Self-Organization in Multi-Agent Swarms via Distributed Computation of Diffeomorphisms

Vishaal Krishnan and Sonia Martínez

Abstract—**In this work, we address the problem of self-organization for multi-agent swarms in 1D and 2D spatial domains. The objective is to achieve a desired density distribution over a continuous spatial domain. Since individual agents in a swarm are not themselves of interest and we are concerned only with the macroscopic objective, we view the swarm as a discrete approximation of a continuous medium and design spatial control laws to shape the density distribution of the continuous medium. The key feature of this work is that the agents in the swarm do not have access to position information nor do they have the capability to measure the distances to their neighbors. Each individual agent is capable of measuring the current local density of agents and can communicate with its neighbors. The agents implement a distributed algorithm, which we call pseudo-localization, to localize themselves in a new coordinate frame, and a distributed control law to converge to the desired spatial density distribution. We start by studying self-organization in one-dimension, which is then followed by the two-dimensional case.**

I. Introduction

The emergence of long-range order or coordination from local interactions in large collectives of dynamic agents is the class of phenomena broadly referred to as self-organization in swarms. Self-organization is a pervasive phenomenon in nature, observed in biological [1] and other natural systems [2]. The development of low-cost robotic sensor, communication, and computation systems has led to the development of large scale robotic counterparts [3], with applications to monitoring, manipulation, and construction. This transition does not merely involve an increase in the size of these networks, but it also introduces new theoretical challenges for analysis and control design. In particular, large groups of agents have some essential characteristics that distinguish them from other smaller-scale multi-agent systems. In a swarm, agents have no individual significance and only the macroscopic objectives are of importance. A swarm largely remains unaffected by the removal of a large, but discrete, number of agents. It is also nearly infeasible to localize all individual agents in a swarm via a global positioning system. Moreover, it is difficult (and needlessly complicated) to specify the global configuration of the swarm using the states of individual agents; instead, it is more appropriate to employ macroscopic quantities such as the swarm spatial density distribution to specify its configuration. From an analysis and control-theoretic viewpoint, the dynamic modeling of swarms can be established by means of PDEs, for which control theoretic tools are less well developed in comparison to ODEs. These theoretical challenges motivate the investigation of self-organization in swarms.

The authors are with the Department of Mechanical and Aerospace Engineering, University of California at San Diego, La Jolla CA 92093 USA (email: v6krishn@ucsd.edu; soniamd@ucsd.edu). This work has been partially supported by the grant AFOSR - 11RSL548

In the literature, Markov-chain based methods have arguably been the most effective at addressing some of the key theoretical problems pertaining to swarm self-organization. By means of it, the spatial domain is partitioned into a finite number of disjoint sub-regions or “bins,” on which a probability distribution is defined. Then, the self-organization problem is reduced to the design of the transition matrix governing the evolution of this probability density function to ensure its convergence to a desired profile. A recent approach to density control using Markov chains is presented in [4], which includes additional conflict-avoidance constraints. In this setting every agent is able to determine the bin to which it belongs at every instant of time, which essentially means that individual agents have self-localization capabilities. Also, the dimensional transition matrix is synthesized in a central way at every instant of time by solving a convex optimization problem. In [5], the authors make use of inhomogeneous Markov chains to minimize the number of transitions to achieve a swarm formation. In this approach, the algorithm necessitates the estimation of the current swarm distribution, and computes the transition Markov matrices for each agent, at each instant of time. The fact that every agent needs to have an estimate of the global state (swarm distribution) at every time may not be desirable or feasible. The localization of each agent still remains to be a main assumption. Under similar conditions, one can find the manuscript [6], which describes probabilistic swarm guidance algorithms.

In this work, we adopt the viewpoint outlined in [7], wherein we make an amorphous medium abstraction of the swarm, an amorphous medium being a manifold with an agent (mobile computational device) at each point. We then model the system using PDEs and design distributed control laws for them. Previous literature on PDE-based methods includes [8], where the authors present algorithms for the deployment of agents onto families of planar curves. Here, the swarm collective dynamics are modeled by the reaction-advection-diffusion PDE and the particular family of curves to which the swarm is controlled to is parametrized by the continuous agent identity in the interval of unit length. An extension of this work to deployment on a family of 2D manifolds in 3D space can be found in [9]. In [10], the authors present a distributed optimal control problem formulation for swarm systems, where microscopic control laws are derived from the optimal macroscopic description using a potential function approach.

In the context of robotic swarms, programmable self-assembly of two-dimensional shapes with a thousand-robot swarm is demonstrated in [11]. These robots are capable of measuring distances to nearby neighbors which they use to localize themselves relative to other localized robots. Each robot then uses its position to implement an edge-following algorithm.

The main contribution of this paper is the development of distributed control laws for the index- and position-free density control of swarms to achieve general 1D and a large class of 2D density profiles. In very large swarms with thousands of agents, particularly those deployed indoors or at smaller scales, presupposing the availability of position information or pre-assignment of indices to individual agents would be a strong assumption. In this paper, in addition to not making the above assumptions, we suppose that the agents are not capable of measuring distances to their neighbors. The agents are only capable of measuring the local density, and in the 2D case, the boundary agents are capable of estimating the normal direction to the boundary.

Under these assumptions, we present distributed pseudo-localization algorithms for 1 and 2 dimensions that agents implement to compute their position identifiers. Since every agent occupies a unique spatial position, we are able to rigorously characterize the resulting position assignment as a one-to-one correspondence between the set of spatial coordinates and the set of position identifiers, which corresponds to a diffeomorphism of the continuum domain. Based on this assignment, control strategies for self-organization in one and two dimensions are provided, under the assumption that motion control of agents is noiseless. The extension to the 2D case leads to new difficulties related to the control of the swarm boundaries. To address these, a variant of the 1D pseudo-localization algorithm is implemented at the boundary during an initialization phase.

II. PRELIMINARIES

Let $\mathbb{R}$ denote the set of all real numbers, $\mathbb{R}_{\geq 0}$ the set of non-negative real numbers, and $\mathbb{R}^n$ the n-dimensional Euclidean space. We use boldface letters to denote vectors in $\mathbb{R}^n$. The norm $|\mathbf{x}|$ of a vector $\mathbf{x} \in \mathbb{R}^n$ is the standard Euclidean 2-norm, unless otherwise specified. Let $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$ denote the gradient operator in $\mathbb{R}^2$. As a short-hand, we let $\frac{\partial}{\partial z}(.) = \partial_z(.)$ for a variable z. We denote by $\left(\frac{\partial f}{\partial x}\right)_g$ the partial derivative of f with respect to x holding the function g constant. We denote by either $\dot{S}$ or $\frac{dS}{dt}$ the time derivative of $S(t)$. Given functions $f, g : \mathbb{R} \to \mathbb{R}$, we write $f = \mathscr{O}(g)$ if there exist positive constants C and c such that $|f(h)| \leq C|g(h)|$, for all $|h| \leq c$. Let $\mathscr{S}$ denote the set of agents in the swarm, and N its cardinal. For the 1D case, let $l \in \mathscr{S}$ denote the leftmost agent, and $r \in \mathscr{S}$ the rightmost one. Let $\mathscr{N}_i$ denote the spatial neighborhood of agent i, which comprises those agents located inside a small ball centered at i. For a spatial domain $M \subset \mathbb{R}^n$, ∂M denotes the boundary of M.

