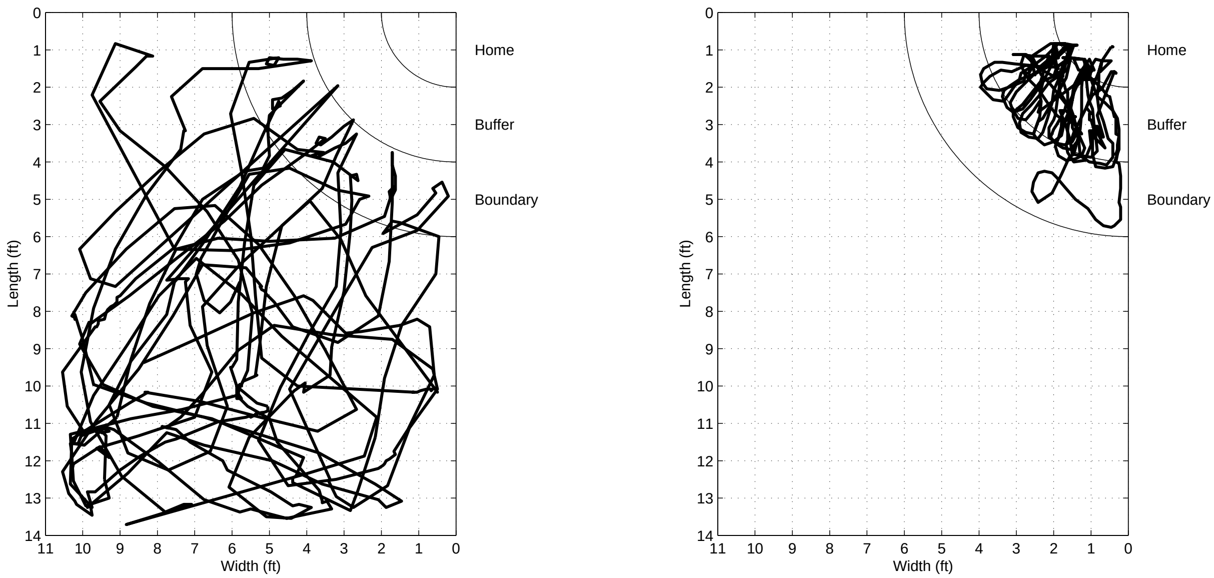


Figure 16: (Left) A typical path of a physical robot in the Search Caste of the caste controller; (Right) a typical path of the robot in the Goal Caste.

The results indicate that the homogeneous controller performs the least work in completing the task, and thus is the most efficient, whereas the caste controller performs the most work and is least efficient.

Although the total distances traveled for the three controllers are statistically different, this is not necessarily true of the distances traveled by the individual robots within a controller. This follows intuitively from the structure of the controllers. In the homogeneous controller where all four robots are behaviorally identical, there is no statistical difference in the distances traveled. In the caste controller, Robot0, Robot1, and Robot2, which comprise the Search Caste, travel similar distances, whereas Robot3 of Goal Caste moves significantly less, as might be expected. In the pack controller, one would expect the less dominant robots to travel less since they spend more time waiting for the dominant robots to deliver pucks. Table 4, with Robot0 as the least dominant and Robot4 as the most dominant, shows that this is the general trend. Although a one-way analysis of variance indicates that there is significant difference among these values, there are too few trials to provide further discrimination using a $t$-test. (The exception is that Robot0 is shown to travel significantly less than Robot2 and Robot3.)

A more qualitative, visual examination of the execution of the controllers is also possible. Figure 15 shows a typical path of one robot in the homogeneous controller. It is clear that the robot searches for pucks, delivers several to Home, and sometimes enters the Boundary without pucks and promptly leaves. Figure 16 (Left) shows a similar path taken by a robot in the Search Caste of the caste controller. The path is much longer than that of the homogeneous controller due to the protracted time of the trial. We also note that the Search Caste very clearly delivers pucks to the Boundary/Buffer line. Figure 16 (Right) shows the complementary path of the Goal Caste collecting pucks from the Boundary/Buffer line and taking them to Home. Figure 17 provides a juxtaposition of typical paths taken by the least dominant and most dominant robot of the pack controller. As expected, the most dominant robot has a path (Right) very similar to that of the homogeneous controller. The least dominant robot, however, has a severely stunted path demonstrating the significance of the time it waits for the more dominant robots to deliver their pucks.

