1

Pervasive *Sensor-less* Networks for Cooperative Multi-Robot Tasks

Keith J. O'Hara and Tucker R. Balch
{kjohara, tucker}@cc.gatech.edu

The BORG Lab
College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332–0250

Summary. A number of researchers are investigating the use of embedded sensor networks to facilitate mobile robot activities. Previous studies focus individual tasks (e.g. navigation to a goal) using networks of several to tens of expensive ($\approx$ \$100) nodes placed by the robots themselves or in predetermined geometric grids. In this work we explore the use of tens up to hundreds of simple and cheap ($\approx$ \$10) sensor-less nodes placed arbitrarily to support a complex multi-robot foraging task. Experiments were conducted in a multi-robot simulation system. Quantitative results illustrate the sensitivity of the approach to different network sizes, environmental complexities, and deployment configurations. In particular, we investigate how performance is impacted by the density and precision of network node placement.

1.1 Introduction and Related Work

We are interested in the application of low-cost, pervasively distributed network nodes to support cooperative multi-robot tasks. In this work we consider a heterogeneous system composed of small, embedded, immobile sensor-less communication nodes and larger mobile robots equipped with sensors and manipulators. The embedded network serves as a pervasive communication and computation fabric, while the mobile robots provide sensing and actuation. We refer to the embedded nodes as forming a *sensor-less* network to distinguish the approach from those where the network nodes also have sensors. In our work the embedded nodes provide only modest computation and communication for the team.

As noted above, we depart from the usual approach where the embedded nodes are equipped with sensors. There are a number of arguments in favor of sensor-less embedded networks. First, the cost and power requirements for simpler embedded nodes is lower, thus enabling us to distribute more of them. Second, it is likely that even for a network with sensor nodes, certain activities for which the nodes do not have sensors will need to be conducted. For

instance, we may want to utilize a network that has already been deployed (e.g. wildfire monitoring), to support a new application (e.g. search and rescue). Unanticipated applications can be addressed on deployed networks by equipping the mobile robots with appropriate sensors. In this way application-specific hardware is concentrated on the smallest portion of the team – the mobile platforms.

The algorithm for guiding the mobile robots essentially works as a distributed variant of the popular wave-front path planning algorithm, or a breadth-first search from the goal, propagating paths from the goal location. The embedded nodes make up the vertices of the path planning graph, and the network connections between them are the edges of the graph. Mobile robots can then use reactive navigation to traverse the graph by visiting the vertices (i.e. the embedded nodes) to the goal. The sensor-less network creates navigation networks for supporting mobile robots in various tasks such as coverage, recruitment, and path planning. We demonstrate all three of these distributed skills in a multi-robot foraging task. We show that a pervasive network of embedded sensor-less nodes can enable a team of mobile robots to accomplish complex tasks effectively. We analyze the sensitivity of the approach to different team sizes, environmental complexities, and deployment configurations.

Parunak et al developed a technique for coordinating multiple unmanned air vehicles (UAVs) using synthetic pheromones. [1, 2] Inspired by pheromone communication in insects, they create potential fields for guiding the UAVs around threats to goal locations in a distributed manner. We do not assume the embedded nodes are arranged uniformly in any structure. We also rely on a distributed dynamic programming solution to the path planning problem, rather than an approach based on the dynamics of insect pheromones.

Like Parunak, Payton et al present an approach for large scale multi-robot control referred to as "Pheromone Robotics" inspired by biology. [3] They use a system based on virtual pheromones, by which a team of mobile robots use short-range communication to accomplish cooperative sensing and navigation. In Payton's work "virtual pheromones" are communicated over an ad hoc network to neighboring robots. In contrast, in our approach information is not distributed by the mobile robots, but rather by the relatively static, embedded nodes scattered throughout the environment.