We now state some well-known results that we will be used in the subsequent sections of this paper.

Lemma 1 (Divergence Theorem [12])**.** *For a vector field* $\mathbf{F}$ *over a region* $M \subseteq \mathbb{R}^n$ *with boundary* ∂M*, the volume integral of the divergence* $\nabla \cdot \mathbf{F}$ *of* $\mathbf{F}$ *over* M *is equal to the surface integral of* $\mathbf{F}$ *over* ∂M*:*

$$\int_M (\nabla \cdot \mathbf{F}) d\mu = \int_{\partial M} \mathbf{F} \cdot \mathbf{ds}. \tag{1}$$

•

For a scalar field U and a vector field $\mathbf{F}$ defined over a region $M \subseteq \mathbb{R}^n$:

$$\int_M \mathbf{F} \cdot \nabla U = \int_{\partial M} U \mathbf{F} \cdot \mathbf{ds} - \int_M U \nabla \cdot \mathbf{F}.$$

Lemma 2 (Leibniz Integral Rule)**.** *Let* $f : M^t (\subset \mathbb{R}^n) \times \mathbb{R} \to \mathbb{R}$*. Then:*

$$\frac{d}{dt}\left(\int_{M^t} f(\mathbf{r},t) d\mu_{\mathbf{r}}\right) = \int_{M^t} \frac{\partial}{\partial t} f(\mathbf{r},t) d\mu_{\mathbf{r}} + \int_{\partial M^t} f(\mathbf{r},t) \mathbf{v} \cdot \mathbf{n},$$

where $\mathbf{v}$ *is the velocity of the boundary and* $\mathbf{n}$ *is the normal to the boundary.* •

Lemma 3 ([13])**.** *Let* $M : \mathbb{R} \to \mathbb{R}^2$ *be a (smoothly) time-varying compact 2-manifold and let* $f : \mathbb{R} \times M \to \mathbb{R}^n$ *be a time-varying (scalar or vector-valued) function on* M*. Let* U *be the energy functional defined as* $U = \frac{1}{2}\int_M |f|^2$*. Then,* $\partial_t U = \int_M f \cdot \left(\frac{df}{dt}\right) + \frac{1}{2}\int_M |f|^2 \nabla \cdot \mathbf{v}$*, where* $\frac{d}{dt} = \partial_t + \mathbf{v} \cdot \nabla$*.* •

Lemma 4 (Poincaré-Wirtinger Inequality [14])**.** *For* $p \in [1, \infty]$ *and* Ω *a bounded connected open subset of* $\mathbb{R}^n$ *with a Lipschitz boundary, there exists a constant* C *depending only on* Ω *and* p *such that for every function* u *in the Sobolev space* $W^{1,p}(\Omega)$*:*

$$\|u - u_\Omega\|_{L^p(\Omega)} \leq C \|\nabla u\|_{L^p(\Omega)},$$

where $u_\Omega = \frac{1}{|\Omega|}\int_\Omega u d\mu$*, and* $|\Omega|$ *is the Lebesgue measure of* Ω*.* •

A. Continuum approximation

Given that N, the number of agents in the swarm, is very large, we will analyze and design the swarm dynamics through a continuum approximation. Let $t \in \mathbb{R}_{\geq 0}$, and let $M(t) \subset \mathbb{R}^n$ describe the spatial region in which the continuum of agents is deployed at time t. We denote $\dot{\mathbf{r}}_i(t) = \mathbf{v}_i$, $\forall i \in \mathscr{S}$, where $\mathbf{r}_i(t) \in M(t)$ is the position of the ith agent in the swarm at time t. Let $\rho : \mathbb{R} \times M \to \mathbb{R}_{>0}$ be the spatial density function, such that $\forall \mathbf{r} \in M(t)$, $\int_{M(t)} \rho(t, \mathbf{r}) d\mu_{\mathbf{r}} = 1$, where $d\mu_{\mathbf{r}}$ is the volume measure in $M(t)$. We assume that $M(t)$ is simply connected and that the boundary $\partial M(t)$ does not self-intersect. By assuming that ρ is smooth, the macroscopic agent dynamics can now be described by the continuity equation [12], assuming that the total number of agents is conserved:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \qquad \forall \mathbf{r} \in M^t, \tag{2}$$

where $\mathbf{v} : \mathbb{R}_{\geq 0} \times M \to \mathbb{R}^n$, and $\mathbf{v}_i(t) = \mathbf{v}(t, \mathbf{r}_i)$.

B. Harmonic maps

A map $\phi : (M, g) \subset \mathbb{R}^2 \to (N, h) \subset \mathbb{R}^2$ (where g and h are Riemannian metrics) is called harmonic if it minimizes the functional:

$$E(\phi) = \int_M |d\phi|^2 dv_g. \tag{3}$$

The Euler-Lagrange equation for the functional E, for Euclidean metrics g and h, which also yields the minimum is given by $\Delta \phi = 0$, the Laplace equation. We now state a lemma from [15] on harmonic diffeomorphisms between Riemann surfaces.

Lemma 5 (Harmonic diffeomorphism [15])**.** *Let (M, g) be a compact surface with boundary and (N, h) a compact surface with non-positive curvature. Suppose that $\psi : M \to N$ is a diffeomorphism onto $\psi(M)$. Assume that $\psi(M)$ is convex. Then there is a unique harmonic map $\phi : M \to N$ with $\phi = \psi$ on ∂M, such that $\phi : M \to \phi(M)$ is a diffeomorphism.* •

We note that the non-positive curvature constraint in the lemma is essentially a constraint on the metric h on N, and the curvature is zero for the Euclidean metric.

III. Self-organization in one dimension

This section presents the pseudo-localization algorithm and derives a distributed control law for the 1D self-organization problem that we aim to solve.

The problem at hand is to ultimately design a distributed control law for the swarm to converge to a desired configuration, which is specified using a density distribution. At $t \in \mathbb{R}_{\geq 0}$, let $M(t) = [0, L(t)] \subset \mathbb{R}$ be the interval in which the agents are distributed, and let $\rho : \mathbb{R} \times M \to \mathbb{R}_{>0}$ be the normalized density function describing the swarm on that interval. Without loss of generality, we place the origin at the leftmost agent of the swarm. We also assume that the leftmost and the rightmost agents, l and r, are aware that they are at the boundary. Let $\rho^* : M^* = [0, L^*] \to \mathbb{R}_{>0}$ be the desired normalized density distribution, $\Theta^*(x) = \int_0^x \rho^*(\bar{x}) d\bar{x}$, $\Theta^*(L^*) = 1$. We would like to achieve $\rho \to \rho^*$ by assuming that agents know the functional form of ρ^*. To do this, an agent is able to measure the local density $\rho(t, x)$ at time t; however, its position x within the swarm is unknown, and, thus, the value of the desired local density $\rho^*(x)$ can not be directly computed. This is precisely the context in which we introduce the pseudo-localization algorithm, which is essentially a distributed algorithm that the agents implement to find a new set of coordinates on the spatial domain $M(t)$ that contains the agents and captures information on the current density $\rho(t, x)$. The distributed control law is then a function of these new coordinates and the measured local density function. Now, let $p^* : \mathbb{R} \to \mathbb{R}_{>0}$, and $\theta^* \in \Theta^*(M^*) = [0, 1]$, such that $p^*(\theta^*) = \rho^*(\Theta^{*-1}(\theta^*)) = \rho^*(x)$. We use the function p^*, which represents the desired density distribution in the transformed coordinate system, to derive the distributed control law.

