Nonsmooth analysis and sonar-based implementation of distributed coordination algorithms

Craig L. Robinson[1], Daniel Block[1], Sean Brennan[2], Francesco Bullo[1], and Jorge Cortés[1]

[1]General Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign

{clrobnsn,d-block,bullo,jcortes}@uiuc.edu

[2]Department of Mechanical Engineering, The Pennsylvania State University

sbrennan@mne.psu.edu

Abstract— This paper investigates the behavior of a group of autonomous robots evolving in a polygonal environment according to a "move away from the closest neighbor" heuristic. We demonstrate that this distributed coordination algorithm optimizes an aggregate cost function that measures how uniformly distributed are the robots in their environment. Our technical approach based on non-smooth analysis and computational geometry unveils a sphere-packing problem. The algorithm is implemented in a testbed of indoor mobile robots equipped with sonar. We develop novel approaches for improving single point sonar scan performance. These algorithms are then shown to have improved reliability, resolution and speed in distributed environments as compared to other scanning methods.

I. INTRODUCTION

One fundamental capability of future mobile and tunable networks of robots will be the ability to perform spatially-distributed sensing tasks including coverage, surveillance, exploration, target detection, and search. These future networks of autonomous vehicles will be able to adapt to changing environments and dynamic situations. They will provide guaranteed quality of service in the presence of failures, and will operate via limited-bandwidth ad-hoc communication links. The algorithms required to achieve these desirable capabilities must be amenable to implementation in a cooperative setting, i.e., they are required to be *distributed, asynchronous, adaptive, and verifiably correct.* Sample references on coordination problems for multi-vehicle networks include [1], [2], [3].

The premise of this paper is based on a class of coverage and deployment problems for networks of mobile robots. Once the optimal sensor coverage is formalized as an aggregate performance metric via methods from geometric optimization, a class of cooperative control algorithms can be designed by generalizing the classic Lloyd algorithm from quantization theory. The resulting control laws are interaction protocols between the mobile sensors and include behaviors such as, "move away from your closest neighbor," and, "move toward the geometric center of your sensing region."

The first objective of this paper is the study of the "move away from your closest neighbor" algorithm in group of robots and is developed in Section II. Tools from nonsmooth analysis, stability and convergence analysis via a LaSalle Invariance Principle and geometric optimization concepts such as sphere-packing and disk-covering functions are used. It turns out that this collection of techniques is well-suited to study a number of coordination problems. The reader is referred to [4] for the comprehensive mathematical treatment.

The second objective of this paper is to investigate the practical issues that arise in the implementation of these algorithms. A decentralized sensor based testbed consisting of several mobile DSP platforms (discussed in Section III) equipped with rotating sonar sensors is used. Successful implementation of the algorithms is largely dependent on the properties and limitations of the inexpensive wide-angle sonar sensors used. These problems are reviewed in Section IV which combined with the implementation limitations in Section IV-A provided the impetus to develop improved methods for single point sonar scan resolution and reliability. Hence, the final objective of this paper is contained in Sections IV-C and IV-E where two new algorithms called the `Max Filter` and `Max Resolver` as well as an extension of the `EERUF` method by Borenstein and Korne [5] are described. These methods prove to be very effective at improving sonar reading reliability and resolution within the context of this investigation in an environment with large amounts of interference. Finally, Section V discusses the combined implementation of the sensing and coordination algorithms, as well as the modifications, adaptations and tradeoffs of the final implementation.

II. INTERACTION ALGORITHMS

A. Preliminaries

Let $\| \cdot \|$ denote the Euclidean distance function and let $v \cdot w$ denote the scalar product of the vectors $v, w \in \mathbb{R}^N$. Throughout the paper, $\mathrm{versus}(v)$ will denote the unit vector in the direction of $v \neq 0$, i.e., $\mathrm{versus}(v) = v/\|v\|$. Recall that a set $S \subset \mathbb{R}^N$ is said to be *convex* if $\lambda x + (1 - \lambda)y \in S$, for all $\lambda \in [0, 1]$ and all $x, y \in S$. Given $S \subset \mathbb{R}^N$, the *convex hull of S*, $\mathrm{co}(S)$, is defined as

$$\mathrm{co}(S) = \big\{ z \in \mathbb{R}^N \mid z = \lambda x + (1 - \lambda)y$$
$$\text{for some } \lambda \in [0, 1] \text{ and } x, y \in S \big\} .$$