Both Batalin et al [4, 5] and Li et al [6] have developed similar approaches using heterogeneous teams composed of mobile nodes and an embedded network. The network of embedded nodes, creates a "Navigation field" [4], which mobile nodes can use to find the their way around. They differ in how they compute this navigation field. Batalin et al use Distributed Value Iteration [4]. In their approach, the embedded nodes use estimated transition probabilities between nodes to compute the best direction to suggest to a mobile robot for moving between a start and goal node. These transition probabilities are established during deployment and both the robots and sensor nodes have synchronized direction sensors (e.g. digital compass). Our approach does not

require the nodes to store transition probabilities, instead we rely on the communication network to establish the navigation paths. Also, in our approach the mobile robots only need a local sense of direction in order to move toward the correct embedded node. Neither the robots or the embedded nodes need any shared sense of direction.

Li et al are able to generate an artificial potential field for navigation based on the obstacles and goals sensed by the network. [6] This potential field is guaranteed to deliver the mobile node to the goal location via an danger-free (obstacle-free) path. The field is created by the embedded nodes propagating goal-ness or danger to neighboring nodes. In our approach the embedded nodes do not have sensors, this capability is provided by the mobile nodes, and thus can not sense obstacles directly. We assume the obstacles are sensed indirectly by the resulting communication topology. The later three of these approaches, as well as ours, use distributed dynamic programming [7] to create the navigation field.

Koenig [8] and Wagner [9, 10] also devise some related methods for doing parallel coverage using simple ant robots that communicate indirectly by leaving indicators in the environment. Batalin et al [5] also use communication nodes as "markers" in aiding mobile robots in the exploration problem. The embedded nodes offer a suggested un-explored direction for the mobile robots to follow. Unlike our approach, their embedded nodes do not communicate with each other, but only to mobile robots.

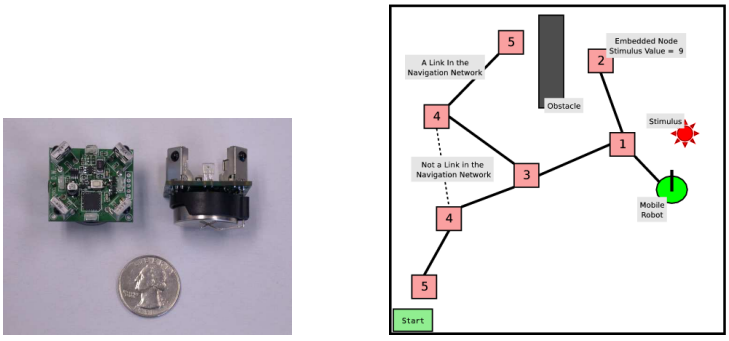
1.2 Problem Statement and Assumptions

Our system is composed of mobile robots with sensors and actuators supported by an embedded immobile network of nodes without environmental sensors. We assume the embedded network nodes have the following capabilities:

- **Limited computation and memory,** on the order of a PIC microprocessor with 2K ROM and 256 bytes of RAM operating at 4 MHz.
- **Short range communication** with adjacent nodes up to 4 meters distant.
- **Communication is blocked by navigation obstacles.**

We assume the robots and embedded nodes communicate using a short-range medium that is occluded by the same objects that occlude navigation (e.g. walls). Line of sight between nodes implies open space for navigation.We have implemented a hardware platform to these specifications, the GNAT (see Fig. 1(a)). The GNAT is a low-power, omni-directional, infrared device costing about 30 dollars to build. The mobile robots in our system are somewhat more capable. We assume they support:

- Communication with embedded nodes;
- Relative bearing estimation to nearby embedded nodes;



(a) The GNAT (b) Navigation Network

Fig. 1.1. (a) The GNAT is a low-power, omni-directional, infrared device. (b) An illustration of a navigation network.

- Local attractor and obstacle sensing; (e.g. food objects);
- Attractor object grasping.

These robot capabilities are sufficient for cooperative foraging in the presence of an embedded network.

Significantly, there are a few assumptions we do not make. In particular, **we do not assume localization or mapping capabilities** on the part of the robots or the embedded nodes. No mobile robots or embedded nodes are expected to perform localization or mapping. Furthermore **we do not assume the environment is static**. Obstacles to navigation can appear and disappear. We expect the network to automatically adapt to dynamic conditions.

