PAPER *IEICE/IEEE Joint Special Section on Autonomous Decentralized Systems Theories and Application Deployments*

Adaptive Flocking of Robot Swarms: Algorithms and Properties

Geunho LEE[†a)], ***Student Member and*** **Nak Young CHONG**[†], ***Nonmember***

SUMMARY This paper presents a distributed approach for adaptive flocking of swarms of mobile robots that enables to navigate autonomously in complex environments populated with obstacles. Based on the observation of the swimming behavior of a school of fish, we propose an integrated algorithm that allows a swarm of robots to navigate in a coordinated manner, split into multiple swarms, or merge with other swarms according to the environment conditions. We prove the convergence of the proposed algorithm using Lyapunov stability theory. We also verify the effectiveness of the algorithm through extensive simulations, where a swarm of robots repeats the process of splitting and merging while passing around multiple stationary and moving obstacles. The simulation results show that the proposed algorithm is scalable, and robust to variations in the sensing capability of individual robots.

key words: *robot swarms, decentralized coordination, local interaction, adaptive flocking*

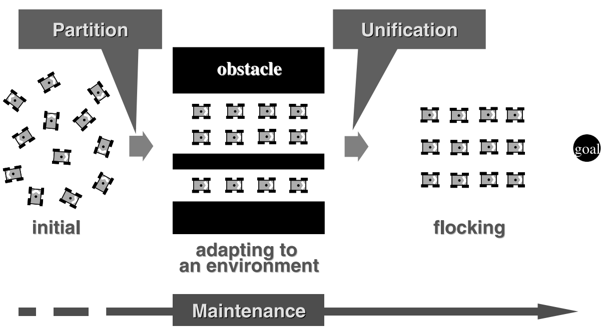


Fig. 1 Concept of adaptive flocking.

1. Introduction

Recently, swarms of mobile robots are expected to be deployed in a wide variety of applications such as exploration, search-and-rescue, medical operations within the human body, and so on [3]. In order to perform those tasks successfully, individual robots need to be controlled to support coordinated swarm behaviors. For the purpose, sociobiology has attracted much attention since living systems exhibit self-organizing and adaptive behavior [1]. This paper is motivated by the observation of schools of fish that exhibit a certain swarm behavior in their environments. For instance, when a school of fish faces obstacles, they avoid collision by splitting themselves into a plurality of smaller swarms, and merge to form a single swarm after they pass around the obstacles. Based on such observation, we propose several swarm behavior rules that enable a swarm of autonomous mobile robots to flock in a complex environment as illustrated in Fig. 1.

Reynolds [2] presented a distributed behavioral model of coordinated animal motion based on fish schools and bird flocks. His work demonstrated that flocking is an example of emergent behavior arising from simple rules. Many flocking strategies reported in the field of swarm robotics can be classified into centralized and decentralized strategies. Centralized strategies [4], [5] employ a central unit that organizes the behaviors of the whole swarm. These strategies usually lack scalability and become technically unfeasible when a large swarm is considered. On the other hand, decentralized strategies are based on interactions between individual robots mostly inspired by evidence from biological systems or natural phenomena. Decentralized strategies can be further divided into biological emergence [6]–[8], behavior-based [9], [10], and virtual physics-based [12]–[14] approaches. Specifically, the behavior-based and virtual physics-based approaches are related to the use of such physical phenomena as crystallization [10], gravitational forces [12], [13], and potential fields [14]. Those works mostly use a force balance between inter-individual interactions exerting an attractive or repulsive force within the influence range, which might over-constrain the swarm and frequently lead to deadlocks. Moreover, the computation of relative velocities or accelerations between robots are needed to obtain the magnitude of the force. Regarding the aspect of calculating the movement position of each robot, accuracy and computational efficiency issues will arise.

In this paper, the swarm behavior emerges in a decentralized way from the local interactions between robots under environmental constraints. In detail, a geometric approach is proposed that enables three neighboring robots to form an equilateral triangle lattice. The plurality of lattices aggregate with each other to self-adjust their shape and size according to the environment condition, eventually forming a single, large swarm. With partially connected mesh topology [18], the proposed method can take advantage of the redundancy provided by a fully connected network topology without the expense and complexity of networking processes. From a practical standpoint, the swarm flocking can be considered as a robust *ad hoc* mobile networking model whose connectivity must be maintained in a cluttered envi-

Manuscript received December 27, 2007.
Manuscript revised April 18, 2008.
[†]The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Nomi-shi, 923-1292 Japan.
a) E-mail: geun-lee@jaist.ac.jp
DOI: 10.1093/ietcom/e91–b.9.2848

ronment.

The rest of this paper is organized as follows. Section 2 presents the robot model and the definition of adaptive flocking problem. Section 3 describes the fundamental motion planning of each robot locally interacting with neighboring robots. Section 4 presents the properties of solutions to the flocking rules. Section 5 provides the results of simulations and discussion. Section 6 draws conclusions.

2. Problem Statement

We consider a swarm of n autonomous mobile robots, where individual robots are denoted respectively by $r_1, \cdots, r_n$. Each robot is modeled as a point, which freely moves on a two-dimensional plane. It is assumed that an initial distribution of robots is arbitrary and distinct. The robots have no leader and no identifiers. They do not share any common coordinate system, and do not retain any memory of past actions that gives inherently self-stabilizing property† [19]. They can detect the positions of other robots within their limited ranges of sensing, but do not have any explicit direct means of communication to each other. Each of the robots executes the same algorithm, but acts independently and asynchronously from other robots. They repeat an endless activation cycle of observation, computation, and motion. At each activation, each robot computes their target position using an algorithm (computation) based on the positions of other robots (observation), and moves toward the computed position (motion) (see Fig. 2). Finally, an obstacle with an even surface is modeled as a polygon-typed object with finite dimensions.

