

Voluntary Retreat for Decentralized Interference Reduction in Robot Swarms*

Siddharth Mayya, Pietro Pierpaoli and Magnus Egerstedt ‡

Abstract—In densely-packed robot swarms operating in confined regions, spatial interference – which manifests itself as a competition for physical space – forces robots to spend more time navigating around each other rather than performing the primary task. This paper develops a decentralized algorithm that enables individual robots to decide whether to stay in the region and contribute to the overall mission, or vacate the region so as to reduce the negative effects that interference has on the overall efficiency of the swarm. We develop this algorithm in the context of a distributed collection task, where a team of robots collect and deposit objects from one set of locations to another in a given region. Robots do not communicate and use only binary information regarding the presence of other robots around them to make the decision to stay or retreat. We illustrate the efficacy of the algorithm with experiments on a team of real robots.

I. INTRODUCTION

Swarm robotic systems are being increasingly deployed in dynamic real-world environments without the need for a central coordinator overseeing the mission (e.g., see [1], [2] and references within). As the size of these swarms increases, robots tend to spend more time and effort avoiding collisions between each other, thereby hurting the overall performance of the swarm [3]. This phenomenon—an inevitable consequence of the competition for physical space—is called multi-robot interference and has been studied in a wide number of contexts (e.g., [4], [5], [6]).

In the context of multi-robot *foraging* [7], [8] and *collection* tasks [9], [10], it has been shown that while adding more robots to the swarm at intermediate robot densities increases the total number of objects collected, the increase in performance is sub-linear—caused by a significant decrease in individual robot performance, e.g., [11], [12]. Owing to the increased operational cost of deploying large robot teams, one would like to obtain the desired overall productivity while utilizing as few robots as possible [13]. Achieving such a trade-off is especially challenging in situations where relatively simplistic robots operate in dynamic environments without a central coordinator.

For a swarm of robots executing a *distributed collection* task, this paper develops a decentralized algorithm which enables individual robots to decide between staying in the region and contributing to the task, or retreating from the region when the negative effects of interference outweigh the

increased productivity of a larger swarm. In order to allow individual robots to make these decisions, the first part of this paper develops an analytical model which describes the fraction of time that a robot spends performing the primary task as opposed to avoiding collisions with other robots. This allows us to compute an optimal robot density that minimizes a cost function in order to achieve a trade-off between the deployed team size and the overall performance of the swarm.

We envision a team of robots collecting objects from randomly scattered “pick-up” points and depositing them at “drop-off” points. The locations of these points are unknown to the robots. Such a distributed collection task is of relevance to multi-robot applications such as search & rescue, mine-clearing, moving waste to recycling stations, transporting passengers to destinations, etc. Furthermore, we assume (i) a constant influx of robots which join the collection task, and (ii) no explicit communication between the robots, thus inspiring the need for a dynamic, yet decentralized mechanism to regulate interference in the domain.

Entomological studies have shown that many social insect colonies possess the ability to regulate interference using decentralized techniques, e.g. [14], [15]. In [16], the authors study the ability of *Solenopsis Invicta* ants to regulate densities in narrow tunnels and prevent the formation of flow-stopping clogs. This is partially attributed to the tendency of individual ants to reverse out of crowded tunnels thus reducing the density. Crucially, it has been shown that ants use inter-ant encounters as the primary sensory mechanism to regulate density and perform functions like task allocation and division of labor [17].

Inspired by these observations from biology, we allow robots to use local inter-robot encounter measurements as the only sensory mechanism to regulate interference. We extend the ideas presented in our previous work [18], and demonstrate that not just physical inter-robot collisions but local “proximity” encounters can allow robots to infer something about the swarm. Using the analytical collision model developed in [19], we allow robots to measure the density of the swarm by counting interactions.

This paper is organized as follows: Section II discusses the contributions of this paper in the context of existing literature. Section III develops analytical models to characterize the frequency of encountering objects and other robots in the domain, given a particular robot density. Section IV leverages these analytical models to compute the optimal density of robots for achieving a desired trade-off between the size of the swarm and the overall performance. In Section V, we

*This work was supported by the Army Research Lab under Grant DCIST CRA W911NF-17-2-0181.

‡The authors are with the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta, USA {siddharth.mayya, pietro.pierpaoli, magnus}@gatech.edu

develop an algorithm which allows each robot to leave the domain if the measured density of robots in the swarm is above the optimal value. In Section VI, the algorithm is deployed on a team of real robots. Section VII concludes the paper.

II. RELATED WORK

The modeling and regulation of interference in swarm robotic systems is a widely studied topic (e.g., see [8], [20], [21]). The effects of interference on the efficiency of a swarm have been studied experimentally e.g., [4], [9], [22], [23], and by developing analytical models [11], [21]. Techniques to regulate interference range from performing “aggression” maneuvers to break deadlocks [4], [24], partitioning robots among tasks [25], and team size selection before deploying the swarm [26].