Obviously, if S is convex, then $S = \mathrm{co}(S)$. In general, $S \subset \mathrm{co}(S)$. If S is a convex set in $\mathbb{R}^N$, let $\mathrm{proj}_S \colon \mathbb{R}^N \to S$ denote the orthogonal projection onto S and let $\mathrm{D}_S \colon \mathbb{R}^N \to \mathbb{R}$ denote the distance function to S. Also, let $\mathrm{Ln}(S)$ denote the least-norm vector in S (see Figure 1).
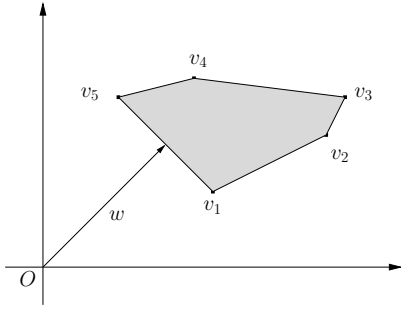
Fig. 1. The convex hull of the set $S = \{v_1, v_2, v_3, v_4, v_5\}$ is shown in grey. The vector w is the least-norm element in $\mathrm{co}(S)$.

Let Q be a convex polygon in $\mathbb{R}^2$. Denote by $\mathrm{int}\,Q$ its interior set and by $\mathrm{Ed}(Q) = \{e_1, \ldots, e_M\}$ its set of edges. Given an edge $e \in \mathrm{Ed}(Q)$, we let n_e denote the unit normal to e pointing toward the $\mathrm{int}(Q)$. Let $P = (p_1, \ldots, p_n) \in Q^n$ denote the location of n generators (or robots) in the space Q. The *Voronoi partition* $\mathcal{V}(P) = (V_1(P), \ldots, V_n(P))$ of Q generated by the points $(p_1, \ldots, p_n) \in Q^n$ is defined by

$$V_i(P) = \{q \in Q \mid \|q - p_i\| \le \|q - p_j\|, \ \forall j \ne i\}.$$

For simplicity, we refer to $V_i(P)$ as V_i. If V_i and V_j share an edge, i.e., $V_i \cap V_j$ is neither empty nor a singleton, then p_i is called a *neighbor* of p_j (and vice-versa). Given a polytope W in $\mathbb{R}^N$, its incenter set, denoted by $\mathrm{IC}(W)$, is the set of the centers of maximum-radius spheres contained in W.

B. The geometric optimization problem

Consider the optimization problem consisting of maximizing the following aggregate cost function

$$\mathcal{H}(P) = \min_{\substack{i,j \in \{1,\ldots,n\} \\ i \ne j, \ e \in \mathrm{Ed}(Q)}} \left\{ \tfrac{1}{2}\|p_i - p_j\|, \mathrm{D}_e(p_i) \right\}$$
$$= \min_{i \in \{1,\ldots,n\}} \left\{ \min_{q \notin \mathrm{int}\,V_i} \|q - p_i\| \right\}.$$

This geometric optimization problem corresponds to the situation where we are interested in maximizing the coverage of the polygon Q in such a way that the radius of the robots do not overlap (in order not to interfere with each other) or leave the environment. Following [4], this problem can be restated as a *sphere-packing problem*: how to maximize the coverage of a region with non-overlapping disks (contained in the region) of minimum radius. The problem reads:

$$\max\{R \mid \cup_{i \in \{1,\ldots,n\}} \overline{B}(p_i, R) \subseteq Q,$$
$$B(p_i, R) \cap B(p_j, R) = \emptyset\}, \quad (1)$$

where $B(p, R) = \{q \in \mathbb{R}^2 \mid \|q - p\| < R\}$ and where $\overline{B}(p, R)$ is its closure.

