Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Bacterium-inspired Robots for Environmental Monitoring

Amit Dhariwal, Gaurav S. Sukhatme and Aristides A. G. Requicha
Department of Computer Science, Center for Robotics and Embedded Systems
University of Southern California, Los Angeles, California, USA
{dhariwal,gaurav,requicha}@usc.edu

Abstract— **Locating gradient sources and tracking them over time has important applications to environmental monitoring and studies of the ecosystem. We present an approach, inspired by bacterial chemotaxis, for robots to navigate to sources using gradient measurements and a simple actuation strategy (biasing a random walk). Extensive simulations show the efficacy of the approach in varied conditions including multiple sources, dissipative sources, and noisy sensors and actuators. We also show how such an approach could be used for boundary finding. We validate our approach by testing it on a small robot (the robomote) in a phototaxis experiment. A comparison of our approach with gradient descent shows that while gradient descent is faster, our approach is better suited for boundary coverage, and performs better in the presence of multiple and dissipative sources.**

I. Introduction

Several phenomena in nature induce gradients in their environment. For example, a fire induces a temperature gradient in its vicinity; an oil spill induces a concentration gradient of oil in the water etc. Detection, seeking and tracking such phenomena in-situ has received some attention recently [1], [2], [3]. The ability to autonomously detect, locate and track such phenomena (the source of the induced gradient) would give scientists a tool to monitor and study ecosystems at an unprecedented level of detail. Typical gradients of interest to scientists studying the ecosystem include temperature, light, salinity, mineral concentration, pH, etc. These problems are difficult because of the time-varying nature of the source, the dynamics of the environment, a multiplicity of (possibly interacting) sources, and finally, a paucity of sensing.

Motivated by these applications and challenges, we are interested in the development of simple, robust, energy efficient and cost-effective techniques which could be used in-situ to locate source phenomena of interest to scientists (for ex., the detection of a mineral pollutant in seawater). In this paper we focus on a 2D version of this problem, and propose a simple strategy for a mobile robot (or multiple robots) to navigate to such a source using gradient information and extremely rudimentary actuation. Our strategy is inspired by the studies of taxis in bacteria [4].

Previous approaches to location and tracking of gradient sources include spiral surge [1], gradient seek [5], sensor arrays [6], swarm intelligence [7]. These approaches are good for application in nearly static environments. Most of these approaches have difficulties tracking sources that are small, weak, mobile or sources whose intensity varies with time. The gradient seeking strategies are susceptible to local minima and plateaus. In the presence of multiple sources they are able to locate and proceed to the closest source enacting a greedy solution.

Another related area is the detection of dynamically changing gradient source boundaries in-situ in a distributed manner [5], [6], [8]. This process involves locating the periphery of the source, a small region which has the sharpest gradient in a region close to the source. A brute force technique is to compute the gradient at every point in the search space, but such a technique is not efficient in terms of energy, time requirements and the amount of processing involved.

In this paper, we present a novel technique based on biased random walk [9] for the detection, seeking and tracking of gradient inducing source phenomena. Our approach is inspired by the way bacteria detect, locate and track nutrient sources in nature [10]. We begin by a discussion of the characteristics of bacterial motion and how it can be adapted and applied to the problem at hand. In section III we describe the simulation platform we created to evaluate our strategy along with a discussion of the results obtained from simulations. Section IV presents an implementation of our approach on the Robomote robot platform and a discussion of the results obtained and their applicability. In section V, we present a comparison between the behavior of a gradient descent strategy and the behavior of a biased random walk model. This is followed by our conclusions and a brief comment on future work we plan to pursue.

II. Bacterial Motion

A. Biased Random Walk as a result of Chemotaxis

Nature presents us with a wide variety of simple biological models which have evolved and refined over time. These models help in sustaining the flora and fauna on our planet and maintaining the ecological balance in diverse habitats. Bacterial motion [10] and its response to the presence of chemical concentration gradients called chemotaxis (or chemokinesis) have been well studied [4], [11], [12]. The response to a chemical stimulus in their vicinity helps bacteria find sources of nutrients which are essential for their survival. Chemotaxis is also observed in various other species of animals for varied purposes including colony formation, predator avoidance and breeding ground location [10].

0-7803-8232-3/04/$17.00 ©2004 IEEE

Bacteria sense chemical concentration using receptors. They are able to detect temporal and spatial changes in chemical concentration based on the fraction of receptors occupied at successive time intervals. An increase in the fraction of occupied receptors is called a positive gradient while a decrease is called a negative gradient. A chemical whose concentration gradient attracts the bacterial cells is called a chemo-attractant.

