electronics

Article

Self-Organized Aggregation Behavior Based on Virtual Expectation of Individuals with Wave-Based Communication

Phan Gia Luan and Nguyen Truong Thinh *

Institute of Intelligent and Interactive Technologies, University of Economics Ho Chi Minh City—UEH, Ho Chi Minh City 700000, Vietnam; luanpg@ueh.edu.vn
* Correspondence: thinhnt@ueh.edu.vn

Abstract: In this study, a microscopic model for a swarm of mobile robots is developed to implement self-organized aggregation behavior. The proposed model relies on the concept of subjective expectation, which is defined as the "minimum wished cluster size" of a robot in the swarm. During the whole process, a robot's expectation constantly changes based on context awareness. This awareness is obtained by employing a low-cost communication system commonly found in swarm robot studies: infrared-based communication. Robots can make their own decisions by comparing their expected and estimated observed cluster sizes, which leads to macroscopic swarm aggregation. However, due to the limitations of local communication and mobility, robots are restricted in their ability to perceive global information, particularly regarding cluster size. Inspired by the slime mold aggregation process, a wave-based communication mechanism is implemented to help robots estimate a cluster size. Moreover, each transmission includes a modulated message that allows robots to explicitly share their knowledge with others. In this way, despite the fact that the robot may not belong to that cluster due to its perception range, it can easily grasp the cluster size when passing the cluster. Once a robot detects a desired cluster, it can approach this cluster with its direction determined by using average origin of wave (AOW) method. The performance of the aggregation algorithm was tested both in simulation and with a real swarm robot. Dispersion metrics and cluster metrics were employed to evaluate the proposed algorithm's performance. The results show that the proposed microscopic model utilizes collective behavior which aggregates all randomly distributed robots into a single aggregate cluster with a reasonable swarm density and evaluation time.

Keywords: self-organized behavior; aggregation; swarm robots; self-organized aggregation; mobile robots

Citation: Gia Luan, P.; Truong Thinh, N. Self-Organized Aggregation Behavior Based on Virtual Expectation of Individuals with Wave-Based Communication. *Electronics* **2023**, *12*, 2220. https://doi.org/10.3390/electronics12102220

Academic Editor: Dah-Jye Lee

Received: 18 April 2023
Revised: 8 May 2023
Accepted: 9 May 2023
Published: 13 May 2023

1. Introduction

The success of collective behaviors exhibited by many species living in groups called "swarms" in nature has highly attracted many robotic researchers in recent decades. The term "swarm robots" can be derived from "swarm intelligence" as the emergence of macro-level behavior in a whole swarm that can be formed from the collaboration of many simple micro-level behaviors of individuals [1]. Swarm robots can do collective tasks without the intervention of a central controller during their operation through cooperation between individuals. Hence, swarm robots have many advantages compared to other types of robot systems: scalability, flexibility and robustness [2]. The robot intercommunication and interaction rules must be appropriately implemented in order to construct a protocol by which a group of robots can cooperate and achieve a global goal. In general, these materials are mostly inspired by the characteristics of society from insects and animals, such as flocks of birds, school fish, ant colonies, and honeybees to unicellular life forms such as slime molds, bacteria, and blood cells.

One of the most basic behaviors of swarms in nature, which can be seen in a wide variety of biological systems ranging from unicellular organisms to social insects and humans, is aggregation. It is crucial for the formation of functional groups of individuals, as it lays the foundation for the establishment of diverse forms of collaboration [3]. In fact, it can be considered a precondition for completing a variety of collective tasks, such as enabling them to resist against natural hazards and strengthening them with sensing capabilities to seek food, resources, and escaping routes, among others. Environmental cues are known to enhance some aggregation behaviors, such as honeybees aggregating on optimal temperature areas (which inspired BEECLUST [4]) and dry wood termites responding to temperature and light gradients [5]. However, other natural aggregations are purely self-organized. Cockroach aggregation [6], aggregation in chick penguins [7], and slime mold multicellular development [8] are fascinating instances of this kind of behavior. This form of aggregation is unique in that it can occur at irregular areas without any environmental stimuli and in the absence of a central controller. Furthermore, it is frequently accomplished in a coherent manner by individuals utilizing very simple navigation strategies and local interaction rules.

The self-organized aggregation behavior has been addressed in numerous swarm robotics projects based on these biological investigations. It is regarded as a precursor to more complicated behaviors, including flocking, self-assembly, and pattern formation. In swarm robotics, there are currently three techniques for self-organized aggregation in swarm robotic systems: the probabilistic approach, evolutionary approach, and potential field approach [9].

The artificial potential field approach is frequently used in robot navigation. It involves calculating the forces that influence how robots move in the workspace with respect to the location of the surrounding robots and obstacles. Inspired by birds and frogs, Melhuish et al. proposed a method which used a chorus consisting of individuals who can approximate the size of the aggregates using variations in sound [10]. Electrical limitations, on the other hand, cause the so-called saturation effect in robot sensors, which restricts the upper bound of input signal intensity which is used to control robot behavior. To avoid suffering from the input signal saturation problem, Belkacem et al. proposed Distance-Weighted K-Nearest Neighbors topology which is revealed in studies in birds flocking and fish schooling [11]. The self-organized aggregation is the result of a method based on an intra-virtual physical connection between neighboring robots. The improved version using the Minkowski metric was introduced in [12]. However, the naive potential field approach does not provide the balance of exploration and exploitation, which results in swarms that may aggregate into many smaller clusters instead of the unique one due to local minimum convergence. A probability approach is often applied to enhance the exploration behavior of the aggregation process.

In the probabilistic approach, the behavior of each robot has a random component that is adjusted in the process of the robot's interaction with the environment. This approach often relies on probabilistic finite state machines (PFSMs) [13–17]. This type of behavior is often found in the natural world in social insects such as bees or cockroaches. Jeanson et al. [18] investigated aggregation in cockroach larvae and developed a model of their behavior. The cockroaches were reported to join and leave clusters with probabilities correlated to the sizes of the clusters.

In the case of the evolutionary control method, aggregation dynamics is achieved by using robot controllers, the parameters of which are selected in the process of artificial evolution. Examples of the controllers, using this method, are neural networks [19]. Depending on the algorithm being used, the inputs of the evolutionary process may include devices able to receive information about the environment, and the outputs may include devices allowing robots to move and communicate with each other. Instead of using neural networks, Katada introduced the implementation of particle swarm optimizer (PSO) to evolve parameters of PFSM [14].

In conclusion, a single aggregate cluster emerges from the continuous aggregation and disaggregation of clusters. In order for a single aggregate cluster to emerge, the behavior of the participating individuals needs to change as a function of cluster size.

In this study, the swarm aggregation is approached in a different way. The key idea of the proposed approach is to create the biggest aggregate cluster; robot always have a desire to join a bigger cluster. Based on these ideas, a microscopic behavior for a swarm of robots is designed that relies on two main factors: subjective expectation which stands for "minimum wished cluster size" and actual cluster size. The first parameter of the proposed approach, subjective expectation, is driven only by the local awareness of each individual. The effects of awareness on robot expectation are simplified into three: disappointment effect, motivation effect and influence effect. These effects change robot expectations over time. By comparing the robot expectation and cluster size, robots can determine their actions: join or leave the cluster.

In this study, the main contribution is to propose a novel microscopic model for randomly placed robots in a closed workspace that allows them to aggregate into a unique cluster. The model does not require robots to have prior knowledge of the workspace, to form a connected communication network, or to have localization capabilities. Furthermore, the communication mechanism inspired by the slime mold aggregation process has been modified and applied to the proposed robot platform, allowing robots to estimate and transfer cluster size. This communication mechanism was often considered as a built-in function of robots in many previous articles [14,16,17]. The Average of Origin Wave (AOW) approach is employed to help the robot estimate the heading to move toward the cluster [20]. Then, the robot can navigate itself to the cluster when the cluster size is desirable. The AOW approach has also been modified to adapt to the proposed communication mechanism. Both simulations and real experiments were used to test the performance of the aggregation behavior. Simulation results show that the proposed microscopic model emerges from collective behavior in which a single aggregate cluster can be formed. The proposed model is applicable with a large-scale swarm of robot in a large arena with the appropriate parameter setting.

The structure of this paper is outlined below. In Section 2, a brief formulation of the aggregation problem studied is provided. Section 3 introduces the new microscopic model proposed. The communication mechanism, inspired by slime mold aggregation, is presented and modified in Section 4. Section 5 explains the navigation algorithms, including the roaming strategy and AOW approach. The experimental setup used to evaluate the proposed model and results are discussed in Section 6. Lastly, in Section 7, future plans will be discussed, and the study will be concluded.

