The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Three Dimensional Deployment of Robot Swarms

Geunho Lee, Yasuhiro Nishimura, Kazutaka Tatara, and Nak Young Chong

Abstract— **This paper addresses the deployment problem for a swarm of autonomous mobile robots initially randomly distributed in 3 dimensional space. A fully decentralized geometric self-configuration approach is proposed to deploy individual robots at a given spatial density. Specifically, each robot interacts with three neighboring robots in a selective and dynamic fashion without using any explicit communication so that four robots eventually form a regular tetrahedron. Using such local interactions, the proposed algorithms enable a swarm of robots to span a network of regular tetrahedrons in a designated space. The convergence of the algorithms is theoretically proved using Lyapunov theory. Through extensive simulations, we validate the effectiveness and scalability of the proposed algorithms.**

I. Introduction

Recently, increasing attention has been paid to autonomous operations of swarms of unmanned vehicles and sensors to provide airborne, undersea, and terrestrial surveillance and exploration. These applications often require individual agents to autonomously configure themselves into a designated space. To achieve and maintain such capabilities, many studies have addressed the effectiveness of decentralized coordination approaches in self-configuration [1][2], pattern generation [3], flocking and tracking [4][5][6] in flat 2 dimensional space. These approaches are mainly based on local interactions between individual robots with limited sensing and communication capabilities, and can be classified into biologically-inspired [3], behavior-based [2][7], and virtual physics-based [1][5][8] approaches. The behavior-based and virtual physics-based approaches use some sort of inter-robot force balance such as spring forces [1], gravitational forces [5], potential fields [8], and other forces. This is because the force-based interaction rules are considered simple yet effective, and provide an intuitive understanding on individual behavior. Moreover, such local interactions may result in lattice-type configurations that offer effective area coverage with redundant connections. Along the same line, but in 3 dimensional space, we propose a geometric approach that enables robots to organize a network of regular tetrahedrons using a partially-connected topology [9].

The 3 dimensional deployment of a robot swarm has been gaining recognition and popularity in very recent years. The 'I-SWARM' project [10] was an attempt to deploy micro-robot swarms for automatic execution of tasks in the small world. Michael *et al*. [11] proposed a planning and control scheme for robot swarms based on abstraction reducing its complexity in the 9 dimensional space which is a product structure of the 6 dimensional Euclidean group and a three-dimensional shape. In [12], several different swarm flocking solutions were investigated in two and three-dimensional spaces and their stability was analyzed. Paul *et al*. [13] presented a coordinated formation flight and reconfiguration of unmanned aerial vehicles (UAVs) based on potential fields using a virtual leader and considering the vehicle's velocities. Chaimowicz *et al*. [14] reported a hierarchical coordination architecture where a few number of UAVs is used to command, control, and monitor swarms of unmanned ground vehicles (UGVs) for urban search and rescue. Meanwhile, little attention has been paid to the configuration control of robot swarms in 3 dimensional space.

Our previous works on the 2 dimensional configuration control problem [2] relied on a set of artificial behavior rules, enabling robots to create equilateral triangle lattices. Through the local behavior rules, a swarm of robots could configure themselves into an area at a uniform interval, aiming to provide enhanced coverage and connectivity. In this paper, we intend to extend this concept to 3 dimensional space. This extension raises several new challenges due to an increase in the degrees of freedom of robot movement. There appears to be a need for a novel interaction method to enable robots to form a 3 dimensional shape. Here we use the geometric tetrahedron model, as it is the simplest shape in 3 dimensional space, and accordingly will reduce the computational burden of calculating inter-robot interactions. The convergence of the proposed algorithms are shown based on Lyapunov theory, leading to asymptotic stability of the desired configuration from an arbitrary distribution. We also perform extensive simulations to demonstrate that a swarm of robots can establish a regular tetrahedral network in a scalable manner according to a given spatial density.

II. Problem Statement

We consider a swarm of mobile robots denoted as $r_1, \cdots, r_n$. It is assumed that an initial distribution of all robots is arbitrary and their positions are distinct. Each robot autonomously moves in 3 dimensional space. Robots have no leader and no identifiers. They do not share any common coordinate system, and do not retain any memory of past actions. They can detect the positions of other robots only within their limited sensing range. In addition, each robot does not communicate explicitly with other robots.

Based on these assumptions, let us consider a robot r_i with its *local coordinates* $\vec{r}_{x,i}$, $\vec{r}_{y,i}$, and $\vec{r}_{z,i}$. The *position* of r_i is given by $\mathbf{p}_i = [p_{i,x}\ p_{i,y}\ p_{i,z}]^T$ (for simplicity, p_i hereafter). p_i is $(0, 0, 0)$ with respect to r_i's local coordinates.