## 8.3 Robustness

During the experimental trials for each controller, we had the opportunity to evaluate group-level robustness. The R2e robots used in the experiments are quite fragile and prone to failure from something as simple as a

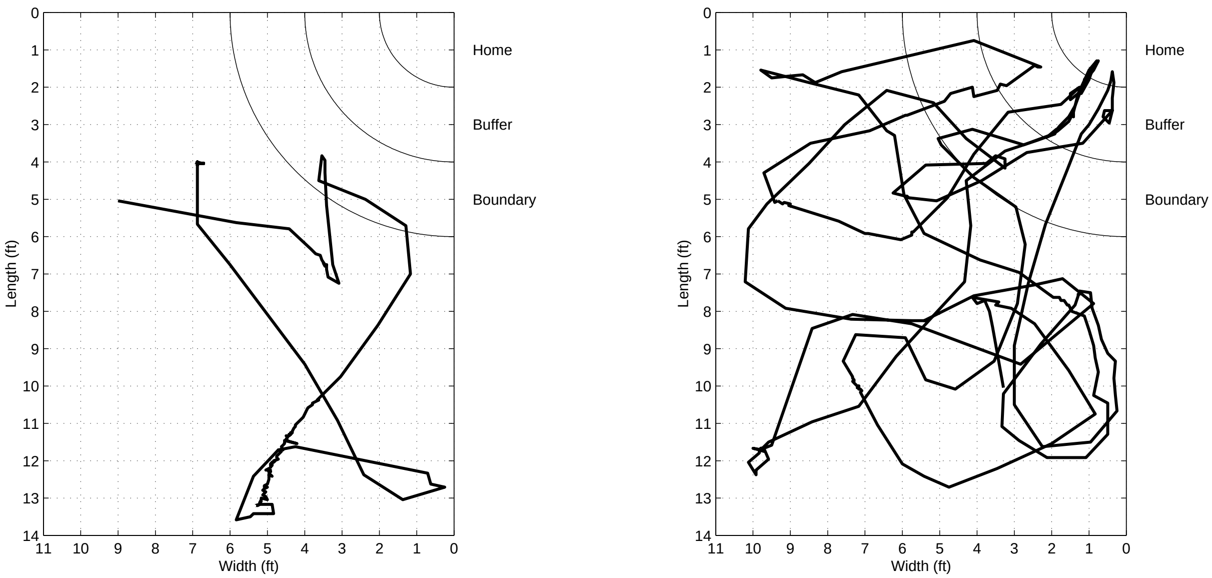


Figure 17: (Left) A typical path of the least dominant robot of the pack controller; (Right) a typical path of the most dominant robot.

buildup of static electricity corrupting memory or causing the robot's computer to crash. There was seldom a trial without multiple failures requiring the failed robots to be restarted. With the homogeneous controller, we noted very clearly that the failure of one robot did not effect the activity of the others. In the pack controller, the less dominant robots of the hierarchy were able to compensate for the failure a dominant robot by using the message passing protocol. If a dominant robot failed while delivering a puck (which occurred at least once per trial), the less dominant robots would stop waiting and begin delivering their pucks. In the caste controller, the Search Caste exhibited group-level robustness similar to the homogeneous controller: the failure of one robot did not affect the other members of the caste. In addition, due to the asynchronicity of interaction between the two castes, the failure of the robot in the Goal Caste did not debilitate the activity of the Search Caste.