The distance between the robot r_i's position p_i and the robot r_j's position p_j is denoted as $dist(p_i, p_j)$. Denote a constant distance as d_u that is finite and greater than zero. Each robot has a limited sensing boundary denoted by *SB*. Then r_i detects the positions of other robots, denoted by $\{p_1, p_2, \cdots\}$, located within its *SB*, and makes a set of the observed positions O_i obtained with respect to its local coordinate system. From O_i, r_i can select two specific robots r_{s1} and r_{s2}, respectively. We call r_{s1} and r_{s2} the neighbors of r_i and denote their positions $\{p_{s1}, p_{s2}\}$ as N_i. Given p_i and N_i, *Triangular Configuration* is defined as a set of three distinct positions $\{p_i, p_{s1}, p_{s2}\}$ denoted by $\mathbb{T}_i$, where we define the internal angle $\angle p_{s1} p_i p_{s2}$ of r_i as α_i. Next, we can define *Equilateral Configuration* denoted by $\mathbb{E}_i$ if and only if all the possible distance permutations $dist(p_{\pi(i)}, p_{\pi(j)})$ in $\mathbb{T}_i$ are equal to d_u. Now we need a measure indicating to which degree $\mathbb{T}_i$ is configured into $\mathbb{E}_i$. Given $\mathbb{T}_i$, we can express all the possible distance permutations as the following matrix termed *Distance Matrix* $\mathbf{D}_i$ with respect to r_i.

$$\mathbf{D}_i = \begin{cases} (dist(p_m, p_n) - d_u)^2 & \text{if } m \neq n \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\{\{p_m, p_n\} | \, p_m, p_n \in \mathbb{T}_i = \{p_i, p_{s1}, p_{s2}\}\}$. We will denote $(dist(p_m, p_n) - d_u)^2)$ for simplicity as $(d_k - d_u)^2$.

It is known that local geometric shapes of a school of tuna form a diamond shape [15], whereby tunas exhibit the following schooling behaviors: maintenance, partition, and unification. Similarly, local interactions in this work is to form $\mathbb{E}_i$ from an arbitrary $\mathbb{T}_i$. Formally, *Local Interactions* is to have r_i maintain d_u with N_i at each time toward forming $\mathbb{E}_i$. Now, we can define the problem of *Adaptive Flocking* for a swarm of robots based on local interactions as follows:

Given $r_1, \cdots, r_n$ located at arbitrarily distinct positions, how to enable the robots to navigate in a coordinated manner adapting to a given environment.

We advocate that adaptive flocking can be achieved by solving the following three constituent sub-problems.

- **Problem-1**(*Maintenance*): Given that robots located at arbitrarily distinct positions, how to enable the robots to flock in a single swarm.
- **Problem-2**(*Partition*): Given that an environmental constraint is detected, how to enable a swarm to split into multiple smaller swarms adapting to the environment.
- **Problem-3**(*Unification*): Given that multiple swarms exist in close proximity, how to enable them to merge into a single swarm.

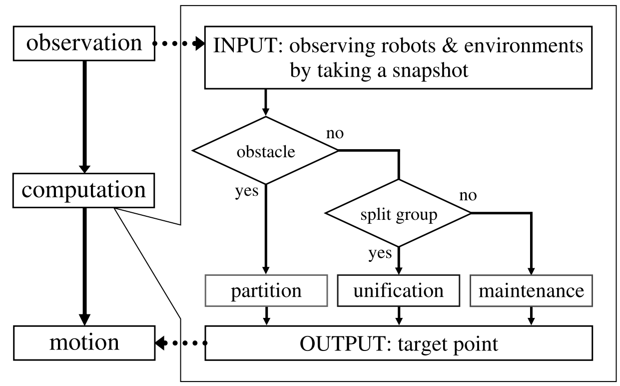


Fig. 2 Adaptive flocking flowchart.

As illustrated in Fig. 2, the input of the solution of adaptive flocking for each time is O_i and the environment constraint with respect to r_i's local coordinate system. The output is the target positions of each robot. When robots detect the constraint within their *SB*, they execute the partition algorithm to adapt their position to the constraint. When they face no constraint, but observe other swarms, they execute the unification algorithm. Otherwise, they basically execute the maintenance algorithm while navigating toward a goal.

In practice, many works on robot swarms use sensor-rich information, memory, and communication means. For

†Self-stabilization is the property of a system which, started in an arbitrary state, always converges toward a desired behavior [20], [21].

example, Nembrini et al. [16] used direct communications, and robots were not required to sense each other's position. Note that if any means of communication are employed, robots need to identify with each other or use a global coordinate or positioning system [16], [17]. In this paper, we attempt to achieve adaptive flocking of robot swarms without taking advantage of rich computational capabilities or communication. This will allow us to develop robot systems in simple, robust, and non-costly ways.

3. Local Interaction

This section explains the local interactions among three neighboring robots. As presented in ALGORITHM-1, the algorithm consists of a function $\varphi_{interaction}$ whose arguments are p_i and N_i at each activation. Consider a robot r_i and its two neighbors r_{s1} and r_{s2} located within r_i's SB. As shown in Fig. 3(a), three robots are configured into $\mathbb{T}_i$ whose vertices are p_i, p_{s1}, and p_{s2}, respectively. First, r_i finds the centroid of the triangle $\triangle p_i p_{s1} p_{s2}$, denoted by p_{ct}, with respect to its local coordinates, and measures the angle ϕ between the line connecting the two neighbors and r_i's horizontal axis. Using p_{ct} and ϕ, r_i calculates the target point p_{ti}. Each robot computes p_{ti} by its current observation of neighboring robots. Intuitively, under ALGORITHM-1, r_i may maintain d_u with its two neighbors at each time. In other words, each robot attempts to form an isosceles triangle for N_i at each time, and by repeatedly doing this, three robots configure into $\mathbb{E}_i$.

