Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Adaptive Self-configurable Robot Swarms Based on Local Interactions

Geunho Lee, *Student Member, IEEE* and Nak Young Chong, *Member, IEEE*

***Abstract*— This paper presents a motion planning framework for a large number of autonomous robots that enables the robots to configure themselves adaptively into an area of an arbitrary geometry. A locally interacting geometric technique provides a unique solution that allows the robots to converge to the uniform distribution by forming an equilateral triangle with their two neighbors. The basic idea underlying the proposed solution is that robots can be thought of as liquid particles that change their relative positions conforming to the shape of the container they occupy. Specifically, it is assumed that robots are not allowed to have the identification number, a pre-determined leader, a common coordinate system, and communication capabilities. Under such minimal conditions, the convergence of the algorithm is mathematically proved and verified through extensive simulations. The results validate the feasibility of applying the algorithm to self-configuration of mobile sensors across the constrained environment.**

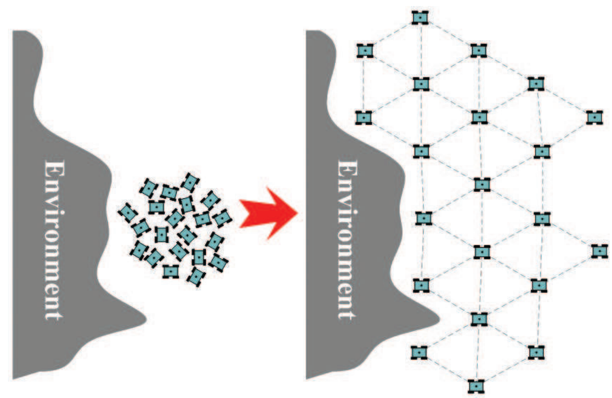


Fig. 1. Concept of adaptive self-configuration in an unknown environment

I. INTRODUCTION

Swarm robotics [1] is gaining increasing attention because a robot swarm is expected to perform a variety of real applications such as environmental or habitat monitoring, exploration, search-and-rescue, and so on. In order to enable a robot swarm to perform the aforementioned tasks adapting to an environment, a motion planning framework is needed for the robots to determine their relative positions from an arbitrary initial distribution. Such frameworks mostly use a balance between inter-individual interactions based on the observations from an organism of animals and insects, or physical phenomena in nature, that we call respectively behavior-based [2-3] and physics-based [4-11] approaches.

Balch and Hybinette [2] suggested the notion of social potentials to achieve robot formations mimicking the process of forming a crystalline structure that holds the molecules into place. Martison and Payton [4] proposed the virtual line force to deploy robots into a regular lattice. Spears *et al.* [5] developed a physics-based framework to achieve the desired deployment using the gravitational force model. Shucker and Bennett [6] introduced the virtual spring forces to maximize coverage and uniformity using the acute angle algorithm. Likewise, many algorithms for mobile sensor network deployment use different types of force, including electromagnetic forces [7], inter-molecular forces [8], and virtual potential fields [9]. These approaches require an effort to adjust parameters to obtain the desired behavior of self-configuration. Most importantly, to the best of our knowledge, no research has been performed under the geometric constraint of the environment.

In this paper, we address adaptive self-configuration of a robot swarm that enables a large number of robots with limited ranges of sensing to configure themselves into a 2-dimensional plane from an arbitrarily initial distribution. Through local interactions between individual robots that attempt to form an equilateral triangle, a robot swarm can eventually be deployed conforming to the shape of the environment as illustrated in Fig. 1. This will provide a systematic approach to adapting to an unknown environment regardless of limited sensing and communication capabilities of the robots. In practice, this adaptive self-configuration enables a robot swarm to strive toward achieving its mission in the presence of changes in environments. For instance, a robot swarm can maintain local geometric configurations while navigating through an environment populated with obstacles [12]. The main contribution of this paper is to provide a simple, distributed swarm self-configuration algorithm that exhibits self-organizing and self-stabilizing features.

The rest of this paper is organized as follows. Section II presents the assumptions about the robot and the definitions of the adaptive self-configuration problem. Section III describes the fundamental concepts in local interaction and the convergence properties of the algorithm. Section IV addresses the proposed adaptive self-configuration algorithm and its properties under geometric constraints. Section V provides the results of the simulations and discussions. Section VI draws conclusions.