2. Problem Formulation

Let us consider a swarm of homogeneous robots $\mathbf{S} = \{R_1, R_2, \ldots, R_N\}$ that consists of N individuals where R_i denotes the ith robot. The individual is based on a two-wheel different-drive platform consisting of two separately controlled wheels. The global position of R_i, $\mathbf{P}_i = \begin{bmatrix} x_{\mathbf{P},i} & y_{\mathbf{P},i} \end{bmatrix}^T$ can be indicated based on the relationship between the reference frame fixed in the workspace and the local frame attached to the center of the physical body of the considered robot. Let considered robot R_i have $\dot{\theta}_{l,i}$ and $\dot{\theta}_{r,i}$, which are the angular speed of the left and right wheels, respectively. Let l_w be the distance between two controlled wheels with size of r_w; the instantaneous linear speed V_i and angular speed ω_i of R_i with respect to the local frame can be deduced from Equation (1).

$$\begin{bmatrix} \dot{\theta}_{r,i} \\ \dot{\theta}_{l,i} \end{bmatrix} = \begin{bmatrix} \frac{1}{r_w} & -\frac{l_w}{r_w} \\ \frac{1}{r_w} & \frac{l_w}{2r_w} \end{bmatrix} \begin{bmatrix} V_i \\ \omega_i \end{bmatrix} \tag{1}$$

To enable the perception of the surrounding environment and intercommunication between individuals, each robot is equipped with a communication system consisting of m infrared modules installed around the periphery of the robot. Each module has one *IR* light-emitting diode (LED) and one IR phototransistor. These modules are symmetrically arranged with fixed spacing of $2\pi/m$ radians, providing full 2π radians coverage. Let r_s and r_c be the environment sensing range and communication range of the robot. Let $\mathbf{N}_i$ be the neighbor set of R_i where a neighbor is defined as a robot which can communicate with R_i. The robot does not have knowledge about $\mathbf{N}_i$, since robots in a swarm do not have any identification. However, the number of neighbors, $|\mathbf{N}_i|$, can be estimated by counting *IR* modules which received a message in a specific time. Due to the light-of-sight properties of infrared modules, each robot just can communicate with a limited number of robots within their range, despite the presence of in-range neighbors. Hence, the interference is also significantly reduced.

The introduced algorithm's aim is to have all of the randomly distributed robots in **S** congregate somewhere in the closed area with a size of A that is not predetermined. Hence, the proposed microscopic model implemented in robots must be capable of navigating the robots to emerge in a single aggregate cluster based on local information from $\mathbf{N}_i$ for R_i. In order to determine the characteristics of the proposed algorithm, some assumptions are clarified:

- Robots do not have information regarding the scenario, including: size and shape of the arena, obstacles position, their global position, and swarm size. However, the maximum size of swarm $N_{\max}$ is stated, i.e., the proposed algorithm is scalable with an upper bound constraint.
- r_s and r_c can be significantly smaller than the size of the arena that leads to individuals not necessarily forming a connected graph according to their initial placement.
- During the aggregation process, individuals can be added or removed at any time.

3. Self-Organized Aggregation Behavior

As a result of the assumptions in Section 2, only the spectators know when the aggregation process is complete. Meanwhile, robots with local information are unable to identify the end of the process. Hence, even if they are already in the largest cluster, robots should continue to search the arena for larger clusters. That means the algorithm should maintain the balance between the robot's exploration and exploitation behaviors. Almost all previous efforts have used a probabilistic technique in which the robot's chances of leaving and staying vary as a function of cluster size, implying that the larger the cluster, the less likely the robot will leave [13–17]. However, in terms of local information, robots cannot acquire this knowledge on their own without a suitable information exchange technique among them. The communication mechanism which provides this kind of knowledge will be considered in next section. In this section, the size of a cluster is assumed to have been estimated by individuals who are members of the considered cluster and successfully transferred to others within the communication range. Let n_i and $\widetilde{n}_i$ be the actual size and estimated size of cluster which R_i belongs to.

In the proposed control strategy, there are two main states: the roaming state and aggregating state. Let s_i represent a state of R_i where $s_i = 1$ stands for the aggregating state and $s_i = 0$ stands for the roaming state. Admit that when $s_i = 0$, then $n_i = \widetilde{n}_i = 0$. The main idea of the proposed approach is to make randomly distributed swarm robots congregate into a single aggregate cluster based on the subjective expectation of each individual instead of using probabilistic parameters. In this study, the subjective expectation of R_i, $n_{e,i}$, is defined as the minimum size of a cluster where R_i wants to join. The expectation of R_i is changed over time based on many factors related to the robot's local awareness. Let ς_i be the rate of change of expectation of R_i, $n_{e,i}$ at time step t can be determined by employing Equation (2).

$$n_{e,i}(t) = n_{e,i}(t-1) + \varsigma_i(t) \tag{2}$$

The value of ς_i at instant time is determined based on the following effects formulated in Equation (3):

- Disappointment effect: When robots do not meet their desired cluster, their expectation decreases over time with a rate of ς_d.
- Motivation effect: When robots meet their desired cluster, their expectation increases over time with a rate of ς_m.

- Influence effect: In aggregating state, the expectation of R_i decreases over time with a rate of $\varsigma_{s,i}$ which is related to $|\mathbf{N}_i|$.

$$\varsigma_i = s_i(\varsigma_m + \varsigma_{s,i}) + (s_i - 1)\varsigma_d \quad (3)$$

The behaviors of robot in each state can be brief as follows. In the beginning, all the randomly placed robots start in the roaming state, and their expectation is randomly assigned in the range of $(1, n_0]$ where n_0 is the maximum initial value of $n_{e,i}$. Then, the robots randomly traverse in the arena to look for the desired cluster. Let $\widetilde{n}_{i,j}$ be an estimated cluster size of $R_{i,j}$ where $R_{i,j} \in \mathbf{N}_i$ and $\mathbf{E}_i$ are a set of cluster sizes estimated by neighbors of R_i, which is obtained via the intercommunication between robots. A cluster is said to be a desired cluster or expected cluster of R_i if $\exists \widetilde{n}_{i,j} \in \mathbf{E}_i | \widetilde{n}_{i,j} + 1 \geq n_{e,i}$ or $\widetilde{n}_i + 1 \geq n_{e,i}$ in case $\widetilde{n}_i$ is determined. As soon as the robot finds the desired cluster, the robot will shift to an aggregating state and move to this cluster. However, in the beginning, all robots in the swarm are in a roaming state, which means there are no aggregations in the arena. However, the expectations of robots are now under disappointment. Hence, their expectations have decreased over time. In case R_i does not meet any expected cluster and $n_{e,i} < 1$, each robot automatically shifts to an aggregating state since obviously $n_i = \widetilde{n}_i = 1$, which is satisfied given condition $\widetilde{n}_i + 1 \geq n_{e,i}$.

In an aggregating state, robots communicate with others to estimate the cluster size and transfer the estimated cluster size to their neighbors. According to a given idea, to create the biggest cluster, robots always have a desire to join bigger clusters. Hence, the robot expectation is now under a motivation effect. This effect raises robot expectations over time with a rate of ς_m and helps balance the exploration and exploitation of the aggregate processes.

However, if the expectations of robots constantly increase over time, a cluster is more likely to vanish. When an interior robot's expectation is large enough, they want to leave the cluster. However, due to collision avoidance, they may not have a chance to leave the cluster, and they will be stuck inside. Then, the disappointment effect takes place, which reduces the expectations of these robots. When the robot expectation drops below the estimated cluster size of their neighbors, they switch back to an aggregating state. This process is repeated over and over, causing their expectations to fluctuate around cluster size. Gradually over time, more and more robots that stay inside a cluster will fall into this situation. Hence, if the robots within a cluster boundary leave the cluster, then the cluster is more likely to vanish.

To avoid this situation, an influence effect is introduced. An influence effect reduces expectation over time based on the relative position of a robot with respect to the cluster. If the robot is located more inside the cluster, it will have less chance to leave and hence less incentive to leave. In a cluster, the relative location of each robot can be indicated by counting the number of its neighbors. Robots that stay in the periphery of a cluster have few neighbors and vice versa. Hence, based on number of neighbors, the change of expectation caused by the influence effect can be determined through Equation (4). Hence, if $|\mathbf{N}_i| = m$, i.e., neighbors completely surround R_i, then $\varsigma_{s,i} = -\varsigma_m$, which results in eliminating the motivation effect.

$$\varsigma_{s,i} = -\varsigma_m \frac{|\mathbf{N}_i|}{m} \quad (4)$$

After the swarm size is estimated and expectation is calculated, individuals can determine whether to stay or leave the aggregate. If the difference of expectation and swarm size is greater than a given threshold Δn_i, then a robot will switch back to its roaming state and leave an aggregate. A given threshold Δn_i should be large enough in order to give the robot enough time to actually leave the aggregate, or the robot will be stuck in an infinite loop of state transition. In order to increase the lifetime of a large cluster,

Δn_i should be positively correlated to n_i. In the proposed model, Δn_i can be calculated by employing Equation (5).

$$\Delta n_i = \sqrt{\widetilde{n}_i} + 3 \quad (5)$$

The pseudo-code in Algorithm 1 describes the whole basic aggregation process of R_i.