In this work do not address the deployment of the embedded nodes. We assume they have already been placed in the environment, but their positions are unknown and the uniformity of their placement can vary. In fact, one objective of this work is to assess the impact on performance with respect to different network sizes, environment complexities, and deployment configurations.

Given the system of robots and network nodes described above, we would like to solve a multi-robot foraging problem. Foraging is a well-studied, canonical multi-robot task [11, 12]. In this task a robot team is initialized at a "homebase" location, from which they should begin to explore the environment in search of attractor (food) objects. Once a cache of attractor objects is discovered, this information should be disseminated to the other robots, along with a means for them to navigate to the cache. Finally, all of the attractor objects should be collected by the robots and delivered to homebase. We have decomposed the overall problem into the following sub-problems:

- Cooperative coverage: enable a team of mobile robots to completely explore the area covered by embedded nodes. For efficiency, robots should avoid traveling through areas already explored by other robots.
- Recruitment: alert the team to new, critical information. In the context of a foraging task, the discovery of a food cache is an example recruitment situation.
- Path planning: Without requiring localization capabilities, provide an efficient route for each robot, located anywhere in the environment, to a goal location.

1.3 Approach

The sensor-less network creates navigation networks for supporting mobile robots in various tasks such as coverage, recruitment, and path planning. We use navigation networks to accomplish three different steps in the task: 1) directing the robots to visit uncovered areas, 2) directing the robots to a discovered attractor cache, and 3) directing them home. Navigation networks for each of these tasks are present in the sensor-less network simultaneously. A mobile robot can then follow whichever navigation network corresponding to it's current sub-task goal. A navigation network is illustrated in Fig. 1(b). We are able to use navigation networks to complete the multi-robot foraging task in complex environments without mapping or localization.

We follow Payton's virtual pheromone technique [3] and assume the communication paths are similar to the navigation paths, and use this to propagate navigation information. By using a short-range communication medium that is occluded by obstacles to navigation, the communication paths carve out free-space. As also pointed out by Payton [3] and Li [6], this results in a kind of distributed physical path-planning. To create a navigation network for a particular goal we use a distributed dynamic programming approach; specifically, we apply the distributed Bellman-Ford algorithm. The Bellman-Ford equation [13] for finding the shortest path from i to j is:

$$D(i,j) = \min_{k \in neighbors} d(i,k) + D(k,j)$$

Where $D(i,j)$ is the path cost from i to j, and $d(i,k)$ is the distance between i and k. It can be used to find the shortest path to a destination from all nodes. The distributed version of Bellman-Ford was created for network routing protocols [14]. In the distributed network routing version, neighbors share their path costs and the distance between nodes is usually measured in hops. We use distributed Bellman-Ford to effectively create a tree of shortest paths from every node to the goal – this tree is the navigation network. The embedded network can be thought of as "routing" the mobile robots to their destination. However, note that the embedded nodes do not know the global, or local, position of their neighbors, so they are not directing the robot in any

direction. Instead, the mobile robot greedily approaches the lowest-valued node currently in its communication range. As it closes in on the node it will come within communication range of that node's parent. The robot continues this until it comes within sensing distance of the goal.

