Engineering Application of Non-Reciprocal-Interaction-Based (NRIB) Model: Swarm Robotic System That Can Perform Spatially Distributed Tasks in Parallel

Takeshi Kano[1], Eiichi Naito[2], Takenobu Aoshima[2], and Akio Ishiguro[1]

[1]Research Institute of Electrical Communication, Tohoku University, Sendai, Miyagi, Japan 980-8577
[2]Business Innovation Division, Panasonic Corporation, Osaka, Japan
Correspondence: tkano@riec.tohoku.ac.jp

Abstract

We propose a simple decentralized control scheme for swarm robots that can perform spatially distributed tasks in parallel, drawing inspiration from the non-reciprocal-interaction-based (NRIB) model we proposed previously. Each agent has an internal state called "workload." Each agent first moves randomly to find a task. When it finds a task, its workload increases, and then it attracts its neighboring agents to ask for their help. Simulation was used to demonstrate the validity of the proposed control scheme.

Recently, we have proposed an extremely simple model of collective behavior based on non-reciprocal interactions by drawing inspiration from friendship formation in human society (Kano et al. (2017a); Kano et al. (2017b)). It was demonstrated via simulations that various patterns emerge by changing the parameters (http://www.riec.tohoku.ac.jp/∼tkano/ECAL_Movie1.mp4). Although this model (hereafter, we refer to Non-Reciprocal-Interaction-Based (NRIB) model) is highly abstract and it is difficult to conclude that it exactly mimics real social phenomena, the NRIB model can potentially contribute to various issues such as understanding the essence of collective behaviors of animals and active matters (Tanaka et al. (2017)) and designing swarm robotic systems.

This study focuses on the application of the NRIB model to the control of swarm robotic systems. Specifically, we extend the NRIB model to provide new insights into the problem of controlling swarm robots that can perform spatially distributed tasks in parallel (*e.g.*, swarm robots that can split into several groups to efficiently clean a room in which dust is spatially distributed) (Aşı k and Akı n (2017); Claes et al. (2015); Ducatelle et al. (2009)). We show via simulations that the robots driven by the proposed control scheme move and perform spatially distributed tasks in a quite reasonable manner.

We explain the proposed model hereafter. Agents exist on a two dimensional field wherein tasks are spatially distributed. Each agent has an internal state referred to "workload." The position and the workload of agent i ($i = 1, 2, \cdots, N$) are denoted by $\mathbf{r}_i$ and C_i, respectively. The amount of tasks at the position $\mathbf{r}$ is denoted by $V(\mathbf{r})$. Each agent can detect the position and the workload of agents that exist in the area S_i, which is defined as the area within the circle whose radius and origin are r_{th} and $\mathbf{r}_i$, respectively.

The basic concept of the proposed control scheme is as follows. Each agent first moves at random to search for a task. Once it finds a task, *i.e.*, it enters an area where $V(\mathbf{r})$ is large, its motion slows down to perform the task. Concurrently, its workload increases, which works to attract its neighboring agents. As a consequence, the attracted agents can perform the task cooperatively. After the task is finished, the workload of the attracted agents decreases. Then, they repel each other and search for another task again.

Thus, the time evolutions of the workload of agent i, C_i, is given by the following equation:

$$\tau \dot{C}_i = \alpha V(\mathbf{r}_i) - C_i, \tag{1}$$

where α and τ are positive constants. Note that C_i is reset to 1 when it exceeds 1. Equation (1) means that C_i becomes large when agent i remains in the area where $V(\mathbf{r})$ is large.

The amount of task $V(\mathbf{r})$ decrements by ϵ in each time step at points where agents exist, because tasks are performed by the agents. Note that $V(\mathbf{r})$ is reset to zero when it becomes negative. Hence, according to Eq. (1), C_i decreases when the task is finished, *i.e.*, $V(\mathbf{r}_i)$ becomes zero.

The time evolution of the position of agent i, $\mathbf{r}_i$, is given by

$$\dot{\mathbf{r}}_i = (1-C_i)\left[\sum_{j\in S_i}\{(aC_j - b)r_{ij}^{-1} - r_{ij}^{-2}\}\mathbf{e}_{ij} + \mathbf{n}_i\right], \tag{2}$$

where r_{ij} denotes the distance between agents i and j, $\mathbf{e}_{ij}$ denotes the unit vector pointing from $\mathbf{r}_i$ to $\mathbf{r}_j$, and a and b are constants satisfying $a > b > 0$. The components of the vector $\mathbf{n}_i$ changes randomly at a certain time interval, which enables random walk of the robot. The term $(aC_j - b)r_{ij}^{-1} - r_{ij}^{-2}$ in Eq. (2) means that agent i approaches and repels from agent j when the workload of agent j, C_j, is high and low, respectively. Thus, agents tend to aggregate in areas where $V(\mathbf{r})$ is large. The term $(1 - C_i)$ denotes the mobility of agent i. Specifically, agent i slows down as C_i increases and stops completely when $C_i = 1$. Hence, the robot performs tasks without moving until the $V(\mathbf{r})$ decreases to some extent.