Remark 2.1: In the definition of $\mathcal{H}$ we employ a $1/2$ correction factor in comparing the pairwise robot distances, $\|p_i - p_j\|$, with the distances to the walls, $\mathrm{D}_e(p_i)$. This factor is necessary to establish the equivalence between minimizing $\mathcal{H}$ and solving the sphere-packing problem (1).

C. The distributed coordination algorithms

Here we design two coordination algorithms with the property that the corresponding closed-loop systems are guaranteed to monotonically increase the value of the performance measure $\mathcal{H}$. Let $i \in \{1, \ldots, n\}$. At a configuration $P \in Q^n$, the following set describes the distances of the generator p_i to the rest of generators and to the boundary of Q,

$$M_i(P) = \left\{ \frac{1}{2}\|p_j - p_i\| \mid j \in \{1, \ldots, n\} \setminus \{i\} \right\}$$
$$\cup \{\mathrm{D}_e(p_i) \mid e \in \mathrm{Ed}(Q)\}.$$

Let $R_i(P)$ denote the minimum of $M_i(P)$. Note that $\mathcal{H}(P) = \min_{i \in \{1,\ldots,n\}} R_i(P)$. Consider the set $S_i(P)$ defined by

$$\mathrm{versus}(p_i - p_j) \in S_i(P) \Leftrightarrow \frac{1}{2}\|p_j - p_i\| = R_i(P),$$
$$n_e \in S_i(P) \Leftrightarrow \mathrm{D}_e(p_i) = R_i(P).$$

Note that if there is a single element (generator or edge of Q) which is nearest to p_i, the set $S_i(P)$ is formed by a single vector. If several elements are equidistant to p_i, then $S_i(P)$ is composed of several vectors (see Figure 2). For implementation purposes, it is also worth noticing that in order to compute $S_i(P)$, we need to compare $\frac{1}{2}\|p_i - p_j\|$ with $\mathrm{D}_e(p_i)$ (see Remark 2.1).
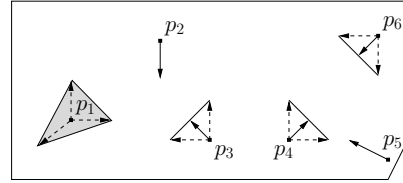

Fig. 2. Illustration of the vector field (2a). At this configuration, the first generator is the only one such that $\mathrm{Ln}(\mathrm{co}(S_1(P))) = 0$.

Now, consider the following dynamical systems describing the evolution of the generators $i \in \{1, \ldots, n\}$,

$$\dot{p}_i = \mathrm{Ln}(\mathrm{co}(S_i(P))). \qquad (2a)$$
$$\dot{p}_i \in \mathrm{IC}(V_i(P)). \qquad (2b)$$

The first system is *nearest-neighbor-distributed* in the sense that $\mathrm{Ln}(\mathrm{co}(S_i(P)))$ depends only by the position of p_i and its nearest neighbors. The second system is *Voronoi-neighbor-distributed* in the sense that $\mathrm{IC}(V_i(P))$ depends only by the position of p_i and its Voronoi neighbors. Note also that the solution to both systems must be understood in the Filippov sense [6]. Classical notions in differential equations such as existence and uniqueness of solutions, invariant sets and stability analysis can also be extended to dynamical systems of this type. In particular, we need the notion of weakly invariant set to state the next result: a set M is said *weakly invariant* if for each $x_0 \in M$, M contains a maximal solution of the corresponding dynamical system in (2).

Theorem 2.2: For the dynamical systems (2a) and (2b), the generators' location $P = (p_1, \ldots, p_n)$ converges asymptotically to the largest weakly invariant set contained in the closure of $A(Q) = \{P \in Q^n \mid R_i(P) = \mathcal{H}_i(P) \Rightarrow p_i \in \mathrm{IC}(V_i)\}$.

The configurations $P \in A(Q) \subset Q^N$ have the property that all the robots with the smallest value of $R_i(P)$ are located at the incenter of their Voronoi regions.