Many biologically inspired techniques allow a group-level division of labor to emerge in order to enhance the efficiency of the swarm, e.g., [7], [27], [28]. In the prey-retrieval task studied in [29], the authors demonstrate via robotic experiments that allowing robots to modify their tendency to participate in the primary task leads to reduced interference. Similar studies predefine participatory tendencies in robots [28], or allow them to adjust it online [7], demonstrating an improvement in efficiency by running experiments.

In contrast to these approaches, this paper develops a decentralized interference reduction mechanism which allows the swarm to adaptively achieve the *optimal* robot density for a desired trade-off between team size and overall productivity. Furthermore, we assume no coordination/communication between the robots and a constant influx of robots joining the task.

Similar to [30], [31], we use the theory of continuous-time Markov chains (CTMCs) to compute the *distribution* of time spent by robots performing the primary task and in collision avoidance. The development of a CTMC framework is borne out of necessity: in order to allow robots to make decisions at an individual level, we require more than a mean-field characterization of the effects of interference. In the next section, we introduce the problem setup and develop analytical models to describe the interference in the swarm.

III. MOTION AND ENCOUNTER MODELS

We consider a homogeneous team of robots—each with a circular footprint of radius r and a sensing skirt of total diameter 2δ around it to detect objects (see Fig. 1). Let v denote the average speed of the robots as they move through the domain.

As illustrated in Fig. 1, the domain $\mathcal{D} \subset \mathbb{R}^2$ contains a set of randomly, but uniformly distributed pick-up locations (marked as \square) and drop-off locations (marked as \times) from where robots can pick up objects and deliver them, respectively. Each robot picks up or drops off an object if it gets within the sensing distance δ of the corresponding location.

In this paper, we consider the scenario where picked up objects do not have a specific destination but can be dropped off at any of the drop-off locations. A robot picks up only

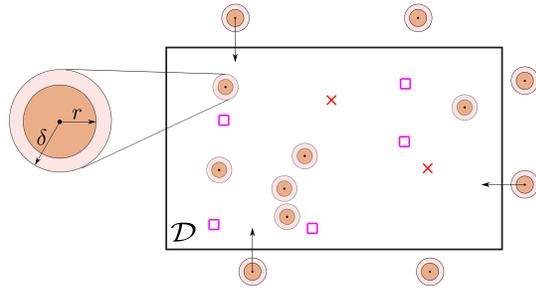


Fig. 1. Distributed Collection Task: A team of robots pick up objects from randomly scattered pick-up locations (denoted by \square) and deliver them to drop-off locations (denoted by \times). An object can be delivered to any drop-off location. Pick-up and drop-off locations are distributed independently with densities λ_p and λ_d , respectively. New robots are stationed outside the domain and enter the domain at a constant rate λ_{in} , symbolized by the arrows.

one object at a time, and delivers it to a destination before picking up another object. We assume that the positions of the pick-up and drop-off points are unknown to the robots, but they know the corresponding non-zero uniform densities, denoted as λ_p and λ_d . Furthermore, new robots initially outside domain \mathcal{D} join the collection task at a constant rate λ_{in} , as illustrated in Fig. 1. The rest of this section develops analytical models to characterize the performance of the swarm and the interference at a *particular* robot density, denoted by λ .

A. Motion Model

The robots do not have any prior knowledge of where the transport locations are, and traverse the domain with trajectories which are ergodic (e.g., [32], [33]) with respect to a uniform distribution. The density distribution $\phi : \mathcal{D} \rightarrow \mathbb{R}$ corresponding to a uniformly ergodic trajectory is given by $\phi(x) = |\mathcal{D}|^{-1}$, $\forall x \in \mathcal{D}$ where $|\mathcal{D}|$ is the area of the domain. This ergodicity property allows us to be general about the motion patterns of the robots—the developed algorithms are applicable as long as the ergodicity conditions are satisfied. Under this ergodicity assumption on the motion of the robots, the distribution of robots will also be uniform in the domain.

B. Inter-Robot Encounters

As robots explore the domain, they cross paths with each other. We define two robots i and j as having an “encounter” when they come within each other’s sensing radius, i.e., when $\|p_i - p_j\| \leq 2\delta$ where p_i represents the location of robot i . At this point, each robot executes a collision avoidance maneuver (e.g., [34], [35]) to resolve the encounter and continues on its intended path.

The encounters experienced by a robot will depend on its motion and the motion of all the other robots in the domain. Similar to our previous work [18], [19], we simplify such complex interactions by making a mean-field approximation and assume that all robots are identical under the effects from other robots in the swarm. There are two relevant questions which arise in this context: how often does a given robot encounter other robots in the domain and how long does it take to resolve an encounter?

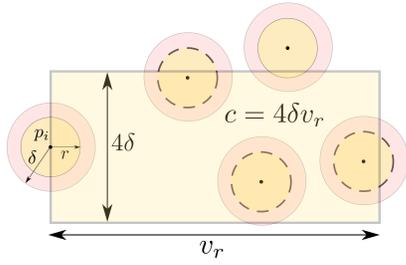


Fig. 2. The effective “sensory” area swept by robot i in unit time is depicted as the shaded region. The expected number of encounters experienced by the robot in unit time will be equal to the number of other robot centers that happen to fall in this region [19], [36], whose area is denoted by c . Accordingly, the robots experiencing an encounter are indicated with dashed outlines. v_r is the average relative speed between the robots, given by (1).