As illustrated in Fig. 3(b), let's consider the circumscribed circle of an equilateral triangle whose center is p_{ct} of $\triangle p_i p_{s1} p_{s2}$ and radius d_r is $d_u/\sqrt{3}$. Under the local interactions, the positions of each robot are determined by controlling the distance d_i from p_{ct} and the internal angle α_i (see Fig. 3(a)). First, the distance is controlled by the following equation.

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

where a is a positive constant. 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.

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

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

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

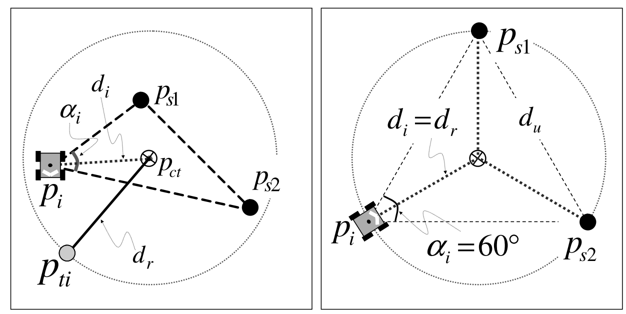
where k' is $3k$. Likewise, the solution of (4) is $\alpha_i(t) = |\alpha_i(0)|e^{-k't} + 60^\circ$ that converges exponentially to 60° as t approaches infinity.

Note that (2) and (4) imply that the trajectory of r_i converges to d_r and 60°, an equilibrium state as termed $[d_r\ 60^\circ]^T$ shown in Fig. 3(b). This also implies that three robots eventually form $\mathbb{E}_i$. In order to prove the convergence of the local interactions, we demonstrate the application of

---

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

```
constant d_u := uniform distance
FUNCTION φ_interaction({p_s1, p_s2}, p_i)
  (p_ct,x, p_ct,y) := centroid(p_s1, p_s2, p_i)
  φ := angle between overline(p_s1 p_s2) and r_i's local horizontal axis
  p_ti,x := p_ct,x + d_u cos(φ + π/2)/√3
  p_ti,y := p_ct,y + d_u sin(φ + π/2)/√3
  p_ti := (p_ti,x, p_ti,y)
```



(a) two control parameters: range and bearing

(b) desired equilateral triangular configuration

Fig. 3 Illustration of two control parameters in local interaction.

Lyapunov stability theory[†] [22]. Now, the desired configuration can be regarded as one that minimizes the energy level of a Lyapunov function.

Consider the following scalar function of the state $\mathbf{x} = [d_i(t)\ \alpha_i(t)]^T$ with continuous first order derivatives:

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

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

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

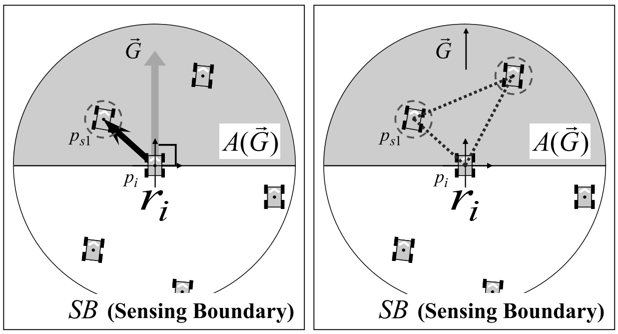
which is obtained by differentiating $f_{l,i}$ to substitute for $\dot{d}_i$ and $\dot{\alpha}_i$. It is evident that (6) is negative definite and the scalar function $f_{l,i}$ is radially unbounded since it tends to infinity as $\| \mathbf{x} \| \rightarrow \infty$. Therefore, the equilibrium state is asymptotically stable, implying that r_i reaches a vertex of $\mathbb{E}_i$.

4. Solution Approaches

4.1 Team Maintenance

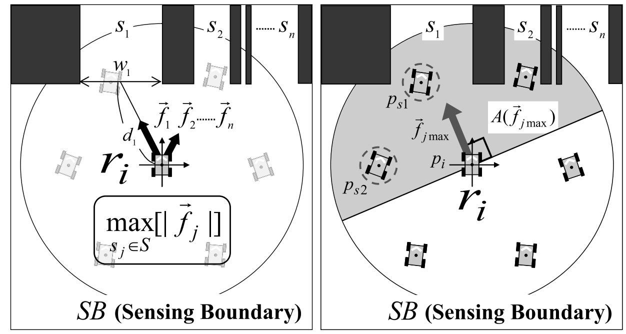
The first problem is how to maintain $\mathbb{E}_i$ with neighboring robots while navigating. A swarm is required to maintain a multitude of equilateral triangle lattices, denoted by $\sum_{i=1}^{n} \mathbb{E}_i$.

[†]Lyapunovs stability theorem states if there exists a scalar function $v(\mathbf{x})$ of the state $\mathbf{x}$ with continuous first order derivatives such that $v(\mathbf{x})$ is positive definite, $\dot{v}(\mathbf{x})$ is negative definite, and $v(\mathbf{x}) \rightarrow \infty$ as $\| \mathbf{x} \| \rightarrow \infty$, then the equilibrium at the origin is asymptotically stable.



(a) 1st neighbor selection (b) 2nd neighbor selection

Fig. 4 Illustration of team maintenance.



(a) favorite vector (b) neighbor selection

Fig. 5 Illustration of team partition.