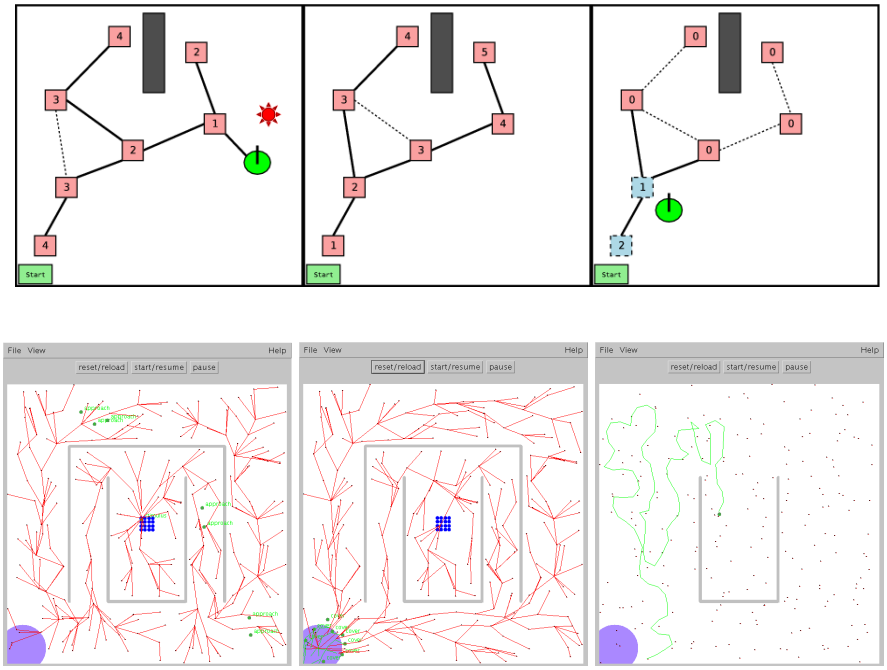
As the mobile robots discover attractors, they broadcast this information to neighboring embedded nodes and navigation trees are created with roots near the attractor. As the information is propagated throughout the network by the embedded nodes, the hop-count, or path-cost, increases. Any mobile robot can then approach the network, access the relevant navigation network, and descend to the root of the tree, eventually reaching the original goal.

Using a static sensor-less network provides several advantages over fully mobile networks and static sensor networks. The first benefit is cost. By having the bulk of our system be composed of cheap communication and computation nodes, we lower the cost and power requirements of the entire team. In addition, the embedded network can be used generally, for instance, when the desired sensor for the application is not known beforehand. Another related advantage is that the bulk of the application-specific hardware is concentrated in the smallest portion of our team – the mobile platforms – allowing the network to be used in a general manner. Another advantage is that the majority of the system is relatively static and connected. We say relatively because the topology can indeed change, but we assume for the most part, the network will be fully connected and fairly static. When using all mobile nodes, as done by Payton [3], we must assure the network stays connected. Because the propagation algorithms use a distributed dynamic programming solution, they can fail when the network becomes disconnected for long periods of time.

The attractor navigation network allows a robot from anywhere in the environment to find a path to an attractor that was sensed by another mobile robot. An illustration of the navigation network for a discovered food-source is shown in Fig. 2(b). It is obvious that if we filled the space with embedded nodes arranged in a grid, and gave the nodes range sensors, we would see a picture very similar to many grid-based planning approaches. The path planning space is approximated by the communication network. But how many nodes are required for this approximation to hold, and how uniformly do they have to be arranged? We address these questions in the experiments below.

The home navigation network is an instance of an attractor navigation network, with the homebase being the attractor. This creates a tree rooted at the homebase, assuring the robots can return home. An illustration is shown in Fig. 2(c).

Next, we consider a mechanism for building a coverage navigation network. It is a straightforward extension of the attractor navigation network, where each unvisited node is an attractor. The mobile nodes are then offered paths to reach the closest unvisited nodes. This approach assures all nodes will be visited. If a node has been visited, it uses the default scheme of propagating one of its neighbor's values. This results in the visited embedded nodes directing the mobile robots into unexplored areas.



(b) Attractor Navigation Network

(c) Home Navigation Network

(d) Coverage Navigation Network

Fig. 1.2. The components of the foraging task. The illustrations depict the embedded nodes as squares, labeled with their value, the mobile robot as the green circle, and the attractor cache as the red star. The solid lines connecting the nodes make up the navigation network, the dotted lines are connections that aren't included in the navigation network. For the screenshots of the TeamBots simulation environment, the blue circles in the center represent the food source, the blue circle in the bottom left corner is the robots' starting position, and the gray lines are walls. (a) An example navigation network for the attractor cache. (b) The homebase navigation network. (c) The coverage navigation network.