III. INDOOR MULTI-ROBOT TESTBED

A group of 5 mobile robots was developed in the UIUC Control Systems Laboratory. The robots are based on a 150MHz 32bit floating point Texas Instruments DSP 6711 mounted on a TI DSP development board. A daughter card was built in house to provide an interface point for a BX24 Micro controller, PWM output, encoder input, A/D and D/A converters and serial communication. The BX24 Micro controller co-ordinates a set of three Daventech SRF04 ultrasonic sensors as well as three infrared distance sensors. The sensors are mounted on a RC servomotor that rotates to incrementally scan the area around the robot. Pittman motors drive the robot wheels and HP optical encoders provide odometry information. A Matrix Orbital LCD screen displays onboard debugging information and data is sent over a RS232 wireless serial connection to a base station PC and to the other robots. Texas instruments Code composer and C compiler were used to program the robots. Figure 3 contains a diagram illustrating the hardware architecture of the multi-robot testbed.
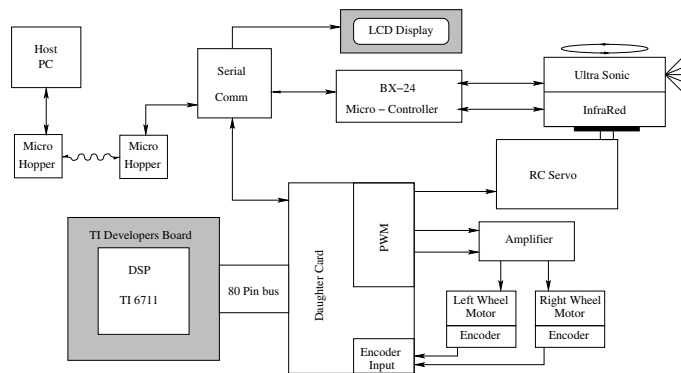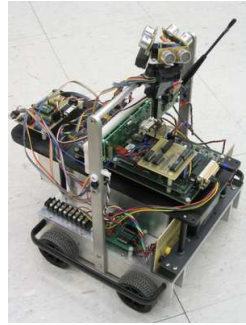

Fig. 3.   Hardware diagram of multi-robot testbed.

IV. DISTRIBUTED SONAR SENSING

A. Implementation limitations

It is important to emphasize the constraints imposed by the problem definition on the implementation. Firstly, the algorithms are to be completely decentralized. Hence, although the robots might communicate among themselves or with a base station, their behavior must not depend on this. Secondly, due to the dynamic nature and undetermined size of the environment, the robot cannot expect to build and keep an

accurate updated map. The occupancy grid methods described by [7], [8] do account for a changing environment but several readings are required to remove an object from the map if it has moved. This is not acceptable in this implementation. Decisions concerning the environment surrounding the robot must be made from only *one* horizon of sonar readings at *one* location. This was the main stimulus to improve single point resolution in sonar scans.

B. Sonar advantages and disadvantages

The Daventech SRF04 Ultrasonic range sensor has several distinct advantages over the Polaroid 6500 sensors which have commonly been used, e.g., see [9], [7], [8]. These advantages include a larger range of measurements (3cm to 6m), lower power consumption (during both firing and quiescent periods), and smaller size. Any ultrasonic sensor comes with a set of problems [10], [11] such as wide beam angle, interference, multiple and specular reflections. Much of the previous work on these issues has focused on map building, e.g., [12] and path planning problems, where probabilistic approaches [8], Kalman filters [12], and stereoscopic approaches [13] have been shown to significantly improve the overall scanning performance. Unfortunately, the distributed aspects of our setting limits the applicability of these approaches.

C. Improving scanning reliability

One of the problems with ultrasonics sensors is direct and indirect interference from stray sonar pulses and reflections causing *very* erroneous readings. It is a serious factor in sonar usage and occurs when vagrant sonic pulses are mistaken for an intended objects reflection. This problem is partially addressed by the error eliminating rapid ultrasonic firing (EERUF) procedure described by Borenstein and Korne in [5]. In the approach sonar sensors are fired in a sequence with different timing. The timings differ between all sonars on the same robot as well as between successive firings of the same sonar. A *predetermined* timing schedule for a *known* number of sensors is developed. The variable timing scheme ensures that interference in successive readings is not constant. Scanning then continues until a pair of readings agree within some tolerance. Borenstein and Korne implement EERUF with a total of 32 sonar firing on two separate robots. Their algorithm successfully rejects 97% of all erroneous readings that would affect naive firing methods. The work [5] contains little indication of the measurement length at which these results are achieved; to the best of our understanding it *appears* that these successful results were obtain for distances typically less than $100cm$. This is an important consideration for the comparison of results in Section IV-D.