1) *Inter-Encounter Time Intervals*: Similar to the analysis conducted in kinetic theory of gases [36] and in [19], [37], if we assume that all robots except the robot under study are stationary, then the expected number of encounters experienced by this robot in unit time will be equal to the number of robot centers which happen to fall into the effective “sensory” area swept by the robot (see Fig. 2). This can be computed as $c = 4\delta v$. But, since all other robots are also moving with an average speed v , the *effective* speed of the robot can be replaced by the average relative speed between the robots, which can be computed as [19],

$$v_r = \frac{4}{\pi}v. \quad (1)$$

Thus, the effective area swept by the robot is $c = 4\delta v_r$. Owing to the ergodic motion model developed in Section III-A, the density of robots over the domain D is uniform and is denoted as λ . Thus, the expected number of encounters experienced by a robot in unit time is given as,

$$\Omega_c(\lambda) = c\lambda = 4\delta\frac{4}{\pi}v\lambda. \quad (2)$$

Although this section characterizes the performance of the swarm for a given robot density λ , the decentralized control algorithm developed in Section V takes into consideration the time-varying densities resulting from the entry and exit of robots.

For a given robot, let $\tau_c(\lambda)$ be the random variable denoting the time interval between entering two successive encounters. Similar to the analysis done in kinetic theory of gases [36], we make the assumption that inter-robot encounters are independent events that are not influenced by each other. This allows us to model $\tau_c(\lambda)$ as an exponentially distributed random variable (see [36] for a discussion of this fact),

$$\tau_c(\lambda) \sim \exp(\Omega_c(\lambda)). \quad (3)$$

In Section III-D, we perform simulations to validate the above developed encounter model for robots moving with uniformly ergodic trajectories and executing a specific collision avoidance algorithm.

2) *Time Spent Resolving an Encounter*: When robots encounter each other, they execute collision avoidance maneuvers. Let τ_r be the random variable which denotes the time it takes for a robot to resolve an encounter with another robot. τ_r will depend on a variety of factors such as the collision avoidance maneuvers used by the robots, velocities before collision, presence of domain boundaries, congestion, etc. The uncertainty associated with performing multiple sub-tasks in robotic/vehicular systems has been modeled using sums of exponential random variables [31], [38]. Similar to these formulations, we model τ_r as an exponentially distributed random variable with a parameter ρ that will vary based on the type of robots and collision avoidance mechanisms used,

$$\tau_r \sim \exp(\rho). \quad (4)$$

In practice, the parameter ρ can be estimated by fitting the empirical encounter resolution times to the distribution in (4). In Section III-D, we illustrate using simulations that τ_r is indeed well-approximated by an exponential distribution for a given collision avoidance algorithm.

C. Object Pick-Up and Drop-Off

Let λ_p and λ_d denote the uniform density of object pick-up and drop-off locations in the domain, respectively. As the robots move through the domain, they pick-up/drop-off objects when they enter within a sensing distance δ of the respective location. Similar to the analysis done in Section III-B, the expected number of locations encountered per unit time will be,

$$\Omega_p(\lambda_p) = 2\delta v\lambda_p \ ; \ \Omega_d(\lambda_d) = 2\delta v\lambda_d, \quad (5)$$

where Ω_p and Ω_d correspond to the expectations for pick-up and drop-off locations, respectively. Furthermore, since transport locations are uniformly random, the time elapsed between encountering two such locations can be modeled as an exponentially distributed random variable. Let τ_p and τ_d denote the time between encountering two successive pick-up and drop-off locations, respectively,

$$\tau_p(\lambda_p) \sim \exp(\Omega_p(\lambda_p)) \ ; \ \tau_d(\lambda_d) \sim \exp(\Omega_d(\lambda_d)). \quad (6)$$

The above developed models play a crucial role in the development of a decentralized mechanism to regulate interference, which is the ultimate goal of this paper. We now introduce a simulation setup where the robots solve the distributed collection task while using a particular collision avoidance algorithm.

D. Simulation Setup

We validate the theoretical models developed in this paper using a simulated team of differential-drive robots solving the distributed collection task while performing a uniformly ergodic random walk in the domain. The robots deploy minimally-invasive barrier certificates [35] to compute safe velocities when they encounter other robots in the domain.