G. Lee, Y. Nishimura, K. Tatara, and N.Y. Chong are with the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan {geun-lee,y.nishimura,k_tatara,nakyoung}@jaist.ac.jp

The *distance* between the robot r_i's position p_i and another robot r_j's position p_j is defined as $dist(p_i, p_j)$. The desired *inter-robot distance* is denoted by d_u. Also r_i has a limited *sensing boundary* denoted by *SB*. Then r_i detects the position $\{p_1, p_2, \cdots\}$ of other robots located within its *SB*, yielding a set of the positions, denoted by O_i, with respect to its local coordinates. From O_i, r_i selects three robots r_{n1}, r_{n2}, and r_{n3} that we call the *neighbors*. The set of positions of the neighbors $\{p_{n1}, p_{n2}, p_{n3}\}$ is denoted by N_i. Given p_i and N_i, the *tetrahedral configuration*, denoted by $\mathbb{T}_i^3$, is defined as a set of the four distinct points (*i.e.*, $\{p_i, p_{n1}, p_{n2}, p_{n3}\}$. Given $\mathbb{T}_i^3$, we can express the *distance permutations* with respect to r_i as the following matrix D_i^3:

$$D_i^3 = \begin{cases} (dist(p_m, p_n) - d_u)^2 & \text{if } m \neq n, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $p_m, p_n \in \mathbb{T}_i^3$. We will denote $(dist(p_m, p_n) - d_u)^2$ for simplicity as $(d_k - d_u)^2$. Now we define the *regular tetrahedron configuration*, denoted by $\mathbb{E}_i^3$, as the configuration in which all the distance permutations of $\mathbb{T}_i^3$ are equal to d_u. Using $\mathbb{T}_i^3$ and $\mathbb{E}_i^3$, we formally define the *local interaction* as follows: Given $\mathbb{T}_i^3$, the local interaction allows r_i to maintain d_u with r_i's neighbors at each time (toward forming $\mathbb{E}_i^3$). Based on the local interaction, we address the *self-configuration* problem as follows:

Given a swarm of robots $r_1, \cdots, r_n$ with arbitrarily distinct positions in 3 dimensional space, how to enable the robots to configure themselves into $\mathbb{E}_i^3$.

Now, we propose a self-configuration solution to the above problem, which is composed of two parts: local interaction (ALGORITHM-1) and neighbor selection (ALGORITHM-2). ALGORITHM-1 enables four neighboring robots to generate $\mathbb{E}_i^3$ of side length d_u from an arbitrary $\mathbb{T}_i^3$. Through ALGORITHM-2, r_i determines three neighbors r_{n1}, r_{n2}, and r_{n3}, to be interacted, among O_i. The self-configuration executes for the input O_i with respect to r_i's local coordinates to output r_i's next movement position every time.

III. GEOMETRIC LOCAL INTERACTION

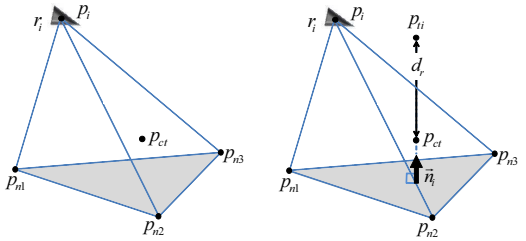
A. Algorithm Description

ALGORITHM-1 LOCAL INTERACTION (code executed by r_i)

FUNCTION $\varphi_{interaction}(\{p_{n1}, p_{n2}, p_{n3}\}, p_i)$

1 centroid p_{ct}: $(p_{ct,x},\ p_{ct,y},\ p_{ct,z})$
2 normal vector $\vec{n}_i$: $[\alpha_i\ \beta_i\ \gamma_i]^T$
3 straight-line equation:
$$\frac{x-p_{ct,x}}{\alpha_i} = \frac{y-p_{ct,y}}{\beta_i} = \frac{z-p_{ct,z}}{\gamma_i} = \mu_i$$
4 parameter μ_i: $\frac{\sqrt{6}}{4\sqrt{\alpha_i^2+\beta_i^2+\gamma_i^2}} d_u$
5 next movement point p_{ti}: $(p_{ti,x},\ p_{ti,y},\ p_{ti,z})$

Let us consider r_i and its three neighbors r_{n1}, r_{n2}, and r_{n3}. As illustrated in Fig. 1-(a), the four robots are configured into $\mathbb{T}_i^3$ whose vertices are p_i, p_{n1}, p_{n2}, and p_{n3}, respectively. First, r_i calculates the centroid of $\mathbb{T}_i^3$, denoted by p_{ct} (=$(p_{ct,x},\ p_{ct,y},\ p_{ct,z})$), with respect to its local coordinates. Then, as shown in Fig. 1-(b), a surface normal vector $\vec{n}_i$ =$[\alpha_i\ \beta_i\ \gamma_i]^T$ to the plane $\triangle p_{n1}p_{n2}p_{n3}$ formed



(a) centroid p_{ct} in $\mathbb{T}_i^3$ (b) next movement point p_{ti}

Fig. 1. Illustration of ALGORITHM-1

by the neighbors is defined, where α_i, β_i, and γ_i denote its direction ratios with respect to r_i's local coordinates. By using the so-called symmetric form, the straight line equation passing through p_{ct} and parallel to $\vec{n}_i$ is defined by

$$\frac{x-p_{ct,x}}{\alpha_i} = \frac{y-p_{ct,y}}{\beta_i} = \frac{z-p_{ct,z}}{\gamma_i} = \mu_i, \quad (2)$$

