

Article

Control Input Design for a Robot Swarm Maintaining Safety Distances in Crowded Environment

Yuki Origane , Yuya Hattori  and Daisuke Kurabayashi 

School of Engineering, Department of Systems and Control Engineering, Tokyo Institute of Technology, Ookayama, Meguro-ku, Tokyo 152-8550, Japan

* Correspondence: origane@irs.sc.e.titech.ac.jp (Y.O.); hattori.yuya@sc.e.titech.ac.jp (Y.H.); dkura@irs.ctrl.titech.ac.jp (D.K.)

Abstract: We consider an autonomous and decentralized mobile robotic swarm that does not require an advanced communication system; moreover, each robot must pass a narrow space preserving the distance with other robots. The control barrier function (CBF) method is useful for robotic swarms to guarantee collision avoidance. However, introducing CBF inequalities can cancel other objectives and sometimes causes a deadlock problem. Therefore, we introduce a coupled oscillator system to generate asymmetric global order by itself to avoid deadlock. By generating an effective global order in the swarm, each robot adequately moves to a target position without requiring high-cost communication systems.

Keywords: robotic swarm; collision avoidance; control barrier function; coupled oscillator system; obstacle region



Citation: Origane, Y.; Hattori, Y.; Kurabayashi, D. Control Input Design for a Robot Swarm Maintaining Safety Distances in Crowded Environment. *Symmetry* **2021**, *13*, 478. <https://doi.org/10.3390/sym13030478>

Academic Editor: Christos Volos

Received: 19 February 2021

Accepted: 11 March 2021

Published: 15 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Several attempts have been made recently to control multi-mobile agent systems [1–10]. For example, Cui et al. performed formation control on underwater vehicles [1]. Miyano et al. proposed a cooperative object transportation method for multi robots [2]. Among them, mobile robotic swarms, which are autonomous, distributed, and do not require advanced communication, are robust to communication disruptions and robot failures. Therefore, they could be applied to systems operating in harsh environments, as well as micro-robot swarms. Multi-mobile robots must satisfy conflicting objectives such as collision avoidance, gathering, and moving to the target simultaneously. We require a swarm system to perform the conflicting objectives only by observable information from each robot. A straightforward method is to express these objectives as virtual forces to determine the input by adding them [11]. However, each objective interferes with each other and complicates parameter tuning. Furthermore, particularly when considering the size of the robot, guaranteeing collision avoidance is challenging, because we cannot predict how the collision avoidance terms are reflected in the input of the robot. In particular, in a crowded environment, robots can undergo the safety distance of other robots.

An idea to solve interference is to express constraints, such as collision avoidance, as inequalities to the input. Control barrier function (CBF) is a method to design the theoretical inequality condition. CBF is a tool that theoretically guarantees that the target system satisfies the constraints [12,13]. Ames et al. [14] applied CBF to a mobile robot and a legged robot. The CBF method was also applied to multiple robots [15–17]. However, inequality constraints using CBF have the limitation that it can cancel other objectives to guarantee constraints. Thus, introducing CBF constraints for multi-robot systems can cause deadlock during movement, as mentioned in [15]. In some situations, to overcome this deadlock problem, deciding control input based only on local observable information, such as relative position and velocity, is insufficient. Therefore, to introduce CBF and avoid deadlock simultaneously, robots must communicate global information and move appropriately

from a macroscopic perspective. However, these tasks are not straightforward for a swarm system because they must not use centralized control and high-cost communication. Each robot must decide only by limited information. For an autonomous decentralized swarm system, moving based on global information is the same as forming a global order by itself.

An alternative method for generating global order is using a coupled oscillator system (COS), where the distribution of the natural frequency of each oscillator generates a global phase gradient field covering all the oscillators. If each robot is equipped with an oscillator, the robots share the same gradient field and can move in an orderly manner.

In this paper, we consider a narrow space-passing task for a swarm with CBF inequalities. We introduce the deadlock problem due to CBF and resolve it by generating an asymmetric global order with COS. The proposed method allows introducing CBF constraints for robotic swarms to avoid deadlock problems. The simulations confirm that the proposed method satisfies the following three requirements—separate constraint and task achievement in input determinants, guarantee to maintain a safe distance between robots, and guide the robots successfully.

This method can be applied to robot systems that need to pass through narrow spaces in restricted situations in future. For example, rescue robots that operate in harsh environments, underwater exploration robots where high-speed communication is difficult.

