Swarm Robotic Odor Localization: Off-Line Optimization and Validation with Real Robots

Adam T. Hayes, Alcherio Martinoli, Rodney M. Goodman
Collective Robotics Group
136-93 California Institute of Technology, Pasadena CA 91125
athayes, alcherio, rogo@caltech.edu
http://www.coro.caltech.edu

Abstract

This paper presents an investigation of odor localization by groups of autonomous mobile robots using principles of Swarm Intelligence. First, we describe a distributed algorithm by which groups of agents can solve the full odor localization task more efficiently than a single agent. Next, we demonstrate that a group of real robots under fully distributed control can successfully traverse a real odor plume, and that an embodied simulator can faithfully reproduce these real robots experiments. Finally, we use the embodied simulator combined with a reinforcement learning algorithm to optimize performance across group size, showing that it can be useful not only for improving real world odor localization, but also for quantitatively characterizing the influence of group size on task performance.

Keywords

Collective Autonomous Robotics, Odor Localization, Plume Traversal, Off-line Optimization, Swarm Intelligence

I. Introduction

This paper presents an investigation of odor localization by groups of autonomous mobile robots using principles of Swarm Intelligence (SI), a computational and behavioral metaphor for solving distributed problems that takes its inspiration from

biological examples provided by social insects. In most biological cases studied so far, robust and capable group behavior has been found to be mediated by nothing more than a small set of simple interactions among individuals and between individuals and the environment[1]. The application of SI principles to autonomous collective robotics aims to develop robust task solving by minimizing the complexity of the individual units and emphasizing parallelism, exploitation of direct or indirect interactions, and distributedness. The main advantages of this approach are three: first, scalability from a few to thousands of units, second, flexibility, as units can by dynamically added or removed without explicit reorganization, and third, increased system robustness, not only through unit redundancy but also through the design of minimalist units. Several examples of collective robotics tasks solved with SI principles can be found in the literature: aggregation[2, 3] and segregation,[4] beacon localization,[5] stick pulling,[6] collective transportation,[7] and foraging.[8]

Solving a task using the SI approach reduces to determining a set of local rules which, when carried out in parallel by a group of agents, has the desired global effect. These rules could involve the control of behavior (software mediated) and/or direct physical interactions (hardware mediated). Because software parameters are easier to manipulate, they are the focus of this study. Each rule can have a set of associated parameters, and once the rules have been chosen, maximizing team performance involves solving a global optimization problem. If a deterministic analytical model describing system performance exists, there are efficient search methods available.[9] However, because SI systems depend heavily on unpredictable agent-to-agent and agent-to-environment interactions, performance is often stochastic, and evaluative, rather than gradient based, search methods are appropriate. This type of control optimization has been extensively studied for the case of a single agent,[10, 11, 12] including the particular type of off-line optimization which is of interest in this paper.[13, 14] Because optimal parameter values can be a function of the number of agents in the system, parameter optimization for each group size is necessary before the influence of the number of agents on system performance can be analyzed. We frame the problem in this way in order to attempt to quantify the advantages of multi-robot systems on this

task, although in a broader framework the number of robots can become another system parameter to be optimized. The notion of optimizing parameters for a particular group size may seem at odds with the flexibility provided by the SI framework, but in fact the robustness provided by this approach allows performance to degrade gracefully as system behavior diverges from optimal.

Recent advances have been made in understanding biological and artificial odor classification and odor localization and tracking as developed in moths[15, 16] and rats[17] in the air, and lobsters[18] and stomatopods[19] in water. Biology utilizes olfaction for a wide variety of tasks including finding others of the same species, communication, behavior modification, avoiding predators, and searching for food. Odors, unlike visual and auditory perceptions, are non-spatial: they possess neither spatial metric nor direction. In contrast, odorant stimuli possess both spatial and temporal character, snaking out complex plumes that can wander over a wide area. This implies that a level of sophistication beyond gradient following is necessary for localization of an odor source.

Animals use a combination of `hardware' (frequency of receptor adaptation, perhaps), `software' (temporal integration and/or spatial integration), and behavioral search strategies (both intrinsic and landmark-based) to locate odor sources. Odor localization is in essence a behavioral problem that varies from animal to animal. While some animals exploit fluid information at different layers (lobster) or several residues on the ground (ants), others can track odors in the air (moths) or use a combination of information (dogs). From an engineering standpoint there are advantages to combining odor tracking with mobile robots, such as in the detection of chemical leaks and the chemical mapping of hazardous waste sites. We are interested in developing groups of small mobile robots that use odor tracking algorithms, multiple sensory modalities (e.g. odometry, anemometry, olfaction), and sensory fusion to search out and identify sources of odor.

The aim of the case study described in this paper is four-fold. Firstly, we describe a distributed algorithm by which groups of agents can solve the full odor localization task

more efficiently than a single agent. Secondly, we demonstrate that a group of real robots under fully distributed control can successfully traverse a real odor plume. Thirdly, we show that an embodied simulator can faithfully reproduce these real robots experiments. Lastly, we establish that this simulator combined with a reinforcement learning algorithm can be used to optimize performance across group size, and thus can be useful not only for systematically improving real world odor localization, but also for quantitatively characterizing the influence of group size on task performance.

II. The Odor Localization Problem

The general odor localization problem addressed in this paper is as follows: find a single odor source in an enclosed 2D area as efficiently as possible. This can be broken down into three subtasks: plume finding - coming into contact with the odor, plume traversal - following the odor plume to its source, and source declaration - determining from odor acquisition characteristics that the source is in the immediate vicinity. Plume finding amounts to a basic search task, with the added complication, due to the stochastic nature of the plume, that a simple sequential search is not guaranteed to succeed. Plume traversing requires more specialized behavior, both to progress in the direction of the source and to maintain consistent contact with the plume. Source declaration does not necessarily have to be done using odor information, as typically odor sources can be sensed via other modalities from short range, but here we propose a solution using no extra sensory apparatus.

A. Biological Inspiration

As an odor source dissolves into a fluid medium, an odor plume is formed. The turbulent nature of fluid flow typically breaks the plume into isolated packets, areas of relative high concentration surrounded by fluid that contains no odor. The task of odor localization thus becomes one of plume traversal, or following the trail of odor packets upstream to the source. This becomes difficult as odor packets become more sparse (due to source intermittency and diffusion below detectable levels) and more dispersed (due to flow meander).

Although the approach of moving slowly and continually sampling odor and flow data to reduce environmental noise is used in nature (starfish) and has been applied to robotic systems,[20, 21] environmental and behavioral constraints (e.g. significant plume sparseness or meander, time critical performance) can render these systems ineffective. In that case, upon sensing an odor signal, a good policy is to move directly upwind, as a good immediate local indication of source direction under such circumstances is the instantaneous direction of flow.[22] When the odor is no longer present, a good strategy is to perform a local search (known as casting in the biological literature) until it is reacquired, as the location of the previous packet encounter provides the best immediate estimate of where the next will occur. This type of surge-cast behavior has been observed in moths,[23] and its performance has been studied in simulation.[16]

The previous work on this odor localization algorithm was aimed at studying biology, which limited the sensory and behavioral time scales investigated. When applying these ideas to robots, however, the separation between algorithm and underlying hardware is much more clear, and it no longer makes sense to constrain behavior strictly by sensory response characteristics. Therefore, in this work key aspects of the search behavior, such as surge duration and casting locality, are treated as algorithm parameters.

B. The Spiral Surge Algorithm

The basic odor localization algorithm used in this study, Spiral Surge (SS), is shown in Figure 1. It consists of different behaviors related to the three different subtasks.

Fig. 1. Spiral Surge odor localization behavior.

Plume finding is performed by an initial outward spiral search pattern (`SpiralGap1`). This allows for thorough coverage of the local space if the total search area is very large and initial information can be provided by the deployment point (an external `best guess' as to source location). Alternatively, if no a priori knowledge is available, a spiral with a

gap much greater than the arena size (producing essentially straight line search paths) provides an effective random search procedure.

Plume traversal is performed using a type of surge algorithm. When an odor is encountered during spiraling, the robot samples the wind direction and moves upwind for a set distance (`StepSize`). If during the surge another odor packet is encountered, the robot resets the surge distance but does not resample the wind direction. After the surge distance has been reached, the robot begins a spiral casting behavior, looking for another plume hit. The casting spiral can be tighter than the plume finding spiral (`SpiralGap2`), as post surge the robot has information about packet density and a thorough local search is a good strategy. If the robot subsequently re-encounters the plume, it will repeat the surging behavior, but if there is no additional plume information for a set amount of time (`CastTime`), the robot will declare the plume lost and return to the plume finding behavior (with a wider, less local, spiral gap parameter).

Source declaration can be accomplished using the fact that a robot performing the plume traversal behavior at the head of a plume will tend to surge into an area where there is no plume information, and then spiral back to the origin of the surge before receiving another odor hit. If the robot keeps track internally of the post spiral inter-hit distances (using odometry, for example, which is sufficient because information must be accurate only locally), a series of small differences can indicate that the robot has ceased progress up the plume, and must therefore be at the source. However, because small inter-hit distances can occur in all parts of the plume, this method is not foolproof, and tuning of the difference threshold (`SrcDecThresh`), as well as the number of observed occurrences before source declaration (`SrcDecCount`), is required to obtain a particular performance within a given plume. See Table I for a summary of individual SS parameters.

SS uses only binary odor information generated from a single plume sensor. This is motivated partially because this is the most simple and reliable type of information that can be obtained from real hardware. There may be more information encoded in fine

plume structure,[24] however, due to the highly stochastic nature of turbulent fluid flow and the odor-packet nature of the plume, it is unclear that more complex sensing - via graded intensity information or larger sensor arrays - would benefit an odor localizing agent when flow information is available through other means.

TABLE I

Spiral Surge Algorithm Parameters

`SpiralGap1`	Initial spiral gap width
`SpiralGap2`	Plume reacquisition spiral gap width
`StepSize`	Surge distance post odor hit
`CastTime`	Length of time before reverting from reacquisition to initial search spiral
`SrcDecThresh`	Significance threshold between consecutive separate odor hits
`SrcDecCount`	Number of significant differences before source declaration

C. Collaborative Spiral Surge

One way to increase the performance of a robot swarm is collaboration. In particular, if collaboration is obtained with simple explicit communication schemes such as binary signaling, the team performance can be enhanced without losing autonomy or significantly increasing complexity at the individual level. Several simple types of communication can be integrated into basic SS. Though this issue is not explored in this paper, the effects of communication strategies can change depending on the environment, so communication type should be a tunable system parameter.

