<table>
<thead>
<tr>
<th>Title</th>
<th>Three Dimensional Deployment of Robot Swarms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Lee, Geunho; Nishimura, Yasuhiro; Tatara, Kazutaka; Chong, Nak Young</td>
</tr>
<tr>
<td>Citation</td>
<td>The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): 5073-5078</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2010-10-18</td>
</tr>
<tr>
<td>Type</td>
<td>Conference Paper</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10119/9553">http://hdl.handle.net/10119/9553</a></td>
</tr>
<tr>
<td>Rights</td>
<td>Copyright (C) 2010 IEEE. Reprinted from The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010, 5073-5078. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of JAIST's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to <a href="mailto:pubs-permissions@ieee.org">pubs-permissions@ieee.org</a>. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th></th>
</tr>
</thead>
</table>
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 \( r_1, \ldots, r_n \). 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 \( p_i = [p_{i,x} \ p_{i,y} \ p_{i,z}]^T \) (for simplicity, \( p_i \) hereafter), \( p_i = (0,0,0) \) with respect to \( r_i \)’s local coordinates.
The distance between the robot $r_i$’s position $p_i$ and another robot $r_j$’s position $p_j$ is defined as $\text{dist}(p_i, p_j)$. The desired inter-robot distance is denoted by $d_u$.

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 $E^3_i$, as the configuration in which all the distance permutations of $T^3_i$ are equal to $d_u$. Using $E^3_i$ and $T^3_i$, we formally define the local interaction as follows: Given $T^3_i$, the local interaction allows $r_i$ to maintain $d_u$ with $r_i$’s neighbors at each time (toward forming $E^3_i$). Based on the local interaction, we address the self-configuration problem as follows:

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

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).

III. GEOMETRIC LOCAL INTERACTION

A. Algorithm Description

**Algorithm-1 Local Interaction**

**FUNCTION** getinteraction($\{p_{n1}, p_{n2}, p_{n3}\}$; $p_i$)

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: $\frac{z-p_{ct,z}}{\gamma_i} = \frac{y-p_{ct,y}}{\beta_i} = \frac{x-p_{ct,x}}{\alpha_i} = \mu_i$
4. parameter $\mu_i$: $4\sqrt{\alpha_i^2 + \beta_i^2 + \gamma_i^2} d_u$
5. next movement point $p_{rt}(p_{ct,x}, p_{ct,y}, p_{ct,z})$

Let us consider $r_i$ and its three neighbors $r_{n1}, r_{n2}, r_{n3}$. As illustrated in Fig. 1-(a), the four robots are configured into $T^3_i$ whose vertices are $p_i, p_{n1}, p_{n2}, p_{n3}$, respectively. First, $r_i$ calculates the centroid of $T^3_i$, 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 by the neighbors is defined, where $\alpha_i$, $\beta_i$, and $\gamma_i$ 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 given by

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

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 $E^3_i$ with side length $d_u$ is $\sqrt{\frac{\pi}{d_u}}$. Then, $p_{ti}$ is decided to be located $\sqrt{\frac{\pi}{d_u}} d_u$ 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: $p_{tx} = \alpha_i \mu_i + p_{ct,x}, p_{ty} = \beta_i \mu_i + p_{ct,y}$, and $p_{tz} = \gamma_i \mu_i + p_{ct,z}$. Accordingly, $\text{dist}(p_{ct}, p_{ti})$ is given as:

$$d_r = \sqrt{(p_{tx} - p_{ct,x})^2 + (p_{ty} - p_{ct,y})^2 + (p_{tz} - p_{ct,z})^2}.$$

Finally, using (2) and (3), $p_{ti}$ is obtained as follows:

$$p_{tx} + \frac{\alpha_i \sqrt{d_r^2}}{4\sqrt{\alpha_i^2 + \beta_i^2 + \gamma_i^2}} d_u, p_{ty} + \frac{\beta_i \sqrt{d_r^2}}{4\sqrt{\alpha_i^2 + \beta_i^2 + \gamma_i^2}} d_u, p_{tz} + \frac{\gamma_i \sqrt{d_r^2}}{4\sqrt{\alpha_i^2 + \beta_i^2 + \gamma_i^2}} d_u.$$

As illustrated in Fig. 2-(a), at time $t$, $r_i$ in $T^3_i(t)$ determines $p_{ct}(t)$ such that the line segment $p_{ct}(t)p_{tri}(t)$ is $d_u$ in length and is perpendicular to $\triangle p_{n1}p_{n2}p_{n3}(t)$. In other words, at $t + 1$, $p_{ct}(t)$ is the circumscribing of $T^3_i(t)$. Similarly, since $r_{n1}, r_{n2}, r_{n3}$ 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 $E^3_i$.