2. Problem Statement

2.1. Target System

We consider N circular mobile robots of radius r that can move in all directions in a two-dimensional plane with obstacles. Suppose that the relative position and velocity with other robots and obstacles can be measured within its visible region r_v . Let a mobile robot i have the dynamics expressed in Equation (1) where $\mathbf{x}_i \in \mathbb{R}^2$ and $\mathbf{u}_i \in \mathbb{R}^2$ represent the position and control input of force, respectively.

$$m\ddot{\mathbf{x}}_i + \eta\dot{\mathbf{x}}_i = \mathbf{u}_i, \quad (1)$$

where m, η are positive constant values denoting mass of each robot and viscosity, respectively. The dot symbol indicates time derivative then $\dot{\mathbf{x}}_i$ is velocity and $\ddot{\mathbf{x}}_i$ means acceleration of each robot. Moreover, we construct $\mathbf{u}_i$ as follows:

$$\mathbf{u}_i = \mathbf{f}_i^O + \mathbf{f}_i^F + \mathbf{f}_i^P, \quad (2)$$

where $\mathbf{f}_i^O, \mathbf{f}_i^F$, and $\mathbf{f}_i^P$ are virtual forces denoting collision avoidance with obstacles, maintaining the distance between robots and task goal (like moving to target) term, respectively. By adding up these virtual forces, each robot decides its control input $\mathbf{u}_i$. The decision making by (2) can achieve autonomous decentralized control. However, interference between each term causes two major problems: (1) parameter tuning is complicated, and (2) there is no guarantee of collision avoidance constraint, for instance, maintaining a safe distance.

For a later simulation, we define $\mathbf{f}_i^O, \mathbf{f}_i^F$, as proposed in [18].

$$\mathbf{f}_i^O = \begin{cases} -k_O \frac{d_o^2 - \|\mathbf{x}_{io}\|^2}{d_o^2} \frac{\mathbf{x}_{io}}{\|\mathbf{x}_{io}\|} & (\|\mathbf{x}_{io}\| < d_o) \\ 0 & (\|\mathbf{x}_{io}\| \geq d_o) \end{cases} \quad (3)$$

$$\mathbf{f}_i^F = -k_F \sum_{j \in \mathcal{N}_i} \left(\frac{1}{\|\mathbf{d}_{ij}\|^3} - \frac{1}{\|\mathbf{d}_{ij}\|^2} \right) e^{-\|\mathbf{d}_{ij}\|} \mathbf{d}_{ij}, \quad (4)$$

where $\mathbf{x}_{io}$ denotes the relative position of the nearest obstacle object, and $\mathbf{d}_{ij} := \frac{\mathbf{x}_{ij}}{r_c}$, $\mathbf{x}_{ij} := \mathbf{x}_j - \mathbf{x}_i$. $\mathcal{N}_i := \{j \in \{1, \dots, N\} \mid \|\mathbf{x}_{ij}\| \leq r_v\}$ is a neighbor set of the robot i . These force terms have the distance parameter d_o , and r_c , and the weight parameters k_O , and k_F . See the Appendix A for more details on these virtual forces.

2.2. CBF Method

To avoid interference between virtual forces, CBF is a useful method. In this section, we review the method of Zeroing CBF (ZCBF). Interested readers can find the details of the method in [13]. Let state $x \in \mathbb{R}^n$ follow the input-affine system below, using input $u \in \mathbb{R}^m$

$$\dot{x} = f(x) + g(x)u, \quad (5)$$

where f and g are locally Lipschitz. Consider $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function and $\mathcal{C} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\}$. Then, $h(x)$ is said to be a (ZCBF) [13] for the set $\mathcal{C}$ if there exists an extended class $\mathcal{K}$ function $\alpha : \mathbb{R} \rightarrow \mathbb{R}$ such that

$$\sup_{u \in \mathbb{R}^m} [\dot{h}(x) + \alpha(h(x))] \geq 0. \quad (6)$$

Subsequently, if the function h is a ZCBF of $\mathcal{C}$ and if Lipschitz continuous u satisfies inequality (7) at any time, set $\mathcal{C}$ becomes forward invariant

$$L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0, \quad (7)$$

where L_f and L_g are the Lie derivatives along f and g , respectively. If we describe the safety condition as $h(x) \geq 0$, and select the control input u to satisfy constraint (7), $x(t_0) \in \mathcal{C} \Rightarrow x(t) \in \mathcal{C} \forall t \geq 0$ is guaranteed; however, we need to be careful regarding the relative degree of h .

As presented above, CBF describes the constraint as an inequality condition of input, which theoretically satisfies the constraint. However, the inequality by CBF cancels other objective terms to guarantee constraints. In this sense, CBF is a “hard” method. CBF’s “hard” input causes a deadlock problem. While one solution is adding perturbation to the control input, it is not enough in some situations. A more practical way is to introduce global information to each robot and generate a global order in the entire swarm.

2.3. COS Method

In this section, we introduce COS, which can generate global order with low-cost communication. We give each robot a new scalar state variable, ϕ_i , and we call it phase. We assume that each robot can observe the phase of its neighboring robots. Based on our assumptions, we update the phase ϕ_i of each robot following a common rule (8) [18,19].

$$\dot{\phi}_i = \Omega_i + \frac{\kappa}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} ({}^i\phi_j - \phi_i). \quad (8)$$

The introduction of phase according to (8) means that we have equipped each robot with a virtual one-dimensional oscillator. In the equation, Ω_i indicates the natural frequency of each oscillator, κ is a common constant gain for interactions, ϕ_i is the phase of oscillator i , and ${}^i\phi_j$ is the phase of oscillator j observed by robot i . Once we assume that observation matches the true value, ${}^i\phi_j = \phi_j$. Since the oscillators of each robot interact with each other, the entire robots constitute a COS. In the COS consisting of oscillators (8), the phase gap of the oscillators converges to a specific value and generate a phase gradient [20].