D. Plume Traversal

This paper will focus on the plume traversal subtask because it contains most of the plume related complexity present in the full odor localization task, and due to experimental limitations it is not feasible to study all phases with real robots at this time. Therefore in the following experiments the full SS algorithm is not employed, as agents are always in `plume traversal' mode. This makes `SpiralGap2` and `StepSize` the parameters of interest, and effectively fixes `CastTime` at infinity. To study plume traversal, we place groups of agents within a starting area at the distal end of an odor plume in an enclosed arena. Over repeated trials we measure the time and distance traveled by the whole group until the first agent comes within a given radius of the plume source (T_{SF}, D_{SF}).

To justify the high density of agents in the plume (which would be unlikely given that in the general problem the plume area is a small percentage of the total search area), we allow explicit communication between the agents that causes all downwind agents (locally determined from previous individual measurement and odometry) to surge toward an agent that has received an odor hit and is initiating its own surge behavior. This provides an attractive force that holds the group together as it traverses the plume and makes the experimental situation (in which many robots are simultaneously within the plume) more reasonable.

Efficiency for the plume traversal task cannot be defined in the general case. Instead, we use two basic measures of task performance: time and group energy (which can be considered proportional to the sum of the individual distances traveled). Since these measures are physically independent, a composite metric incorporating a particular weighting of these two basic factors can be considered.

$$Q = \alpha T_{SF} + \beta D_{SF} \qquad (1)$$

$$P = \frac{\alpha T_{MIN} + \beta D_{MIN}}{Q} \qquad (2)$$

Q is an arbitrary weighting of time and distance. By choosing specific values for α and β, the appropriate relationship can be generated for evaluating any particular application. The form of P ensures that for any exponent α or β greater than 0, the optimal system will achieve a performance of 1, and any that require more time or distance will have a performance less than 1. In this study we experimentally determine the optimum values for the given task (T_{MIN}, D_{MIN}) from a real robot executing the optimal behavior (a straight line path from start to goal areas at maximum speed). Maximum speed, which determines the relationship between the time and distance values, is determined by the maximum safe operating speed of the real robot in the given environment.

III. Materials and Methods

A. Real Robots

We use Moorebots, as shown in Figure 2. The plume traversal arena is 6.7 by 6.7 m, and the robots are 24 cm in diameter. In addition to the standard configuration,[25] each robot is equipped with four infra-red range sensors for collision avoidance, a single odor sensor tuned to sense water vapor, and a hot wire anemometer. Note that the hardware requirements needed by the SS algorithm could be fulfilled by robots much simpler and smaller than Moorebots (e.g. Khepera[26] or even Alice[27] robots). This is consistent with one of the main characteristics of the SI approach that calls for minimization of the individual complexity. The use of more sophisticated robots such as Moorebots in these experiments is motivated by their superior user interface which in turn allows extended capabilities for monitoring, debugging, and implementation flexibility.

Fig. 2. A Moorebot equipped with wind, odor, and proximity sensors, as well as markings for overhead tracking.

The odor sensor detects the presence of an airborne substance through a change in the electrical resistance of a chemically sensitive carbon-doped polymer resistor.[28] We generate a water plume using a pan of hot water and an array of fans. Mapping the plume using a random walk behavior (see Figure 3) over long periods reveals a stable

plume boundary. A stationary environment is critical because it reduces the number of trials required to produce accurate performance measurements.

Fig. 3. Plume hits received by 6 real robots over 1 hour while performing a random walk behavior.

The anemometer is enclosed in a tube that gives it unidirectional sensitivity, which, combined with a scanning behavior, allows the robot to measure wind direction. A wind map of 2102 individual samples averaged spatially is shown in Figure 4. Although the wind field is relatively simple compared to those found in real environments, it does fluctuate somewhat over time due to eddies generated by interaction between the mean flow and the walls of the room. These fluctuations produce variations in the instantaneous wind measurements, which must be dealt with by the plume traversal algorithm.

Fig. 4. Average wind direction in plume traversal arena as measured by the real robots. Plume source at upper right. Arrow lengths are proportional to the mean flow magnitude at the tail of each arrow.

An overhead camera tracking system, combined with a radio LAN among the robots and an external workstation, is used to log position data during the trials, reposition the robots between trials, and emulate the binary communication signals. Trials of different group size are interleaved and inactive robots are automatically positioned at recharging stations. The arena layout, as seen from the overhead camera, is shown in Figure 5.

Fig. 5. Real robot arena as seen from overhead camera.

B. Embodied Simulation

We used Webots,[29] a 3D sensor-based, kinematic simulator, originally developed for Khepera robots,[26] to systematically investigate the performance of SS in simulation. This embodied simulator has previously been shown to generate data that closely

matches real Khepera[6, 30, 31] and Moorebot[5] experiments, so we were confident that real robot behavior was accurately captured.

Fig. 6. Webots plume traversal arena with average plume intensity map.

The physical arena was captured in Webots, as shown in Figure 6. To properly capture the plume stimulus, we incorporated a series of leaky source 2D plume images generated in a water flume by Philip Roberts and Donald Webster at Georgia Tech.[32] Such 'plume movies', even though they do not capture the influence of the agents on plume dynamics, offer a good approximation to the discretized (packet-like) nature of odor stimulus received in real environments. We scaled the recorded plume data to imitate the average speed and envelope of the real plume data (see Figure 7 and Figure 3), and tuned the odor sensitivity threshold (higher threshold leads to less odor information) based on performance observed in our real arena. Odor hit frequency differences between the real and simulated maps are due to different polling rates of the respective measurement systems and differences in response bandwidth of the real and simulated sensors. Flow information was taken directly from the real robot data (as shown in Figure 4) and introduced into the embodied simulations.

Fig. 7. Plume hits received by 6 simulated robots over 1 hour.

IV. Off-Line Machine Learning Optimization

Machine learning in multi agent systems has been the subject of much recent study.[33] The main design issues that must be addressed by these systems involve diversity in the control (homogeneous vs. heterogeneous) and the type of reward signal (local vs. global). Optimization algorithms allowing controller differentiation or using local reinforcement signals face a daunting credit assignment problem, because it is difficult to determine which actions of which agent are responsible for the outcomes observed in the system. Learning under these conditions has been addressed with the help of explicit communication,[34] a priori information about proper task completion,[35] and careful alignment of individual and group performance metrics,[36, 37] although none of

these approaches can be easily applied to the problem of plume traversal. Extensive peer-to-peer communication is undesirable, because the overhead of providing and maintaining each agent's unique identification, as well as a possibly exponentially increasing number of messages, makes it difficult to scale to large group sizes. There is no efficient way to determine progress up the plume (measuring packet densities at different points would be time consuming and possibly unreliable), so breaking the task down into subtasks that can be rewarded more directly is not possible. And there is no individual metric that captures performance more clearly than the group metric used because performance is inherently defined at a system-wide level (contrast this to a task in which the goal is to get as many plume hits as possible-- here the group performance can be broken down clearly into the sum of individual performances, each of which may provide a more salient reinforcement signal to an individual agent).

The use of homogeneous controllers with a global reward signal offers another answer to the credit assignment problem for off-line control optimization. By making the learning agent operate in the space of algorithm parameters, and providing only measures of group performance, there effectively becomes one agent and one reward signal and the credit assignment problem no longer applies.[38] This may be an extreme simplification to the problem of learning in distributed multi-agent systems, but it provides a way to optimize team performance when evaluation is expensive, as is often the case with real-world environments that include a strong stochastic component.

The optimization procedure for the plume traversal task involves the off-line tuning of two parameters, `SpiralGap2` and `StepSize`. Optimization for each group size allows meaningful comparison of performance across group size. In this initial study the selection of design points (i.e., parameter pairs over which to optimize) is done a priori, although there are techniques for selecting them adaptively[14, 39] which may be utilized in further studies. The possible parameter space is bounded to include a wide range of behaviors. Small spiral gaps and step sizes (in comparison to the arena size) induce more local search patterns which benefit from robot proximity to the source, while larger parameter values produce global searches which cannot take advantage of robot

location information, but likewise are not adversely affected when the robot wanders away from the source. The design points are selected from a logarithmically spaced grid. This spacing is chosen to reflect the fact that for this task the performance surface is more highly sloped in the more local (smaller parameter) region of the search space.

Once the design points x_i ($i = 1 \ldots N$, where N is the total number of points), are selected, the optimization is performed as follows:

1) Initialize the set of active points A to include all x_i.
2) At each cycle j, simulate a trial at each x_i in A, storing the result $Q = y_i^j$ in Y_i. Q is defined in eq. (1). Each Y_i represents a set of performance values generated using the input parameters x_i.
3) Let $Q_{Max} = \max_i E(Y_i)$ and $E(Y_m) = Q_{Max}$
   For each $x_i \in A$, if
   $\eta Q_{Max} > Q_{Max} - E(Y_i) + \gamma SE(Y_m - Y_i)$

   where i != m, remove x_i from A and place it in S_m. $E(x)$ represents the expected value of x. $SE(x)$ represents the standard error of x. S_i is the set of points that have been determined to produce a performance that is to some degree of certainty within some margin of the performance of x_i.

4) For each $x_i \in A$, if $Y_i < Y_k$, for some $x_k \in A$, as determined by a Kolmogorov-Smirnov test to confidence level ε, remove x_i from A and return all members of S_i to A.

5) If more than one remains x_i in A, go to Step 2.

At the end of the process, the remaining point x_{Max} represents the best guess at the optimum performance. Q is defined in eq. (1). This algorithm is defined by the initial design choice method and three parameters: η, γ, and ε. η defines the margin around x_{Max} in which it is defined to be not cost effective to further optimize (e.g., if $\eta = .1$ then if

all remaining options are determined to be within 10% of each other, the optimization stops). γ defines the desired level of certainty of achievement of the margin defined by η. ε defines the level of certainty that the Kolmogorov-Smirnov test makes proper decisions. Basically, on each iteration all active points are sampled, and then Step 3 removes input points that perform close to the current estimate of best performance, and Step 4 eliminates points that perform detectably worse than another active point. *Q* is used so that equal weight is given to all measurements in calculating the sample mean, and poor performances do not effectively become 0.