A. Pseudo-localization algorithm for the swarm in 1D

Here, we present the pseudo-localization algorithm for the distributed computation of diffeomorphism for the swarm. This algorithm involves simple computation steps and local communication with the immediate left and right neighbors' of each agent. We first describe the idea of the coordinate transformation (diffeomorphism) we employ and construct a PDE that converges asymptotically to this diffeomorphism. We then discretize the PDE to obtain the distributed pseudo-localization algorithm. This will allow us to analyze and design a distributed control law in terms of the macroscopic density function ρ in Subsection III-B.

We use the cumulative distribution function to construct a coordinate transformation from the spatial domain to the unit interval $[0, 1]$. We define $\Theta : M = [0, L] \to [0, 1]$ as

$$\Theta(x) = \int_0^x \rho(\bar{x}) d\bar{x}. \tag{4}$$

such that $\Theta(L) = 1$.

Lemma 6. *Given $\rho : M \to \mathbb{R}_{>0}$, the mapping $\Theta : M \to [0, 1]$ is a diffeomorphism and $\Theta(M) = [0, 1]$.* •

The proofs for lemmas have not been presented here owing to constraints on space. For detailed proofs for the lemmas in this paper, the reader is referred to [13].

Our goal now is to set up a partial differential equation with appropriate boundary conditions such that it yields the diffeomorphism Θ as its asymptotically stable steady state solution. Let $X : \mathbb{R} \times M \to \mathbb{R}$, such that $(t, x) \mapsto X(t, x) \in \mathbb{R}$, and v is the velocity field, such that:

$$\begin{aligned} &\partial_t X = \frac{1}{\rho} \partial_x \left(\frac{\partial_x X}{\rho} \right) - v \partial_x X, \\ &X(t, 0) = 0, \\ &X(t, L(t)) = \beta(t), \\ &X(0, x) = 0. \end{aligned} \tag{5}$$

From (4), we observe that $\partial_x \left(\frac{\partial_x \Theta}{\rho} \right) = 0$. Letting $w = X - \Theta$, the error, and from (5), we get:

$$\begin{aligned} &\partial_t w = \frac{1}{\rho} \partial_x \left(\frac{\partial_x w}{\rho} \right) - v \partial_x w, \\ &w(t, 0) = 0, \\ &w(t, L) = \beta(t) - 1, \\ &\partial_t w(t, L) = -w(t, L), \\ &w(0, x) = -\Theta. \end{aligned} \tag{6}$$

Lemma 7 ([13])**.** *The system (5), with $v = 0$ (stationary condition) converges asymptotically to the diffeomorphism $\Theta = \int_0^x \rho(\bar{x}) d\bar{x}$.* •

We now discretize (5) to obtain the distributed pseudo-localization algorithm. Let $X_i = X(x_i)$, where x_i is the position of the i-th agent. We identify the agent i with its coordinate $\Theta(x) = \theta \in [0,1]$. From (4), we get $\rho(x) = \partial_x \Theta(x)$. It follows that $\partial_x(\cdot) = \partial_\theta(\cdot)\partial_x\theta = \partial_\theta(\cdot)\rho$. Therefore, $\frac{1}{\rho}\partial_x(\cdot) = \partial_\theta(\cdot)$. From (5), we have:

$$\frac{dX}{dt} = \partial_t X + v\partial_x X = \frac{1}{\rho}\partial_x\left(\frac{\partial_x X}{\rho}\right) = \partial_\theta\left(\partial_\theta X\right) = \frac{\partial^2 X}{\partial\theta^2}. \tag{7}$$

Discretizing (7), with $\frac{dX}{dt} = \frac{X_i(t+1)-X_i(t)}{\Delta t}$ and $\frac{\partial^2 X}{\partial\theta^2} = \frac{X_{i+1}-2X_i+X_{i-1}}{(\Delta\theta)^2}$, and the choice $3\Delta t = (\Delta\theta)^2$, and from (5), we get for $i \in \mathscr{S}\setminus\{l,r\}$:

$$\begin{aligned}
&X_i(t+1) = X_i(t) + \frac{1}{3}\sum_{j=i-1}^{i+1}\left[X_j(t) - X_i(t)\right],\\
&X_l(t) = 0,\\
&X_r(t) = \beta(t),\\
&X_i(0) = 0.
\end{aligned} \tag{8}$$

which is the distributed pseudo-localization algorithm.

B. Distributed Control and Analysis

In this subsection, we propose a distributed feedback control law to achieve $\rho \to \rho^*$ and $w \to 0$ through a distributed control input v and a boundary control $\beta(t)$. We refer the reader to [16] for an overview of Lyapunov-based methods for the stability of distributed systems.

From (2) we have:

$$\partial_t\rho = -\partial_x(\rho v), \tag{9}$$

which along with (5) constitutes the dynamics of the swarm. We assume that the agent at position x at time t is able to measure $\rho(t,x)$. However, the agents in the swarm do not have access to their positions, and therefore cannot access $\rho^*(x)$, which could be used to construct such feedback. To circumvent this problem, we propose a scheme in which the agents use the position identifier X involved in the pseudo-localization algorithm to access $p^*\circ X(t,x)$, using this as their dynamic set-point. The idea is to then design a distributed control law and a boundary control law such that $\rho \to p^*\circ X$ and $X \to \Theta^*$, to obtain $\rho \to p^*\circ\Theta^* = \rho^*$.

Consider the distributed control law, defined as follows for all time t:

$$\begin{aligned}
&v(t,0) = 0,\\
&\partial_x v = (\rho - p^*\circ X) - \frac{\partial_X p^*}{\rho(\rho + p^*\circ X)}\partial_x\left(\frac{\partial_x X}{\rho}\right).
\end{aligned} \tag{10}$$

And the boundary control law:

$$\begin{aligned}
&X(t,0) = 0,\\
&\beta_t = 2 - \beta(t) - \left.\frac{X_x}{\rho}\right|_{L(t)}.
\end{aligned} \tag{11}$$

It is worth noting here that the agents implementing the control laws (10) and (11) do not require position information, because for the agent at position x, $\rho(t,x)$ is a measurement, $X(t,x)$ is the pseudo-localization variable (through which $p^*\circ X(t,x)$ can be accessed). Discrete versions of control laws will be derived in Section III-C.

