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

Vishaal Krishnan

PhD Student
Mechanical and Aerospace Enginering
University of California, San Diego
v6krishn@ucsd.edu

22nd International Symposium on Mathematical Theory of Networks and Systems

Advisor **Prof. Sonia Martínez**

Multi-Agent Swarms

- Swarms are large collectives of dynamic agents
- Agents interact with one another via sensing and/or communication

Robotic swarms

- Large-scale, distributed nature potentially offers robustness
- Propelled by the development of low-cost sensing, communication and computational systems

Figure: The Kilobot; A Kilobot swarm; Robots constructing a structure

- Applications include monitoring, manipulation and construction

Self-organization in Swarms

Emergence of long-range order from local interactions in large-scale multi-agent systems

Self-organization in nature

Figure: Fish school; Fractal patterns in Broccoli; Bacterial colony

Self-organization in engineered systems

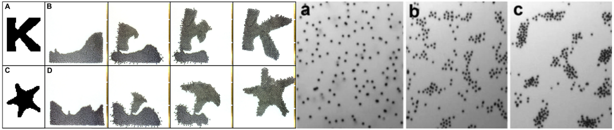


Figure: Kilobots; Micromotors

Modeling Swarms

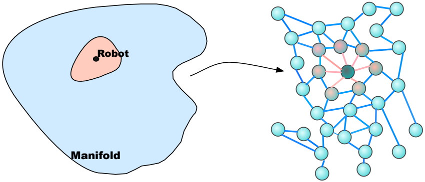
Properties of swarms

- Individual agents are insignificant, only macroscopic objectives matter
- Performance unaffected by the removal of a small number of agents
- Specifying the states of individual agents is impractical and ineffective
- Macroscopic quantities (spatial density profile, etc) are more appropriate to specify swarm configuration
- Agents interact (sense/communicate) with **nearest spatial neighbors**

Modeling Swarms

Continuum abstraction of swarms

- Viewing a swarm as a discretization of a continuous space (manifold)
- Every point in the manifold is a computational device (agent)



- Objectives specified in the continuum domain
- Algorithms and control laws designed for the continuum

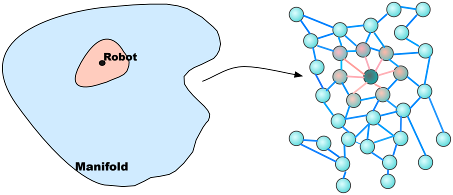
Selected References

Continuum abstractions of multi-agent/computing systems not new:

- **Spatial computing**
  - "Space-time programming," Jacob Beal
- **Agent coordination for curve / surface formation**
  - "Multi-agent deployment in 3D via PDE control," Qi, Vazquez, Krstic, 2015
  - "Leader-enabled deployment onto planar curves," Frihauf, Krstic, 2011

(require pre-assignment of indices to agents)

Modeling Swarms - Continuum Abstraction



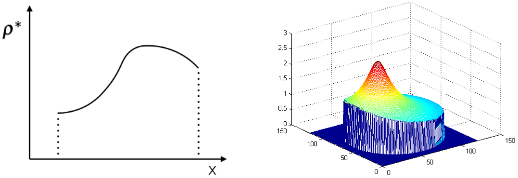
- Number of agents N very large ($N \to \infty$)
- Agents embedded in a bounded open set $\Omega \subset \mathbb{R}^d$ with spatial density distribution ρ
- Density distribution normalized ($\int_\Omega \rho = 1$)
- **Conservation of agents**: $\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0$, v is the velocity field
- The above is also the **continuity equation** from fluid dynamics

Outline of the Talk

I. Problem formulation
II. Outline of approach
III. Self-organization in one dimension
IV. Self-organization in two dimensions
V. Conclusions and Future work

Problem Formulation

Achieve a macroscopic spatial density profile $\rho^* : M^* \to \mathbb{R}_{>0}$ over a target spatial domain M^*



Assumptions

- Agents have computation and communication capabilities
- Agents can measure local density, **but do not have position information** (key aspect of this work)
- Agents do **not** have pre-assigned indices
- Agents know the **true x and y directions**
- Boundary agents can determine outward unit normal to the boundary