## 8.4 Evaluation

Using the analyses presented above we can now discuss the relative desirability of the three controllers. All three are desirable in that they exhibit good group-level robustness. The tradeoff between time and interference captures the relative performance. The homogeneous implementation requires the least time but does not result in the least interference, whereas the pack implementation exhibits the least total interference and least interference per unit time, but takes longer overall. Thus, we must decide which criterion is more important or what kind of compromise we wish to make in the final controller choice. If we can sacrifice some performance time for decreased robot interference, then the pack implementation appears to be the best choice. This solution applies to conservative systems where collisions and the possibility of equipment damage outweighs the required time. In contrast, if total time or energy expenditure is the critical factor, such as in domains where the items to be collected are toxic or dangerous, or robot power is limited, then the homogeneous implementation is the better choice. From this analysis we also observe that the caste implementation does not appear to be a satisfactory solution under any criterion, and may be discarded from consideration.

Although our analysis does not identify one controller that is clearly superior in all respects, it does provide information to make an intelligent decision regarding the tradeoffs between the homogeneous and pack controllers. The designer may decide that one of the controllers sufficiently satisfies the requirements for the task, or might wish to investigate other variations for a more suitable controller. The latter decision is facilitated by the ability to build behavior-based controllers that are easy to modify and evaluate in an expeditious manner, as we have demonstrated here.

## 8.5 Heterogeneity and Performance

Until now we have avoided a quantitative evaluation of the heterogeneity demonstrated by our three controllers. The reason for this is twofold:

1. Quantification of the heterogeneity of a multi-robot system can be subjective and ill-defined.
2. Regardless of how well-defined heterogeneity is, the link between it and performance may be unreliable.

We will consider each of these points in more depth.

Heterogeneity in multi-robot systems remains ill-defined partially because to date there has been little work exploring its quantification. One notable exception is work by Balch on *simple social entropy* and *hierarchical social entropy* (Balch, 1997; Balch, 2000), including a chapter in this book. Both are based on information entropy (Shannon & Weaver, 1963) and provide metrics for quantifying *behavioral* differences in a group of robots.

For illustrative purposes, we use simple social entropy which takes the form $\sum_{i=1}^{M} p_i \log_2(p_i)$, where $M$ is the number of (behavioral) classes of robots, and $p_i$ is the proportion of robots in the $i$th class. According to this measure, our homogeneous controller has one class containing all four robots, giving $p_1 = 1$ and a social entropy of 0, indicating homogeneity. The caste controller has two classes with $p_1 = 1/4 = .25$ and $p_2 = 3/4 = .75$, giving a social entropy of 0.81, indicating some heterogeneity. Though seemingly straightforward, calculating the social entropy of the pack controller introduces the dilemma of subjectivity. If we consider that all of the robots have the same controller and behave similarly, it seems clear that the group is homogeneous and has a social entropy of 0. If, however, we consider that each robot has a defined position in the hierarchy and behaves differently with respect to the other robots, then it appears that there are four classes, each containing one robot. This results in a social entropy of 2.0, indicating maximum heterogeneity. Is the pack controller fully homogeneous or heterogeneous? The fact that both views seem justified helps illustrate our first point: even with a well-defined metric, heterogeneity may still be subjective.

The situation is further complicated if the system contains multiple forms of heterogeneity. The caste controller for instance exhibits not only behavioral heterogeneity but also spatial heterogeneity since the robots occupy different regions. We can quantify spatial heterogeneity using a variation of Balch's social entropy. The Search Caste contains 3 robots occupying 141 square feet ($\text{ft}^2$) of space, for a total of $3 \times 141 = 423$ robot-$\text{ft}^2$. The Goal Caste contains 1 robot and occupies 13 $\text{ft}^2$ of space, for a total of $1 \times 13 = 13$ robot-$\text{ft}^2$. In our calculation, $p_1 = 423/436 = 0.970$ and $p_2 = 13/436 = 0.030$, giving a *spatial* entropy of 0.19 and indicating a small amount of heterogeneity. The question now is how to describe the overall heterogeneity of the caste controller. Should each type of heterogeneity (behavioral and spatial) be defined separately, or should the two numbers be combined into a single value? In the latter case, how should each number be weighted? The influence of each type of heterogeneity could depend on the task the robots are performing and the structure of the environment. Any weighting may thus have to be derived (likely empirically) for the exact scenario. If this is not possible, the overall heterogeneity of the system could remain ambiguous or ill-defined. The addition of other forms of heterogeneity (e.g., involving morphology or sensors) could further complicate the matter.