II. PROBLEM STATEMENT

We consider a swarm of autonomous mobile robots $\{r_1, \cdots, r_n\}$. Each robot is modeled as a point, that freely

Manuscript received April 5, 2007; revised August 6, 2007

The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan (phone: +81-761-51-1248; fax: +81-761-51-1149; e-mail: {geun-lee, nakyoung}@jaist.ac.jp)

1-4244-0912-8/07/$25.00 ©2007 IEEE.

moves on a 2-dimensional plane with limited ranges of sensing. The robots have no leader and no prior knowledge of their identification number. They do not share any common coordinate system, and do not retain any memory of past actions. They can detect the position of other robots in close proximity, but are not allowed to communicate explicitly with them. Instead of the direct communication method, robots are able to locally interact by observing locations of other robots. Each of the robots executes the same algorithm, but acts independently and asynchronously from other robots. Time is represented as an infinite sequence of discrete instants $t_1, t_2, \cdots$. At a time instant, one of the following three actions will be taken by the robots:

- **Observation**: The robot r_i detects the position $\{p_1, p_2, \cdots\}$ of other robots located within its sensing range *SB*, and makes the observation set O_i of the obtained positions with respect to its local coordinate system.
- **Computation**: r_i performs the computation according to the local interaction algorithm to be proposed, yielding the target position p_{ti}.
- **Motion**: r_i moves to p_{ti} and returns to the observation state.

Each of the robots repeats an endless activation cycle of observation, computation, and motion. Now the adaptive self-configuration problem of a swarm of mobile robots in this work can be stated as follows.

Given that a swarm of robots $r_1, \cdots, r_n$ located at arbitrarily distinct positions in a two-dimensional plane, how can the robots configure themselves into equilateral triangular lattices adapting to the environment?

It is assumed that the robots can exactly determine the position of other robots using sonar [13] or infrared detectors [5], and distinguish between other robots and obstacles in the environment.

III. Local Interaction

This section describes our local interaction algorithm that enables to generate an equilateral triangular lattice by cooperation of three neighboring robots (See ALGORITHM-1).

A. Description of the Local Interaction Algorithm

All robots are initially located at distinct positions. Among them, consider a robot r_i and its two neighbors $s1$ and $s2$ located within r_i's sensing boundary *SB*. Hereafter, we denote the constant uniform distance interval by d_u, and the position of r_i, $s1$, and $s2$ by p_i, p_{s1}, and p_{s2}, respectively. As shown in Fig. 2-(a), three robots configure into a triangle whose vertices are p_i, p_{s1}, and p_{s2}, respectively. Fig. 2-(b) illustrates that r_i finds the centroid p_{ct} of the configured triangle $\triangle p_i p_{s1} p_{s2}$ with respect to its local coordinates, and measures the angle ϕ between the line connecting two neighbors and r_i's horizontal axis. Using p_{ct} and ϕ, r_i calculates its target position p_{ti}.

ALGORITHM-1 LOCAL INTERACTION (code executed by the robot r_i at the point p_i)

constant d_u := uniform distance
FUNCTION $\varphi_{interaction}(\{p_{s1}, p_{s2}\}, p_i)$
$(ct_x, ct_y) := centroid(\{p_{s1}, p_{s2}, p_i\})$
ϕ := angle between $\overline{p_{s1}p_{s2}}$ and r_i's local horizontal axis
$target_x := ct_x + d_u \cos(\phi \pm \pi/2)/\sqrt{3}$
$target_y := ct_y + d_u \sin(\phi \pm \pi/2)/\sqrt{3}$
$p_{ti} := (target_x, target_y)$

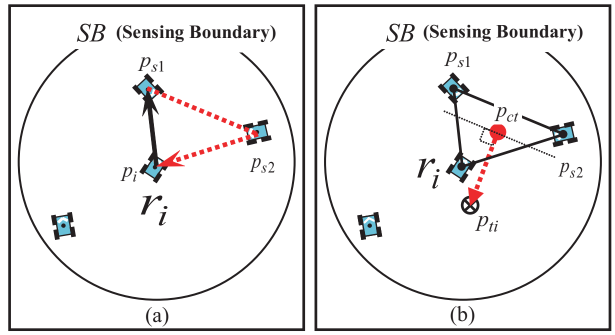


Fig. 2. Illustration of ALGORITHM-1 (a) triangular configuration, (b) target point computation

