

Article

Collective Cognition on Global Density in Dynamic Swarm

Phan Gia Luan and Nguyen Truong Thinh * 

Institute of Intelligent and Interactive Technologies, University of Economics Ho Chi Minh City,
Ho Chi Minh City 700000, Vietnam; luanpg@ueh.edu.vn

* Correspondence: thinhnt@ueh.edu.vn

Abstract: Swarm density plays a key role in the performance of a robot swarm, which can be averagely measured by swarm size and the area of a workspace. In some scenarios, the swarm workspace may not be fully or partially observable, or the swarm size may decrease over time due to out-of-battery or faulty individuals during operation. This can result in the average swarm density over the whole workspace being unable to be measured or changed in real-time. The swarm performance may not be optimal due to unknown swarm density. If the swarm density is too low, inter-robot communication will rarely be established, and robot swarm cooperation will not be effective. Meanwhile, a densely-packed swarm compels robots to permanently solve collision avoidance issues rather than performing the main task. To address this issue, in this work, the distributed algorithm for collective cognition on the average global density is proposed. The main idea of the proposed algorithm is to help the swarm make a collective decision on whether the current global density is larger, smaller or approximately equal to the desired density. During the estimation process, the swarm size adjustment is acceptable for the proposed method in order to reach the desired swarm density.

Keywords: swarm robots; collective cognition; hard-core point process; collective decision making



Citation: Gia Luan, P.; Thinh, N.T. Collective Cognition on Global Density in Dynamic Swarm. *Sensors* **2023**, *23*, 4648. <https://doi.org/10.3390/s23104648>

Academic Editors: Ankit A. Ravankar, Jose Victorio Salazar Luces and Abhijeet Ravankar

Received: 11 April 2023

Revised: 1 May 2023

Accepted: 8 May 2023

Published: 11 May 2023



Copyright: © 2023 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

Swarm robotics is a sub-field of multi-agent robotics that employs distributed controllers, where inter-robot local communication is the key component. Due to the use of distributed controllers in a self-organized manner, swarm robot systems are expected to have three main advantages: flexibility, scalability and robustness. However, to design self-organized mechanisms is challenging, as desired behavior at the macroscopic level must be accomplished based on a set of behaviors programmed in each individual robot. Many task-based controllers have been designed for self-organized swarm robot systems such as collective transporting [1–3], collective foraging [4,5], collective decision-making [6,7] and collective exploration [8]. These controllers primarily focus on the scalability of swarm systems, in which changes to the swarm size within a reasonable range do not dramatically affect the performance of the swarm robots.

The average performance of a robot swarm which uses previous controllers actually depends on the swarm density, and it only depends on the swarm size if the operating area is kept constant. The analyses in previous works [1,3,9] has shown that the performance of swarms can be optimized by determining a suitable swarm density. In high-density environments, robots are compelled to put in more effort and attention to address inter-robot collision avoidance tasks rather than concentrating on their core responsibility, which lowers the swarm's overall performance. If the robot density is raised much further, performance will likely decrease to negligible levels due to interference halting almost all motion. However, when there are too few robots present in a large area, robot collaborations are noticeably ineffective, and the same outcome could be observed. In some cases, low-density swarms can have more impact than high density situations, as observed in self-organized aggregation processes [10,11]. Hence, in order to optimize the performance of the swarm, the swarm density must be predetermined.

Even if the theoretical optimal swarm density is determined, the performance of the swarm may not reach its maximum value due to unmeasurable swarm density in real-world situations. Generally, swarm density is defined as the number of robots occupying a unit area [9]. However, in many cases, swarm density is unknown due to the lack of knowledge about the workspace. Even if the workspace is fully determined, the varying swarm size is also a significant issue in measuring swarm density. Faulty or out-of-battery robots are factors that make swarm density vary with respect to time. With large swarm sizes, more robots may break, get lost or may intentionally be removed, causing the swarm density to decrease at runtime. Real-time changes in density have an impact on the swarm performance, and if there is no compatible swarm size change, the performance of the swarm will be greatly reduced.

There are many previous studies that can help swarm robots collectively determine swarm density. Sheng Zhao et al., proposed the Smoothed Particle Hydrodynamic (SPH) model which uses the weighted sum of distances to its nearby robots within a certain range as the robot density [12]. This information is locally utilized by a robot which does not imply any collective information. Inter-robot collisions are a simple and effective way to derive swarm density. Many social insect colonies, according to entomological studies, can monitor density through inter-individual collision in order to manage interference with decentralized mechanisms [13,14]. Siddharth, et al., who were inspired by these species, created a model that was used to compute the swarm density using the frequency of inter-robot interactions [15]. The approach has been used for certain behaviors such as group gathering [16] and task allocation [17].

This work introduces an approach that enables swarms of robots to collectively determine whether their current swarm density is greater, smaller or comparable to a given desired swarm density. The fundamental idea of the approach is to enable the robots to estimate the swarm density collectively by evaluating how frequently they encounter their neighbors, which allows the swarm as a whole to make decisions. The relationship between encounter rate and swarm density is modeled based on the nearest neighbor distance distribution in the hard-core point process introduced in [18]. To ensure the swarm has shared information to make the same decision, the simple distributed average consensus strategy is proposed. The proposed process addresses four main challenges of designing collective swarm density cognition: (1) swarm density is a global feature that can only be efficiently detected collectively by applying the distributed average consensus technique; (2) we face a tradeoff between filtering noise and reacting quickly to changes; (3) the swarm density estimator only depends on swarm size and environment area and does not require knowledge of additional parameters such as environment shape, robot speed or robot kinematics, as long as the robot motion is ergodic; (4) the method should be able to assess changes in swarm density over time to reach optimal density. The proposed method is verified by numerous simulations and is also deployed on a team of real robots. By analyzing the results, the collective swarm density cognition method has an applicable response to the change of swarm density with an acceptable error when a significant number of changes occurs in the swarm density.

2. Task Formulation

Consider a finite swarm of identical robots, denoted by $\mathbf{S}(t) = \{R_1, R_2, \dots\}$, which are randomly distributed in a compact, connected planar domain $D \subset \mathbb{R}^2$ where $R_i \in \mathbf{S}(t)$ denotes the i^{th} robot. The swarm size can be varied during the operation of the swarm, and it is denoted as $N(t)$. Let r_r denote the robot footprints' actual physical radius. Each robot is equipped with sensors that helps it perceive the presence of obstacles and other robots with a detecting skirt of the total diameter $2r_s$ around it. The same range is applied to the robots' communication system, which enables robots to exchange explicit information with their neighbors. Additionally, it is expected that any D can fit all $N(t), \forall t \geq 0$ robots without

encountering any challenging geometric packing issues. Let $\lambda(t)$ denote the population density of swarm $\mathbf{S}(t)$ over domain D which can be defined as Equation (1).

$$\lambda(t) = \frac{N(t)}{|D|} \quad (1)$$

where $|\cdot|$ denotes the set area operator. Following the preliminary introduction, it is assumed that the swarm operates in a fully or partially unknown domain, and the size of the swarm may vary. Hence, λ is a time-varying variable that can only be estimated. However, in this study, we do not attempt to propose the method used to estimate λ . The key issue we will tackle is how to answer the question of whether a swarm's actual swarm density surpasses, falls short of or is roughly equal to a given desired swarm density, λ_d , by introducing a distributed algorithm that aids swarms. Furthermore, we assume that robots in a swarm do not have any prior knowledge of the size of the workspace, and the decision of robots in the swarm is based solely on the information related to the inter-robot encounter rate.