For a set of simulation parameters ($\lambda = 0.99$, $\delta = 0.11$, $v = 1$), Fig. 3a plots the histogram of the empirically obtained inter-encounter times, alongside the theoretical values

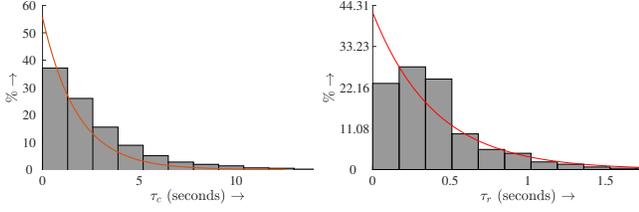


Fig. 3. Histogram of the inter-encounter time τ_c and the encounter resolution time τ_r for a team of simulated robots performing the distributed collection task. In 3a, the inter-encounter time distribution is well-approximated by an exponential random variable with mean given in (2). 3b shows the best-fit exponential distribution to the encounter resolution times. Best-fit exponential parameter: $\rho = 2.44$.

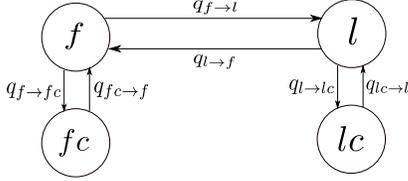


Fig. 4. Each robot can be in four different states when performing the collection task: free (f), free and colliding (fc), loaded with an object (l), loaded and colliding (lc). State transitions are characterized by a continuous-time Markov chain whose transition rates are illustrated.

corresponding to the same density given by (3). Figure 3b plots a histogram of the time taken by robots to resolve encounters and plots that against a best-fit exponential curve (MATLAB `fitdist`: $\rho = 2.44$). As seen in both cases, for the given choice of collision-avoidance algorithm, the empirical distributions are well-approximated by the developed encounter models.

IV. CONTINUOUS-TIME MARKOV MODEL

At a particular time t , any robot can be described as belonging to one of four distinct states defined by $\mathcal{V} = \{f, l, fc, lc\}$: free and ready to pick-up an object (f), free and colliding with another robot (fc), carrying an object (l), and colliding while carrying an object (lc). The possible transitions between states is illustrated in Fig. 4.

Owing to the stochastic nature of the motion of robots and distribution of transport locations, the transition of a robot from one state to another will be stochastic. Let $X(t)$ be the stochastic variable which represents the current state of a chosen robot at time t . It does not matter which robot we pick since they are all identical under the mean-field approximation. The memoryless properties of the inter-robot encounter and object-transport models developed in (3), (4), and (6) imply that the stochastic variable $X(t)$ evolves according to a continuous-time Markov process [39] over the possible states $\mathcal{V} = \{f, l, fc, lc\}$.

The probability per unit time of making a transition from state $i \rightarrow j$, $i, j \in \mathcal{V}$ is characterized by the transition rate $q_{i \rightarrow j}$. Impossible transitions are characterized by $q_{i \rightarrow j} = 0$. The following theorem derives expressions for the transition rates and computes the fraction of time spent by each robot in the different states \mathcal{V} .

Theorem 1: Consider a team of robots performing the distributed collection task as defined in Section III-A. Let $X(t) \in \mathcal{V}$ be the stochastic variable denoting the current state of a given robot at time t . Utilizing the collision as well as object transport models developed in (3), (4), (6), the fraction of time spent by each robot in a state $j \in \mathcal{V}$ can be computed by evaluating the unique stationary distributions

$$\pi_j = \lim_{t \rightarrow \infty} \text{Prob}(X(t) = j), j \in \mathcal{V},$$

whose expressions are given as,

$$\pi_f = \frac{\lambda_d / \lambda_p}{1 + \lambda_d / \lambda_p (1 + \frac{c\lambda}{\rho} (1 + \lambda_p / \lambda_d))},$$

$$\pi_l = \frac{\lambda_p}{\lambda_d} \pi_f; \pi_{fc} = \frac{c\lambda}{\rho} \pi_f; \pi_{lc} = \frac{c\lambda}{\rho} \pi_l,$$

where $c = 4\delta v_r$.

Proof: The transition rates characterizing the CTMC illustrated in Fig. 4 can be defined as follows.

- $q_{f \rightarrow fc}$ and $q_{l \rightarrow lc}$ represent the rates at which a free (or loaded) robot encounters another robot in the domain, which is given by (2).
- $q_{fc \rightarrow f}$ and $q_{lc \rightarrow l}$ represent the rates at which a robot resolves an encounter when it is either free or loaded. From (4), this is simply the parameter ρ .
- $q_{f \rightarrow l}$ and $q_{l \rightarrow f}$ represent the rate at which a robot picks up objects and drops them off, respectively, given by (5).