Bacteria produce motion by the movement of their flagellum [13]. A counter clockwise flagellar rotation results in a smooth swim motion in a straight line in a particular direction (we call this a run) while a clockwise rotation of the flagellum causes the bacterium to randomly reorient itself in a new direction (we call this a tumble), which is the direction for the next run. Motion alternates between these two stages (run and tumble).

The duration of the run (which is related to the mean free path) is dependent on the concentration gradient that is sensed in the vicinity of the bacterial cell. In the absence of a gradient, the run length is independent of the direction of motion and the bacterium executes a random walk. In the presence of a positive gradient, the frequency of tumbling is reduced resulting in a longer run length [11]. The presence of a negative gradient does not have any effect on the tumbling frequency. This change of tumbling frequency in response to concentration gradient results in chemotaxis, allowing bacteria to move towards sources of nutrients. Informally, chemotaxis is a biased random walk.

B. A Robotic Implementation

Based on the description of bacterial motion presented in section II-A, it is clear that a strategy based on biased random walk can be used to locate and track gradient sources. This biologically-inspired algorithm can be implemented on a group of robots with simple sensing and actuation. The strategy of such a bacteria-like robot can be summarized as “sense and move”. A robotic node executing a biased random walk has very small requirements in terms of memory since only the last sensor reading needs to be stored. The processing requirements are minimal since the only processing required is comparison between successive sensor readings (gradient computation). Only a minimal amount of motion control is required to hold the heading of the robot in a particular direction for a particular duration of time (depending on bias levels).

III. Simulation Experiments

A. Methodology

To validate our ideas and explore the possible implications of a bacterial motion-based approach for locating and tracking gradient sources and their boundaries, we designed a simulation platform.

We created a model of the world as a uniform two dimensional grid of dimension 2000 * 2000 Units (Fig. 1). We chose this size for our model because this would help us generate areas of very low/negligible gradient concentrations within the grid. We initialize one or more sources of gradient (S_i) at randomly chosen positions in the grid. The intensity (or gradient concentration) at any point in the grid due to these sources was modeled to follow an inverse square law distribution given by

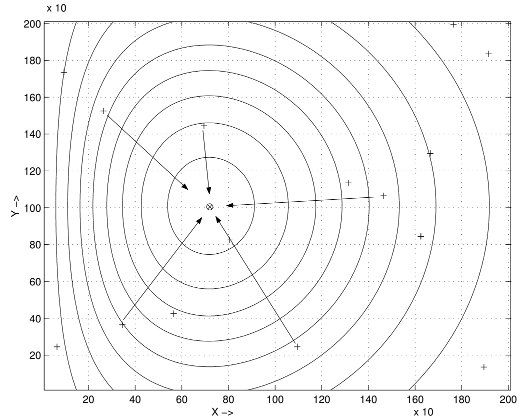

Fig. 1

Simulation Grid Setup

$$Intensity(x, y) = \frac{1}{K} \sum_{i=0}^{m} \frac{q_i}{r_i^2} \tag{1}$$

Equation (1) gives the intensity that can be sensed at a point (x, y) on the grid in the presence of m gradient sources, q_i is the intensity of the source S_i, K is a constant of proportionality and r_i is the distance between the grid point (x, y) and the center of source S_i.

All our simulations considered a set of 100 robots deployed in the grid. The deployment was either random (using the values derived from a uniform distribution over the grid region) or based on the requirements of the simulation. At each time step each of these robots could either move to one of its 8 neighboring grid points or change its direction of motion or stay at its position if it had reached the gradient source.

We simulated a biased random walk with a Mean Free Path (MFP) of 10 units, i.e., under the absence of a concentration gradient, each robot would move 10 units of distance along the grid in a particular direction before tumbling and changing its direction of motion randomly. However, if the robot senses a positive change in gradient, it decreases its tumbling frequency thus increasing the run-length resulting in a biased random walk (Fig. 2(b)). The typical bias value we used in our simulations is 10% of the MFP which is similar to the bias values observed in nature for bacterial motion [10].