Consider a triangle with three vertices p_a, p_b, and p_c that represent the position of three robots A, B, and C as shown in Fig. 3. Let α, β, and γ denote the internal angles of the triangle, respectively. Each robot located at the vertex of $\triangle p_a p_b p_c$ may move to the new position p_{ta}, p_{tb}, and p_{tc} computed by ALGORITHM-1. The internal angles of $\triangle p_{ta} p_{tb} p_{tc}$ are α', β', and γ', respectively. Let p_{ct} denote the centroid of $\triangle p_a p_b p_c$. Also, let p denote the point projected from p_{ct} onto $\overline{p_a p_b}$. Similarly, let q indicate the point projected from p_{ct} onto $\overline{p_a p_c}$. If we consider a quadrangle $p_a p p_{ct} q$, the angles of p and q are right angles. Therefore, $\angle p p_{ct} q$ becomes $180° - \alpha$. From Fig. 3, $\angle p_{tb} p_{ct} p_{tc}$ is equal to $\angle p p_{ct} q$. $\triangle p_{tb} p_{ct} p_{tc}$ is an isosceles triangle since $\overline{p_{ct} p_{tb}}$ and $\overline{p_{ct} p_{tc}}$ is identical ($d_u/\sqrt{3} = \sqrt{3}/2 \times d_u \times 2/3$). Hence, α of $\triangle p_a p_b p_c$ is equal to $2a$ in the figure. With the same manner, β and γ become $2b$ and $2c$, respectively. Moreover, we see that α' of $\triangle p_{ta} p_{tb} p_{tc}$ is $(\beta + \gamma)/2$ (or equal to $(b + c)$). Likewise, β' indicates $(\alpha + \gamma)/2$ and γ' does $(\alpha + \beta)/2$. Accordingly, α' is given by $(\beta + \gamma)/2$. Now the relation between internal angles can be rewritten as a function of time to give the following equation:

$$\alpha(t+1) = (\beta(t) + \gamma(t))/2, \tag{1}$$

where t and $t+1$ represent the current time instant and the next time instant. Thus, the internal angle of r_i at $t+1$ is obtained by dividing the sum of internal angles of two neighbors observed at t with 2. Intuitively, r_i may maintain d_u with two neighbors. In other words, each robot attempts to form an isosceles triangle at each time instant, and by repeatedly doing this, three robots configure into an equilateral triangle with a side length d_u.

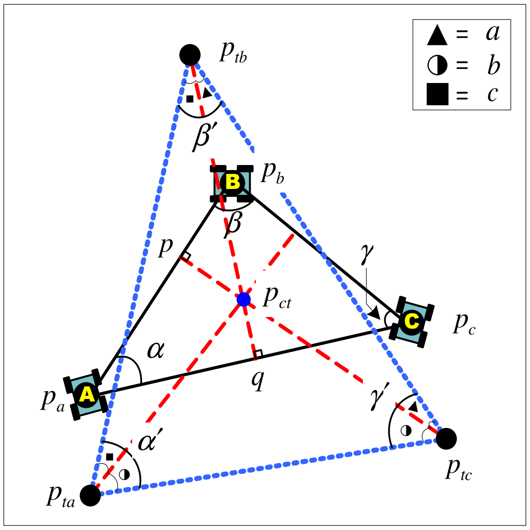


Fig. 3. Robots attempt to form an isosceles triangle

B. Convergence of Local Interactions

Let's consider the circumscribed circle of an equilateral triangle whose center is p_{ct} of $\triangle p_i p_{s1} p_{s2}$ configured from three positions occupied by three robots and radius is $d_u/\sqrt{3}$. Under the local interaction algorithm, motion planning for the robots is performed by controlling the distance from p_{ct} and the internal angle (See Fig. 4-(a)).

First, the distance is controlled by the following equation

$$\dot{d}_i(t) = -a\Big(d_i(t) - d_r\Big), \tag{2}$$

where a is a positive constant and d_r represents the length $d_u/\sqrt{3}$. Indeed, the solution of (2) is $d_i(t) = |d_i(0)|e^{-at} + d_r$ that converges exponentially to d_r as t approaches infinity.

Secondly, the internal angle is controlled by the following equation

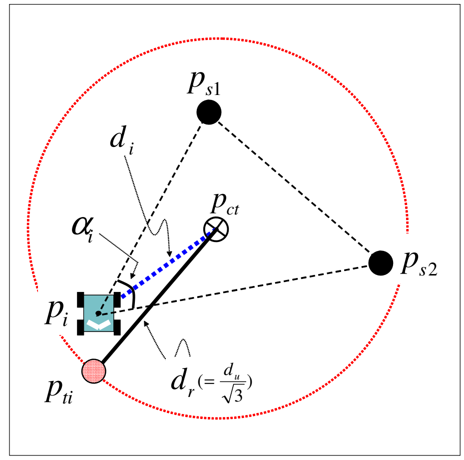
$$\dot{\alpha}_i(t) = k\Big(\beta_i(t) + \gamma_i(t) - 2\alpha_i(t)\Big), \tag{3}$$

where k is a positive number. Because the total internal angle of a triangle is $180°$, (3) can be re-written as

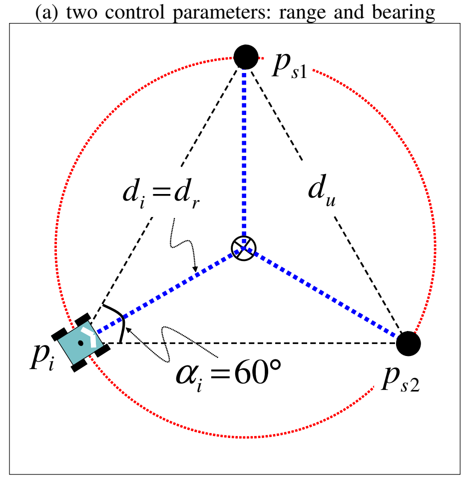
$$\dot{\alpha}_i(t) = k'\Big(60° - \alpha_i(t)\Big), \tag{4}$$

where k' is $3k$. Likewise, the solution of (4) is $\alpha_i(t) = |\alpha_i(0)|e^{-k't} + 60°$ that converges exponentially to $60°$ as t approaches infinity.