Algorithm 1: Aggregation algorithm. Basic pseudo-code for robot R_i used for the aggregation process. This algorithm uses two parallel threads: *main_thread* and *communication_thread*. This code below represents *main_thread* that uses the results of *communication_thread*

Inputs: ς_d, ς_m
Global variables: s_i, $\widetilde{n}_i$, $n_{e,i}$
Initialization: $s_i \leftarrow 0,\ \widetilde{n}_i \leftarrow 0,\ n_{e,i} \leftarrow random \in (1, n_0]$
While true **do**
  If $s_i = 0$ **then**
    Execute *roaming_motion*;
    $\mathbf{E}_i \leftarrow$ communication_thread
    If $\max_{\widetilde{n}_{i,j}}(\mathbf{E}_i) \geq n_{e,i} + 1$ **or** $n_{e,i} < 1$ **then** $s_i \leftarrow 1$
  Else
    Execute *aggregating_motion*;
    $\widetilde{n}_i \leftarrow$ communication_thread
    $\Delta n_i \leftarrow \sqrt{\widetilde{n}_i} + 3$
    If $n_{e,i} > n_i + \Delta n_i$ **then** $s_i \leftarrow 0$
  end
  $\varsigma_i \leftarrow s_i(\varsigma_m - \varsigma_{s,i}) + (s_i - 1)\varsigma_d$
  $n_{e,i} \leftarrow n_{e,i} + \varsigma_i$
end

---

4. Communication Mechanism

In the previous section, an IR-based communication mechanism was proposed to help robots perceive the size of clusters and exchange this knowledge among robots. Actually, the instantaneous cluster sizes are almost impossible to be determined by individuals due to their mobility and communication constraints. This information can only be estimated. To estimate the cluster size, analogy-measurable cues such as sound intensity or light intensity are aided to the swarm system [13,15]. However, these measurement systems are often saturated when the size of a cluster is large. In the absence of cues, a communication system with an appropriate information exchange strategy still can extract cluster size but with the presence of latency. According to infrared-based communication, there are a few estimation methods which can be applicable [21–23]. Nevertheless, these techniques require the communication of complex modulated messages, which do not optimize the communication system in terms of time.

Inspired by the slime mold aggregation process [8], wave-based approaches were introduced. This mechanism promises to reduce latency in robot networks. Varughese et al. introduced a vast range of behaviors of swarm robots by utilizing a communication mechanism called the wave-oriented swarm paradigm, which is inspired by fireflies and the slime mold [24]. There are other nature-inspired communication mechanism options available, such as pheromone communication [25,26] and bioluminescence [27]. However, the wave-oriented swarm paradigm offers a unique approach to the problem of aggregation, and has been shown to be effective in a wide range of behaviors [24]. However, this method requires the swarm robots to form a connected graph at the beginning, since the intercommunication between robots allows only single-bit information signal in each transmission, i.e., the content of message is not explicit. If a robot is not in a cluster but within a communication range, it should pay attention to a cluster for a while to estimate cluster size before deciding whether to join or proceed. This process can be redundantly repetitive, since robots cannot distinguish among many clusters. Hence, the communication mechanism should help robots have the capability to explicitly transfer their estimated cluster size to overcome this situation.

In this study, a communication mechanism is proposed that is inspired by the slime mold aggregation process but uses modulated messages instead of single-pulse information like in ordinary mechanism. Each transmission of robot is modulated onto the medium by using binary amplitude shift keying (B-ASK). Each transmission contains n+1 bits, including 1 start bit and n-message bits. Connectionless protocol is implemented to optimize the transfer time. The transmission is said to be successful if n + 1 bits are completely received. The overview of the proposed mechanism used to estimate the cluster size is outlined briefly as follows.

As soon as a robot switches to the aggregating state, it will begin the process of estimating the cluster size. During this process, the robot's communication system will go through three states: active, inactive and refractory. At the beginning, the robot is in an inactive state, in which it waits for a message from its neighbors. If any message is received and recognized, the robot will enter the active state. In this state, the robot broadcasts its currently estimated cluster size to its neighbors for a time interval T_{act}, which is the required time for message transmission. This state transition will trigger a series of transmissions throughout the entire cluster. To avoid a broadcast storm across the cluster, due to the signal bouncing back and forth between robots, instead of going back to an inactive state, robots will enter the refractory state. In the refractory state, the robot's communication system is disabled for a time interval T_{ref} after successfully transmitting its message. Then, the robot switches back to an inactive state and continues to wait for messages from its neighbors.

Moreover, in an inactive state, robots have a chance to switch to an active state with probability p_t to self-trigger a message that is equivalent to initializing a wave in a cluster. The self-triggered message of a robot causes all the remaining robots in the cluster who are in the inactive state to be triggered to broadcast their owned knowledge, resulting in a wave-like propagation of messages. This kind of wave have three main characteristics:

- A wave has only one source;
- A wave spreads away from the source;
- A wave is eliminated by colliding with refractory robots.

From the first two wave properties, it can be derived that a wave is capable of affecting all robots in the cluster but only once per robot. From there, the idea of estimating cluster size is given by counting the number of waves between the two adjacent self-triggered messages. By this idea, a robot should take the effect of $N-1$ waves before initializing a wave in a cluster of N robots. Hence, each individual can estimate cluster size by counting the number of messages relayed between its own self-triggered messages. However, the durations between two self-triggered messages $T_{p,i}$ has the form of a Poisson distribution, so the estimated value significantly fluctuates over time. For this reason, $\widetilde{n}_i$ is determined by taking the average value of m_b successfully estimated processes.

However, the wave may not propagate throughout the whole cluster due to suspension by refractory robots caused by other waves. In this way, the estimated cluster size is often smaller than the actual cluster size. This effect cannot be eliminated due to the strong randomness of the method. However, a select priorate p_t can reduce this effect in each individual. Furthermore, the estimated cluster sizes received from neighbors can be used to reduce the effect of a stochastic component of the proposed method over the cluster.

By employing the proposed mechanism, robots which are in an aggregating state can estimate the cluster size, and roaming robots can obtain the information when they are in the communication range with aggregated robots. The overall process is implemented to communicate threads that provide the estimated cluster size. The communication thread is described in the form of pseudo-code in the next section.

5. Navigation Algorithm

In roaming state, a simple finite state machine is implemented for robot motion to avoid collision based on measuring the reflective infrared signal. Let p_{thresh} be the proximity sensor threshold and $p_{i,front}$, $p_{i,left}$ and $p_{i,right}$ denote the measured value of reflective *IR* light pulse from the front, left and right sensors of R_i, respectively. All infrared sensors' measured values are positively correlative to distance to obstacles. In this way, the FSM implemented for the proposed robot platform used for roaming is described in Algorithm 2.

Algorithm 2: Roaming_motion.

Input: p_{thresh}
$p_{i,front}, p_{i,left}, p_{i,right} \leftarrow$ ***communication_thread***
If $p_{i,front} > p_{thresh}$ **then**
  If $p_{i,left} > p_{thresh}$ **then**
    If $p_{i,right} > p_{thresh}$ **then** ***Move forward***
    Else ***Turn left***
  Else
    If $p_{i,right} > p_{thresh}$ **then** ***Turn right***
    Else ***Move backward***
Else
  If $p_{i,left} < p_{i,right}$ **then** ***Turn right***
  Else ***Turn left***
End

Note that the process of measuring $p_{i,front}$, $p_{i,left}$ and $p_{i,right}$ would be performed if there is no received message for a while. The value of these measurements can be incorrect due to incoming infrared signals from other sources. However, since these measurements require very short time to acquire value, the fault probability is also very small.

During the roaming state, a robot observes many clusters and checks expectations to make a decision. Even if the robot has already determined the cluster to join, there are still other problems that need to be solved in order for the robot to successfully aggregate. The most major problem is navigating the robot to the desired cluster. In this study, the navigation algorithm that utilizes the proposed communication mechanism to help robots identify the direction from where most signals originate from, which will be referred to as the average origin of wave (AOW) [24], is applied. Each time robot in cluster receives a message, they will store the direction of the source in a list $\boldsymbol{\alpha}_\mathbf{i} = \{\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,k}\}$ where $k = \widetilde{n}_i$. $\alpha_{i,j}$, where $j \in \{1, 2, \ldots, k\}$ is defined as the estimated bearing angle with respect to the local frame of R_i.

$$\alpha_{i,j} = \text{atan2}\left(\sum_{l=1}^{m} p_{i,l} \sin(\beta_{i,l}), \sum_{l=1}^{m} p_{i,l} \cos(\beta_{i,l})\right) \tag{6}$$

where $p_{i,l}$ and $\beta_{i,l}$ are the peak of the signal measured from the received message on the lth sensor and heading of the lth sensor with respect to the local frame of R_i. Each time a robot triggers a wave, the estimated origin of the wave is determined by finding the mean value of $\boldsymbol{\alpha}_\mathbf{i}$, which is denoted as $\widetilde{\alpha}_i$.

$$\widetilde{\alpha}_i = \text{atan2}\left(\frac{1}{k}\sum_{l=1}^{k} \sin(\widetilde{\alpha}_{i,l}), \frac{1}{k}\sum_{l=1}^{k} \cos(\widetilde{\alpha}_{i,l})\right) \tag{7}$$

Due to the randomness of cycle length, the values of $\widetilde{\alpha}_i$ significantly fluctuate. Hence, the estimated AOW is obtained by the average list of value $\widetilde{\alpha}_i$ throughout m_c-cycles. However, this approach relies on parameters from the communication mechanism. The wave initiation probability p_t should be large enough for every agent to ping in a slot in which to keep the waves from colliding with each other as low as possible, therefore providing a more accurate estimation of the average origin of message. The proposed AOW method is implemented in the communication thread, which is described briefly in pseudo-code below (Algorithm 3).