Theorem 1. *The system,* (5) *and* (9)*, with the control laws* (10) *and* (11) *is asymptotically stable at* $\rho = \rho^*$*,* $X = \theta^*$ *a.e.*

Proof. Consider the candidate control Lyapunov functional V:

$$V = \frac{1}{2}\int_0^{L(t)}|\rho - p^*\circ X|^2 dx + \frac{1}{2}\int_0^{L(t)}\frac{|w_x|^2}{\rho(x)}dx + \frac{1}{2}|w|^2(L(t)).$$

Taking the time derivative of V along the dynamics (9), along with the control law (10) and (11), we get (steps omitted owing to space constraints. The reader is referred to [13] for an extended proof):

$$\begin{aligned}
\dot{V} = &-\frac{1}{2}\int_0^{L(t)}(\rho + p^*\circ X)|\rho - p^*\circ X|^2 dx - \left|\frac{\partial_x w}{\rho} + w\right|^2_{L(t)}\\
&-\int_0^{L(t)}\frac{1}{\rho}\left|\partial_x\left(\frac{\partial_x w}{\rho}\right)\right|^2 dx.
\end{aligned}$$

Clearly, $\dot{V} \le 0$. Therefore, the system converges to $(\dot{V})^{-1}\{0\}$. The following hold a.e.:

$$\begin{aligned}
&\rho = p^*\circ X,\\
&\partial_x\left(\frac{\partial_x w}{\rho}\right) = 0 \Rightarrow \frac{\partial_x w}{\rho} = a \Rightarrow w(x) = a\int_0^x \rho,\\
&w(L) = a\int_0^L \rho = a,\\
&\left(\left.\frac{\partial_x w}{\rho}\right|_L + w(L)\right) = \left.\frac{w(L)\rho}{\rho}\right|_L + w(L) = 0 \Rightarrow w(L) = 0.
\end{aligned} \tag{12}$$

We have $\rho = p^*\circ X$ a.e. Our aim is to prove that $\rho = \rho^*$, which is the case when $X = \Theta^*$. Therefore, it remains to be shown that $\rho = p^*\circ X \Rightarrow X = \Theta^*$. Now, we have $X(x) = \Theta(x) = \int_0^x \rho(\tilde{x})d\tilde{x}$ (since $w = X - \Theta = 0$ from (12)). Thus, from Lemma 6, X is a diffeomorphism. Now, we define $p : [0,1] \to \mathbb{R}_{>0}$ such that $p\circ X(x) = \rho(x)$, and because $X(x) = \Theta(x)$, we have $X_x(x) = \rho(x) = p\circ X(x)$, and with a slight abuse of notation identifying $X(x)$ with X, we get $\frac{dX}{dx} = p(X) > 0\ \forall X \in [0,1]$. We therefore have:

$$x = \int_0^{X(x)}\left(p(\tilde{X})\right)^{-1}d\tilde{X}.$$

Recall from the definition of p^* and (4) that $p^*\circ\Theta^*(x) = \rho^*(x)$, and $\Theta^*_x(x) = \rho^*(x) = p^*\circ\Theta^*(x) \Rightarrow \frac{d\Theta^*}{dx} = p^*(\theta^*) > 0$, where $\theta^* = \Theta^*(x)$. Therefore:

$$x = \int_0^{\Theta^*(x)}\left(p^*(\tilde{X})\right)^{-1}d\tilde{X}.$$

From the above two equations, $\int_0^{X(x)}\left(p(\tilde{X})\right)^{-1}d\tilde{X} = \int_0^{\Theta^*(x)}\left(p^*(\tilde{X})\right)^{-1}d\tilde{X}$, and since $X = \Theta$, $p^*\circ X(x) = \rho(x) = p\circ X(x)$. Therefore:

$$\int_0^{X(x)}\left(p^*(\tilde{X})\right)^{-1}d\tilde{X} = \int_0^{\Theta^*(x)}\left(p^*(\tilde{X})\right)^{-1}d\tilde{X},$$

and p^* is strictly positive $\Rightarrow X(x) = \Theta^*(x)\ \forall x$, and we obtain $\rho(x) = p^* \circ X(x) = p^* \circ \Theta^*(x) = \rho^*(x)$. Therefore, the system (9) with control laws (10) and (11) is asymptotically stable at $\rho = \rho^*$, $X = \Theta^*$ a.e. $\square$

C. Discrete Implementation

In this subsection, we discretize (10) and (11) to obtain implementable laws for a finite number of agents $i \in \mathscr{S}$, and a numerical simulation is later presented in Section V.

Let $i \in \mathscr{S} \setminus \{l, r\}$. First note that $\partial_x v = (\partial_\theta v)(\partial_x \Theta) = (\partial_\theta v)\rho$. Using backward differencing and recalling that $\Delta\theta = \varepsilon$, we can write:

$$\partial_x v = \rho_i \frac{v_i - v_{i-1}}{\Delta\theta} = \rho_i \frac{v_i - v_{i-1}}{\varepsilon},$$

where ρ_i is agent i's density measurement.

Recall that, from Section III-A, $\frac{1}{\rho}\frac{\partial}{\partial x}\left(\frac{X_x}{\rho}\right) = \frac{1}{\varepsilon^2}\sum_{j=i-1}^{i+1}(X_j - X_i)$. With $\kappa = \frac{1}{2\varepsilon}$, from (10) and the above equation, we obtain the law for agent i as:

$$\begin{aligned} v_i = v_{i-1} &- \frac{2\kappa}{\rho_i(\rho_i + p^*(X_i))}\left(\frac{p^*(X_{i+1}) - p^*(X_{i-1})}{X_{i+1} - X_{i-1}}\right) \\ &\times \sum_{j=i-1}^{i+1}(X_j - X_i). \end{aligned}$$

With $v_l = 0$. Now, (11) is discretized to:

$$\partial_t \beta = 1 - \beta(t) - 2\kappa(X_r - X_{r-1}).$$

IV. Self-organization in two dimensions

In this section, we present the two-dimensional self-organization problem. Although our approach to the 2D problem is fundamentally similar to the 1D case, we encounter a problem in the two-dimensional case that did not require consideration in one dimension, and it is the need to control the shape of the spatial domain in which the agents are distributed. We overcome this problem by controlling the shape of the domain with the agents on the boundary, while controlling the density distribution of the agents in the interior.

Let $M \in \mathbb{R}^2$ be the spatial domain in which the agents are distributed, and $\rho : \mathbb{R} \times M \to \mathbb{R}_{>0}$ the density function defined on the domain. Without loss of generality, we shift the origin to a point on the boundary of the domain, such that $(0,0) \in \partial M$. Let $\rho^* : M^* \to \mathbb{R}_{>0}$ be the desired density distribution, where M^* is the target spatial domain. Just as in the 1D case, the agents do no have access to their positions.

In what follows we present our strategy to solve this problem, which we divide into three stages. In the first stage the agents converge to the target spatial domain M^* with the boundary agents controlling the shape of the domain. In stage two, the agents implement the pseudo-localization algorithm to compute the coordinate transformation. In the third stage, the boundary agents remain stationary and the agents in the interior converge to the desired density distribution. We begin by presenting the pseudo-localization algorithm.