Problem formulation

Conservation law

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0 \tag{1}$$

$v(t, r)$ is the velocity of agent at r at time t

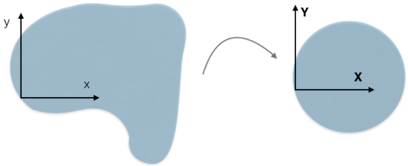
The **objective** is to design $v(t, r)$ to achieve $\rho \to \rho^*$

However, the agents do not know their position r, which implies that

- The agents cannot access the value of the desired local density $\rho^*(r)$
- A velocity input $v(t, r)$, with argument r, cannot be implemented

Outline of Approach

1. **Construct a coordinate transformation (Diffeomorphism)** $r \mapsto R^*$



   Map spatial position r to an artificial coordinate $R^*(r)$

2. **Design a distributed algorithm** for the agents to compute $R^*(r)$

3. **Determine desired density profile (offline)** in transformed coordinates

   Density profile p^* in the new coordinates, such that $p^* = \rho^* \circ {R^*}^{-1}$

4. **Design a distributed control law** to achieve $\rho \to p^* \circ R^* = \rho^*$ under the dynamics in Equation (1)

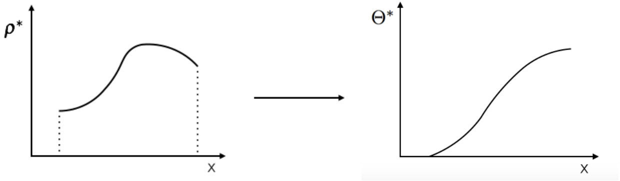
Self-organization in 1D - Coordinate Transformation

1. What coordinate transformation?

For every $\rho : [0, L] \to \mathbb{R}_{>0}$, there is a natural coordinate transformation Θ given by:

$$\Theta(x) = \int_0^x \rho(x)dx$$

the cumulative distribution function (with $\Theta(L) = 1$)



Self-organization in 1D - Coordinate Computation

2. Distributed algorithm to compute the coordinate transformation

We use the diffusion PDE to converge to Θ

$$\begin{aligned}
&\partial_t X = \frac{1}{\rho}\partial_x \left(\frac{\partial_x X}{\rho}\right) \\
&X(t,0) = \alpha(t) \\
&X(t,L) = \beta(t) \\
&\partial_t \alpha(t) = -\alpha(t) \\
&\partial_t \beta(t) = 1 - \beta(t) \\
&X(0,x) = X_0(x)
\end{aligned}$$

Lemma

The above PDE system converges asymptotically to Θ

Self-organization in 1D - Coordinate Computation

Discretizing the PDE system over a **line graph** yields the averaging consensus dynamics

$$X_i(t+1) = X_i(t) + \frac{1}{3}\sum_{j=i-1}^{i+1}[X_j(t) - X_i(t)]$$

$$X_l(t) = 0$$

$$X_r(t) = \beta(t)$$

$$X_i(0) = X_{0\,i}$$

where the index $'l'$ is for the leftmost agent and $'r'$ the rightmost

This is the distributed algorithm implemented **synchronously** by the agents to compute the coordinate transformation

Self-organization in 1D - Distributed Control

3. **Determine density profile** p^* such that $p^* = \rho^* \circ \Theta^{*-1}$

   where Θ^* is the CDF corresponding to ρ^*

4. **Distributed control law** to achieve $\rho \to p^* \circ X$

The dynamics of the swarm + distributed computation of coordinate transform

$$
\begin{aligned}
&\partial_t \rho = -\partial_x(\rho v) \\
&\partial_t X = \frac{1}{\rho}\partial_x \left(\frac{\partial_x X}{\rho} \right) - v\partial_x X \\
&X(t,0) = 0 \\
&X(t,L(t)) = \beta(t) \\
&X(0,x) = X_0(x)
\end{aligned}
\tag{2}
$$