Even given an adequate measure for all forms of heterogeneity and their combination, the lack of a clear connection between the performance of the system and degree of heterogeneity it exhibits remains a concern. In our work in this chapter, we have compared several aspects of performance among our three controllers, including interference, time-to-completion, and energy expenditure. The important caveat is that these results are not completely general. They are dependent upon the structure of the environment, the physical characteristics of the robots, and the exact details of the controllers. In other words, the same task on different robots in a different environment might give very different results. Adding or removing heterogeneity to the system may improve or degrade performance depending on the details of the system and the aspect of heterogeneity being changed. As in the second point, one may not be able to rely on the results of a heterogeneity/performance comparison generalizing to another situation.

We have seen that the heterogeneity of a multi-robot system can be difficult to quantify, and once quantified is of uncertain relation to performance. Based on our experimental results in foraging, it is not clear how a study of this relationship helps the designer improve a multi-robot system. We therefore do not focus our analysis in this chapter upon that topic. Our hope is that the capability to expeditiously build, modify, and evaluate multi-robot controllers, as we have demonstrated, will help facilitate the future study of issues in group robotics, such as the uses of heterogeneity analysis.

# 9 Conclusions

We have demonstrated the successful application of behavior-based control to the task of distributed multi-robot collection. Our focus has been on developing controllers that are robust to noise and robot failures, and easily modified to facilitate development of the variation that sufficiently satisfies the requirements for the task. Three versions of the collection task were presented: an initial homogeneous controller, and two heterogeneous variations (pack and caste) derived from the spatio-temporal manipulation of physical interference. All three versions were evaluated in a spatio-temporal context using interference, time-to-completion, and distance traveled as the main diagnostic parameters. This work demonstrates that given a good substrate for development (e.g., a useful set of behaviors), it can be relatively easy to implement, evaluate, and compare multi-robot controllers.

# Acknowledgments

The research reported here was conducted at the Interaction Laboratory at the Brandeis University Volen Center for Complex Systems and the Computer Science Department, and at the University of Southern California Robotics Research Labs and Computer Science Department. The work was supported in part by the Office of Naval Research Grant N00014-95-1-0759, by the National Science Foundation Infrastructure Grant CDA-9512448, by the Office of Naval Research Grant N00014-00-1-0140, and by the Sandia National Labs Grant 3065.

# References

Arkin, R. C. (1998). *Behavior-based robotics.* The MIT Press: Cambridge, Massachusetts.

Arkin, R. C., & Ali, K. S. (1994). Reactive and telerobotic control in multi-agent systems. *From Animals to Animats: International Conference on Simulation of Adaptive Behavior* (pp. 473–478). MIT Press.

Arkin, R. C., Balch, T., & Nitz, E. (1993). Communication of behavioral state in multi-agent retrieval tasks. *IEEE International Conference on Robotics and Automation* (pp. 588–594). IEEE Computer Society Press.

Arkin, R. C., & Hobbs, J. D. (1993). Communication and social organization in multi-agent systems. *From Animals to Animats: International Conference on Simulation of Adaptive Behavior* (pp. 486–493). MIT Press.

Balch, T. (1997). Social entropy: a new metric for learning multi-robot teams. *Proceedings, 10th International Florida AI Symposium (FLAIRS-97).* Daytona Beach, FL.

Balch, T. (2000). Hierarchical social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, *8*, 209–237.

Beckers, R., Holland, O., & Deneubourg, J. (1994). From local actions to global tasks: Stigmergy and collective robotics. *Artificial Life IV, Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems* (pp. 181–189). MIT Press.

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

Brooks, R. A. (1990). *The behavior language; user's guide* (Technical Report AIM-1227). MIT AI Lab.

Brooks, R. A. (1991). Intelligence without reason. *Proceedings, IJCAI-91.*

Cao, Y. U., Fukunaga, A. S., & Kahng, A. B. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, *4*, 1–23.