Note that (2) and (4) imply that the trajectory of r_i converges to d_r and $60°$, an equilibrium state shown in Fig. 4-(b). This also implies that three robots eventually form an equilateral triangle with d_u. In order to prove the correctness, we will take advantage of stability based on Lyapunov's theory [14]. The stability theorem states if there exists a scalar function $f_{l,i}$ of the state $\mathbf{x} = [d_i(t)\ \alpha_i(t)]^T$ with continuous first order derivatives such that $f_{l,i}$ is positive definite, $\dot{f}_{l,i}$ is negative definite, and $f_{l,i} \to \infty$ as $\parallel \mathbf{x} \parallel \to \infty$, then equilibrium at a specific state $[d_r\ 60°]^T$ is asymptotically stable. The desired configuration is one that minimizes the energy level of the scalar function.



(a) two control parameters: range and bearing



(b) desired equilateral triangular configuration

Fig. 4. Illustration of two control parameters in local interaction

Consider the following scalar function:

$$f_{l,i} = \frac{1}{2}(d_i - d_r)^2 + \frac{1}{2}(60° - \alpha_i)^2 \tag{5}$$

This scalar function is always positive definite except $d_i \neq d_r$ and $\alpha_i \neq 60$. The derivative of the scalar function is given by

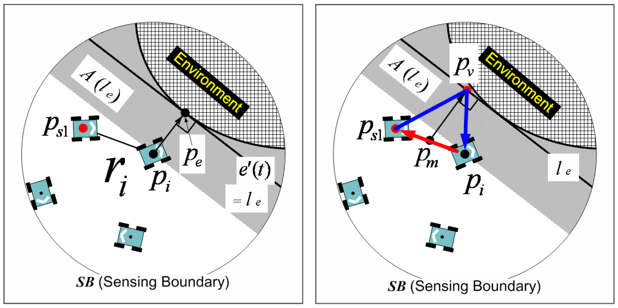
$$\dot{f}_{l,i} = -(d_i - d_r)^2 - (60° - \alpha_i)^2, \tag{6}$$

which is obtained by differentiating $f_{l,i}$ using (2) and (4) to substitute for $\dot{d}_i$ and $\dot{\alpha}_i$. Eq. (6) is negative definite. The scalar function $f_{l,i}$ is radially unbounded since it tends to infinity as $\parallel \mathbf{x} \parallel \to \infty$. Therefore, the equilibrium state is asymptotically stable, implying that r_i reaches a vertex of the desired triangle.

Now we prove the convergence of the algorithm for n robots. The n-order scalar function $\mathbf{F}$ is defined as

$$\mathbf{F} = \sum_{i=1}^{n} f_{l,i}\Big(d_i(t),\ \alpha_i(t)\Big). \tag{7}$$

It is straightforward to verify that $\mathbf{F}$ is positive definite and $\dot{\mathbf{F}}$ is negative definite. $\mathbf{F}$ is radially unbounded since it tends to infinity as t approaches infinity. Consequently, n robots move toward the equilibrium state.



(a) detecting an environment (b) approaching an environment

Fig. 5. Illustration of uniform adaptation algorithm

Finally, as mentioned earlier, the robots have no memory (oblivious). Hence, the algorithm uses the function $\varphi_{interaction}$ whose arguments consist of the position set of the robot and its two neighbors at the current time instant. The return value is the target position of the robot at the next time instant. It has been proven that the oblivious strategy yields a self-stabilizing algorithm [1] [15].

IV. Adaptive Self-Configuration

This section describes how to deploy the robots at a uniform interval conforming to the geometry of the environment using local interactions. We assume that the geometry of the environment can be represented by a continuous function.