As illustrated in Fig. 4(a), r_i adjusts $\vec{G}$, termed the goal direction, with respect to r_i's local coordinates and computes O_i at the time t. Here, let $A(\vec{G})$ denote the area of goal direction defined within r_i's SB. Next, r_i checks whether there exists a neighbor in $A(\vec{G})$. If any robots exist within $A(\vec{G})$, r_i selects the first neighbor r_{s1} located the shortest distance away from p_i that gives p_{s1}. Otherwise, r_i spots a virtual point p_v located some distance d_v away from p_i along $\vec{G}$, which gives p_{s1}. As shown in Fig. 4(b), the second neighbor r_{s2} is selected such that the total distance from p_{s1} to p_i passing through p_{s2} is minimized. As a result, p_{ti} can be obtained by $\varphi_{interaction}$ in Algorithm-1.

Under the maintenance algorithm, r_i attempts to find two neighbors within SB at each time and then form $\mathbb{E}_i$. Again, in order to examine the convergence property of the algorithm, we will apply Lyapunov's theory with a scalar function given by

$$f_{m,i} = \sum_{\mathbb{T}_i}(d_k - d_u)^2 + f_{l,i}, \tag{7}$$

where $f_{l,i}$ indicates the scalar function of local interactions in (5), $\sum_{\mathbb{T}_i}(d_k - d_u)^2$ is defined as the constant value $\mathbf{D}_i$ associated with $\mathbb{T}_i$ at each time (see (1)). A symmetric matrix $\mathbf{D}_i$ can be said to be positive definite, if $\mathbf{x}^T\mathbf{D}_i\mathbf{x} > 0$ for every nonzero $\mathbf{x}$ [23]. Moreover the term $f_{l,i}$ is always positive definite except $d_i \neq d_r$ and $\alpha_i \neq 60$. (If $\mathbb{T}_i$ is equal to $\mathbb{E}_i$, it is easily seen that $\sum_{\mathbb{T}_i}(d_k - d_u)^2$ reaches 0, resulted from $d_r = d_u/\sqrt{3}$.) The derivative of the scalar function is given by $\dot{f}_{m,i} = \dot{f}_{l,i}$. From (6), the derivative is negative definite. Therefore, the equilibrium state is asymptotically stable, implying that r_i reaches a vertex of the desired triangle.

Next, the collective scalar function $\mathbf{F}_m$ of a swarm of robots is a nonzero function with the property that any solution of the set of algebraic constraints on range and bearing (see Fig. 3(b)) is closely related to a set of equilibria for $\{r_i|1 \leq i \leq n\}$ and vice versa. Without loss of generality, the collective scalar function is a diminished energy function with a scalar potential. Therefore, the scalar function $\mathbf{F}_m$ for a swarm of n robots is defined as $\mathbf{F}_m = \sum_{i=1}^{n} f_{m,i}$. It is straightforward to verify that $\mathbf{F}_m$ is positive definite and $\dot{\mathbf{F}}_m$ is negative definite. Consequently, a swarm of n robots converges into $\mathbb{E}_i$ for their N_i.

4.2 Team Partition

When a swarm of robots detects an obstacle in its path, each robot is required to determine its direction toward the goal avoiding the obstacle. In this work, each robot determines their direction by using the relative degree of attraction of the passageway [24], termed the favorite vector $\vec{f}$, whose magnitude is given by

$$|\vec{f}_j| = |w_j/d_j^2|. \tag{8}$$

In Fig. 5(a), s_j denotes the passageway with width w_j, and d_j denotes the distance between the center of w_j and p_i. Note that if r_i can not exactly measure w_j beyond its SB, w_j may be shortened. Now the passageways can be represented by a set of favorite vectors $\{\vec{f}_j|1 \leq j \leq n\}$ and then r_i selects the maximum magnitude of $\vec{f}_j$ denoted as $|\vec{f}_j|_{\max}$. As shown in Fig. 5(b), r_i defines a maximum favorite area $A(\vec{f}_{j\max})$ based on the direction of $|\vec{f}_j|_{\max}$ within its SB. Next, r_i checks whether there exists a neighbor in $A(\vec{f}_{j\max})$. If neighbors are found, r_i selects r_{s1} located the shortest distance away from itself to define p_{s1}. Otherwise, r_i spots a virtual point p_v located at d_v in the direction of $|\vec{f}_j|_{\max}$ to define p_{s1}. Finally r_{s2} is selected such that the total distance from p_{s1} to p_i passing through p_{s2} is minimized. As a result, p_{ti} can be obtained by $\varphi_{interaction}$ in Algorithm-1.

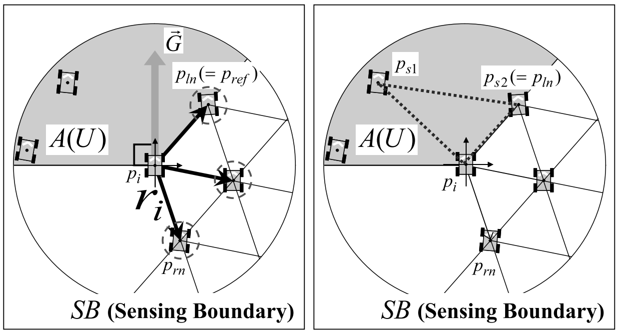
Note that $|\vec{f}_j|_{\max}$ forces r_i move a certain direction of p_{ti}. $|\vec{f}_j|_{\max}$ can be regarded as motion planning given simply by $|\vec{f}_j|_{\max} = \ddot{f}_{l,i}$. Unless r_i collides with any obstacles while locally interacting, then we can prove the convergence into $\mathbb{E}_i$ as detailed below.

