Artif Life Robotics (2016) 21:405–410
DOI 10.1007/s10015-016-0331-4

SPECIAL FEATURE: ORIGINAL ARTICLE

Adaptive role assignment for self-organized flocking of a real robotic swarm

Yasuda Toshiyuki[1] · Shigehito Nakatani[1] · Akitoshi Adachi[1] · Masaki Kadota[1] · Kazuhiro Ohkura[1]

Received: 11 May 2016 / Accepted: 18 July 2016 / Published online: 31 October 2016
© ISAROB 2016

Abstract Self-organized flocking of robotic swarms has been investigated for approximately 20 years. Most studies are based on a computer animation model named Boid. This model reproduces flocking motion by three simple behavioral rules: collision avoidance, velocity matching, and flock centering. However, flocking performance depends on how these rules are configured and no guideline for the configuration exists. This paper investigates real robot flocking where individuals can switch their roles depending on the situations. Robots can move as leaders or followers, and the roles are dynamically allocated using stochastic learning automata. The flocking performance is evaluated, and swarming behavior is analyzed in a scenario where robots consecutively travel between two landmarks.

Keywords Swarm Robotics · Flocking · Boid · Leader–follower

1 Introduction

Self-organized coordinated motion is commonly observed in animal societies. Swarming behaviors increase the survival rate and reduce the energy consumption of individual animals [1]. Similarly, the navigation and sensing performance of artificial systems can be improved by employing a swarm of robots.

This decade has witnessed growing interest in the research field of the collective behavior of large robot groups, known as Swarm Robotics (SR) [2]. Swarm robotic systems (SRSs) consist of many simple autonomous robots that operate without global controllers. SRSs are expected to have several advantages against a single high-performance robot, such as robustness, flexibility, and scalability. Collective behavior in robotic swarms emerges from the self-organization processes as in biological swarms.

SRSs are usually assigned biological swarming tasks such as aggregation [3], foraging [4], and collective transport [5]. In most studies on flock-like motions of SRSs (see for example, [6, 7]), robots are designed based on the well-known computer animation model Boid [8]. Reynolds showed that flocking in the Boid model can be developed using three simple rules: collision avoidance, velocity matching, and flock centering. However, flocking performance depends on how these rules are configured and no guideline for the configuration exists.

This paper thus investigates swarm robot flocking by introducing multiple rule configurations that can be switched depending on situations. The remainder of this paper is organized as follows. Our task allocation approach for flocking behavior in mobile robots is described in Sect. 2. Section 3 shows experimental results. Conclusions and future research directions are presented in Sect. 4.

2 Method

Boid-based flocking has been successfully demonstrated in various computer simulations, but cannot be easily implemented in real robots. The main reason is that accurate

This work was presented in part at the 1st International Symposium on Swarm Behavior and Bio-Inspired Robotics, Kyoto, Japan, October 28–30, 2015.

✉ Yasuda Toshiyuki
yasu@hiroshima-u.ac.jp

[1] Graduate School of Engineering, Hiroshima University, 1-4-1, Kagamiyama, Higashi-Hiroshima 739-8527, Japan

neighbor information, especially on the orientations, is difficult to obtain in modern technological systems for mechanical and electrical reasons. To overcome this difficulty, mobile robots are frequently programmed with communication functions that enable flocking behaviors [6, 7]. Explicit communication, however, presents an additional challenge because it requires external computation or communication devices.

Another difficulty in implementing Boid to real robots is that no guideline on how to coordinate collision avoidance, velocity matching, and flock centering does not exist. Trial-and-error based weighting for the rules is typically conducted.

In this paper, the first problem is simply mitigated by providing robots information on their destination. The heading direction of robots tends to be aligned toward the destination. For the second problem, multiple weighting parameter sets are provided, and a learning mechanism for which parameter set to use is implemented.