where $\mu_i \in R$ denotes a parameter increasing or decreasing as r_i travels along the line. Let p_{ti} (= $(p_{ti,x}, p_{ti,y}, p_{ti,z})$) denote the goal of the next movement. Note that the radius d_r of the circumscribed sphere of $\mathbb{E}_i^3$ with side length d_u is $\frac{\sqrt{6}}{4}d_u$. Then, p_{ti} is decided to be located $\frac{\sqrt{6}}{4}d_u$ away from p_{ct} on the straight line of (2). Now, the three parametric line equations passing through p_{ct} and p_{ti} along $\vec{n}_i$ are derived: $p_{ti,x} = \alpha_i\mu_i + p_{ct,x}$, $p_{ti,y} = \beta_i\mu_i + p_{ct,y}$, and $p_{ti,z} = \gamma_i\mu_i + p_{ct,z}$. Accordingly, $dist(p_{ct}, p_{ti})$ is given as: $d_r = \sqrt{(p_{ti,x}-p_{ct,x})^2 + (p_{ti,y}-p_{ct,y})^2 + (p_{ti,z}-p_{ct,z})^2}$. μ_i is then given by

$$\mu_i = \frac{\sqrt{6}}{4\sqrt{\alpha_i^2+\beta_i^2+\gamma_i^2}} d_u. \quad (3)$$

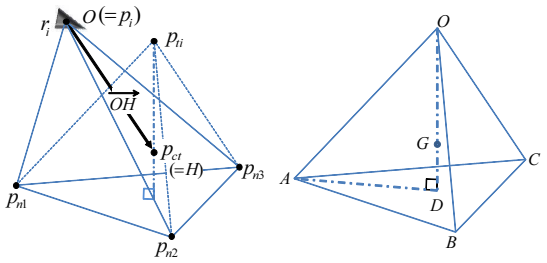
Finally, using (2) and (3), p_{ti} is obtained as follows: $(p_{ct,x} + \frac{\alpha_i\sqrt{6}d_u}{4\sqrt{\alpha_i^2+\beta_i^2+\gamma_i^2}},\ p_{ct,y} + \frac{\beta_i\sqrt{6}d_u}{4\sqrt{\alpha_i^2+\beta_i^2+\gamma_i^2}},\ p_{ct,z} + \frac{\gamma_i\sqrt{6}d_u}{4\sqrt{\alpha_i^2+\beta_i^2+\gamma_i^2}})$.

As illustrated in Fig. 2-(a), at time t, r_i in $\mathbb{T}_i^3(t)$ determines $p_{ti}(t)$ such that the line segment $\overline{p_{ct}p_{ti}}(t)$ is d_r in length and is perpendicular to $\triangle p_{n1}p_{n2}p_{n3}(t)$. In other words, at $t+1$, $\overline{p_{ct}p_{ti}}(t)$ is the circumradius of $\mathbb{T}_i^3(t)$. Similarly, since r_{n1}, r_{n2}, and r_{n3} also execute the same algorithm, it is easily seen that $p_{ct}(t)$ at t is the orthocenter $H(t)$ at $t+1$. By repeatedly running ALGORITHM-1 every time, the four robots eventually configure themselves into $\mathbb{E}_i^3$.

B. Geometric Interpretation

Here we examine the geometric relation between p_{ct} and H. Let us consider a regular tetrahedron $p_ip_{n1}p_{n2}p_{n3}$ whose centroid is G (=p_{ct}) and side length is d_u. In Figs. 2-(b), (c), and (d), p_i, p_{n1}, p_{n2}, and p_{n3} are denoted, for simplicity, as O, A, B, and C, respectively. The point D is the projection of O onto the equilateral triangle $\triangle ABC$. The vectors $\overrightarrow{OA}$, $\overrightarrow{OB}$, $\overrightarrow{OC}$, $\overrightarrow{OD}$, and $\overrightarrow{OG}$ are denoted hereafter as $\vec{a}$, $\vec{b}$, $\vec{c}$, $\vec{d}$, and $\vec{g}$, respectively. Since $\vec{d} \perp \triangle ABC$, the following relations hold:

$$\begin{aligned} \vec{d} \cdot \overrightarrow{AB} &= 0 \\ \vec{d} \cdot \overrightarrow{BC} &= 0 \end{aligned}. \quad (4)$$



(a) position vector $\overrightarrow{OH}$ (b) centroid $G(= p_{ct})$

(c) circumcenter P (d) orthocenter H

Fig. 2. Geometric interpretation of ALGORITHM-1

Now $\vec{d}$, $\overrightarrow{AB}$, and $\overrightarrow{AC}$ can be represented as $\vec{d} = \vec{a} + \overrightarrow{AD}$, $\vec{b} - \vec{a}$, and $\vec{c} - \vec{a}$, respectively. Using the linear independence of $\overrightarrow{AB}$ and $\overrightarrow{AC}$, $\vec{d}$ can be rewritten in the following form:

$$\begin{aligned} \vec{d} &= \vec{a} + x\overrightarrow{AB} + y\overrightarrow{AC} \\ &= \vec{a} + x(\vec{b} - \vec{a}) + y(\vec{c} - \vec{a}) \\ &= (1 - x - y)\vec{a} + x\vec{b} + y\vec{c}, \end{aligned} \tag{5}$$

where x and y are scaling coefficients that can be determined by substituting (5) into (4) given by

$$\begin{aligned} \{(1 - x - y)\vec{a} + x\vec{b} + y\vec{c}\} \cdot (\vec{b} - \vec{a}) &= 0 \\ \{(1 - x - y)\vec{a} + x\vec{b} + y\vec{c}\} \cdot (\vec{c} - \vec{a}) &= 0 \end{aligned}. \tag{6}$$

$\vec{d}$ is finally given by

$$\vec{d} = (1 - \frac{1}{3} - \frac{1}{3})\vec{a} + \frac{1}{3}\vec{b} + \frac{1}{3}\vec{c} = \frac{1}{3}(\vec{a} + \vec{b} + \vec{c}), \tag{7}$$

which shows that D is the centroid of $\triangle ABC$. It is also easily seen that $\vec{g}$ is given by

$$\vec{g} = \frac{3}{4}\{\frac{1}{3}(\vec{a} + \vec{b} + \vec{c})\} = \frac{1}{4}(\vec{a} + \vec{b} + \vec{c})(= \overrightarrow{p_i p_{ct}}). \tag{8}$$

Consequently, $\overrightarrow{GD}$ is perpendicular to $\triangle ABC$.