We performed another set of simulations where the intensity of the source varied over time. The change in intensity could be attributed to a variety of reasons such as inherent nature of the source to dissipate its energy over time, occlusion of the source, consumption of the source (for ex., a nutrient source being consumed by bacteria over time). The intensity of the source (S_i) at any time t is given by

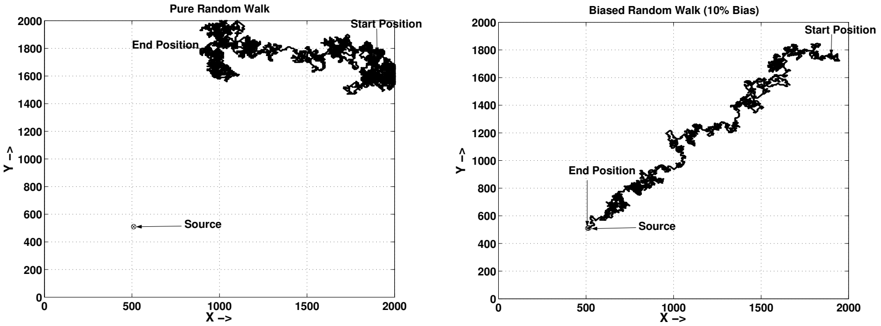

Fig. 2
PURE RANDOM WALK VS BIASED RANDOM WALK

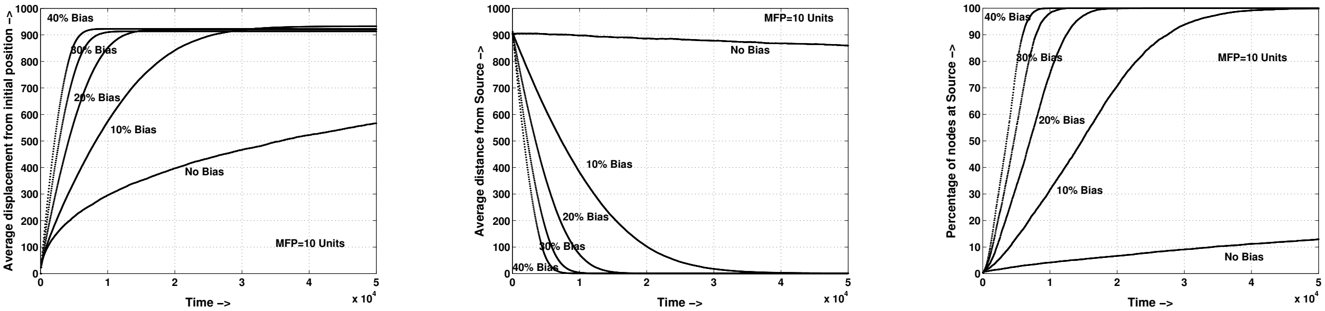

Fig. 3
THE EFFECTS OF VARYING BIAS. THE LARGER THE BIAS, THE QUICKER THE CONVERGENCE. THE DATA ARE VISUALIZED USING THREE METRICS: A. ROBOT DISPLACEMENT VS. TIME (LEFT), B. DISTANCE BETWEEN ROBOT AND SOURCE VS. TIME (MIDDLE), AND C. PERCENTAGE OF ROBOTS AT SOURCE VS. TIME (RIGHT).

$$q_i(t) = (q_i(0)e^{-k_1 t}) - k_2 \sum_{j=0}^{t} N_{ij} \qquad (2)$$

where $q_i(0)$ is the initial intensity of source S_i, k_1 and k_2 are constants whose values are set based on the type of source we are trying to model, N_{ij} is the number of robots at source S_i depleting its energy at time instant j. The first term models the decrease in intensity due to the energy dissipated by the source over time while the second term models the decrease in source intensity due to consumption by the robots near it.

The results we present in sections III-B and V represent average values obtained over 10^4 trials to filter out the effects of noise and outliers.

B. *Simulation Results*

Our simulations modeled the biased random walk strategy under a wide variety of simulation models including various gradient setup schemes (decaying vs. constant source, linear vs. exponentially decaying gradients with distance) and robot deployment strategies. In our trials, once a robot reaches a particular gradient source, it stays there as long as the source remains sufficiently active (i.e., intensity remains above some threshold value). The gradient source is modeled as a circular disc of radius 5 units. We designed a set of metrics which would help understand and evaluate our approach.

Our first metric is the average displacement of the robot from its initial position. We monitor this over time as the simulation proceeds. The second metric we use is the average distance of the robot from the source over time.