Figure 1 shows the range of each behavioral rule based on Boid model. The repulsion and attraction rules enable the robot to avoid collisions and remain close to its neighbors, respectively. The repulsion works for robots within the range of r_{rep} from the center of the corresponding robot. The attraction rule is applied if neighboring robots are in the range of r_{rep} to r_{att}. The acceleration rule is a biomimetic rule inspired by surf scoter behavior [9]. This rule gives robots a preference for neighbors in the frontal sector with the central angle of θ_f. The heading direction vector is calculated as

$$\mathbf{h} = a\mathbf{R} + b\mathbf{A} + c\mathbf{L}, \quad (1)$$

where **R**, **A**, and **L** denote the repulsion vector, attraction vector, and leader vector (landmark-heading vector), respectively. The leader vector is expected to be utilized as an alignment rule in the original Boid. Constants a, b, and c are set within the range [0.0, 1.0].

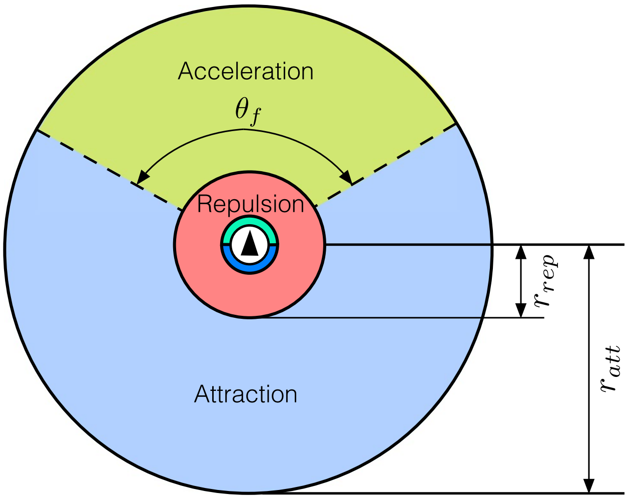


Fig. 1 Rule areas

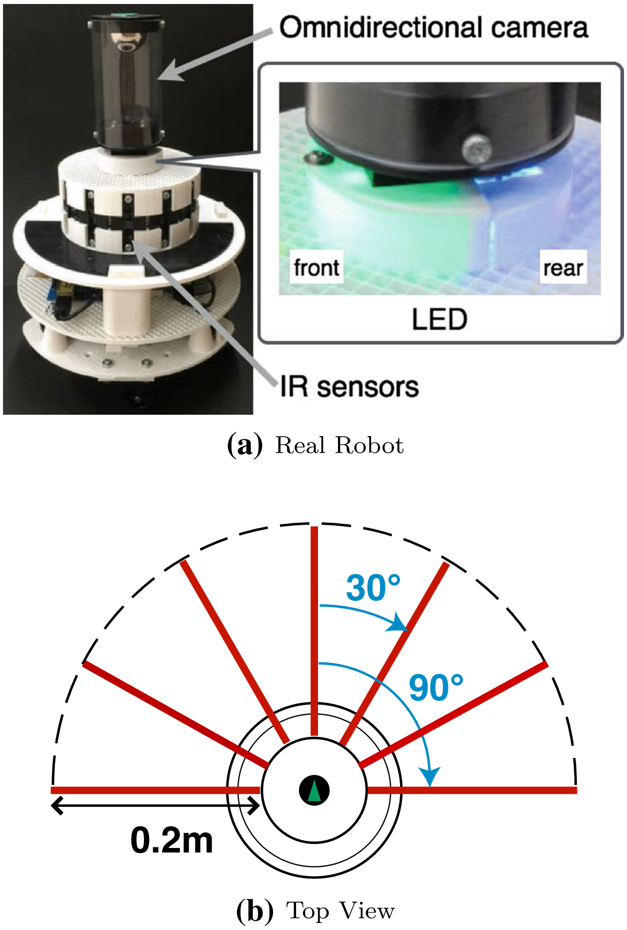


Fig. 2 Two-wheeled robot with IR sensors and an omni-vision camera

In this paper, robots have a mechanism for switching their roles depending on the situations. Robots can move as leaders or followers, and the roles are dynamically allocated using stochastic learning automata (SLA). Leaders have a preference for moving toward their destination ($b < c$), and followers for attracting to neighbors ahead ($b > c$). Whether outputs of SLA are correct or not is evaluated based on the distribution of neighboring robots. Leaders/followers prefer neighbors behind/ahead rather than those ahead/behind, respectively.

