2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Decentralized Stochastic Control of Robotic Swarm Density: Theory, Simulation, and Experiment

Hanjun Li[1], Chunhan Feng[2], Henry Ehrhard[3], Yijun Shen[1], Bernardo Cobos[1], Fangbo Zhang[1], Karthik Elamvazhuthi[4], Spring Berman[4], Matt Haberland[1*], and Andrea L. Bertozzi[1]

***Abstract*— This paper explores a stochastic approach for controlling swarms of independent robots toward a target distribution in a bounded domain. The robot swarm has no central controller, and individual robots lack both communication and localization capabilities. Robots can only measure a scalar field (e.g. concentration of a chemical) from the environment and from this deduce the desired local swarm density. Based on this value, each robot follows a simple control law that causes the swarm as a whole to diffuse toward the target distribution. Using a new holonomic drive robot, we present the first confirmation of this control law with physical experiment. Despite deviations from assumptions underpinning the theory, the swarm achieves the theorized convergence to the target distribution in both simulation and experiment. In fact, simulated and experimental performance agree with one another and with our hypothesis that the error from the target distribution is inversely proportional to the square root of the number of robots. This is evidence that the algorithm is both practical and easily scalable to large swarms.**

I. INTRODUCTION

In order to reduce the cost and complexity of a robotic swarm, individual robots may lack capabilities often assumed to be essential for robot control. We are interested in the control of swarms without a central controller, robot localization information, or communication capabilities. That is, each robot must act independently with only locally measurable information, yet it still must contribute to the group effort.

For instance, consider the proposal of [1] to pollinate crops using a swarm of robotic bees. Even the most advanced insect-sized flying robots are severely weight constrained [2], imposing a tradeoff between the addition of communication or localization modules and even more essential capabilities, such as energy storage capacity. If the density of robots required to pollinate the field is proportional to the density of flowers, and if the local density of flowers can be measured with an inexpensive, lightweight sensor (e.g. airborne chemical or color sensor), the swarm can achieve the desired density using the simple law described in [3] without need for a central controller, radio communication, or GPS.

This paper presents the first experimental validation of this control law[1] and assessment of whether the law is robust to differences between the assumptions of the theory and practical scenarios. Our paper is also the first to demonstrate experimentally that the distribution of the robotic swarm converges to the desired distribution as $\frac{1}{\sqrt{N}}$, where N is the number of robots. While it is known that a pure random walk control yields $\frac{1}{\sqrt{N}}$ convergence of a swarm to the uniform distribution (e.g. [5]), this rate of convergence has not yet been shown for any stochastic control law that directs the swarm toward a specific non-uniform distribution.

This paper is organized as follows. Section II summarizes the mathematical background for the control law. Section III details our simulation methods, introduces the error metric used to assess our results, and presents the results for two target swarm density distributions. Section IV describes the testbed we used, the design of our robot, the design of the experiments, and results for the same two target distributions. Section V compares simulated and experimental results to one another and to our hypothesized convergence rate, and we conclude and discuss future work in Section VI.

II. THEORY

We begin with the result of [3]: assuming that the desired density of a swarm is proportional to a measurable feature of the robot's environment (i.e. "a scalar field"), the swarm will tend toward the desired density if *each robot follows a random walk with speed inversely correlated to the square root of the scalar field at the robot's instantaneous position*.

More precisely, the individual robots are to move according to:

$$d\mathbf{X}(t) = D(\mathbf{X}(t))dW + d\psi(t) \tag{1}$$

where $\mathbf{X}(t) \in \mathbb{R}^2$ is the stochastic process of the robot's position in domain $\Omega \subset \mathbb{R}^2$ at time t, D is a function $\mathbb{R}^2 \to \mathbb{R}$ representing the control law that scales the robot's velocity determined by the standard Wiener process $W(t)$, and $\psi(t)$ is a function[2] that reflects the robot specularly from its boundary $\partial\Omega$. Our specific choice of control law is given by

$$D(\mathbf{x}) = \frac{1}{\sqrt{F(\mathbf{x})}} \tag{2}$$

where F is a scalar field on the domain representing the target distribution, and $\mathbf{x} \in \mathbb{R}^2$ is the robot's position. This unique choice of control law D causes the density of the

*Corresponding author: haberland@ucla.edu
[1]UCLA Department of Mathematics, Los Angeles, CA 90095
[2]Nankai Univerity, School of Physics, Tianjin 300071, China
[3]Grinnell College, Mathematics and Statistics, Grinnell, IA 50112
[4]Arizona State University, School for Engineering of Matter, Transport, and Energy, Tempe, AZ, 85281

The authors gratefully acknowledge the support of the following grants: NSF DMS-1045536, NSF CMMI-1435709, NSF CMMI-1436960, and the Cross-Disciplinary Scholars in Science and Technology Program.