Next, Fig. 2-(c) illustrates the circumscribed sphere of a regular tetrahedron $OABC$ whose circumcenter is P and radius is d_r in length. Denoting the vector from O to P by $\vec{p}$, we reexpress d_r by

$$d_r = |\vec{p}|^2 = |\vec{a} - \vec{p}|^2 = |\vec{b} - \vec{p}|^2 = |\vec{c} - \vec{p}|^2. \tag{9}$$

Here $|\vec{a} - \vec{p}|^2$ is given by $\langle \vec{a} - \vec{p}, \vec{a} - \vec{p} \rangle = |\vec{a}|^2 - 2\langle \vec{a}, \vec{p} \rangle + |\vec{p}|^2$. From the above relation, $\langle \vec{a}, \vec{p} \rangle$ is equal to $\frac{|\vec{a}|^2}{2}$. Similarly, $\langle \vec{b}, \vec{p} \rangle$ and $\langle \vec{c}, \vec{p} \rangle$ are equal to $\frac{|\vec{b}|^2}{2}$ and $\frac{|\vec{c}|^2}{2}$, respectively. These relations can be represented in matrix form:

$$M^T \vec{p} = \frac{1}{2} \begin{bmatrix} |\vec{a}|^2 \\ |\vec{b}|^2 \\ |\vec{c}|^2 \end{bmatrix}, \tag{10}$$

where M is given by $[\vec{a}\ \vec{b}\ \vec{c}]$. Eq. (10) can be rewritten as

$$\vec{p} = \frac{1}{2}(M^T)^{-1} \begin{bmatrix} |\vec{a}|^2 \\ |\vec{b}|^2 \\ |\vec{c}|^2 \end{bmatrix} = \frac{1}{2} M (M^T M)^{-1} \begin{bmatrix} |\vec{a}|^2 \\ |\vec{b}|^2 \\ |\vec{c}|^2 \end{bmatrix}. \tag{11}$$

As illustrated in Fig. 2-(d), H is the orthocenter of $OABC$. Here we denote the vector from O to H by $\vec{h}$. Based on the orthocenter property, the following relations hold: $\vec{a} \cdot (\vec{h} - \vec{b}) = \vec{b} \cdot (\vec{h} - \vec{c}) = \vec{c} \cdot (\vec{h} - \vec{a}) = 0$, yielding the following relations:

$$\vec{a} \cdot \vec{h} = \vec{b} \cdot \vec{h} = \vec{c} \cdot \vec{h} = k, \tag{12}$$

where k is a nonzero constant. These relations can be represented in matrix form:

$$M^T \vec{h} = \begin{bmatrix} k \\ k \\ k \end{bmatrix}. \tag{13}$$

(13) can be rewritten as:

$$\vec{h} = M G^{-1} \begin{bmatrix} k \\ k \\ k \end{bmatrix}, \tag{14}$$

where G is $M^T M$ (see (11)) given by

$$G = \begin{bmatrix} |\vec{a}|^2 & k & k \\ k & |\vec{b}|^2 & k \\ k & k & |\vec{c}|^2 \end{bmatrix}. \tag{15}$$

Using (15), $[k\ k\ k]^T$ in (14) can be given by

$$\begin{bmatrix} k \\ k \\ k \end{bmatrix} = \frac{1}{2} \left(G \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} |\vec{a}|^2 \\ |\vec{b}|^2 \\ |\vec{c}|^2 \end{bmatrix} \right). \tag{16}$$

Substituting (16) into (14) gives the following equation.

$$\vec{h} = \frac{1}{2} M G^{-1} \left(G \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} |\vec{a}|^2 \\ |\vec{b}|^2 \\ |\vec{c}|^2 \end{bmatrix} \right). \tag{17}$$

Using (11), (17) is reduced to the form given by

$$\vec{h} = \frac{1}{2}(\vec{a} + \vec{b} + \vec{c}) - \vec{p}. \tag{18}$$

From (18), the relation between $\vec{p}$ and $\vec{h}$ is obtained as follows:

$$\vec{h} + \vec{p} = \frac{1}{2}(\vec{a} + \vec{b} + \vec{c}). \tag{19}$$

Since $\vec{p}$ is identical to $\vec{h}$ in $OABC$, (19) is rewritten as

$$\vec{h} = \vec{p} = \frac{1}{4}(\vec{a} + \vec{b} + \vec{c}) = \vec{g}. \tag{20}$$

Based on the facts given above, the following properties are understood. First, $p_{ct}(t)$ can be regarded as $H(t+1)$. Secondly, $\overrightarrow{OH}$ ($\vec{g}$ in (8)) in Fig. 2-(a) can be used to describe the position vector of r_i. Moreover, using (8) and (9), $\overrightarrow{p_{ti}p_i}$ in Fig. 2-(a) is written as

$$\overrightarrow{p_{ti}p_i} = \overrightarrow{OH} - \overrightarrow{p_{ti}p_{ct}} = \vec{g} - \vec{p}. \tag{21}$$