Fig. 3 presents results from simulations performed to study the effect of varying bias levels on the speed of convergence of the robots towards a single source. As can be seen in Fig. 3, even when no bias is present, the robots do move and explore a lot of space. As expected, in the absence of a bias, the robots wander around without showing any particular progress towards the source. An introduction of a very small amount of bias of the order of 10% results in a rapid movement of robots towards the source of the gradient. The exploration phenomena was observed to be independent of initialization conditions as long as the other simulation parameters (bias etc.) remained

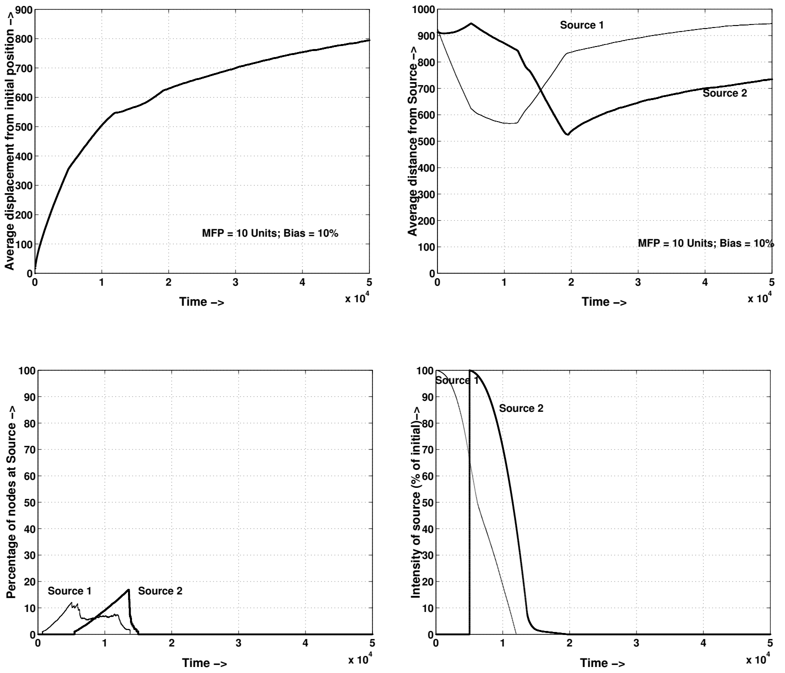

Fig. 4

Multiple Sources. a.Robot displacement vs. Time, b.Distance from source vs. Time, c.Percentage of robots at source 1 & 2, and d.Intensity of source 1 & 2

the same.

Fig. 4 presents the results from another set of simulations where we initialize a single gradient source at the start of simulation (t = 0s). A second source is introduced at t = 5000s. Both sources are modeled as dissipative energy sources, with dissipation rates proportional to the number of robots that have reached them respectively.

Another feature to observe is how well the robots did in terms of reaching an individual gradient source, how fast and how many of them reached which source? This gives us a third metric to evaluate our model wherein we monitor the average number of robots reaching a particular source over time. This metric is more meaningful when plotted alongside the source intensity variations. Figs. 3(c) and, 4(c) show how the robots reach the source as the simulation proceeds in time. They also demonstrate how the appearance of a second source attracts some of the robots and pulls them away from the first source. As the intensity of the source starts decreasing, some of the robots start moving away from it and once it disappears, all the robots which were near it start executing a random walk in search of other sources and move away from it. The results from the multiple source experiment (Fig. 4) also demonstrate that this technique does not show any preferential movement towards a particular source based on its time of appearance.

We performed another set of simulations with multiple sources having different intensities and the results were as we expected. All the sources were tracked. The weaker sources got a comparatively smaller fraction of the robots because the gradients they set up were very small, but on an average all sources were well covered. Another set of simulations were performed to study how the performance of the algorithm varied in presence of linear, cubic and exponential gradients. The results were similar since the decision to run or tumble is governed by the sign of the change in gradient and not by the magnitude of the change. After executing a *run*, the robot essentially takes a new sensor reading and compares it with the previous sensor reading taken before it started the current run. The decision to tumble or continue in the same direction is then based on the sign of the difference in the sensed gradient values and does not depend on the absolute sensor readings. This also makes the system tolerant to static sensor errors.

We performed another set of simulations introducing a Gaussian error in the decision function itself. These errors model the non-static sensor errors and actuation errors (i.e., motion of the robot might not be the same as the command signal applied). Fig. 5 presents the results in the presence of 0%, 6%, 20% and 40% error. Even in the presence of 40% error, the robots still converge to the gradient source.

Another set of simulations were performed to understand how well the robots spread around a gradient source i.e., do the robots reach a gradient source all at the same place? We modeled a circular source with a diameter of 45 units and studied the effect of deploying the robots randomly, uniformly and at a single location in the grid at the start of the simulation.