3 Experiments

3.1 Mobile robot

Figure 2 shows the two-wheeled mobile robot used in this paper. The robot is 28 cm tall and 18 cm in diameter. It is equipped with seven distance sensors and an omnidirectional camera. Objects within the sensor range, such as neighboring robots and walls, can be detected. If an object is in the sensing range of a distance sensor, robot i tries to avoid collisions by generating the repulsion force as follows:

$$\mathbf{h}_i = \sum_k d_k e^{i\phi_k}, \tag{2}$$

where d_k is the value of the distance sensor k attached at $\phi_k = \frac{\pi}{6}k$ ($k \in \{0, 1, ..., 6\}$). When neighbors are in the acceleration area, robot i moves toward them by generating an attractive vector described in the next section.

3.2 Heading direction vector

When n_{rep} robots are in the repulsion area, the repulsion vector $\mathbf{r}$ is generated from information obtained by the omnidirectional camera:

$$\mathbf{R} = -\frac{1}{n_{rep}} \sum g(d_{ij})\mathbf{u}_{ij}, \tag{3}$$

$$d_{ij} = \left|\mathbf{x}_j - \mathbf{x}_i\right|, \tag{4}$$

$$\mathbf{u}_{ij} = \frac{(\mathbf{x}_j - \mathbf{x}_i)}{d_{ij}}, \tag{5}$$

where $\mathbf{x}_i$ denotes the position vector of the ith robot, and $g(\cdot)$ is a normalizing function. The attraction vector for n_{att} robots in the attraction area is calculated as:

$$\mathbf{A} = \frac{1}{n_{\mathrm{att}}} \sum g(d_{ij})\mathbf{u}_{ij}. \tag{6}$$

Besides, the leader vector is determined as

$$\mathbf{L} = \mathbf{h}_{\mathrm{l}} - \mathbf{h}_{\mathrm{c}}, \tag{7}$$

where $\mathbf{h}_{\mathrm{l}}$ and $\mathbf{h}_{\mathrm{c}}$ are the direction for a target landmark and the current heading direction of a robot.

3.3 Robot behavior

The vectors defined in the previous subsection are transformed into signals that actuate the two side-wheels of the mobile robot. The rotation velocities of the left and right wheels, denoted N_{L} and N_{R}, respectively, are calculated as

$$u = \begin{cases} (\mathbf{h} \cdot \mathbf{h}_{\mathrm{c}})\, u_{\max} & \text{if } \mathbf{h} \cdot \mathbf{h}_{\mathrm{c}} \geq 0.5, \\ u_{\min} & \text{otherwise} \end{cases} \tag{8}$$

$$\omega = (\angle \mathbf{h}_c - \angle \mathbf{h})K_p, \tag{9}$$

$$N_{\mathrm{R}} = \left(u - \frac{\omega}{2} l\right)\frac{60}{2\pi r}, \tag{10}$$

$$N_{\mathrm{L}} = \left(u + \frac{\omega}{2} l\right)\frac{60}{2\pi r}, \tag{11}$$

