2018 IEEE 14th International Conference on Control and Automation (ICCA)
June 12-15, 2018. Anchorage, Alaska, USA

Modelling Emergent Swarm Behavior Using Continuum Limits for Environmental Mapping

Megan Emmons[1], Anthony A. Maciejewski[1], and Edwin K. P. Chong[1]

Abstract— **Robotic swarms are comprised of simple, individual robots but can collectively accomplish complex tasks through frequent interactions with other robots and the environment. One pertinent objective for swarms is mapping unknown, potentially hazardous environments. We show that even without communication or localization, the emergent behavior of a swarm observed at one area can be used to infer the presence of obstacles in an unknown environment. The main body of this work focuses on how partial differential equation (PDE) models of emergent swarm behavior can be derived by applying continuum limits to approximate discrete-time rules for individual robots in continuous-time. We illustrate our approach by demonstrating how obstacles can be located by comparing swarm observations to a base library of PDE models. As supported in this work, the PDE models accurately capture identifying characteristics of the emergent behavior and are solved in a few seconds allowing for fast feature identification.**

I. INTRODUCTION

Swarms are a relatively new focus area within the robotics community but take inspiration from well-observed biological systems. Like a colony of ants, a robotic swarm is comprised of simple individual robots that can collectively accomplish complex tasks via frequent interactions with other robots and the environment [1]. Multi-robot systems provide an increased level of redundancy and may outperform individual, more sophisticated robots by working cooperatively to accomplish a given task [2]. Leveraging the number of robots, swarms have a variety of applications in surveillance, medical treatment, and exploration [3]. Swarms have additional advantages in exploration; very simple, inexpensive robots can quickly reveal important features of an unknown environment, as shown in this paper.

Even in a worst-case scenario where robots within a swarm are reduced to random motion with no communication, their distribution can reveal important environmental information. The shape of an environment influences how robots are dispersed through the domain so observing the density of robots at one location can provide information about features throughout the environment, particularly the presence of obstacles. By way of analogy, the position of downstream blockages in a river can be inferred by observing the flow of water upstream. The identification requires no communication between robots and only local, rather than global, knowledge—making the system fully scalable. Identifying features becomes a matter of developing appropriate models and matching observed swarm behavior with the corresponding model. During the identification process, the emergent behavior of a swarm is more informative than individual behaviors, just as the river flow is more significant than individual water droplets.

For physical implementations, individual robots are programmed so local behaviors are known. The emergent behavior can be observed from a physical implementation or modeled through simulation but both methods are extremely time consuming, especially as swarm and environment size increase. In this paper, we propose a methodology to map discrete-time, probabilistic behaviors of individual robots to a partial differential equation (PDE) model of the emergent behavior. The PDE model is quickly solved to determine the emergent behavior at any desired time and position and thus allows for fast feature identification in an unknown environment.

This paper is organized as follows. In Section II, a brief summary of related work is presented before we introduce the continuum limit methodology in Section III. A detailed derivation of the PDE model for a one-dimensional (1D) random walk scenario is presented first to provide a theoretical foundation. The 1D model also allows for more intuitive visualization and validation of the proposed methodology. We then apply the same methodology to a two-dimensional (2D) environment. Validation of both 1D and 2D models is given in Section IV. In Section V we provide a preview into how the PDE models can be used to infer the boundary-types of a simulated environment. The preview includes a illustrative simulated scenario. Finally, we conclude the work in Section VI and present some directions for future work.

II. RELATED WORK

A detailed review of early research in swarm robotics is provided by Navarro and Matía [1] where they also present important characteristics of swarm robotics. Citing Şahine [4], Navarro and Matía consider a swarm to be comprised of many autonomous robots with only local sensing capabilities. Another key feature is scalability of the system. The proposed methodology herein encompasses the swarm definition by assuming robots with no localization or communication.

Many swarm researchers have focused on designing control strategies to produce a desired emergent behavior. Strategies include the use of event-triggered controllers [5], chemical reaction models [6], [7], and potential fields [8]. These strategies all focus on developing rules for individual robots, in essence taking a top-down approach by dictating local rules to produce a desired final state.

[1]Emmons, Maciejewski, and Chong are with the Dept. of Electrical and Computer Engineering, Colorado State University. They can be contacted at: mremmons@rams.colostate.edu, aam@colostate.edu, edwin.chong@colostate.edu

Some researchers have linked partial differential equation (PDE) models to emergent behavior of the swarm. Elamvazhuthi and Berman [9] use a set of advection-diffusion-reaction PDEs to model swarm behavior but still take a top-down approach. The parameters of the PDE are optimized to achieve a desired emergent behavior and then the individual robots behave according to the tuned PDE.

This paper proposes a reverse approach. Rather than imposing a high-level control strategy and influencing individual robot actions to generate a desired distribution, we propose a methodology to map individual robot actions to the natural emergent behavior generated in varying environments. Berger et al. [10] similarly take a bottom-up approach to swarm modeling by proposing the use of compressive subspace learning to identify emergent behavior. The subspace learning still lacks a direct correlation between individual robot actions and the emergent behavior. It also does not provide information about environmental features and requires extensive computations.