As $\vec{p}$ in (20) overlaps with $\vec{g}$, r_i eventually converges into one vertex of $\mathbb{E}_i^3$. Thirdly, $\sqrt{\alpha_i^2+\beta_i^2+\gamma_i^2}$ in (3) is $|\vec{n}_i|$ with respect to $\triangle p_{n1}p_{n2}p_{n3}$. Since $|\vec{n}_i|$ is given by $|\overrightarrow{p_{n1}p_{n2}} \times \overrightarrow{p_{n1}p_{n3}}|$ in $\triangle p_{n1}p_{n2}p_{n3}$, μ_i can be represented as

$$\mu_i = \frac{\frac{\sqrt{6}}{4}d_u}{|\overrightarrow{p_{n1}p_{n2}}||\overrightarrow{p_{n1}p_{n3}}|\sin(\angle p_{n2}p_{n1}p_{n3})}. \quad (22)$$

When r_i converges into one vertex of $\mathbb{E}_i^3$ with d_u, μ_i converges to $\frac{1}{\sqrt{2}d_u}$, where $\triangle p_{n1}p_{n2}p_{n3}$ in $\mathbb{E}_i^3$ becomes an equilateral triangle.

C. Motion Control

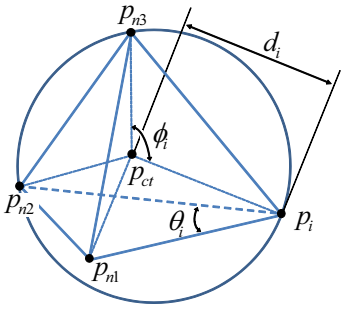


Fig. 3. Motion control rules for individual robots

As shown in Fig. 3, let us consider the circumscribed sphere of a regular tetrahedron $p_ip_{n1}p_{n2}p_{n3}$ whose center is p_{ct} and radius is d_r. From the geometric properties given above, we control the distance d_i from p_i to p_{ct} and the two internal angles θ_i and ϕ_i between $\overrightarrow{p_{n1}p_i}$ and $\overrightarrow{p_{n2}p_i}$, and between $\overrightarrow{p_ip_{ct}}$ and $\overrightarrow{p_{n3}p_{ct}}$, respectively, to generate the motion of r_i.

First, d_i is controlled by the following equation:

$$\dot{d}_i(t) = -a(d_i(t) - d_r), \quad (23)$$

where a is a positive constant. The solution to (23) is given by $d_i(t) = |d_i(0)|e^{-at} + d_r$, which converges exponentially to d_r as t approaches infinity. Secondly, θ_i is represented by

$$\theta_i = \cos^{-1}\frac{\langle\overrightarrow{p_{n1}p_i},\ \overrightarrow{p_{n2}p_i}\rangle}{|\overrightarrow{p_{n1}p_i}||\overrightarrow{p_{n2}p_i}|},$$

where θ_i equals $\frac{\pi}{3}$ in a regular tetrahedron. To create the desired angle θ_r, the following equation is used.

$$\dot{\theta}_i(t) = b(\theta_r - \theta_i(t)), \quad (24)$$

where b is a positive constant. The solution to (24) is given by $\theta_i(t) = |\theta_i(0)|e^{-bt} + \theta_r$, which converges exponentially to θ_r as t approaches infinity. Thirdly, ϕ_i is represented by

$$\phi_i = \cos^{-1}\frac{\langle\overrightarrow{p_ip_{ct}},\ \overrightarrow{p_{n3}p_{ct}}\rangle}{|\overrightarrow{p_ip_{ct}}||\overrightarrow{p_{n3}p_{ct}}|},$$

where $\cos\phi_i$ equals $-\frac{1}{3}$ in a regular tetrahedron. To create the desired angle ϕ_r satisfying $\phi_i = cos^{-1}(-\frac{1}{3})$, the following equation is used.

$$\dot{\phi}_i(t) = c(\phi_r - \phi_i(t)), \quad (25)$$

where c is a positive constant. The solution to (25) is given by $\phi_i(t) = |\phi_i(0)|e^{-ct} + \phi_r$, which converges exponentially to ϕ_r as t approaches infinity. Note that (23), (24), and (25) imply that the trajectory of r_i converges to an equilibrium state $\mathbf{x}_e = [d_r\ \theta_r\ \phi_r]^T$. We use Lyapunov theorem [15] to show the convergence of motion to the state $\mathbf{x} = [d_i(t)\ \theta_i(t)\ \phi_i(t)]^T$.