Although the coverage solution generated is not optimal in the sense of shortest circuit to visit all the nodes (i.e. the traveling salesman problem) it does assure all nodes are visited. Depending on the configuration of the embedded nodes, this can assure systematic coverage of the terrain, even though neither the robots or the embedded nodes have any localization capabilities or a map. In addition, both algorithms can be used in dynamic coverage scenarios by changing their state to unvisited after a certain amount of time has passed. The algorithm works well for both single and multi-robot exploration. An illustration of the coverage algorithm is given in Fig. 2(d).

1.4 Experiments

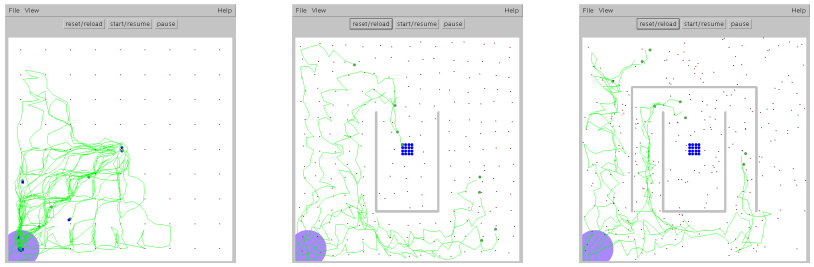
We combined the three navigation networks (attractor, homebase and coverage) to complete a multi-robot foraging task. We implemented this system in the TeamBots multi-robot simulation environment. The control systems were encoded using the Clay behavioral architecture [15]. In all the experiments we used 8 mobile robots with grippers, and 16 attractors in a group to represent the attractor cache to be exploited. All the robots had limited sensing and communication ranges of 4 meters that were occluded by obstacles. We tested the technique in three different $36x36m^2$ environments of increasing complexity. The three environments are shown in Figs. 3(a), 3(b), and 3(c).

Two key factors impacting how the system performs are the number of embedded nodes and how they are placed. As mentioned previously, when we have a large number of embedded nodes deployed uniformly we effectively have a real grid-world and the navigation networks are accomplishing distributed path-planning. Much work has dealt with trying to optimally deploy a sensor network for these tasks. In this research, however, we assume that the network is approximately uniformly distributed, but with random placement error. Placement error in a real system could be due to error in deployment or changes over time. Since our approach does not depend on the embedded nodes being localized, it is robust to changes in placement.

We ran experiments with 81, 121, 169, 225, 289, and 361 embedded nodes. Additionally, we varied the error in placement using the following technique. First, we placed the nodes uniformly across the space, then added error to each node's position by some random amount, the average distance from original position was varied: from 0, .5, and 2, to 10 meters. In the case of 10m average error, placement is essentially uniform random. Each experimental configuration was run 10 times. The graphs show mean performance, with errorbars denoting standard deviations.

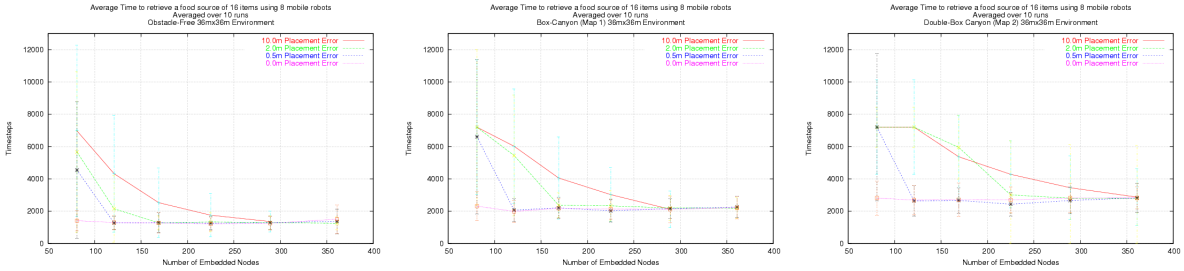
We present the results of the complete foraging task. Space limitations preclude presenting the individual analyses of the coverage and delivery subtasks. The results show the average time, in timesteps, to deliver each of the 16 attractors. A delivery time of 7200 timesteps (simulation timeout) were used for undelivered attractors. The results of the first obstacle free environment are shown in Fig. 4(a). The results of the second and third more complex environments are shown in Figs. 4(b) and 4(c). We see that as we increase the error in placement and the complexity of the environment, more nodes are needed to maintain the same level of performance. This is due to the fact that with a small number of nodes and large amount of error, the navigation network is disconnected and isn't able to guide the robots in the foraging task. Instead, they must rely on a random walk to cover the space and purely local reactive navigation.