The methodology proposed herein provides a direct mapping between local and emergent behaviors for varying environments. A PDE model is obtained based on local behavior rules. The boundary conditions of the PDE model encode environmental features—namely open-space or obstruction. The model derivation is based on the work of Zhang et al. [11] who apply continuum limits to derive PDE models for large-scale wireless networks.

III. Continuum Limit Methodology

A. Derivation of 1D Random Walk Model

To fully illustrate the proposed methodology, we start with a one-dimensional (1D) environment where robots can only move left or right. In this scenario, the robots are performing a random walk. The limited capability of the robot model demonstrates how even simple robots can be used to map unknown environments and represents a worse-case scenario in the spectrum of robot capabilities. The environment itself is composed of N bins evenly distributed in a one-dimensional line, $D = (0, 1)$, where N specifies the desired spatial resolution of the environment. Two additional bins, denoted $n = 0$ and $n = N + 1$ for the left and right-most bins respectively, map to the boundaries of the interval. There are M robots distributed in the N interior bins. At every discrete-time step, each robot randomly chooses to move left or right one bin with probability P_L or $(1 - P_L)$ respectively. As a result, the number of robots in each bin, the *bin occupancy*, varies over time according to a random process.

Let $\mu_{n,k}$ be the occupancy of bin n at time k. We wish to characterize the dynamics of $\mu_{n,k}$ over time. The value of $\mu_{n,k+1}$ depends on the number of robots moving from adjacent bins and hence is a random function of $\mu_{n-1,k}$ and $\mu_{n+1,k}$. To precisely characterize the evolution of bin occupancies over time, we use the method in [11] and define $\vec{\mu}_k = [\mu_{1,k}, \ldots, \mu_{N,k}]$ as the vector of bin occupancies at time k. Assume that for each robot, the choice to move left or right is independent over robots, time, and also of $\vec{\mu}_k$. Taking the conditional expectation of $\mu_{n,k+1}$ given $\vec{\mu}_k$, we obtain

$$E[\mu_{n,k+1}|\vec{\mu}_k] = P_L \mu_{n+1,k} + (1 - P_L)\mu_{n-1,k}. \quad (1)$$

The right-hand side of (1) has the form $f(\vec{\mu}_k)$. From this expression, we can write the mean-field equation, similar to the process in [11],

$$\vec{m}_{k+1} = f(\vec{m}_k), \quad (2)$$

the n^{th} component of which is

$$m_{n,k+1} = P_L m_{n+1,k} + (1 - P_L)m_{n-1,k}. \quad (3)$$

For simplicity, it is assumed robots move left or right with equal probability so

$$P_L = \frac{1}{2}, \quad (4)$$

and the mean occupancy of bin n at time $k + 1$ reduces to

$$m_{n,k+1} = \frac{1}{2}[m_{n+1,k} + m_{n-1,k}]. \quad (5)$$

The mean change in the occupancy of bin n between discrete-time steps k and $k + 1$ can be expressed as

$$\Delta m_n = \frac{1}{2}[m_{n+1,k} + m_{n-1,k}] - m_{n,k}. \quad (6)$$

Equation (6) represents the discrete-time change in mean bin occupancy for interior bins along a one-dimensional line. To map the discrete-time model to continuous time, the following substitutions are made:

$$\begin{array}{rcl} n \pm 1 & \rightarrow & s \pm \Delta s \\ k + 1 & \rightarrow & t + \Delta t. \end{array} \quad (7)$$

The continuous-time model for a one-dimensional random walk becomes

$$\Delta m(s, t) = \frac{1}{2}[m(s + \Delta s, t) + m(s - \Delta s, t)] - m(s, t). \quad (8)$$

The continuous-time model can be represented by a partial differential equation by first performing a second-order Taylor Series expansion where

$$\begin{aligned} m(s \pm \Delta s, t) = \quad & m(s, t) \pm \tfrac{\partial m(s,t)}{\partial s}\Delta s \\ & + \tfrac{1}{2}\tfrac{\partial^2 m(s,t)}{\partial s^2}\Delta s^2 + o(\Delta s^2). \end{aligned} \quad (9)$$

Substituting (9) into (8) and performing algebraic simplifications, the continuous-time model then becomes

$$m(s, t + \Delta t) - m(s, t) = \frac{1}{2}\frac{\partial^2 m(s, t)}{\partial s^2}\Delta s^2 + o(\Delta s^2). \quad (10)$$

Now let $\Delta t = \Delta s^2$ and divide both sides of (10) accordingly to reach

$$\frac{m(s, t + \Delta t) - m(s, t)}{\Delta t} = \frac{1}{2}\frac{\partial^2 m(s, t)}{\partial s^2} + \frac{o(\Delta s^2)}{\Delta s^2}. \quad (11)$$

Taking the limit as $\Delta t \to 0$, the continuous-time model converges to the standard heat equation

$$\frac{\partial m(s, t)}{\partial t} = \frac{1}{2}\frac{\partial^2 m(s, t)}{\partial s^2} \quad (12)$$

for interior bins. The interior model of (12) becomes increasingly accurate as the variables of interest, namely environment size (N) and number of robots (M), approach infinity (see [11]). To fully define the PDE solution, it is necessary to determine boundary conditions and define corresponding initial conditions.