A. Description of Adaptive Self-Configuration Algorithm

At the time instant t, r_i detects the first neighbor $s1$ located the shortest distance. First of all, we assume that the surface geometry of an environment can be represented by a continuous function $e(t)$ without discontinuity. As illustrated in Fig. 5-(a), when detecting the environment, r_i defines a point p_e projected from p_i onto the environment surface with the minimum distance d_e and computes the tangent $e'(t)$ to the environment surface at p_e. It is obvious that $e'(t)$ is perpendicular to the vector $\overrightarrow{p_i p_e}$, termed the environment direction. Let $A(l_e)$ denote the area in the environment direction within *SB*. That is, $A(l_e)$ is the area between the surface of the environment and the line passing through p_i and parallel to $e'(t)$. If no neighbors exist in $A(l_e)$ or if the condition $d_e \leq \frac{\sqrt{3}d_u}{2}$ is satisfied, in order to approach the environment, r_i computes the midpoint p_m of $\overline{p_i p_{s1}}$ from which the virtual point p_v is projected onto $e'(t)$. Now p_v is considered as p_{s2} as illustrated in Fig. 5-(b). Otherwise, to approach other robots, $s2$ is selected such that the total distance from p_{s1} to p_i through p_{s2} is minimized. Now with p_{s1} and p_{s2}, r_i can compute the next target point p_{ti} by $\varphi_{interaction}$ in ALGORITHM-1. When three robots are all aligned, the centroid p_{ct} is set to the center point of the line segment between p_{s1} and p_{s2}. If r_i is located on the line segment, p_{ti} is set to the point $\frac{\sqrt{3}d_u}{2}$ away from p_{ct}. Otherwise, p_{ti} is set to the point $\frac{d_u}{\sqrt{3}}$ away from p_{ct}.

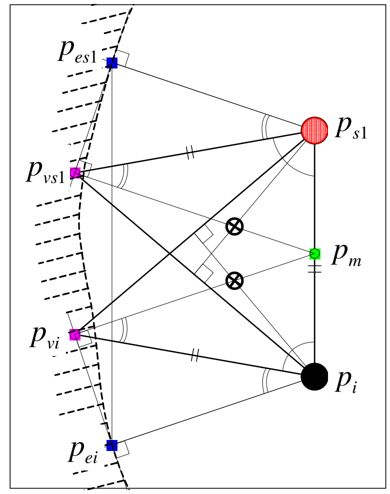
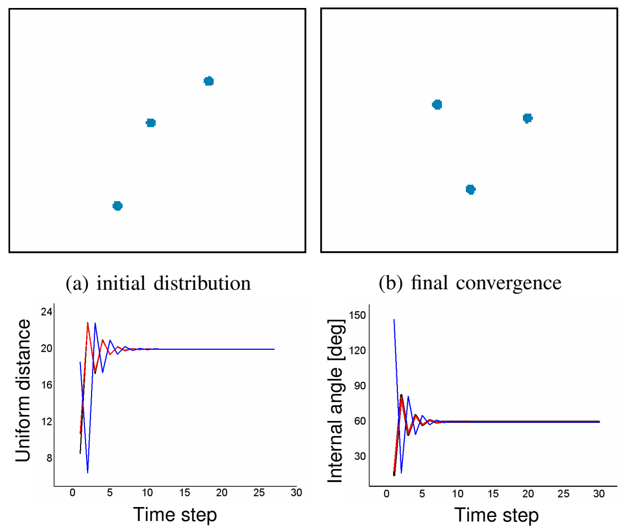


Fig. 6. Uniform adaptation to shapes of surfaces



(a) initial distribution (b) final convergence

(c) distance variation (d) internal angle variation

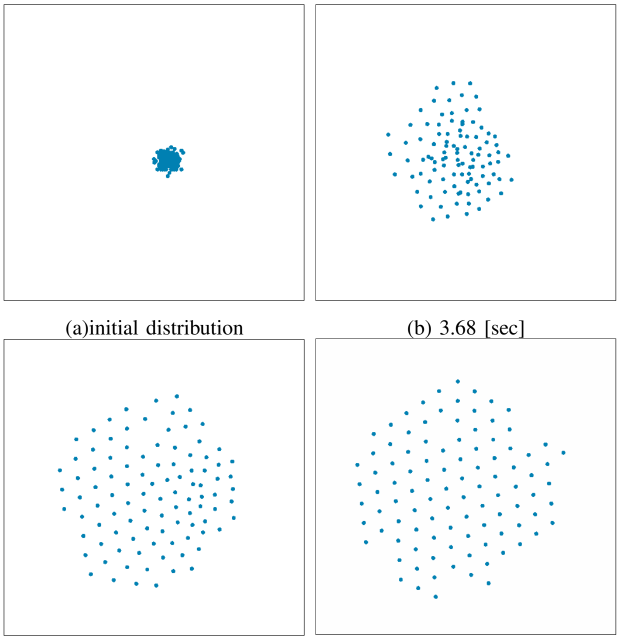
Fig. 7. Simulation results of self-configuration for three robots

B. Uniform Adaptation Property

It is verified in Section III that three neighboring robots are able to form an equilateral triangle with distance d_u. Without loss of generality, r_i can converge into an isosceles triangle with a neighbor and a virtual point. Now we show that the robots maintain a constant uniform distance d_u with each other while conforming to the geometry of the environment. As shown in Fig. 6, this requires that, the vector $\overrightarrow{p_i p_{s1}}$ with distance d_u should be parallel to $\overrightarrow{p_{ei} p_{es1}}$, where p_{ei} (or p_{es1}) indicates the point projected from p_i (or p_{s1}) onto the environment surface.