We presented a technique for using a pervasive network of embedded sensor-less nodes to support multi-robot exploration and navigation. We showed that with enough embedded nodes, the distributed physical path plan-



(a) Map 0 (b) Map 1 (c) Map 2

Fig. 1.3. The simulation environments. (a) Map 0 is the first obstacle-free environment. A sample configuration is shown with 81 embedded nodes distributed uniformly throughout the environment without any error in placement. (b) Map 1 is a more complicated environment with a box canyon. A sample configuration is shown with 225 embedded nodes distributed uniformly with .5 m of error in placement. (c) Map 2 the most complicated environment with two box canyons. A sample configuration is shown with 289 embedded nodes distributed uniformly with 10m of error in placement.



(a) Map 0 (b) Map 1 (c) Map 2

Fig. 1.4. The average time to deliver each attractor as a function of the number of embedded nodes, the error in their placement, and the environment.

ning works even with very random, non-uniform, deployment of the embedded nodes in complex environments. In contrast, without enough nodes to form a connected network, the approach does not work since the network can not guide the robots. This approach also fails when the communication paths and navigation paths differ. One possible solution would be to use real navigation experiences to reinforce the paths in navigation networks. We are currently evaluating the technique on real robots.

References

1. H. V. D. Parunak, S. Brueckner, and J. Sauter, "Synthetic pheromone mechanisms for coordination of unmanned vehicles," in *Proceedings of First International Conference on Autonomous Agents and Multi-Agent Systems*, 2002, pp. 449–450.
2. H. V. D. Parunak, M. Purcell, and R. O'Connell, "Pheromones for autonomous coordination of swarming uavs," in *Proceedings of First AIAA Unmanned Aerospace Vehicles, Systems,Technologies, and Operations Conference*, 2002.
3. D. Payton, M. Daily, R. Estowski, M. Howard, and C. Lee, "Pheromone Robotics," *Autonomous Robots*, vol. 11, pp. 319–324, 2001.
4. M. Batalin and G. Sukhatme, "Sensor network-based multi-robot task allocation," *Proceedings of International Conference on Intelligent Robots and Systems (IROS 2003)*, October 2003.
5. ——, "Coverage, exploration and deployment by a mobile robot and communication network," *Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks*, 2003.
6. Q. Li, M. DeRosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," *The 2nd International Workshop on Information Processing in Sensor Networks*, 2003.
7. D. P. Bertsekas, "Distributed dynamic programming," *IEEE Transactions on Automatic Control*, vol. 27, no. 3, pp. 610–616, 1982.
8. S. Koenig, B. Szymanski, and Y. Liu, "Efficient and inefficient ant coverage methods," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 41–76, 2001.
9. I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein, "Smell as a computational resource - a lesson we can learn from the ant," *Proceedings of the ISTCS'96 - 4'th Israeli Symposium on Theory of Computing and Systems*, June 1996.
10. ——, "Distributed covering by ant-robots using evaporating traces," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 05, pp. 918–933, October 1999.
11. T. Balch and R. C. Arkin, "Communication in Reactive Multiagent Robotic Systems," *Autonomous Robots*, vol. 1, no. 1, pp. 27–52, 1994.
12. D. Goldberg and M. Mataric, *Robot Teams*. A K Peters Ltd., 2002, ch. Design and Evaluation of Robust Behavior-Based Controllers for Distributed Multi-Robot Collection Tasks.
13. R. E. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.
14. L. Ford and D. Fulkerson, *Flows in Networks*. Princeton, NJ: Princeton University Press, 1962.
15. T. Balch, "Clay: Integrating motor schemas and reinforcement learning," Georgia Institute of Technology, Tech. Rep. GIT-CC-97-11, 1997.