A Reinforcement Learning Scheme of Adaptive Flocking Behavior

Masahito Tomimasu[1], Koichiro Morihiro[2,1], Haruhiko Nishimura[3], Teijiro Isokawa[1], and Nobuyuki Matsui[1]

[1] University of Hyogo, 2167 Shosha, Himeji, 671-2201, JAPAN

E-mail: tomimasu@comp.eng.himeji-tech.ac.jp,{isokawa,matsui}@eng.u-hyogo.ac.jp

[2] Hyogo University of Teacher Education, 942-1, Yashiro, 673-1494, JAPAN

E-mail: mori@info.hyogo-u.ac.jp

[3] University of Hyogo, 1-3-3 Higashikawasaki-cho, Chuo-ku, Kobe, 650-0044, JAPAN

E-mail: haru@ai.u-hyogo.ac.jp

Abstract

Flocking by birds, herding by land animals, or schooling by fishes is well-known collective behavior in nature. Many previous observations suggest that there are no leaders who control the behavior of the group. Several models have been proposed for describing the flocking behavior (we call the aggregate motions only as flocking from now on). In these models, a rule is given to each of individuals a priori for their interactions in reductive and rigid manner. Instead of this, we propose a new framework for self-organized flocking of agents by reinforcement learning. It will become important to introduce a learning scheme for making collective behavior in artificial autonomous distributed systems. The behavior of agents is demonstrated and evaluated through computer simulations and it is shown that the flocking behavior of agents emerges as a result of learning.

1 Introduction

Bird-flocking or fish-schooling is well-known collective behavior in nature. Many previous observations suggest that there are no leaders to control the behavior of the group; rather it emerges from the local interactions among individuals in the group. Several models have been proposed for describing the flocking behavior. In these models, a rule is given to each of individuals a priori for their interactions[1][2][3]. This reductive and rigid approach is plausible for modeling flocks of biological organisms, for they seem to inherit the ability of making a flock. However what is more, it will become important to introduce a learning scheme for making collective behavior. In a design of artificial autonomous distributed system, fixed interactive relationships among agents (individuals) lose the robustness against nonstationary environments. It is necessary for agents to be able to adjust their parameters of the ways of interactions. Some learning framework to form individual interaction will be of importance. In addition to securing the robustness of system, this framework will give a possibility to design systems easier, because it determines the local interactions of agents adaptively as a certain function of the system.

In this paper, we propose an adaptive scheme for self-organized making flock of agents. Each of agents is trained in its perceptual internal space by Q-learning, which is a typical reinforcement learning algorithm[4][5][6]. The behavior of agents is demonstrated and evaluated through computer simulations.

2 Reinforcement Learning

2.1 Q-learning

Machine learning that gives a computer system an ability to learn has been developed and used in various situations. A lot of learning algorithms and methods are proposed for the system to acquire step-by-step the desired function. Reinforcement learning is originated in experimental studies of learning in psychology. Almost all reinforcement learning algorithms are based on estimating value functions. The system gets only an evaluative scalar feedback of a value function from its environment, not an instructive one as in supervised learning. Q-learning is known as the best-understood reinforcement learning algorithm. The value function in Q-learning consists of values decided from a state and an action, which is called Q-value. In Q-learning, proceedings on learning consist of acquiring a state(s_t), deciding an action(a_t), receiving

a reward(r) from an environment, and updating Q-value($Q(s_t, a_t)$). Q-value is updated by the equation written as follows:

$$Q(s_{t+1}, a_{t+1}) = Q(s_t, a_t) + \alpha[r + \gamma \max_{a' \in A(s')} Q(s', a') - Q(s_t, a_t)] \quad (1)$$

where A denotes a set of actions, α is the learning rate($0 < \alpha \leq 1$), γ is the discount rate($0 \leq \gamma \leq 1$).

2.2 Action Choice Generator

In the reinforcement learning, many kinds of exploration policies have been proposed as a process of trial and error such as ϵ-greedy, softmax, and weighted roulette action selection. Here, we adopt softmax action selection, and the rule is given as follows:

$$p(a|s) = \frac{\exp\{Q(s,a)/T\}}{\sum_{a_i \in A} \exp\{Q(s,a_i)/T\}} \quad (2)$$

where T is a positive parameter called the temperature. High temperatures cause the actions to be all (nearly) equi-probable, and low temperatures cause a greater difference in selection probability for actions that differ in their value estimates.

3 Model and Method

In this section, we introduce a scheme of perceptual internal space as the Q-value coordinates in the situation that an agent perceives (finds) another one among the others.