We first consider the case of the indented wall in Fig. 6. Let p_{vi} and p_{vs1} indicate the virtual point for r_i and $s1$, respectively. Note that, due to the limited ranges of SB, r_i is not able to identify whether $\overrightarrow{p_i p_{ei}}$ is parallel to $\overrightarrow{p_{s1} p_{es1}}$. According to the geometry of the environment, p_{vi} may

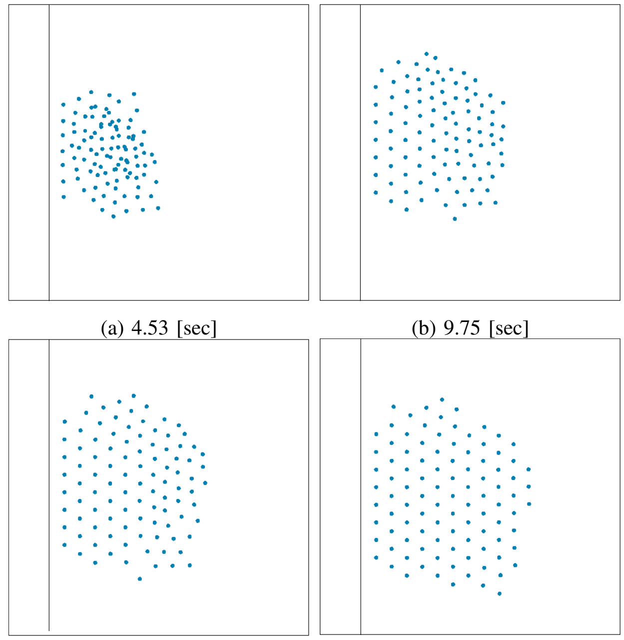
[1] Self-stabilization is the property of a system which, started in an arbitrary state, always converges toward a desired behavior [16] [17].



(a)initial distribution (b) 3.68 [sec]

(c) 6.60 [sec] (d) $d_{1\%}$: 25.55 [sec]

Fig. 8. Simulation results of self-configuration for 100 robots



(a) 4.53 [sec] (b) 9.75 [sec]

(c) 12.38 [sec] (d) $d_{1\%}$: 32.02 [sec]

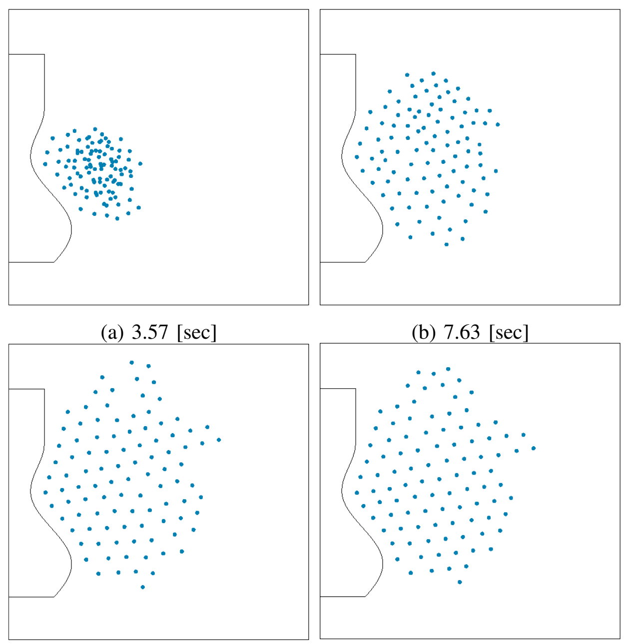
Fig. 9. Simulation results of self-configuration over a flat surface

vary since $e'(t)$ changes. Thus, p_{vi} and p_{vs1} may not be coincident.