Algorithm 3: Communication_thread. Pseudo-code used to operate communication system of R_i

Input: p_t, T_{act}, T_{ref}, m_b, m_c
Global variables: $state$, $\widetilde{n}_i$, list_$\widetilde{n}_i$, $n_{temp,i}$, $\boldsymbol{\alpha}_\mathbf{i}$, list_$\widetilde{\alpha}_i$, $\widetilde{a}_{temp,i}$
While true do
  While $s_i = 0$ **do Update E**$_i$, $p_{i,front}$, $p_{i,left}$, $p_{i,right}$
  $state \leftarrow inactive$, $buff_i \leftarrow \{0\}$, $n_{temp,i} \leftarrow 0$, $\widetilde{n}_i \leftarrow 0$
  While $s_i = 1$ **do**
    If $state = refractory$ **then**
      If $t_{ref} \leq 0$ **then** $state \leftarrow inactive$ **Else** $t_{ref} \leftarrow t_{ref} - \Delta t$
    If $state = active$ **then**
      If $t_{act} \leq 0$ **then** $state \leftarrow refractory$, $t_{ref} \leftarrow T_{ref}$ **Else** $t_{act} \leftarrow t_{act} - \Delta t$
    If $state = inactive$ **then**
      If ***receive message*** **then**
        Update $p_{i,1}, p_{i,2}, \ldots, p_{i,m}$
        Wait for ***message;***
        If ***message*** **is recognized then**
          $state \leftarrow active$, $t_{act} \leftarrow T_{act}$, $n_{temp,i} \leftarrow n_{temp,i} + 1$
          Append ***estimated bearing angle*** **to** $\boldsymbol{\alpha}_\mathbf{i}$
      Else if $p_t < random \in [0,1]$ **then**
        $state \leftarrow active$, $t_{act} \leftarrow T_{act}$
        If $|\text{list\_}\widetilde{n}_i| \geq m_b$ **then pop first element of** list_$\widetilde{n}_i$
    append $n_{temp,i}$ **to** list_$\widetilde{n}_i$
    $\widetilde{n}_i \leftarrow$ average value of list_$\widetilde{n}_i + 1$
    $\widetilde{a}_{temp,i} \leftarrow$ mean value of $\boldsymbol{\alpha}_i$
        If $|\text{list\_}\widetilde{\alpha}_i| \geq m_c$ **then pop first element of** list_$\widetilde{\alpha}_i$
    append $\widetilde{a}_{temp,i}$ **to** list_$\widetilde{\alpha}_i$
    $\widetilde{\alpha}_i \leftarrow$ mean value of list_$\widetilde{\alpha}_i$
      Else Update $p_{i,front}$, $p_{i,left}$, $p_{i,right}$
      End
    End
  End
End

According to $\widetilde{\alpha}_i$, a robot can approach the desired cluster. Note that $\widetilde{\alpha}_i$ is considered as the desired heading with respect to the local frame of R_i since the components that are used to calculate $\widetilde{\alpha}_i$ are also with respect to the local frame of R_i. Hence, $\widetilde{\alpha}_i$ can be directly used as the desired heading deviation. By using the proportional parameter κ, the angular speed ω_i can be determined by employing Equation (8).

$$\omega_i = \kappa\widetilde{\alpha}_i, \widetilde{\alpha}_i \in (-\pi, \pi] \tag{8}$$

By the naive approach, the aggregating motion is basically formed by a sequence of three motions (Algorithm 4). Firstly, the robot will turn with a rate of ω_i until the required heading is within the desired heading error tolerance $\widetilde{\alpha}_{thresh}$. Then, the robot will move forward until $p_{i,front}$ is less than some threshold. Finally, the robot will stop. However, because agents determine the desired heading from which they receive the most pings, the AOW rarely provides a geometrical center of the swarm [24], resulting in a loosely connected cluster. In order to create a tighter form (without obstacles, they usually create a circle form), after turning with a rate of ω_i, a robot will perform a roaming motion for a while. That leads to the robot having a greater chance of moving deeper inside the cluster instead of becoming stuck after approaching its neighbors.

Algorithm 4: Aggregating_motion

Input: κ
If $|\widetilde{\alpha}_i| > \widetilde{\alpha}_{thresh}$ **then** $\omega_i \leftarrow \kappa\widetilde{\alpha}_i$ **,** Turn with rate of ω_i
Else Perform ***roaming_motion*** for a while

6. Performance Evaluation

The performances of the proposed model are evaluated through tests conducted in both simulation and real experiments. All tests use the same robot platform, which is shown from a top and perspective view in Figure 1. The robot platform is cylindrical in shape with a radius r_R of 25 mm and height of 40 mm, respectively. The inter-wheel l_w and radius of each wheel r_w are 40 mm and 20 mm, respectively. The robot is equipped with six infrared modules with the order shown in Figure 1. With six infrared modules, the maximum number of neighbors of each robot obviously is six also. Without adding information, the default communication range r_c and default sensing range r_s of these infrared modules are approximately 180 mm and 20 mm, respectively. For convenience, the proximity sensor threshold p_{thresh} is set to a value which is equivalent to the sensing range r_s. Finally, the robot linear speed can reach 35 mm/s. However, due to the communication speed not being fast enough, hence, the maximum linear speed $V_{R,\max}$ is set to 20 mm/s according to the experiments.

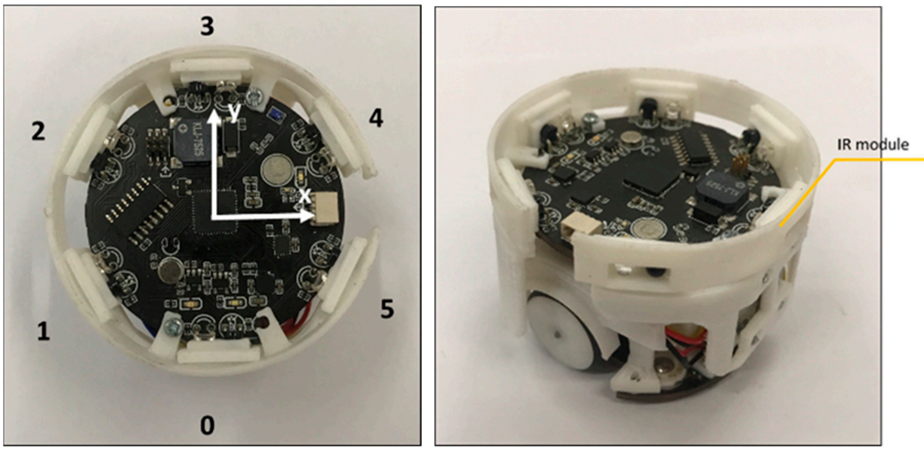

Figure 1. Robot platform used in this study. **Left**: The top view of the robot platform with IR-module order and local frame notations. **Right**: The side view of the robot platform.

To evaluate the proposed model, two performance evaluation metrics that were proposed in [28] and have been used in many previous studies [14,25,29] were employed in this study. They are the dispersion metric and the cluster metric. The dispersion metric evaluates the model by measuring the total distance between the individuals' position $\mathbf{P}_i$ and the centroid of swarms, which is defined as the average value of the robots' position $\frac{1}{N}\sum_{i=1}^{N}\mathbf{P}_i$. In this way, the aggregation process aims to minimize the dispersion of the swarm. However, since the distance between robots depends on the physical size r_R and negative feedback caused by proximity sensor threshold p_{thresh}, hence, the dispersion of the swarm is normalized by $\frac{1}{4(r_R+r_s)^2}$. Let q_d be the dispersion of swarm at time t. The q_d can be determined by employing Equation (9).

$$q_d = \frac{1}{4(r_R + r_s)^2}\sum_{i=1}^{N}\left\|\mathbf{P}_i - \frac{1}{N}\sum_{i=1}^{N}\mathbf{P}_i\right\|^2 \tag{9}$$

The second employed metric is a cluster metric, which is indicated by the ratio of the largest cluster size in the arena and the swarm size. If two robots are in an aggregating state and the distance between them is less than $r_R + r_c$, they are said to be two adjacent robots. Based on this condition, a cluster size can be determined by a recursive search algorithm implemented in observers. By sorting cluster sizes, the largest one is determined. Let q_c be the cluster metric of the swarm. The q_c value can be determined by employing Equation (10).

$$q_c = \frac{\underset{n_i}{\mathrm{Argmax}}(\mathbf{S})}{N} \tag{10}$$

However, the focus is on the steady state of the aggregation process in which a swarm robot forms a single aggregate cluster which is equivalent to $q_c = 1$. The aggregation process is not directly analyzed in this study based on these parameters. There are two evaluated parameters derived from the proposed metrics: time to complete aggregation process t_{cap} and dispersion of aggregation at steady-state q_{ds}. In some cases, when a single aggregate cluster cannot be formed, q_{ds} can be considered as the mean of q_d for the whole process. The t_{cap} is defined as the time in which 75% of robots in a swarm form a single cluster the first time or $q_c = 0.75$. The reason for choosing 75% is to balance the exploitation and exploration behavior. Due to the balance of exploitation and exploration behavior, the dynamic cluster is formed in which it is almost impossible to archive $q_c = 1$ in any case. Meanwhile, q_{ds} is applied to evaluate the dispersion of the swarm cluster and is defined as the mean of q_d when a single aggregate cluster is formed.