This optimization procedure is related to that described by Yakowitz,[13] with several important distinctions. First, instead of drawing the design points on the fly from an a priori pdf, all such input points are defined from the outset of the optimization process. This enables the designer to tailor the state space coverage to the evaluation resources available. Second, rather than refining estimates across all points until some pre-determined stopping time, this method removes design points from consideration as they are determined to be inferior or definitively within some defined range of the current optimal point, thereby allocating more evaluation time to the most promising input points. Thirdly, this method compares design points via evaluating entire sample distributions using a non-parametric test as opposed to via the sample mean. It is feasible to store all of the performance data and compare distributions rather than sample means or some other central tendency measurement because data generation is considered to be expensive (which is true for embodied simulation and even more so for real robots), and data analysis comparatively cheap. A non-parametric comparison test is used because we do not want to assume anything about the performance distribution a priori, nor do we want to take enough samples to generate an appropriate model. This process assumes stationarity of the environment during the optimization process, and also a close correspondence between training and deployment environments.

V. Results and Discussion

A. Real Robots

We tested real robot plume traversal performance using two sets of SS parameters and two control experiments. As previously stated, only `SpiralGap2` and `StepSize` are considered because we are looking only at the plume traversal aspect of the task. `SS1` represents a non-local search in that its search paths are straight and its surges extend to the boundaries of the arena. `SS2` uses a smaller spiral gap and surge length to perform a more local exploration of the arena. `Random Odor` uses `SS2` parameters, and receives odor hits that are generated from the time sequence of `SS2` odor hits but are not correlated with robot position in the arena. This control experiment investigates whether an algorithm incorporating precise odor packet location information is more efficient than a blind upwind surging behavior. An alternative experiment could be to decouple the wind source from the odor source by creating a wind field with an array of fans, but due to practical limitations in our experimental set-up, the Random Odor case was easier to implement and provided equivalent information from a proof-of concept point of view. `Random Walk` takes straight line paths and random avoidance turns at boundaries (using no odor or flow information) to provide a traversal performance baseline. Specific parameters relating to the real robot tests are listed in Table II. 15 trials of each group size were run for `SS1`, `SS2` and `Random Odor`, and 30 trials were run for `Random Walk` due to the high variance of performance values.

TABLE II

Plume Traversal Parameter Values

Agent Speed	.325 m/s
Arena Length	6.7 m
Plume Length	9 m
Plume Speed	1 m/s
Src Dec Radius	.88 m
Plume:Arena Area	1:2.3
Goal:Arena Perimeter	1:18.0
α,β	1
T_{Min}	19.0 s

D_{Min}	6.2 m
SS1: SpiralGap2	1785 km
SS1: StepSize	9.1 m
SS2: SpiralGap2	.357 m
SS2: StepSize	.91 m

Figures 8 and 9 show that for all conditions studied, traversal time decreases with group size while group distance traveled increases. This indicates, as expected for a search task, that as time becomes more important to performance than energy usage (i.e., $\alpha > \beta$ in eq. (1)), larger group sizes will be preferred.

Figure 10 shows that while single robots are generally most efficient in this arena (for $\alpha = \beta = 1$), SS1 gives the best results for each group size (significant via K-S test to $p < .01$ for group size $\in \{1, 2, 3\}$), demonstrating successful real robot plume traversal. Random Odor performs worse than SS2 for all group sizes (significant as above for group size $\in \{1, 2, 4, 6\}$), indicating that location of odor information is an important aspect of the search algorithm. This means that SS is actually plume tracing rather than simply localizing the source of the wind, because if it were only wind localizing, one would expect Random Odor to perform exactly the same as SS2. Also, SS2 performs worse than SS1 (significant as above for all group sizes), suggesting that local search is not a good strategy in this small arena where the goal-to-search perimeter ratio is high (i.e., it is likely to find the goal by chance). Random Walk retains relatively constant performance across group size, and at the larger group sizes its performance tends to approach the optimal observed performance. This suggests that as a search arena becomes overcrowded, random movement becomes the best strategy. All error bars in the plots represent standard error.

Fig. 8. Normalized time across group size for real robot trials. Lower values are better.

Fig. 9. Normalized distance across group size for real robot trials. Lower values are better.

Fig. 10. Performance across group size for real robot trials. Higher values indicate better performance.

B. Embodied Simulations

We successfully reproduced the real robot performance data in Webots, as shown in Figure 11. Data represents 1000 trials per group size. All parameters in Table II apply to the Webots data as well. Only `SS1` for group size of one robot produces significantly different results (as determined by a 2-tailed K-S test with $p < .01$) between Webots and the real robots, and even in this case the error bars overlap. Because our Webots data closely matches our available real robot data, it is reasonable that further simulated experiments will accurately reflect real world behavior.

Fig. 11. Performance of real robot and Webots trials across group size. Higher values indicate better performance.

C. Optimization with the Embodied Simulator

Due to the large goal-to-search perimeter ratio in real robot arena, there is no advantage to local search, and `SS1` represents the optimal parameter set. This intuitive result was confirmed by the optimization process, although it is somewhat disappointing because `SS1` represents essentially a degenerate case of SS plume traversal. In a larger arena, the traversal task becomes more difficult, and the local search properties of SS should become more valuable.

To examine this hypothesis we enlarged the arena to 16 times its original area and optimized the plume traversal performance across group size. The simulated plume in the larger arena remained the same length and speed as before, and to make it more realistic the cross-plume scaling was eliminated. Without the proximity of walls and fans to create turbulent flow, the plume structure most likely observed in a large arena is best

represented by the structure found in the original flume data. Similarly, the wind data from the real arena is no longer applicable, so wind direction values were generated using 10% white noise from the plume axis. All other parameters remained the same as in the previous trials.

Optimization was performed 10 times for each group size using the parameters shown in Table III. Repetition was necessary because the full SS algorithm is not being used, so when the agent loses the plume, it takes a long time for the local search spiral to be reacquired. This results in heavy tailed performance distributions, which are difficult to rank correctly. Use of full SS (which increases the optimization dimension) will make the performance distributions better behaved, and the optimization results will be more consistent.

TABLE III

Optimization Parameter Values

N	81 (9x9)
`SpiralGap2` range	.14-28.3 m
`StepSize` range	.37-9.1 m
η	.1
γ	1.96
ε	.001

The best parameters observed in the results of the optimization process (over all 10 trials) are shown in Table IV. SpiralGap2 remains constant across group size, while StepSize increases. One would expect both parameter values to increase with the number of robots, because larger parameter values correspond to less local search. Larger group sizes endowed with a less local search behavior cover space more efficiently and can afford an increased risk of losing contact with the plume since the task will be accomplished by only one of them reaching the source. It is likely that SpiralGap2 simply has a much greater influence on search locality (as its effect compounds over time), and due to the coarse granularity of the search space, all group

sizes peak at the same value. This hypothesis could be tested by focusing the parameter search in this region and repeating the optimization process, or by enhancing the optimization method to choose attractive design points on the fly.

Performance from 4000 trials using the best values for each group size is shown in Figure 12. For these particular task constraints and time-energy weighting, 3 robots perform most efficiently. Smaller group sizes are too likely to lose the plume, and larger group sizes waste energy with overlapping search areas. The optimal group size (3) in the larger arena is larger than the optimal group size (1) in the smaller arena, and although the data is not yet complete (larger arena and group sizes have yet to be tested) we expect that as search area size increases, because the penalty for plume loss will be come more severe, larger group sizes will become optimal. This concept can be generalized to state that the more difficult the plume is to stay in contact with (i.e. due to high sparseness or meander), the larger the number of robots in the optimal group size. We are currently testing this hypothesis, as it could be used to help determine ideal deployment numbers based on source concentration expectation and atmospheric conditions. In addition, supporting the data observed in the real robot arena, the raw data from these optimized trials, shown in Figure 13, indicates that as time becomes more valued over energy used, larger group sizes will become optimal.

TABLE IV

Best SS Parameter Values for Large Arena

Group Size	SpiralGap2 [m]	StepSize [m]
1	.62	1.82
2	.62	1.82
3	.62	2.60
4	.62	3.72
5	.62	3.72
6	.62	3.72

Fig. 12. Optimized performance on plume traversal task of Webots trials across group size in larger arena. Higher values indicate better performance.

Fig. 13. Normalized time and distance across group size for optimized Webots trials in large arena. Lower values indicate better performance.

VI. Conclusion

In this paper we have described a distributed algorithm for solving the full odor localization task, and shown that group performance can exceed that of a single robot. We have demonstrated that one subtask, plume traversal, can be successfully accomplished by real robots. Furthermore, we have established that an embodied simulator can accurately replicate the real robots results, and shown that it can be used to optimize performance across group size. Thus, it is useful not only for improving real world odor localization, but also for quantitatively characterizing the influence of group size on task performance under the constraints of the SI architecture.

Furthermore, our data indicates that in this search task, as completion time becomes more valued over total energy used, larger group sizes become optimal. Also, larger movement parameter values can result in more efficient search for larger group sizes, and larger group sizes are favored as it becomes more difficult to maintain plume contact. These types of observations may eventually result in formulae for narrowing the range of possible optimal parameter values based on odor localization task parameters, which will decrease search costs.

Our eventual goal, achievement of near optimal performance on the full odor localization task in the real world, will require efficient search of a large parameter space through a combination of accurate simulation and efficient off-line machine learning techniques.

Acknowledgments

We would like to thank Owen Holland, Alan Winfield, Sanza Kazadi, Jim Pugh, Robert Enright, Ladd Van Tol, and Andrew Lundsten for their contributions on various hardware and software aspects of the project. We would also like to thank our collaborators in the DARPA-ONR Chemical Plume Tracing Program for their valuable input. This work is supported in part by the Center for Neuromorphic Systems Engineering as part of the National Science Foundation Engineering Research Center Program under grant EEC-9402726. Support has also been received from the DARPA Dog's Nose program under grant DAAK60-97-K-9503. This work is also supported in part by the Office of Naval Research under grant N00014-98-1-0821 "Chemical Plume Tracing Via Biological Inspiration and Electronic Olfaction", and by the Army Research Office under MURI grant DAAG55-98-1-0266 "Understanding Olfaction: From Detection to Classification". Adam Hayes is supported by a National Science Foundation Graduate Research Fellowship.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, Oxford University Press, New York, US, 1999.