By the convergence of local interactions, it is obvious that $\triangle p_i p_{s1} p_{vi}$ and $\triangle p_{s1} p_i p_{vs1}$ are the isosceles triangles with d_u (the length of $\overline{p_i p_{s1}}$, $\overline{p_i p_{vi}}$, and $\overline{p_{s1} p_{vs1}}$ are the same). Since the two triangles have the same height of $\frac{\sqrt{3}d_u}{2}$, they are congruent. Also, since $\angle p_{s1} p_i p_{vi}$ and $\angle p_i p_{s1} p_{vs1}$, and $\angle p_{s1} p_{vs1} p_m$ and $\angle p_i p_{vi} p_m$ have the same measure, $\angle p_{vi} p_i p_{ei}$ and $\angle p_{vs1} p_{s1} p_{es1}$ have the same measure. Hence, $\triangle p_{vi} p_i p_{ei}$ and $\triangle p_{vs1} p_{s1} p_{es1}$ are congruent. Since $\overline{p_i p_{ei}}$ and $\overline{p_{s1} p_{es1}}$ have the same length and $\angle p_{s1} p_i p_{ei}$ and $\angle p_i p_{s1} p_{es1}$ have the same measure, the quadrangle $p_i p_{s1} p_{es1} p_{ei}$ is an isosceles trapezoid. Thus, it is readily evident that $\overrightarrow{p_i p_{s1}}$ is parallel to $\overrightarrow{p_{ei} p_{es1}}$. The conformity condition for the case of the flat wall can be satisfied with the same manner.

V. Simulation Results

The proposed self-configuration algorithm terminates when all robots converge into the distance $d_u \pm 1\%$ with their two neighbors, which is denoted as $d_{1\%}$. Fig. 7 shows that how three robots converge into an equilateral triangle, where Fig. 7-(a) and (b) display the initial and the final position of the robots. Fig. 7-(c) and (d) indicate the variations in the distance and the internal angle that eventually converge into d_u and 60°. Fig. 8 shows that 100 robots configure themselves into a uniformly distributed pattern with distance interval d_u over the empty plane. We tested extensive simulations in a variety of initial distributions and compared the total deployment time. For 30 kinds of initial distributions, the deployment time is summarized as follows: 27.36 [sec] for $d_{1\%}$, 42.86 [sec] for $d_{0.1\%}$, and 61.48 [sec] for $d_{0.01\%}$. Specifically, each robot interacts with only two neighbors, which ensures that the motion of the robot is less affected than other approaches employing a large number of neighbors and the computational load decreases.

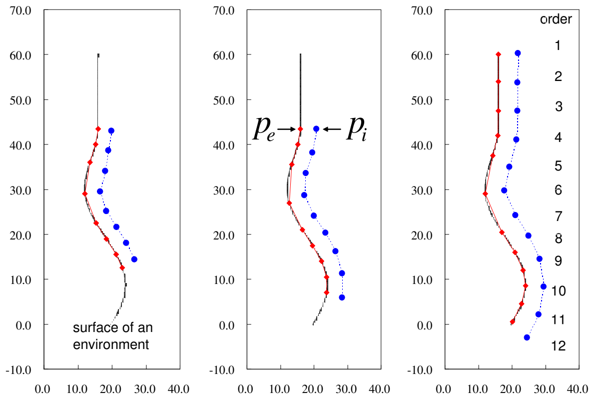


(a) 3.57 [sec] (b) 7.63 [sec]

(c) 13.64 [sec] (d) $d_{1\%}$: 41.54 [sec]

Fig. 10. Simulation results of self-configuration over a curved surface

Figs. 9 and 10 show that how 100 robots self-configure into a geometrically-constrained environment. As mentioned in Section IV, all robots could eventually converge to the uniformly distributed position conforming to the environment geometry. It is evident from Fig. 10 that even the robots that do not detect the environment was able to conform to the geometry of the environment by just interacting with their neighbors. Fig. 11 shows the simulation results when



(a) d_u: 3.5 (b) d_u: 5.5 (c) d_u: 7.5

Fig. 11. Surface conforming of Fig. 10 according to the changes in distance

TABLE I

COMPARISON OF GRADIENT OF LINE SEGMENTS IN FIG. 11 [DEG]

order	Fig. 11-(a)		Fig. 11-(b)		Fig. 11-(c)	
	grad. of p_e	grad. of p_i	grad. of p_e	grad. of p_i	grad. of p_e	grad. of p_i
1-2	78.055	77.186	78.055	77.946	90.000	89.607
2-3	67.071	67.871	67.028	66.688	90.000	89.956
3-4	77.983	76.983	84.868	84.492	87.938	87.395
4-5	-63.524	-63.520	-56.535	-57.197	70.676	71.112
5-6	-49.474	-50.151	-48.831	-49.309	75.748	75.723
6-7	-50.205	-51.387	-52.975	-53.566	-59.991	-59.801
7-8	-58.213	-58.116	-66.250	-67.583	-49.421	-49.693
8-9			-86.746	-87.205	-58.376	-57.714
9-10					-78.585	-78.331
10-11					74.721	75.175
11-12					55.529	56.340