However, the rate of encounters experienced by the robots will depend on their motion in the domain. We initially restrict the motion characteristics of the robots in their workspace in order to make specific claims about inter-robot encounters. Let $\mathbf{x}_i(t) \in D$ be the position of i^{th} robot at time t . Then, for any given subset $U \subseteq D$, the average time spent by the robot in that set should be equal to the ratio defined in Equation (2). If the Equation (2) is held in the swarm, then we can state that the motion pattern of robots in the swarm is ergodic.

$$\lim_{T \rightarrow \infty} \int_{t=0}^T \frac{1}{T} \sum_{i=1}^N \mathbf{1}(\mathbf{x}_i(t) \in U) dt = \frac{|U|}{|D|} \quad (2)$$

The developed encounter model is applicable as long as the ergodicity assumptions are satisfied, which enables us to make generalizations about the types of motion patterns that arise from the task's execution. The consistently ergodic trajectories of the robots imply that their density over the domain D at any given instant is also uniform. In our study, the motion of robots in a swarm can be summarized by the following steps: (1) a robot moves forward; (2) if a robot comes close to the workspace boundaries or collides with other robots, it turns in a randomly selected direction and reverts back to step (1). As long as the ergodicity assumption is still valid, the other motion models can be used even when robots are performing other tasks while executing the suggested mechanism.

3. Encounter Rate

In the first section, we stated that the robots performing the collective task in a distributed manner are equipped with sensors that can help the robot detect the presence of its neighbor within the range of r_s . We first denote that $\mathbf{F}_i(t) \subset \mathbf{S}(t)$ is the set of robots in range of r_s within R_i , called neighbor set of R_i , in which if $R_j \in \mathbf{F}_i(t)$, then $\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| < r_r + r_s$ must be satisfied. Hence, we define a robot as having an "encounter" when a neighbor set of given robots is not empty, $\mathbf{F}_i(t) \neq \emptyset$. Based on the given motion pattern of robots in the swarm, upon colliding with another robot, each robot executes a collision avoidance maneuver to prevent the collision and continue along its trajectory. Consequently, the presence of the complement state of "encounter" is called "disjoin" where no robots are present in the sensory range of a given robot or $\mathbf{F}_i(t) = \emptyset$.

We first need to construct a model for how frequently encounters happen among the robots, or the encounter rate, in order to be able to use encounter information to estimate the swarm density. The robot's encounter rate can be calculated from its time in the encounter and disjoin states. However, the durations indicating how long a robot stays in an encounter or disjoin state are stochastic variables due to the randomness of the motion model. Obviously, these random variables are dependent on swarm density as well as on different distribution parameters with regard to how quickly the task of

avoiding collisions may be solved and the robot speed. These factors can be modeled, but they can also complicate the estimation process and reduce its generality. Firstly, the collision avoidance strategy and time spent on different locomotion methods—for example, differential-drive robots and omnidirectional robots that share physical parameters—are significantly different. Furthermore, although robot density is maintained as constant, the ratio of encounter duration to total process duration varies during the process due to the varying speed of robots. To ensure the generalizability of the proposed method, the measurement of travelled distance in which these states are occupied are implemented instead of measuring in terms of time. From now on, s will be used as the discrete index parameter indicating the travelled distance of a given robot. We should note that the travelling distances among robots in a swarm are not the same with respect to time.

Hence, now, the states of each robot in the swarm can be modelled as a continuous stochastic chain $\{X(s) : s \geq 0\}$ defined by index parameter s and state space $\{0, 1\}$ where 0 denotes the “disjoin state” and 1 denotes the “encounter state”. Due to the exponential decrease in probability of travelling distance for a single encounter event and single disjoin event, and their independence from each other, this process can be considered as a continuous Markov process with an index parameter in terms of the travelled distance. The travelled distance on the single encounter event and single disjoin event can be modelled as the exponential distribution with rate parameters λ_e and λ_d , respectively. Let π_e be the average encounter rate of robots in the swarm, and $\pi_e(t)$ can be defined as a stationary probability of the encounter state at time t .

$$\pi_e(t) = \frac{\lambda_d(t)}{\lambda_e(t) + \lambda_d(t)} \quad (3)$$

However, because λ_e and λ_d are unknown variables and depend on the number of robots in the swarm, it is possible to estimate π_e solely by using estimators for λ_e and λ_d . Let $\hat{\pi}_{e,i}$ be the encounter rate estimator of R_i , which can be defined as Equation (4) by applying the moment method.

$$\hat{\pi}_{e,i}(t) = \frac{\hat{\lambda}_{d,i}(t)}{\hat{\lambda}_{e,i}(t) + \hat{\lambda}_{d,i}(t)} \quad (4)$$

where $\hat{\lambda}_{e,i}$ and $\hat{\lambda}_{d,i}$ are the estimators of the inverse of the mean of travelled distance on the encounter state and disjoin state of the given continuous Markov process estimated by R_i . The estimation procedure can be carried out by using the maximum likelihood estimation method on two sets of random samples. The first set consists of the length of travelled distance a robot spends in the encounter state, while the second set consists of the sum of its travelled distance spent in both states. k represents the size of each set. The value of k can be viewed as the number of cycles of the Markov process that a particular robot has gone through. However, directly determining the value of $\hat{\pi}_{e,i}$ can be very complicated. To ease this problem, the auxiliary estimator z_i is employed which is defined in Equation (5).

$$z_i(t) = \frac{\sum_{j=1}^{k_i} s_{e,i,j}}{\sum_{j=1}^{k_i} (s_{e,i,j} + s_{d,i,j})} \quad (5)$$

where $s_{e,i,j}$ and $s_{d,i,j}$ are the travelled distance of the j^{th} encounter event and j^{th} disjoin event measured by R_i . Both $s_{e,i,j}$ and $s_{d,i,j}$ are random variables which have a probability density function represented by an exponential model. Hence, if the change of average global density is slow enough, then we can approximate that $\sum_{j=1}^{k_i} s_{e,i,j} \sim \Gamma(k, k\lambda_e)$ and $\sum_{j=1}^{k_i} (s_{e,i,j} + s_{d,i,j}) \sim \Gamma(k, k\lambda_c)$ where $\Gamma(a, b)$ represents the gamma distribution parameter-

ized in term of shape parameter a and an inverse scale parameter b ; λ_c can be defined as Equation (6).

$$\lambda_c = \frac{\lambda_d + \lambda_e}{\lambda_d \lambda_e} \quad (6)$$

According to [19], the probability distribution of $z_i(t)$ can be approximate to the generic beta model of the second kind, i.e., $z_i \sim GB2\left(k, k, 1, \frac{\lambda_c}{\lambda_e}\right)$ where the first three parameters are shape parameters and λ_c/λ_e is the scale parameter. The distribution of z_i is defined in terms of the probability density function (Equation (7)).

