Decentralized Control of Swarm Density via PDE-Constrained Optimization

by

Ziyi Wang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2024

Author's Declaration

I hereby declare that I am the sole author of this report. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The control of swarms of autonomous agents has gained significant attention in fields such as robotics, transportation, and biological systems. Centralized control strategies, while effective, often suffer from scalability issues and high communication demands, motivating the development of decentralized approaches. Decentralized control allows individual agents to make decisions based on locally available information, offering improved scalability and robustness against failures. Achieving comparable performance to centralized methods while maintaining efficiency and reducing energy consumption remains a key challenge.

In this work we present a decentralized control framework for swarm density regulation using Partial Differential Equation (PDE)-constrained optimization, implemented with a Model Predictive Control (MPC) paradigm. We propose a novel formulation which allows each agent in the swarm to optimize its control input using locally available information only. Extensive numerical experiments show the effectiveness of the developed decentralized control framework in reaching desired swarm densities, and compare its performance to centralized control strategies. Specifically, the decentralized approach, with an appropriate choice of parameters, manages to outperform the centralized approach in terms of energy consumption, with little trade off in performance.

Acknowledgements

I would like to express my gratitude to my supervisors, Professor Roberto Guglielmi and Professor Gennaro Notomista, for their invaluable guidance, insightful feedback, and continuous support throughout this work. Their direction, expertise, and encouragement have been profoundly enriching over the past year.

I am also grateful to my family for their love, patience, and support, which have motivated me every step of the way.

Finally, I extend my thanks to my friends, especially my colleagues from the labs, for their constant support and inspiration.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advancement in low-cost robot design [18, 14] has significantly expanded the range of applications of swarm robotics, including precision agriculture, environmental monitoring, and emergency response and rescue [19]. The control of swarm systems constituted by large-scale agents has received increasing attention [15, 6], due to its ability to solve complex problems through decentralized, collective behavior, leading to scalable, flexible, and robust solutions [19].

One effective approach to achieve the control of swarms is through density control. By considering the swarm as a continuum, one can control its density via a velocity field applied to the swarm [20, 7]. Each agent moves according to the value of the velocity field at its position.

This approach has several advantages. First, unlike position-based control, which requires controlling the dynamics of every individual agent, density control focuses only on the swarm density dynamics, reducing complexity and enhancing scalability. Second, in some practical problems, it is more convenient to define control targets in the form of density rather than assigning exact positions to agents. In such cases, density control is not only advantageous but essential. Finally, density can easily incorporate the effects of random noise, such as measurement errors, operational errors, and environmental disturbances, through the diffusion term, providing a natural and intuitive way to model these uncertainties.

However, Such approach typically requires a central controller to acquire the positions of all agents, compute the control variables, and distribute them to each agent. As the number of agents in the swarm and the area of the domain increase, the computational

burden and communication overhead might grow significantly, making centralized control less scalable and more prone to the central controller failures.

Therefore, it is desirable to have a *decentralized* control framework, where the controller synthesis process is solely based on local information [2]. In this way, the approach results in a scalable technique that is robust to individual robot failures and uncertainties [17, 23].

If a control method can combine these two approaches, it would incorporate the advantages of both. On one hand, it can be applied to problems where the control target is defined as a density, while also easily accounting for random noise. On the other hand, thanks to decentralized control, it offers scalability and robustness, making it suitable for large-scale and uncertain environments.

The combination of the two approaches, density control and decentralized control, has been exploited in [5], where the author discretized the state space into a finite union of subspaces and considered how agents transition between these subspaces. This results in a finite-state discrete-time Markov chain, coupled with a decentralized control law for the probability density function.

A different approach is taken in [12], based on the stochastic differential equation (SDE) of each agent dynamics and the design of a specific control law to achieve the target swarm density.

In [13], the authors set the swarm density as the target to achieve voluntary retreat, preventing interference between robots that could reduce overall efficiency, and designed a feedback decentralized control to regulate this behavior. Another feedback decentralized control that relies only on local density measurements is proposed in [3] for a discrete-time Markov process model.

Unlike the aforementioned works, we consider applying decentralized control to the continuous-time evolution of swarm density. In this paper we propose a new approach to decentralized optimal density control, providing the following contributions:

(1) We formulate a novel decoupled centralized control problem and a decentralized density control problem

(2) We design the optimal control for the decentralized density control problem using Partial Differential Equation (PDE)-constrained optimization implemented with a model predictive control (MPC) paradigm, introducing a novel approximation where, during a short time step, agents other than the one optimizing are considered static

(3) We extensively analyze the results of numerical experiments to show how different parameters influence the decentralized control and we compare the proposed approach to centralized control

Chapter 2

Background

In this section, we introduce the centralized swarm density control problem. This problem has been extensively studied, and numerous results including well-posedness and the existence of optimal controls have been obtained, see [20, 9, 4], just to name a few.

2.1 Density Control and Fokker-Planck equation

Consider the dynamics of a swarm of N agents within a domain $\Omega \subset \mathbb{R}^n$ with boundary Γ over time $[0, T]$. The position $X_i(t) \in \Omega$ of each agent i, $i = 1, \cdots, N$ at time $t \in [0, T]$ moves according to an SDE driven by the velocity field $v(x, t)$, $x \in \Omega$, that plays the role of the control input, perturbed by a Wiener process $W(t)$, as follows

$$\begin{cases} dX_i(t) = v(X_i, t) + \sqrt{2\mu} dW(t) + n(X_i(t)) d\psi(t) \\ X_i(0) = X_{i,0}, \end{cases} \tag{2.1}$$

where μ is the diffusion coefficient, $n(x)$ is the normal vector at $x \in \Gamma$, $\psi(t) \in \mathbb{R}$ describes reflecting boundary conditions to ensure that agents remain within Ω, and $X_{i,0}$ is the initial position of agent i. The model in (2.1) is an accurate representation of physical agents affected by process and measurement noises [21]. As in [20], we suppose there is no interaction, collision or communication between the agents. For large N, the evolution of the spatial density $\rho(x, t)$ of the swarm of agents is described by the Fokker-Planck

equation [1]:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (-\mu \nabla \rho + v\rho) = 0 & \text{in } \Omega, \\ (-\mu \nabla \rho + v\rho) \cdot n = 0 & \text{on } \Gamma, \\ \rho(x,0) = \rho_0(x) & \text{in } \Omega, \end{cases} \tag{2.2}$$

where $\rho_0(x)$ is the initial density of the swarm.