Self-organization in 1D - Distributed Control

Theorem

The System (2) *with the control law*

$$
\begin{aligned}
v(t,0) &= 0 \\
\partial_x v &= (\rho - p^* \circ X) - \frac{\partial_X p^*}{\rho(\rho + p^* \circ X)} \partial_x \left(\frac{\partial_x X}{\rho} \right) \\
X(t,0) &= 0 \\
\beta_t &= 2 - \beta(t) - \left. \frac{X_x}{\rho} \right|_{L(t)}
\end{aligned}
$$

is asymptotically stable at $\rho = \rho^*$ *and* $X = \Theta^*$

Self-organization in 1D - Discrete Control Law

The control law above is discretized to obtain

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

$$v_l = 0$$

$$\partial_t \beta = \frac{\beta(t+1) - \beta(t)}{\Delta t} = 1 - \beta(t) - 2\kappa \left(X_r - X_{r-1} \right)$$

Simulation

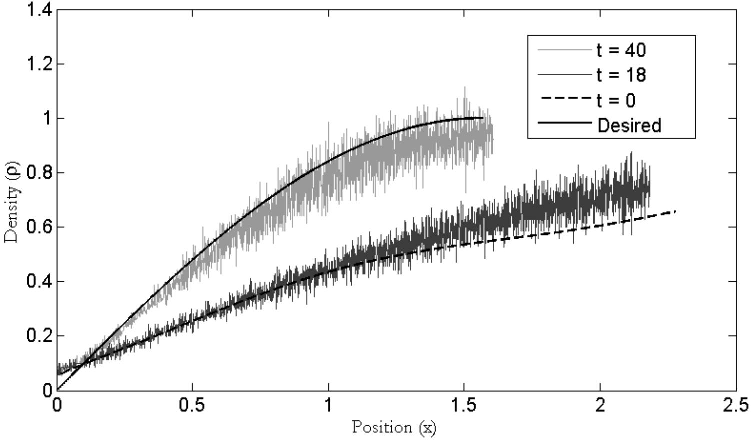


Figure: Density $\rho(x)$ plotted against position x at different instants of time

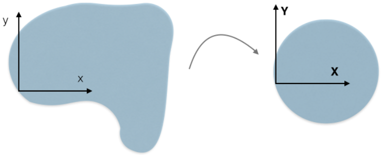
Self-organization in 2D

Difficulty in the 2D case

- Shape control through boundary + density control in the interior

Coordinate transformation

- The **harmonic map** is used to contruct a diffeomorphism
- Agents map themselves harmonically onto a unit disk



- Boundary agents first **implement the 1D algorithm** and map themselves onto a unit circle
- Interior agents implement a **heat flow based algorithm** to compute the diffeomorphism

Harmonic Maps

A map $\mathbf{R} = (X, Y) : M \subset \mathbb{R}^2 \to \mathbb{R}^2$ is called harmonic if it minimizes the functional:

$$E(\mathbf{R}) = \int_M (|\nabla X|^2 + |\nabla Y|^2) d\mu,$$

The Euler-Lagrange equation for the functional E which also yields the minimum is given by $\Delta X = 0$ and $\Delta Y = 0$ (Laplace equations)

Lemma

Let M be a compact surface with boundary and $N \subset \mathbb{R}^2$ another compact surface. Suppose that $\psi : M \to N$ is a diffeomorphism onto $\psi(M)$ such that $\psi(M)$ is convex. Then there is a unique harmonic map $\mathbf{R} : M \to N$ with $\mathbf{R} = \psi$ on ∂M, such that $\mathbf{R} : M \to \mathbf{R}(M)$ is a diffeomorphism.

Heat Flow

Let $\mathbf{R} = (X, Y) : M \to N$ be a harmonic map

$$\begin{cases} \Delta X = 0 \\ \Delta Y = 0 \end{cases} \quad \text{for } \mathbf{r} \in \text{int } M$$

$$\mathbf{R} = \psi \quad \text{on } \partial M$$

Lemma

Solutions to the heat flow equation

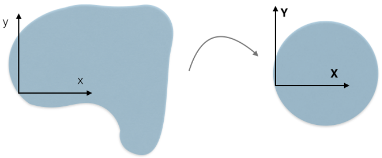
$$\begin{cases} \partial_t X = \Delta X \\ \partial_t Y = \Delta Y \end{cases} \quad \textit{for } \mathbf{r} \in \textit{int } M$$