[1] [4] also applies a stochastic swarm control law to physical robots, but it is not concerned with achieving a specific non-uniform swarm distribution.
[2]This function cannot be expressed explicitly; see [6].

swarm to be linearly proportional to the scalar field F for a large number of robots N and time t. That is, it is proven in [3] that if $\rho(\mathbf{x}, t)$ is the probability density function of a random process $\mathbf{X}(t)$ satisfying Equation 1, then

$$\lim_{t\to\infty} \|\rho(\cdot, t) - \rho_\Omega(\cdot)\|_1 = 0 \tag{3}$$

where $\rho_\Omega(\mathbf{x}) = F(\mathbf{x})/||F(\cdot)||_1$ is the target density distribution and the L^1 norm of a function $f(\mathbf{x})$ is defined as

$$||f(\cdot)||_1 = \int_{R^2} |f(\mathbf{x})| d\mathbf{x}.$$

In the case of crop pollination, the scalar field might be the concentration of a chemical emitted by the crop, or simply the intensity of a color that distinguishes the crop from its surroundings. Such a scalar field corresponds to a target swarm distribution with a high concentration of robotic bees over the crop area, and low swarm density elsewhere.

Intuitively, where the scalar field F and thus the desired density of robots is high, the robots will tend to move slowly, or linger. Where the scalar field F and thus the desired density of robots is low, the robots will tend to move quickly, or evacuate. Thus the diffusive behavior of the Wiener process is guided by the simple control law D to achieve the swarm goal.

III. SIMULATION

To provide a performance benchmark for experiment, we simulate the model of Equations 1 and 2 for two target swarm densities represented by the scalar fields of Figure 1.

A. Method

We begin by introducing real-world complexities in the simulation that are not accounted for in the theory of Section II. First, due to the finite clock rate of the microprocessor and, more importantly, the inertia of the robot, Brownian motion must be approximated as a discrete-time continuous-space random walk. Thus for simulation, the differential Equation 1 for each robot takes the form of a difference equation

$$\mathbf{x}^{j+1} = \mathbf{x}^j + \sqrt{2\Delta t} D(\mathbf{x}^j)\mathbf{Z}^j \tag{4}$$

where $\mathbf{x}^j = \mathbf{x}(t_j)$, $\Delta t = t_{j+1} - t_j$ is the (constant) duration of each time step, and $\mathbf{Z}^j$ is a vector of independent, normally distributed random variables with zero mean and unit variance. Note that it will be useful to rewrite Equation 4 as

$$\mathbf{x}^{j+1} = \mathbf{x}^j + v^j \Delta t \hat{\mathbf{Z}}^j \tag{5}$$

where v^j is the speed of the robot during the time step, and $\hat{\mathbf{Z}}^j = \frac{\mathbf{Z}^j}{|\mathbf{Z}^j|}$ is a unit vector in the direction of $\mathbf{Z}^j$.

In addition to discrete-time operation, real robots have a finite maximum speed $v_{\max}$, and thus

$$v^j = \min\left(\sqrt{2}\Delta t^{-\frac{1}{2}} D(\mathbf{x}^j) Z^j, v_{\max}\right), \tag{6}$$

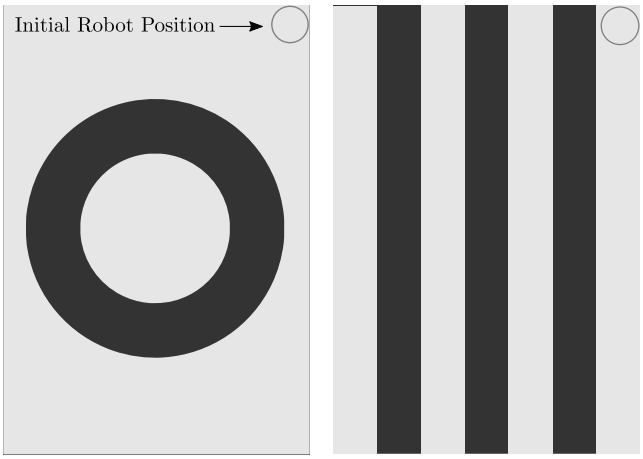
where $Z^j = \left|\mathbf{Z}^j\right|$ is the magnitude of $\mathbf{Z}^j$.

Finally, a real robot with only a local scalar sensor cannot easily determine the orientation of the boundary relative to its current velocity, and thus cannot reflect specularly. Instead, when the robot senses contact with the boundary, it simply reverses direction for the duration of the time step. Therefore, if $\mathbf{x}^{j+1}$ calculated according to Equation 5 is found to be outside the boundary, we calculate a coefficient $\alpha = \frac{\Delta t_a - \Delta t_b}{\Delta t_a + \Delta t_b}$, where Δt_a is the duration until the robot would reach the boundary at its present speed, and $\Delta t_b = \Delta t - \Delta t_a$ is the remainder of the time step. Then

$$v^j = \alpha \min\left(\sqrt{2}\Delta t^{-\frac{1}{2}} D(\mathbf{x}^j) Z^j, v_{\max}\right). \tag{7}$$

Note that although [5] explains the disadvantages of boundary control laws other than specular reflection, it does not consider the boundary control law presented here. Simulation for a uniform desired distribution within a circular boundary suggests that our law does not cause distortion near the edges, as some other boundary control laws do.