Here, Lyapunov's theory is applied again to show the convergence of r_i using the positive definite scalar function $f_{p,i}$ given by

$$f_{p,i} = \frac{1}{2}(\dot{f}_{l,i})^2 + f_{l,i} + \sum_{\mathbb{T}_i}(d_k - d_u)^2. \tag{9}$$

It is evident that the differentiation of $f_{p,i}$ gives

$$\dot{f}_{p,i} = \dot{f}_{l,i}(\ddot{f}_{l,i}) + \dot{f}_{l,i} = \dot{f}_{l,i}(\ddot{f}_{l,i} + 1). \tag{10}$$



(a) unification area (b) neighbor selection

Fig. 6 Illustration of team unification.

which is negative definite. Therefore, based on Lyapunov's theory, the motion planning of r_i under partition converges into $\mathbb{E}_i$ while avoiding any obstacles.

Now we examine the convergence of partition for a swarm of n robots. It is straightforward to define $\mathbf{F}_p = \sum_{i=1}^{n} f_{p,i}$, and show that $\mathbf{F}_p$ is positive definite. It is also evident that $\dot{\mathbf{F}}_p$ is negative definite. Consequently, a swarm of n robots can be split into multiple swarms according to $|\vec{f_j}|_{\max}$ within their SB while avoiding obstacles.

4.3 Team Unification

In order to enable the multiple swarms in close proximity to merge into a single swarm, r_i adjusts $\vec{G}$ with respect to its local coordinates and defines the position set of robots D_u located within the range of d_u. Let $ang(\vec{m}, \vec{n})$ be an angle between two arbitrary vectors $\vec{m}$ and $\vec{n}$. As shown in Fig. 6(a), r_i computes $ang(\vec{G}, \overrightarrow{p_i p_{uk}})$, where $\overrightarrow{p_i p_{uk}}$ is the vectors starting from p_i to p_{uk} of D_u, and defines a neighbor point p_{ref} that gives the minimum $ang(\vec{G}, \overrightarrow{p_i p_{uk}})$ between $\vec{G}$ and $\overrightarrow{p_i p_{uk}}$. Starting from $\overrightarrow{p_i p_{ref}}$, r_i checks whether there exists a neighbor point p_{ul} which belongs to D_u within the area obtained by rotating $\overrightarrow{p_i p_{ref}}$ 60 degrees clockwise. If there exists p_{ul}, r_i finds another neighbor point p_{um} using the same method starting from $\overrightarrow{p_i p_{ul}}$. Unless p_{ul} exists, r_i defines p_{ref} as p_{rn}. Similarly, r_i can decide a specific neighbor point p_{ln} while rotating 60 degrees counterclockwise from $\overrightarrow{p_i p_{ref}}$. The two points, denoted as p_{rn} and p_{ln}, are located at the farthest point in the right-hand or left-hand direction of $\overrightarrow{p_i p_{ref}}$, respectively. As illustrated in Fig. 6(b), a unification area $A(U)$ is defined as the common area between $A(\vec{G})$ in SB and the rest of the area in SB, where no element of D_u exists. Then, r_i defines a set of robots in $A(U)$ and selects the first neighbor r_{s1} located the shortest distance away from p_i in $A(U)$. The second neighbor position is defined such that the total distance from p_{s1} to p_i can be minimized through either p_{rn} or p_{ln}. As a result, p_{ti} can be obtained as using $\varphi_{interaction}$.

The unification algorithm aims to merge multiple smaller swarms into a single, large swarm. In other words, the solution enables the robots located on the boundaries of the swarms to increase the number of neighbor robots which form $\mathbb{E}_i$ centering r_i within SB. Therefore, r_i attempts to reach the maximum number of desired configurations given by $\mathtt{max}[\sum_{k=1}^{s}(\mathbb{E}_i)_k]$, where s will not exceed a maximum of 6 since the configuration is a hexagon composed of 6 equilateral triangle lattices. Depending on the current location of each robot, s varies from 1 to 6.

We define the scalar function related to unification with respect to r_i as $f_{u,i}$ given by

$$f_{u,i} = \sum_{k=1}^{w}(f_{l,i})_k + f_{l,i} + \sum_{\mathbb{T}_i}(d_k - d_u)^2 \tag{11}$$

where w is less than $\mathtt{max}$[s]. Using (11), $\mathtt{max}[\sum_{k=1}^{s}(\mathbb{E}_i)_k]$ can be re-written as follows:

$$f_{u,i} = \mathtt{min}[\sum_{k=1}^{s}(f_{l,i})_k] \tag{12}$$

where s is greater than or equal to $w + 1$ and is less than or equal to 6. Note that, when r_i approaches another swarm, (11) implies that $f_{u,i}$ forces r_i to minimize $\sum_{k=1}^{w}(f_{l,i})_k$. If $f_{u,i}$ decreases, we can predict that r_i becomes stable, namely, $\mathtt{min}[\sum_{k=1}^{s}(f_{l,i})_k]$. By doing this repeatedly, the smaller split groups will be merged.

In order to examine the convergence property of r_i, Lyapunov's theory is applied with positive definite scalar function (11). Differentiating $f_{u,i}$ gives