3.1 Perceptual Internal Space for Each Agent

We employ a configuration where N agents that can move to any direction are placed in a two-dimensional field. The agents act in discrete time, and their velocities are 1 body-length(1 BL). At each time-step an agent (agent i) finds other agent (agent j) among N-1 agents. In the perceptual internal space, the state s_t of $Q(s_t, a_t)$ for the agent i is definded as [R], the maximum integer not surpassing the Euclidean distance from agent i to agent j, R. As the action a_t of $Q(s_t, a_t)$ four kinds of action patterns (a_1, a_2, a_3, a_4) are taken as follows, shown in Fig.1.

a_1 : Attraction to agent j

a_2 : Parallel positive orientation to agent j
$(\mathbf{m_a} \cdot (\mathbf{m_i} + \mathbf{m_j}) \geq 0)$

a_3 : Parallel negative orientation to agent j
$(\mathbf{m_a} \cdot (\mathbf{m_i} + \mathbf{m_j}) < 0)$

a_4 : Repulsion to agent j

where $\mathbf{m_a}$ is the directional vector of a_t, $\mathbf{m_i}$ and $\mathbf{m_j}$ are the velocity vectors of agent i and agent j, respectively, with $|\mathbf{m_a}|=|\mathbf{m_i}|=|\mathbf{m_j}|=1$(BL). Agent i moves according to $\mathbf{m_i}$ at each time-step, and $\mathbf{m_i}$ is updated by

$$\mathbf{m_i} \leftarrow \frac{(1-\kappa)\mathbf{m_i} + \kappa \mathbf{m_a}}{|(1-\kappa)\mathbf{m_i} + \kappa \mathbf{m_a}|} \quad (3)$$

where κ is a positive parameter($0 \leq \kappa \leq 1$).

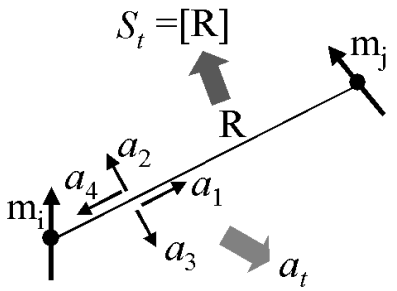


Figure 1: Constitution of perceptual internal space for each agent

3.2 Learning Method for Each Agent

In our proposed model, we prepare the reward for (s_t, a_t) of each agent as shown in Table 1, where R_1,R_2, and R_3 have the relationship of $R_1 < R_2 < R_3$.

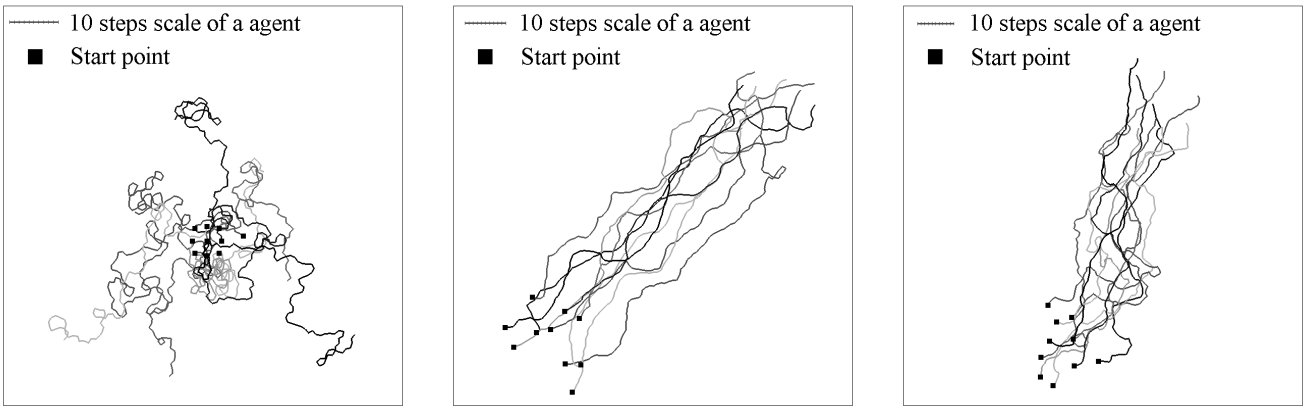
Table 1: Reward r preparation for the selected action a_t in the state s_t=[R]

s_t	$0 <$[R]$\leq R_1$		$R_1 <$[R]$\leq R_2$		$R_2 <$[R]$\leq R_3$	
a_t	a_4	a_1,a_2,a_3	a_2	a_1,a_3,a_4	a_1	a_2,a_3,a_4
r	1	-1	1	-1	1	-1

The learning of the agents proceeds according to a positive or negative reward. In case [R]>R_3, agent i cannot perceive agent j, and then receives no reward and choose an action from the four action patterns(a_1, a_2, a_3, a_4) randomly. In case $0 <$[R]$\leq R_3$, agent i can perceive another agent with the probability in proportion to $R^{-\beta}$, where β is a positive parameter. This means that the smaller R value is, the easier the agent at that position is selected.