A. Pseudo-localization algorithm for the swarm in 2D

In this subsection, we present the pseudo-localization algorithm for the agents in the interior of the spatial domain. We first describe the idea of the coordinate transformation (diffeomorphism) we employ and construct a PDE that converges asymptotically to this diffeomorphism. We then discretize the PDE to obtain the distributed pseudo-localization algorithm.

We use the idea of harmonic maps to construct a coordinate transformation (diffeomorphism) from the spatial domain of the swarm to a unit disk.

Let $M \in \mathbb{R}^2$ (compact) be the spatial domain and $N = \{(x,y) \in \mathbb{R}^2 : (x-1)^2 + y^2 \leq 1\}$ the unit disk. The manifolds M and N are both equipped with Euclidean metrics $g_{ij} = h_{ij} = \delta_{ij}$. Let $\Gamma : \partial M \to [0,1)$ be a parametrization of the boundary of M. Let $\psi : \bar{M} \to N$ be any diffeomorphism that takes the following form on the boundary of M:

$$\psi(\Gamma^{-1}(\gamma)) = (1 - \cos(2\pi\gamma), \sin(2\pi\gamma)), \quad \gamma \in [0,1), \tag{13}$$

and we know that $\Gamma^{-1}[0,1) = \partial M$. From Lemma 5, there is a unique harmonic diffeomorphism $\mathbf{R} : M \to N$ such that $\mathbf{R} = \psi$ on ∂M. We know, by definition, that $\mathbf{R} = (X, Y)$ satisfies:

$$\begin{gathered} \begin{cases} \Delta X = 0, \\ \Delta Y = 0, \end{cases} \quad \text{for } \mathbf{r} \in \text{int } M, \\ \mathbf{R} = \psi, \quad \text{on } \partial M, \end{gathered} \tag{14}$$

where Δ is the Laplace operator.

Now, we define a function $p^* : N \to \mathbb{R}_{>0}$ by $p^* = \rho^* \circ \mathbf{R}^{*-1}$, (where $\mathbf{R}^*$ is the harmonic diffeomorphism from M^* to N) the image of the desired spatial density distribution on the unit disk.

We now construct a PDE that asymptotically converges to the harmonic diffeomorphism, which we then discretize to obtain the distributed pseudo-localization algorithm. We use the heat flow equation as the basis for the algorithm, which yields a harmonic map as its asymptotically stable steady state solution. Let $M \subset \mathbb{R}^2$ be a compact spatial domain, N be the unit disk as defined before, $\mathbf{R} = (X, Y) : M \to N$, and $\mathbf{v}$ the velocity field, such that:

$$\begin{gathered} \begin{cases} \partial_t X = \Delta X - \nabla X \cdot \mathbf{v}, \\ \partial_t Y = \Delta Y - \nabla Y \cdot \mathbf{v}, \end{cases} \quad \text{for } \mathbf{r} \in \text{int } M, \\ \mathbf{R} = \psi, \quad \text{on } \partial M. \end{gathered} \tag{15}$$

The heat flow equation (Equation (15) with $\mathbf{v} = 0$) has been studied extensively in the literature. For well-known existence and uniqueness results, we refer the reader to [15].

Lemma 8 ([13])**.** *The system* (15) *with* $\mathbf{v} = 0$ *(stationary case), is asymptotically stable at the harmonic map* (14). •

We now discretize (15) to get the distributed pseudo-localization algorithm. We have $\rho : M \to \mathbb{R}_{>0}$, the density distribution of the swarm on the domain M. We view the swarm as a discrete approximation of the domain M with density ρ, and it follows that the PDE (15) is approximated by a distributed algorithm by the swarm. Equivalently, the task of discretization is to propose a candidate distributed algorithm, which would yield the heat flow equation via a continuum approximation. We begin with a weighted Laplacian-based consensus algorithm, owing to the connection between Laplacian consensus and the heat flow equation.

$$X_i(t+1) = X_i(t) + \frac{1}{(d_i+1)} \sum_{j\in\mathcal{N}_i} w_{ij}(X_j - X_i), \qquad (16)$$

where $d_i = \sum_{j\in\mathcal{N}_i} w_{ij}$. The same holds for Y, which we skip at this stage for brevity. Noting that the neighbours $j \in \mathcal{N}_i$ of i are also the spatial neighbours of i in M, such that $r_j \in B_\varepsilon(r_i)$ (a ball of radius ε centered at r_i), and using $X_i(t+1) - X_i(t) = \frac{dX}{dt}\Delta t$, we make a continuum approximation of (16):

$$\frac{dX}{dt}\Delta t = \frac{1}{(d(\mathbf{r})+1)} \int_{B_\varepsilon(\mathbf{r_i})} \rho(\mathbf{s}) w(\mathbf{r}_i,\mathbf{s})(X(t,\mathbf{s}) - X(t,\mathbf{r_i}))d\mu_{\mathbf{s}},$$

where $w \in C^\infty(M \times M)$, $w(\mathbf{r}_i,\mathbf{r}_j) = w_{ij}$ and $d(\mathbf{r}) = \int_{B_\varepsilon(\mathbf{r_i})} \rho(\mathbf{s}) w(\mathbf{r},\mathbf{s}) d\mu_{\mathbf{s}}$. Now, for very small ε (neglecting the $\mathcal{O}(\varepsilon^3)$ term), the above equation reduces to:

$$\frac{dX}{dt}\Delta t = k\rho w \left[\Delta X + 2\frac{\nabla\rho}{\rho}\cdot\nabla X + 2\frac{\nabla_{\mathbf{s}} w}{w}\cdot\nabla X\right],$$

where $k = \frac{1}{4\varepsilon}\int_{B_\varepsilon(\mathbf{r_i})}(\mathbf{s}-\mathbf{r})\cdot(\mathbf{s}-\mathbf{r})d\mu_{\mathbf{s}}$, $\nabla_{\mathbf{s}} w$ is the gradient of w w.r.t the second argument $\mathbf{s}$ in $w(\mathbf{r},\mathbf{s})$. With the choice $w(\mathbf{r},\mathbf{s}) = \frac{1}{\rho(\mathbf{s})}$ (where in the discrete setting this corresponds to $w_{ij} = \frac{1}{\rho_j}$), where w_{ij} is the estimate of $w(\mathbf{r}_i,\mathbf{r}_j)$, and with the choice $\Delta t = k$, we obtain:

$$\frac{dX}{dt} = \frac{\partial X}{\partial t} + \mathbf{v}\cdot\nabla X = \Delta X,$$

which is the PDE (15). Therefore, after substituting in (16), we get the distributed pseudo-localization algorithm for the agents in the interior of the swarm to be:

$$\begin{aligned} X_i(t+1) &= X_i(t) + \frac{1}{(d_i+1)} \sum_{j\in\mathcal{N}_i} \frac{1}{\rho_j}(X_j - X_i), \\ Y_i(t+1) &= Y_i(t) + \frac{1}{(d_i+1)} \sum_{j\in\mathcal{N}_i} \frac{1}{\rho_j}(Y_j - Y_i). \end{aligned} \qquad (17)$$

For the agents on the boundary, we have:

$$\mathbf{R}_i = (X_i, Y_i) = \psi_i,$$

where ψ_i is the estimate of $\psi(\mathbf{r}_i)$ for $\mathbf{r}_i \in \partial M$.