The simulation is performed in MATLAB using only standard arithmetic and control flow statements, as the difference equations are inherently explicit. $v_{\max} = 17.5$in/s is chosen to reflect the maximum speed of the physical robot described in Section IV. Because the maximum achievable speed of the physical robot depends on the direction of movement, $v_{\max}$ is really the minimum, over all movement directions, of the maximum achievable speed. The time step $\Delta t = 0.5$s is chosen to be relatively small to avoid error due to time discretization, but large enough to permit the robot to accelerate to each new velocity well within the time step.



(a) **The Ring Pattern.** The inner radius is $r_1 = 11.4$in; outer radius is $r_2 = 20.6$in.

(b) **The Rows Pattern.** Each row has a width of $d = 6.87$in.

Fig. 1: **The Scalar Fields.** Each pattern above specifies a scalar field

$$F(\mathbf{x}) = \begin{cases} 36 & \text{if } \mathbf{x} \in \Omega \cap \Gamma, \\ 1 & \text{if } \mathbf{x} \in \Omega \setminus \Gamma, \end{cases}$$

used in simulation and experiment, where:

- $\Omega = \{\mathbf{x} : x_1 \in [0, w], x_2 \in [0, h]\}$,
- (ring) $\Gamma = \{\mathbf{x} : r_1^2 < (x_1 - \frac{w}{2})^2 + (x_2 - \frac{h}{2})^2 < r_2^2\}$,
- (rows) $\Gamma = \{\mathbf{x} : x_1 \in [d, 2d] \cup [3d, 4d] \cup [5d, 6d]\}$.

with (x_1, x_2) Cartesian coordinates of $\mathbf{x}$, $w = 48$in and $h = 70$in. In short, the goal of the controller is to achieve a swarm density $36\times$ higher in dark regions than in light regions.

B. Error Metric

As the simulation is defined in terms of a finite number of individual robots rather than robot density, it is not immediately obvious how to define a metric for how well the robot swarm matches the target distribution. A naïve method would involve discretizing the simulation space into bins, calculating the number of robots in each bin, and comparing these to the desired values calculated from the target distribution. This is not a good error metric, however, because the choice of bin size is arbitrary. Furthermore, the metric approaches zero (no error) as the bin size approaches the entire space, and the metric approaches unity (or complete error) as the bin size is reduced to zero. A more useful error metric follows from replacing the position of the robot, a single point, with a distribution representing how well it performs its task in space. In particular, we center a Gaussian function at the i^{th} robot's position $\mathbf{x}_i(t)$ at time t,

$$G_i^\delta(\mathbf{x}, t) = \frac{1}{2\pi\delta^2} \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_i(t)|^2}{2\delta^2}\right), \tag{8}$$

where δ represents the radius of the region in which the robot effectively performs its task. We combine the Gaussians for each robot into a single function and normalize the result[3]

$$\rho_N^\delta(\mathbf{x}, t) = \frac{1}{N} \sum_{i=1}^{N} G_i^\delta(\mathbf{x}, t), \tag{9}$$

where N is the number of robots. This 'Gaussian blob function' gives a continuous representation of how well the robots cover the space at a given time. We calculate an error metric by comparing the Gaussian blob function to the target density distribution using the L^1 norm,

$$e_N^\delta(t) = \int_\Omega \left|\rho_N^\delta(\mathbf{x}, t) - \rho_\Omega(\mathbf{x})\right| d\mathbf{x}. \tag{10}$$

The metric makes conceptual sense as in most applications an individual robot will not fulfill its task only at a single point but rather over some area, and the robot's ability to perform its task diminishes further from the robot's center. The parameter δ is meaningful because it describes the robot's effective task space.

The upper bound for this error metric is 2. This is approached for any number of robots N only when δ is small, all robots are located in some region where the target density is zero, and no robots are located where the target density is nonzero. The lower bound for this error metric is zero. However, for a general target distribution, this can only be approached when δ is small and N is large[4]. More precisely, we hypothesize that $\forall \epsilon > 0$

$$\lim_{t\to\infty} \lim_{\delta\to 0} \lim_{N\to\infty} P(e_N^\delta(t) < \epsilon) = 1. \tag{11}$$

That is, $e_N^\delta(t)$ converges in probability to 0 as $t \to \infty$, $\delta \to 0$, and $N \to \infty$.

[3]We assume δ is small enough relative to Ω such that the integral of each Gaussian over Ω is nearly 1.

[4]Of course, this can also be achieved if the desired distribution can be represented exactly as the sum of N Gaussians with the given δ.

Of course, for finite t and N and nonzero δ, the value of the error will be somewhere between 0 and 2. We can consider three components of the error separately. The error due to t arises because the control law needs time to guide the robots from their initial distribution to steady state. The error due to δ can be likened to the difficulty in painting a detailed image with a large paintbrush. Finally, the error due to N can also be called the 'sampling error': by the Central Limit Law, for a fixed $\mathbf{x}$ and t, $\sqrt{N}\left(\rho_N^\delta(\mathbf{x}, t) - \rho(\mathbf{x})\right)$ converges in distribution to the standard normal distribution $\mathcal{N}(0, 1)$. Hence we conjecture that the sampling error will vary linearly with $\frac{1}{\sqrt{N}}$, which is common for similar scenarios in the literature (e.g. [5], [7]). We will assess whether our experimental results confirm this in Section V.