the uniform distance is changed to (a) 3.5, (b) 5.5, and (c) 7.5, respectively. In the figure, the black bold line shows the outline of the environment. The blue dots indicate p_i of the robot and the red dots on the outline display p_e projected from p_i (see Fig. 5 in Section IV). The red dotted and blue solid line segments represent $\overline{p_{ei}p_{es1}}$ and $\overline{p_ip_{s1}}$, respectively, illustrated in Fig. 6. As expected, each robot could be distributed uniformly regardless of the changes in the uniform distance while conforming to the environment. Table I shows that the average error rate over the entire set of gradients is about 0.84 [%]. It is readily evident from the table that $\overline{p_ip_{s1}}$ is closely parallel to $\overline{p_{ei}p_{es1}}$. If we take the nonuniform curvature of the outline into consideration, the robots was able to conform as closely as possible to the uneven surface.

VI. CONCLUSION

In this paper, we presented a local interaction algorithm between neighboring robots, enabling a large-scale swarm of robots to self-configure into various two-dimensional planes. The robots were assumed to have no individual identification, no determined leader, no common coordinates, no memory for past actions, and no communication capability. They were allowed to interact with two dynamically selected neighbors by observing other robots in their sensing range. Based on the geometric approach to forming an equilateral triangle, the swarm could be uniformly self-deployed, and moreover adapt to an unknown environment. The proposed algorithm featuring decentralized, self-organized, and self-stabilizing design was proved mathematically and verified by simulations. Our analysis and simulation results show that the proposed adaptive self-configuration is a simple and efficient approach to uniform deployment of a robot swarm in a changing environment. As a first step toward real-world implementations, we intend to apply this algorithm to build an *ad hoc* mobile robotic sensor network with uniform spatial density where measurement errors exist.

REFERENCES

[1] E. Sahin. "Swarm robotics: from sources of inspiration to domains of application," SAB2004, E. Sahin and W. M. Spears (eds.), Lecture Notes in Computer Science, Vol.3342, Springer, pp.10-20, 2005
[2] T. Balch and M. Hybinette. "Social potentials for scalable multi-robot formations," Proc. IEEE International Conference on Robotics and Automation, pp.73-80, 2000
[3] B. Werger and M. J. Mataric. "From insect to internet: situated control for networked robot teams," Annals of Mathematics and Artificial Intelligence, Vol.31, No.1-4, pp.173-198, 2001
[4] E. Martison and D. Payton. "Lattice formation in mobile autonomous sensor arrays," SAB2004, E. Sahin and W. M. Spears (eds.), Lecture Notes in Computer Science, Vol.3342, Springer, pp.98-111, 2005
[5] W. Spears, D. Spears, J. Hamann, and R. Heil. "Distributed, physics-based control of swarms of vehicles," Autonomous Robots, Vol.17, No.2-3, pp.137-162, 2004
[6] B. Shucker and J. K. Bennett. "Scalable control of distributed robotic macrosensors," Proc. 7th International Symposium on Distributed Autonomous Robotic Systems, 2004
[7] A. Howard, M. J. Mataric, and G. S. Sukhatme. "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," Proc. 6th International Symposium on Distributed Autonomous Robotic Systems, pp.299-308, 2002
[8] N. Heo and P. K. Varshney. "A distributed self spreading algorithm for mobile wireless sensor networks," Proc. IEEE Wireless Communication and Networking Conference, 2003
[9] Y. Zou and K. Chakrabarty. "Sensor deployment and target localization based on virtual forces," Proc. IEEE Infocom Conference, 2003
[10] R. Cohen and D. Peleg. "Local algorithms for autonomous robots systems," Sirocco2006, P. Flocchini and L. Gasieniec (eds.), Lecture Notes in Computer Science, Vol.4056, Springer, pp.29-43, 2006
[11] G. Wang, G. Cao, and T. L. Porta. "Movement-assisted sensor deployment," Proc. IEEE Infocom Conference, pp.2469-2479, 2004
[12] G. Lee, Y. Hanada, N. Y. Chong, and C. Kim. "Adaptive flocking algorithm for robot swarms: lessons from a school of fish," Proc. 13th International Conference on Advanced Robotics, 2007
[13] G. Lee and N. Y. Chong. "Decentralized formation control for a team of anonymous mobile robots," Proc. of 6th Asian Control Conference, pp.971-976, 2006
[14] J. E. Slotine and W. Li. *Applied nonlinear control*, Prentice-Hall, 1991
[15] I. Suzuki and M. Yamashita. "Distributed anonymous mobile robot: formation of geometric patterns," SIAM Journal of Computing, Vol.28, No.4, pp.1347-1363, 1999
[16] S. Dolev. *Self-Stabilization*, MIT Press, 2000
[17] M. Schneider. "Self-stabilization," ACM Computing Survey, Vol.25, No.1, pp.45-67, 1993