In this formulation, for any measurable set $A \subset \Omega$, $\int_A \rho(x,t)dx$ is the expected number of agents in A at time t. We refer to this as the *coupled system* to distinguish it from the decoupled system discussed later.

By considering the density of the swarm instead of the agents' positions, the system reduces to a single partial differential equation instead of N individual stochastic differential equations. The probability density functions of the agents, which further induce the probabilities of their positions, are *coupled* in the dynamic of the swarm density. Such coupling relationship make the system hard to be decentralized.

Defining the set of admissible velocity fields $\mathcal{V}_{ad} = \{v \in L^2(0,T;L^\infty(\Omega)) : \|v(x,t)\| \leq v_M \text{ a.e. } x \in \Omega \text{ and } t \in [0,T]\}$ as in [20], the coupled centralized optimal control problem to reach a target density ρ_T is formulated as:

$$\min_{\rho, v \in \mathcal{V}_{ad}} J_1(\rho, v) = \frac{1}{2}\int_\Omega (\rho(x,T) - \rho_T(x))^2 d\Omega + \frac{\alpha}{2}\int_0^T \int_\Omega \|v(x,t)\|^2 d\Omega\, dt \tag{2.3}$$

subject to (2.2), which can be tackled using PDE-constrained optimization techniques as in [20, 9, 11].

2.2 PDE-Constrained Optimization

In this paper, we consider the adjoint state method and its first-order optimality condition to solve PDE-constrained optimization problems [16]. The objective of this approach is to compute the reduced gradient of the cost function with respect to the control inputs, thereby enabling gradient descent on the control input to achieve local, and potentially global, optimal control.

For a general PDE-constrained optimization problem stated as below:

$$\begin{aligned} \min_{u \in U, x \in X} \quad & J(u,x) = \hat{J}(u) \\ \text{s.t.} \quad & F(u,x) = 0, \end{aligned}$$

where u is the control input, $x(u)$ is the state of the system, and J, F represent the cost function and dynamic equations respectively. The cost function can be written in a reduced form $J(u, x) = \hat{J}(u)$ with the mapping $x = x(u)$.

The adjoint state method can be broadly described in three steps:

(1) Introduce the adjoint state λ and construct the Lagrangian of the original problem.

$$L(x, \lambda, u) = J(u, x) + < \lambda, F(u, x) >$$

(2) Derive the equation for the adjoint state based on the optimality condition. That is, impose the variation of L with respect to x to 0.

$$L'_x[\delta x] = 0$$

(3) Compute the reduced gradient of the cost function with respect to the control inputs according to the variation of L with respect to u. Notably, we have:

$$< \nabla \hat{J}(u), \delta u >= L'_u[\delta u]$$

.

Once we have the reduced gradient $\nabla \hat{J}(u)$, we can perform gradient descent to reach an optimal control.

Chapter 3

Decoupled System

Deriving a decentralized control system from the coupled one (2.3) is challenging because the dynamics apply to the swarm as a whole, rather than to individual agents. For this reson, a decoupled system, where each agent has its own dynamics and control input, is necessary to formulate the decentralized control problem.

For each agent i, the evolution of its position's probability density is given by a Fokker-Planck equation:

$$\begin{cases} \dfrac{\partial \rho^{(i)}}{\partial t} + \nabla \cdot (-\mu \nabla \rho^{(i)} + v^{(i)} \rho^{(i)}) = 0 & \text{in } \Omega, \\ (-\mu \nabla \rho^{(i)} + v^{(i)} \rho^{(i)}) \cdot n = 0 & \text{on } \Gamma, \\ \rho^{(i)}(x, 0) = \rho_0^{(i)}(x) & \text{in } \Omega. \end{cases} \tag{3.1}$$

Although (3.1) shares the same form as the coupled system (2.3), their interpretations differ. In the coupled system, when the number of agents becomes sufficiently large, the system can be regarded as a continuum, where the swarm density represents the mass distribution from a macroscopic perspective. In contrast, in the decoupled system, the density is interpreted as the probability distribution of the agents' positions from a microscopic perspective.

Figure 3.1 illustrates the difference. While both densities are the same Gaussian, the left figure shows how the large-scale swarm distributed according to the density, meanwhile the right figure tells what's the possible positions of a single agent.

The probability density of an individual agent represents the uncertainty regarding its position, allowing for the inclusion of factors such as measurement errors, operational errors, and even intentionally introduced errors.