We will present a simple example. Suppose that four robots are lined up in a straight line (Figure 1). In the figure we show the phase of each robot (oscillator) as a clock.

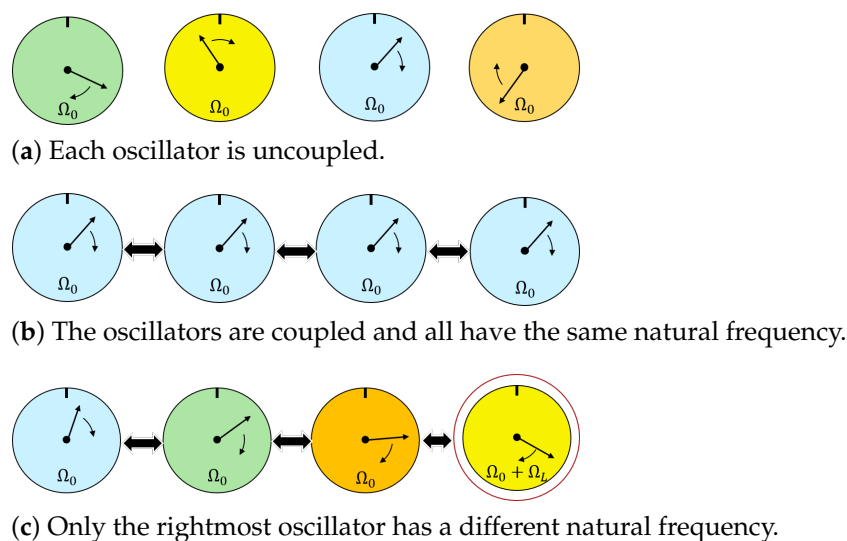


Figure 1. Synchronization of linearly aligned oscillators.

When there is no coupling between the oscillators ($k = 0$), the phase of each oscillator is determined individually (Figure 1a). Then, if the neighboring oscillators are coupled ($k > 0$), the oscillators are synchronized. If the natural frequency of each oscillator Ω_i is the same, phase of each oscillator will converge to the same value (Figure 1b). In contrast, Increasing the natural frequency of the rightmost oscillator (circled in red) produces a phase gradient over the entire system as shown in Figure 1c.

As like example, there will be a spatial gradient in the phase value according to the spatial distribution of Ω_i as shown in Figure 2. By observing the phase value of the oscillator, each robot knows the gradient at each position. Each robot follows a globally shared gradient field (9), and then moves in an orderly manner. Therefore, by appropriately specifying Ω_i , we can generate a global order in a robotic swarm.

$$f_i^P = k_P \frac{F_i^P}{\|F_i^P\|}, \quad F_i^P = \sum_{j \in \mathcal{N}_i} (\phi_j - \phi_i) \frac{x_{ij}}{\|x_{ij}\|}. \quad (9)$$

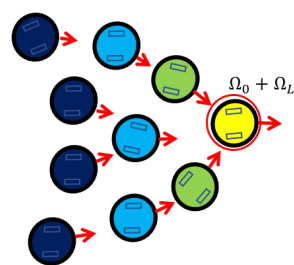


Figure 2. Image of guidance using the phase gradient. The robot's color gradation indicates the gradient of the phase value of the oscillator. The robot marked by a red circle is the leader.

2.4. Objective

In this study, we deal with passing a narrow space problem for robotic swarms [9]. The description of this task is presented later. For this task, our swarm system must satisfy the following three requirements:

- (i) Separate constraint and task achievement in input determinants;
- (ii) Guarantee to maintain a safe distance between robots;
- (iii) Successfully guide the robots;

Requirements (i) and (ii) are satisfied by introducing a CBF constraint. However, CBF inequalities cause deadlock and disturb satisfying requirement (iii). To overcome this deadlock problem, we use COS to generate a global order.

In this study, we set the narrow space as shown in Figure 3.

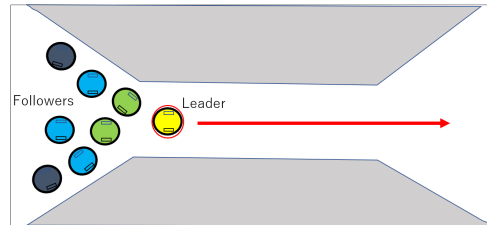


Figure 3. Image of a narrow passage task for mobile robots. The robot marked by a red circle is the leader, and the other robots are followers.

We assume that a leader robot knows the target path of the robots. This assumption applies to situations where humans control specific robots and cases such as the odor source search problem, where some robots detect the odor source and use the phase gradient information to guide the entire swarm.