B. Localization of the boundary agents

Here, we present a localization algorithm for the agents on the boundary. We assume that the agents have a boundary detection capability (can approximate the normal to the boundary), the ability to communicate with neighbors immediately on either side along the boundary curve, and can compute the density of agents along the boundary.

Let $M \subset \mathbb{R}^2$ be a compact 2-manifold with boundary ∂M and let $(0,0) \in \partial M$ (i.e., one of the agents on the boundary of the swarm is considered to be the origin). First, the agents on ∂M implement the distributed 1D pseudo-localization algorithm presented in Section [1D pseudo]. This yields a parametrization of the boundary $\Gamma : \partial M \to [0,1)$, with $\Gamma(0,0) = 0$, such that the closed curve which is the boundary ∂M is identified with the interval $[0,1)$. We have that, for $\gamma \in [0,1)$, $\Gamma^{-1}(\gamma) \in \partial M$. For $\gamma \in [0,1)$, let $s(\gamma)$ be the arc length of the curve ∂M from the origin, such that $s(0) = 0$ and $\lim_{\gamma\to 1} s(\gamma) = l$. We assume that the boundary agents have access to the unit normal $\mathbf{n}(\gamma)$ to the boundary, and thus the unit tangent $\mathbf{s}(\gamma)$.

Let $q : [0,l) \to \mathbb{R}_{>0}$ denote the normalized density of agents on the boundary, such that we have $\int_0^l q(s)ds = 1$. Now the 1D localization algorithm of Section III-A serves to provide a 2D localization as follows. Note that $\frac{ds}{d\gamma} = \frac{1}{q(\gamma)}$, and $(dx,dy) = \mathbf{s}ds$, which implies $(dx,dy) = \frac{1}{q(\gamma)}\mathbf{s}(\gamma)d\gamma$. Therefore, we get the position of the boundary agent γ, $(x(\gamma),y(\gamma)) = \int_0^\gamma \frac{1}{q(\bar\gamma)}\mathbf{s}(\bar\gamma)d\bar\gamma$, and the arc-length $s(\gamma) = \int_0^\gamma \frac{1}{q(\bar\gamma)}d\bar\gamma$, which is implemented in the discrete setting by finite differencing, since the agents have access to $q(\cdot)$ and $\mathbf{s}(\cdot)$. This way, the boundary agents are localized at time $t = 0$, and they update their position estimates using their velocities, for $t \geq 0$.

C. Distributed control in the continuum domain and analysis

In this section, we derive the distributed feedback control law to converge to the desired density distribution over the target domain in the two-dimensional case. The swarm dynamics are given by:

$$\begin{aligned} \partial_t\rho &= -\nabla\cdot(\rho\mathbf{v}), \quad \text{for } \mathbf{r} \in \text{int } M(t), \\ \partial_t\mathbf{r} &= \mathbf{v}, \quad \text{on } \partial M(t). \end{aligned} \qquad (18)$$

1) Stage 1: In this stage, the objective is for the swarm to converge to the target spatial domain M^*.

Let $\mathbf{r}^* : [0,1] \to \partial M^*$ be the closed curve describing the boundary. Let $\mathbf{e}(\gamma) = \mathbf{r}(\gamma) - \mathbf{r}^*(\gamma)$ be the position error of agent γ on the boundary, where $\mathbf{r}(\gamma)$ is the actual position of agent γ computed as presented in section IV-B.

Let $\phi : \mathbb{R} \times M \to \mathbb{R}$ be a smooth function. Let $M(t)$ be the spatial domain at time t. We will use a heat flow-based method to design the distributed control law for swarm motion as follows:

$$\partial_t\phi = \begin{cases} \Delta\phi, & \text{for } \mathbf{r} \in \text{int } M(t), \\ -\frac{1}{2}|\nabla\phi|^2 - \mathbf{e}\cdot\mathbf{n} - \nabla\phi\cdot\mathbf{n}, & \text{on } \partial M(t), \end{cases} \qquad (19)$$

And:

$$\mathbf{v} = \begin{cases} \nabla\phi, & \text{for } \mathbf{r} \in \text{int } M(t), \\ (\nabla\phi\cdot\mathbf{n})\mathbf{n} - (\mathbf{e}\cdot\mathbf{s})\mathbf{s}, & \text{on } \partial M(t), \end{cases} \qquad (20)$$

where $\mathbf{n}$ and $\mathbf{s}$ are the unit normal and unit tangent to the boundary.

Theorem 2. *The system* (18)*, with the distributed control law* (19) *and* (20) *asymptotically converges to the target spatial domain M^*.*

Proof. We consider an energy functional E_1 given by:

$$E_1 = \frac{1}{2}\int_{M(t)} |\nabla\phi|^2 + \frac{1}{2}\int_{\partial M(t)} |\mathbf{e}|^2.$$

The time derivative $\partial_t E_1$, under the dynamics (18) and the control law (19) and (20) is given by (we omit the steps owing to space constraints [13]):

$$\partial_t E_1 = -\int_{M(t)} |\Delta\phi|^2 - \int_{\partial M(t)} |\nabla\phi\cdot\mathbf{n}|^2 - \int_{\partial M(t)} |\mathbf{e}\cdot\mathbf{s}|^2.$$

Clearly, $\partial_t E_1 \leq 0$, which implies that the system converges to $(\partial_t E_1)^{-1}\{0\}$ (say, at time t_∞), and $\partial_t E_1 = 0$ if and only if $\Delta\phi = 0$ a.e in $M(t_\infty)$, $\nabla\phi\cdot\mathbf{n} = 0$ and $\mathbf{e}\cdot\mathbf{s} = 0$ on $\partial M(t_\infty)$. It follows that:

$$\begin{aligned}\int_{M(t_\infty)} |\nabla\phi|^2 &= \int_{M(t_\infty)} \nabla\phi\cdot\nabla\phi \\ &= \int_{\partial M(t_\infty)} \phi(\nabla\phi\cdot\mathbf{n}) - \int_{M(t_\infty)} \phi\Delta\phi = 0.\end{aligned}$$

Hence, $\nabla\phi = 0$ a.e in $M(t_\infty)$, and from (19) we have $\partial_t\phi = 0$ necessarily on $\bar{M}$, which implies that the system converges to $\mathbf{e} = \mathbf{e_n}\mathbf{n} + \mathbf{e_s}\mathbf{s} = 0$, which corresponds to the domain M^*. Therefore, the swarm converges to the target spatial domain M^* at the end of Stage 1. □

2) Stage 2: Stage two consists of the agents in the swarm implementing the pseudo-localization algorithm presented in Section IV-A. Since the agents are distributed across the target spatial domain M^*, implementing the pseudo-localization algorithm yields the coordinate transformation $\mathbf{R}^*$. We therefore have $\partial_t\mathbf{R}^* = 0$, which implies that $\frac{d\mathbf{R}^*}{dt} = \partial_t\mathbf{R}^* + \nabla(\mathbf{R}^*)\mathbf{v} = \nabla(\mathbf{R}^*)\mathbf{v}$, which will be used in Stage 3.