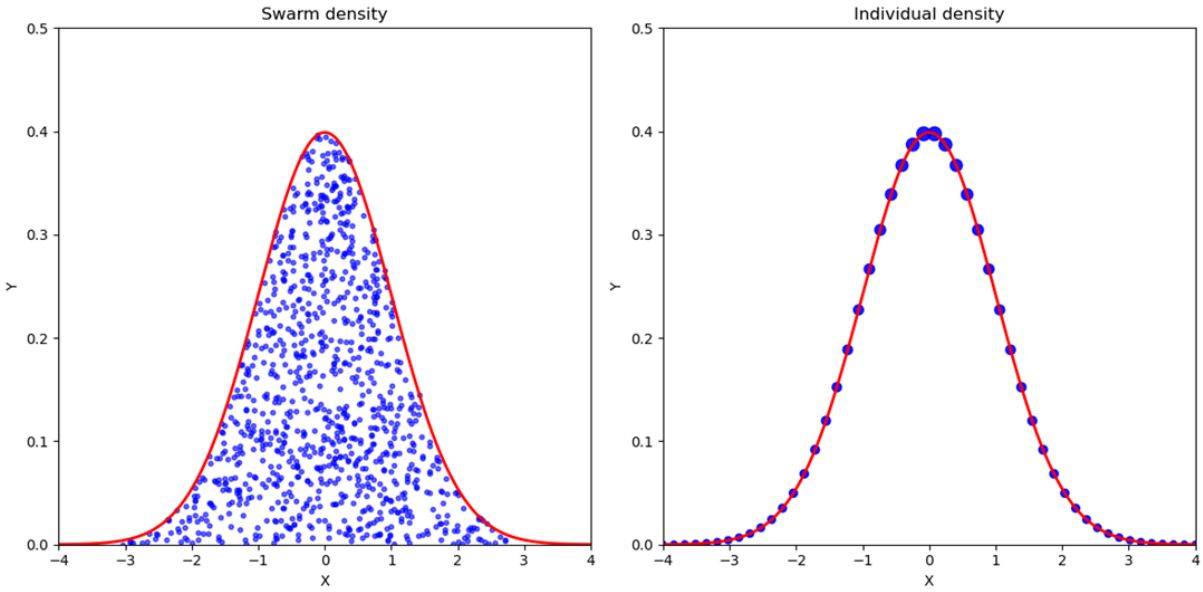


Figure 3.1: Different interpretations of the swarm density and individual density. The points on the left figure represent different agents in a large scale system, while the points on the right figure are the possible positions of one individual agent.

Given that swarm density represents the average of individual agents' probability densities, controlling each agent's probability density—using the velocity field as the control variable—enables control over the swarm density as well. The optimal control problem can then be formulated as

$$\min_{\rho^{(1)},\cdots,v^{(1)},\cdots} \quad J_2(\rho^{(1)},\cdots,v^{(1)},\cdots) = \frac{1}{2}\int_\Omega \left(\frac{\sum_{i=1}^N \rho^{(i)}(x,T)}{N} - \rho_T(x)\right)^2 d\Omega + \frac{\alpha}{2}\sum_{i=1}^N \int_0^T \int_\Omega \|v^{(i)}(x,t)\|^2 d\Omega\, dt \quad (3.2)$$

subject to (3.1) for any $i \in \{1, ..., N\}$. Before proceeding with the analysis of the optimal control problem, the following proposition shows that the decoupled system (3.2) is equivalent—in terms of cost of the optimal solution—to the coupled system (2.3) when regularization term is not applied, that is, when $\alpha = 0$ in (2.3) and (3.2).

Proposition 1. *Let $\alpha = 0$ in (2.3) and (3.2). For any feasible solution $\rho^{(i)}$, $v^{(i)}$ of the decoupled system (3.2), there exists a feasible solution ρ, v of the coupled system (2.3) that has the same swarm density evolution and cost. Moreover, for any feasible solution ρ, v of the coupled system (2.3), there exists a feasible solution $\rho^{(i)}$, $v^{(i)}$ of the decoupled system (3.2) that has the same swarm density evolution and cost.*

Proof. Suppose a swarm system satisfies both the coupled system (2.3) and the decoupled system (3.2). Summing up all the equations in the decoupled system, we have:

$$\frac{\partial(\sum_{i=1}^N \rho^{(i)})}{\partial t} + \nabla \cdot (-\mu\nabla(\sum_{i=1}^N \rho^{(i)}) + \sum_{i=1}^N (v^{(i)}\rho^{(i)})) = 0 \quad (3.3)$$

The relation between the swarm density ρ and agents' densities $\rho^{(i)}$ can be stated as $\rho = \sum_{i=1}^N \rho^{(i)}/N$. Comparing (3.3) with (2.3), we have the relation between the velocity fields:

$$v = \frac{\sum_{i=1}^N v^{(i)}\rho^{(i)}}{\sum_{i=1}^N \rho^{(i)}}.$$

For any feasible solution $\rho^{(i)}$, $v^{(i)}$ of the decoupled system (3.2), let

$$v = \frac{\sum_{i=1}^N v^{(i)}\rho^{(i)}}{\sum_{i=1}^N \rho^{(i)}},\ \rho_0 = \frac{\sum_{i=1}^N \rho_0^{(i)}}{N}.$$

The solution ρ to (2.3), with ρ_0, v being the initial density and velocity field, satisfies $\rho = \sum_{i=1}^{N} \rho^{(i)}/N$, and thus $J_1(\rho, v) = J_2(\rho^{(i)}, v^{(i)})$.

On the other hand, for any feasible solution ρ, v of the coupled system (2.3), let $v^{(i)} = v$ and $\rho_0^{(i)} = \rho_0$. The solution ρ of (3.1), with $\rho_0^{(i)}$, $v^{(i)}$ being the initial densities and velocity fields, satisfies $\rho = \sum_{i=1}^{N} \rho^{(i)}/N$, and thus $J_1(\rho, v) = J_2(\rho^{(i)}, v^{(i)})$. □

In conclusion, the two systems are equivalent in the sense that for any feasible solution for one system there exists an equivalent solution that has the same swarm density evolution and realizes the same cost. As a corollary of the above result, the two systems share the same optimal cost.