When calculating $e_N^\delta(t)$, the values of N and t are given but δ must be chosen. Robotic bees for pollination would likely have very short manipulator arms, or none at all, and so the size of their task space would not be much greater than that of their bodies. Hence, we choose $\delta = 2$in, the approximate radius of our holonomic drive robot.

The level of this error metric that corresponds with acceptable performance also depends on the particular task, but we seek to interpret the values of the error metric achieved by a swarm of N robots in a task-independent way. Two approaches are to compare the value of the error metric achieved by the robot swarm to:

- the probability density function of error metric values that follows from random sampling of the robot positions from the desired distribution and
- the global extrema, over all possible robot configurations, for the error metric.

The approximate probability density function of error metric values for the ring distribution is shown in Figure 2. However, finding the global extrema may not be feasible for an arbitrary number of robots N and desired distribution, so we use instead the highest and lowest values of the error metric we have found by thoughtful trial and error.

C. Results

Figure 3 shows sample configurations and density distributions, as measured by the Gaussian blob function ρ_N^δ, achieved in simulation of $N = 200$ robots for both of the target distributions of Figure 1.

Despite deviations from theoretical assumptions, the swarm of robots still tends toward a distribution proportional to the scalar field. For the target distribution as the ring pattern, the minimum error value achieved in simulation, $e_N^\delta = 0.4295$, lies at 7.2% of the range between the minimum error value 0.3089 and maximum error value 1.9956 achieved by manual placement. For the target distribution as the row pattern, the error value $e_N^\delta = 0.5366$ is within 12.5% of the minimum error value 0.3549, considering the maximum error value of 1.8115. More detailed analysis of the convergence to the desired distribution and comparison of results to Figure 2 will be provided in Section V.

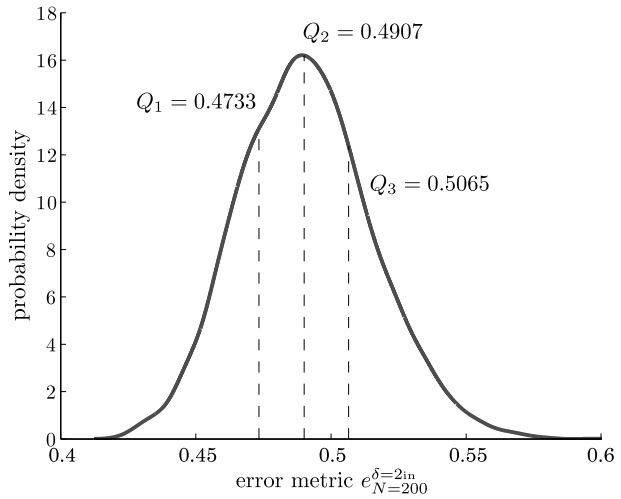
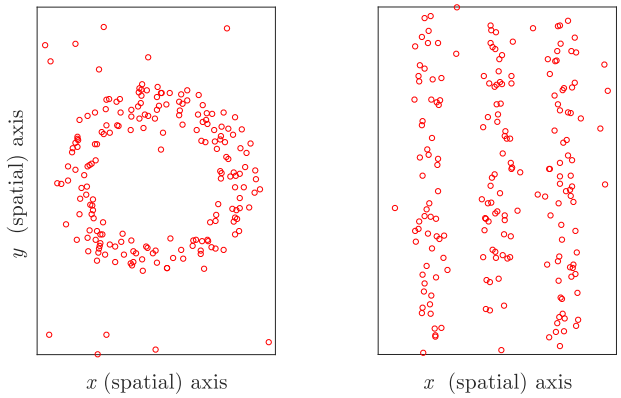
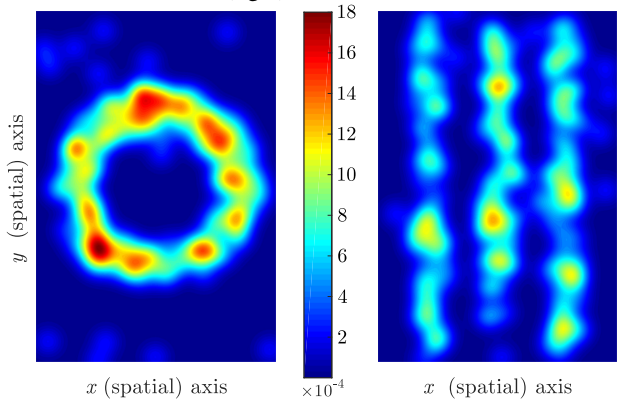


Fig. 2: **Probability density function of the error metric for robot configurations randomly sampled from the ring distribution.** This curve was generated by randomly sampling $N = 200$ robot positions from the ring distribution, calculating the value of the error metric with $\delta = 2$in, and repeating 100000 times. The MATLAB routine `ksdensity` was used to approximate the continuous probability density function from the resulting histogram. Lines for the 25^{th}, 50^{th}, and 75^{th} percentiles are labeled Q_1, Q_2, and Q_3, respectively. In Section V, we will compare the values of the error metric achieved by the control algorithm to this probability density function to assess whether the performance is acceptable.