For the leader to guide the followers, the natural period Ω_i of each oscillator is set as follows:

$$\Omega_i = \begin{cases} \Omega_0 + \Omega_L & i \text{ is Leader} \\ \Omega_0 & i \text{ is Follower.} \end{cases} \quad (10)$$

Throughout the task, Ω_i of each oscillator is kept constant. By following the setting mentioned above, a phase gradient is formed around the leader, and followers are guided in the direction of the leader.

3. Constraints by CBF for Robot's Modeling

3.1. CBF for a Swarm

We introduce the CBF constraint for our swarm system. Let be the control period T_c and $q = [q_1, q_2]^T = [x_{ij}, \dot{x}_{ij}]^T$. The safety function $h(q)$ between agent i, j is described as follows:

$$h(q) = \|q_1\| + q_2^T \frac{q_1}{\|q_1\|} T_c - r_s, \quad (11)$$

where r_s denote the minimum safety distance. From each agent system, Equation (1), q can be expressed as follows:

$$\dot{q} = f(q) + g(q)(u_j - u_i). \quad (12)$$

Consider $f(q) = (q_2, -\frac{\eta}{m}q_2)^T$, $g(q) = (0, \frac{1}{m})^T$. Using Equations (11) and (12), the input constraint of each agent corresponding to Equation (7) can be written as follows:

$$L_f h(q) - 2L_g h(q)u_i + \gamma h(q) \geq 0 \quad (13)$$

where $\gamma > 0$ is a parameter denoting the constraint. and this parameter denotes naively of constraint. If we determine u_i satisfying inequality constraint (13), $h(q) \geq 0$ is theoretically satisfied. This implies that requirement (b) is fulfilled. Moreover, we construct the raw control input $\hat{u}_i$ as follows:

$$\hat{u}_i = \begin{cases} u^{trace} & i \text{ is Leader} \\ f_i^P + f_i^F & i \text{ is Follower.} \end{cases} \quad (14)$$

u^{trace} is the force following the target path given to the leader robot. For follower robots, f_i^O is deleted from Equation (2), because this force only denotes the collision avoidance constraint. f_i^F is also a constraint term. However, this term remains because it also aims

to ensure an appropriate distance between individual robots. Finally, we determine the robot's control input using the Quad Program form as follows:

$$\begin{aligned} u_i &= \arg \min_{u_i \in \mathbb{R}^2} \frac{1}{2} \|u_i - \hat{u}_i\|^2 \\ \text{s.t. } L_f h(q) - 2L_g h(q)u_i + \gamma h(q) &\geq 0. \end{aligned} \quad (15)$$

By determining the input in the form of Equations (14) and (15), we separate constraints from other objectives (requirement (i)) and obtain the theoretical guarantee of collision avoidance (requirement (ii)). We succeed in writing the constraints in the form of inequality using CBF. The separation of constraints and achievement is performed; therefore, the requirement (i) is achieved.

3.2. Deadlock Solution by COS

We introduce an example of a deadlock problem due to CBF and propose a method to resolve it. The follower robot moves in the direction of the leader robot by f_i^P . It sometimes moves around in front of the leader robot, as shown in Figure 4. This is particularly true when the leader slows down or changes direction. Once the leader robot is encircled, the inequality condition of the collision avoidance constraint prevents the leader from moving, and the entire robot stops. We call this the deadlock problem due to leader encirclement. We have to prevent the occurrence of this deadlock problem in order to fulfill the requirement (iii).

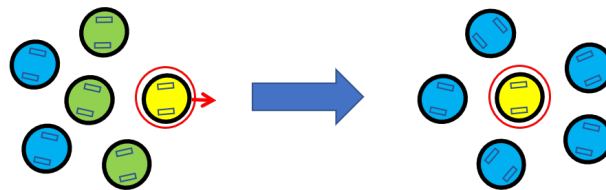


Figure 4. Leader robots is encircled by Follower robots.

We add geometrical information to the oscillator as a phase shift based on the relative direction to resolve this problem.

$${}^i\phi_l = \phi_l + c|\theta_{li}|, \quad (16)$$

where robot l is the leader, c is a positive constant, and $\theta_{li} \in [-\pi, \pi]$ denotes the direction in which follower i exists relative to the direction of movement of the leader l . This equation changes the phase gradient to be asymmetric in the direction of the leader. This equation leads the phase to be higher on opposing the movement direction of the leader (asymmetry). By coupling the oscillators, the robots share the gradient field with this tendency. Thus, an orderly movement exists that does not disturb the movement of the leader.

We have another method for avoiding deadlock, for example, by including perturbations in the robot's input. However, small perturbations to $\hat{u}_i$ are cancelled out by the CBF. Large perturbations prevent task achievement and cause the swarm to shred. Perturbations to u_i violate the CBF condition and thus violate the safe distance. Our method achieves deadlock avoidance without affecting the swarm and preserves the CBF condition.