Given the transition rates $q_{i \rightarrow j}$, $i, j \in \mathcal{V}$, we can compute the stationary distribution of the CTMC by constructing the generator matrix G [39], which is given as,

$$G = \begin{bmatrix} -(c\lambda + m\lambda_p) & m\lambda_p & c\lambda & 0 \\ m\lambda_d & -(c\lambda + m\lambda_d) & 0 & c\lambda \\ \rho & 0 & -\rho & 0 \\ 0 & \rho & 0 & -\rho \end{bmatrix},$$

where $c = 4\delta v_r$ and $m = 2\delta v$. The stationary distribution $\pi = (\pi_f, \pi_l, \pi_{fc}, \pi_{lc})^T$ can be computed by solving the equation $\pi^T G = 0$. The uniqueness of the stationary distribution π follows from the irreducibility and positive-recurrence properties of the embedded discrete-time Markov chain corresponding to the CTMC illustrated in Fig. 4. ■

Since $q_{l \rightarrow f}$ represents the rate at which robots deliver objects and become free, the expected number of objects delivered per robot per unit time, denoted by T , is given as,

$$T(\lambda) = q_{l \rightarrow f} \pi_l = 2\delta v \lambda_d \pi_l(\lambda).$$

For the simulation setup introduced in Section III-D, Fig. 5 plots the total number of objects delivered by the robots per unit time per unit area $T(\lambda)\lambda$, against the density λ (parameters: $v = 0.1$, $\delta = 0.11$, $\lambda_p = 0.82$, $\lambda_d = 0.66$, $\rho = 2.44$). As seen, the actual productivity of the swarm closely matches the theoretically expected output, barring deviations due to edge effects and the stochastic nature of inter-robot encounters. This paper is primarily concerned with the performance of the swarm at intermediate densities—Fig. 5 does not consider very large densities where complete robot jamming can be expected, e.g., [4].

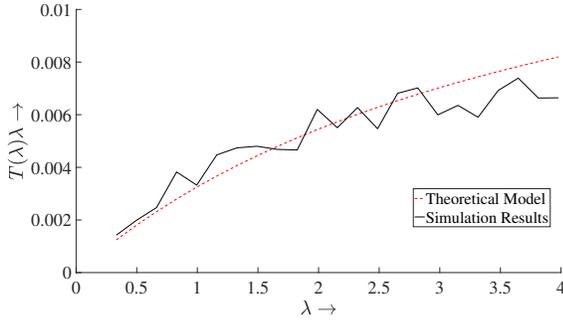


Fig. 5. The theoretical evolution of $T(\lambda)\lambda$ as the density of robots (measured in robots/ m^2) increases is closely matched with the values obtained from simulations with robots employing barrier certificates for collision avoidance. As seen, beyond a certain density, the total output of the swarm increases sub-linearly with the robot density since individual robots spend more time avoiding collisions than transporting objects. This paper is primarily concerned with the performance of the swarm at intermediate densities—the catastrophic jamming occurring at higher densities is not discussed.

Inspired by the sub-linearity in the total output of the swarm at intermediate densities (see Fig. 5), we would like to compute the optimal density of deployed robots so as to achieve a balance between the swarm size (characterized by λ) and the total output of the swarm (characterized by $T(\lambda)\lambda$). Such a trade-off can be encoded in the following optimization problem,

$$\operatorname{argmin}_{\lambda \geq 0} J(\lambda) = \frac{C}{T(\lambda)\lambda} + \lambda,$$

where λ is the density of robots deployed in the domain, and C is chosen to trade-off the cost of deploying robots and the effectiveness of the swarm at performing the task. Setting the gradient of $J(\lambda)$ to zero and solving analytically, we get the optimal density of robots,

$$\lambda^* = \operatorname{argmin}_{\lambda \geq 0} J(\lambda) = \sqrt{\frac{C(1 + \frac{\lambda_d}{\lambda_p})}{2\delta v \lambda_d}}. \quad (7)$$

In the next section, we allow a robot swarm to autonomously regulate its density to the optimal value by enabling individual robots to retreat from the domain when they sense a higher-than-optimal density.

V. DECENTRALIZED DENSITY CONTROL

The primary aim of this section is to develop an algorithm that allows individual robots to retreat from the domain when they detect a higher-than-optimal density. The first part of this section develops an ensemble-level control framework for reducing the robot density. A decentralized implementation is later discussed. We also develop an estimation algorithm that allows each robot in the domain to measure the time-varying density of robots $\lambda(t)$. We assume that robots enter and exit the domain by driving through any section of the boundary of domain \mathcal{D} (as depicted in Fig. 1).

A. Ensemble Closed-Loop Control

Given the density of pick-up and drop-off locations λ_p and λ_d , if the density of robots at time t , denoted as $\lambda(t)$, is

higher than the optimal λ^* , we would like a certain fraction of the robots to leave the domain (we ignore the degenerate case when $\lambda^* = 0$). Let $\epsilon\lambda$ be the number of robots leaving per unit area per unit time. Then, the population dynamics in the domain can be expressed as,

$$\dot{\lambda}(t) = \lambda_{in} - \epsilon(\lambda(t))\lambda(t), \quad (8)$$

where λ_{in} denotes the known and constant number of robots added to the swarm per unit area per unit time (see Section I). The following choice of ϵ feedback-linearizes the above system while driving the density of the swarm to λ^* using a “one-sided” proportional controller,

$$\epsilon(\lambda(t)) = \begin{cases} \frac{1}{\lambda(t)}[\lambda_{in} - K_p(\lambda^* - \lambda(t))], & \lambda > \lambda^* \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$

where K_p is an appropriately chosen proportional gain. A proportional controller was chosen primarily as a simple demonstration of the control mechanism; in general, based on the desired performance requirements, any controller can be used. Applying (9) in (8), we get the ensemble closed loop population dynamics,

$$\dot{\lambda}(t) \equiv \begin{cases} K_p(\lambda^* - \lambda(t)), & \lambda(t) > \lambda^* \\ \lambda_{in} & , \text{ otherwise} \end{cases}. \quad (10)$$

The control parameter $\epsilon(\lambda(t))$ can be interpreted as the probability per unit time with which a robot should leave the domain as a function of the density $\lambda(t)$. Thus, in order for each robot to individually compute ϵ , it must estimate the density of robots in the domain.