6.1. Simulation

In this test, robots are simulated in a 2D environment where each robot is considered as a disk with the same parameters described above. The default size of the test arena is 5000×5000 mm^2. The simulation was conducted with the default swarm size $N = 100$. The robot is programmed to have a 3% loss rate for receiving messages from its neighbor due to noise. The message transmission time is fixed to 10 ms.

Due to the strong randomness of experimental contexts and the communication mechanism, the results of the aggregation process based on the proposed metrics are significantly deviated. Hence, to investigate any context parameters or controller parameter, the final result should be based on the statistics of data of many experiments. Moreover, the limitation of computational time per experiment can take up to more than one hour to complete the aggregation process. To evaluate the proposed approach, only 20 runs of each experiment were performed. The statistics values of these evaluations are reported, which gives an estimated general performance regarding the introduced metrics. However, to accelerate the evaluations, evaluating t_{cap} and q_{ds} is performed in different conditions. When evaluating t_{cap}, the default condition is employed. Meanwhile, to evaluate q_{ds}, swarm robots will be distributed in a very small area in the arena to force them to form a connected communication network in the whole swarm. The default initial constant and parameters used in simulation are listed in Table 1. In this way, four evaluations are conducted: the effect of swarm density, effect of the communication range, presence of obstacle and flexibility of the proposed algorithm. By these default settings, if the swarm robot randomly distributed in the arena, then $q_{ds} > 380$. The single aggregate cluster should have $q_{ds} < 100$ and can be maintained with $q_c > 0.75$ over time.

Table 1. Constants and default parameters used in simulations and real experiments.

Parameter	Description	Value
ς_d	Rate of change of desire under disappointment effect	0.3/s
ς_m	Rate of change of desire under motivation effect	1/s
p_t	Wave initiation probability	0.004
T_{active}	Duration of active state	10 ms
$T_{refractory}$	Duration of refractory state	15 ms

Table 1. *Cont.*

Parameter	Description	Value
κ	Proportional parameter in heading control	0.2
p_{thresh}	Distance between robots in cluster	r_s
m_b	Estimated swarm size buffer length	10
m_c	Average origin of wave buffer length	5

6.1.1. Effect of Expectation Rate

The rate of change of expectation ς_i is under three effects, which are represented by two parameters: ς_d and ς_m, which can be considered as controller parameters. Hence, in this section, the parameters are analyzed to optimize the aggregation process in terms of time. The first parameter ς_d causes ς_i to decrease over time during the roaming state. Hence, ς_d should be selected appropriately to help robots have enough time to explore the arena but should not produce a redundant exploration time. If ς_d is too small or even zero, the convergence of the aggregation process emerges in a short time. However, the motivation effect keeps each robot's expectation growing over time during the aggregating state, causing robots to gradually leave the single aggregate cluster. That leads to the swarm disbanding, which is an undesirable phenomenon. On the other hand, if ς_d is too large, robot expectation drops rapidly. That leads each robot to easily return to the cluster it just left, and there will be many small clusters in the arena. However, if the roaming speed is high enough, it can remain in the same explored space. Hence, to optimize the process in terms of time, the test is performed with the roaming speed set to $V_{R,\max}$ when ς_d varies in a given set {0.001, 0.01, 0.1, 1.0, 10.0}. The results of this test are shown in Figure 2.

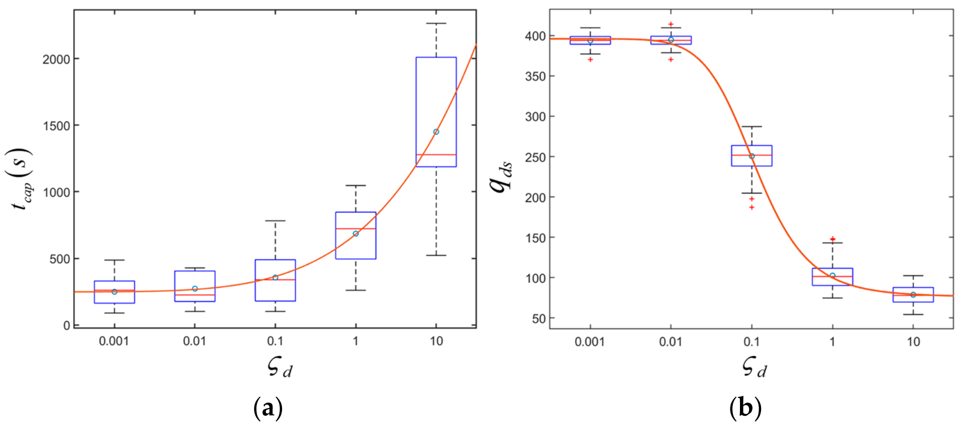

Figure 2. (**a**) The effect of ς_d on t_{cap} in simulation. (**b**) The effect of ς_d on q_{ds} in simulation.

According to the results in Figure 2, the effect of ς_d on the swarm aggregation behavior is predictable. The larger ς_d is, the slower the convergence rate of the aggregation behavior. Moreover, a single aggregate cluster can be formed in any range of ς_d but cannot be maintained in any range. Here, ς_d should have a minimum value of 1 to satisfy the single aggregate cluster conditions.

Based on the results shown in Figure 2, $\varsigma_d = 1$ is chosen to optimize the aggregation process in terms of time. However, selecting ς_m is more complicated. ς_m both increases and decreases robot expectation, depending on the number of neighbors. ς_m has an impact on the life cycle of the cluster. If ς_m is too small, the robot's behavior is more inclined toward exploitation and vice versa. If the aggregation process has not finished yet, large ς_m values accelerate the convergence. However, with large ς_m, the robot returns to leaving the cluster, forming a high dynamic cluster. That leads to suboptimal aggregation behavior.

In conclusion, high motivation results in fast convergence but unstable aggregation. According the results in Figure 3, ς_m should be chosen smaller than or at most equal to 1 to satisfy the q_{ds} condition. The simulation with $\varsigma_m > 3$ is difficult to converge in most cases, and t_{cap} cannot be recorded. This situation also occurs when $\varsigma_m < 3$; the convergence of the aggregation process takes a very long time, which cannot be performed in the given simulation hardware. In order to optimize the process in terms of time, $\varsigma_m = 1$ is chosen.

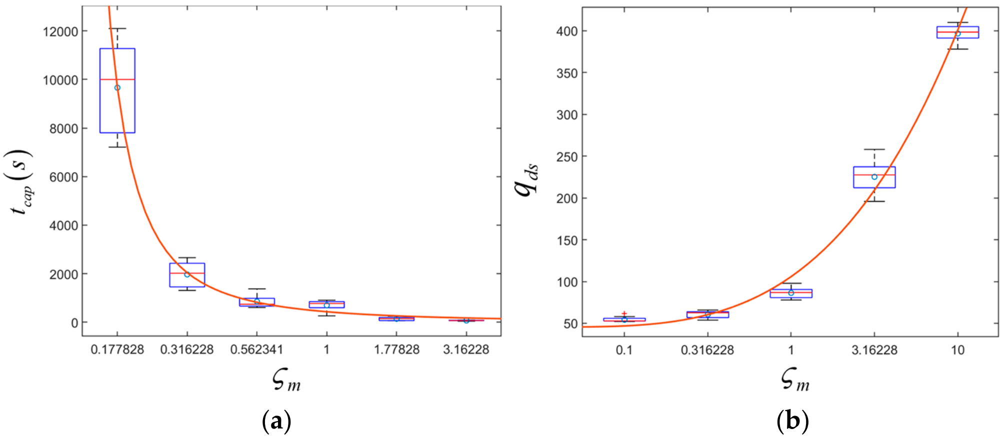

Figure 3. (**a**) The effect of ς_m on t_{cap} in simulation. (**b**) The effect of ς_m on q_{ds} in simulation.