4 Simulations and Results

To demonstrate our proposed scheme in computer simulations, we take the following experimental conditions : $\alpha = 0.1$, $\gamma = 0.7$ in Eq.(1), $T = 0.5$(under



(a) Under learning, in the range $0 - 100$ steps

(b) Under learning, in the range $4900 - 5000$ steps

(c) After learning, in the range $500 - 600$ steps

Figure 2: The trajectories of agents under and after learning in the case of $N = 10$, and (R_1,R_2,R_3)=(4,20,50)

learning) in Eq.(2), $\kappa = 0.5$ in Eq.(3), and $\beta = 0.5$ for the distance dependence of $R^{-\beta}$. The total number of trials is set to 5000 time-step through all simulations. Under these conditions we check whether agents make a flock for the parameters N and (R_1,R_2,R_3).

4.1 N=10 with (R_1 ,R_2, R_3) = (4, 20, 50) Case

We simulated our model in the case of the number of agents N=10, and R_1=4(BL), R_2=20(BL) and R_3=50(BL). Figures 2(a),(b), and(c) show the trajectories in the range $0 - 100$ steps, $4900 - 5000$ steps under learning, and in the range $500 - 600$ steps after learning. In Fig.2(a) each of the agents changes its direction very often, but it keeps the direction for long time-step with the others in Figs.2(b) and (c). This indicates that the learning succeeded in flocking. In order to evaluate how the agents make flocking behavior quantitatively, we introduce a measure $|\mathbf{M}|$ of the uniformity in direction :

$$|\mathbf{M}| = \frac{1}{N}\left|\sum_{i=1}^{N} \mathbf{m_i}\right| \tag{4}$$

The value of $|\mathbf{M}|$ becomes closer to 1 when the directions of agents increase their correspondence. Figure 3 shows the time-step dependence of $|\mathbf{M}|$ in this case. The transition of $|\mathbf{M}|$ evolves good except for the fluctuation owing to the exploration effect in every time-step. To remove these large variations we further take the average of 100 events by repeating the above simulation with various random series in exploration. As a result, Fig.4 is obtained in which the value of $|\mathbf{M}|$ increases up to near 0.9.

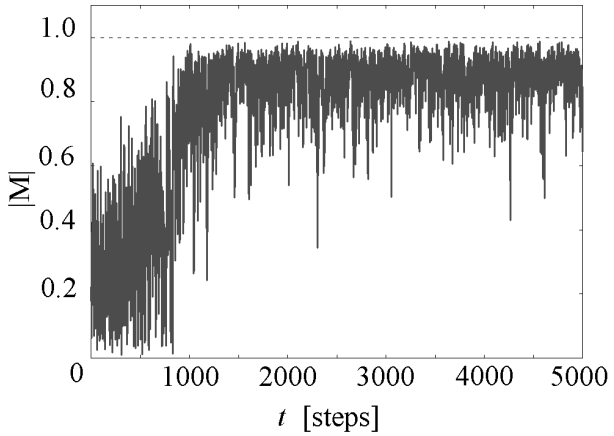


Figure 3: The time-step dependence of $|\mathbf{M}|$ (Eq.(4)) for the case in Fig.2