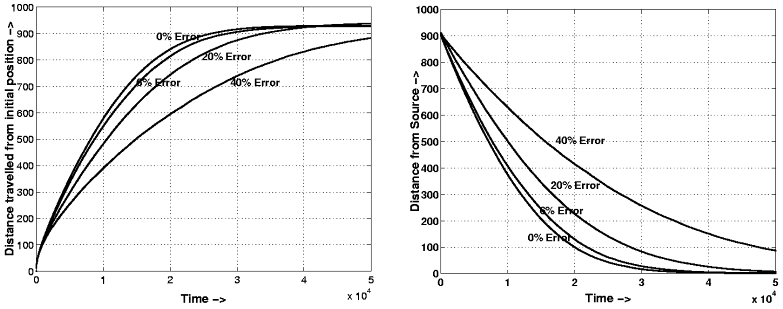

Fig. 5
EFFECT OF INTRODUCTION OF ERRORS IN THE DECISION FUNCTION

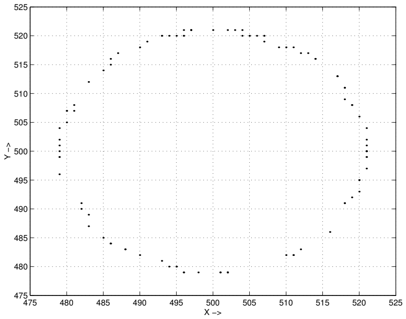

Fig. 6
EFFECTIVENESS AT BOUNDARY DETECTION

Fig. 6 presents the result of initializing all the robots at a single location at one corner of the grid. As can be seen, the robots approach the source from all directions. Similar results were obtained from random and uniform initial robot deployments. A point to note here is that even if all the robots were deployed in very close vicinity of the source, some spread along the periphery of the source was achieved. The results from irregularly shaped sources were also in agreement with the above observations.

C. Discussion

From the above results we conclude that a strategy based on biased random-walk implemented on a group of mobile robots can effectively track multiple time-varying gradient sources at the same time. Moreover, such a set of robots require very minimal amount of controls to be developed. The only control element is the change in the length of the run in response to the sensed gradient change.

An analysis of our simulation results indicates that we can trade bias for speed vs. efficiency. Higher bias values accelerate the movement of the robots towards the source but might not be good if the source is mobile or has a variable intensity. Lower bias values result in longer response times for the robots to reach the source but are more effective in tracking weak, mobile and intensity-varying sources. Our results also indicate that it is possible to determine a range of bias values to track a gradient source with any desired source tracking characteristics.

The source boundary detection results also highlight the suitability of our approach to a set of applications where the source dimensions are comparatively large and we need to track its complete boundary (e.g. the boundary of an underwater plume or the boundary of an oil spill). For such applications the algorithm should be able to achieve a sufficient spread of robots along the entire boundary.

IV. EXPERIMENTS WITH THE ROBOMOTE ROBOT

Validation experiments were carried out on the Robomote (Fig. 7(a)) test bed developed at the Robotic Embedded Systems Lab at the University of Southern California. The capabilities of this small, matchbox-sized, two-wheeled robomote include moving along a straight line for a specified duration and/or distance and the ability to turn in place by a specified angle. For a more detailed description of the platform the reader is referred to [14].

Mica-I motes developed by Crossbow were used to give control commands to the robomote using TinyOS [15]. We used two basic components *move* and *rotate* to provide control commands to the robomote for performing a biased random walk. A light source was placed at one end of the test bed to generate a photo (light) gradient. A basic sensor board with a photo sensor was mounted on the robomote (Fig. 7(b)) to sense the photo gradient. The experimental setup can be seen in Fig. 7(c). Color blobs were mounted on top of the robomote to help detect its position on the test bed. The position of the robomote was tracked using an overhead camera and a tracker [16].

We positioned the robomote at a distance d (d = 40 cm, 80 cm, 120 cm) from the source at the start of the experiment and tracked its position as it executed a biased random walk on the test bed following the photo gradient. Each of the d values constituted a circular arc on the table of radius d units from the center of the light source.