4. Simulations

In this section, we verify that the robotic swarm using the proposed method satisfies requirements (ii) and (iii). The simulation comprises a narrow passage simulation. The unit of x_i is the meter and the unit of u_i is the newton in the simulation. We simulated 20 robots within a $16 \text{ m} \times 16 \text{ m}$ plane field, as shown in Figure 5. To evaluate a requirement (iii), we introduce a criterion. We determine the center of the swarm from the average of the robot positions. When the center of the swarm enters the target area (a circle of radius 2 (m) from the target point), we define that the swarm guidance has been achieved.

In the simulations, $T_c = 0.02$ (s). and other parameters are listed in Table 1. The parameter selection for the robotic system and COS is based on [20–22]. The parameters of the CBF were selected experimentally. Simulations were performed using MATLAB R2020a. In particular, *quadprog* function in the optimization toolbox of MATLAB was implemented.

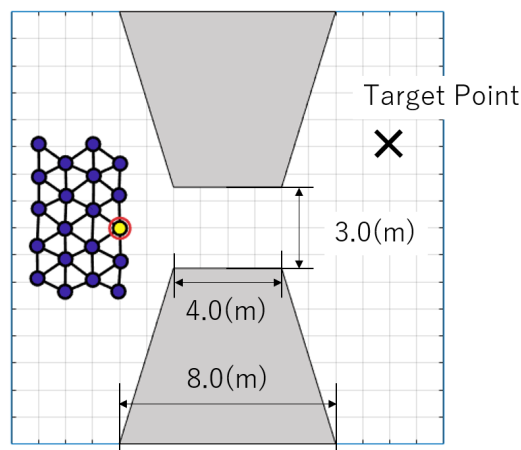


Figure 5. Simulation field and initial position of the robots.

Table 1. Parameters Specification.

Parameter	Value	
m	1	mass of each robot
k_d	20	damping constant
r_v	1.5	visible range
k_f	40	weight constant of f_i^F
r_c	1	standard distance between robots
k_p	4	weight constant of f_i^P
k_o	40	weight constant of f_i^O
d_o	0.5	robot's avoiding distance with obstacle
r_s	0.8	safety distance between robots
r_{so}	0.4	robot's safety distance with obstacle
γ	4	constant value of CBF constraint
Ω_0	0.4	common natural period
Ω_L	0.04	leader's natural period changing
κ	11.5	coupling intensity
c	0.3	weight of direction effect

First, we check the distances between robots to verify requirement (ii). Figure 6 shows smallest distance between each robot. Figure 6a shows the case without the CBF constraint; we can see that the minimum distance between robots (blue curve) violates the safety distance (red line). This result implies that a collision between robots can occur. In contrast, in the CBF constraint case shown in Figure 6b, the minimum distance of robots maintain the safety distance. By introducing a CBF constraint, the robots avoid a collision. Therefore, requirement (ii) is fulfilled.

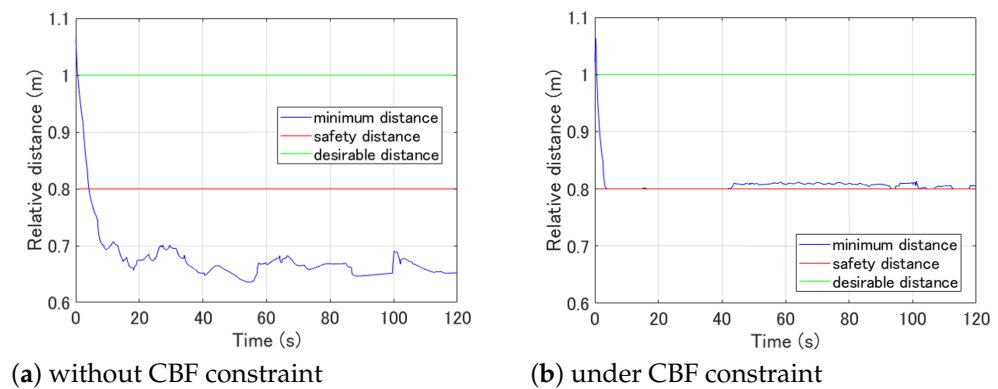


Figure 6. The minimum distance between each robot. Blue lines denote minimum distance between robots. Red lines are safety distance to be preserved. Green lines are standard distance.

Subsequently, we verify the deadlock problem due to CBF. Figure 7 presents a snapshot of the robotic swarm with CBF trying to pass a narrow space. The red circled robot is the leader. As seen in the figure, after the simulation started, the follower robots encircle a marked leader robot, and the entire robotic swarm stops moving. In contrast, in the simulation shown in Figure 8, the robotic swarm successfully passes a narrow space. In this simulation, each robot follows global order generated by COS as described in Section 3.2. Follower robots do not move forward to the leader's direction, and the leader robot completes the entire swarm. We can see the asymmetric global order (phase gradient). See, for example, Figure 8a. Among the robots that are near the leader, the robots on the opposite side of the leader's movement direction are yellow (the phase is higher), while the other robots are orange or green (the phase is lower). This tendency is also transmitted to the following robots. This shows that the COS forms a global order (phase gradient) that is asymmetric with respect to the leader's direction of movement.