1) Boundary Conditions for 1D Random Walk: Equation (12) describes the one-dimensional random walk scenario for internal bins, $n \in [1, N]$, in continuous time but does not address the boundary conditions which are represented in discrete time by bins 0 and $N+1$. To derive continuous-time rules for the boundary bins, it is important to note that all internal bins must follow the rule described by (5). Let general bin a represent a boundary bin, b the adjacent internal bin, and c the internal bin adjacent to b.

Two different types of boundaries are considered: sinks and walls, roughly corresponding to open exits and obstacles, respectively. When a robot chooses to move into bin a where a is a sink, the robot is removed from the system. The discrete-time model for the population of bin b at time $k+1$ is therefore

$$m_{b,k+1} = \frac{1}{2}m_{c,k} \tag{13}$$

but the internal rule requires

$$m_{b,k+1} = \frac{1}{2}m_{a,k} + \frac{1}{2}m_{c,k}. \tag{14}$$

The two equations can both be satisfied by choosing

$$m_{a,k} = 0 \ \forall k. \tag{15}$$

Physically, the discrete rule means the sink boundary bin should have an occupancy of zero at every discrete-time step which agrees with the initial intent of the model–robots that enter a sink are instantly removed. Extending the discrete rule into continuous time results in a Dirichlet boundary condition,

$$m(a,t) = 0 \ \forall t, \tag{16}$$

for the PDE in (12).

Discrete- and continuous-time rules for a wall boundary are a bit more complicated but follow the same general process. A robot that chooses move 'into' a wall boundary actually stays in its current bin for that time step. The population of bin b at time $k+1$ is therefore represented in discrete time as

$$m_{b,k+1} = \frac{1}{2}m_{b,k} + \frac{1}{2}m_{c,k} \tag{17}$$

Comparing to (14), both conditions are satisifed by choosing

$$m_{a,k} = m_{b,k} \ \forall k. \tag{18}$$

Noting that bin b is adjacent to bin a and using the substitutions from (7), the continuous-time rule becomes

$$m(a,t) - m(a-\Delta s, t) = 0 \ \forall t. \tag{19}$$

Dividing both sides by Δs and taking the limit as $\Delta s \to 0$, a Neumann boundary condition,

$$\frac{\partial m(a,t)}{\partial s} = 0 \ \forall t, \tag{20}$$

is obtained for the PDE model.

2) Initial Conditions for 1D Random Walk Scenario: With the internal PDE and boundary conditions determined, only the initial conditions are needed to fully define the PDE model. Assume that M robots are initially distributed in the N internal bins so bin j has occupancy determined by

$$m_{j+1} = \frac{\pi * M}{2(N+1)}\sin(\frac{\pi * j}{N+1}), \ j = 0, 1, \ldots, N-1 \tag{21}$$

where the actual population is rounded to the nearest integer. This initial distribution approximating a half sine wave was chosen as a simple example to validate the PDE model but could correspond to a physical system where the majority of robots are inserted far away from potential obstacles (boundaries in this illustration). To map the occupancy of robots in bins $1 \to N$ to the continuous time domain, $D = (0, 1)$, the PDE should have initial condition

$$m(s, 0) = \sin(\pi s). \tag{22}$$

B. Extension of Methodology to 2D Random Walk

Though the 1D case illustrates our proposed methodology, 2D environments are a more realistic representation for our domain of interest. When moving from 1D to 2D, the same general steps are applied. Now N bins are evenly distributed in a rectangle, $\mathbb{R}^{N_x \times N_y}$ where $N_x = \{1, 2, \ldots, X\}$ and $N_y = \{1, 2, \ldots, Y\}$, in order to designate the desired spatial resolution. Two additional bins are required for each row, denoted as 0 and $X+1$, and column, 0 and $Y+1$, to define the interval boundaries. At every discrete-time step, each of the M robots now randomly decides to move either up, right, down, or left one bin with probabilities P_U, P_R, P_D, or P_L respectively. To be a proper probability model, an additional constraint where the sum of the probabilities for each of the four different moves is equal to one applies. The mean occupancy, m, of interior bin $[i, j]$ at time $k+1$ is once again the sum of robots in neighboring bins which choose to move into bin $[i, j]$, mathematically expressed as

$$\begin{aligned} m_{[i,j],k+1} \quad = \quad & P_U m_{[i,j-1],k} + P_R m_{[i-1,j],k} + \\ & P_D m_{[i,j+1],k} + P_L m_{[i+1,j],k}. \end{aligned} \tag{23}$$

For simplicity, we assume that the robots move to one of the four available bins with equal probability:

$$P_U = P_R = P_D = P_L = \frac{1}{4}. \tag{24}$$

Using a similar methodology as the 1D case, the mean occupancy of bin $[i, j]$ at time $k+1$ therefore reduces to

$$\begin{aligned} m_{[i,j],k+1} \quad = \quad & \tfrac{1}{4}(m_{[i,j-1],k} + m_{[i-1,j],k} + \\ & m_{[i,j+1],k} + m_{[i+1,j],k}). \end{aligned} \tag{25}$$