B. Estimation

The robots performing the distributed collection task are not equipped with sophisticated sensors, but can detect encounters with other robots as they move through the domain, as defined in Section III-A. Let $e_i(t)$ denote the total number of robots currently in an encounter with robot i . New encounters experienced by robot i can be characterized by an index variable,

$$H_i(t) = \begin{cases} 1, & \text{if } \exists t_1 < t \text{ s.t., } \forall x \in [t_1, t), e_i(x) < e_i(t) \\ 0, & \text{otherwise} \end{cases}.$$

Denote $y_{iL}(t)$ as the total number of encounters experienced by robot i in a time interval $[t - L, t]$, $t \geq L$,

$$y_{iL}(t) = \int_{t-L}^t H_i(t) dt. \quad (11)$$

Given this measurement and the encounter model described by (3), the Maximum-likelihood Estimate (MLE) [40] of the swarm density for robot i at time $t > L$ is,

$$\hat{\lambda}_i(t) = \frac{y_{iL}(t)}{4\delta \frac{d}{\pi} v L}. \quad (12)$$

The measurement horizon L should be chosen so as to achieve a trade-off between accuracy of estimate and the ability of the robot to track changing densities.

C. Decentralized Implementation

Using the maximum likelihood density estimate $\hat{\lambda}_i(t)$, each robot computes its probability of leaving the domain $\epsilon(\hat{\lambda}_i(t))$ using (9). The operations of each robot, executed every dt seconds in software, is illustrated in the following algorithm.

Algorithm 1 Voluntary Retreat Algorithm

- 1: Initialize $\hat{\lambda}_i = 0, k = 0$
 - 2: Given $\lambda_p, \lambda_d, \lambda_{in}$, and C , compute λ^*
 - 3: **for** each time $t = kdt, t > L$ **do**
 - 4: Compute $y_{iL}(t)$ from (11) \leftarrow *detecting encounters*
 - 5: Compute $\hat{\lambda}_i(t)$ from (12)
 - 6: Compute $\epsilon(\hat{\lambda}_i(t))$ from (9)
 - 7: Leave the domain with probability $\epsilon(\hat{\lambda}_i(t))dt$
 - 8: $k = k + 1$
 - 9: **end for**
-

Before deployment, each robot computes the optimal density λ^* using the density of transport locations and influx rate of robots. After an initial wait of L seconds (during which time robots are collecting encounter information), each robot computes the MLE estimate in step 5 and the resulting ϵ in step 6. Finally, the robot flips a biased coin to leave the domain with a probability $\epsilon(\hat{\lambda}_i(t))dt$.

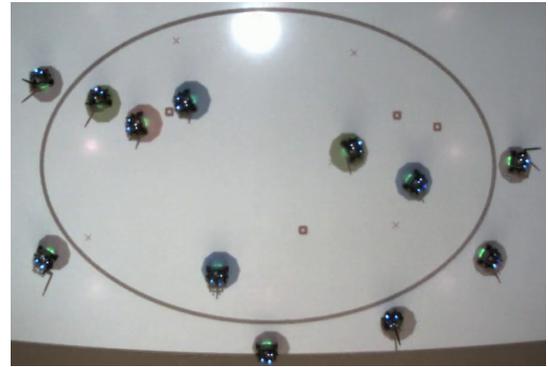
Each robot i will have a different estimate of the density; consequently, $\epsilon(\hat{\lambda}_i(t))$ will vary from one robot to another. While this implies that robots behave differently from each other, we demonstrate in the next section that, for a team of real world robots, the algorithm indeed allows the robots to regulate the density of the swarm to the optimal value.

VI. EXPERIMENTAL RESULTS

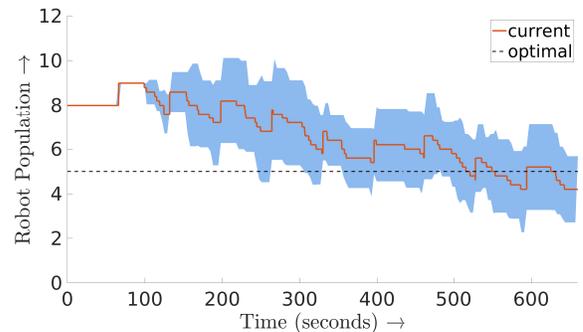
The developed algorithm was deployed on the Robotarium [41], a remotely accessible swarm robotics testbed. The experiment was initialized with 8 robots starting inside the elliptical domain (with $1.2m$ and $0.8m$ as the semi-major and semi-minor axis length) shown in Fig. 6a, deploying minimally-invasive barrier certificates [35] for collision avoidance. An additional 4 robots were initially placed outside and entered the domain at a steady rate. Pick-up (\square) and drop-off (\times) locations are marked on the domain.