Simulations were conducted to demonstrate the validity of the proposed control scheme. The periodic boundary condition is employed. Areas where $V(\mathbf{r})$ is large are initially distributed, and the initial positions of the agents are set to be random (Fig. 1(a)). The initial workload value is set to be 0.2 for all agents. Parameter values, which were determined by trial-and-error, are as follows: $N = 30$, $a = 2.0$, $b = 0.5$, $\alpha = 3.0$, $\tau = 2.0$, $\epsilon = 0.0005$, $r_{th} = 10$. The time step is 0.002. The x and y components of the vector $\mathbf{n}_i$ are set to random values within the range of $[-0.25, 0.25]$, and are updated every 20000 time steps.

The result is shown in Fig. 1 (the movie can be downloaded from http://www.riec.tohoku.ac.jp/∼tkano/data.mpeg). Agents aggregate in several areas where $V(\mathbf{r})$ is large, and their workload becomes large (arrows in Fig. 1(b)). They remain in these areas for a while, and then $V(\mathbf{r})$ decreases, *i.e.*, the tasks are performed (arrow in Fig. 1(c)). When $V(\mathbf{r})$ becomes almost zero, the workload becomes small and the agents begin to distribute while moving randomly (Fig. 1(d)). Then, when one of the agents enters an area where $V(\mathbf{r})$ is large, its workload increases and its neighboring agents aggregate in the area to perform the task (arrow in Fig. 1(e)). This process is repeated until the tasks are almost finished (Fig. 1(f)). In sum, agents are autonomously divided into several groups to perform spatially distributed tasks in parallel by using the proposed control scheme.

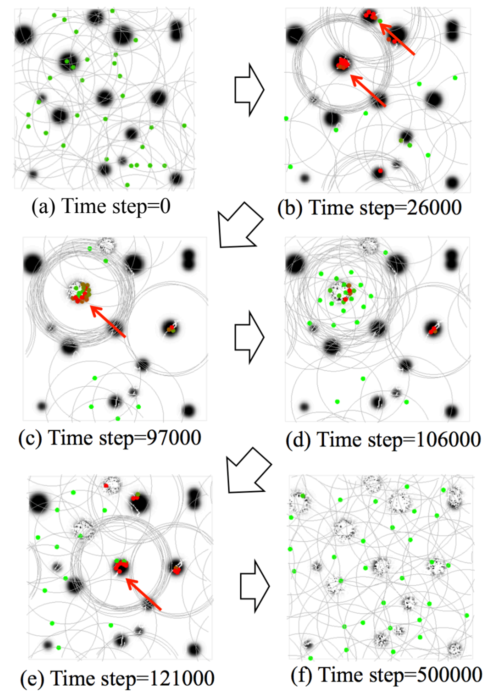

Figure 1: Simulation result. Areas in which $V(\mathbf{r}_i)$ is large are denoted by a dense color. Agents with high and low workload are colored by red and green, respectively. Gray circles denote the sensor range of the agents. The meanings of the red arrows are explained in the main text.

In conclusion, we proposed a decentralized control scheme for swarm robots, inspired by the NRIB model. In spite of the simplicity of the proposed control scheme, it was demonstrated via simulation that spatially distributed tasks were performed in parallel by the agents. Although comparison with other methods and solving several technical issues, *e.g.*, how to measure $V(\mathbf{r})$, r_{ij}^{-1} and $\mathbf{e}_{ij}$ in real situations are still needed, we believe that the proposed control scheme could be used for various practical applications in which several spatially distributed tasks need to be performed efficiently. Moreover, it may be applicable when chasing moving targets, *e.g.* capturing fish and chasing criminals.

In future, we would like to examine the applicability of the proposed control scheme on an unstructured environment, *e.g.*, a field on which several obstacles exist. We would also like to develop real robots to validate the proposed control scheme in the real world.

Acknowledgements

The authors would like to thank Prof. Ken Sugawara of Tohoku Gakuin University and Naoki Matsui of Tohoku University for their insightful suggestions.

References

Aşı k, O. and Akı n, H. (2017). Effective multi-robot spatial task allocation using model approximations. In Behnke, S., Sheh, R., Sarı el, S., and Lee, D., editors, *RoboCup 2016: Robot World Cup XX. RoboCup 2016. Lecture Notes in Computer Science*, volume 9776, pages 243–255.

Claes, D., Robbel, P., Oliehoek, F., Tuyls, K., Hennes, D., and Hoek, W. (2015). Effective approximations for multi-robot coordination in spatially distributed tasks. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 881–890.

Ducatelle, F., Förster, A., Di Caro, G., and Gambardella, L. (2009). New task allocation methods for robotic swarms. In *Proceedings of 9th IEEE/RAS Conference on Autonomous Robot Systems and Competitions*.

Kano, T., Osuka, K., Kawakatsu, T., and Ishiguro, A. (2017a). Mathematical analysis for non-reciprocal-interaction-based model of collective behavior. *Journal of the Physical Society of Japan*, 86:124004.

Kano, T., Osuka., K., Kawakatsu, T., Matusi, N., and Ishiguro (2017b). A minimal model of collective behaviour based on non-reciprocal interactions. In *Proceedings of 14th European Conference on Artificial Life*, pages 237–244. Cambridge, MA: The MIT Press.

Tanaka, S., Nakata, S., and Kano, T. (2017). Dynamic ordering in a swarm of floating droplets driven by solutal marangoni effect. *Journal of the Physical Society of Japan*, 86:101004.