$$\dot{f}_{u,i} = \dot{f}_{l,i} + \sum_{k=1}^{w}(\dot{f}_{l,i})_j, \tag{13}$$

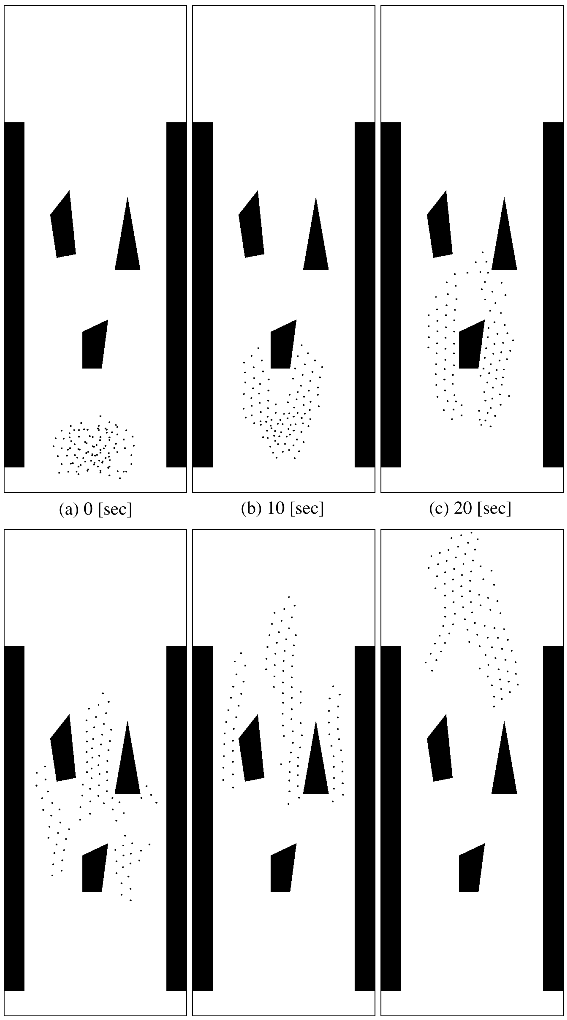
which can be simplified to $\dot{f}_{u,i} = \sum_{k=1}^{s}(\dot{f}_{l,i})_k$. It is evident that $\dot{f}_{u,i}$ is negative definite. Therefore, based on Lyapunov's theory, the motion of r_i under unification converges into $\sum_{k=1}^{s}(\mathbb{E}_i)_k$.

Now we examine the convergence of unification for a swarm of n robots. Using (11), if the scalar function is defined as $\mathbf{F}_u = \sum_{i=1}^{n} f_{u,i}$, it is straightforward to verify that $\mathbf{F}_u$ is positive definite and $\dot{\mathbf{F}}_u$ is negative definite. Consequently, a swarm of n robots converges into $\sum_{i=1}^{n}\left(\mathtt{max}[\sum_{k=1}^{s}(\mathbb{E}_i)_k]\right)$.

5. Simulation Results

We set the distance d_v between p_v and p_i to 1.2 times longer than d_u and the range of SB to 3.5 times longer than d_u. A stationary goal is assumed as a light source and located at a long distance. Moreover, we assume that each robot can detect the goal direction through light emitted from the source. Our simulations start from the scenario that a swarm of robots navigates toward the goal while adapting to an unknown environment like an exploration application.

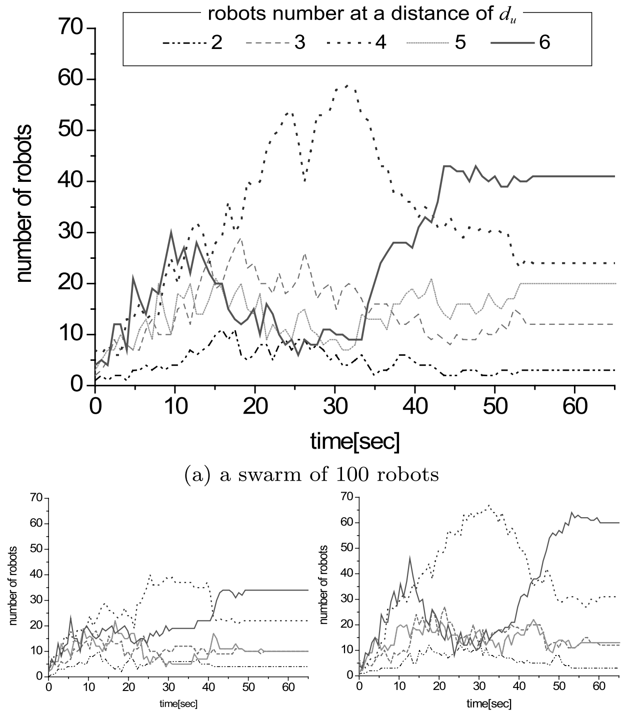
The first simulation demonstrates how a swarm of robots adaptively flocks in an environment populated with obstacles. In Fig. 7, the swarm navigates toward the goal located at the upper center point. On the way to the goal, some of the robots detect an obstacle that forces the swarm split into two groups in Fig. 7(b). The rest of the robots just



(a) 0 [sec] (b) 10 [sec] (c) 20 [sec]

(d) 30 [sec] (e) 40 [sec] (f) 50 [sec]

Fig. 7 Simulation results of adaptive flocking under static environmental conditions.



(a) a swarm of 100 robots

(b) a swarm of 80 robots (c) a swarm of 120 robots

Fig. 8 Changes in the number of neighboring robots located at a constant distance of d_u from each robot in Fig. 7.

follow their neighbors moving ahead. After being split into two swarms, each swarm maintains the geometric configuration while navigating in Fig. 7(c). Note that the robots that could not identify the obstacle follow the moving direction of proceeding robots. Figs. 7(d) and (e) show that two swarms merge and/or split again into smaller swarms due to other obstacles. In Fig. 7(f), the robots successfully pass through the obstacles.

We now investigate properties of the proposed algorithm by changing the number of robots in a swarm to 80, 100, and 120, respectively. Figure 8 presents the changes in the number of neighboring robots at a uniform distance of d_u from each robot while traveling toward a goal in Fig. 7. We can largely divide their behavior into the following four time periods in Fig. 8(a). First, during the first 10 sec., each robot generated an equilateral triangle of a side length of d_u with their neighbors, which resulted in a significant increase of the number of neighbors at a distance of d_u. Secondly, from 10 sec. to 40 sec., the number of robots accompanied by 6 neighboring robots decreased, while the number of robots accompanied by 4 neighboring robots increased. If we take a close look at Fig. 7(d), during this period, the swarm was split into multiple smaller groups due to the obstacles in its path. Thirdly, from 40 sec. to 50 sec., the multiple groups were re-united and the number of neighboring robots located at d_u gradually increased. Lastly, after the unification period, there were no changes in the number of neighboring robots at d_u, maintaining a single swarm through the local interactions of individual robots.