$$f_{z_i}(z) = \frac{\lambda_e \Gamma(2k)}{\lambda_c \Gamma^2(k)} \left(\frac{z \lambda_e}{\lambda_c}\right)^{k-1} \left(1 + \frac{z \lambda_e}{\lambda_c}\right)^{-2k} \quad (7)$$

In which $\Gamma(\cdot)$ is the gamma function of a given parameter. Note that if $k < 1$, then the mean of z_i is infinite and if $k < 2$, then the variance of z_i is not defined. Hence, to be able to determine these values, k should be chosen as greater than 2. We assume that $k > 2$; then, the mean and variance of z_i are defined in Equations (8) and (9), respectively.

$$E(z_i) = \frac{\lambda_c}{\lambda_e} \frac{k}{k-1} \quad (8)$$

$$\text{var}(z_i) = \left(\frac{\lambda_c}{\lambda_e}\right)^2 \frac{k(2k-1)}{(k-2)(k-1)^2} \quad (9)$$

According to Equation (8), $\hat{\pi}_{e,i}$ can be estimated by $E(z_i)$ via Equation (10).

$$\hat{\pi}_{e,i} = E(z_i) \frac{k-1}{k} \quad (10)$$

The choice of k affects both the correctness of the estimated value $\hat{\pi}_{e,i}$ and the sensitivity of robots to the change of average global density. If k is increased, then the range of possible values of $\hat{\pi}_{e,i}$ is narrowed down to $E(\hat{\pi}_{e,i})$, but in order to notice changes in the swarm density, the robot needs to take a longer time. Another issue is that the time required to finish k samples depends on the swarm density when the number of samples k is fixed, rather than using a fixed sampling time. In other words, if the swarm density approaches 0, collecting k samples could take an infinitely long time. Therefore, the maximum sample period in terms of travelled distance S_s is provided as a solution to this issue. During the sampling time, if any measured event duration $s_{e,i,j}$ or $s_{d,i,j}$ of R_i exceeds S_s , then the estimating process will be terminated and considered as completed. In this case, if $\sum_{j=1}^{k_i} s_{e,i,j} < \sum_{j=1}^{k_i} s_{d,i,j}$, then $\hat{\pi}_{e,i} := 0$ and vice versa, $\hat{\pi}_{e,i} := 1$. The selection of S_s affects the minimum and maximum measurable value of $\hat{\pi}_{e,i}$ and the sensitivity of the robot to the change of swarm density. In the next section, we will estimate the likelihood of a robot encountering another robot at a specific time $\hat{\pi}_{e,i}(t)$ as a function of the partial distribution of robots throughout the relevant domain.

4. Nearest Neighbor Distance Distribution

Under the definition of an encounter state, an encounter is said to occur if the given robot, whose position is denoted as x_i , has at least one neighbor that is equivalent to the distance from x_i to the nearest neighbor of a given robot, whose position is denoted as x_j and is less than $r_r + r_s$. Hence, the encounter rate or the ratio that R_i experienced the encounter state in terms of travelled distance can also be determined by the probability that the random variable denoting the distance between R_i and its nearest neighbor R_j is less than $r_r + r_s$. Due to the time invariance of the encounter rate, this kind of probability only depends on the partial distribution of swarm robots over the domain. Hence, the patterns of geometrical swarm over the interested domain must be constructed first.

Point processes are most commonly used as a mechanism to interpret patterns of geometrical objects—for example, the position of swarm robot in our case. In this study, we describe the spatial distribution of the swarm of robots in a certain workspace using a homogeneous point process. Each robot is represented by a point in the point process, which also indicates where it is in the workspace. The Poisson process is the most basic type of point process. The selection of completely independent random points without observing the minimum distance is modelled by the Poisson process. However, the Poisson process is not suitable for modelling the position distribution of robots since it assumes that the locations of the robots are independent and uniformly distributed, which does not take into account the physical footprint of the robots or the fact that they cannot overlap.

The purely random selection of positions without observing a minimum distance is modelled by the Poisson process. Meanwhile, a hard-core process is a stochastic point process in which successive events maintain a specified minimum distance from one another. It is suitable for modelling the partial distribution of robots in a swarm. From the perspective of stochastic geometry, there exist n -dimensional point fields generated by hard-core processes using the center of n -dimensional balls that do not overlap and have a given diameter. Depending on the way in which the points are generated, different hard-core processes with different properties can be described.

Performing a dependent-thinning approach on an underlying “parent” spatial Poisson process, where points are gradually deleted until the appropriate hard-core requirements are satisfied by all the remaining points, is a typical technique for creating hard-core point processes. In this study, we focus on MHC type-II, fully detailed in [20]. To construct an MHC type-II process, we start with a real number $\delta > 0$ and a parent stationary parent Poisson process denoted by Φ_p , having a homogeneous intensity λ_p (points per unit area) and then for each point $\mathbf{x} \in \Phi_p$, a uniformly distributed random mark $M(\mathbf{x}) \sim U(0, 1)$ is assigned. If the mark of point $\mathbf{x} \in \Phi_p$ is not the lowest relative to all other points in a ball with radius δ , $b(\mathbf{x}, \delta)$, around it, that point will be marked for removal. Only after this test is done for all points in Φ_p , the points that were flagged-for-removal are removed. The retained points constitute the resulting MHC process Φ_{hc} . The distribution of robots over D can be modelled by using Φ_{hc} with parent Φ_p over D . Due to the removal of some points in the origin process, the point density of Φ_{hc} is not larger than the one in Φ_p . Let λ_{hc} denote the hard-core point process density; the relation between λ_{hc} and λ_p can be formulated as the following:

$$\lambda_p \pi \delta^2 = \ln \left(\frac{1}{1 - \lambda_{hc} \pi \delta^2} \right) \quad (11)$$

where δ is also called the hard-core distance of Φ_{hc} which is illustrated in Figure 1. In our case, δ can also be interpreted as the minimum distance between two robots, i.e., $\delta = 2r_r$. Furthermore, the distribution of the point in Φ_{hc} over D represents the partial distribution of the swarm robot over the same domain, resulting in $\lambda_{hc} = \lambda$. According to the proposed motion model, in this study, the MHC process Φ_{hc} with population density λ is used to describe the partial distribution of the swarm robot over D . Based on the constructed partial distribution of the swarm robots, the probability of experiencing an encounter state of R_i will be calculated.

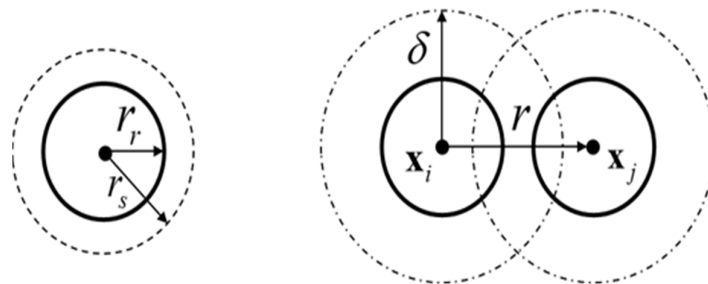


Figure 1. (Left): the geometric parameters of the robot; (right): the geometric parameters between robots.