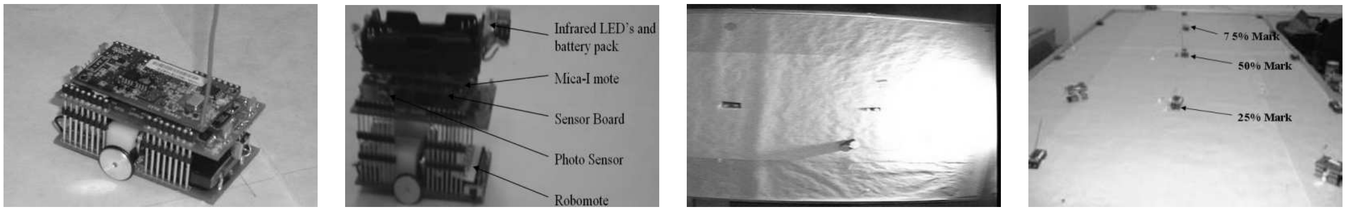

Fig. 7
ROBOMOTE AND THE TESTBED

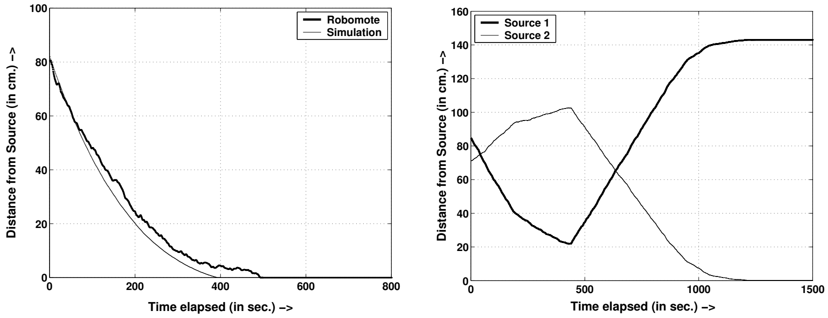

Fig. 8
ROBOT BEHAVIOR IN RESPONSE TO (A) SINGLE AND (B) MULTIPLE GRADIENT SOURCES

A fixed value for *d* still leaves the position of the robomote on the arc and its initial heading as random variables which can be either towards or away from the source. A small circular arc of radius 5 cm around the source was considered as the source radius. We were interested in measuring if the robomote reaches the source and if so in how much time? The speed of the robomote was set at 2 cm/s for the experiments. We chose a MFP of 5 cm and a bias factor of 40%.

Once switched on at the distance *d*, the robomote starts off by taking a sample using the photo sensor (S_i). It starts moving along a straight line in the direction of its current heading for a distance and/or duration as specified in the random walk parameter MFP (run). At this point it takes another photo sensor reading and compares it with the previous reading from the photo sensor. If it senses no change or a negative change in gradient, it randomly chooses a new heading direction (θ_i) and rotates in place to orient to that heading. If a positive change in gradient was sensed, it continues its motion for an additional distance specified by its bias value before randomly computing the new heading and making a turn (tumble). In either case the run and tumble cycle is repeated. The experiment terminates when the robomote reaches the source.

We repeated the experiment 75 times for each of the d values with random starting location and orientation of the robomote along the arc and averaged our position readings over the collected data sets. We believe this gave us a good enough set to evaluate the effectiveness of our approach. The graph for the metrics we proposed in the previous section can be seen in Fig. 8(a). The results from the robomote platform agree with our simulation results.

We repeated the experiment with two equal intensity sources present at the same time at opposite corners of the test bed and started the robomote at distances d (d = 25%, 50% (center) and 75% position on the test bed) (Fig. 7(d)). Contrary to the expectation that the robomote would have moved to the closest source, we observed what we had expected, i.e., the source where the robomote finally ended was independent of the starting orientation and location of the robomote at the start of the experiment.

With the same setup as the previous experiment (Fig. 7(d)), we carried out another set of experiments. We started with only one source switched on initially at t = 0s. At t = 180s we switched on the second source which was located at the other end of the test bed. This was followed by switching off the first source completely at t = 435s. We repeated the experiment for the different values of *d* (d = 25%, 50% (center) and 75% position on the test bed). The results we obtained on hardware platform (Fig. 8(b)) were in agreement with our simulation results.

V. COMPARISON WITH GRADIENT DESCENT STRATEGY

We are trying to locate a gradient source by essentially following gradients. An obvious question arises: why not just use a simple gradient descent algorithm? In this section we

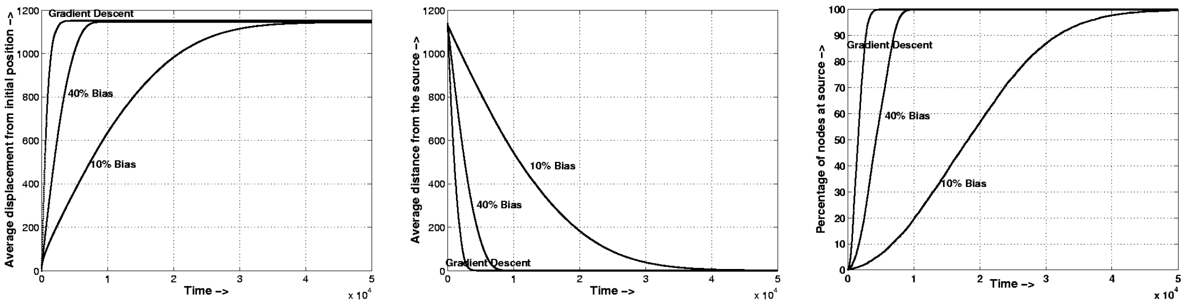

Fig. 9

COMPARISON OF GRADIENT DESCENT STRATEGY WITH BIASED RANDOM WALK IN PRESENCE OF A SINGLE SOURCE

compare the performance of our biased random walk approach with a simple gradient descent algorithm.