[2] R. Beckers, O.E. Holland, and J.L. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics," in Proc. of the Fourth Workshop on Artificial Life, R. Brooks and P. Maes, Eds., Boston, MA, 1994, pp. 181-189, MIT Press.

[3] A. Martinoli, A. J. Ijspeert, and F. Mondada, "Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots," Robotic and Autonomous Systems, vol. 29, pp. 51-63, 1999.

[4] O.E. Holland and C. Melhuish, "Stigmergy, self-organization, and sorting in collective robotics," Artificial Life, vol. 5, pp. 173-202, 1999.

[5] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Comparing distributed exploration strategies with simulated and real autonomous robots," in Proc. of the fifth Int. Symp. on

Distributed Autonomous Robotic Systems DARS-2000, L. E. Parker, G. Bekey, and J. Barhen, Eds., Knoxville, Tennessee, October 2000, pp. 261-270, Springer Verlag.

[6] A. J. Ijspeert, A. Martinoli, A. Billard, and L. M. Gambardella, "Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment," Autonomous Robots, Vol. 11, No. 2, pp. 149-171, 2001.

[7] C. R. Kube and E. Bonabeau, "Cooperative transport by ants and robots," Robotics and Autonomous Systems, vol. 30, pp. 85-101, 2000.

[8] M. J. B. Krieger and J.-B. Billeter, "The call of duty: Self-organised task allocation in a population of up to twelve mobile robots," Robotics and Autonomous Systems, vol. 30, pp. 65-84, 2000.

[9] J. Barhen and D. R. Protopopescu, "Trust: A deterministic algorithm for global optimization," Science, vol. 276, pp. 1094-1097, 1997.

[10] R. A. Jarvis, "Optimization strategies in adaptive control: A selective survey," IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-5, no. 1, pp. 83-94, 1975.

[11] R. S. Sutton and A. G. Barto, Reinforcement Learning, MIT Press, Cambridge, MA, 1998.

[12] D. Floreano and Mondada F., "Evolution of homing navigation in a real mobile robot," IEEE Transactions on Systems, Man and Cybernetics, vol. 26, pp. 396-407, June 1996.

[13] S. Yakowitz and J. Mai, "Methods and theory for off-line machine learning," IEEE Automatic Control, vol. 40, no. 1, pp. 161-165, January 1995.

[14] S. Yakowitz, P. L'Ecuyer, and F. Vazquez-Abad, "Global sthocastic optimization with low-dispersion point sets," Operations Research, vol. 48, no. 6, pp. 939-950, November-December 2000.

[15] R. T. Carde and A. Mafra-Neto, "Effect of pheromone plume structure on moth orientation to pheromone," in Perspectives on Insect Pheromones. New Frontiers, R. T. Carde and A. K. Minks, Eds., pp. 275-290. Chapman and Hall, N.Y., 1996.

[16] J. H. Belanger and M. A. Willis, "Adaptive control of odor guided locomotion: Behavioral flexibility as an antidote to environmental unpredictability," Adaptive Behavior, vol. 4, pp. 217-253, 1996.

[17] U. Bhalla and J. M. Bower, "Multi-day recording from olfactory bulb neurons in awake freely moving rats: Spatial and temporally organized variability in odorant response properties," J. of Computational Neuroscience, vol. 4, pp. 221-256, 1997.

[18] J. Atema, "Eddy chemotaxis and odor landscapes: Exploration of nature with animal sensors," Biological Bull., vol. 191, pp. 129-138, 1996.

[19] M. J. Weissburg, "From odor trails to vortex streets: Chemo and mechanosensory orientation in turbulent and laminar flows," in Orientation and Communication in Arthropods, M. Lehrer, Ed. Birkhauser, Basel, 1997.

[20] T. Nakamoto, H. Ishida, and T. Moriizumi, "A sensing system for odor plumes," Analytical Chemistry, vol. 71, no. 15, pp. 531A-537A, August 1999.

[21] S. Kazadi, R. Goodman, D. Tsikata, and H. Lin, "An autonomous water vapor plume tracking robot using passive resistive polymer sensors," Autonomous Robots, vol. 9, no. 2, pp. 175-188, 2000.

[22] C. T. David, J. S. Kennedy, J. S. Ludlow, and J. N. Perry, "A re-appraisal of insect flight towards a point source of wind-borne odor," Journal of Chemical Ecology, vol. 8, pp. 1207-1215, 1982.

[23] N. J. Vickers and T. C. Baker, "Reiterative responses to single strands of odor promote sustained upwind flight and odor source location by moths," Proceedings of the National Academy of Sciences USA, vol. 91, pp. 5756-5760, 1994.

[24] D. R. Webster, S. Rahman, and L. P. Dasi, "On the usefulness of bilateral comparison to tracking turbulent chemical odor plumes," Limnology and Oceanography, 2001, To appear.

[25] A.F.T. Winfield and O.E. Holland, "The application of wireless local area network technology to the control of mobile robots," Microprocessors and Microsystems, vol. 23, pp. 597-607, 2000.

[26] F. Mondada, E. Franzi, and P. Ienne, "Mobile robot miniaturization: A tool for investigation in control algorithms," in Proc. of the Third International Symposium on Experimental Robotics ISER-93, T. Yoshikawa and F. Miyazaki, Eds., Kyoto, Japan, 1993, pp. 501-513, Springer Verlag.

[27] G. Caprari, P. Balmer, R. Piguet, and R. Siegwart, “The autonomous microbot ‘alice’: a platform for scientific and commercial applications,” in Proc. of the Ninth Int. Symp. on Micromechatronics and Human Science, Nagoya, Japan, November 1998, pp. 231-235.

[28] M. S. Freund and N. S. Lewis, "A chemically diverse conducting polymer-based electronic nose," Proceedings of the National Academy of Sciences USA, vol. 92, pp. 2652, 1995.

[29] O. Michel, "Webots: Symbiosis between virtual and real mobile robots," in Proceedings of the First International Conference on Virtual Worlds, VW'98, Paris, France, July 1998, pp. 254-263, Springer Verlag.

[30] A. Martinoli, A. J. Ijspeert, and L. G. Gambardella, "A probabilistic model for understanding and comparing collective aggreagtion mechanisms," in Proc. of the Fifth Int. European Conf. on Artificial Life ECAL-99, Lecture Notes in Computer Science, D. Floreano, F. Mondada, and J.-D. Nicoud, Eds., pp. 575-584. Springer Verlag, Lausanne, Switzerland, September 1999.

[31] A. Martinoli, Swarm Intelligence in Autonomous Collective Robotics: From Tools to the Analysis and Synthesis of Distributed Control Strategies, Ph.D Thesis Nr. 2069, EPFL, Lausanne, Switzerland, October 1999.

[32] D. R. Webster, P. J. W. Roberts, and L. Ra'ad, "Simultaneous dptv/plif measurements of a turbulent jet," Experiments in Fluids, vol. 30, pp. 65-72, 2001.

[33] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," Autonomous Robots, vol. 8, no. 3, 2000.

[34] L. E. Parker, "Lifelong adaptation in heterogeneous multi-robot teams: Response to continual variation in individual robot performance," Autonomous Robots, vol. 8, no. 3, pp. 239-267, 2000.

[35] M. J. Mataric, "Reinforcement learning in the multi-robot domain," Autonomous Robots, vol. 4, no. 1, pp. 73-83, March 1997.

[36] A. Murciano, J. R. Millan, and Z. Zamora, "Specialization in multi-agent systems through learning," Biological Cybernetics, vol. 76, pp. 375-382, 1997.

[37] T. Balch, Behavioral Diversity in Learning Robot Teams, Ph.D Thesis, College of Computing, Georgia Institute of Technology, December 1998.

[38] C. Versino and L. M. Gambardella, "Learning real team solutions," in DAI Meets Machine Learning, Lecture Notes in Artificial Intelligence, G. Weiss, Ed., pp. 298-311. Springer Verlag, Berlin, 1997.

[39] J. R. Koehler, A. A. Puhalskii, and B. Simon, "Estimating functions evaluated by simulation: A bayesian/analytic approach,"
The Annals of Applied Probability, vol. 8, no. 4, pp. 1184-1215, 1998.

Captions

Fig. 1. Spiral Surge odor localization behavior.

Fig. 2. A Moorebot equipped with wind, odor, and proximity sensors, as well as markings for overhead tracking.

Fig. 3. Plume hits received by 6 real robots over 1 hour while performing a random walk behavior.

Fig. 4. Average wind direction in plume traversal arena as measured by the real robots. Plume source at upper right. Arrow lengths are proportional to the mean flow magnitude at the tail of each arrow.

Fig. 5. Real robot arena as seen from overhead camera.

Fig. 6. Webots plume traversal arena with average plume intensity map.

Fig. 7. Plume hits received by 6 simulated robots over 1 hour.

Fig. 8. Normalized time across group size for real robot trials. Lower values are better.

Fig. 9. Normalized distance across group size for real robot trials. Lower values are better.

Fig. 10. Performance across group size for real robot trials. Higher values indicate better performance.

Fig. 11. Performance of real robot and Webots trials across group size. Higher values indicate better performance.

Fig. 12. Optimized performance on plume traversal task of Webots trials across group size in larger arena. Higher values indicate better performance.

Fig. 13. Normalized time and distance across group size for optimized Webots trials in large arena. Lower values indicate better performance.