To construct the nearest neighbor distance model, a geometric function must be determined first. According to Figure 1, let r be the distance between R_i and its nearest neighbor; $l_1(r, \delta)$ be the area of the symmetrical lens formed by the intersection of $b(\mathbf{x}_i, \delta)$ and $b(\mathbf{x}_j, \delta)$; and $l_2(r, \delta)$ be the area of the asymmetrical lens formed by the intersection of $b(\mathbf{x}_i, r)$ and $b(\mathbf{x}_j, \delta)$. $l_1(r, \delta)$ and $l_2(r, \delta)$ are illustrated in Figure 2 and can be defined as in Equations (12) and (13), respectively.

$$l_1(r, \delta) = \begin{cases} 2\delta^2 \cos^{-1}\left(\frac{r}{2\delta}\right) - \frac{1}{2}r\sqrt{4\delta^2 - r^2}, & 0 < r \leq 2\delta \\ 0, & r > 2\delta \end{cases} \quad (12)$$

$$l_2(r, \delta) = \begin{cases} \pi r^2, & 0 < r < \frac{\delta}{2} \\ r^2 \cos^{-1}\left(1 - \frac{\delta^2}{2r^2}\right) + \delta^2 \cos^{-1}\left(\frac{\delta}{2r}\right) - \frac{1}{2}\delta\sqrt{4r^2 - \delta^2}, & r \geq \frac{\delta}{2} \end{cases} \quad (13)$$

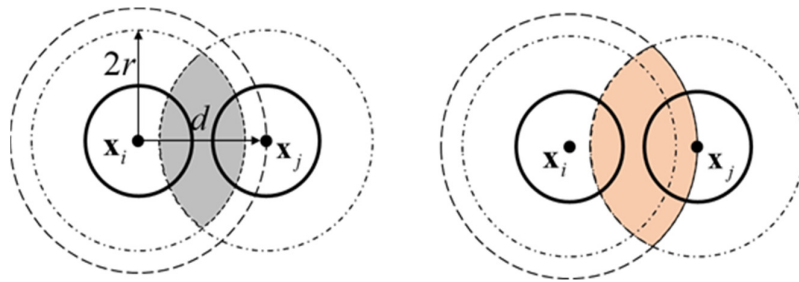


Figure 2. (Left): the black shaded region is the lens formed by the intersection of $b(\mathbf{x}_i, \delta)$ and $b(\mathbf{x}_j, \delta)$; (right): the orange shaded region is the lens formed by the intersection of $b(\mathbf{x}_i, r)$ and $b(\mathbf{x}_j, \delta)$.

According to [18], the cumulative probability function of the distance between a given point and its neighbor, who is likewise provided by the MHC process Φ_{hc} , can be approximated as Equation (14).

$$F_{\mathbf{R}}(R) = 1 - \exp\left(-\frac{2\pi\lambda_p^2}{\lambda_{hc}} \int_{\delta}^R r\kappa_1(r, \delta)dr\right) \quad (14)$$

where $\mathbf{R}$ is the random variable of nearest neighbor distance of a given point and $\kappa_1(r, \delta)$ can be expressed by Equation (15) with the components; $a = \lambda_p\pi\delta^2$, $b_1 = \lambda_p l_1(r, \delta)$ and $b_2 = \lambda_p l_2(r, \delta)$ are the expected numbers of the point occupying $b(\mathbf{x}_i, \delta)$, $l_1(r, \delta)$ and $l_2(r, \delta)$ in the parent Poisson process Φ_p , respectively.

$$\kappa_1(r, \delta) = \frac{1 - \exp(-a)}{a^2 - ab_2} + \frac{\exp(b_2 - 2a) - 1}{2a^2 + b_2^2 - 3ab_2} + \frac{\frac{1 - \exp(-b_1 + b_2 - a)}{b_1 - b_2 + a} + \frac{\exp(b_2 - 2a) - 1}{2a - b_2}}{(a - b_1)} \quad (15)$$

As stated, the likelihood that the random variable expressing the distance between the given robot and its nearest neighbor is smaller than $r_r + r_s$ can also be used to calculate the encounter rate or probability that the given robot experienced an encounter state in terms of travelled distance. Hence, according to (3) and (14), the relation between nearest neighbor distance distribution and encounter rate is determined as in Equation (16).

$$\pi_e = F_{\mathbf{R}}(r_r + r_s) \quad (16)$$

Based on Equation (16), if the desired swarm density λ_d is given, the desired encounter rate can be defined. Let $\pi_{e,d}$ be the preassigned desired encounter rate for the swarm, and let $Q \in \{-1, 0, 1\}$ be the desired decision of the swarm, where the value of Q is determined

based on the comparison between the current encounter rate and the desired encounter rate, as well as the threshold error σ .

$$Q = \begin{cases} -1 & \pi_e - \pi_{e,d} < \sigma \\ 0 & |\pi_e - \pi_{e,d}| \leq \sigma \\ 1 & \pi_e - \pi_{e,d} > \sigma \end{cases} \quad (17)$$

However, due to the randomness of encounter rate estimators, the values of Q_i , the decision of R_i based on $\hat{\pi}_{e,i}$ instead of π_e , are different among swarm robots. This variation increases when $\pi_e \rightarrow \pi_{e,d}$ or $\sigma \rightarrow 0$. The technique that allows the entire swarm to make the same conclusion is introduced in the following section.

5. Collective Decision Making

According to Section 3, the estimated encounter rate $\hat{\pi}_{e,i}$ of R_i can be determined by Equation (5). Once each robot has computed $\hat{\pi}_{e,i}$, they need to achieve consensus on the global value. However, the randomness of $\hat{\pi}_{e,i}$ prevents Q_i from sharing the same value across the whole swarm. This problem falls into a broader class of distributed consensus or agreement problems in multi-agent coordination and wireless sensor networks [21]. These protocols allow a swarm to reach an agreement on a quantity of interest by exchanging information over the communication network. Consensus algorithms have the attractive property that, at termination, the computed value is available throughout the swarm, so a swarm user can query any robot and immediately receive a response, rather than waiting for the query and response to propagate to and from a fusion center. Additionally, there is no single-dead-point where the result of in-swarm computation can be biased due to undesired values from failed robots. However, previous consensus algorithms require a fixed network topology and multi-hop data transmission. In this section, we propose an implementation of the quasi-distributed average consensus algorithm that does not rely on multi-hop communication and can adapt to a dynamic swarm. The main idea of the proposed method is that each robot updates its value by a weighted sum of values from itself and its neighbor, i.e.,

$$\pi_{e,i}(n_i + 1) = \sum_{j=1}^N W_{ij}(n_i) \pi_{e,j}(n_i) \quad (18)$$

where n_i is a discrete-time index taking values in the non-negative integers. In this section, we only consider $\hat{\pi}_{e,j}$ in the consensus process, and we use n_i as its index value. Obviously, we have that $\hat{\pi}_{e,j}(0) = \hat{\pi}_{e,j}$, where $\hat{\pi}_{e,j}$ represents the estimated encounter rate which is obtained by estimating the process. Here, $W_{ij}(n_i)$ is the linear weight on $\hat{\pi}_{e,j}(n_i)$ at R_i . Now, the question is how to choose the weight $W_{ij}(n_i)$ such that every $\hat{\pi}_{e,i}(n_i)$ converges to the average of their initial value, i.e.,

$$\lim_{n_i \rightarrow \infty} \hat{\pi}_{e,i}(n_i) = N^{-1} \sum_{i=1}^N \hat{\pi}_{e,i}(0) \quad (19)$$