(a) Robot positions at $t = 601.5$s for ring distribution with $N = 200$ robots (left) and at $t = 361.5$s for row distribution with $N = 200$ robots (right).



(b) Gaussian blob function $\rho^{\delta=2\text{in}}_{N=200}(t = 601.5\text{s})$ for ring distribution (left) and $\rho^{\delta=2\text{in}}_{N=200}(t = 361.5\text{s})$ for row distribution (right).

Fig. 3: **Sample simulation results for both target distributions.** The times t are those at which the error metric e^{δ}_{N} is minimal over a simulation 800s long. The robot positions and Gaussian blob function, with $\delta = 2$in and $N = 200$ robots, show that the control law guides robots toward the target distribution.

IV. EXPERIMENT

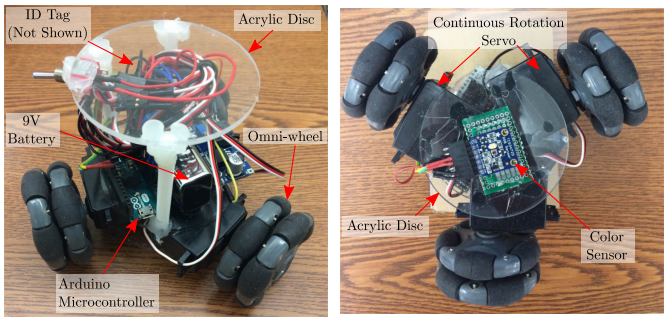
To test the robustness of the control law in a real world scenario, we designed and built a new holonomic drive robot (i.e. one that can translate in any direction independent of its orientation, [8]), programmed it to follow the control laws used in the simulation, and performed many trials with one robot to reveal the behavior of a swarm of many robots. In our spatial coverage scenario, it is possible to experimentally measure swarm behavior using only a single robot because the control law requires no interaction among robots, and the assumption that there are no collisions is mostly valid in the case of small robots in a relatively large domain (e.g. robotic bees pollinating a field).

A. Methods

1) Testbed: Large format prints of the scalar field patterns of Figure 1 were placed on the floor of the testbed. The print is protected by a thin, transparent sheet of plastic. The plastic is secured to the floor at the edges with thick strips of red tape, which also mark the boundary of the testbed.

2) Robot: The chassis of the robot consists of three Parallax High Speed Continuous Rotation Servos (Product ID 900-00025) arranged at equal angles and sandwiched between 3in diameter, 0.125in thick acrylic discs. Each servo drives a 58mm Nexus Robot omni-wheel (RB-Nex-57), which rolls like a typical wheel but also slides freely relative to the floor along the axis of rotation. The servos are controlled via 8-bit pulse width modulation (PWM) by an Arduino Micro (Atmel ATmega32u4 with additional components, developed by Adafruit Industries and Arduino) programmable microcontroller. A 9V lithium-ion rechargeable battery powers the Arduino directly and is regulated to 6V by a 1.25V-35V 3A Adjustable Step-Down Voltage Regulator (RB-See-365) to power the servos. An additional acrylic disc supported by three $1/4$in diameter PTFE rods is used to attach an identification tag. For easy assembly and disassembly, all electronic components are attached to the robot chassis using self-locking mushroom head fastener tape and all structural components are bonded with hot melt glue. A hand-soldered perfboard, connected to components using ribbon cable and 0.1in breakaway headers, distributes signals and power. The complete robot, minus the identification tag, is shown in Figure 4.

The scalar field and boundary are sensed by a digital



(a) View from above (b) View from bottom

Fig. 4: **The Kiwi-drive Robot.** For clarity of other components, identification tag visible in Figure 5 is not shown.

RGB color sensor (TCS34725) fixed to the underside of the chassis, which illuminates the scalar field print with white LEDs and measures the red, green, and blue components of the reflected light. The robot distinguishes the scalar field from the boundary by color: black represents a high value of the scalar field, white represents a low value of the scalar field, and red represents the boundary.

3) Data Collection: Two overhead cameras record the motion of the robot. We process the video with OpenCV [9] in Python to track the position of the centroid of the robot through time. We found the position measured by computer vision tracking to be within 2cm of ground measurement for a variety of robot positions throughout the space.

4) Experiment Design: We placed the robot in the specified initial position on top of the scalar field print, started recording data with the overhead cameras, and initiated the control algorithm on the robot. After 180s[5], the video recording stops automatically, and we manually turned the robot off. This process was repeated 200 times for each of the scalar field patterns.