Let us consider the following scalar function:

$$f_i(d_i,\theta_i,\phi_i) = \frac{1}{2}(d_i - d_r)^2 + \frac{1}{2}(\theta_r - \theta_i)^2 + \frac{1}{2}(\phi_r - \phi_i)^2. \quad (26)$$

This scalar function is always positive definite except when $d_i \neq d_r$, $\theta_i \neq \theta_r$, and $\phi_i \neq \phi_r$. The derivative of the above scalar function is given by

$$\dot{f}_i = -a(d_i - d_a)^2 - b(\theta_r - \theta_i)^2 - c(\phi_r - \phi_i)^2,$$

which is negative definite. The scalar function is radially unbounded, since it tends to infinity as $|\mathbf{x}| \to \infty$. Therefore, $\mathbf{x}_e$ is asymptotically stable, implying that r_i reaches a vertex of the desired regular tetrahedron. Now we show the convergence of the algorithm for n robots. The 4-order scalar function $\mathbf{F}_4$ is defined:

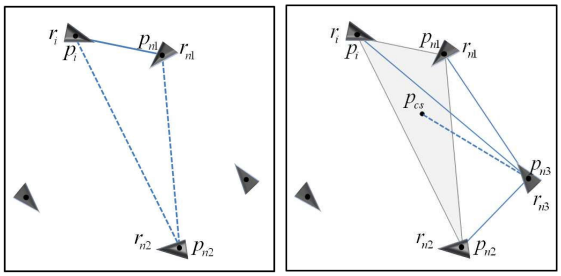
$$\mathbf{F}_4 = \sum_{i=1}^{4} f_i(d_i,\theta_i,\phi_i). \quad (27)$$

It is straightforward to verify that $\mathbf{F}_4$ is positive definite and $\dot{\mathbf{F}}_4$ is negative definite. $\mathbf{F}_4$ is radially unbounded, since it tends to infinity as t approaches infinity. Consequently, 4 robots move toward $\mathbf{x}_e$.

IV. Neighbor Selection Algorithm

ALGORITHM-2 NEIGHBOR SELECTION (code executed by r_i)

Function $\varphi_{configuration}(O_i, p_i)$
1 $p_{n1} := \min_{p\in O_i - \{p_i\}} [dist(p_i, p)]$
2 $p_{n2} := \min_{p\in O_i - \{p_i, p_{n1}\}} [dist(p_{n1}, p) + dist(p, p_i)]$
3 p_{cs} := centroid of $\triangle p_ip_{n1}p_{n2}$
4 $p_{n3} := \min_{p\in O_i - \{p_i, p_{n1}, p_{n2}\}} [dist(p_{cs}, p)]$



(a) selection of r_{n1} and r_{n2} (b) selection of r_{n3}

Fig. 4. Illustration of ALGORITHM-2

In order to form $\mathbb{T}_i^3$, r_i needs to select and interact with three neighbors r_{n1}, r_{n2}, and r_{n3} from O_i. The first neighbor r_{n1}, whose position is denoted by p_{n1}, is selected as the one located the shortest distance away from r_i. Next, as shown in Fig. 4-(a), the second neighbor r_{n2}, whose position is p_{n2}, is determined such that the length of the perimeter of $\triangle p_ip_{n1}p_{n2}$ is minimized. Here, we denote the centroid of $\triangle p_ip_{n1}p_{n2}$ by p_{cs}. Then, as illustrated in Fig. 4-(b), the

third neighbor r_{n3}, whose position is p_{n3}, is chosen such that the distance between p_{cs} and p_{n3} is minimized. The set of neighbor positions N_i is the input to ALGORITHM-1.

V. CONVERGENCE BY THE SELF-CONFIGURATION

The self-configuration process is achieved using ALGORITHM-1 and ALGORITHM-2, yielding a multitude of regular tetrahedrons, denoted by $\sum_{i=1}^{n} \mathbb{E}_i^3$. Specifically, r_i determines and changes its neighbors at each time, whereby it continues to configure itself to be converging toward the equilibrium state $\mathbf{x}_e = [d_r \ \theta_r \ \phi_r]^T$. Without loss of generality, the convergence to $\mathbb{E}_i^3$ can be analyzed based on the concept of the minimal energy level. We use Lyapunov's theory with the scalar function given by

$$f_{sc,i} = \sum_{\mathbb{T}_i^3} (d_k - d_u)^2 + f_i, \tag{28}$$

where f_i is given in (26) and $\sum_{\mathbb{T}_i^3}(d_k - d_u)^2$ is defined as the constant value associated with $\mathbb{T}_i^3$ at each time (see (1)). A symmetric D_i^3 is said to be positive definite, if $\mathbf{x}^T D_i^3 \mathbf{x} > 0$ for every nonzero $\mathbf{x} = [d_i(t) \ \theta_i(t) \ \phi_i(t)]^T$ [16]. Thus, from (26), (28) is always positive definite except when $d_i \neq d_r$, $\theta_i \neq \theta_r$, and $\phi_i \neq \phi_r$. (If $\mathbb{T}_i^3$ is equal to $\mathbb{E}_i^3$, it is easily seen that $\sum_{\mathbb{T}_i^3}(d_k - d_u)^2$ reaches 0.) The derivative of (28) is given by

$$\dot{f}_{sc,i} = \dot{f}_i = -a(d_i - d_a)^2 - b(\theta_r - \theta_i)^2 - c(\phi_r - \phi_i)^2. \tag{29}$$

Eq. (29) is negative definite. Finally, the scalar function $f_{s,i}$ is radially unbounded since it tends to infinity as $|\mathbf{x}| \rightarrow \infty$. Therefore, $\mathbf{x}_e$ is asymptotically stable, implying that r_i reaches a vertex of $\mathbb{E}_i^3$ from an arbitrary $\mathbb{T}_i^3$ by (28) without overlapping each other.