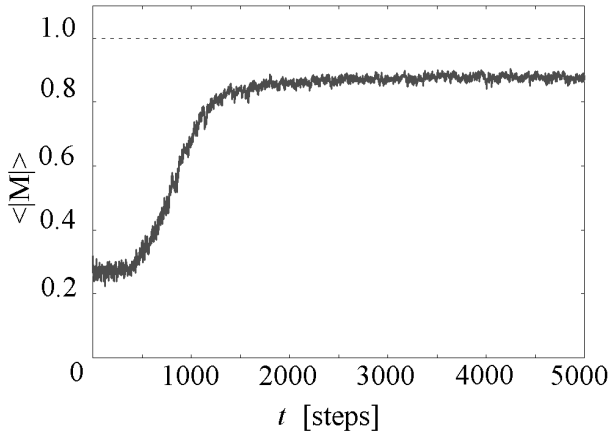
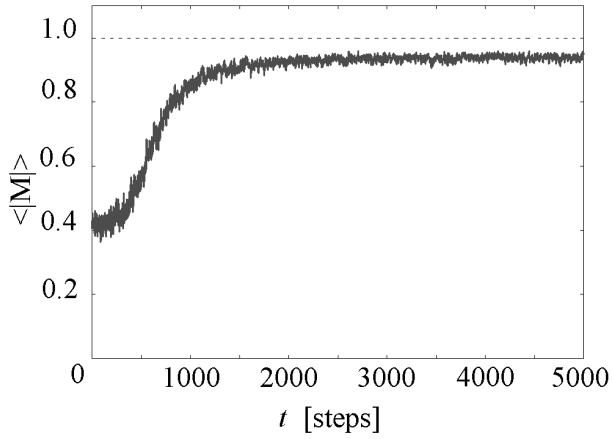
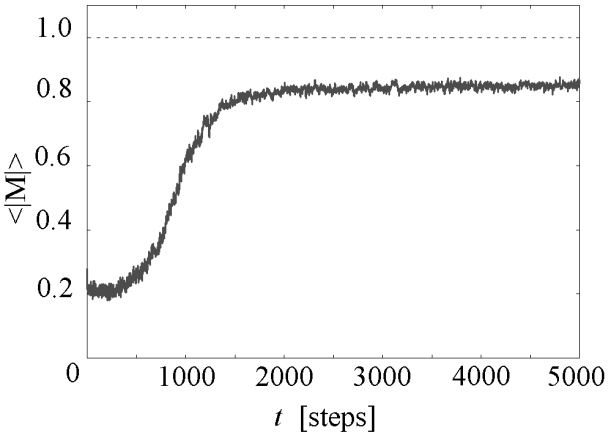


Figure 4: The step-time dependence of the averaged $|\mathbf{M}|$ in 100 events of Fig.3 case



(a) 4 agents case



(b) 16 agents case

Figure 5: The time-step dependence of the averaged $|\mathbf{M}|$ in 100 events in the cases of N=4 and 16

4.2 N=4, 16 with (R_1,R_2,R_3)=(4,20,50) Cases

We simulated our model in the cases of 4 and 16 agents with (R_1,R_2,R_3) = (4, 20, 50). Figures 5(a) and (b) show the transition of averaged $|\mathbf{M}|$ in 100 events respectively. It is found that the performances are good in both cases depending a little on the number of agents.

4.3 N=10 with Individual (R_1,R_2,R_3) Case

The case of 10 agents with (R_1,R_2,R_3) = (3,16,40), (3,17,45), (4,19,47), (4,18,48), (4,20,50), (4,21,51), (4,22,52), (5,25,45), (5,23,53), (6,25,55) was furthermore simulated. From Fig.6 the performance is as well as the case with common (R_1,R_2,R_3) in section 4.1.

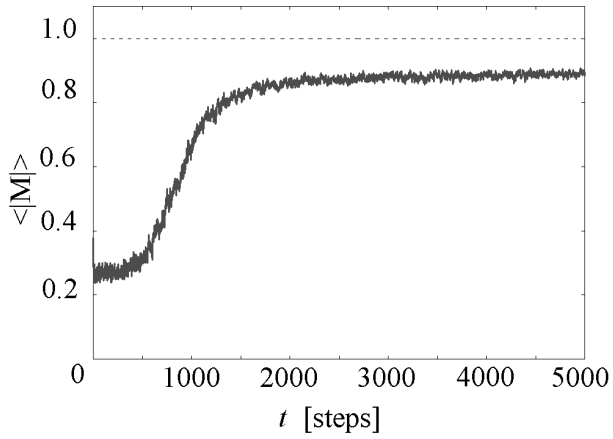


Figure 6: The time-step dependence of the averaged $|\mathbf{M}|$ in 100 events in the case of N=10 with individual (R_1,R_2,R_3)

5 Conclusion

We proposed a scheme for autonomously making flock of agents by reinforcement Q-learning and could show the flocking behavior of agents emerges as a result of learning in simulations. In order to confirm whether our scheme is effective under various environments, we proceed further investigations on various parameters and on introducing different kind of agents such as predators.

References

[1] I. Aoki, "A Simulation Study on the Schooling Mechanism in Fish, " *Bulletin of the Japanese Society of Scientific Fisheries*, Vol.48, No.8, pp.1081-1088, 1982.

[2] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model, " *Comp. Graph*, Vol.21, No.4, pp.25-34, 1987.

[3] A. Huth and C. Wissel, "The Simulation of the Movement of Fish Schools, " *J. theor. Biol*, Vol.156, pp.365-385, 1992.

[4] R. S. Sutton and A. G. Barto, Reinforcement Learning, Cambridge, The MIT press, 1982.

[5] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement Learning : A Survey, " *Journal of Artificial Intelligence Research*, Vol.4, pp.237-285, 1996.

[6] C. J. C. H. Watkins and P. Dayan, "Q-learning, " *Machine Learning*, Vol.8, pp.279-292, 1992.