The change in the mean occupancy of bin $[i, j]$ between time k and $k+1$ can be expressed as

$$\begin{aligned} \Delta m_{[i,j]} \quad = \quad & \tfrac{1}{4}(m_{[i,j-1],k} + m_{[i-1,j],k} + m_{[i,j+1],k} + \\ & m_{[i+1,j],k}) - m_{[i,j],k}. \end{aligned} \tag{26}$$

Equation (26) represents the mean discrete-time change in bin population for interior bins in a two-dimensional rectangle. To map the discrete-time model to continuous time, the following substitutions are made much as in the 1D case:

$$\begin{array}{rcl} [i \pm 1, j \pm 1] & \rightarrow & [x \pm \Delta x, y \pm \Delta y] \\ k+1 & \rightarrow & t + \Delta t. \end{array} \tag{27}$$

The continuous-time model for the internal, two-dimensional random walk becomes

$$\begin{gathered} m([x,y], t+\Delta t) - m([x,y],t) = \\ \tfrac{1}{4} m([x, y-\Delta y], t) + \tfrac{1}{4} m([x-\Delta x, y], t) + \\ \tfrac{1}{4} m([x, y+\Delta y], t) + \tfrac{1}{4} m([x+\Delta x, y], t) - \\ m([x,y],t). \end{gathered} \tag{28}$$

By using a second-order Taylor Series expansion to approximate the continuous-time model of (28), performing algebraic simplification, defining $\Delta t = \Delta x^2$ and $\Delta t = \Delta y^2$ so both sides of the equation can be divided appropriately, and taking the limit as $\Delta t \rightarrow 0$, the continuous-time model converges to the standard multi-dimensional heat equation

$$\frac{\partial m(\vec{x}, t)}{\partial t} = \frac{1}{4} \nabla^2 m(\vec{x}, t) \tag{29}$$

where in this scenario

$$\vec{x} = [x, y] \in (0,1)^2. \tag{30}$$

1) Boundary Conditions in 2D: As for the 1D scenario, the key to boundary conditions is ensuring all internal bins follow the same rule as described by (25). The discrete-rule is satisfied for sink boundaries by again imposing a Dirichlet boundary condition. Similarly, the discrete-rule is satisfied for a wall boundary by imposing a modified Neumann boundary condition,

$$\vec{d} \cdot \nabla_x m(a, t) = 0 \ \forall t, \tag{31}$$

where $\vec{d}$ represents the direction of the boundary bin with respect to the internal bin of interest.

2) Initial Conditions in 2D: For the 2D random walk simulation, M robots are initially distributed in the N internal bins so bin $[i, j]$ has occupancy determined by

$$\begin{aligned} m_{i,j} &= M \tfrac{\pi}{2(Y+1)} \sin(\tfrac{\pi * i}{Y+1}) \tfrac{\pi}{2(X+1)} \sin(\tfrac{\pi * j}{X+1}), \\ & i = 0, 1, \ldots, Y-1, \ j = 0, 1, \ldots, X-1 \end{aligned} \tag{32}$$

with $m_{i,j}$ rounded to the nearest integer. Mapping the discrete bin occupancy to continuous time as in the 1D scenario reveals the PDE should have initial conditions:

$$m([i,j], 0) = \sin(\pi * i) \sin(\pi * j). \tag{33}$$

IV. Validation of Partial Differential Equation Models for Emergent Behavior

A. Overview

To evaluate the effectiveness of the proposed methodology for deriving a PDE model of emergent swarm behavior, scripts were written in MATLAB to simulate a swarm of robots exploring an environment using simple random walk motion. Robots are distributed in finite-sized bins evenly spaced throughout the environment. During every discrete-time step, each robot moves to a neighboring bin with appropriate probability. The MATLAB-based simulation records the population of all bins for each discrete-time step. Similarly, the corresponding PDE model is solved to produce a solution showing the time evolution of robot occupancy throughout the environment.

B. Scaling for 1D Random Walk Scenarios

We again start with the 1D model for validation before expanding into 2D. To simulate the 1D random walk, M robots are distributed across N bins according to the initial conditions of (21). The corresponding internal PDE model is given by (12). Appropriate scaling factors are required for time, space, and population to accurately compare the simulation and corresponding PDE.

For the 1D line, a total of $N + 2$ bins are used in the simulation to account for boundary bins. Each bin is represented by its mid-point. The left-most midpoint corresponds to the left boundary of the PDE, namely 0. Similarly, the midpoint of the right-most bin is associated with the right PDE boundary. Hence, to correlate the simulation which has a variable number of bins to the fixed $(0, 1)$ PDE interval, the simulation is scaled by

$$\Delta s = \frac{1}{N+1}. \tag{34}$$

Increasing N improves the spatial resolution because more bins are mapped to the fixed $(0, 1)$ interval. With the spatial scaling specified by the user, temporal scaling is determined by $\Delta t = \Delta s^2$ as previously described.