We begin by presenting an evaluation of the performance of the simple gradient descent strategy in simulation. We use the same simulation framework that we developed in section III-A. We simulated a gradient descent strategy with a Mean Free Path (MFP) of 10 units, i.e., under the absence of a positive concentration gradient in the current direction of motion, the robot would move 10 units of distance along the grid in that direction before tumbling and changing its direction of motion randomly and then repeat the run and tumble stages. However, if the robot sensed a positive change in gradient, it continues to move in the direction it was moving, for another MFP units, before checking for the gradient change again.

The results from the simulation are presented in Fig. 9 along with results for the biased random walk simulations (repeated here for easier comparison). The results clearly indicate that a simple gradient descent scheme performs better than our biased random walk for small bias values. However, as the bias levels are increased, the two become comparable.

Sources in nature seldom occur by themselves or without interference from each other. In order to compare the two techniques under more realistic conditions, we carried out another set of simulations with multiple sources. We started the simulation with a single source at one corner of the grid at time t = 0s. At time t = 1500s. we initialized a second source at the opposite corner of the grid and tracked the performance of the gradient descent algorithm over time. At t = 10000s we turned off the first source and continued tracking the motion of the robots.

The results from the simulation are presented alongside the results for biased random walk in Fig. 10. As can be seen from the results, the gradient descent strategy performs better than the biased random walk approach in the presence of a single gradient source both in terms of the number of robots which reach the source (Fig. 10(c)) as well as the time taken by them to reach the source. With the introduction of a second source, the gradient descent strategy keeps following the first source and the number of robots at the first source keeps increasing and very few robots reach the second source (Fig. 10(d)). On the other hand, the results from the biased random walk are quite impressive. Both gradient sources are simultaneously tracked and receive a good share of the number of robots reaching them. The introduction of more gradient sources results in some robots tracking each one of them as long as there are some available robots. Thus for the purpose of tracking multiple gradient sources, our algorithm clearly outperforms the gradient descent approach.

Our last set of simulations verifies the source coverage obtained by the gradient descent algorithm. We initialized all the robots at the same location on the grid. The gradient descent algorithm terminated with all the robots on one quarter of the source boundary (nearest their initial position), whereas (as presented in the previous section), the biased random walk approach resulted in a spread all around the source. From the above set of results, clearly the biased random walk approach we propose outperforms the simple gradient descent strategy for the applications we consider.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented an algorithm based on biased random walk [9] for the detection, seeking and tracking of gradient inducing source phenomena. Our approach is inspired by the way bacteria detect, locate and track nutrient sources in nature [10]. Through a wide set of simulation and experimental work on robots, we demonstrated how our strategy is well suited to varied conditions including multiple sources, dissipative sources, and noisy sensors and actuators. We also show how our approach could be used for boundary finding. We validated our approach by testing it on a small robot (the robomote) in a phototaxis experiment. A comparison of our approach with gradient descent shows that while gradient descent is faster for tracking a single source, our approach performs better in the presence of multiple and dissipative sources and is also better suited for boundary coverage.

Our algorithm is robust in the sense that it is guaranteed to work as long as there is a gradient. The inherent randomness in the algorithm prevents the robots from landing in a local