3) Stage 3: In this stage, the boundary agents of the swarm remain stationary and interior agents converge to the desired density distribution.

Consider the distributed control law, defined as follows for all time t:

$$\begin{cases} \frac{d\mathbf{v}}{dt} = -\rho\nabla(\rho - p^*\circ\mathbf{R}^*) + (\mathbf{v}\cdot\nabla)\mathbf{v} - \mathbf{v}, & \text{for } \mathbf{r}\in \text{int } M^*, \\ \mathbf{v} = 0, & \text{on } \partial M^*, \end{cases} \tag{21}$$

where $\frac{d\mathbf{v}}{dt}$ at $\mathbf{r}\in M$ is the acceleration of the agent at $\mathbf{r}$, the control input. Using the relation $\frac{d}{dt} = \partial_t + \mathbf{v}\cdot\nabla$, it follows from (21) that $\partial_t\mathbf{v} = -\rho\nabla(\rho - p^*\circ\mathbf{R}^*) - \mathbf{v}$.

Theorem 3. *The system* (18) *with $M(t) = M^*$, under the distributed control law* (21)*, is asymptotically stable at $\rho = \rho^*$ almost everywhere.*

Proof. We consider an energy functional E_3 given by:

$$E_3 = \frac{1}{2}\int_{M^*} |\rho - p^*\circ\mathbf{R}^*|^2 + \frac{1}{2}\int_{M^*} |\mathbf{v}|^2.$$

The time derivative $\partial_t E_3$, is given by (steps omitted owing to space constraints [13]):

$$\partial_t E_3 = -\int_{M^*} |\mathbf{v}|^2.$$

Hence, the system converges to $(\partial_t E_3)^{-1}\{0\}$, which corresponds to $\mathbf{v} = 0$ and $\partial_t\mathbf{v} = 0$. This therefore implies that $\nabla(\rho - p^*\circ\mathbf{R}^*) = 0$ a.e., from the choice of $\partial_t\mathbf{v}$. Using the Poincare-Wirtinger inequality, we get that $\|(\rho - p^*\circ\mathbf{R}^*) - (\rho - p^*\circ\mathbf{R}^*)_{M^*}\| \leq C\|\nabla(\rho - p^*\circ\mathbf{R}^*)\| = 0$, where $(\rho - p^*\circ\mathbf{R}^*)_{M^*} = \frac{1}{|M^*|}\int_{M^*}(\rho - p^*\circ\mathbf{R}^*)$. Since $\int_{M^*}\rho = \int_N p^* = \int_{M^*} p^*\circ\mathbf{R}^* = 1$, we have $(\rho - p^*\circ\mathbf{R}^*)_{M^*} = 0$, and therefore $\rho = p^*\circ\mathbf{R}^*$ a.e. in M^*. We know that $p^*\circ\mathbf{R}^* = \rho^*$ and therefore, $\rho = p^*\circ\mathbf{R}^* = \rho^*$, which is the desired density distribution. □

D. Discrete Implementation

1) Computing the Jacobian: When the steady-state is reached in the pseudo-localization algorithm (17) (i.e., $X_i(t+1) = X_i(t)$ and $Y_i(t+1) = Y_i(t)$), we have, $\forall\ i\in\mathscr{S}$:

$$X_i = \frac{\sum_{j\in\mathscr{N}_i}\frac{1}{\rho_j}X_j}{\sum_{j\in\mathscr{N}_i}\frac{1}{\rho_j}},\quad Y_i = \frac{\sum_{j\in\mathscr{N}_i}\frac{1}{\rho_j}Y_j}{\sum_{j\in\mathscr{N}_i}\frac{1}{\rho_j}}, \tag{22}$$

where i is the index of the agent located at $\mathbf{r}\in M$ and $\mathscr{N}_i$ is the set of agents in a disk-shaped neighborhood $B_\varepsilon(\mathbf{r})$ of area ε centered at $\mathbf{r}$. We assume that the agents have the capability in their hardware to perturb the disk of communication $B_\varepsilon(\mathbf{r})$ (by moving an antenna, for instance). The Jacobian $J = \nabla\mathbf{R}$, where $\mathbf{R} = (X,Y)$ is computed through perturbations to $\mathscr{N}_i$ (i.e., the neighborhood $B_\varepsilon(\mathbf{r})$) and using the approximations:

$$\partial_x X = \frac{X(\mathbf{r}+\delta x) - X(\mathbf{r})}{\delta x},\quad \partial_y X = \frac{X(\mathbf{r}+\delta y) - X(\mathbf{r})}{\delta y},$$

and similarly for Y. Now, $X(\mathbf{r}+\delta x)$ is computed as in (22) for $\mathscr{N}_i^{\delta x}$, the set of agents in $B_\varepsilon(\mathbf{r}+\delta x)$ and $X(\mathbf{r}+\delta x)$ from $B_\varepsilon(\mathbf{r}+\delta y)$.

2) Computing the spatial gradient using the Jacobian: Let $J(\mathbf{r}) = \nabla\mathbf{R}(\mathbf{r})$ be the Jacobian of the diffeomorphism $\mathbf{R}: M\to N$ at $\mathbf{r}\in M$. Clearly, J is non-singular. Let $\nabla = (\partial_x,\partial_y)$ and $\bar{\nabla} = (\partial_X,\partial_Y)$, where $\mathbf{R} = (X,Y)$. We have $\partial_x = (\partial_x X)\partial_X + (\partial_x Y)\partial_Y$ and $\partial_y = (\partial_y X)\partial_X + (\partial_y Y)\partial_Y$. Therefore, $\nabla = J^\top\bar{\nabla}$. For a smooth function $f: M\to\mathbb{R}$, we have, $\nabla f = J^\top\bar{\nabla} f$, and the agents numerically compute $\bar{\nabla}$ by:

$$\left(\frac{\partial f}{\partial X}\right)_i = \frac{1}{|\mathscr{N}_i|}\sum_{j\in\mathscr{N}_i}\frac{f_j - f_i}{X_j - X_i},\quad \left(\frac{\partial f}{\partial Y}\right)_i = \frac{1}{|\mathscr{N}_i|}\sum_{j\in\mathscr{N}_i}\frac{f_j - f_i}{Y_j - Y_i},$$

where i is the index of the agent located at $\mathbf{r}\in M$ and $\mathscr{N}_i$ is the set of agents in a disk-shaped neighbourhood of area ε centered at $\mathbf{r}$.

3) Computing the spatial gradient without the Jacobian: In the absence of a Jacobian estimate, we use the following alternative method for computing an approximate spatial gradient estimate. This is used in Stage 1 of the self-organization process. Let $\bar{f}(\mathbf{r})$ be the mean value of f over a small disk-shaped neighbourhood of area ε centered at $\mathbf{r}$, where $\bar{f}(\mathbf{r}) = \frac{1}{\varepsilon}\int_{B_\varepsilon(\mathbf{r})} f d\mu = \frac{1}{|\mathscr{N}_i|}\sum_{j\in\mathscr{N}_i} f_j$. We have from [13], $\overline{\left(\overline{\left(\frac{\partial f}{\partial x}\right)},\overline{\left(\frac{\partial f}{\partial y}\right)}\right)} = \frac{1}{\varepsilon}\left(\frac{\partial\bar{f}}{\partial x},\frac{\partial\bar{f}}{\partial y}\right)$, which the agents use as the estimate of the gradient ∇f in the numerical algorithms.