We check whether the criteria of requirement (iii) are satisfied. Figure 9 shows distances from the center of the swarm to the target point. Figure 9a shows the case where deadlock occurs. That the center of the swarm (blue line) does not reach the target area (red line). On the other hand, when the global order by COS avoids deadlock (Figure 9b), the center of the swarm reach the target area. From the simulation results shown in Figure 7–9, we verified that the global order by COS suppresses the deadlock problem, thus, verifying the requirement (iii).

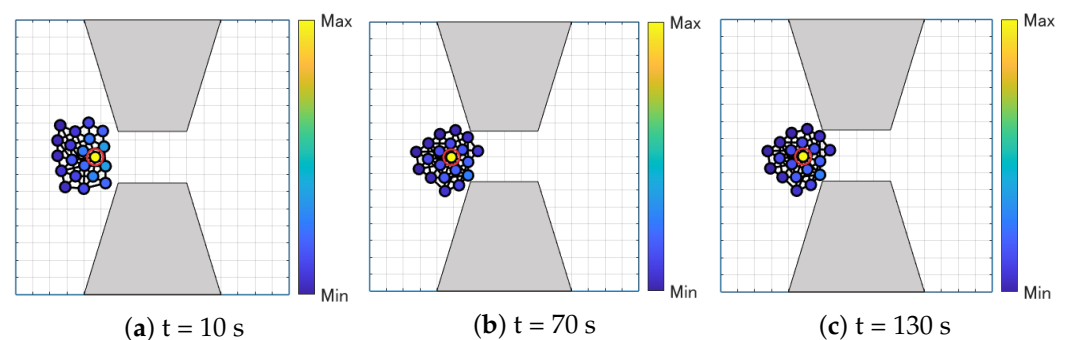


Figure 7. Guidance simulation results. Deadlock problem due to leader encirclement occurs. The color of each robot shows its phase value.

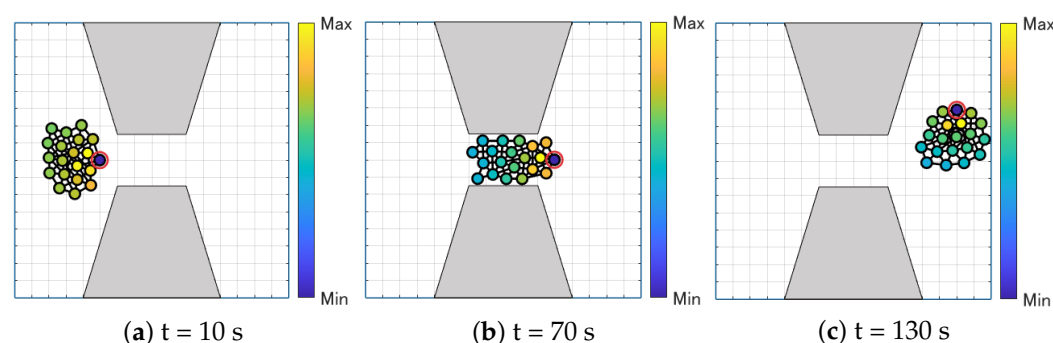


Figure 8. Simulation results under geometrical information to oscillator. Robots successfully pass a narrow space. The color of each robot shows its phase value.

Therefore, we confirmed that the proposed method satisfies all requirements (i), (ii), and (iii).

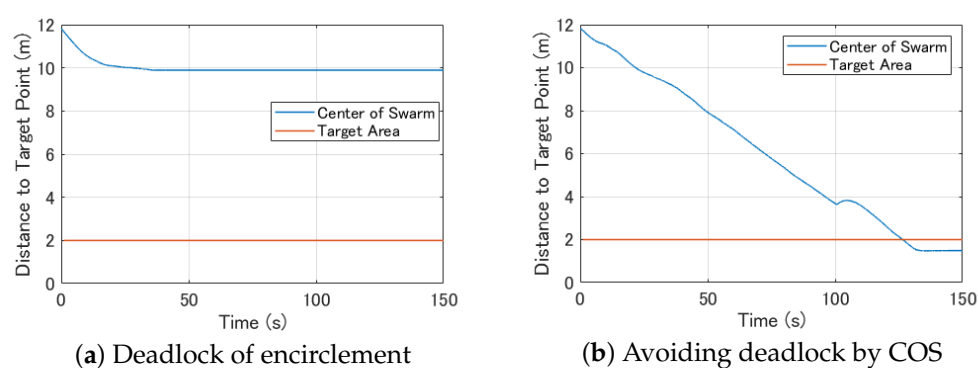


Figure 9. Distance from the center of the swarm to the target point. Red line shows the boundary of the target area to be entered.