We can solve the optimal control problem using PDE-constrained optimization introduced in chapter 2 [20].

Introducing the adjoint state $\nu^{(i)}$, we formulate the Lagrangian of the system as

$$\begin{aligned} &L(\rho^{(i)}, \nu^{(i)}, v^{(i)}) = \\ &J(\rho^{(i)}, v^{(i)}) - \sum_{i=1}^{N} \int_0^T \int_\Omega \Big(\frac{\partial \rho^{(i)}}{\partial t} + \nabla(-\mu \nabla \rho^{(i)} + v^{(i)} \rho^{(i)}) \Big) \nu^{(i)}. \end{aligned} \tag{3.4}$$

The following equation is crucial for calculating the variations.

$$\begin{aligned} &\int_0^T \int_\Omega [\frac{\partial \rho^{(i)}}{\partial t} + \nabla(-\mu \nabla \rho^{(i)} + v^{(i)} \rho^{(i)}] \nu^{(i)} d\Omega dt = \\ &\qquad \int_\Omega \rho^{(i)}(x,T) \nu^{(i)}(x,T) - \int_\Omega \rho^{(i)}(x,0) \nu^{(i)}(x,0) - \int_0^T \int_\Omega \rho^{(i)} \frac{\partial \nu^{(i)}}{\partial t} d\Omega dt \\ &+ \int_0^T \int_\Gamma (\mu \rho^{(i)} \nabla \nu^{(i)}) \cdot n d\Gamma dt - \int_0^T \int_\Omega \mu \rho^{(i)} \Delta \nu^{(i)} d\Omega dt - \int_0^T \int_\Omega \rho^{(i)} v^{(i)} \cdot \nabla \nu^{(i)} d\Omega dt \end{aligned} \tag{3.5}$$

With the help of (3.5), we calculate the variations of (3.4) with respect to the state $\rho^{(i)}$ to derive the adjoint system.

$$\begin{aligned} &L'_{\rho^{(i)}}[\delta \rho^{(i)}] = \\ &\int_\Omega \delta \rho^{(i)}(x,T) \left(\frac{\sum_{i=1}^{N} \rho^{(i)}(x,T)}{N} - \rho_T(x) - \nu^{(i)}(x,T) \right) \\ &+ \int_0^T \int_\Omega \delta \rho^{(i)} (\mu \Delta \nu^{(i)} + \frac{\partial \nu^{(i)}}{\partial t} + v^{(i)} \cdot \nabla \nu^{(i)}) \\ &- \int_0^T \int_\Gamma \delta \rho^{(i)} (\mu \nabla \nu^{(i)} \cdot n). \end{aligned}$$

The adjoint system is obtained by setting the variation with respect to $\rho^{(i)}$ to 0 and is given by the backward equation

$$\begin{cases} \dfrac{\partial \nu^{(i)}}{\partial t} + \mu \Delta \nu^{(i)} + v^{(i)} \cdot \nabla \nu^{(i)} = 0 & \text{in } \Omega, \\ \nabla \nu^{(i)} \cdot n = 0 & \text{on } \Gamma, \\ \nu^{(i)}(x,T) = \dfrac{\sum_{j=1}^{N} \rho^{(j)}(x,T)}{N} - \rho_T(x)\,. \end{cases} \tag{3.6}$$

Similarly, we calculate the variations of (3.4) with respect to the control input $v^{(i)}$ to derive first order optimality condition and the reduced gradient.

$$L'_{v^{(i)}}[\delta v^{(i)}] = \\ \alpha \int_0^T \int_\Omega v \cdot \delta v^{(i)} d\Omega dt + \int_0^T \int_\Omega (\rho^{(i)} \nabla \nu^{(i)}) \cdot \delta v^{(i)} d\Omega dt.$$

The first-order optimality condition is obtained by setting the variation with respect to the control variable $v^{(i)}$ to 0. In this way, we also obtain the gradient of the reduced cost function $\tilde{J}(v^{(i)}) = J(\rho^{(i)}(v^{(i)}), v^{(i)})$, which reads as:

$$\nabla \tilde{J}(v^{(i)}) = \alpha v^{(i)} + \rho^{(i)} \nabla \nu^{(i)}. \tag{3.7}$$

The reduced gradient is used to perform gradient descent in search of the optimal control.

Chapter 4

Decentralized System

We can directly decentralize the decoupled system; however, a more interesting question arises when agents have only limited local information.

For this reason, we suppose that the agents can only sense the target density and the presence of other agents within a limited circular area $D_i(t)$, depicted for each agent as a dashed circle in Fig. 4.1. Each disk $D_i(t)$ is centered at the agent i's position, and thus changes over time. With these premises, each agent must solve the optimal control problem:

$$\min_{\rho^{(i)},v^{(i)}} \quad J(\rho^{(i)},v^{(i)}) = \frac{1}{2}\int_\Omega \left(\frac{\rho^{(i)}(x,T) + \sum_{j\neq i}\rho^{(j)}(x,T)\chi_{D_i(T)}}{N} - \rho_T\chi_{D_i(T)} \right)^2 + \frac{\alpha}{2}\sum_{i=1}^{N}\int_0^T\int_\Omega \|v^{(i)}(x,t)\|^2 \quad (4.1)$$

subject to (3.1), where both terms of the cost functional depends only on the target density and the densities of other agents localized in the sensing area $D_i(T)$.

It is important to note that, due to the limited information of each agent, it can be inaccurate if not impossible for an agent to predict the final-time swarm density, and the appearance or departure of other agents within its sensing area can cause significant oscillations. Therefore, we let each agent solve their optimal control assuming the other agents are static, that is, we assume $D_i(t) = D_i(0) = D_i$, $\rho^{(j)}(t) \equiv \rho^{(j)}(0)$, $t \in [0,T]$.