For the following parameters: $\lambda_p = 1.32, \lambda_d = 1.32, C = 0.03, \delta = 0.1, v = 0.1$, each robot computes the optimal swarm density using (7). The robots perform an unbiased random walk in the domain while encountering other robots. Using Algorithm 1 (with parameters $L = 90, dt = 0.03, K_p = 0.03, \lambda_{in} = 0.02$), each robot flips a biased coin to decide whether it should stay or retreat.

The experiment was repeated 5 times to average the results, and demonstrate the consistent performance of the voluntary retreat algorithm. Figure 6b plots the mean of the true robot population against the optimal value, along with the standard deviation (represented by the shaded region). As seen, the rate at which robots retreat reduces as the true population approaches the optimal value, in accordance with



(a)



(b)

Fig. 6. Experimental verification of the Voluntary Retreat Algorithm on a team of 12 real robots operating on the Robotarium. An overhead projector is used to project additional information onto the robot arena. The experiment begins with 8 robots inside the elliptical domain seen in 6a. Robots stationed outside enter at a steady rate λ_{in} . Blue and red circles projected around each robot signify *free* (f) and *loaded* (l) robots, respectively. Green circles indicate that a robot has decided to retreat from the domain. These robots drive out of the elliptical domain via any section of the boundary. The robot population data in 6b is averaged over 5 experimental runs to demonstrate the consistent performance of the algorithm, with one standard deviation depicted by the shaded region.

the closed-loop dynamics given in (10). Even with variations in the density estimates of the robots as well as the constant influx of new robots (indicated with upwards jumps in Fig. 6b), the closed loop algorithm allows the population to settle around the optimal value.

VII. CONCLUSION

The technique presented in this paper allows a robot swarm to autonomously curb the negative effects of spatial interference by enabling individual robots to either stay and perform the distributed collection task, or retreat so as to reduce the density. Each robot uses only binary information regarding the presence of other robots around it to estimate the density of the swarm, and decides whether to stay or voluntarily retreat from the domain. We demonstrate that the swarm regulates the density to an optimal value, which is analytically computed to achieve a trade-off between swarm size and overall productivity. Multi-robot experiments demonstrate the efficacy of the developed algorithm.