In a truly distributed system it is difficult to determine the global number of sensors as well as to obtain global synchronization of the robot's internal clocks and hence it is very difficult to devise a *global* timing schedule for sonar firing. In addition, in this investigation it was found that in an environment with several (5) *static* mobile robots each with 3 sonar implementing the EERUF method there was sufficient
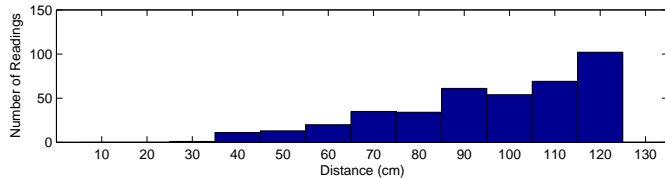
Fig. 4. Error distribution at a distance of 120cm with 15 Sonar mounted on 5 robots in the environment.

cross talk to continuously cause interference and often prevent two successive sonar readings from being within tolerance. The result is very long delays (in the order of seconds) for obtaining a valid EERUF reading. In other words the EERUF method appears to be not scalable in distributed systems.

To overcome these problems, we propose, and later compare, a number of single-point sonar-scan algorithms. We refer to the following scheme as the Queue Checker algorithm. Let $\epsilon > 0$ be a tolerance, and let n be the size of a queue where we record successive sonar readings in a FIFO fashion. If a new reading differs by less than ϵ from a certain number $k < n$ of readings in the queue, then the reading is accepted as a valid Queue Checker measurement. In addition, the firing schedule is *randomized* and halts as soon as a valid measurement is obtained. This allows other sensors to fire alone, faster, and with less probability of interference. The randomization reduces the possibility of consecutive interference occurring as a result of other sonar firing at the same rate. The performance of the algorithm can be tuned by appropriate selections of the parameters ϵ, k, and n.

After considering a set of naive sonar data as in Figure 4 it is evident that the measurements are strongly left skewed, i.e., erroneous measurements are always *smaller* than the actual distance. Hence, the *greatest* distance observed is relative to the *closest* object. Using this observation, the Max Filter filter collects a series of sonar distance readings and chooses the maximum value as a valid measurement.

A further means to improving the sonar performance is to adjust the sensitivity and maximum range of the sensors. Limiting the maximum sensitivity ignores weaker reflections and interference. Scanning is also faster as the sonar does not have to wait as long to detect reflections. The sensitivity can be set dynamically and was implemented throughout this paper.

D. Experimental performance of sensing algorithms

The performance of each sonar scan algorithm was tested in a reproducible environment that simulated the testbed environment. The experiments were conducted with a variety of distances, objects and number of sonar. A brief summary of the results are shown in Table I.

Considering Table I it is significant to note the following:

- Unfiltered sonar measurements produce very poor measurements particularly with longer distance measurements.

- The EERUF method proves to be reliable but slow with respect to the other methods. This is because a *sequence* of readings differ from one another according to the overall error distribution; see Figure 4. However, the EERUF algorithm was found to be more effective at shorter distances (<100cm); this is consistent with the results reported in [5].
- The Queue Checker approach improves the readings reliability only at short distances. At larger distances the measurement error distribution is such that it is more likely to match a series of incorrect readings than select the correct distance. In comparison, the Max Filter algorithm improves reading reliability at all distances.
- The time required for a reading using the Queue Checker method increases exponentially with increasing number of values to match where as the Max Filter time increases linearly.