Figure 1

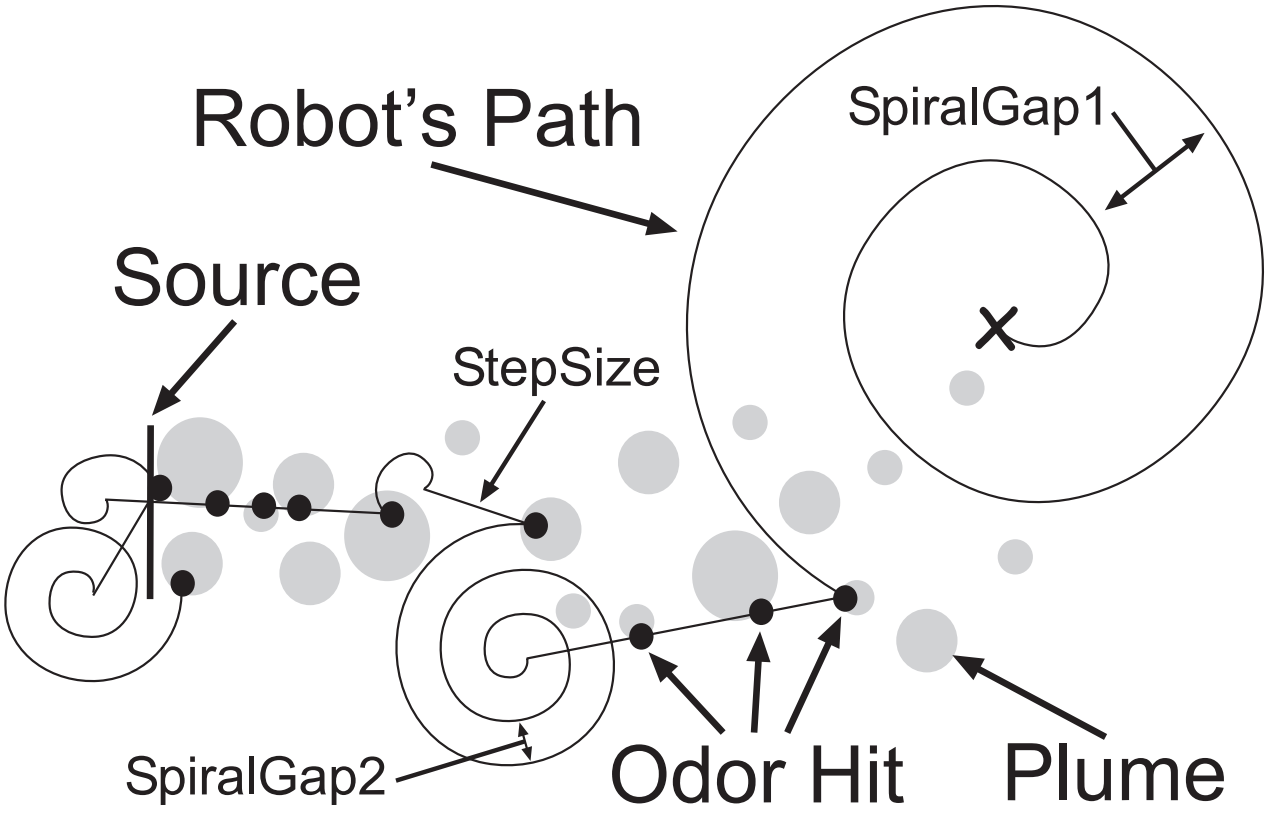

Robot's Path
SpiralGap1
Source
StepSize
SpiralGap2
Odor Hit
Plume

Figure 2

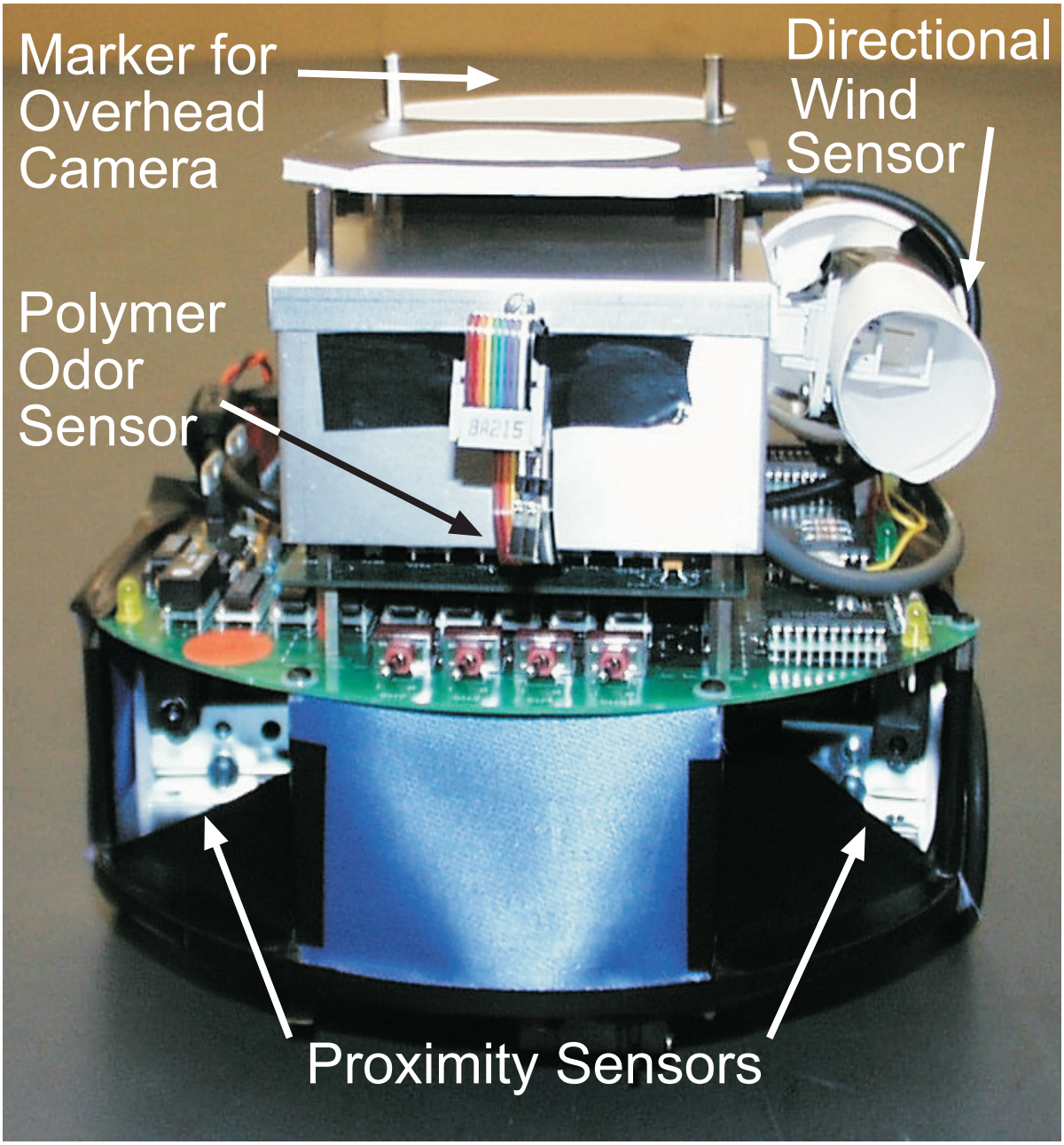

Marker for Overhead Camera
Directional Wind Sensor
Polymer Odor Sensor
Proximity Sensors