Such approximations may be accurate only assuming that the final time T is sufficiently small. For this reason, we adopt a MPC paradigm as in [8]. In this way, each agent solves

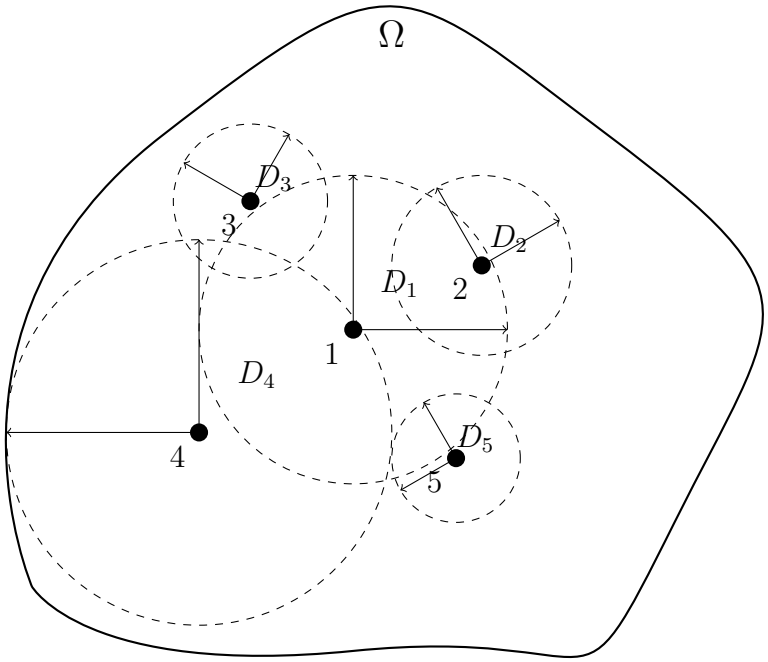


Figure 4.1: Depiction of five agents (numbered black dots) with their local sensing areas D_i at the moment t (dashed circle). The arrows originating from each agent represent the axes of their own local references.

the optimal control using similar reduced gradient (3.7) over a very small time horizon, and then iterates this process until convergence to the target density. The algorithm is described in 1.

Algorithm 1 MPC for agent i solving (4.1)

1: Assign the agents with Gaussians as their densities.
2: Sense the target density ρ_T and other agents' densities ρ_0^{others} inside sensing area.
3: Initial guess on velocity field $v^{(i)}$.
4: Solve the Fokker-planck equation with v.
5: Solve the adjoint equations (3.6).
6: Calculate the reduced gradient (3.7) and conduct gradient descent on v.
7: Move the agent according to the velocity field and random noise.
8: Repeat step 4-7 until desired accuracy or final time is reached.

Another significant difference between (4.1) and (3.2) is the restriction of the spatial domain in which system dynamics are considered. This is achieved by means of the indicator functions $\chi_{D_i(T)}$ in the cost of (4.1). This results in a significantly lower computational cost, as discussed quantitatively in the next section. On the other hand, since the domain for solving the problem differs from that in the centralized problem, the boundary conditions also need to be changed. In the centralized problem (2.3), reflecting Neumann boundary conditions are applied because agents cannot leave the domain Ω. However, when the solution is restricted to a smaller neighborhood, agents can leave the domain D_i. Considering that if an agent leaves the region under the effect of the optimal control, it should not return within a short period, an absorbing Dirichlet condition [10] is more appropriate. Additionally, if the boundary Γ_i of agent i's neighborhood D_i intersects with the boundary Γ of Ω, a mixed boundary condition is used. With these approximations,

the decentralized control problem (4.1) becomes:

$$
\begin{aligned}
&\min_{\rho^{(i)},v^{(i)}} \quad J(\rho^{(i)},v^{(i)}) = \\
&\frac{1}{2}\int_{D_i}\left(\frac{\rho^{(i)}(x,T)+\sum_{j\neq i}\rho^{(j)}(x,0)\chi_{D_i}}{N}-\rho_T(x)\chi_{D_i}\right)^2 \\
&+\frac{\alpha}{2}\sum_{i=1}^{N}\int_0^T\int_{D_i}\|v^{(i)}(x,t)\|^2 \qquad (4.2)\\
&\text{s.t.} \quad \begin{cases} \frac{\partial\rho^{(i)}}{\partial t}+\nabla\cdot(-\mu\nabla\rho^{(i)}+v^{(i)}\rho^{(i)})=0 & \text{in } D_i,\\ \rho^{(i)}=0 & \text{on } \Gamma_i\backslash\Gamma,\\ (-\mu\nabla\rho^{(i)}+v^{(i)}\rho^{(i)})\cdot n=0 & \text{on } \Gamma_i\bigcap\Gamma,\\ \rho^{(i)}(x,0)=\rho_0^{(i)}(x) & \text{in } D_i. \end{cases}
\end{aligned}
$$

The differences between (4.2) and (4.1) are threefold. First, in the cost function, the target density and the densities of other agents are replaced by their initial time values instead of the final time values. Second, the solution domain of the system is restricted to a smaller neighborhood D_i. Third, Dirichlet boundary conditions are applied.

Chapter 5

Numerical Experiments

We conducted several numerical experiments to show the performance of the proposed controllers and to illustrate their properties. Section 5.2 shows that decoupled centralized control can effectively achieve the target density, thereby validating the proposition. In Section 5.3 we report the results of the experiments performed by varying two important parameters in decentralized control, namely the sensing area D_i and the time step T. Section 5.4 introduces a metric of energy consumption in order to quantitatively compare the proposed controllers.