E. Improving scanning resolution

A persisting problem of using sonar is the wide beam angle as shown in Figure 5. In many approaches to improving
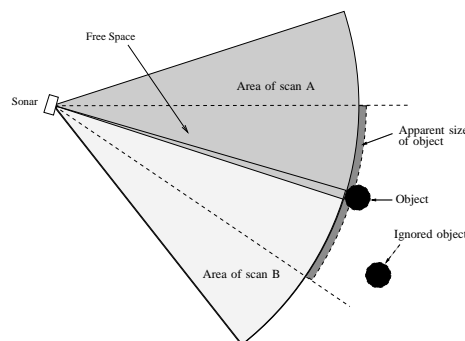

Fig. 5. As a result of the wide beam angle (i) distant objects are not observed and (ii) the 'size' of an object is increased.

sonar resolution the robot moves through an environment and uses readings from multiple, individual and subsequent sonar firings, from different locations, to improve the resolution of the observed objects. A popular approach is the *occupancy grid* method developed by Moravec and Elfes [8] in which the probability of an object being located in discretized regions of space is computed. Certainty is 'added' along an arc at the distance recorded by the sonar (the 'apparent object size' in Figure 5), and 'subtracted' in wedge shape of free space between the robot and the object. A recent approach by Choset, Nagatani and Lazar [9] represents each wide sonar reading as a single arc. Intersections of subsequent arcs then indicate the location of an object. McKerrow [14] proposes fitting line segments through successive sonar arcs that met certain criteria. The result of all of these methods is a representation of the environment that is significantly better than a naive sonar model. However, their application to our problem is severely limited as there is only *one* set of sonar readings from an unknown location available at any time (cf. Section IV-A).

As a result of these limitations, we developed a new ap-

TABLE I

SUMMARY OF SONAR TEST RESULTS FOR THE MAX FILTER AND QUEUE CHECKER METHODS. 10 ELEMENTS WERE USED IN THE QUEUE CHECKER METHOD. THE TIME ENTRIES WERE RECORDED FOR SCANNING AT 120CM.

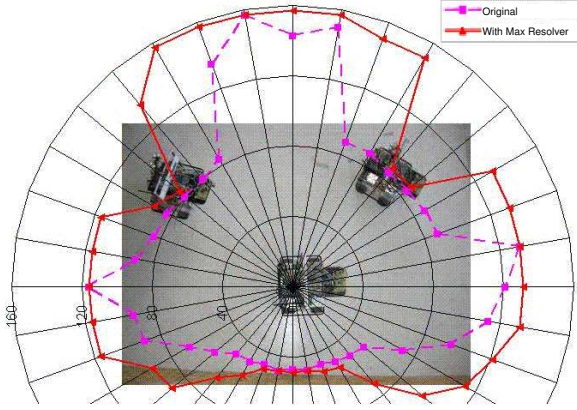|			Percentage errors in readings								
Actual Distance (cm)	Sonar Alone	EERUF	Number to match in Queue Checker				Number of readings for Max Filter				
			1	2	3	4	2	3	4	12	20
80	18%	0%	7%	4%	4%	4%	3%	1%	0%	-	-
100	47%	4%	41%	32%	23%	20%	40%	24%	17%	4%	-
120	66%	18%	83%	89%	92%	98%	53%	49%	41%	9%	2%
Time (Relative)	0.5	40	1.3	2.5	4.3	9.9	0.8	1.1	1.4	3.6	5.9

Fig. 6. Illustration of improvement scanning resolution using the Max Resolver algorithm.

proach called the Max Resolver in which a set of adjacent readings in a single scan are compared and used to improve the resolution of the complete horizon of distance measurements. The effect of a wide beam angle is to record a distance as the minimum distance to *any* object within the beam angle as shown in Figure 5. This has the effect of enlarging an object in the sonar view. The Max Resolver algorithm uses the maximum distance in a set of readings that would lie within the beam angle as the actual distance. This approach attempts to "invert" the wide beam angle effect and has the effect of "sharpening" the sonar readings. Figure 6 shows the original and modified sonar distance measurements and illustrates the effect of implementing the Max Resolver on the robot in the actual environment.

V. EXPERIMENTAL RESULTS

The steps in implementing the algorithm are outlined below:

(i) The micro-controller fires the 3 ultrasonic sensors for use in the Max Filter or Queue Checker algorithms.

(ii) A valid reading is returned to the DSP and stored. The RC servo then rotates the sonar to obtain a full 360° horizon of distance measurements. Ultrasonic scanning is halted.