**B. Geometric Interpretation**

Here we examine the geometric relation between $p_{ct}$ and $H$. Let us consider a regular tetrahedron $p_1p_2p_3p_4$ whose centroid is $G$ and side length is $d_u$. In Figs. 2-(b), (c), and (d), $p_1, p_2, p_3, p_4$ are denoted for simplicity, as $O, A, B, C$, respectively. The point $D$ is the projection of $O$ onto the equilateral triangle $\triangle ABC$. The vectors $\vec{OA}$, $\vec{OB}$, $\vec{OC}$, $\vec{OD}$, and $\vec{OG}$ 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:

$$\vec{d} \cdot \vec{AB} = 0, \quad \vec{d} \cdot \vec{BC} = 0.$$
Now \( \overrightarrow{d}, \overrightarrow{AB}, \) and \( \overrightarrow{AC} \) can be represented as \( \overrightarrow{d} = \overrightarrow{a} + \overrightarrow{AD} \), \( \overrightarrow{b} - \overrightarrow{a} \), and \( \overrightarrow{c} - \overrightarrow{a} \), respectively. Using the linear independence of \( \overrightarrow{AB} \) and \( \overrightarrow{AC} \), \( \overrightarrow{d} \) can be rewritten in the following form:

\[
\overrightarrow{d} = \overrightarrow{a} + x\overrightarrow{AB} + y\overrightarrow{AC} = \overrightarrow{a} + x(\overrightarrow{b} - \overrightarrow{a}) + y(\overrightarrow{c} - \overrightarrow{a}) = (1 - x - y)\overrightarrow{a} + x\overrightarrow{b} + y\overrightarrow{c},
\]

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

\[
M = \overrightarrow{c} \cdot \overrightarrow{b} = 0.
\]

\( \overrightarrow{d} \) is finally given by

\[
\overrightarrow{d} = (1 - \frac{1}{3} - \frac{1}{3})\overrightarrow{a} + \frac{1}{3}\overrightarrow{b} + \frac{1}{3}\overrightarrow{c} = \frac{1}{3}(\overrightarrow{a} + \overrightarrow{b} + \overrightarrow{c}),
\]

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

\[
\overrightarrow{g} = \frac{3}{4}\left(\frac{1}{3}(\overrightarrow{a} + \overrightarrow{b} + \overrightarrow{c})\right) = \frac{1}{4}(\overrightarrow{a} + \overrightarrow{b} + \overrightarrow{c}) = \overrightarrow{p} - \overrightarrow{c}_{D}.
\]

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 \( \overrightarrow{p} \), we reexpress \( d_{r} \) by

\[
d_{r} = |\overrightarrow{p}|^2 = |\overrightarrow{a} - \overrightarrow{p}|^2 = |\overrightarrow{b} - \overrightarrow{p}|^2 = |\overrightarrow{c} - \overrightarrow{p}|^2.
\]

Here \( |\overrightarrow{a} - \overrightarrow{p}|^2 \) is given by \( (\overrightarrow{a} - \overrightarrow{p}, \overrightarrow{a} - \overrightarrow{p}) = |\overrightarrow{a}|^2 - 2(\overrightarrow{a}, \overrightarrow{p}) + |\overrightarrow{p}|^2 \).