Next, the collective scalar function $\mathbf{F}_{sc}$ of a swarm of robots is a nonzero function with the property that any solution to the set of motion equations is closely related to a set of equilibria for $\{r_i | 1 \leq i \leq n\}$ and vice versa. Without loss of generality, $\mathbf{F}_{sc}$ is a diminishing energy function. Now we prove the convergence of the algorithm for a swarm of n robots. The n-order scalar function $\mathbf{F}_{sc}$ is defined as

$$\mathbf{F}_{sc} = \sum_{i=1}^{n} f_{sc,i} = \sum_{i=1}^{n} \sum_{\mathbb{T}_i^3} (d_k - d_u)^2 + \sum_{i=1}^{n} f_i. \tag{30}$$

From (28), it is straightforward to verify that $\mathbf{F}_{sc}$ is positive definite and $\dot{\mathbf{F}}_{sc}$ is negative definite. $\mathbf{F}_{sc}$ is radially unbounded since it tends to infinity as t approaches infinity. Consequently, a swarm of n robots converges into $\sum_{i=1}^{n} \mathbb{E}_i^3$.

VI. SIMULATION RESULTS

To validate the effectiveness and scalability of the self-configuration scheme, we performed extensive simulations. We assumed that the radius of r_i's SB is 1.25 times d_u. The self-configuration process terminates when the distances between neighbors converge to d_u within a tolerance of 1%, denoted by $d_{1\%}$.

Fig. 5 shows the process of how the four robots initially randomly positioned converge into $\mathbb{E}_i^3$ over time, where the individual positions and trajectories are plotted. Fig. 6 shows the inter-robot distance and angle variations among neighbors according to the activation steps. Here, the distance asymptotically converged to d_u that was set to 10 (unit) in length, and the internal angles of three triangular faces that meet at each vertex converged to 60 degrees. Fig. 7 shows how μ changes according to d_u that was set to 10 and 14.14 (unit), respectively. μ converged to the desired values given by (3) and (22).

Figs. 8, 9 and 10 respectively show that 10, 25, and 50 robots self-configure themselves into a network of regular tetrahedrons. Through these simulations, it could be verified that our proposed algorithms converge within a finite time, and are scalable due to its distributed nature. Individual local geometries interact with each other to reach a global equilibrium state in an autonomous manner. Fig. 11 shows distance variations between r_i and N_i during the self-configuration process in Fig. 8. Regardless of changing their neighbors, a swarm of robots could converge to the desired $\mathbb{E}_i^3$. From a practical point of view, individual robots interact with only three neighbors, which ensures that their motion is minimally affected by each other. To the best of our knowledge, ours is the first solution to the 3 dimensional swarm self-configuration problem. Other possible 3 dimensional self-configuration approaches could employ a larger number of neighbors. Accordingly, the computational load increases.

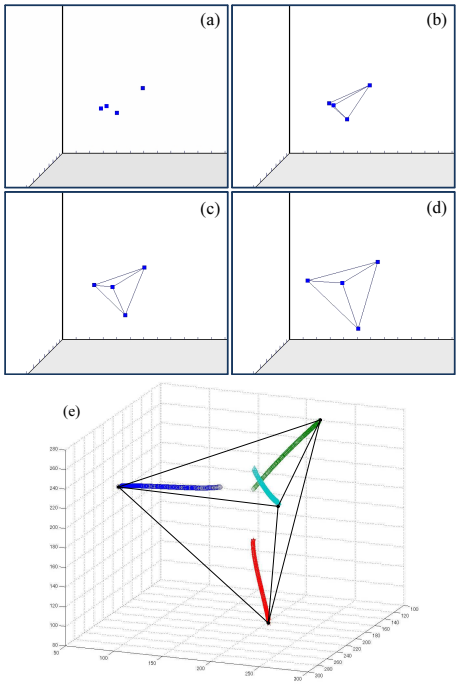
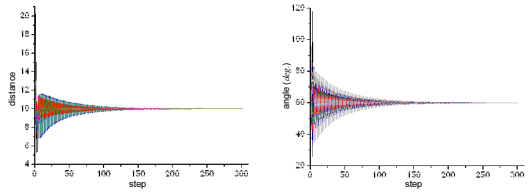


Fig. 5. Simulation result of local interactions among 4 robots ((a) initial distribution, (b) 1 (sec.), (c) 4 (sec.), (d) $d_{1\%}$: 8 (sec.), (e) trajectories)

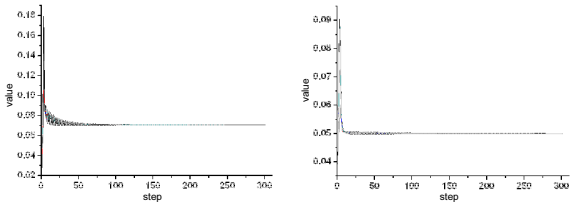
VII. CONCLUSIONS

In this paper, we addressed the 3 dimensional self-configuration problem for a swarm of autonomous mobile robots. We proposed a fully distributed geometric approach whereby individual robots could be configured into a network



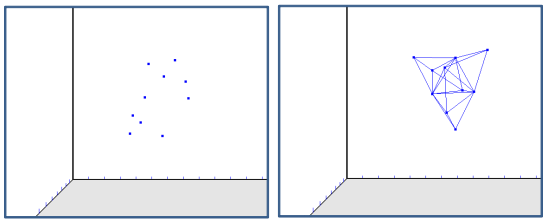
(a) distance variation (b) angle variation