Equation (18) can be interpreted as a local average if the value of $W_{ij}(n_i)$ is chosen to follow the nearest neighbor rules proposed in [22]. This weight can simplify the communication method since it depends only on the value of the neighbor set and the number of neighbors.

$$W_{ij}(n_i) = \begin{cases} \frac{1}{1+|\mathbf{F}_i(t_i(n_i))|} & R_j \in \mathbf{F}_i(t_i(n_i)) \text{ or } i = j \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

in which $t_i(n_i)$ represents the time that R_i determines $\pi_{e,i}(n_i + 1)$. To simplify the notation, vector notation is used. Let $\hat{\pi}_e(m) = [\hat{\pi}_{e,1}(n_1), \hat{\pi}_{e,2}(n_2), \dots, \hat{\pi}_{e,N}(n_N)]^T$ be the vector that represents the set of the estimated encounter rate of the swarm, in which m is a discrete-time

index value. The value of m can be interpreted as the total number of non-simultaneous encounter rate updates. This way, the distributed averaging algorithm (13) can be written as Equation (21).

$$\hat{\pi}_e(m+1) = \mathbf{W}(m)\hat{\pi}_e(m) \quad (21)$$

where the weight matrix $\mathbf{W}(m) \in \mathbb{R}^{N \times N}$ is given by (20) and it satisfies the conditions required for the asymptotic average consensus $\mathbf{W}(m)\mathbf{1} = \mathbf{1}$, $\mathbf{1}^T\mathbf{W}(m) = \mathbf{1}^T$ and $\rho(\mathbf{W}(m) - N^{-1}\mathbf{1}\mathbf{1}^T) < 1$, in which $\mathbf{1}$ stands for the vector of ones and $\rho(\cdot)$ is the spectral radius of a given matrix. If Equation (16) is expanded by recursively replacing $\pi_{e,j}(t)$ by Equation (16), then the value of $\hat{\pi}_e(m+1)$ can be derived from $\hat{\pi}_e(0)$.

$$\hat{\pi}_e(m+1) = \left(\prod_{i=0}^m \mathbf{W}(i) \right) \hat{\pi}_e(0) \quad (22)$$

By applying the weight matrix with components defined in Equation (20) to Equation (22), Morse et al. [22] proved that every individual in the robot swarm will converge to the same value as in Equation (19), resulting in Equation (23).

$$\lim_{m \rightarrow \infty} \prod_{i=0}^m \mathbf{W}(i) = N^{-1}\mathbf{1}\mathbf{1}^T \quad (23)$$

To make the proposed method able to be applied, the inter-robot communication mechanism must be clarified first. The first requirement of the communication mechanism is that each robot must have both broadcast and directional communication. To do that, infrared-based communication is employed [23]. Fortunately, this type of communication using an infrared medium can prevent most signals, and messages created by infrared transceivers can be modulated. The second requirement is bidirectional communication in which if a robot receives any messages from its neighbor, it has the responsibility to reply to the sender.

The consensus algorithm is asynchronously applied to the robot as soon as the robot acquires $\hat{\pi}_{e,i}(0)$. In this process, the robot will broadcast its current estimated encounter rate $\hat{\pi}_{e,i}(n_i)$ to its neighbors if it has no received messages for the duration S_q in terms of travelling distance. According to the second requirement, all robots receiving that message will reply to their current estimated encounter rate first; then, it applies the recorded value to Equation (18). After receiving all replies, R_i also applies all recorded values to Equation (18).

Equation (18) is the ideal result that can make a member of $\hat{\pi}_e(m)$ converge to $N^{-1}\mathbf{1}\mathbf{1}^T\hat{\pi}_e(0)$; by this, the decisions of the whole swarm approach is totally collective. However, to help the swarm make the decision based on Equation (17), the sign to indicate the complete process must be pointed out. Let $range(\mathbf{a})$ denote the range of the arbitrary set of number $\mathbf{a}$; the main purpose of this method is narrowing down $range(\hat{\pi}_e(m))$ to a value that can make the swarm have a collective sense about the estimated value. Let ϑ indicate the acceptable range of $\hat{\pi}_e$. Obviously, if $range(\hat{\pi}_e(m)) \leq \vartheta$, then the condition in Equation (24) must be held.

$$P(|\hat{\pi}_{e,i}(n_i) - \hat{\pi}_{e,j}(n_j)| \leq \vartheta) = 1, \forall i, R_j \in \mathbf{F}_i(t(n_i)) \quad (24)$$

Based on this condition, the consensus process of the swarm can be considered as finished. Let S_u be the discrete sampling duration of $|\hat{\pi}_{e,i}(n_i) - \hat{\pi}_{e,j}(n_j)|$ in terms of the travelling distance of robots in the swarm. S_u will start counting if the condition in Equation (24) is satisfied. During the travelled distance S_u , if the given condition is continuously satisfied, then the consensus process is considered as finished. At any time, if the above condition is not satisfied, S_u will be reset and pause until Equation (24) is satisfied again. Therefore, S_u should be chosen so as to achieve a trade-off between the steady-state of convergence and the ability of the robot to track changing densities. In case of $\hat{\pi}_{e,i}(0) = 0$

or $\hat{\pi}_{e,i}(0) = 1$, if the robot has not received a message during the travelled distance S_s in the consensus process, the robot will terminate the consensus process. In this case, the decision is based on $\hat{\pi}_{e,i}(0)$.

6. Dynamic Swarm Case

The iterative estimate and consensus processes must be used alternatively to be able to adjust to changes in swarm density in real time. The robots will perform these processes at different speeds due to the randomness of the process completion time, though, when the robot instantly switches to the estimating process to estimate the new value as soon as the consensus process is complete. The opinions of the robots in the swarm may no longer be consistent if this procedure is out of phase among them.

After consensus is reached and before the entire process starts over again, an intermediate phase called synchronization is implemented in order to address this issue. The procedure helps robots that have completed the consensus process to wait for other robots that are still in the process of reaching the consensus, without synchronizing the clocks of the robots in the swarm. Figure 3 provides a summary of the full procedure for the swarm's collective perception of density tailored to dynamic swarm.

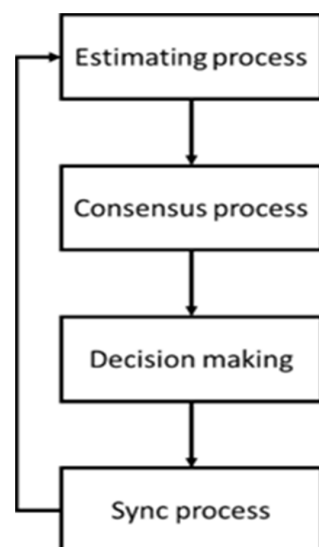


Figure 3. The whole process of proposed method.

As shown in Figure 3, the robot will transition to the sync procedure after completing the consensus process. The robot will keep track of the travelled distance S_w during this operation. The sync procedure ends and the robot resumes to the estimating process if the counter reaches S_w . The robot will not automatically send any messages during the sync process. However, if it receives a message from another robot from the consensus phase, it will reply with its decision rather than an estimated value $\hat{\pi}_{e,i}$. The asymptotic average consensus is maintained in this manner. Robotic consensus processes can be considered completed when the correlation between the robot's present estimated value and the intended value matches the majority decision the robot has received. The comparison between the estimated value at the time and the desired value will lead to the decision of the robots in this scenario.