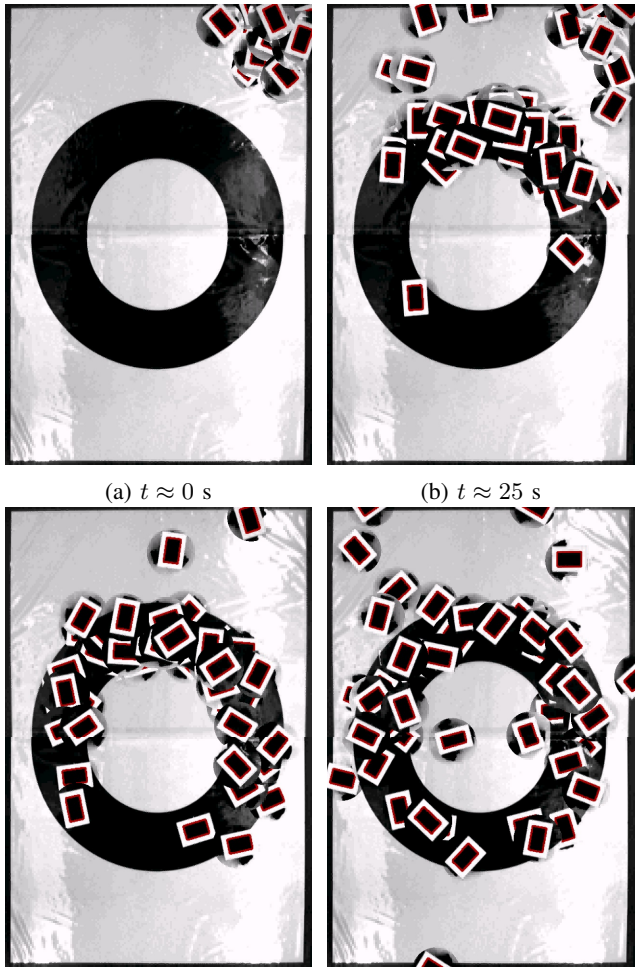
B. Results

In addition to bounded speed and the modified boundary control law, another deviation between the physical robot and theoretical assumptions is imperfect control of velocity due to limited control signal (PWM) precision, nonlinear response of the motors with respect to control signal, wheel slipping, and robot inertia. In spite of this, the ensemble of robot final positions in physical experiment achieves the target distribution, and results of physical experiment match those of simulation very closely.

Figure 5 depicts the convergence of the 'swarm' to the desired distribution through time-lapse images. Figure 6 shows sample configurations and density distributions, as measured by the Gaussian blob function of Equation 9, achieved in physical experiment with $N = 199$ robot runs[6]

[5]This was chosen to be 20% longer than the 150s required for steady state to be achieved in early simulations for the ring pattern. Unfortunately, this was before the maximum speed of the robot was accurately measured, so experiments only capture transient behavior.

[6]Data from one of the robot runs was found to be invalid after experimentation was complete.

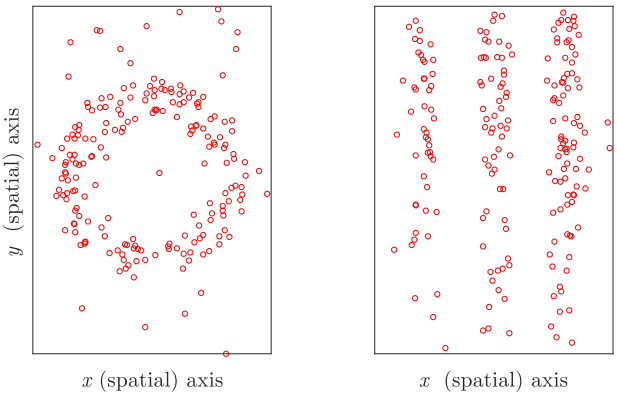


(a) $t \approx 0$ s (b) $t \approx 25$ s

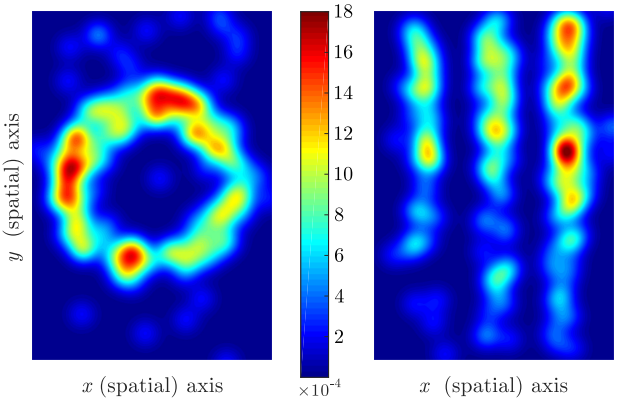
(c) $t \approx 80$ s (d) $t \approx 170$ s

Fig. 5: **Sample robot configurations for ring target distribution.** As results were obtained by running one robot 200 times rather than 200 robots once, each frame above was produced by sampling a random subset of 50 runs and superposing images taken by the overhead cameras during those runs. The robot, as seen in Figure 4, is obscured by an identification tag, visible as a solid black rectangle within a white rectangle. The thin line outlining the identification tag was generated in post-processing, and the circular 'shadow' around each robot is an artifact of the (selective) superposition of multiple images.

for the ring pattern and $N = 200$ robot runs for the row pattern. For the target distribution as the ring pattern, the minimum error value achieved in experiment, $e_N^\delta = 0.4816$, lies at 10.2% of the range between the minimum error value 0.3089 and maximum error value 1.9956 achieved by manual placement. For the target distribution as the row pattern, the error value $e_N^\delta = 0.6477$ is within 20.1% of the minimum error value 0.3549, considering the maximum error value of 1.8115. These error numbers are higher in experiments than simulation because the experiment did not run for as long as the simulation and did not have a chance to settle into a steady state. Nonetheless, it is clear that the control law



(a) Robot positions at $t = 175.5$s for ring distribution with $N = 199$ robots (left) and at $t = 160.5$s for row distribution with $N = 200$ robots (right). The exceptionally careful reader will note that there are only 198 robot markers on the row distribution plot on the right, not 200. This is because in two trials, the robot was slightly outside the boundary (and in the process of 'bouncing' back in) when the robot positions were recorded.