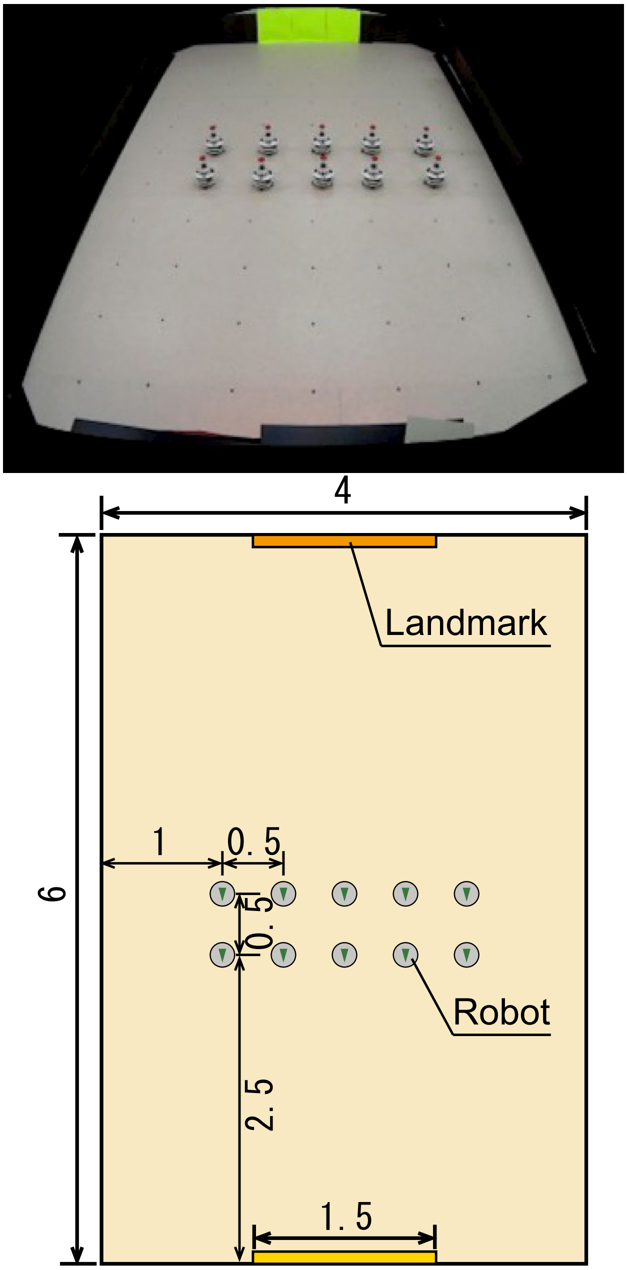


Fig. 3 Environment

where $u_{\max}$ and $u_{\min}$, respectively, denote the maximum and minimum speed of the robot, $K_p = 0.5$ is a proportional gain, l is the distance between the left and right wheels, and r is the wheel radius.

3.4 Setups

The above flocking approach was evaluated on real robots placed in a 6 m × 4 m experimental field surrounded by walls. The initial locations of the 10 deployed robots are shown in Fig. 3. The task for the robots is making round

Table 1 The parameters

Parameter	Value
Weight of the repulsion vector for leaders a_l	0.3
Weight of the attraction vector for leaders b_l	0.3
Weight of the landmark vector for leaders c_l	1.0
Weight of the repulsion vector for followers a_f	0.3
Weight of the attraction vector for followers b_f	1.0
Weight of the landmark vector for followers c_f	0.3
Repulsion range r_{rep}	0.2 m
Attraction range r_{att}	1.0 m
Acceleration area angle θ_f	120°
Learning coefficient α	0.02
Learning coefficient β	0.02

trips between two landmarks located on the both of the shorter walls as many as possible. At the beginning of each experiment, all the robots are oriented in the same direction and start moving toward the landmark ahead. After reaching the target landmark, robots turn around and move toward the other landmark. By repeating this movement, the robots can make round trips.

Four types of robotic swarms are evaluated: (i) all leaders ($b < c$), (ii) all followers ($b > c$), (iii) random leader–follower allocation at each time step, and (iv) SLA-based leader–follower switching. Five runs, each of 300-s duration, are executed for each swarm. The fixed parameters are listed in Table 1. SLA-based leader–follower switching in the robotic swarms (iv) is conducted every 10 time steps, approximately 5 s, during the experimental runs online. When a robot selects a leader rule at time t, the probabilities for selecting a leader/follower rule, P_l and P_f, respectively, are updated as follows:

$$P_l(t+1) = \begin{cases} P_l(t) + \alpha(1 - P_l(t)) & \text{if } N_f \geq N_b \\ (1-\beta)P_l(t) & \text{otherwise} \end{cases} \quad (12)$$

$$P_f(t+1) = \begin{cases} (1-\alpha)P_f(t) & \text{if } N_f \geq N_b \\ \beta + (1-\beta)P_f(t) & \text{otherwise} \end{cases} \quad (13)$$

where α and β are learning coefficients. N_f and N_b denote the numbers of neighbors in front and back at $t + 1$, respectively. The probability update after selecting a follower rule is executed as in Eqs. (12) and (13).