From the above relation, \( (\overrightarrow{a}, \overrightarrow{p}) \) is equal to \( |\overrightarrow{a}|^2 \). Similarly, \( (\overrightarrow{b}, \overrightarrow{p}) \) and \( (\overrightarrow{c}, \overrightarrow{p}) \) are equal to \( |\overrightarrow{b}|^2 \) and \( |\overrightarrow{c}|^2 \), respectively. These relations can be represented in matrix form:

\[
M^T\overrightarrow{p} = \frac{1}{2} \left[ \begin{array}{c} |\overrightarrow{a}|^2 \\ |\overrightarrow{b}|^2 \\ |\overrightarrow{c}|^2 \end{array} \right],
\]

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

\[
\overrightarrow{p} = \frac{1}{2}(M^T)^{-1} \left[ \begin{array}{c} |\overrightarrow{a}|^2 \\ |\overrightarrow{b}|^2 \\ |\overrightarrow{c}|^2 \end{array} \right] = \frac{1}{2} M(M^T)^{-1} \left[ \begin{array}{c} |\overrightarrow{a}|^2 \\ |\overrightarrow{b}|^2 \\ |\overrightarrow{c}|^2 \end{array} \right].
\]

As illustrated in Fig. 2-(d), \( H \) is the orthocenter of \( OABC \). Here we denote the vector from \( O \) to \( H \) by \( \overrightarrow{h} \).

Based on the orthocenter property, the following relations hold: \( \overrightarrow{a} \cdot (\overrightarrow{h} - \overrightarrow{b}) = \overrightarrow{b} \cdot (\overrightarrow{h} - \overrightarrow{c}) = \overrightarrow{c} \cdot (\overrightarrow{h} - \overrightarrow{a}) = 0 \), yielding the following relations:

\[
\overrightarrow{a} \cdot \overrightarrow{h} = \overrightarrow{b} \cdot \overrightarrow{h} = \overrightarrow{c} \cdot \overrightarrow{h} = k,
\]

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

\[
M^T\overrightarrow{h} = \begin{bmatrix} k \\ k \\ k \end{bmatrix}.
\]

(13) can be rewritten as:

\[
\overrightarrow{h} = MG^{-1} \begin{bmatrix} k \\ k \\ k \end{bmatrix},
\]

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

\[
G = \begin{bmatrix} |\overrightarrow{a}|^2 & k & k \\ k & |\overrightarrow{b}|^2 & k \\ k & k & |\overrightarrow{c}|^2 \end{bmatrix}.
\]

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

\[
G = \frac{1}{2} \left( G \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} |\overrightarrow{a}|^2 \\ |\overrightarrow{b}|^2 \\ |\overrightarrow{c}|^2 \end{bmatrix} \right).
\]

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

\[
\overrightarrow{h} = \frac{1}{2} MG^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} |\overrightarrow{a}|^2 \\ |\overrightarrow{b}|^2 \\ |\overrightarrow{c}|^2 \end{bmatrix}.
\]

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

\[
\overrightarrow{h} = \frac{1}{2}(\overrightarrow{a} + \overrightarrow{b} + \overrightarrow{c}) - \overrightarrow{p}.
\]

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

\[
\overrightarrow{h} + \overrightarrow{p} = \frac{1}{2}(\overrightarrow{a} + \overrightarrow{b} + \overrightarrow{c}.
\]

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

\[
\overrightarrow{h} = \overrightarrow{p} = \frac{1}{4}(\overrightarrow{a} + \overrightarrow{b} + \overrightarrow{c}) = \overrightarrow{g}.
\]

Based on the facts given above, the following properties are understood. First, \( p_{c}(t) \) can be regarded as \( H(t + 1) \).

Secondly, \( \overrightarrow{OH} (\overrightarrow{g} \text{ in } (8)) \) in Fig. 2-(a) can be used to describe the position vector of \( r_{c} \). Moreover, using (8) and (9), \( p_{c}(\overrightarrow{p}) \) in Fig. 2-(a) is written as

\[
\overrightarrow{p}_{c}(\overrightarrow{p}) = \overrightarrow{OH} - \overrightarrow{p}_{c} = \overrightarrow{g} - \overrightarrow{p}.
\]
As $\vec{p}$ in (20) overlaps with $\vec{q}$, $r_i$ eventually converges into one vertex of $E_3^1$. Thirdly, $\sqrt{\alpha_i^2 + \beta_i^2 + \gamma_i^2}$ in (3) is $|\vec{h_i}|$ with respect to $\triangle p_{n1}p_{n2}p_{n3}$. Since $|\vec{h_i}|$ is given by \[ |p_{n1}p_{n2} \times p_{n1}p_{n3}| \] in $\triangle p_{n1}p_{n2}p_{n3}$, $\mu_i$ can be represented as
\[ \mu_i = \frac{\sqrt{d_{ui}}}{|p_{n1}p_{n2}|} \sin(\angle p_{n1}p_{n2}p_{n3}) \]  
(22)

When $r_i$ converges into one vertex of $E_3^1$ with $d_{ui}$, $\mu_i$ converges to $\frac{1}{\sqrt{d_{ui}}}$, where $\triangle p_{n1}p_{n2}p_{n3}$ in $E_3^1$ becomes an equilateral triangle.