REFERENCES

- [1] C. Kitts and M. Egerstedt, "Design, control, and applications of real-world multirobot systems [from the guest editors]," *IEEE Robotics & Automation Magazine*, vol. 15, no. 1, pp. 8–8, 2008.
- [2] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *International Workshop on Swarm Robotics*. Springer, 2004, pp. 10–20.
- [3] D. Goldberg and M. J. Mataric, "Interference as a tool for designing and evaluating multi-robot controllers," in *Aaai/iaai*, 1997, pp. 637–642.
- [4] A. Rosenfeld, G. A. Kaminka, and S. Kraus, "A study of scalability properties in robotic teams," in *Coordination of large-scale multiagent systems*. Springer, 2006, pp. 27–51.
- [5] G. Pini, A. Brutschy, M. Birattari, and M. Dorigo, "Interference reduction through task partitioning in a robotic swarm," in *Sixth International Conference on Informatics in Control, Automation and Robotics-ICINCO*, 2009, pp. 52–59.
- [6] H. Hamann, "Superlinear scalability in parallel computing and multi-robot systems: Shared resources, collaboration, and network topology," in *International Conference on Architecture of Computing Systems*. Springer, 2018, pp. 31–42.
- [7] E. Castello, T. Yamamoto, F. Dalla Libera, W. Liu, A. F. Winfield, Y. Nakamura, and H. Ishiguro, "Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach," *Swarm Intelligence*, vol. 10, no. 1, pp. 1–31, 2016.
- [8] D. A. Shell and M. J. Mataric, "On foraging strategies for large-scale multi-robot systems," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 2717–2723.
- [9] R. Beckers, O. E. Holland, and J.-L. Deneubourg, "From local actions to global tasks: Stigmergy and collective robotics," in *Prerational Intelligence: Adaptive Behavior and Intelligent Systems Without Symbols and Logic*. Springer, 2000, pp. 1008–1022.
- [10] A. Martinoli, A. J. Ijspeert, and L. M. Gambardella, "A probabilistic model for understanding and comparing collective aggregation mechanisms," in *European Conference on Artificial Life*. Springer, 1999, pp. 575–584.
- [11] K. Lerman and A. Galstyan, "Mathematical model of foraging in a group of robots: Effect of interference," *Autonomous Robots*, vol. 13, no. 2, pp. 127–141, 2002.
- [12] E. Ostergaard, G. Sukhatme, and M. Mataric, "Emergent bucket brigading," in *Autonomous Agents*, vol. 37, 2001, pp. 2219–2223.
- [13] A. T. Hayes, "How many robots? group size and efficiency in collective search tasks," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 289–298.
- [14] V. Fourcassié, A. Dussutour, and J.-L. Deneubourg, "Ant traffic rules," *Journal of Experimental Biology*, vol. 213, no. 14, pp. 2357–2363, 2010.
- [15] A. Dussutour, V. Fourcassié, D. Helbing, and J.-L. Deneubourg, "Optimal traffic organization in ants under crowded conditions," *Nature*, vol. 428, no. 6978, p. 70, 2004.
- [16] J. Aguilar, D. Monaenkova, V. Linevich, W. Savoie, B. Dutta, H.-S. Kuan, M. Betterton, M. Goodisman, and D. Goldman, "Collective clog control: Optimizing traffic flow in confined biological and robophysical excavation," *Science*, vol. 361, no. 6403, pp. 672–677, 2018.
- [17] D. M. Gordon, *Ant encounters: interaction networks and colony behavior*. Princeton University Press, 2010.
- [18] S. Mayya, P. Pierpaoli, G. Nair, and M. Egerstedt, "Collisions as information sources in densely packed multi-robot systems under mean-field approximations," in *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, July 2017.
- [19] S. Mayya, P. Pierpaoli, G. Nair, and M. Egerstedt, "Localization in densely packed swarms using interrobot collisions as a sensing modality," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 21–34, Feb 2019.
- [20] H. Hamann, "Towards swarm calculus: Urn models of collective decisions and universal properties of swarm performance," *Swarm Intelligence*, vol. 7, no. 2-3, pp. 145–172, 2013.
- [21] Y. Khaluf, C. Pinciroli, G. Valentini, and H. Hamann, "The impact of agent density on scalability in collective systems: noise-induced versus majority-based bistability," *Swarm Intelligence*, vol. 11, no. 2, pp. 155–179, 2017.
- [22] R. C. Arkin, T. Balch, and E. Nitz, "Communication of behavioral state in multi-agent retrieval tasks," in *1993 Proceedings of IEEE International Conference on Robotics and Automation*. IEEE, 1993, pp. 588–594.
- [23] D. Goldberg and M. J. Mataric, "Robust behavior-based control for distributed multi-robot collection tasks." University of Southern California Los Angeles United States, Tech. Rep., 2000.
- [24] R. T. Vaughan, G. S. Sukhatme, and M. J. Mataric, "Go ahead, make my day: Robot conflict resolution by aggressive competition," in *Intl. Conf. on Simulation of Adaptive Behavior (SAB)*. Citeseer, 2000.
- [25] R. Vaughan and A. Lein, "Adaptive multi-robot bucket brigade foraging," *Artificial Life*, vol. 11, p. 337, 2008.
- [26] M. Schneider-Fontán and M. J. Mataric, "A study of territoriality: The role of critical mass in adaptive task division," in *From animals to animats IV*. Citeseer, 1996.
- [27] E. Bonabeau, G. Theraulaz, and J.-L. Deneubourg, "Fixed response thresholds and the regulation of division of labor in insect societies," *Bulletin of Mathematical Biology*, vol. 60, no. 4, pp. 753–807, 1998.
- [28] M. J. Krieger, J.-B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, no. 6799, p. 992, 2000.
- [29] T. H. Labella, M. Dorigo, and J.-L. Deneubourg, "Division of labor in a group of robots inspired by ants' foraging behavior," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 1, no. 1, pp. 4–25, 2006.
- [30] T. W. Mather and M. A. Hsieh, "Synthesis and analysis of distributed ensemble control strategies for allocation to multiple tasks," *Robotica*, vol. 32, no. 2, pp. 177–192, 2014.
- [31] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927–937, 2009.
- [32] L. Bobadilla, F. Martinez, E. Gobst, K. Gossman, and S. M. LaValle, "Controlling wild mobile robots using virtual gates and discrete transitions," in *American Control Conference (ACC)*, 2012. IEEE, 2012, pp. 743–749.
- [33] D. A. Shell, C. V. Jones, and M. J. Mataric, "Ergodic dynamics by design: A route to predictable multi-robot systems," in *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*. Springer, 2005, pp. 291–297.
- [34] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [35] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.
- [36] F. Reif, *Fundamentals of statistical and thermal physics*. Waveland Press, 2009.
- [37] H. Hamann, *Space-time continuous models of swarm robotic systems: Supporting global-to-local programming*. Springer Science & Business Media, 2010, vol. 9.
- [38] R. Russell and T. Urban, "Vehicle routing with soft time windows and erlang travel times," *Journal of the Operational Research Society*, vol. 59, no. 9, pp. 1220–1228, 2008.
- [39] S. Karlin, *A first course in stochastic processes*. Academic press, 2014.
- [40] L. Le Cam, "Maximum likelihood: an introduction," *International Statistical Review/Revue Internationale de Statistique*, pp. 153–171, 1990.
- [41] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1699–1706.