The final scaling is to ensure comparable bin occupancies. The simulation is normalized by dividing the occupancy of each bin by the total number of robots in a simulation - namely M - so from (21) has a maximum possible value of

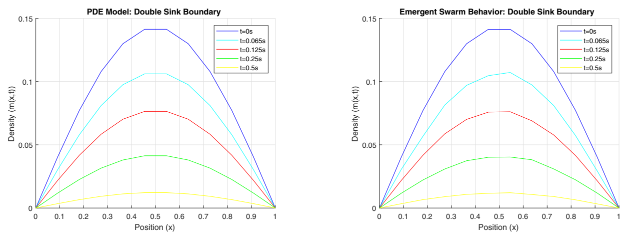
$$z_{\max} = \frac{\pi}{2(N+1)}. \tag{35}$$

The PDE is initialized according to (22) which has a maximum value of one so for comparison the PDE model must be multiplied by $z_{\max}$ as well. With these scaling factors, a simulation can be compared to its associated PDE model to produce nearly identical bin populations with improved accuracy for increased values of M and N as supported in the following section.

C. Results of 1D Random Walk

To evaluate the effectiveness of the derived PDE model, the occupancy of each bin was plotted per time step and compared to the corresponding PDE solution at equivalent time samples. Both techniques generate a surface plot, showing the bin occupancy at every position for each time, so slices of each surface were taken at equivalent times for better comparison. Using the scaling described in the previous section, the simulation and PDE generate very similar results. The time required to generate simulation results depends on the number of robots and the number of bins whereas the PDE solution is independent of both factors and was consistently solved in less than 0.8 seconds.

Running a simulation with sink boundaries at both ends to represent an obstacle-free environment and using $N = 10$ bins with $M \approx 10,000$ robots results in the expected behavior for both the PDE model and simulation as shown in Fig. 1. Even with this comparatively low resolution environment, the simulation required more than 21 times the computation time of the PDE.

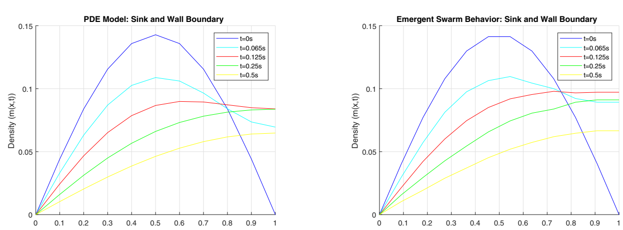


(a) PDE Model:
0.39 sec to generate

(b) Simulation Results:
8.45 sec to generate

Fig. 1: Comparison of spatial occupancies for a 1D environment at fixed time intervals for the (a) PDE model and (b) discrete, 10-bin simulation with two sink boundaries.

Introducing a wall boundary at bin $n = N + 1$ to represent an obstacle causes a build-up in the occupancy of the adjacent bin throughout the simulation though the sink boundary allows for a steady decrease in bin occupancy as shown in Fig. 2. A slight difference in occupancy at the wall boundary between the simulation and PDE is the result of the low number of bins and the scaling of (34). In the simulation, robots only occupy N bins but the PDE necessarily considers $N + 1$ bins. As $N \to \infty$, the difference becomes negligible and even with $N = 10$ is minor.

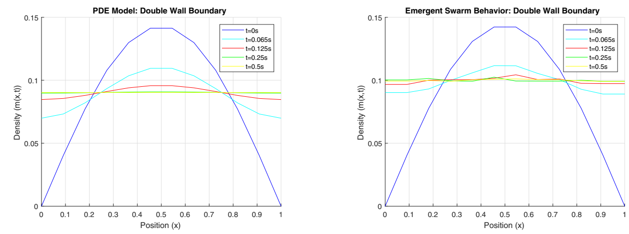


(a) PDE Model:
0.56 sec to generate

(b) Simulation Results:
19.77 sec to generate

Fig. 2: Comparison of spatial occupancies for a 1D environment at fixed times for the (a) PDE model and (b) discrete, 10-bin simulation for a left sink and right wall boundary.

When both boundaries become obstructed, denoted as wall boundaries, the difference between the PDE and simulation becomes more pronounced because the steady-state solution is non-zero. With two wall boundaries, robots cannot 'escape' the simulation so instead approach a uniform distribution. A uniform distribution for the simulation results in a mean bin occupancy of 0.1 because $N = 10$ bins are available. By contrast, the PDE necessarily settles to $1/(N+1)$ or approximately 0.09, an artifact of the continuum limit mapping. As shown in Fig. 3, the PDE model nonetheless presents a distinguishable distribution when compared to the other boundary scenarios.



(a) PDE Model:
0.64 sec to generate

(b) Simulation Results:
19.67 sec to generate

Fig. 3: Comparison of spatial occupancies for a 1D environment at fixed time intervals for the (a) PDE model and (b) discrete, 10-bin simulation with two wall boundaries.

The overall behavior between the actual simulation and the PDE model for the double wall scenario becomes increasingly similar as the number of bins is increased. Using $N = 50$ increases the spatial resolution of the simulation and decreases the impact of the $N + 1$ artifact; hence, a closer correlation between the emergent behavior of the swarm and the PDE model is observed. The error between the PDE model and emergent behavior, averaged over ten simulation runs, is presented in Fig. 4 for the scenario with $N = 10$ as the solid lines and $N = 50$ as the dashed lines at three different sample times. Each 10-bin simulation required less than 47 seconds to run while the 50-bin simulations each took over 30 minutes. The PDE in both scenarios required approximately 7 seconds to complete.