7. Performance Evaluation

The performances of the proposed model are evaluated according to the test which is conducted in a real experiment. All tests use the same robot platform which has a top and perspective view illustrated in Figure 4. The robot platform is cylindrical in shape with a radius r_r of 25 mm and height h_r of 40 mm, respectively. The inter-wheel distance l_w and radius of each wheel r_w are 40 mm and 10 mm, respectively. Given that the robot is

equipped with 6 infrared modules, as shown in Figure 4, arranged in a specific order, the maximum number of neighbors of robot $|F_i|_{\max}$, $\forall i \in \{1, 2, \dots, N(t)\}$ obviously is 6.

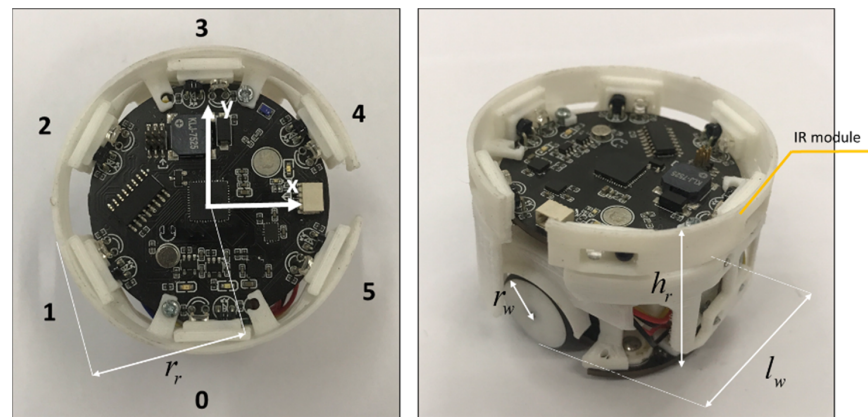


Figure 4. The robot platform utilized in this research is depicted in the image. On the left-hand side, there is a top view of the robot platform, including the IR-module with annotation numbers ranging from 0 to 5 that indicate their order and local frame notations. On the right-hand side, there is a side view of the robot platform.

Without adding information, the default communication range and default sensing range r_s of these infrared modules are approximately 180 mm. Finally, the robot's linear speed can reach 35 mm/s. Due to the communication speed not being fast enough, the maximum linear speed has been set to 20 mm/s according to our experiments. The default initiation constants and parameters used in this section were listed in Table 1.

Table 1. Constants and default parameters used in experiments.

Parameter	Description	Value
k	Number of samples which are used to estimate encounter rate in estimating process.	20
S_s	The maximum sample value in terms of travelled distance in estimating process.	5 m
S_q	The maximum duration in terms of travelled distance to self-broadcast current estimated encounter rate in consensus process.	1 m
S_w	The counter in terms of travelled distance in sync process.	10 m
θ	Acceptable range.	0.001
σ	Threshold error.	0.02

The test area for the swarm robots in this experiment is a square-shaped workspace measuring $2000 \times 2000 \text{ mm}^2$, marked by a black line to indicate the boundaries. Three infrared sensors are installed at the bottom of each robot to determine the boundaries of the work area. The message transmission process, which contains the message protocol and estimated value time of the robot, takes approximately 10 ms.

Since the swarm's decision-making is based on the estimated encounter rate of the robots, the value that all robots converge to is crucial. Therefore, the convergence of estimated encounter rates of the swarm is first evaluated under the static swarm size. In this case, we set $\theta = 0$, i.e., the consensus process will continue to run until the experiment is terminated by the experimenter. Three experiments, in which the swarm size is set to 20, 40 and 60, are conducted to evaluate the convergence. According to Equations (11) and (16), the desired encounter rates $\pi_e \in \{0.4811, 0.7339, 0.865\}$ correspond to $N \in \{20, 40, 60\}$. In order to obtain the results, a central server was set up for the task of collecting data from all robots in the swarm. Each robot was equipped with a Wi-Fi communication system previously connected to the server. The robot will continuously stream its data to the server, and the type of data collected depends on the experiment. For this experiment, robots

presenting in the workspace will continuously transmit their current estimated encounter rate to the server. The results of these experiments are plotted in Figure 5.

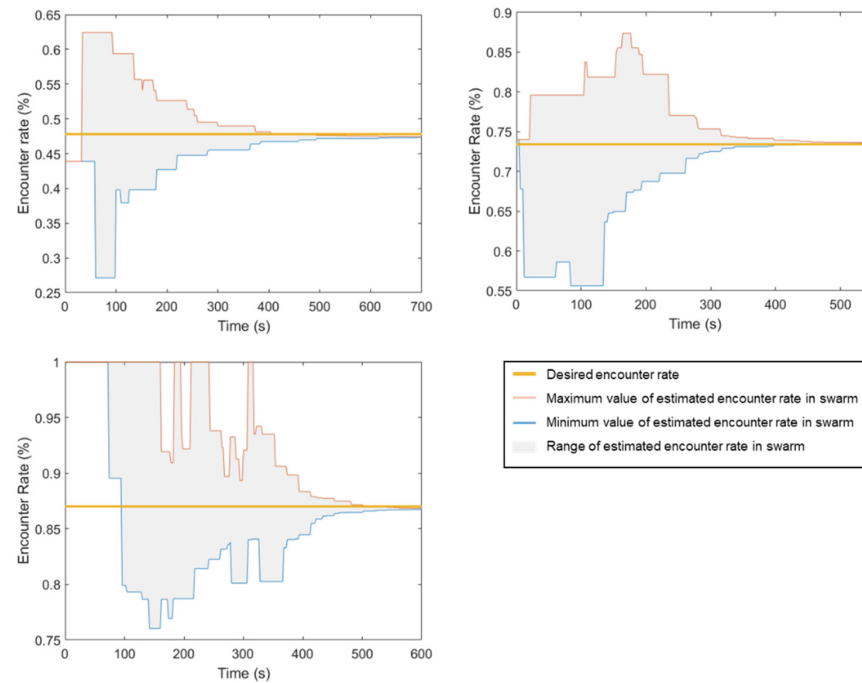


Figure 5. The results of experiments conducted to evaluate the convergence of the estimated encounter rate in the swarm are presented for a static swarm size scenario. The experiments were conducted for a swarm size of 20 (**top-left**), 40 (**top-right**), and 60 (**bottom-left**).

In Figure 5, at the initial time of all conducted experiments, the estimated encounter rate of the robots is unknown, so the time from the beginning to the appearance of the first complete estimated robot is not recorded. The values $\min(\hat{\pi}_e)$ and $\max(\hat{\pi}_e)$ are only recorded when any robot enters the consensus process; therefore, at the initial time on the graphs, $\min(\hat{\pi}_e) = \max(\hat{\pi}_e)$. However, soon after, as other robots join the consensus process, the range between the maximum and minimum estimated encounter rates begins to increase, leading to a rise in $range(\hat{\pi}_e)$. The consensus process gradually decreases $range(\hat{\pi}_e)$ until the values converge if $t \rightarrow \infty$. During the consensus process, not all communication between robots is successful, which can cause the swarm to deviate from the original mean value of the encounter rate. Specifically, in the case of $\pi_e = 0.4811$, at $t = 650s$, when $range(\hat{\pi}_e) < \vartheta$ satisfies the consensus process condition, then $0.4736 \leq \hat{\pi}_{e,i} \leq 0.474$, $\forall R_i \in S$. However, this deviation is not significant enough to affect the collective decision of the swarm.