5.1 Numerical Method and Parameters

We consider 1-d problems on the interval $[0, 1]$, using the Finite Element Method (FEM) for space discretization and Crank-Nicolson scheme for time discretization. We use IPOPT [22] for the NLP optimization. For the control input velocity fields we use the same discretization as for the state.

We adopt the optimize-then-discretize approach, that is we first derive the optimality conditions in the continuous setting and then substitute the discretized variables into the optimality conditions. Specifically, if we have a discretized control input $v^{(i)}(x,t) = \sum_{j=1}^{M} v_j^{(i)}(t)\phi_j$, the corresponding reduced gradient can be expressed as:

$$\frac{\partial \hat{J}(v)}{\partial v_j^{(i)}} = \int_{D_i} (\rho^{(i)}\phi_j \cdot \nabla \nu^{(i)} + \alpha \phi_j \cdot v^{(i)}) d\Omega. \tag{5.1}$$

To derive the fully discretized systems, we need the help of the following matrices.

$$
\begin{aligned}
(M)_{ij} &= \int_\Omega \phi_i(x)\phi_j(x)d\Omega \\
(A)_{ij} &= \int_\Omega \mu\phi_i'(x)\phi_j'(x)d\Omega \\
(B_{1,k})_{ij} &= \int_\Omega \phi_i(x)\phi_j'(x)\phi_k(x)d\Omega \\
(B_{2,k})_{ij} &= \int_\Omega \phi_i(x)\phi_j(x)\phi_k'(x)d\Omega \\
(C_k)_{ij} &= \delta_{iM}\delta_{jM}\delta_{kM} - \delta_{i1}\delta_{j1}\delta_{k1}, \quad \delta_{ij} \text{ is the Kronecker delta.}
\end{aligned}
$$

The main dynamics of the agents' states and adjoint states are then given by:

$$
\begin{aligned}
M\frac{d\rho^{(i)}}{dt} + A\rho^{(i)} + \sum_{k=1}^{M} v_k^{(i)}(t)(B_{1,k} + B_{2,k} - C_k)\rho^{(i)} = 0 \\
M\frac{d\nu^{(i)}}{dt} - A\nu^{(i)} + \sum_{k=1}^{M} v_k^{(i)}(t)B_{1,k}\nu^{(i)} = 0.
\end{aligned}
$$

And the fully discretized version of the reduced gradient (5.1) is given as follows:

$$
\frac{\partial \hat{J}(v)}{v_j^{(i)}} = \rho^{(i)}B_{1,j}\nu^{(i)} + N\alpha(Mv^{(i)})_j
$$

In all experiments, we set the number of agents $N = 5$, the diffusion coefficient $\mu = 0.01$, and the regularization coefficient $\alpha = 0.01$. Additionally, a velocity constraint of $\|v(x,t)\|_\infty \leq 0.05$ is imposed to better simulate real-world scenarios. We change the number of FEM nodes according to the sensing radius such that the distance between each node stays constant 0.02.The standard deviation of the Gaussians assigned to the agents is 0.05.

5.2 Decoupled Centralized System

We first demonstrate that the decoupled system (3.2) successfully achieves the target density, as illustrated in Fig. 5.1. The orange line shows the target density ρ_T as a normal distribution centered at 0.5 with standard deviation being 0.1. The blue line represents the swarm density starting from the sum of five Gaussians assigned to the five agents. As time evolves, the swarm density converges to the target density.

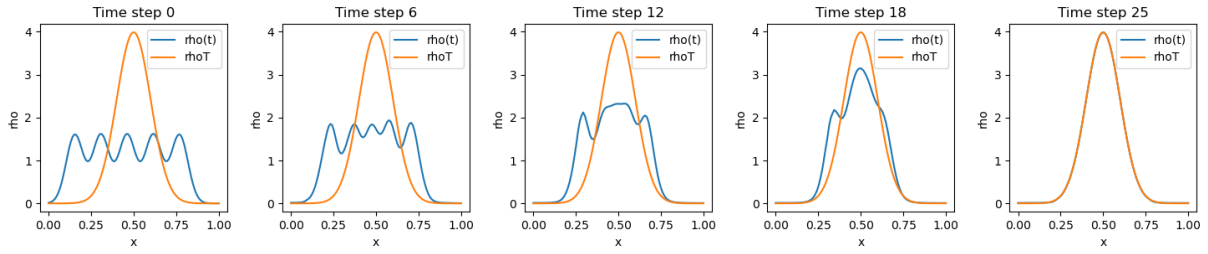


Figure 5.1: Evolution of swarm density according to the solution of the decoupled centralized problem (3.2). The density of five agents (blue line) evolves over time (with a time step of 0.1s) from an initial sum of Gaussians to the target density (orange line).

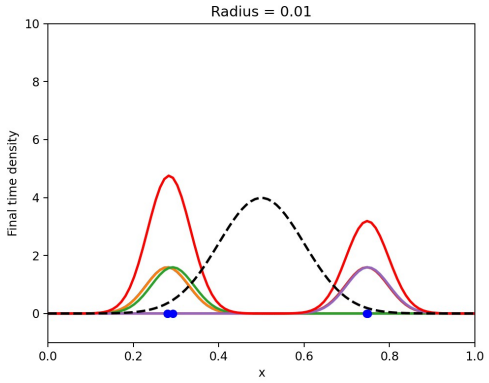
5.3 Decentralized System

The decentralized problem (4.1) are reported for varying sensing radii and time step. The adjustments to the sensing area are accompanied by corresponding changes in the number of nodes used to approximate the velocity fields. The results are shown in Fig. 5.2.