C. Motion Control

![Motion control rules for individual robots](image)

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

First, $d_i$ is controlled by the following equation:
\[ \dot{d_i}(t) = -a(d_i(t) - d_r), \]  
(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, $\theta_i$ is represented by
\[ \dot{\theta_i}(t) = b(\theta_i(t) - \theta_r(t)), \]  
(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 $\theta_r$ as $t$ approaches infinity. Thirdly, $\phi_i$ is represented by
\[ \dot{\phi_i}(t) = c(\phi_i(t) - \phi_r(t)), \]  
(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 $\phi_r$ as $t$ approaches infinity. Note that (23), (24), and (25) imply that the trajectory of $r_i$ converges to an equilibrium state $x_c = [d_r, \theta_r, \phi_r]^T$. We use Lyapunov theorem [15] to show the convergence of motion to the state $x_c$.

Let us consider the following scalar function:
\[ f_i(d_i, \theta_i, \phi_i) = \frac{1}{2}(d_i - d_r)^2 + \frac{1}{2}(\theta_i - \theta_r)^2 + \frac{1}{2}(\phi_i - \phi_r)^2. \]  
(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
\[ \hat{f_i} = -a(d_i - d_r)^2 - b(\theta_i - \theta_r)^2 - c(\phi_i - \phi_r)^2, \]  
which is negative definite. The scalar function is radially unbounded, since it tends to infinity as $|x| \to \infty$. Therefore, $x_c$ 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 $F_4$ is defined:
\[ F_4 = \sum_{i=1}^{4} f_i(d_i, \theta_i, \phi_i). \]  
(27)

It is straightforward to verify that $F_4$ is positive definite and $F_4$ is negative definite. $F_4$ is radially unbounded, since it tends to infinity as $t$ approaches infinity. Consequently, $n$ robots move toward $x_c$.

IV. NEIGHBOR SELECTION ALGORITHM

**Algorithm-2: Neighbor Selection (code executed by $r_i$)**

1. $p_{n1} := \min \{ \text{dist}(p_i, p) \}$
2. $p_{n2} := \min \{ \text{dist}(p_{n1}, p) + \text{dist}(p, p_i) \}$
3. $p_{cs} := \text{centroid of } \triangle p_{n1}p_{n2}$
4. $p_{n3} := \min \{ p_i, p_{n1}, p_{n2} \} [\text{dist}(p_{cs}, p)]$

![Illustration of Algorithm-2](image)

In order to form $T_4$, $r_i$ 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 $\triangle p_{n1}p_{n2}p_{n3}$ is minimized. Here, we denote the centroid of $\triangle p_{n1}p_{n2}p_{n3}$ 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 \(\mathbf{x}_c = [d_i \ \theta_i \ \phi_i]^T\). 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

\[
f_{sc,i} = \sum_{i=1}^{3} (d_k - d_u)^2 + f_i, \quad (28)
\]

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

\[
\dot{f}_{sc,i} = \dot{f}_i = -a(d_i - d_u)^2 - b(\theta_i - \theta_a)^2 - c(\phi_i - \phi_a)^2. \quad (29)
\]

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

Next, the collective scalar function \(F_{sc}\) 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, \(F_{sc}\) is a diminishing energy function. Now
we prove the convergence of the algorithm for a swarm of
n robots. The n-order scalar function \(F_{sc}\) is defined as

\[
F_{sc} = \sum_{i=1}^{n} f_{sc,i} = \sum_{i=1}^{n} \sum_{i=1}^{3} (d_k - d_u)^2 + \sum_{i=1}^{n} f_i. \quad (30)
\]

From (28), it is straightforward to verify that \(F_{sc}\) is positive
definite and \(F_{sc}\) is negative definite. \(F_{sc}\) 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 \(r_i’s\) S\(B\) 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

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

5077
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.

REFERENCES