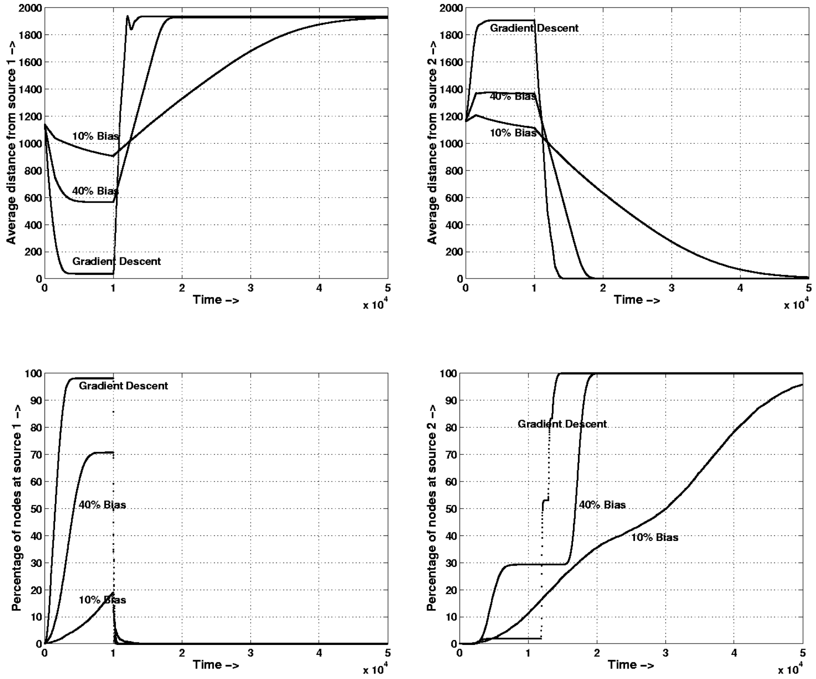

Fig. 10

COMPARISON OF GRADIENT DESCENT STRATEGY WITH BIASED RANDOM WALK IN PRESENCE OF MULTIPLE SOURCES

minimum. The algorithms insensitivity to inherent sensor errors and limited tolerance to actuation errors makes it a good choice for low-cost sensors. The algorithm is scalable and ideal for implementation on very simple, low-cost robots. Cost and scalability are the major issues for the applications we outlined in section I.

In the future, we will carry out surface-water experiments for detection and tracking of one or more dissipating color dye sources. We will also extend our algorithm to perform distributed data fusion of the data from multiple sensors on the robot.

ACKNOWLEDGMENT

The authors thank Prof. David A. Caron and Beth Stauffer, Department of Biological Science, University of Southern California for their comments. This work is funded in part by grants CCR-0120778, EIA-0121141, IIS-0133947 from the National Science Foundation.

REFERENCES

[1] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Distributed odor source localization," in *Special Issue on Artificial Olfaction*, H. T. Nagle, J. W. Gardner, , and K. Persaud, Eds., vol. 2, no. 3, IEEE Sensors. IEEE Press, 2002, pp. 260–271.

[2] F. Grasso, T. Consi, and J. Mountain, D. Atema, "Locating odor sources in turbulence with a lobster inspired robot," in *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, From Animals to Animats 4*. Cape Cod, MA: MIT Press, 1996.

[3] R. Russell, T. D., R. Deveza, and A. Mackay-Sim, "A robotic system to locate hazardous chemical leaks," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1995, pp. 556–561.

[4] J. Adler, "Chemotaxis in bacteria," *Science*, vol. 153, pp. 708–716, 1966.

[5] H. Ishida, T. Nakamoto, T. Moriizumi, T. Kikas, and J. Janata, "Plume-tracking robots: A new application of chemical sensors," *Biological Bulletin*, no. 200, pp. 222–226, Apr 2001.

[6] H. Ishida, M. Hirayama, J. Janata, D. Webster, and T. Moriizumi, "Sensor arrays for tracking of underwater chemical plumes," IEICE, Tech. Rep. 253, 2001.

[7] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Swarm robotic odor localization," in *Proceedings of the IEEE Conf. on Intelligent Robots and Systems IROS-01*, Oct 2001.

[8] B. Zhang, G. S. Sukhatme, and A. A. G. Requicha, "Adaptive sampling for marine microorganism monitoring," Center for Robotics and Embedded Systems, University of Southern California, Tech. Rep. CRES-04-005, Feb 2004.

[9] W. Alt, "Biased random walk models for chemotaxis and related diffusion approximations," *J. Math. Biol.*, vol. 9, pp. 147–177, 1980.

[10] H. C. Berg, *Random Walks in Biology*. Princeton University Press, 1983.

[11] H. C. Berg and D. A. Brown, "Chemotaxis in Escherichia coli analysed by three-dimensional tracking," *Nature*, vol. 239, pp. 500–504, Oct 1972.

[12] S. D. Muller, J. Marchetto, S. Airaghi, and P. Koumoutsakos, "Optimization based on bacterial chemotaxis," in *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, Feb 2002.

[13] H. C. Berg and R. A. Anderson, "Bacteria swim by rotating their flagellar filaments," *Nature*, vol. 245, pp. 380–382, 1973.

[14] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "A tiny mobile robot platform for large-scale sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2002)*, Washington DC, May 2002.

[15] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for network sensors," in *ASPLOS 2000*, Cambridge, Nov 2000.

[16] A. Howard, "Mezzanine user manual; version 0.00," Institute for Robotics and Intelligent Systems, University of Sourthern California, Tech. Rep. IRIS-01-416, 2001.