Fig. 6. Distance/angle variations among neighbors during local interactions



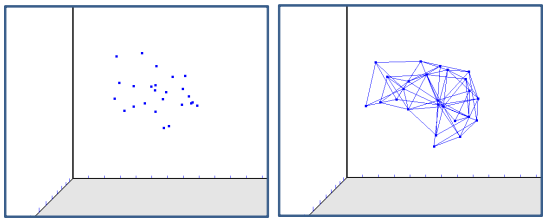
(a) μ with d_u = 10 (b) μ with d_u = 14.14

Fig. 7. μ variation according to d_u during local interactions



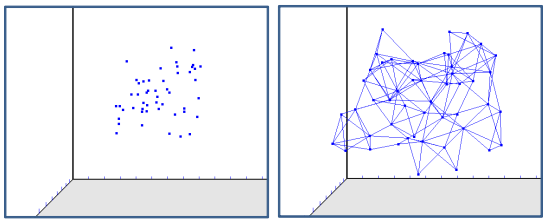
(a) initial distribution (b) $d_{1\%}$: 31 (sec.)

Fig. 8. Simulation result of self-configuration for 10 robots



(a) initial distribution (b) $d_{1\%}$: 83 (sec.)

Fig. 9. Simulation result of self-configuration for 25 robots



(a) initial distribution (b) $d_{1\%}$: 155 (sec.)

Fig. 10. Simulation result of self-configuration for 50 robots

of regular tetrahedrons through local interactions. Specifically, robots were allowed to dynamically select and interact only with three neighbors. It is believed to be a cost-effective solution from a computational point of view. Collecting this local behavior, the swarm as a whole could be deployed in a 3 dimensional space. The convergence of the algorithm was proved theoretically and verified through extensive simulations. Our analysis and simulation results indicated that the proposed self-configuration method can be applied to the formation control of autonomous mobile robots in a distributed and scalable way.

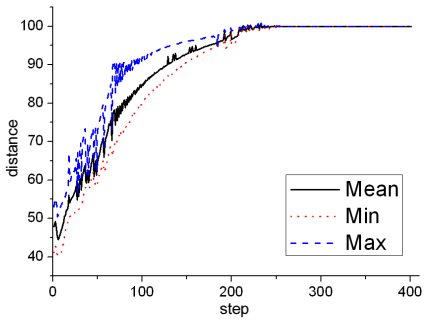


Fig. 11. Distance variations among neighbors during self-configuration in Fig. 8

REFERENCES

[1] B. Shucker, T. D. Murphey, and J. K. Bennett, "Convergence-preserving switching for topology-dependent decentralized systems," IEEE Trans. on Robotics, 24(6):1405-1415, 2008

[2] G. Lee and N. Y. Chong, "A geometric approach to deploying robot swarms," Annals of Mathematics and Artificial Intelligence," 52(2-4):257-280, 2008

[3] Y. Ikemoto, Y. Hasegawa, T. Fukuda, and K. Matsuda, "Graduated spatial pattern formation of robot group," Information Science, 171(4):431-445, 2005

[4] G. Lee and N. Y. Chong, "Adaptive flocking of robot swarms: algorithms and properties," IEICE Trans. on Communications, E91-B(9):2848-2855, 2008

[5] D. Zarzhitsky, D. Spears, and W. Spears, "Distributed robotics approach to chemical plume tracing," Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, 4034-4039, 2005

[6] S. Kamath, E. Meisner, and V. Isler, "Triangulation based multi target tracking with mobile sensor networks," Proc. IEEE Int. Conf. Robotics and Automation, 3283-3288, 2007

[7] B. Werger and M. J. Mataric, "From insect to internet: situated control for networked robot teams," Annals of Mathematics and Artificial Intelligence, 31(1-4):173-198, 2001

[8] J. Reif and H. Wang, "Social potential fields: a distributed behavioral control for autonomous robots," Robotics and Autonomous Systems, 27(3):171-194, 1999

[9] S. Ghosh, K. Basu, and S. K. Das, "An architecture for next-generation radio access networks," IEEE Network, 19(5):35-42, 2005

[10] J. Seyfried, M. Szymanski, N. Bender, R. Estana, M. Thiei, and H. Worn, "The I-SWARM project: intelligent samll world autonomous robots for micro-manipulation," Proc. Int. Conf. the Simulation of Adaptive Behavior, 3342:70-83, 2005

[11] N. Michael, C. Belta, and V. Kumar, "Controlling Three Dimensional Swarms of Robots," Proc. IEEE Int. Conf. Robotics and Automation, 964-969, 2006

[12] R. Olfati-Saber, "Flocking for mult-agnet dynamic systems: algorithms and theory," IEEE Trans. on Automatic Control, 51(3):401-420, 2006

[13] T. Paul, T. R. Krogstad, and J. T. Gravdahl, "UAV formation flight using 3D potential field," Proc. Mediterranean Conf. on Control and Automation, 1240-1245, 2008

[14] L. Chaimowicz and V. Kumar, "Aerial shepherds: coordination among UAVs and swarms of robots," *Distributed Autonomous Robotic Systems 6*, R. Alami, R. Chatila, and H. Asama (eds.), 243-252, Springer, 2007

[15] J. E. Slotine and W. Li, *Applied nonlinear control*, Prentice-Hall, 1991

[16] C.-T. Chen, *Linear system theory and design*, 3rd edn. Oxford University Press, 1999