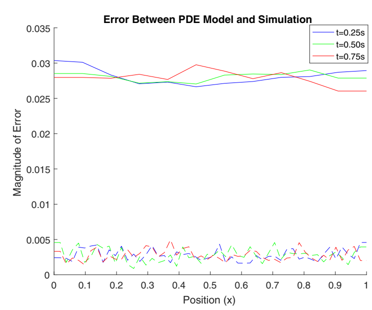


Fig. 4: Comparison of error between the PDE model and emergent behavior in simulated 10-bin (solid line) and 50-bin (dashed line) environments for double wall boundary conditions.

Although increasing the spatial resolution improves the model accuracy, a resolution of $N = 10$ bins was used for all three 1D scenarios. This chosen resolution balanced simulation time and model accuracy. Figure 5 shows the error between the robot densities for the simulation and PDE model in a 1D environment with double sink boundaries as the number of robots and the environment resolution are systematically increased. The model accuracy improves as both variables increase but the amount of improvement decreases after $N \approx 10$ bins. It is further worth noting the significant difference in generation time: increasing the resolution from $N = 10$ to $N = 30$ internal bins increased the run-time from 14 seconds to over 15 minutes. Due to the long simulation time required for higher environment resolution and the evidence from Fig. 5 showing diminishing improvement in accuracy, the subsequent work maintains smaller N to focus on the effectiveness of the PDE models.

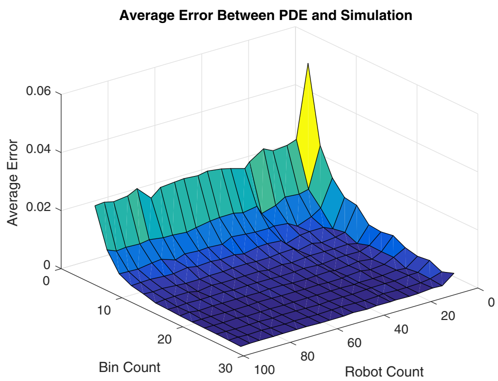


Fig. 5: The PDE model for a 1D, double sink environment becomes increasingly accurate as the number of bins and number of agents increase.

D. Evaluation of 2D Model

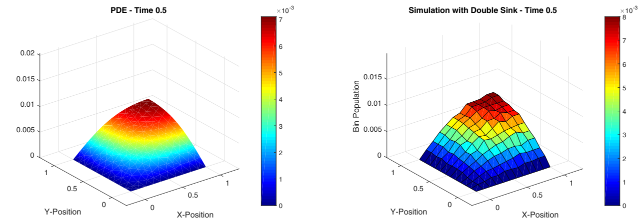
With the 1D models validated, we again apply a similar strategy to investigate the more representative 2D models. Similar scaling is applied to the 2D PDE model and simulation to compare both techniques graphically. With N bins now distributed in a 2D configuration, time samples of robot distribution are surface plots rather than lines and a greater combination of boundary conditions exist. Systematically varying the boundary conditions for a 10×10 bin environment with $M \approx 100,000$ confirmed that the PDE model again effectively captured the emergent behavior with solutions obtained in seconds rather than minutes or even hours.

As an illustrative example, Fig. 6 shows the bin occupancies for a double wall, double sink boundary scenario. Such a scenario could represent a room corner. The PDE model captures the overall shape of the emergent behavior with a slightly lower magnitude at the wall boundaries–a discrepancy once again caused by the low bin count.

V. Environmental Mapping from PDE Model

A. Simple Illustrative Use Case

Thus far, we have shown that PDE models derived from individual random walk behavior of robots using a continuum limit methodology reasonably capture the emergent behavior of a robotic swarm in varying environments. The PDE model is increasingly accurate at modeling the distribution of robots as the simulation resolution increases but the solution is



(a) PDE Model: 1.7 sec to generate

(b) Simulation Results: 171.5 sec to generate

Fig. 6: PDE solution and simulation at one time sample with two wall boundaries and two sink boundaries.

significantly faster through the PDE. In this section, we will show how the PDE models can be used to infer environmental boundary conditions very quickly, but first we propose a relevant physical example.

Assume you are in a mine tunnel that has recently experienced a partial collapse. Two exits, either going left or right, are in the tunnel. You no longer know the state of either exit as one or both may now be blocked. Rather than expend energy searching both directions, you deploy a swarm of robots. After some time you see how many robots are near you and from that observation are able to determine the state of both potential exits.

Many challenges need to be addressed before the proposed scenario is physically realizable, but this work lays a promising theoretical foundation. PDE models for each potential scenario can be quickly generated and the observed robot occupancy in the middle can be compared to each PDE to find the nearest model, as demonstrated next.

B. Boundary Identification for 1D Environment

Three different boundary condition pairs were considered for the 1D environment: double sink, double wall, and mixed. Using $M \approx 100,000$, $N = 10$, and an initial distribution placing the majority of robots in bins furthest away from the boundaries, only a few iterations (time steps) were required before the simulations revealed a distinguishable distribution of robots throughout the environment. The associated PDE models also captured the unique distribution but much more quickly.