$$\mathbf{R} = \psi \quad \textit{on } \partial M$$

converge asymptotically to the harmonic map above

Self-organization in 2D - Coordinate Computation

- **Boundary agents** implement the 1D algorithm
  This yields a parametrization of the boundary $\Gamma : \partial M \to [0, 1)$
- Each agent on the boundary is identified by a unique $\gamma \in [0, 1)$
- They then map themselves onto a unit circle through the map $R(\gamma) = (X, Y)(\gamma) = (1 - \cos(2\pi\gamma), \sin(2\pi\gamma))$ which is a diffeomorphism



- The interior agents now implement **heat flow** to converge to the **harmonic diffeomorphism** (coordinate transformation)

Self-organization in 2D - Coordinate Computation

The distributed algorithm for the computation of coordinates is obtained by **discretizing the heat flow equation**

$$X_i(t+1) = X_i(t) + \kappa \sum_{j \in \mathcal{N}_i(t)} \frac{1}{d_j(t)} (X_j(t) - X_i(t))$$

$$Y_i(t+1) = Y_i(t) + \kappa \sum_{j \in \mathcal{N}_i(t)} \frac{1}{d_j(t)} (Y_j(t) - Y_i(t))$$

where $d_j = |\mathcal{N}_j|$ is the number of neighbors of agent j

Self-organization in 2D - Distributed Control

Approach We divide the 2D self-organization process into three stages

- **Stage 1:** Agents converge to the target spatial domain M^*, with boundary agents controlling the shape of the domain
- **Stage 2:** Agents implement the distributed algorithm to compute the coordinate transformation
- **Stage 3:** Boundary agents remain stationary, agents in the interior converge to the desired density distribution

Self-organization in 2D - Distributed Control

The swarm dynamics are given by:

$$\partial_t \rho = -\nabla \cdot (\rho \mathbf{v}), \quad \text{for } \mathbf{r} \in \mathring{M}(t),$$
$$\partial_t \mathbf{r} = \mathbf{v}, \quad \text{on } \partial M(t).$$

Stage 1

Boundary agents first localize themselves

$$(x(\gamma), y(\gamma))^T = \int_0^{\gamma} \frac{\mathbf{s}(\tau)\, \mathrm{d}\tau}{q(\tau)}$$

with tangent to boundary $\mathbf{s}$ and normalized density of boundary agents q

Boundary agents know the **desired boundary curve** $(x^*(\gamma), y^*(\gamma))$

Self-organization in 2D - Distributed Control Stage 1

Theorem

The swarm with the control law

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

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

converges asymptotically to the target spatial domain M^*

- Heat flow based distributed control law

Self-organization in 2D - Distributed Control Stage 3

- Swarm converged to target spatial domain at the end of Stage 1
- In **Stage 2**, agents implement distributed coordinate computation algorithm described earlier
- In **Stage 3** the interior agents converge to the desired density profile

Theorem

The swarm with the control law

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

converges asymptotically to the desired density profile ρ^*

Simulation

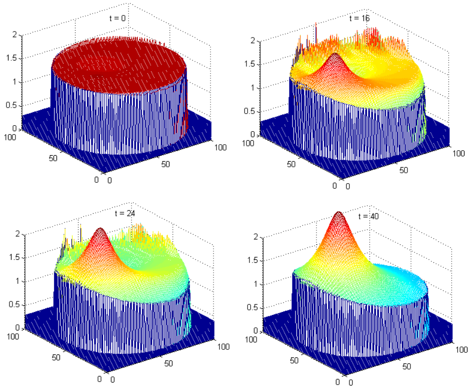


Figure: Evolution of density distribution

Conclusions

- Continuum abstraction of swarms was made owing to the macroscopic nature of objectives
- Objective was to design distributed control laws for spatial self-organization of swarms in 1D and 2D without position measurements
- Coordinate transformations constructed (the **CDF** in 1D and **harmonic diffeomorphism** in 2D) for position-free control laws
- Distributed algorithms for computation of coordinate transformations were designed
- Distributed control laws for self-organization designed with transformed coordinates

Conclusions

Thank you! Questions?