5. Conclusions

We required autonomous decentralized swarms with a leader to pass through narrow spaces while maintaining a safe distance. While CBF guarantees a safe distance between robots, it causes a deadlock problem due to encirclement. To prevent leader encirclement, we equipped the swarm with virtual oscillators to generate an asymmetric global order (phase gradient). We verified the effectiveness of the proposed method by simulations. Then we achieved to do both introducing CBF and deadlock avoidance by generating global order by COS. Proposed method gives a way to introduce CBF constraint for robotic swarm avoiding deadlock problem. In the future, we need to consider swarm robots that can perform a variety of tasks in each environment. We have to consider not only leader encircled but also other types of deadlock. In this study, the parameters of CBF were selected experimentally. We are considering a method to select these parameters theoretically. It is also important to investigate how constraints such as CBF affect the characteristics of a swarm system.

Author Contributions: Y.O., Y.H., and D.K. conceived of and designed the system and the algorithm. Y.O. performed analysis and simulations, while Y.O. and D.K. wrote the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Sekisui Chemical Grant Program (Shizen-ni Manabu Monotsukuri).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Description of Virtual Forces

We describe form of virtual forces (3), (4) used in the simulation (Figure A1). We determined the equations form as proposed in [18]. We show these equations again.

$$f_i^O = \begin{cases} -k_O \frac{d_o^2 - \|x_{io}\|^2}{d_o^2} \frac{x_{io}}{\|x_{io}\|} & (\|x_{io}\| < d_o) \\ 0 & (\|x_{io}\| \geq d_o) \end{cases} \quad (A1)$$

$$f_i^F = -k_F \sum_{j \in \mathcal{N}_i} \left(\frac{1}{\|d_{ij}\|^3} - \frac{1}{\|d_{ij}\|^2} \right) e^{-\|d_{ij}\|} d_{ij} \quad (A2)$$

f_i^O is a virtual force to give robots the property of avoiding obstacles where x_{io} denotes the relative position of the nearest obstacle object. d_o is distance parameter where robots start avoiding movement. Figure A2a shows magnitude of f_i^O . f_i^F is a virtual force for maintaining the distance between robots where $d_{ij} := \frac{x_{ij}}{r_c}$, $x_{ij} := x_j - x_i$. This virtual force contributes to both collision avoidance and cohesion. r_c represents the desired distance between robots. When $\|x_{ij}\| > r_c$, f_i^F produces attraction, and when $\|x_{ij}\| < r_c$, it produces repulsion. Figure A2b shows the magnitude of f_i^F in function of the change in x_{ij} .

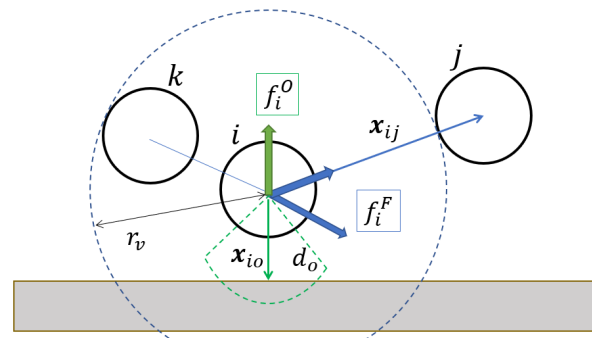


Figure A1. Virtual forces of a robot i . The green dashed line shows range of d_o . f_i^O Since the distance to the brown obstacle x_{io} is smaller than d_o , f_i^O is represented by the green arrow. The blue dashed line represents the range of r_c for robot i . Robot i is repelled by robot k inside the r_c and attracted by robot j outside the r_c .

f_i^O and f_i^F have weight parameters k_O and k_F , respectively. In practice, we generate these virtual forces in the robot by programming it. We have to adjust the balance of the virtual forces appropriately by changing the values of the weight parameters.

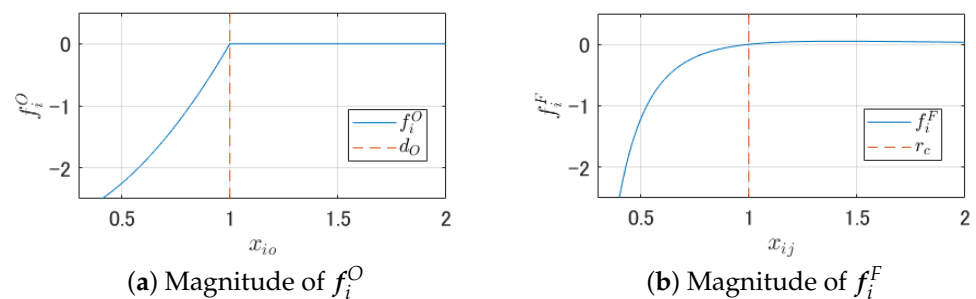


Figure A2. Magnitude of f_i^O with x_{io} and f_i^F with x_{ij} . We set $d_o = r_c = 1$ and $k_O = 3, k_F = 1$.