By observing the number of robots in a central bin (furthest from the boundaries) and comparing the occupancy to each of the three PDE models, it is possible to use a least-squares error to determine which model is most similar to the observed behavior and therefore determine the boundary conditions. To demonstrate the effectiveness of this simple approach, a simulation for each boundary condition pair was executed in MATLAB with the middle bin occupancy compared to the three potential PDE models. As shown in Fig. 7, the simulation consistently and reliably converged to the appropriate model within 15 time steps.

Observing the middle bin is the worst-case scenario because it is furthest from the discriminating features and would be ineffective in determining whether the wall was to

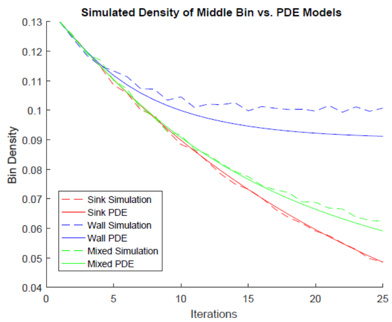


Fig. 7: Convergence of central bin occupancy for 10-bin simulated environment to corresponding PDE model reliably occurs in 15 discrete-time steps.

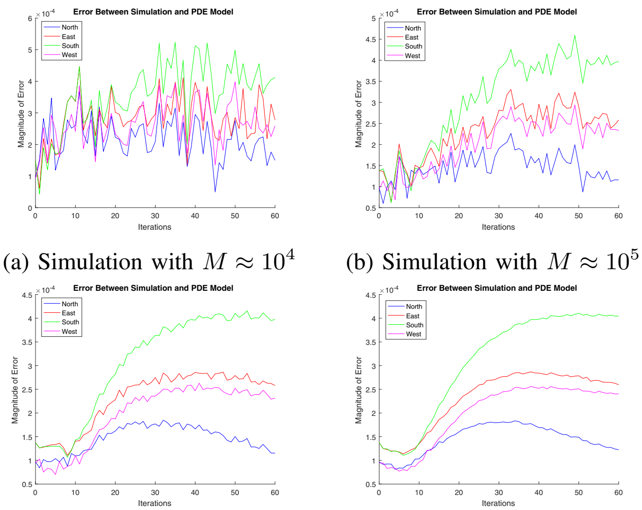
the ‘left’ or ‘right’ in the case of the sink and wall boundary scenario. Nonetheless, Fig. 7 demonstrates the potential of using the proposed continuum limit methodology to develop environmental models for quick identification. The presence or absence of an obstacle was determined reliably after simulated robots within the swarm moved 15 times in a worse-case scenario where communication was absent and the robots were reduced to random movements.

C. Doorway Detection in 2D Environment

Looking toward physical implementations, a 2D environment is more common but the same general methodology can be applied. Consider an office room with no light and a single doorway. It may be impossible to identify where the doorway is after a partial building collapse and extremely dangerous for survivors in the room to search for the door. A safer option may be to use a swarm of robots to safely locate the exit.

To demonstrate the feasibility of this illustrative scenario, a simulation and corresponding PDE models were developed using the presented methodology. The simulation consisted of a 10×10 bin interior environment. Additional boundary bins were placed around the environment, all corresponding to walls except for four consecutive, central bins in the ‘north’ wall that were modeled by sinks to represent the available doorway. Four separate PDE models were developed–all identical for the internal model but with varying boundary conditions to represent a central doorway in either the north, east, south, or west wall.

The bin occupancy for four central bins in the simulation was compared to corresponding points in each PDE model to create a 4×1 difference vector. The norm of the difference vector, hereby referred to as the error, for each PDE model was averaged over ten simulations. The averaged error was then plotted at each time step to show the proximity of the simulation to each of the four PDE models over time. Results are shown in Fig. 8. When $M \approx 10^4$, a statistical discernment exists between the error for each PDE model. Increasing the number of robots by an order of magnitude clarifies the separation of the error between the models with the north model having the lowest error which is encouraging as it matches the simulation. Further increasing the number of robots refined the separation between models.



(a) Simulation with $M \approx 10^4$ (b) Simulation with $M \approx 10^5$

(c) Simulation with $M \approx 10^6$ (d) Simulation with $M \approx 10^7$

Fig. 8: Error between the central observation area for a 10×10 simulated environment with a four-bin doorway in the north wall and each of the four PDE models for doorways in the north, east, south, and west walls.

It is important to not misconstrue the data from Fig. 8. Identifying which of the four walls contains the single doorway does not require $M \approx 10^5$ robots nor are error plots the desired method for environment detection. We show these plots to illustrate the convergence properties of the PDE model and to support the theoretical basis of the proposed methodology. With $M \approx 10^4$ or fewer robots, see Fig. 8(a), there is already a significant difference in the statistical properties associated with the four possible environments. In the following section, we will show how this statistical difference can be physically visualized.