Figure 3

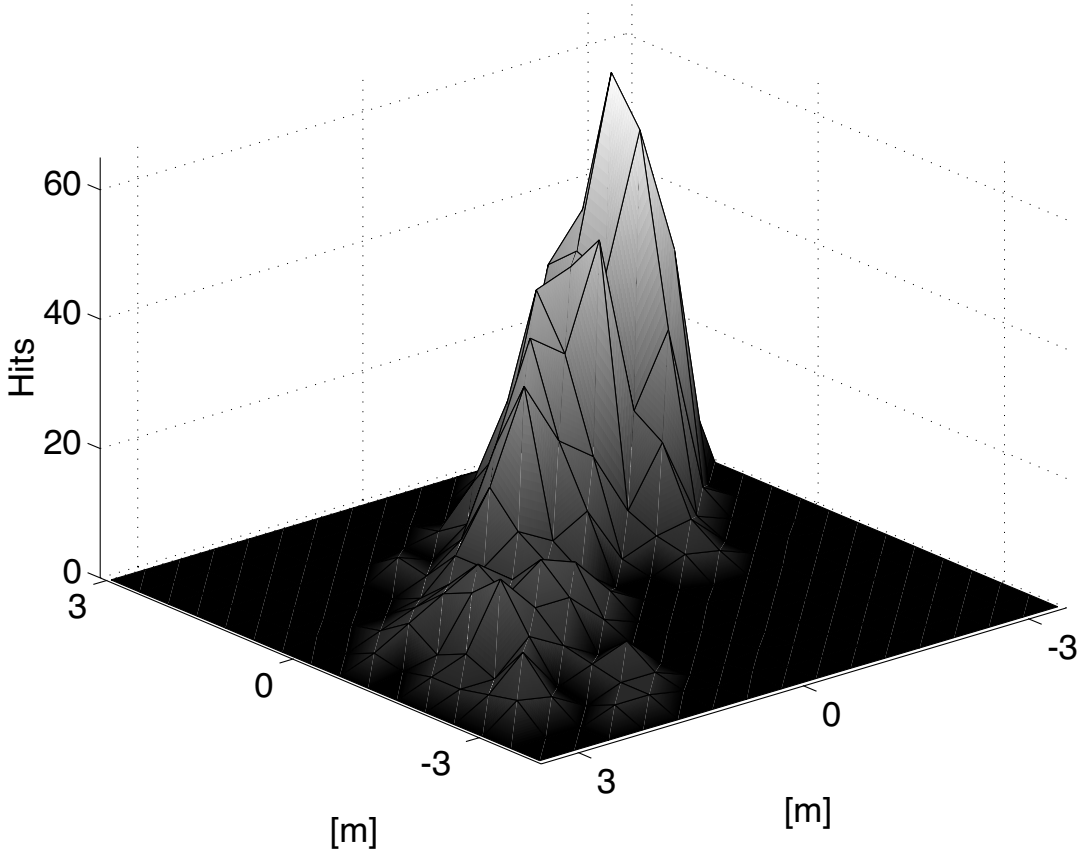


Figure 4

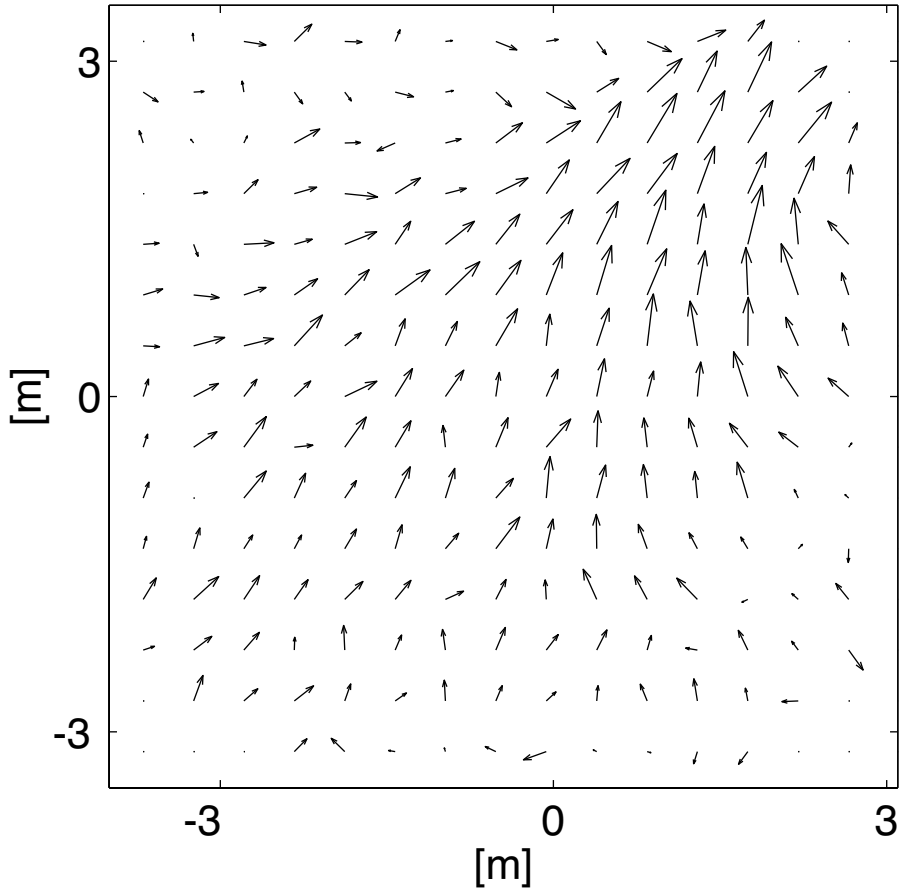


Figure 5

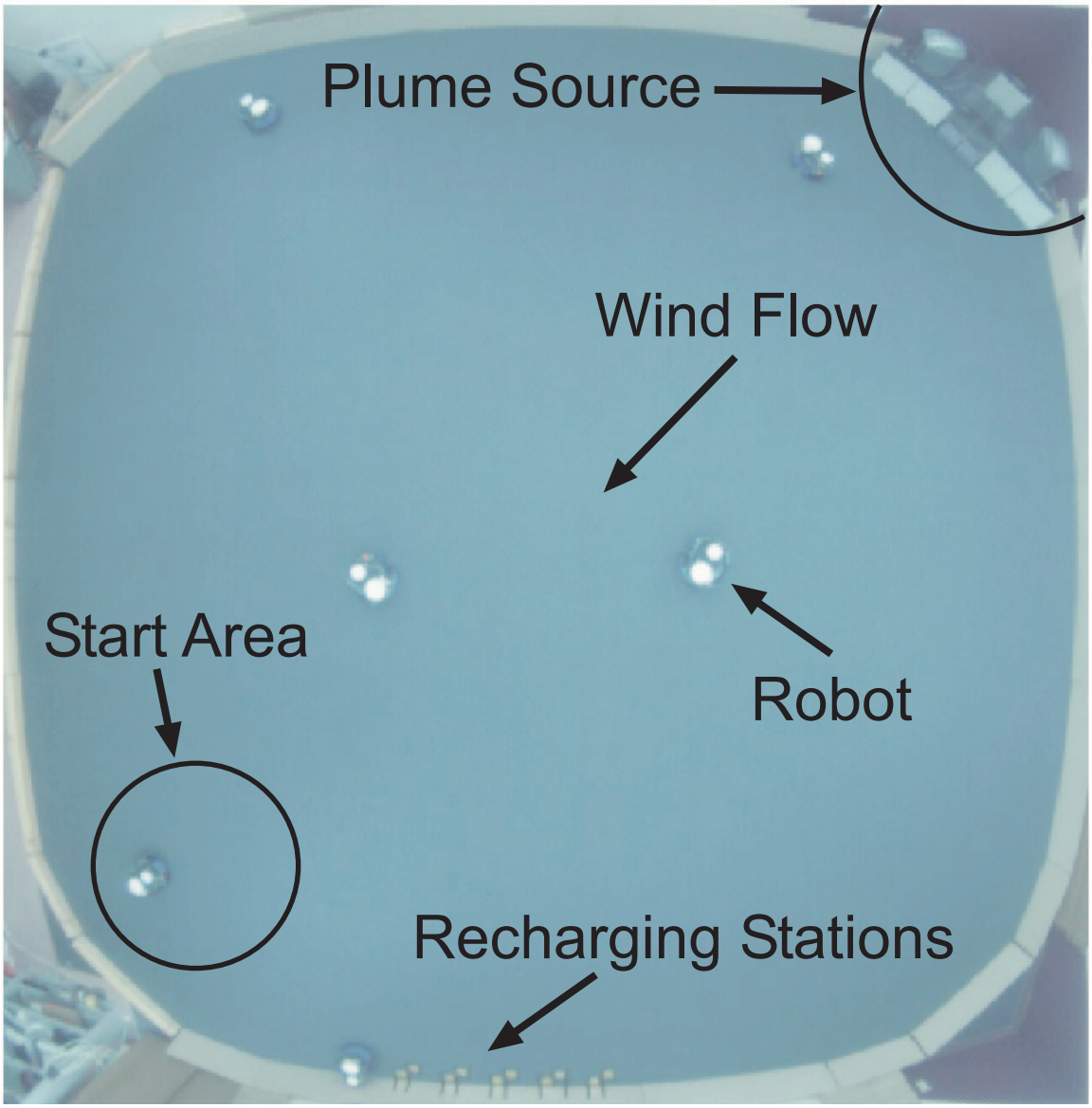

Plume Source
Wind Flow
Start Area
Robot
Recharging Stations

Figure 6

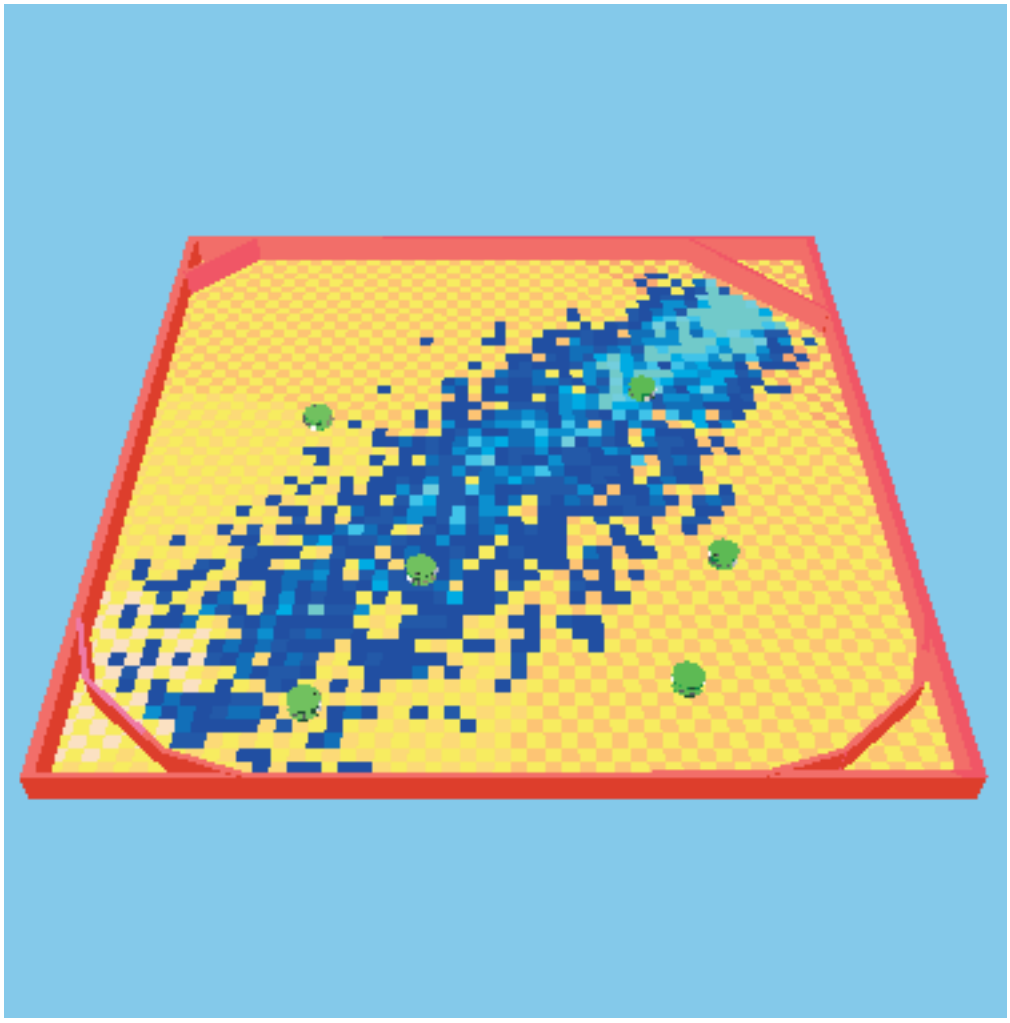


Figure 7

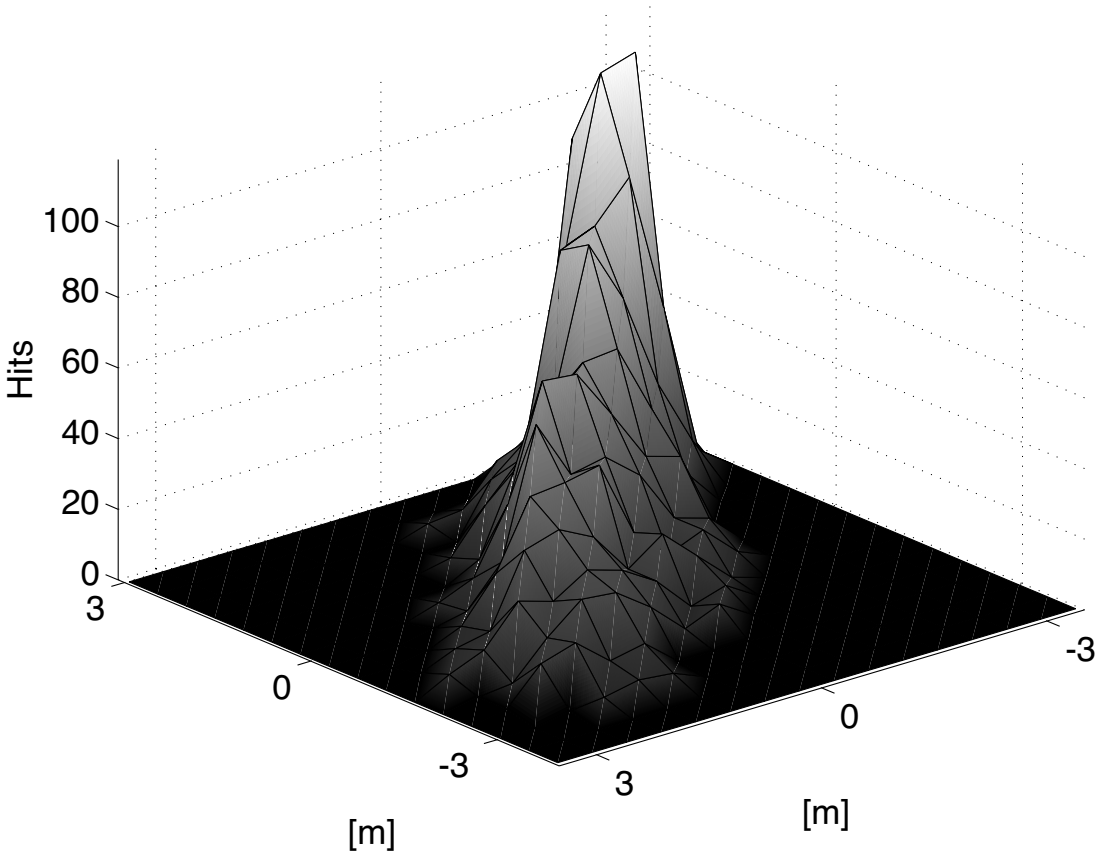


Figure 8

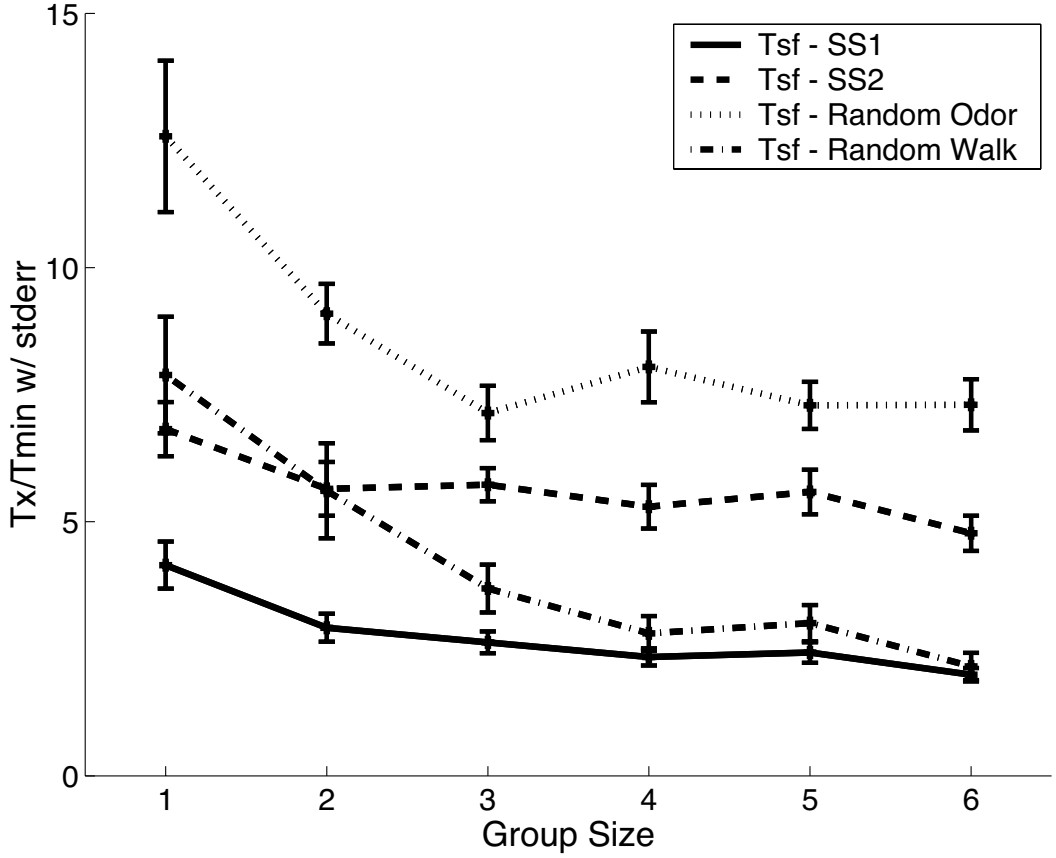

Tsf - SS1
Tsf - SS2
Tsf - Random Odor
Tsf - Random Walk