(b) Gaussian blob function $\rho^{\delta=2\text{in}}_{N=199}(t = 175.5\text{s})$ for ring distribution (left) and $\rho^{\delta=2\text{in}}_{N=200}(t = 160.5\text{s})$ for row distribution (right).

Fig. 6: **Sample results from physical experiment for both target distributions.** The times t are those at which the error metric e^{δ}_{N} is minimized. The robot positions and Gaussian blob function show that the control law guides the robots toward the target distribution. However, the experiments did not run long enough to achieve steady state. This is particularly apparent for the row distribution as the concentration in the top right, where all robots began, remains higher than elsewhere.

tended to produce the desired distribution, and we will show in the following section that experiment agreed remarkably well with simulation with respect to the transient behavior.

V. DISCUSSION

A. Time Convergence

Figure 7 shows the value of the error metric through time for both simulation and physical experiment with the target distribution as the ring pattern. Note that maximum robot speed is the only constant measured from the physical robot

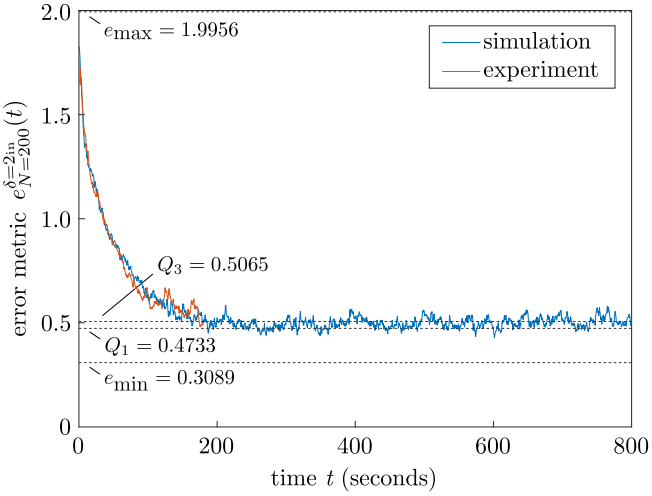


Fig. 7: **Error metric e^{δ}_{N} over time for both simulation and physical experiment with the target distribution as the ring pattern**. Lines for the minimum and maximum error achieved by trial and error (as described in Section III-B), labeled $e_{\min}$ and $e_{\max}$, and for the 25^{th} and 75^{th} percentile errors from Figure 2, labeled Q_1 and Q_3, are included for comparison.

that is used as an input to the simulation; no other constants are used to tune the simulation to match experiments. Nonetheless, the time rate of convergence between the two appear indistinguishable within the stochastic nature of the control law.

As physical experiment ends at $t_f = 180$s but agrees remarkably well with simulation until then, the 'steady state' behavior is studied only in simulation. Note that the error values mostly lie between the lines labeled Q_1 and Q_3, that is, the 25^{th} and 75^{th} percentile error values when robot configurations are randomly sampled from the target distribution (originally presented in Figure 2). This suggests that the steady state coverage of this control law is as good as can be expected, given its stochastic nature.

B. N-Convergence

To test the convergence of the error metric as the number of robots increases, we calculated the error metric for subsets of the trials performed with the ring distribution, specifically $e^{\delta=2\text{in}}_{n}(t)$ for five values of n (10, 20, 40, 80, and 190). Rather than selecting a single value for t, we used the average of $e^{\delta=2\text{in}}_{n}(t)$ over the last 60 seconds. Also, rather than selecting a single subset of size n from our N robot trials (199 for experiment, 200 for simulation), we calculated the error statistic for $\lfloor \frac{199}{n} \rfloor$ disjoint subsets and averaged the results. Figure 8 shows that in simulation this averaged error metric $\bar{e}^{\delta=2\text{in}}_{n}$ drops linearly with respect to $\frac{1}{\sqrt{n}}$ as expected; the coefficient of determination for the best fit line is 0.99. Results from physical experiment agree, although because steady state was not achieved in experiment, the error metric values are always slightly higher for experiment than simulation.

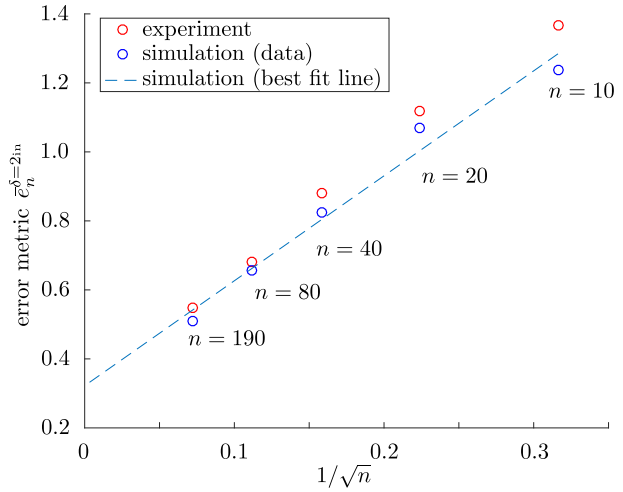


Fig. 8: **Relationship between the error metric $\bar{e}_n^\delta$ and number of robots** n ($\delta = 2$in, ring pattern). Note that the error tends toward a nonzero value as n increases. This is primarily due to the nonzero δ, as discussed in Section III-B. Nonetheless, the decrease is quite linear with respect to $\frac{1}{\sqrt{n}}$, as expected.