3.5 Results

Flocking behavior was evaluated by two criteria: the number of round trips and largest aggregate. The connectivity of robots i and j is defined if both robots are in the sensing range [10]. The cohesiveness of the flock is ratio of the largest aggregate, ϕ, to the whole flock. We evaluated the swarm behavior via movies recorded by a camera on the ceiling of our experimental room.

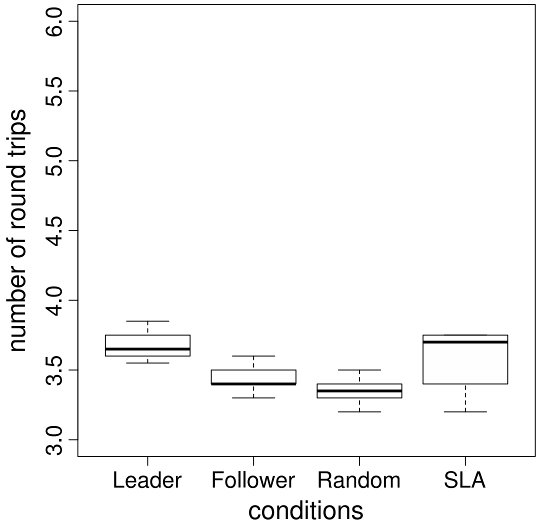


Fig. 4 Number of round trips

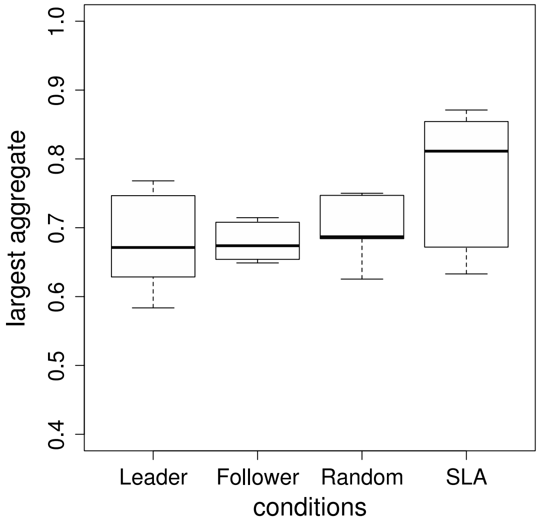
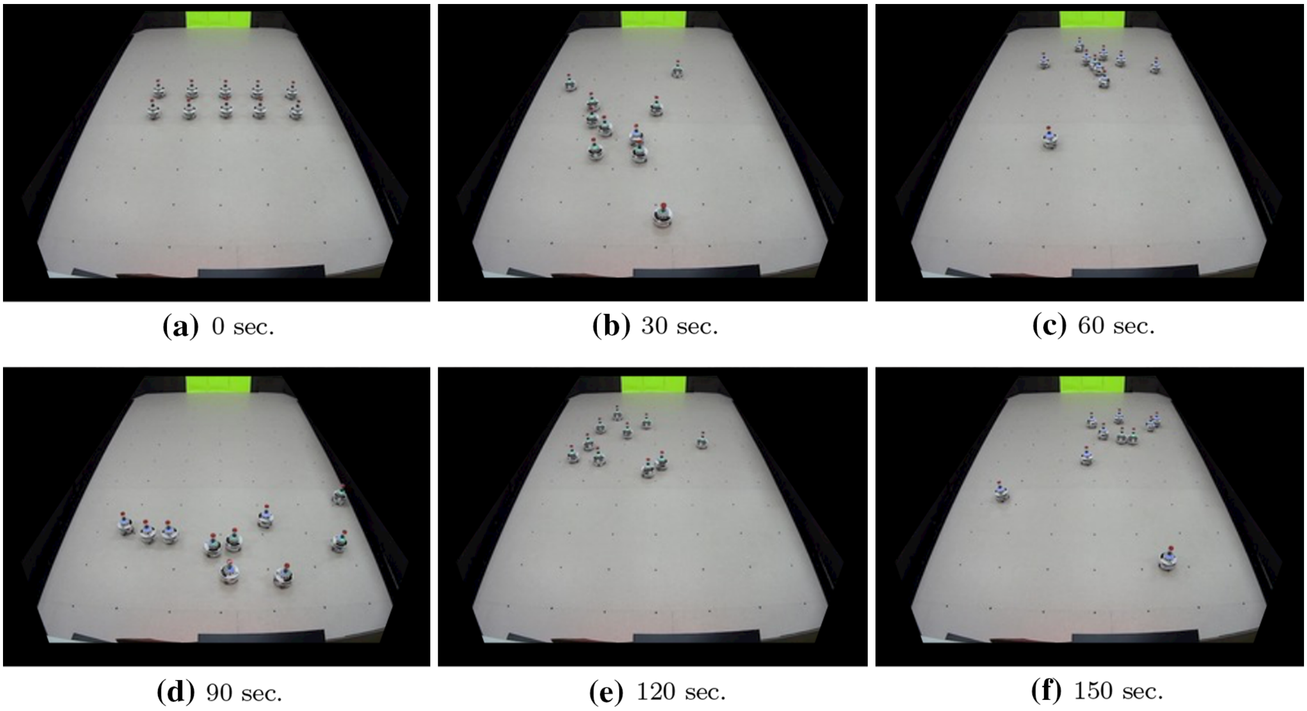