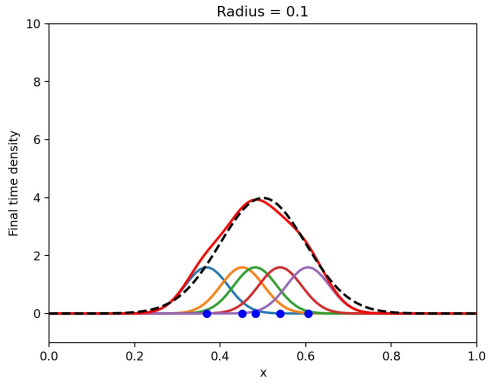
The blue dots on the x-axis are the agents, with the shorter Gaussians centered at them being the assigned densities. The dashed line represents the target density, a Gaussian centered at the 0.5. The red lines represent the swarm densities. It is expected that a larger sensing area would end up with a better result. Indeed, when the sensing area decreases from 0.1 to 0.01, agents in the central region even move away from the target toward the endpoints. While a sensing area of 0.5 slightly overshoots the target value within a given time compared to a sensing area of 0.1, its dynamics are smoother, showing the trade-offs between the two configurations.

As discussed in Section 4, the length of each MPC step could be critical to the result. To demonstrate its effects, we also change the time step and report the results of the simulations in Fig. 5.3, which shows the final time densities. We observe that the shorter the time horizon, the closer the result fits the target density. Particularly, when the time horizon is extended to 1, agents once again diverge from the target. However, when the time horizon is shorter, the differences are not very pronounced. Considering that a shorter time horizon within a given final time requires more steps, and consequently more computation, it becomes crucial to select an appropriate time horizon that ensures a good result without heavy computational cost.

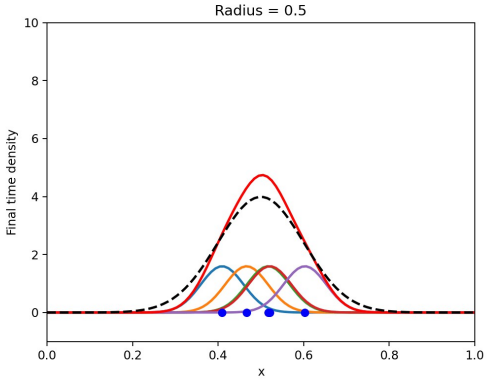
It is interesting to note how Figures 5.2(a) and 5.3(d) demonstrate similar lack of convergence. Such behavior reveals a certain connection between time step and sensing



(a) Radius=0.01

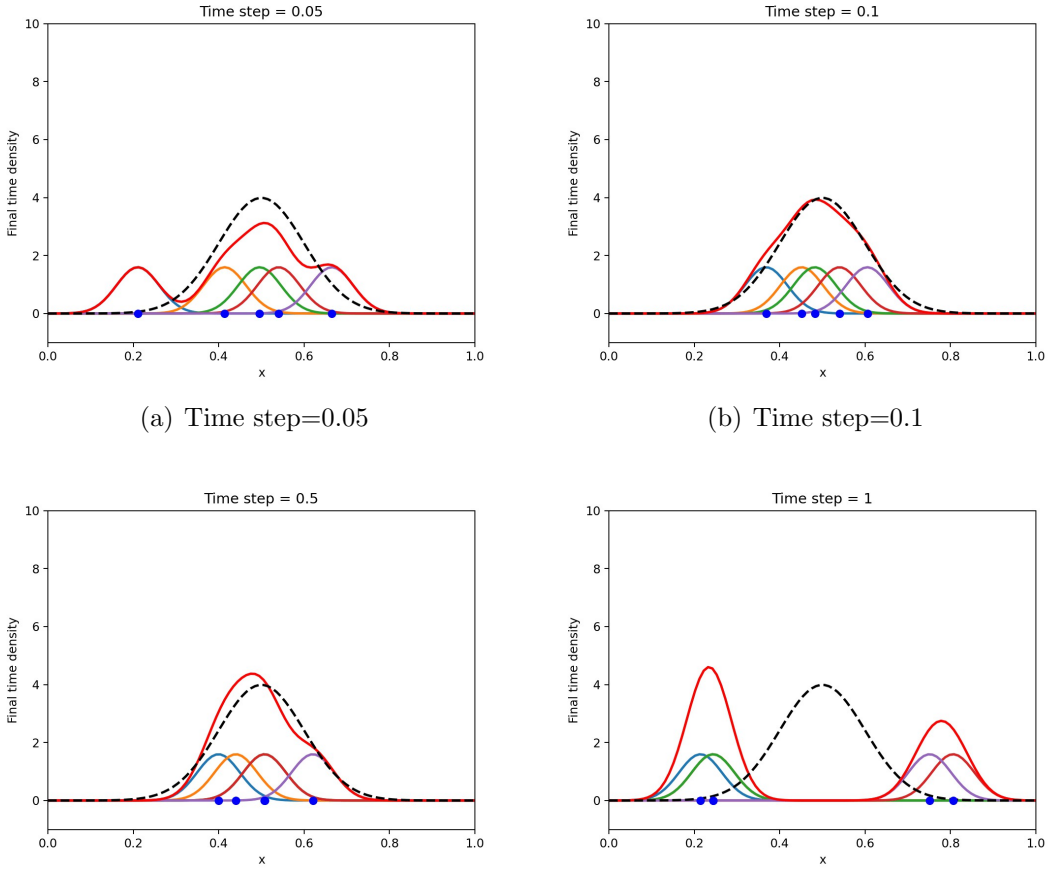


(b) Radius=0.1



(c) Radius=0.5

Figure 5.2: Final density achieved by the agents under the control of (4.2), for varying sensing radii. The blue dots represent the five agents, while their density distributions are depicted as colored lines. The red line represents the swarm density (sum of the agents' distributions), and the dashed line represents the target density.