Tx/Tmin w/ stderr
Group Size
0
5
10
15
1
2
3
4
5
6

Figure 9

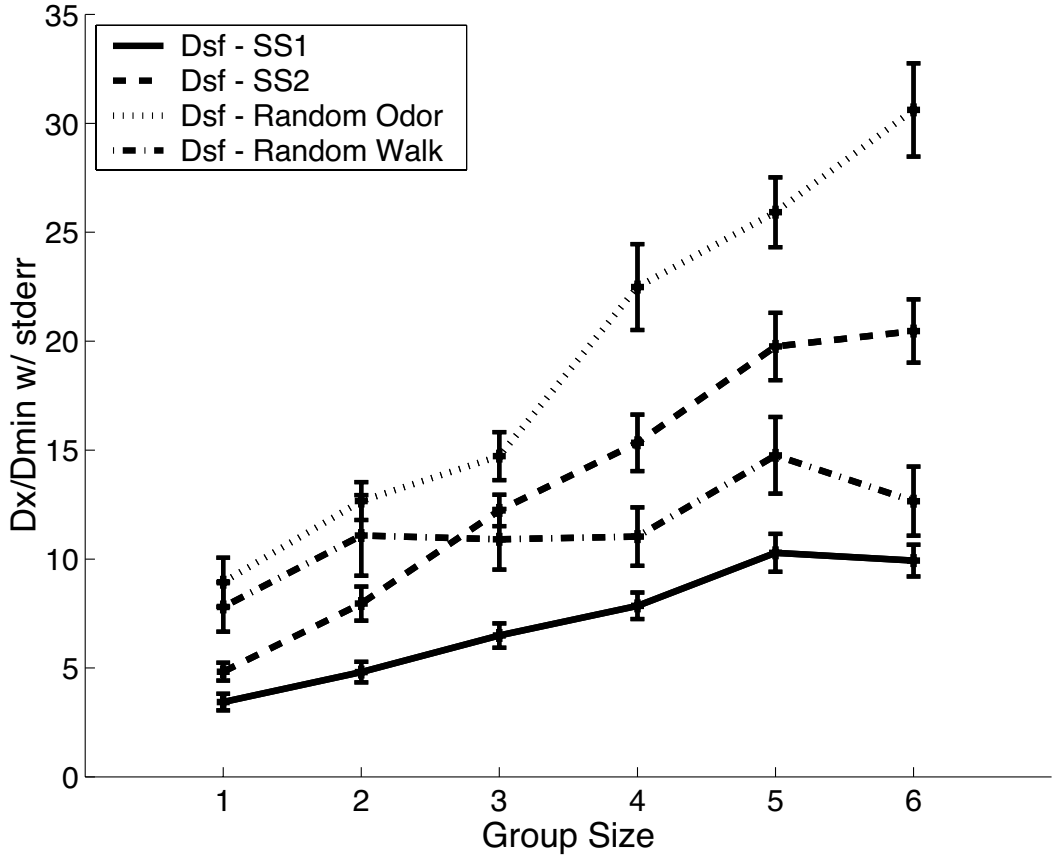

Dsf - SS1
Dsf - SS2
Dsf - Random Odor
Dsf - Random Walk
Dx/Dmin w/ stderr
Group Size
0
5
10
15
20
25
30
35
1
2
3
4
5
6

Figure 10

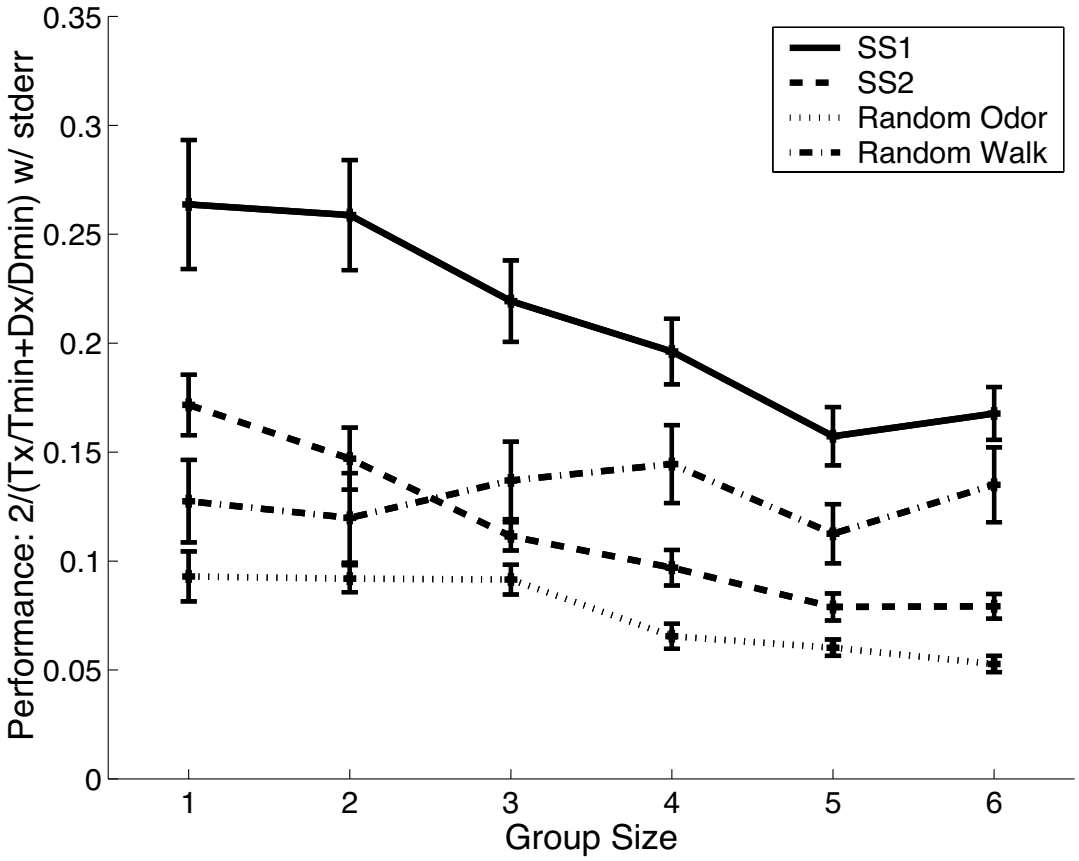

SS1
SS2
Random Odor
Random Walk
Performance: 2/(Tx/Tmin+Dx/Dmin) w/ stderr
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1
2
3
4
5
6
Group Size