Moreover, Fig. 8 demonstrates the stability and the network connection of a swarm for the proposed algorithm when the participated different numbers of robots are applied to it over the same environmental condition. First, regardless of the environmental constraints, each robot attempts to form $\mathbb{E}_i$ for two selected neighbors at a distance of d_u at each time. In other words, it means stability of the motion planning by locally interacting with the neighbors of each robot while traveling. Secondly, the simulation results present the network connection of the swarm representing the changes of the number of neighboring robots located at a uniform distance d_u while adapting to an environment. Similarly, the proposed algorithm is evaluated in a changing environment as presented in Fig. 9 under the same conditions as the previous static environment. Due to one continuously moving obstacle, two swarms traded robots with each other.

We now examine the effect of the changes in each robot's *SB* on swarm behaviors. We set *SB* to 2 times, 3.5 times, 6 times, and 100 times d_u, respectively, in Fig. 10. Each robot is heading toward a goal located on the right side

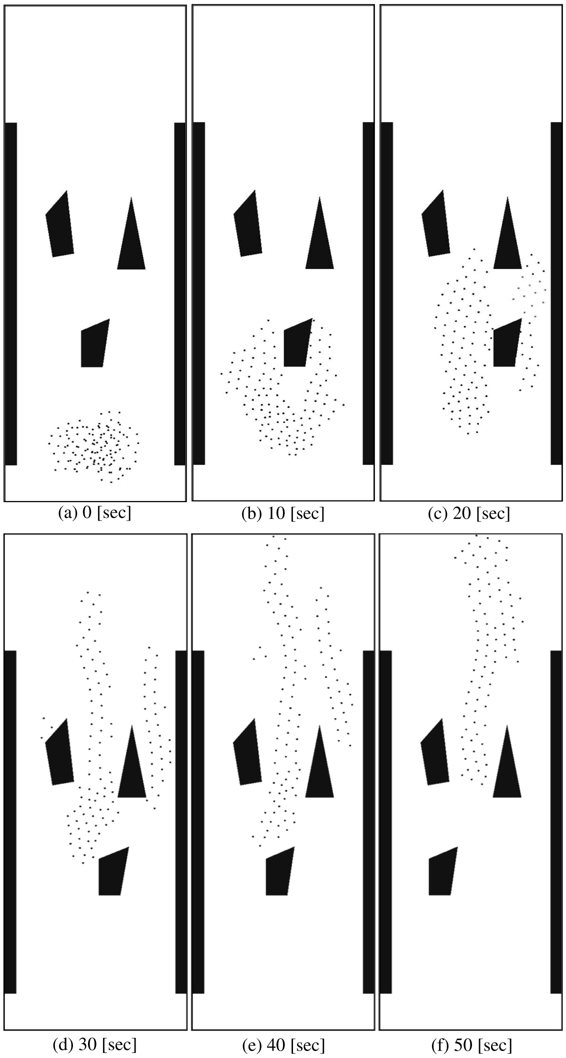


Fig. 9 Simulation results of adaptive flocking under dynamic environmental conditions.

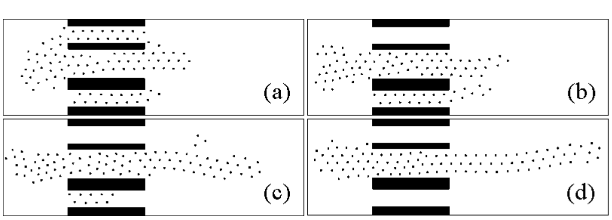


Fig. 10 Simulation results of adaptive flocking according to changes in sensing boundary (a) 2 times d_u, (b) 3.5 times d_u, (c) 6 times d_u, (d) 100 times d_u.

of the figure. Figure 10(a) shows that only the leading edge of the swarm detected the obstacles and selected their path. Other robots were split into three swarms by interacting with the leading edge robots. In contrast, Fig. 10(d) shows that a single swarm was maintained, since all the robots were able to observe the obstacles from a long distance and select their favorite passageway $|\vec{f_j}|_{\max}$ with the largest width. Figure 11 shows the variations in the number of neighbor robots at a

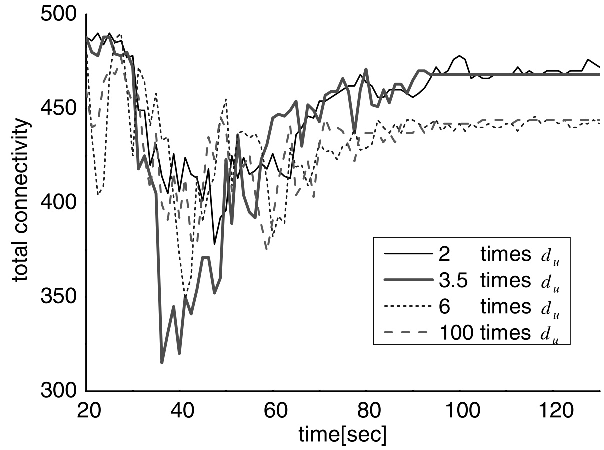


Fig. 11 Variations in the number of neighboring robots located at a constant distance of d_u in Fig. 10.

constant distance of d_u in Fig. 10 according to *SB*. The larger *SB* has each robot, the less fluctuation occurs in the number of neighbor robots at d_u.