Fig. 5 Largest aggregate

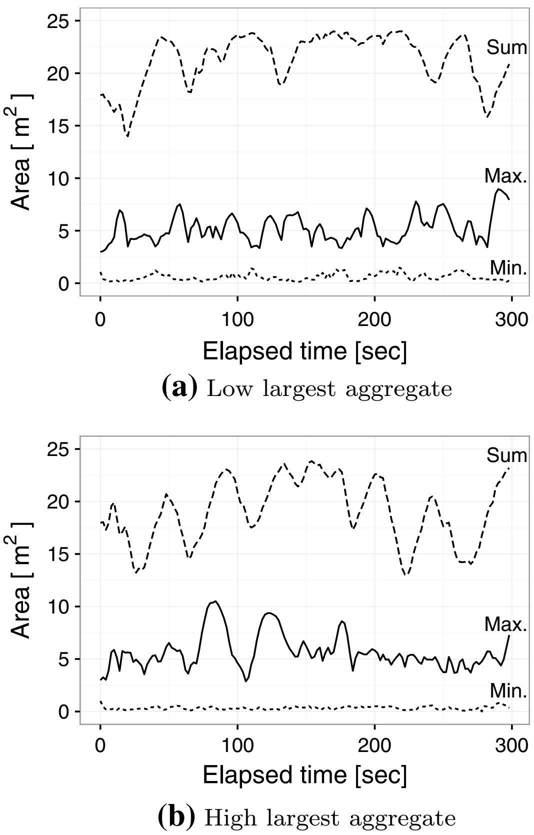
Figures 4 and 5 present the number of successful round trips and the largest aggregate metrics. A swarm with all leaders made the biggest number of round trips. Judging from the largest aggregate metric, however, robots behave not cooperatively but individually. The largest aggregate performance of the robotic swarms is improved when a learning mechanism is implemented. It can be said that dynamic task allocation in a robotic swarm can improve the coherence of flocking.

Figure 6 shows snapshots of a flocking mobile robotic swarm in a trial. It can be observed that the flock formation


(a) 0 sec. **(b)** 30 sec. **(c)** 60 sec.

(d) 90 sec. **(e)** 120 sec. **(f)** 150 sec.