(iii) The Max Resolver algorithm modifies the distance horizon and finds the centroid of the closest object. Neighboring measurements are incorporated into the object if they are sufficiently close and the the new object centroid is computed. (Object size is limited to 90° subtended at the sonar.)

(iv) A correction to ensure that the robot does not perceive an adjacent wall segment as the next closest object is applied and the previous step is repeated until 3 unique objects have been identified.

(v) If the distance to all *three* objects are within tolerance the robot stays in the "central region." Otherwise it moves away from the closest object (or bisector of the two closest) according to (2a).

(vi) Visualization data is sent to the PC and the robot returns to step 1.

In some implementations only one robot would scan at any time. This was done to reduce the possibility of sonar interference but was slow. The algorithms were successfully implemented with up to 5 robots as shown in Figure 7. There are several tradeoffs that need to be made for a successful implementation:

Number of readings: 36 was sufficient. Fewer readings caused the Max Resolver algorithm to eliminate some objects while taking more increased scan time but did not significantly increase accuracy.

Size of central region: Should initially be large and then gradually reduced. This reduces close proximity collisions due to robots moving simultaneously but slows the performance.

Distance to Move: Reducing the distance moved each step takes longer but reduces close quarter collisions and produces behavior which closely represents the simulated results.

Parameters of Max Filter: Reliable and fast results were obtained for $\epsilon = 5cm$, $k = 4$, and $n = 10$.

Sonar Sensitivity and Range: Can be reduced for speed and accuracy but needs to be dynamically adjusted.

Robot Recognition: To implement the 1/2 distance rule for the sphere packing (Remark 2.1) robots and wall objects must be identified. Leonard and Durrant-Whyte [10] use *regions of constant depth* (RCD). In this investigation a RCD of up to 60° was found for walls and 20° to 45° for a robot. Hence, any object smaller than 45° was considered to be a robot. This was successful but sometimes inconsistent.

Local minima play a larger role in the implementation (compared to simulation) because the introduction of a central region enlarges the space where a robot is in equilibrium.
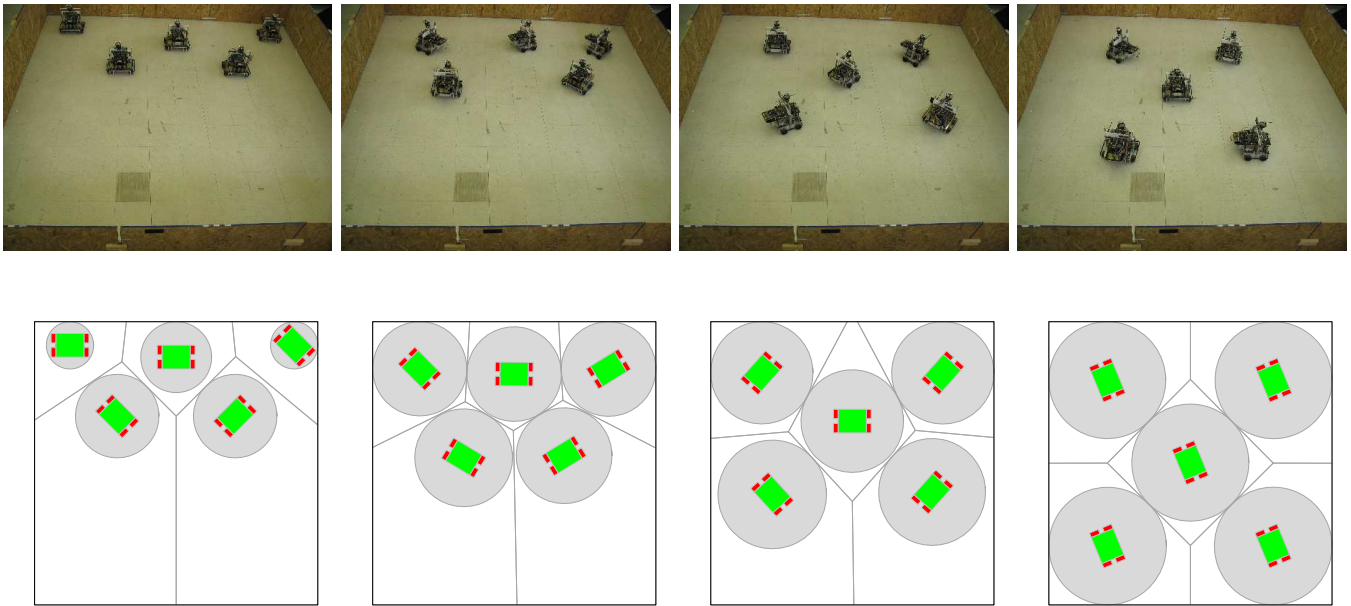
Fig. 7. The upper left (right) figures illustrate the initial (final) locations of 5 robots in a typical 5 robot implementation. The lower figures represent a computer simulation ran in `Mathematica` with identical initial condition.