V. Numerical Simulation

In this section, we present a numerical simulations of swarm self-organization, that is, of the control laws presented in Sections III-B and IV-C.

A. 1D Self-organization

In the simulation of the 1D case, we consider a swarm of $N = 10000$ agents, the desired density distribution given by $\rho^*(x) = a\sin(x) + b$, where $a = 1 - \frac{\pi}{2N}$ and $b = \frac{1}{N}$, $x \in \left[0, \frac{\pi}{2}\right]$. We use a kernel-based method as a scheme to approximate the continuous density function. We discretize the spatial domain with $\Delta x = 0.001$ units, and use an adaptive time step scheme. The self-organization begins from an arbitrary initial density distribution. Figure 1 shows the initial density distribution, an intermediate distribution and the final distribution. We observe that there is convergence to the desired density distribution, even with noisy density measurements.

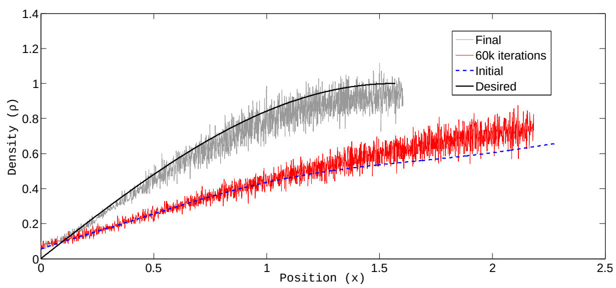

Fig. 1: Numerical Simulation of a 10000-agent swarm

B. 2D Self-organization

In the simulation of the 2D case, we focus on Stage 3 of the self-organization process, where the agents already distributed over the target spatial domain, converge to the desired density distribution. The target spatial domain, a circle of radius 0.5 units, given by $M^* = \{(x, y) \in \mathbb{R}^2 : (x - 0.6)^2 + y^2 \leq 0.25\}$, with the desired density distribution ρ^* given by $\rho^*(x, y) = \frac{1}{\left((x-0.4)^2+y^2\right)^{0.3}}$. The initial density distribution of the swarm is uniform, and the distributed control law of Stage 3 in Section IV-C, following the discretization scheme outlined in Section IV-D is implemented. Figure 2 shows the spatial density error plot, where $e(\rho) = \int_{M^*} |\rho - \rho^*|^2$ is the spatial density error.

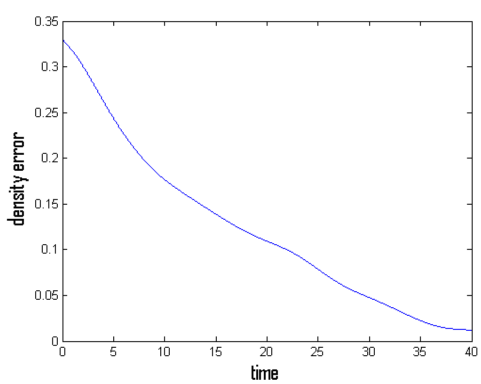

Fig. 2: Spatial density error $e(\rho) = \int_{M^*} |\rho - \rho^*|^2$ vs time

VI. Conclusions

In this paper, we considered the problem of self-organization in multi-agent swarms, in one and two dimensions, respectively. The primary contribution of this paper is the analysis and design of position and index-free distributed control laws for swarm self-organization for a large class of configurations. This was accomplished through the introduction of a distributed pseudo-localization algorithm that the agents implement to find their position identifiers, which then use in their control laws. The validation of the results for more general non-simply connected domains will be considered in the future. An extension to this work will involve the characterization of constraints on the local density function to capture finite robot sizes and collision avoidance constraints, as well as identifying the dynamic restrictions on densities to reflect the dynamical constraints of robots.

VII. Acknowledgement

The authors would like to thank Prof. Lei Ni at the UCSD Mathematics Department for his valuable inputs.

References

[1] S. Camazine, *Self-organization in biological systems*. Princeton University Press, 2003.
[2] G. M. Whitesides and B. Grzybowski, “Self-assembly at all scales,” *Science*, vol. 295, 2002.
[3] M. Rubenstein, C. Ahler, N. Hoff, A. Cabrera, and R. Nagpal, “Kilobot: A low cost robot with scalable operation designed for collective behaviors,” *Journal of Robotics and Autonomous Systems*, vol. 62, 2014.
[4] N. Demir and B. Açıkmeşe, “Density control for decentralized autonomous agents with conflict avoidance,” in *IFAC World Congress*, 2014.
[5] S. Bandyopadhyay, S. J. Chung, and F. Y. Hadaegh, “Inhomogeneous markov chain approach to probabilistic swarm guidance algorithms,” in *Int. Conf. on Spacecraft Formation Flying Missions and Technologies*, 2013.
[6] I. Chattopadhyay and A. Ray, “Supervised self-organization of homogeneous swarms using ergodic projections of markov chains,” *IEEE Transactions on Systems, Man, & Cybernetics. Part B: Cybernetics*, vol. 39, 2009.
[7] J. M. Jonathan Bachrach, Jacob Beal, “Composable continuous-space programs for robotic swarms,” *Neural Computing and Applications*, vol. 19, no. 6, pp. 825–847, 2010.
[8] P. Frihauf and M. Krstic, “Leader-enabled deployment onto planar curves: A pde-based approach,” *IEEE Transactions on Automatic Control*, vol. 56, 2011.
[9] J. Qi, R. Vazquez, and M. Krstic, “Multi-agent deployment in 3-d via pde control,” *IEEE Transactions on Automatic Control*, vol. 60, 2015.
[10] G. Foderaro, S. Ferrari, and T. A. Wettergren, “Distributed optimal control for multi-agent trajectory optimization,” *Automatica*, vol. 50, 2014.
[11] M. Rubenstein, A. Cornejo, and R. Nagpal, “Programmable self-assembly in a thousand-robot swarm,” *Science*, vol. 345, 2014.
[12] A. J. Chorin and J. E. Marsden, *A mathematical introduction to fluid mechanics*, vol. 3. Springer, 1990.
[13] V. Krishnan and S. Martínez, “Self-organization in multi-agent swarms via distributed computation of diffeomorphisms.” Extended version at http://fausto.dynamic.ucsd.edu/sonia/papers/data/2015-VK-SM.pdf, 2015.
[14] G. Leoni, *A first course in Sobolev spaces*, vol. 105. American Mathematical Society Providence, RI, 2009.
[15] J. Eells and L. Lemaire, “Deformations of metrics and associated harmonic maps,” *Proceedings Mathematical Sciences*, vol. 90, 1981.
[16] M. Krstic and A. Smyshlyaev, *Boundary control of PDEs: A course on backstepping designs*, vol. 16. SIAM, 2008.