Based on the results of the experiments, it can be affirmed that the convergent rate of the encounter rate is dependent on the density of the swarm, but not in a linear fashion. In general, as the number of robots in the swarm increases, the number of transmissions required for achieving a consensus also tends to increase. This is because each robot needs to communicate its current state to its neighbors in order to reach a consensus, and the number of neighbors increases with the number of robots. As a result, the overall communication overhead increases. However, this relationship may not always be straightforward, as the nature of the coordination problem, communication protocol and other factors can also impact the number of transmissions required. In our case, since the workspace remains constant, the encounter rate increases as the swarm size increases. As a result, the consensus time does not depend linearly on the swarm density.

To be able to demonstrate that the method is available for a swarm of robots with varying sizes, an experiment was performed on an initial swarm size $N(0) = 20$, and a new robot was added every minute. The desired encounter rate $\pi_{e,d}$ is set to 0.8. The

robot swarm is tasked with continuously making collective decisions about whether the current population of robots will satisfy a given desired encounter rate. The results of the experimental process are shown in two graphs: (1) $range(\hat{\pi}_e)$, $E(\hat{\pi}_e)$ and π_e over time; (2) different number of robot decisions over time. Snapshots of this experiment are recorded and shown in Figure 6.

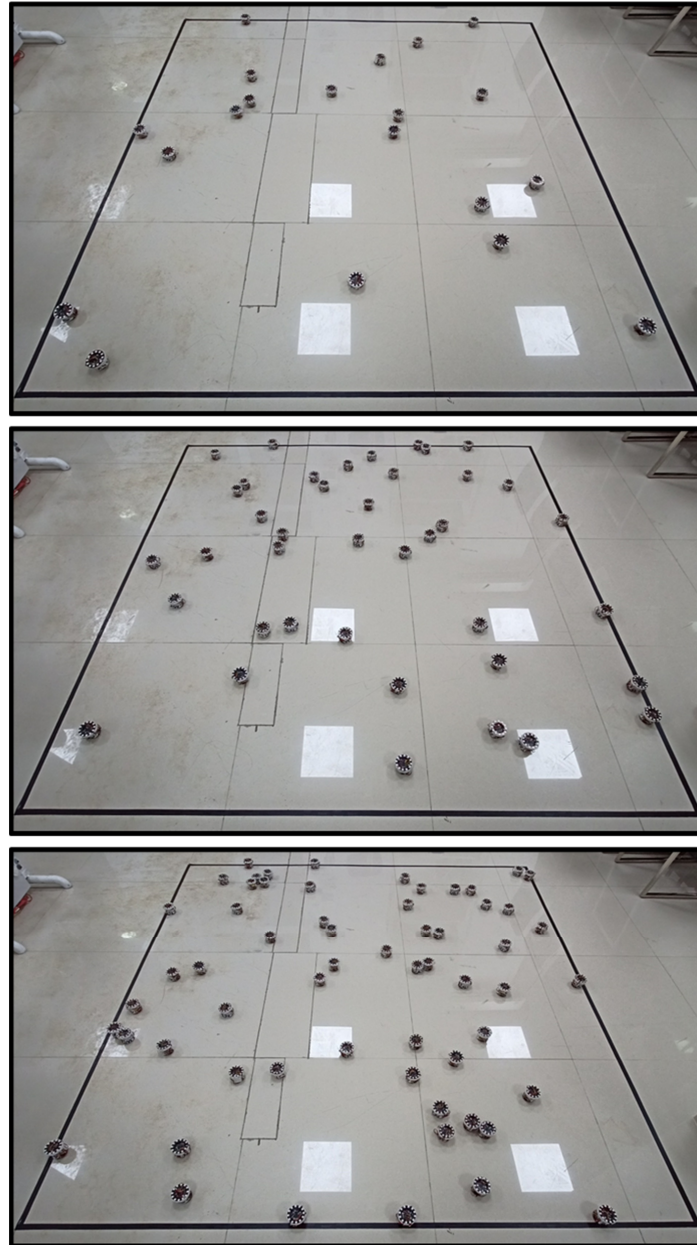


Figure 6. The snapshots of the second experiments used to evaluate the performance of the proposed method on swarms with varying sizes are shown. From top to bottom, the number of robots in the workspace are 20, 40 and 60, respectively.

According to Figure 7, the mean of the estimated encounter rate over the swarm is underestimated compared to the actual encounter rate due to the latency of the proposed algorithm. When new robots are added, the desired encounter rate increases instantly, but the encounter rate estimation value for previous robots depends on the robot density in the past. These robots only update the new value when they complete the sync process and start a new estimation process. Therefore, if the change rate of the desired encounter rate increases at a higher rate, this requires the algorithm to respond faster; otherwise, the difference between the desired encounter and mean of the estimated encounter rate will increase significantly, resulting in the robot's inability to make accurate decisions.

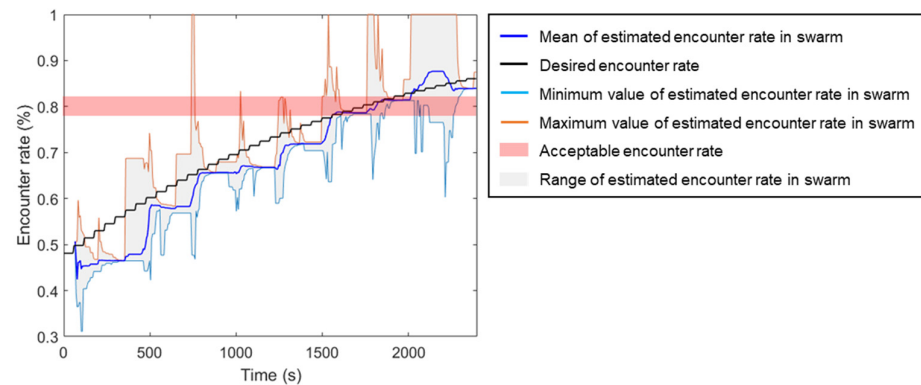


Figure 7. The recording of range of estimated encounter rate $range(\hat{\pi}_e)$, mean of estimated encounter rate $E(\hat{\pi}_e)$ and desired encounter rate π_e over time during the experiment.

In Figure 8, it can be observed that the consensus process, most of the time, enables the swarm to make a collective decision, except for the newly added robots that are still in the estimating process. Additionally, the sync process ensures that the robots return to the estimating process almost simultaneously. Consequently, the decision delay depends on the total time taken from when the robot starts estimating the encounter rate to the end of the sync process. The threshold error and the rate of change of the encounter rate must be selected to match the latency of the method. In this case, with $\sigma = 0.02$ and the swarm growth rate as 1 robot/min, the duration of satisfying the desired encounter rate is about 350 s. This interval allows the method to respond in time, and all robots capable of making a decision at this time interval can make the appropriate decision.