References

1. Cui, R.; Ge, S.S.; How, B.V.E.; Choo, Y.S. Leader–follower formation control of under actuated autonomous underwater vehicles. *Ocean Eng.* **2010**, *37*, 1491–1502. [\[CrossRef\]](#)
2. Miyano, T.; Romberg, J.; Egerstedt, M. Distributed Force/Position Optimization Dynamics for Cooperative Unknown Payload Manipulation. In Proceedings of the 2020 59th IEEE Conference on Decision and Control (CDC), Jeju, Korea, 14–18 December 2020.
3. Shimoyama, N.; Sugawara, K.; Mizuguchi, T.; Hayakawa, Y.; Sano, M. Collective Motion in a System of Motile Elements. *Phys. Rev. Lett.* **1996**, *76*, 3870–3873. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Hayakawa, T.; Matsuzawa, T.; Harat, S. Formation Control of Multi-Agent Systems with Sampled Information. In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 13–15 December 2006; pp. 4333–4338.
5. Angeli, D.; Bliman, P.A. Stability of leaderless discrete-time multi-agent systems. *Math. Control Signs Syst.* **2006**, *18*, 293–322. [\[CrossRef\]](#)
6. Mohanarajah, G.; Hayakawa, T. Influence of Stochastic Communication Loss on the Stability of a Formation of Multiple Agents. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 341–346.
7. Dimarogonas, D.V.; Kyriakopoulos, K.J. On the Rendezvous Problem for Multiple Nonholonomic Agents. *IEEE. Trans. Autom. Control* **2007**, *52*, 916–922. [\[CrossRef\]](#)
8. Pimenta, L.C.; Pereira, G.A.; Michael, N.; Mesquita, R.C.; Bosque, M.M.; Chaimowicz, L.; Kumar, V. Swarm Coordination Based on Smoothed Particle Hydrodynamics Technique. *IEEE. Trans. Autom. Control* **2013**, *29*, 383–399. [\[CrossRef\]](#)
9. Sakai, D.; Fukushima, H.; Matsuno, F. Leader–Follower Navigation in Obstacle Environments While Preserving Connectivity Without Data Transmission. *IEEE Trans. Control. Syst. Technol.* **2017**, *26*, 1233–1248.
10. Suzuki, K.; Tsukidate, T.; Nakada, T.; Shimizu, M.; Ishiguro, A. Self-assembly through the Local Interaction between “Embodied” Nonlinear Oscillators with Simple Motile Function. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 1319–1324.
11. Gazi, V.; Passino, K.M. *Swarm Stability and Optimization*; Springer: Berlin/Heidelberg, Germany, 2011.
12. Ames, A.D.; Grizzle, J.W.; Tabuada, P. Control barrier function based quadratic programs with application to adaptive cruise control. In Proceedings of the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December 2014.
13. Ames, A.D.; Xu, X.; Grizzle, J.W.; Tabuada, P. Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Trans. Autom. Control* **2017**, *62*, 3861–3876. [\[CrossRef\]](#)
14. Ames, A.D.; Coogan, S.; Egerstedt, M.; Notomista, G.; Sreenath, K.; Tabuada, P. Control Barrier Functions: Theory and Applications. In Proceedings of the 2019 18th European Control Conference (ECC), Naples, Italy, 25–28 June 2019; pp. 3420–3431.
15. Borrmann, U.; Wang, L.; Ames, A.D.; Egerstedt, M. Control Barrier Certificates for Safe Swarm Behavior. *IFAC PapersOnLine* **2015**, *48*, 68–73. [\[CrossRef\]](#)
16. Endo, M.; Ibuki, T.; Sampei, M. Collision-free Formation Control for Quadrotor Networks Based on Distributed Quadratic Programs. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 3335–3340.
17. Ibuki, T.; Wilson, S.; Yamauchi, J.; Fujita, M.; Egerstedt, M. Optimization-Based Distributed Flocking Control for Multiple Rigid Bodies. *IEEE Robots. Autom. Lett.* **2020**, *5*, 1891–1898. [\[CrossRef\]](#)
18. Kurabayashi, D.; Choh, T.; Cheng, J.; Funato, T. Adaptive Formation Transition among a Mobile Robot Group based on Phase Gradient. *J. Robot. Mechatron.* **2010**, *22*, 467–474. [\[CrossRef\]](#)
19. Horayama, K.; Kurabayashi, D.; Ahmad, S.; Hashimoto, A.; Moriyama, T.; Choh, T. Guidance of Robot Swarm by Phase Gradient in 3D Space. In *International Conference on Intelligent Robotics and Applications*; Springer: Cham, Switzerland, 2016; p. 9834.
20. Hashimoto, A.; Kurabayashi, D.; Moriyama, T.; Choh, T. Asymmetric Global Order to Control a Group of Anonymous Mobile Robots by Using Loose-Coupled Local Interactions. *Trans. JSME* **2013**, *79*, 144–154. (In Japanese)
21. Yamashita, S.; Kurabayashi, D.; Hattori, Y. Autonomous division and integration of anonymous agents by using interaction between oscillators. *Trans. JSME* **2018**, *84*, 1–14. (In Japanese)
22. Yokoo, K.; Kurabayashi, D.; Hattori, Y. State Analysis in Navigation of Multiple Autonomous Mobile Robots by Local interaction. *Robomech* **2019**. (In Japanese)