Generally, ς_m is also positively correlated to ς_d in order to balance between exploration and exploitation behaviors. Hence, ς_m and ς_d should be investigated simultaneously. However, due to the limitation of computational time, ς_d is investigated while ς_m is set to 1.

6.1.2. Effect of Swarm Size

In this section, the aggregation performance is examined to see how it is affected by the swarm size. During this test, the swarm size is varied when the arena size is constant. Eight sets of experiments with $N \in \{25, 50, 75, 100, 125, 150, 175, 200\}$ are investigated. In this section, the aggregation behavior is evaluated using only t_{cap}, since the dispersion metric depends on the swarm size. The results of the test are shown in Figure 4.

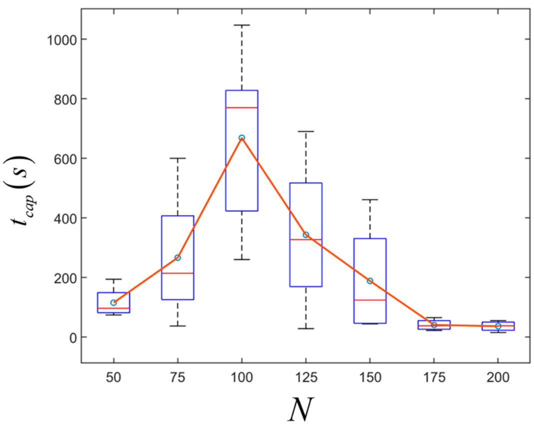

Figure 4. The effect of N on t_{cap} in simulation.

In a low-density swarm, the aggregation process seems to converge faster despite having the same communication range. However, the trend remains the same as long as $N \geq 50$. Due to high motivation, the robot cannot stay long in a small cluster, and hence, it can easily switch back to a roaming state. For swarms with a size lower than 50, the robot is unlikely to form large clusters, and so the aggregation process is almost impossible with a low robot density. Hence, the t_{cap} value of a swarm with a size of 25 cannot be recorded.

The interquartile range of results in the low-density swarm is also lower. Generally, the interquartile range of t_{cap} is caused by the local convergence in the aggregation process.

This phenomenon appears when many clusters of the same size emerge. The maximum value of t_{cap} is archived when the swarm forms two clusters of equal size. Moreover, this phenomenon will become more common as the size of the swarm increases. However, if the size is increased enough for the swarm to form a continuous network at the beginning, this phenomenon seems to be absent or occurs with very little frequency. As the results in Figure 4 show, the swarm which has a size of larger than 150 almost converges at the beginning of the aggregation process.

Thus, with the initially selected parameters, the aggregation process can be completed with a large swarm size of 50. For swarms with a size of less than 50, ς_m and ς_d should be small enough to help robots have enough time to explore in an arena with a low swarm size and increase the lifetime of the cluster.

To illustrate the quality of the aggregation method, the captures of the aggregation process of typical cases in a set of experiments with swarm sizes of 50, 100 and 200 are shown in Figure 5. In case of $N = 50$, the convergence of the process is very fast. However, due to the swarm size being small compared to ς_m, the aggregation of the swarm is unstable. This problem is solved when $N = 100$, but t_{cap} also increases rapidly.

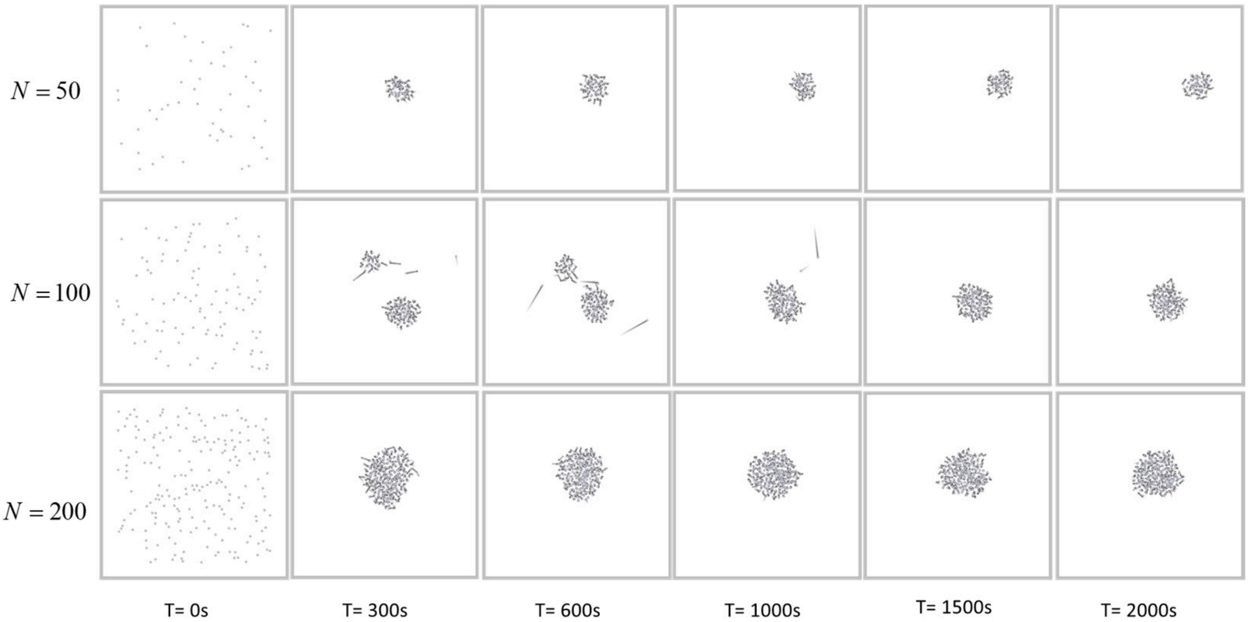

Figure 5. Snapshots of simulated aggregation processes with swarm sizes of 50, 100 and 200, respectively.

6.1.3. Effect of Communication Range

In this section, the effect of communication range r_c on the aggregation process is evaluated. This factor has a great influence on capability of the proposed model. By increasing r_c, a single aggregate cluster can be archived in a swarm with a size of less than 50. By the default settings, a communication range r_c varies in a set {175, 263, 350, 438, 525}, and t_{cap} values are evaluated. The results of this test are recorded in Figure 6.

According to the result, the effect of r_c on the aggregation process follows the same trend as that of swarm size N. For short transmission distances, small clusters will have less chance of contacting roaming robots than large clusters. So, in this case, the large clusters grow very quickly, and the process quickly converges. As the transmission distance increases, the probability of the robot participating in small and large clusters also gradually becomes equal, so the possibility of local convergence is also more likely. This leads to an increase in the interquartile range of t_{cap}. However, when the propagation distance is increased to a large enough value that the swarm can form a continuous network involving most of the robots at the beginning, both the mean of t_{cap} and its deviation decrease rapidly.

As the results in Figure 6 show, with the given arena size, if $r_c \geq 525$, a single aggregate cluster can be formed immediately after the process starts.

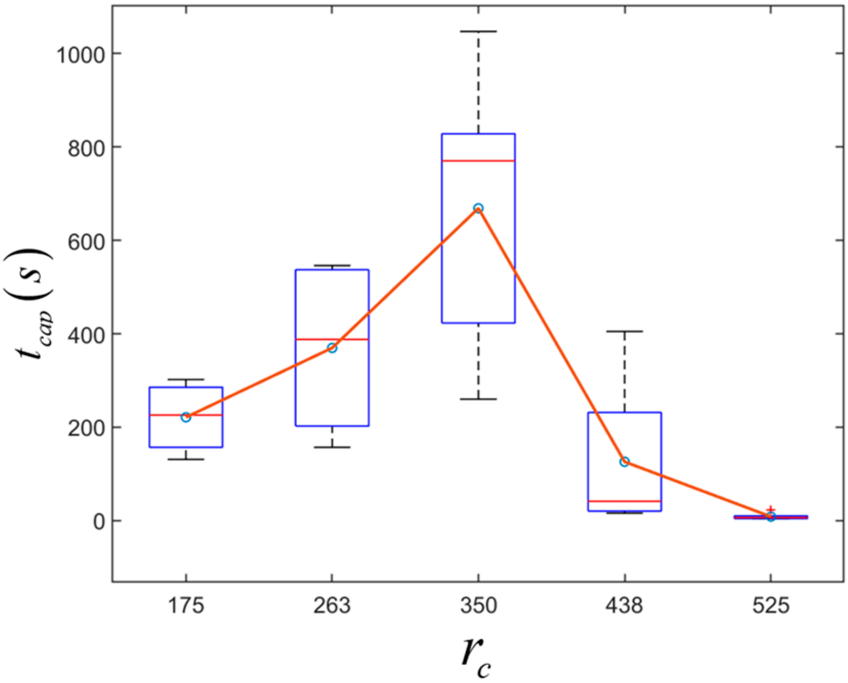

Figure 6. The effect of r_c on t_{cap} in simulation.