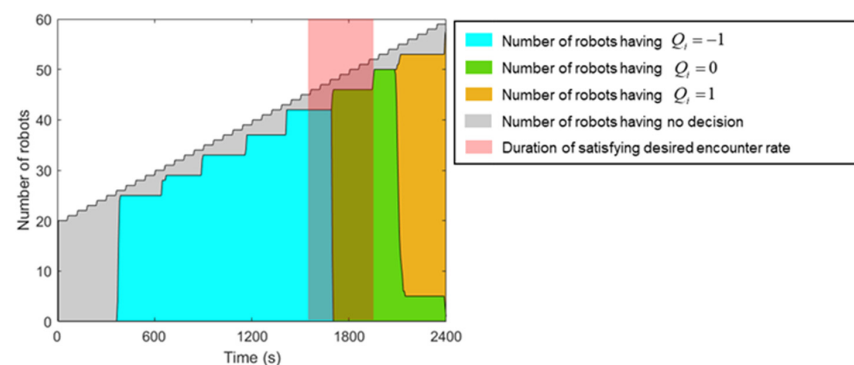


Figure 8. Number of robots at each decision and total number of robots in the swarm at each time point of the experiment.

8. Conclusions

In this study, a methodology that enables robot swarms to collectively assess whether their actual swarm density exceeds, is insufficient or is approximately comparable to a specified desired swarm density is introduced. The main goal of the proposed method is to enable the robots to collectively estimate the swarm density by assessing how frequently

they interact with their neighbors, which will enable the swarm to come to a consensus. The proposed method does not require robots to have knowledge about their workspace, swarm size or their localization in the workspace. However, the trajectories of the robots in the swarm must be uniformly ergodic and the communication system must be a bidirectional communication with explicit messages. The experiments conducted on our proposed swarm robot platform show that the method is applicable to a dynamic swarm.

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, P.G.L.; 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 University of Economics Ho Chi Minh City–UEH University, 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.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research is supported by the University of Economics Ho Chi Minh City, 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. Groß, R.; Dorigo, M. Towards group transport by swarms of robots. *Int. J. Bio-Inspired Comput.* **2009**, *1*, 1–13. [\[CrossRef\]](#)
2. Wilson, S.; Pavlic, T.P.; Kumar, G.P.; Buffin, A.; Pratt, S.C.; Berman, S. Design of ant-inspired stochastic control policies for collective transport by robotic swarms. *Swarm Intell.* **2014**, *8*, 303–327. [\[CrossRef\]](#)
3. Sugawara, K.; Correll, N.; Reishus, D. Object Transportation by Granular Convection Using Swarm Robots. In *Distributed Autonomous Robotic Systems*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 135–147.
4. Campo, A.; Dorigo, M. Efficient Multi-foraging in Swarm Robotics. In *European Conference on Artificial Life*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 696–705.
5. Talamali, M.S.; Bose, T.; Haire, M.; Xu, X.; Marshall, J.A.R.; Reina, A. Sophisticated collective foraging with minimalist agents: A swarm robotics test. *Swarm Intell.* **2020**, *14*, 25–56. [\[CrossRef\]](#)
6. Valentini, G.; Brambilla, D.; Hamann, H.; Dorigo, M. Collective Perception of Environmental Features in a Robot Swarm. In *International Conference on Swarm Intelligence*; Springer: Cham, Switzerland, 2016.
7. Ebert, J.T.; Gauci, M.; Nagpal, R. Multi-feature collective decision making in robot swarms. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, Stockholm, Sweden, 10–15 July 2018.
8. Huang, X.; Arvin, F.; West, C.; Watson, S.; Lennox, B. Exploration in Extreme Environments with Swarm Robotic System. In Proceedings of the 2019 IEEE International Conference on Mechatronics (ICM), Ilmenau, Germany, 18–20 March 2019; Volume 1.
9. Wahby, M.; Petzold, J.; Eschke, C.; Schmickl, T.; Hamann, H. Collective change detection: Adaptivity to dynamic swarm densities and light conditions in robot swarms. In *ALIFE 2019: The 2019 Conference on Artificial Life*; MIT Press: Cambridge, MA, USA, 2019.
10. Soysal, O.; Sahin, E. A Macroscopic Model for Self-organized Aggregation in Swarm Robotic Systems. In *International Workshop on Swarm Robotics*; Springer: Berlin/Heidelberg, Germany, 2006.
11. Cambier, N.; Frémont, V.; Trianni, V.; Ferrante, E. Embodied Evolution of Self-organised Aggregation by Cultural Propagation. In *International Conference on Swarm Intelligence*; Springer: Cham, Switzerland, 2018.
12. Zhao, S.; Ramakrishnan, S.; Kumar, M. Density-based control of multiple robots. In Proceedings of the 2011 American Control Conference, San Francisco, CA, USA, 29 June–1 July 2011.
13. Gazi, V.; Passino, K. Stability Analysis of Social Foraging Swarms. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2004**, *34*, 539–557. [\[CrossRef\]](#)
14. Shiwakoti, N.; Sarvi, M.; Rose, G.; Burd, M. Animal dynamics based approach for modeling pedestrian crowd egress under panic conditions. *Procedia—Soc. Behav. Sci.* **2011**, *17*, 438–461. [\[CrossRef\]](#)
15. Mayya, S.; Pierpaoli, P.; Nair, G.; Egerstedt, M. Localization in Densely Packed Swarms Using Interrobot Collisions as a Sensing Modality. *IEEE Trans. Robot.* **2018**, *35*, 21–34. [\[CrossRef\]](#)
16. Mayya, S.; Pierpaoli, P.; Egerstedt, M. Voluntary Retreat for Decentralized Interference Reduction in Robot Swarms. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.

17. Mayya, S.; Wilson, S.; Egerstedt, M. Closed-loop task allocation in robot swarms using inter-robot encounters. *Swarm Intell.* **2019**, *13*, 115–143. [\[CrossRef\]](#)
18. Al-Hourani, A.; Evans, R.J.; Kandeepan, S. Nearest Neighbor Distance Distribution in Hard-Core Point Processes. *IEEE Commun. Lett.* **2016**, *20*, 1872–1875. [\[CrossRef\]](#)
19. McDonald, J.B.; Xu, Y.J. A generalization of the beta distribution with applications. *J. Econ.* **1995**, *66*, 133–152. [\[CrossRef\]](#)
20. Teichmann, J.; Ballani, F.; van den Boogaart, K.G. Generalizations of Matérn’s hard-core point processes. *Spat. Stat.* **2013**, *3*, 33–53. [\[CrossRef\]](#)
21. Ren, W.; Beard, R.; Atkins, E. A survey of consensus problems in multi-agent coordination. In Proceedings of the 2005 American Control Conference, Portland, OR, USA, 8–10 June 2005.
22. Jadbabaie, A.; Lin, J.; Morse, A. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Control* **2003**, *48*, 988–1001. [\[CrossRef\]](#)
23. Arvin, F.; Samsudin, K.; Ramli, A.R. Development of IR-Based Short-Range Communication Techniques for Swarm Robot Applications. *Adv. Electr. Comput. Eng.* **2010**, *10*, 61–68. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.