Drogoul, A., & Ferber, J. (1992). From tom thumb to the dockers: Some experiments with foraging robots. *From Animals to Animats II* (pp. 451–459). The MIT Press: Cambridge, Massachusetts.

Fontán, M. S., & Matarić, M. J. (1998). Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, *14*.

Goldberg, D., & Matarić, M. J. (1999a). *Augmented markov models* (Technical Report IRIS-99-367). University of Southern California.

Goldberg, D., & Matarić, M. J. (1999b). Coordinating mobile robot group behavior using a model of interaction dynamics. *Proceedings, The Third International Conference on Autonomous Agents (Agents '99).* Seattle, Washington: ACM Press.

Goldberg, D., & Matarić, M. J. (2000a). *Detecting regime changes with a mobile robot using multiple models* (Technical Report IRIS-00-382). USC Institute for Robotics and Intelligent Systems.

Goldberg, D., & Matarić, M. J. (2000b). Reward maximization in a non-stationary mobile robot environment. *Proceedings, Agents 2000.*

Goldsmith, S. Y., Feddema, J. T., & Robinett, R. D. (1998). Analysis of decentralized variable structure control for collective search by mobile robots. *Sensor Fusion and Decentralized Control in Robotic Systems* (pp. 40–47). Boston, Massachusetts: SPIE.

Hasegawa, Y., Ito, Y., & Fukuda, T. (2000). Behavior coordination and its modification on brachiation-type mobile robot. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation.* San Francisco, CA: IEEE.

Holland, O., & Melhuish, C. (2000). Stigmergy, self-organization, and sorting in collective robotics. *Artificial Life*, *5*.

Lindström, M., Orebäck, A., & Christensen, H. I. (2000). Berra: A research architecture for service robots. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation.* San Francisco, CA: IEEE.

Lund, H. H., & Pagliarini, L. (2000). Robocup jr. with lego mindstorms. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation.* San Francisco, CA: IEEE.

Matarić, M. J. (1992a). Behavior-based systems: Key properties and implications. *IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligent Control Systems* (pp. 46–54). Nice, France.

Matarić, M. J. (1992b). Minimizing complexity in controlling a collection of mobile robots. *Proceedings, IEEE International Conference on Robotics and Automation* (pp. 830–835). Nice, France.

Matarić, M. J. (1995). Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, *16*, 321–331.

Parker, L. E. (1992). Adaptive action selection for cooperative agent teams. *From Animals to Animats: International Conference on Simulation of Adaptive Behavior* (pp. 442–450). MIT Press.

Parker, L. E. (1994). *Heterogeneous multi-robot cooperation.* Doctoral dissertation, MIT.

Pirjanian, P. (1998). *Multiple objective action selection & behavior fusion using voting.* Doctoral dissertation, Institute of Electronic Systems, Alborg University, Denmark.

Pirjanian, P., Christensen, H. I., & Fayman, J. A. (1998). Application of voting to fusion of purposive modules: An experimental investigation. *Robotics and Automation*, *23*, 253–266.

Pirjanian, P., & Matarić, M. J. (2000). Multi-robot target acquisition using multiple objective behavior coordination. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*. San Francisco, CA: IEEE.

Rosenblatt, J., Willams, S., & Durrant-Whyte, H. (2000). Behavior-based control for autonomous underwater exploration. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*. San Francisco, CA: IEEE.

Shannon, C. E., & Weaver, W. (1963). *Mathematical theory of communication*. Univerisity of Illinois Press.

Tan, K.-H., & Lewis, M. A. (1996). Virtual structures for high-precision cooperative robotic control. *IEEE/TSJ International Conference on Intelligent Robots and Systems*. IEEE Computer Society Press.

Vaughan, R. T., Støy, K., Sukhatme, G. S., & Matarić, M. J. (2000). Whistling in the dark: Cooperative trail following in uncertain localization space. *Proceedings of the Fourth International Conference on Autonomous Agents* (pp. 187–194). ACM Press: New York.

Werger, B. B., & Matarić, M. J. (1996). Robotic "food" chains: Externalization of state and program for minimal-agent foraging. *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior* (pp. 625–634). The MIT Press: Cambridge, Massachusetts.