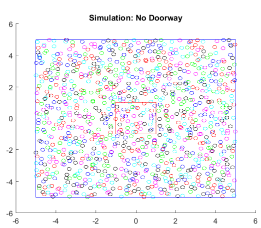
Nonetheless, Fig. 8 supports the use of PDE models in place of simulations for determining emergent swarm behavior and using the resulting robot density to identify environmental features such as unobstructed doorways. Further work is needed to define the correlation between robot count, environment size, features, and resolution. More sophisticated robot behaviors will also impact these four key variables and will be our next focus. However, here we have provided a theoretical foundation for a worst-case scenario in terms of robot exploration. Robots unable to communicate and that are reduced to random motion can still be used to identify in which direction a doorway lies by observing the density of robots in another area.

D. Illustration of Methodology for Particle Collisions

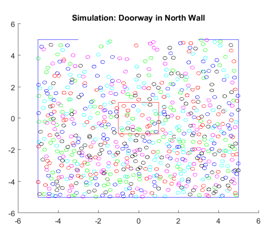
Our ultimate goal is to have a physical realization of a swarm wherein the observed robot density in one area can be used to infer the presence of obstacles in an unknown environment. This work provides the beginning theoretical foundation for achieving our goal but as a further illustration of how robot density in one region can be used to determine the presence of obstacles in another, even in a worse-case scenario, consider the collision of particles in a gas.

In a simplified model, each particle travels a straight trajectory until a collision occurs, at which point the particle is deflected in a way to conserve momentum [12]. As the number of particles in a bounded area increases, individual particle motions also become increasingly random and approach the behavior we have assumed for robots in this preliminary work. By extension, there should be an observable difference in the number of particles present at the center of unique environments–a difference that can distinguish between the possible environments.

We illustrate the particle analogy in MATLAB. Initially, the square environment is fully bounded so no particles can escape and the particles approach a uniform distribution as expected. When a doorway is added to the north wall, paralleling the doorway detection simulations in Section V-C, particles are able to escape and hence fewer are present in the environment. Snapshots of both environments are shown in Fig. 9 at equivalent times. The center red square aids in comparing the density of particles in the middle of the environment. Fewer particles are clearly present in the observation area for Fig. 9(b) and hence one can conclude that environment contains the doorway.



(a) Distribution of particles with no doorway



(b) Distribution of particles with a single doorway

Fig. 9: A comparison of particle density in two distinct environments confirms the density in one area is distinguishable for different environments.

Robots are capable of much more sophisticated behaviour than gas particles but this work is a first step toward modelling emergent behavior with PDEs and developing models to identify environmental features from observed swarm behavior. Gas particles approach the random motion assumed for the initial robot models in this work and provide a stepping stone toward physical demonstration of our proposed methodology.

VI. CONCLUSIONS

This foundational work proposes the use of continuum limits to model the emergent behavior of a swarm of robots with stochastic behaviors. By extending discrete local rules into a continuous-time domain, a PDE model is obtained. The base PDE model describes the interior behavior of robots in the environment while the PDE boundary conditions encode environmental features—namely open space or obstacle.

Solutions to PDEs can be quickly computed and are independent of the number of robots and largely independent of environment size. By contrast, large-scale stochastic networks like robotic swarms require many hours or even days to run with the time depending on both the number of robots and the network or environment size. As a result, PDEs can serve to provide quickly generated environmental models.

Different environmental features cause unique, distinguishable distributions of robots in the environment so observations at one location can be correlated to environmental features in another location. As shown in this preliminary work, observing the number of robots in a central location can be used to identify the presence of sink or wall boundaries by comparing to already determined PDE models. The robots considered in this work represent a worse-case scenario where no communication or advanced decision making is implemented and motion is random. Nonetheless, valuable information can be obtained as partially illustrated in this work.

Future work will focus on extending the complexity of the robot behaviors modeled, introducing internal obstacles, and implementing the results on physical robots in real-world environments. Additional investigations into the relationship between number of robots, environment size, and exploration time will also be investigated to provide confidence bounds on the results obtained.

REFERENCES

[1] I. Navarro and F. Matía, "An introduction to swarm robotics," *ISRN Robotics*, vol. 2013, 2012.

[2] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[3] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[4] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *International Workshop on Swarm Robotics*. Springer, 2004, pp. 10–20.

[5] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, May 2012.

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

[7] S. Berman, A. Halasz, 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, Aug 2009.

[8] L. E. Barnes, M. A. Fields, and K. P. Valavanis, "Swarm formation control utilizing elliptical surfaces and limiting functions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 6, pp. 1434–1445, Dec 2009.

[9] K. Elamvazhuthi and S. Berman, "Optimal control of stochastic coverage strategies for robotic swarms," in *2015 IEEE International Conference on Robotics and Automation*, May 2015, pp. 1822–1829.

[10] M. Berger, L. M. Seversky, and D. S. Brown, "Classifying swarm behavior via compressive subspace learning," in *2016 IEEE International Conference on Robotics and Automation*, May 2016, pp. 5328–5335.

[11] Y. Zhang, E. K. P. Chong, J. Hannig, and D. Estep, "Approximating extremely large networks via continuum limits," *IEEE Access*, vol. 1, pp. 577–595, 2013.

[12] R. M. Logan and R. Stickney, "Simple classical model for the scattering of gas atoms from a solid surface," *The Journal of Chemical Physics*, vol. 44, no. 1, pp. 195–201, 1966.