6.2. Experiments

The simulation results are reviewed in this section, which are obtained by conducting experiments with robots, as shown in Figure 1. The aggregation studies are carried out with 16 robots on a $120 * 240$ mm^2 white flat arena. The expectation rates obtained from the simulation cannot be applied in the case of low swarm sizes, as described in the effect of the swarm size section. Additionally, the arena size limitation has a significant impact on the method's performance. As a result, several tests must be carried out in order to achieve adequate q_c and q_d that adapt to small swarm sizes before an evaluation can be made.

The systems utilized to track and observe robot information are used to derive values of q_c and q_d in these tests. These devices are just used to monitor and track robot states throughout the aggregation process and have no impact on the swarm's performance. One camera, hung upside down 200 mm from the arena, and a server make up the tracking system. The camera connects wirelessly and sends streaming photos to the server, which processes the images to track the movement of swarm robots. Furthermore, each robot is equipped with a Wi-Fi module that is built into the robot's main board. These Wi-Fi modules are used to send the state of the robot and its neighbors to the server in order to obtain the q_c and q_d values.

To test the effect of ς_d to q_c and q_d, five sets of experiments which consist of $\varsigma_d = \{0.001, 0.01, 0.1, 1, 10\}$ are performed with 10 trials per set. The results shown in Figure 7 have trends similar to the one in simulation (Figure 2). However, because the arena and swarm sizes are so small in comparison to the simulation, the time it takes for robots to explore the arena is cut in half, and local convergence is infrequent, resulting in lower t_{cap}. When $\varsigma_d < 0.1$, t_{cap} is only about 70s, but the single aggregation cluster has quickly vanished. To optimize the aggregation process, $\varsigma_d = 0.1$ is selected.

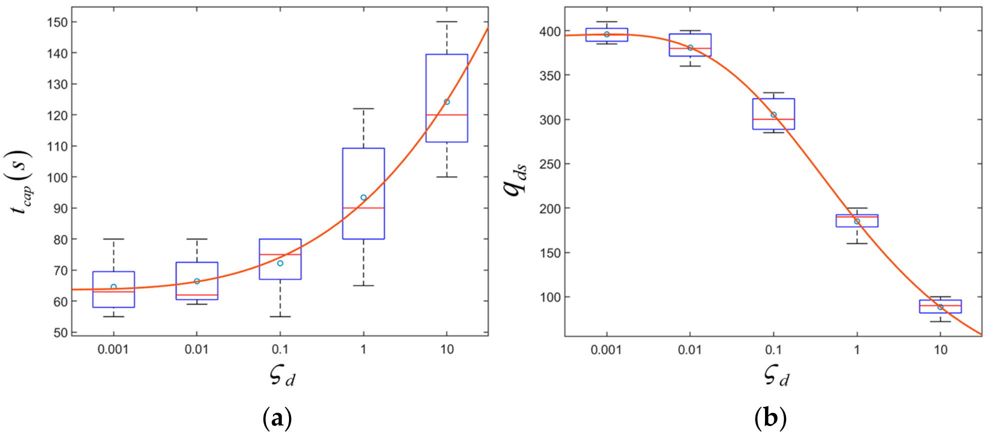

Figure 7. (**a**) The effect of ς_d on t_{cap} in the real experiment. (**b**) The effect of ς_d on q_{ds} in the real experiment.

The influence of ς_m is also investigated with values of 0.001, 0.01, 0.1, 1 and 10, with the results divided into two cases. In the first one, when $\varsigma_m \leq 1$, the relationship between ς_m and t_{cap} in experiments is virtually constant with values comparable to those shown in Figure 7a with corresponding ς_d. This is due to the fact that the size of the arena and the size of the swarm in tests are unlikely to generate local optimal results. That means that throughout their exploration, swarm robots are more likely to identify a single aggregate cluster rather than construct a single member cluster. As a result, with $\varsigma_m \leq 1$, the robot has enough time to participate in the single aggregation cluster before members of this cluster leave due to increasing expectations. Meanwhile, when $\varsigma_m = 10$, the expectation growth of members in the single aggregation cluster is very large compared to the default value of ς_d. Hence, the swarm will always rapidly fill in their expectations and leave a cluster, and the swarm will spend the majority of its time roaming. As a result, t_{cap} is unable to be recorded and q_{ds} grows exceedingly huge. The relationship between ς_m and q_{ds} may be observed in Figure 8. Furthermore, $\varsigma_m \leq 0.1$ provides a high-quality aggregation cluster with $q_{ds} < 80$. Based on these observations, ς_d can be an arbitrary chosen value less than 0.1. However, to avoid the case of local optimization, where many clusters of the same size are formed, ς_d should be large enough to promote the emergence of a single aggregation cluster. Hence, $\varsigma_m = 0.1$ is chosen to optimize the aggregation process in both a typical case and a local optimization case.

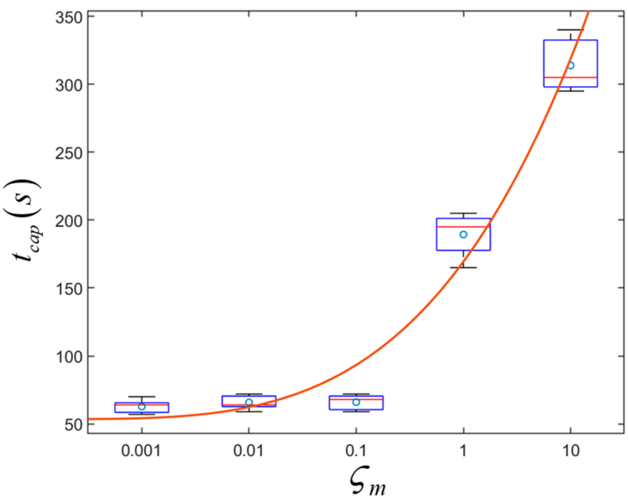

Figure 8. The effect of ς_m on q_{ds} in a real experiment.

Figure 9 shows captures of a typical aggregation process in a real scenario with $\varsigma_m = 0.1$ and $\varsigma_d = 0.1$. This process has $t_{cap} = 75s$ and $q_{ds} \approx 60$, which meet the requirements of the aggregation process results. In the first capture, when $T = 0$, the robots used in the process were manually distributed in the arena with $q_d > 400$. After receiving the aggregation instruction from the server, all of the robots in the swarm undertake the aggregate task at the same time. In the first 20 s, robots generate a slew of small, disjointed clusters. However, due to the tiny arena size, these clusters immediately linked with one another, resulting in a single aggregation cluster after around 60 s. The process continues until the collection stabilizes, and then, q_{ds} is recorded.

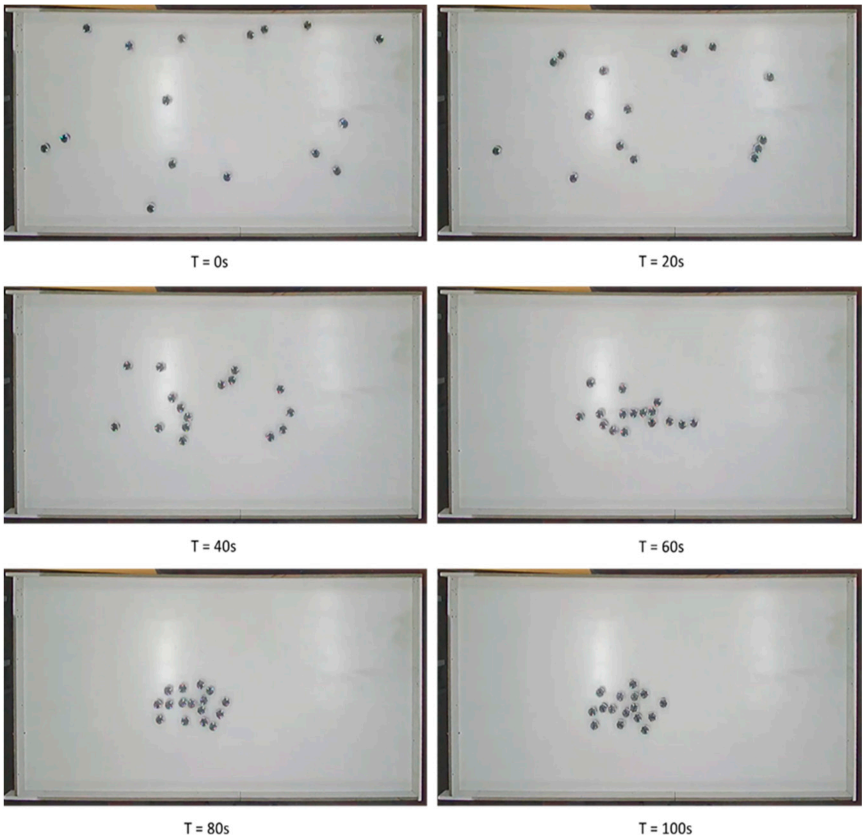

Figure 9. The snapshots of the aggregation process with $\varsigma_m = 0.1$ and $\varsigma_d = 1$ in a real experiment.