Combined with the time-convergence results, this gives confidence that the control law is correctly implemented in both simulation and experiment, and furthermore, that despite conditions of the theory not being met, the control law itself scales to large numbers of robots and is robust enough to work effectively under real-world conditions.

C. Future Work

We have assumed that the robots are small relative to the environment or that collisions between robots are inconsequential and would not affect the net behavior of the swarm, given its stochastic nature. However, we would like to consider the case in which collisions must be avoided to prevent damage to the robots. If each robot were equipped with a rangefinder to sense the relative location of other robots, the addition of suitable repulsive "forces" between all nearby robots would prevent collision while preserving the diffusive nature of the swarm. However, we have found in preliminary simulation that this can slow convergence to the desired distribution and increase the steady state error, especially in regions where high density is desired, so additional work is necessary.

The simulation and experiment have relied on the robots having a holonomic drive system, that is, the robot can translate in any direction independent of orientation. Application of the control algorithm to non-holonomic robots, such as those that steer like tanks or cars, may be relatively straightforward: the *average* velocity of the robot during a time step is calculated as for the holonomic robot, but a path to the destination defined by the average velocity must be planned. We have performed simulation and preliminary physical experiments with a tank-drive robot that simply rotates toward the destination before following a straight-line path to it, but additional work is required to confirm that this is the most efficient approach.

While this control law might be sufficient to drive robot bees close to target flowers, robots would need to switch to a different control law, such as that of [10], to land on flowers and perform pollination work. Future work includes implementing robot behavioral switches between "active" (flying) and "passive" (pollinating) states.

Note that theory regarding convergence of the control law is valid even when the desired distribution is not binary. In preliminary simulations with a smoothly varying scalar field, the algorithm still seems robust. The controller is also valid when the robot cannot sense a scalar field from its environment directly, but instead knows its own location and thus the corresponding value of a pre-assigned scalar field. For example, the strategy could be used for public safety operations in which robots patrol with density proportional to a known or expected threat. These possibilities should be tested in future experiments.

VI. CONCLUSION

We have presented experimental validation of simple, decentralized, stochastic control law that guides a swarm of entirely independent robots toward a desired distribution. Despite significant discrepancies between the experimental platform and the theoretical assumptions upon which the controller is based, the swarm achieves the desired distribution in practice at time rates comparable to simulation. The experimental results even show that the asymptotic error of the swarm is proportional to $\frac{1}{\sqrt{N}}$, where N is the number of robots, as hypothesized. This performance and robustness suggest that the control law would be an effective choice for distributing large swarms of inexpensive robots with limited sensing and no communication or localization abilities.

REFERENCES

[1] S. Berman, V. Kumar, and R. Nagpal, "Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 378–385.

[2] K. Y. Ma, P. Chirarattananon, S. B. Fuller, and R. J. Wood, "Controlled flight of a biologically inspired, insect-scale robot," *Science*, vol. 340, no. 6132, pp. 603–607, 2013.

[3] K. Elamvazhuthi, C. Adams, and S. Berman, "Coverage and field estimation on bounded domains by diffusive swarms," in *IEEE Conference on Decision and Control (CDC)*, Las Vegas, NV, 2016, pp. 2867–2874.

[4] A. Prorok, N. Correll, and A. Martinoli, "Multi-level spatial modeling for stochastic distributed robotic systems," *International Journal of Robotics Research*, vol. 30, no. 5, pp. 574–589, 2011.

[5] P. Szymczak and A. Ladd, "Boundary conditions for stochastic solutions of the convection-diffusion equation," *Physical review E*, vol. 68, no. 3, p. 036704, 2003.

[6] H. Tanaka *et al.*, "Stochastic differential equations with reflecting boundary condition in convex regions," *Hiroshima Mathematical Journal*, vol. 9, no. 1, pp. 163–177, 1979.

[7] G. A. Pavliotis, *Stochastic processes and applications*, ser. Texts in Applied Mathematics. Springer-Verlag, 2014.

[8] R. Rojas and A. G. Förster, "Holonomic control of a robot with an omnidirectional drive," *KI-Künstliche Intelligenz*, vol. 20, no. 2, pp. 12–17, 2006.

[9] Itseez, "Open source computer vision library," http://www.opencv.org/, 2015.

[10] M. Graule, P. Chirarattananon, S. Fuller, N. Jafferis, K. Ma, M. Spenko, R. Kornbluh, and R. Wood, "Perching and takeoff of a robotic insect on overhangs using switchable electrostatic adhesion," *Science*, vol. 352, no. 6288, pp. 978–982, 2016.