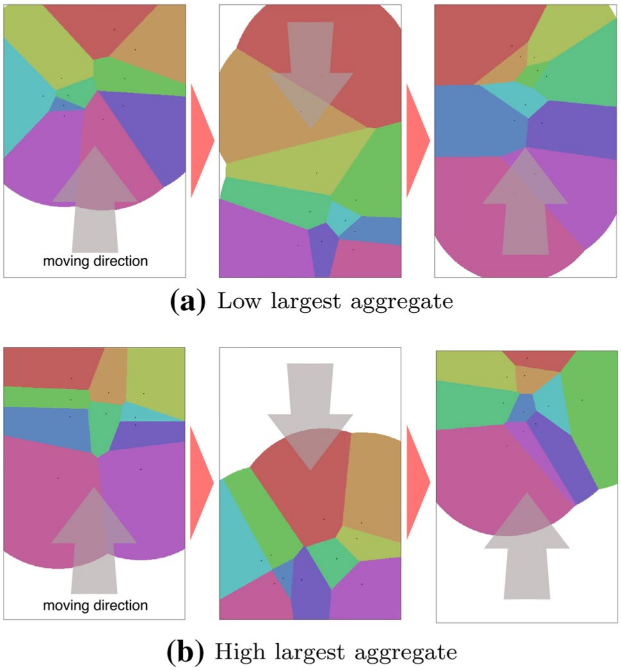
Fig. 6 Snapshots of flocking behavior



(a) Low largest aggregate

(b) High largest aggregate

Fig. 7 Transition of coverage areas



(a) Low largest aggregate

(b) High largest aggregate

Fig. 8 Flocking behavior of a robotic swarm and its coverage areas as Voronoi diagrams (100–200 s)

dynamically changes during round trips. Figure 7 shows the transitions of the coverage areas represented as Voronoi diagrams, a partitioning of the experimental filed into regions based on distance to robots, in experimental runs where largest aggregates ϕ are low/high. The closest pair of robots corresponds to two adjacent cells in the diagram. Since robots can detect neighbors in their sensing range, r_{att}, cell sizes are also restricted. The more coherently the robots perform, the smaller the coverage areas are. Several peaks in the sum of the coverage areas are observed in each experiment. It can be said from this observation that robots repeat the cycle of aggregation and dispersion throughout the experimental runs. When a robotic swarm displays good flocking behavior, peaks tend to be steep. Figure 8 depicts the position and the sensing range of each robot in the middle of experimental runs. As robots separate and turn around individually, the coverage area expands.

4 Conclusions

Self-organized coordinated motion in swarming robots can improve the navigation and sensing performance of the robotic system. This paper investigated a flocking model of SRS that can switch leading/following behavior according to the situations. In this task allocation model, a stochastic learning automaton is used to determine which behavior to be selected. The aggregate and order performances of the model were evaluated in real robot experiments.

As future scope, we plan on introducing more robots into the physical experiments. We are also interested in whether our approach is extendible to flying robots. Flocking behavior of robots that tune weights during the runtime will be investigated.

References

1. Parrish K, Viscido SV, Grünbaum D (2002) Self-organized fish schools: an examination of emergent properties. Biol Bull 202(3):296–305
2. Şahin E (2005) Swarm robotics : from source of inspiration to domain of applications, swarm robotics, SAB2004 International Workshop, Santana Monica, CA, USA, July 2004. Revised Selected Papers, LNCS 3342:10–20
3. O. Soysal, and E. Şahin (2005) Probabilistic aggregation strategies in swarm robotic systems. In: Proc. of the 2005 IEEE Swarm Intelligence Symposium, pp 325–332
4. Liu W, Winfield AFT, Sa J, Chen J, Dou L (2007) Towards energy optimization: emergent task allocation in a swarm of foraging robots. Adapt Behav 15(3):289–305
5. Groß R, Dorigo M (2009) Towards group transport by swarms of robot. Int J Bio-Inspired Comput 1(1–2):1–13
6. Turgut AE, Çelikkant H, Gökçe F, Şahin E (2008) Self-organized flocking in mobile robot swarms. Swarm Intell 2(2–4):97–120
7. Ferrante E, Turgut AE, Mathews N, Birattari M, Dorigo M (2010) Flocking in stationary and non-stationary environments : a novel communication strategy for heading alignment. Lect Notes Comput Sci 6239:331–340
8. Reynolds C (1987) Flocks, herds and schools: a distributed behavioral model, computer graphics, vol 21, no.4 (Proceedings of ACM SIGGRAPH ’87). ACM Press, pp 25–34
9. Lukeman R, Li YX, Edelstein-Keshet L (2010) Inferring individual rules from collective behavior. Proc Natl Acad Sci 107(28):12576–12580
10. Çelikkanat H, Şahin E (2010) Steering self-organized robot flocks through externally guided individuals. Neural Comput Appl 19(6):849–865