6. Conclusion

In this paper, we presented a decentralized algorithm of adaptive flocking, enabling a swarm of autonomous mobile robots to navigate toward achieving a mission while adapting to a complex environment. Through local interactions by observing the positions of neighboring robots, each robot could maintain a uniform distance to their neighbors, and adapt the direction of heading and geometric shape. The algorithm was proved to be convergent using Lyapunov stability theory. Furthermore, we verified the effectiveness of the proposed strategy using our in-house simulator. The simulation results clearly demonstrated that the proposed algorithm is a simple yet robust approach to autonomous navigation of robot swarms in a changing, cluttered environment. In practice, because our robot model is very weak, this algorithm is easily implementable on a wide variety of resource-constrained mobile robots and platforms.

References

[1] E.O. Wilson, Sociobiology: The new synthesis, Harvard University Press, 1976.

[2] C.W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," Comput. Graph., vol.21, no.4, pp.25–34, 1987.

[3] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," Proc. 8th International Conference on the Simulation of Adaptive Behavior (LNCS), vol.3342, pp.10–20, 2005.

[4] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," IEEE Trans. Robot. Autom., vol.20, no.3, pp.120–145, 2005.

[5] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," IEEE Trans. Robot. Autom., vol.17, no.6, pp.947–951, 2001.

[6] F. Mondada, G.C. Pettinaro, A. Guignard, I. Kwee, D. Floreano, J.-L. Deneubourg, S. Nolfi, L.M. Gambardella, and M. Dorigo, "Swarm-bot: A new distributed robotic concept," Autonomous Robots, vol.17, no.2-3, pp.193–221, 2004.

[7] G. Folino and G. Spezzano, "An adaptive flocking algorithm for spatial clustering," Proc. 7th International Conference on Parallel Problem Solving from Nature (LNCS), vol.2439, pp.924–933, 2002.
[8] M. Shimizu, T. Mori, and A. Ishiguro, "A development of a modular robot that enables adaptive reconfiguration," Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.174–179, 2006.
[9] P. Ogren and N.E. Leonard, "A convergent dynamic window approach to obstacle avoidance," IEEE Trans. Robot. Autom., vol.21, no.2, pp.188–195, 2005.
[10] T. Balch and M. Hybinette, "Social potentials for scalable multirobot formations," Proc. IEEE International Conference on Robotics and Automation, pp.73–80, 2000.
[11] 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.
[12] D. Spears, W. Kerr, and W. Spears, "Physics-based robot swarms for coverage problems," International Journal on Intelligent Control and Systems, vol.11, no.3, pp.124–140, 2006.
[13] D. Zarzhitsky, D. Spears, and W. Spears, "Distributed robotics approach to chemical plume tracing," Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1–6, 2005.
[14] J.M. Esposito and T.W. Dunbar, "Maintaining wireless connectivity constraints for swarms in the presence of obstacles," Proc. IEEE International Conference on Robotics and Automation, pp.946–951, 2006.
[15] S. Stocker, "Models for tuna school formation," Mathematical Biosciences, vol.156, pp.167–190, 1999.
[16] J. Nembrini, A. Winfield, and C. Melhuish, "Minimalist coherent swarming of wireless networked autonomous mobile robots," Proc. 7th International Conference on Simulation of Adaptive Behavior, pp.373–382, 2002.
[17] M. Lam and Y. Liu, "ISOGIRD: An efficient algorithm for coverage enhancement in mobile sensor networks," Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1458–1463, 2006.
[18] S. Ghosh, K. Basu, and S.K. Das, "An architecture for next-generation radio access networks," IEEE Netw., vol.19, no.5, pp.35–42, 2005.
[19] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robot: Formation of geometric patterns," SIAM J. Comput., vol.28, no.4, pp.1347–1363, 1999.
[20] S. Dolev, Self-Stabilization, MIT Press, 2000.
[21] M. Schneider, "Self-stabilization," ACM Comput. Surv., vol.25, no.1, pp.45–67, 1993.
[22] J.E. Slotine and W. Li, Applied nonlinear control, Prentice-Hall, 1991.
[23] C.-T. Chen, Linear system theory and design, 3rd ed., Oxford University Press, 1999.
[24] D. Halliday, R. Resnick, and J. Walker, Fundamentals of physics, 5th ed., Wiley, 1997.

Geunho Lee received his Ph.D. degree in information science from Japan Advanced Institute of Science and Technology (JAIST) in 2008. He worked as a researcher at Microsystems Research Center in Korea Institute of Science and Technology (KIST) from 2002 to 2004. Currently, he has been a postdoctoral researcher in JAIST. Research interests include decentralized controls for robot swarms and rehabilitation robotic systems.

Nak Young Chong received his B.S., M.S., and Ph.D. in Mechanical Engineering from Hanyang University, Seoul, Korea in 1987, 1989, and 1994, respectively. From 1994–1998, he was a senior researcher at Daewoo Heavy Industries Ltd. During the period, he was visiting researcher at the Mechanical Engineering Laboratory in Tsukuba, Japan from 1995–1996. After Daewoo, he spent 1 year at the Korea Institute of Science and Technology. From 1998–2003, he was on the research staff of the National Institute of Advanced Industrial Science and Technology in Tsukuba, Japan. In 2003, he joined the faculty of the Japan Advanced Institute of Science and Technology as Associate Professor of Information Science. Dr. Chong served as a co-chair of the IEEE RAS Technical Committee on Networked Robots from 2004–2006. He also served as a co-chair of the Fujitsu scientific systems Robotics working group from 2004–2006, and Robot Information Processing working group from 2006–2008, respectively. He visited Northwestern University in 2001. He is currently on research leave and visiting professor at Georgia Tech. He is the Korea Robotics Society director of international cooperation, and a member of IEEE, RSJ, and SICE.