7. Conclusions

In this study, the microscopic model for the aggregation behavior of swarm robots has been proposed with the presence of two main control parameters: subjective expectation and estimated cluster size. The proposed model consists of the communication mechanism which helps robots estimate the cluster size and navigate them to the desired cluster. All of the test is based on the results of two factors t_{cap} and q_{ds} that derived from the cluster metric and dispersion metric, respectively. Model parameters have been chosen by employing simple statistical methods and perspicuous analysis. Some scenario parameters are also investigated to evaluate the performance of aggregation behavior when the proposed model is implemented.

The introduced model is scalable, but it still has some constraints due to the constant control parameters. According to the given value of the control parameters in the previous section, the proposed model cannot help a swarm with size of below 50 aggregate into a single aggregate cluster. However, a single aggregate cluster can be truly formed by employing this model with reasonably selected parameters.

The proposed model can be improved to remove its limitation, but it is out of the scope of this study. In the future, the proposed model can be extended as follows:

- Control parameters ς_d and ς_m can be varying over time based on a history log of the previous estimated cluster sizes and the ratio of roaming time and aggregating time. In this way, proposed model can adapt to a low-density swarm size;
- By analyzing the effect of wave initiation probability p_t on the cluster size estimating performance and the history log of the previous estimated cluster size, p_t can be controlled by an individual instead of initializing it from the beginning.
- Previously studied models would be applied in experiments to evaluate and compare their performance to the one of the proposed models. The robustness of the model will be further assessed in more complex and realistic environments.

Author Contributions: Conceptualization, N.T.T.; methodology, N.T.T.; software, P.G.L.; validation, P.G.L.; formal analysis, P.G.L.; resources, N.T.T.; data curation, P.G.L.; writing—original draft preparation, P.G.L.; writing—review and editing, N.TT.; visualization, P.G.L.; supervision, N.T.T.; project administration, N.T.T.; funding acquisition, N.T.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the University of Economics Ho Chi Minh City, Vietnam. This work belongs to the project grant No: B2022-SPK-02 funded by Ministry of Education and Training, and hosted by Ho Chi Minh City University of Technology and Education, Vietnam.

Data Availability Statement: Not applicable.

Acknowledgments: This research is supported by the University of Economics Ho Chi Minh City, Vietnam, and received funding from Ministry of Education and Training of Vietnam. The authors would also like to thank the Ho Chi Minh City University of Technology and Education in Vietnam for providing study facilities.

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

References

1. Nayyar, A.; Nguyen, N.G. Introduction to swarm intelligence. In *Advances in Swarm Intelligence for Optimizing Problems in Computer Science*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018; pp. 53–78.
2. Hamann, H. *Swarm Robotics: A Formal Approach*; Springer: Berlin/Heidelberg, Germany, 2018.
3. Nedjah, N.; Junior, L.S. Review of methodologies and tasks in swarm robotics towards standardization. *Swarm Evol. Comput.* **2019**, *50*, 100565. [CrossRef]
4. Schmickl, T.; Hamann, H. BEECLUST: A swarm algorithm derived from honeybees. In *Bio-Inspired Computing and Communication Networks*; CRC Press: Boca Raton, FL, USA, 2011; pp. 95–137.
5. Cabrera, B.J.; Rust, M.K. Behavioral responses to light and thermal gradients by the western drywood termite (Isoptera: Kalotermitidae). *Environ. Entomol.* **1996**, *25*, 436–445. [CrossRef]
6. IsHII, S.; Kuwahara, Y. An aggregation pheromone of the German cockroach *Blattella germanica* L.(Orthoptera: Blattellidae): I. Site of the pheromone production. *Appl. Entomol. Zool.* **1967**, *2*, 203–217. [CrossRef]
7. Ancel, A.; Gilbert, C.; Poulin, N.; Beaulieu, M.; Thierry, B. New insights into the huddling dynamics of emperor penguins. *Anim. Behav.* **2015**, *110*, 91–98. [CrossRef]
8. Newell, P.C. How cells communicate: The system used by slime moulds. *Endeavour* **1977**, *1*, 63–68. [CrossRef] [PubMed]
9. Shlyakhov, N.E.; Vatamaniuk, I.V.; Ronzhin, A.L. Survey of methods and algorithms of robot swarm aggregation. *J. Phys. Conf. Ser.* **2017**, *803*, 012146. [CrossRef]
10. Melhuish, C.; Holland, O.; Hoddell, S. Convoying: Using chorusing to form travelling groups of minimal agents. *Robot. Auton. Syst.* **1999**, *28*, 207–216. [CrossRef]
11. Khaldi, B.; Harrou, F.; Cherif, F.; Sun, Y. Self-organization in aggregating robot swarms: A DW-KNN topological approach. *Biosystems* **2018**, *165*, 106–121. [CrossRef] [PubMed]
12. Khaldi, B.; Harrou, F.; Cherif, F.; Sun, Y. Improving robots swarm aggregation performance through the Minkowski distance function. In Proceedings of the 2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE), Barcelona, Spain, 12–15 February 2020.
13. Soysal, O.; Sahin, E. Probabilistic aggregation strategies in swarm robotic systems. In Proceedings of the 2005 IEEE Swarm Intelligence Symposium, Pasadena, CA, USA, 8–10 June 2005.
14. Katada, Y. Evolutionary design method of probabilistic finite state machine for swarm robots aggregation. *Artif. Life Robot.* **2018**, *23*, 600–608. [CrossRef]
15. Soysal, O.; Bahçeci, E.; Şahin, E. Aggregation in swarm robotic systems: Evolution and probabilistic control. *Turk. J. Electr. Eng. Comput. Sci.* **2007**, *15*, 199–225.

16. Bayindir, L.; Sahin, E. Modeling self-organized aggregation in swarm robotic systems. In Proceedings of the 2009 IEEE Swarm Intelligence Symposium, Nashville, TN, USA, 30 March–2 April 2009.
17. Soysal, O.; Şahin, E. A macroscopic model for self-organized aggregation in swarm robotic systems. In *International Workshop on Swarm Robotics*; Springer: Berlin/Heidelberg, Germany, 2006.
18. Jeanson, R.; Rivault, C.; Deneubourg, J.L.; Blanco, S.; Fournier, R.; Jost, C.; Theraulaz, G. Self-organized aggregation in cockroaches. *Anim. Behav.* **2005**, *69*, 169–180. [CrossRef]
19. Trianni, V.; Groß, R.; Labella, T.H.; Şahin, E.; Dorigo, M. Evolving aggregation behaviors in a swarm of robots. In Proceedings of the Advances in Artificial Life: 7th European Conference, ECAL 2003, Dortmund, Germany, 14–17 September 2003; Springer: Berlin/Heidelberg, Germany, 2003.
20. Ferrante, E.; Turgut, A.E.; Mathews, N.; Birattari, M.; Dorigo, M. Flocking in stationary and non-stationary environments: A novel communication strategy for heading alignment. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Krakov, Poland, 11–15 September 2010; Springer: Berlin/Heidelberg, Germany, 2010.
21. Wang, H.; Rubenstein, M. A Fast, Accurate, and Scalable Probabilistic Sample-Based Approach for Counting Swarm Size. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020.
22. Neumann, N.G. Two Algorithms for Leader Election and Network Size Estimation in Mobile ad hoc Networks. Ph.D. Thesis, Texas A&M University, College Station, TX, USA, 2005.
23. Ding, H.; Hamann, H. Sorting in swarm robots using communication-based cluster size estimation. In Proceedings of the Swarm Intelligence: 9th International Conference, ANTS 2014, Brussels, Belgium, 10–12 September 2014; Springer: Cham, Switzerland, 2014.
24. Varughese, J.C.; Hornischer, H.; Zahadat, P.; Thenius, R.; Wotawa, F.; Schmickl, T. A swarm design paradigm unifying swarm behaviors using minimalistic communication. *Bioinspiration Biomim.* **2020**, *15*, 036005. [CrossRef] [PubMed]
25. Purnamadjaja, A.H.; Iskandar, J.; Russell, R.A. Pheromone communication simulation for mobile robots using java 3d. In Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007), Melbourne, VIC, Australia, 11–13 July 2007.
26. Meng, Y.; Kazeem, O.; Muller, J.C. A hybrid ACO/PSO control algorithm for distributed swarm robots. In Proceedings of the 2007 IEEE Swarm Intelligence Symposium, Honolulu, HI, USA, 1–5 April 2007.
27. Sharma, A.; Lakhwani, K. Implementation of Garbage Collection and Recycling Problem Using an Optimized Glowworm Swarm Optimization Algorithm. *Think India J.* **2019**, *22*, 188–213.
28. Gauci, M.; Chen, J.; Li, W.; Dodd, T.J.; Groß, R. Self-organized aggregation without computation. *Int. J. Robot. Res.* **2014**, *33*, 1145–1161. [CrossRef]
29. Mısır, O.; Gökrem, L.; Can, M.S. Fuzzy-based self organizing aggregation method for swarm robots. *Biosystems* **2020**, *196*, 104187. [CrossRef]