Figure 11

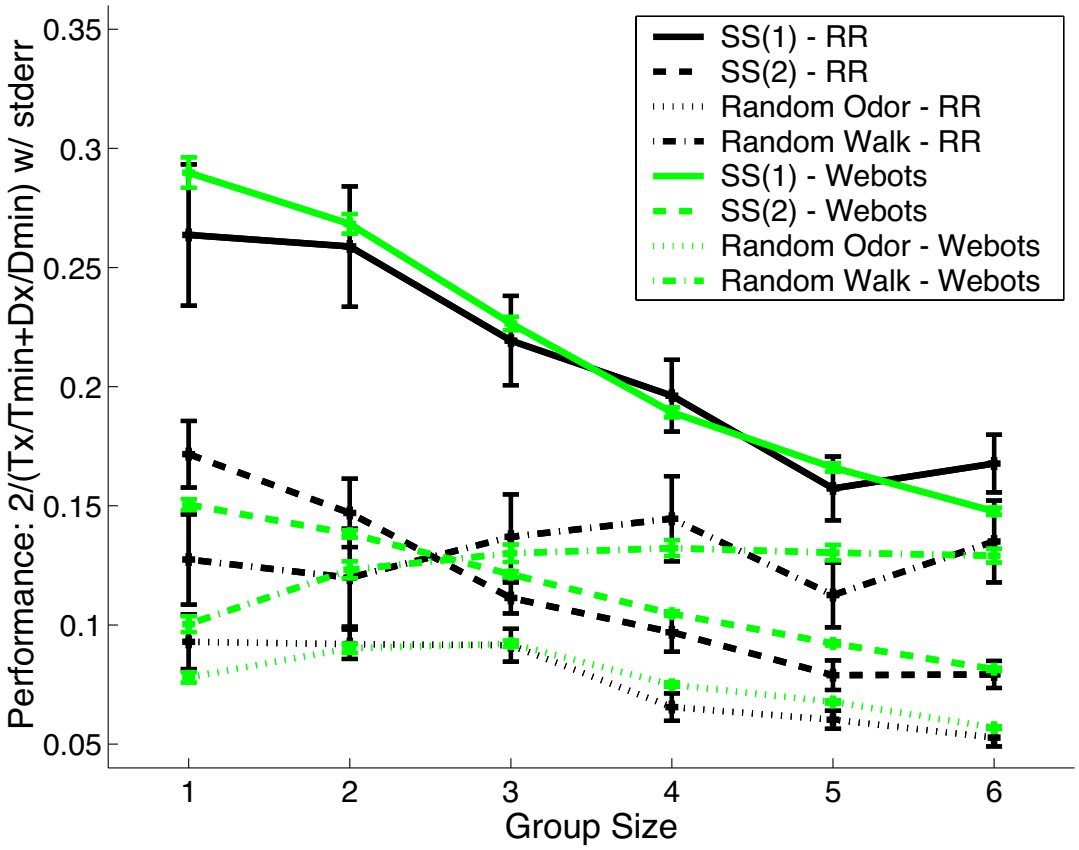

SS(1) - RR
SS(2) - RR
Random Odor - RR
Random Walk - RR
SS(1) - Webots
SS(2) - Webots
Random Odor - Webots
Random Walk - Webots
Performance: 2/(Tx/Tmin+Dx/Dmin) w/ stderr
0.35
0.3
0.25
0.2
0.15
0.1
0.05
1
2
3
4
5
6
Group Size

Figure 12

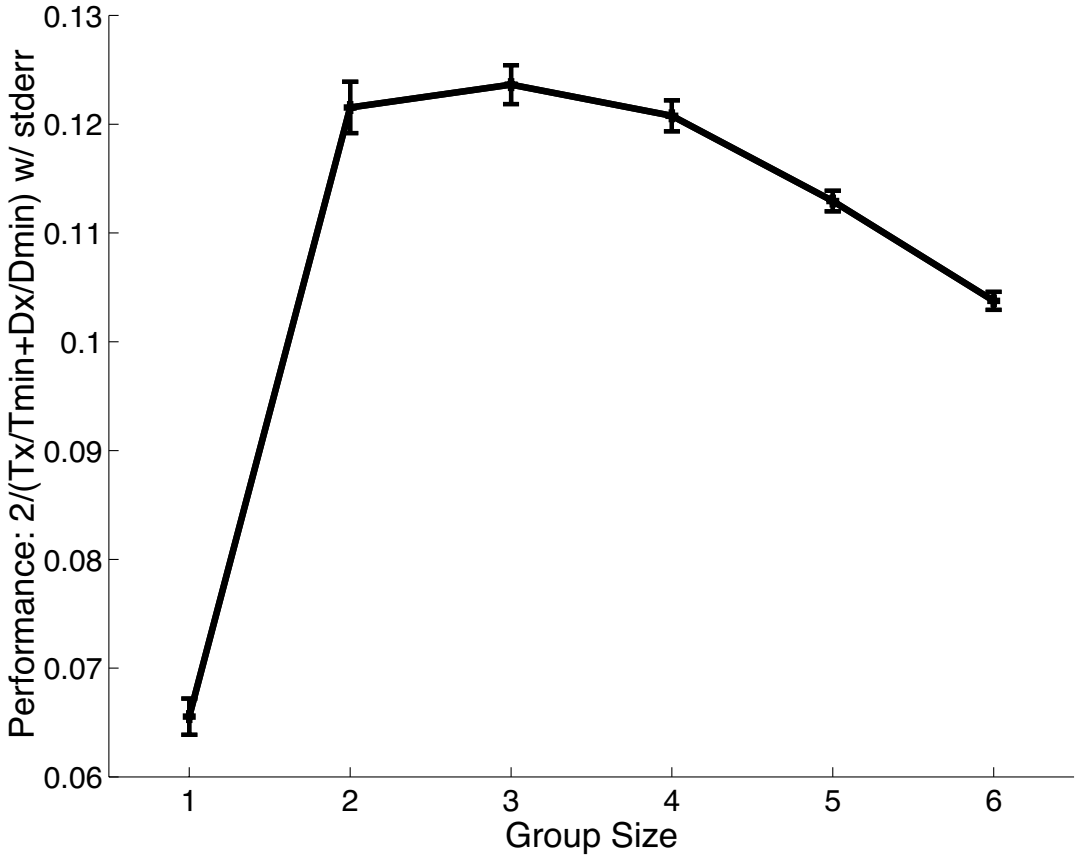
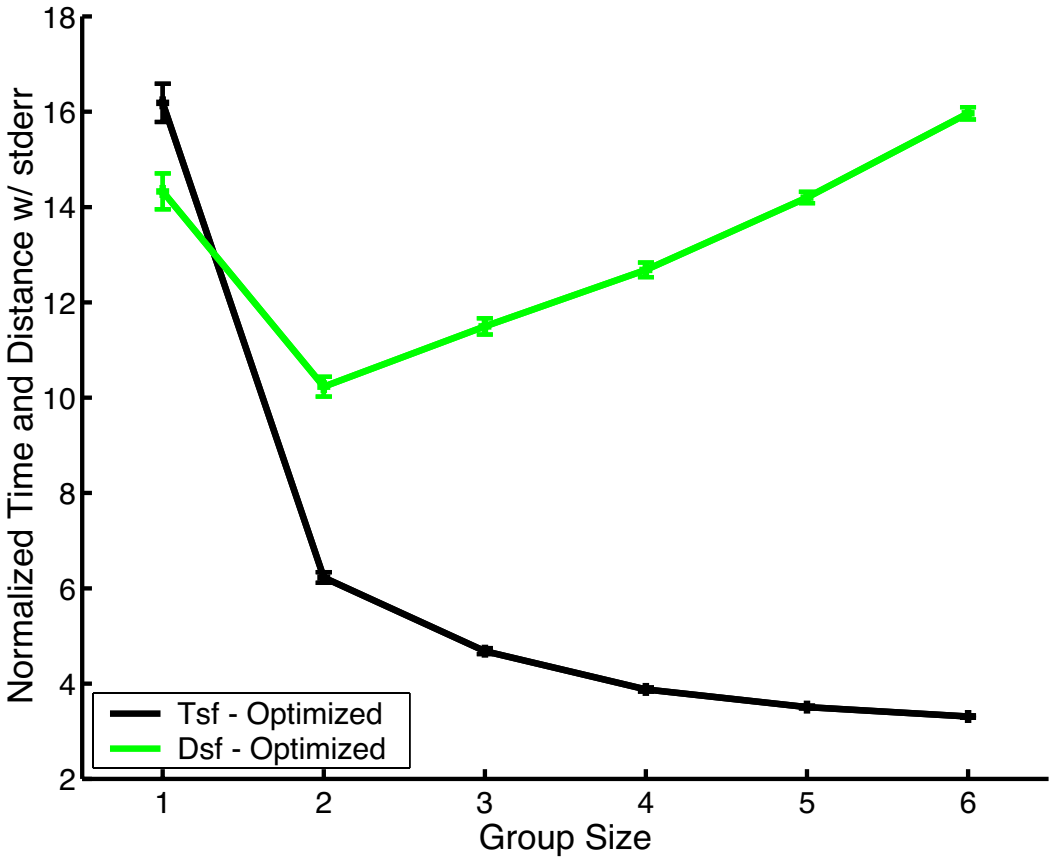

Performance: 2/(Tx/Tmin+Dx/Dmin) w/ stderr
0.13
0.12
0.11
0.1
0.09
0.08
0.07
0.06
1
2
3
4
5
6
Group Size

Figure 13


Normalized Time and Distance w/ stderr
Group Size
Tsf - Optimized
Dsf - Optimized
18
16
14
12
10
8
6
4
2
1
2
3
4
5
6