A typical run (Figure 7) took about 3 minutes but can be reduced to less than a minute by making the tradeoffs discussed above. Finally, there are often several possible equilibrium configurations with different regions of attraction. While at equilibrium an erroneous sonar readings might cause an unexpected movement. However, quite conveniently, the system was found to often settle into another equilibrium configuration with a larger region of attraction.

VI. CONCLUSIONS

We have formalized optimal sensor coverage as an aggregate performance metric via methods from geometric optimization. We characterized the asymptotic behavior of the "move away from your closest neighbor" algorithm using tools from nonsmooth analysis. We have presented experimental results from a fully distributed implementation on a sonar based testbed. These successful results were dependent on good sonar sensor data and two novel methods: the `Max Filter` and the `Queue Checker`. We found the `Max Filter` algorithm to be faster and more reliable at longer distances than the `EERUF` and `Queue Checker` methods. We also develop and implement the `Max Resolver` algorithm for improving single point sonar scanning resolution. Our experimental setup provides valuable insight into practical issues regarding the realization of distributed sensor based exploration algorithms. We refer the interested reader to [15] for more details.

Acknowledgments

REFERENCES

[1] T. Balch and R. Arkin, "Behavior-based formation control for multirobot systems," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[2] A. Howard, M. J. Matarić, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, 2002.

[3] H. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking agents with varying interconnection topology," *Automatica*, July 2003. Rejected.

[4] J. Cortés and F. Bullo, "Coordination and geometric optimization via distributed dynamical systems," *SIAM Journal on Control and Optimization*, vol. 44, no. 5, pp. 1543–1574, 2005.

[5] J. Borenstein and Y. Koren, "Error eliminating rapid ultrasonic firing for mobile robot obstacle avoidance," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 132–138, 1995.

[6] A. F. Filippov, *Differential Equations with Discontinuous Righthand Sides*, vol. 18 of *Mathematics and Its Applications*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1988.

[7] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal Robotics Automation*, vol. 3, pp. 249–265, 1987.

[8] H. Moravec and A. Alfes, "High resolution maps for wide angle sonar," in *IEEE Int. Conf. on Robotics and Automation*, (St. Louis, MO), pp. 116–121, Mar. 1985.

[9] H. Choset, K. Nagatani, and N. A. Lazar, "The arc-transversal median algorithm: a geometric approach to increasing ultrasonic sensor azimuth accuracy," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 513– 521, 2003.

[10] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*, vol. 175 of *Kluwer International Series in Engineering and Computer Science*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1992.

[11] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, & Cybernetics*, vol. 19, no. 5, pp. 1179–1187, 1989.

[12] B. Schiele and J. L. Crowley, "A comparison of position estimation techniques using occupancy grids," in *IEEE Int. Conf. on Robotics and Automation*, (San Diego, CA), pp. 1628–1634, May 1994.

[13] J. Borenstein and S. Shoval, "Angular position measurement using ultrasonic sensors," in *Proceedings of the ANS Conference on Robotics and Remote Systems*, (Pittsburgh), Apr. 1999. Electronic Proceedings.

[14] P. J. McKerrow, "Echolocation: From range to outline segments," *Robots and Autonomous Systems*, vol. 11, no. 4, pp. 205–211, 1993.

[15] C. L. Robinson, "Distributed sensor-based exploration algorithms," Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL, Dec. 2003.