(a) Time step=0.05

(b) Time step=0.1

(c) Time step=0.5

(d) Time step=1

Figure 5.3: Final density achieved by the agents under the control of (4.2), for varying time steps. The blue dots represent the five agents, while their density distributions are depicted as colored lines. The red line represents the swarm density (sum of the agents' distributions), and the dashed line represents the target density.

Table 5.1: Performance comparison

Parameter	Iteration	Stopping Time	Commun	Sensing	Computation
Centralized&Coupled	45.3	4.53	5.45	45.3	2061.10
Sensing 0.1 Time 0.1	48.3	4.83	0	2.415	84.70
Sensing 0.5 Time 0.1	42.7	4.27	0	53.375	2576.15
Sensing 0.1 Time 0.05	95.5	4.78	0	4.775	201.95
Sensing 0.1 Time 0.5	16.9	8.45	0	0.845	85.30

radius, though the specific nature of the relationship requires further investigation in future research.

5.4 Performance Comparison

To compare the performance of the three approaches introduced in this paper, we define a metric for energy consumption which accounts for computation, sensing, and communication. We set a common initial density, target density, and an error threshold of 0.05 for the L_2 norm of the difference between the swarm density and the target density. The MPC scheme stops when this error threshold is reached. At the end, we record the error, stopping time, and computation time. For the energy consumption related to sensing and communication, crucial in practical applications, we assume a quadratic dependence on the sensing area and communication distance, respectively—this reflects the typical decay in signal strength and sensing accuracy with increasing distance, consistently with the inverse square laws observed in physical systems. Finally, since for the decentralized problem (4.1) uses MPC to control the positions of agents, for fairness, we similarly applied MPC to the centralized problem. Furthermore, considering that random noise is introduced in the simulation, each experiment is repeated 5 times, and the average is taken.

Table 5.1 reports the results of the comparisons. Here, *Iteration* represents the number of MPC steps executed, and *Stopping time* corresponds to the product of iteration and time step. In the centralized problem, sensing is only performed by a central unit, and thus the Sensing cost each iteration is uniformly 1. In contrast, in the decentralized problem (4.1), the sensing cost each iteration is the square of the sensing radius multiplied by the number of agents. For the centralized problem, we define the communication cost each iteration as $\min_i \sum_{j=1}^{N}(X_i - X_j)^2$, whereas there is no communication cost in the decentralized problem (4.1). For the centralized problem, the computation cost is evaluated as the

average computational time of the machine[1]. For the decentralized problem, since each agent can independently solve for its optimal control using algorithm 1, we employed parallel programming for the simulations. Thus, the total computation cost is evaluated as the product of the average computation time and the number of agents.

As shown Table 5.1, the stopping times of the centralized and decentralized approaches are quite similar, being nearly identical across experiments, except for the decentralized group with a time step of 0.5. Regarding energy consumption, apart from the group with a sensing area of 0.5, the centralized approach consistently exhibits higher costs compared to the decentralized approach. We can then conclude that, with an appropriate choice of parameters, the decentralized approach has similar performance compared to the centralized one in terms of stopping time, outperforming it in terms of energy consumption.

The impact of time step and sensing radius can also be observed within the decentralized experiment groups. Similar to the trends shown in Fig. 5.3, when the time step is set to 0.05 and 0.1, the stopping times are nearly the same. However, when further increased to 0.5, the stopping time rises significantly. This indicates that stopping time remains comparable when the time step is within a certain range, but further increases might lead to an increase in stopping time. For the sensing area, a value of 0.5 results in a slight reduction in stopping time compared to 0.1, but the difference is small. Thus, while increasing sensing radius can decrease stopping time, its effect is negligible. On the other hand, from the cost perspective, increasing the sensing area significantly raises both the sensing cost and computation cost. While increasing the time step does prolong stopping time, it also decreases the number of iterations, thereby significantly lowering the sensing cost.

[1]All computations were conducted on a Mechrevo 16S laptop with an AMD Core Ryzen 7 7840H CPU and 16 GB RAM running Windows 11.

Chapter 6

Conclusion

As a conclusion, we proposed a decentralized control framework for swarm density control through a combination of PDE-constrained optimization and MPC. We achieve decentralization by restricting the sensing area of what we called the *decoupled centralized problem*. The decentralized control method demonstrates the ability to accurately reach desired target densities with lower computational, sensing, and communication cost, compared to the centralized controller. Extensive numerical experiments illustrate the effects of sensing area and optimization time horizon on system performance.

Several promising directions remain for future exploration. First, we aim to conduct more complicated simulations involving larger number of agents and multimodel target densities to better evaluate the algorithm's robustness and scalability. Second, the algorithm can be implemented on real robots to validate its practical feasibility and performance in real-world scenarios. Third, designing a specific form of communication between agents might improve the performance as it enable information sharing between agents. Lastly, providing theoretical guarantees, especially the sensitivity with respect to the target density. With such sensitivity, the time approximation we made would be guaranteed to have a certain level of accuracy.

References

[1] Andrew J Archer and Markus Rauscher. Dynamical density functional theory for interacting brownian particles: stochastic or deterministic? *Journal of Physics A: Mathematical and General*, 37(40):9325, 2004.

[2] Lubomir Bakule. Decentralized control: An overview. *Annual reviews in control*, 32(1):87–98, 2008.

[3] Shiba Biswal, Karthik Elamvazhuthi, and Spring Berman. Decentralized control of multiagent systems using local density feedback. *IEEE Transactions on Automatic Control*, 67(8):3920–3932, 2021.

[4] Tim Breitenbach and Alfio Borzì. The pontryagin maximum principle for solving fokker–planck optimal control problems. *Computational Optimization and Applications*, 76(2):499–533, 2020.

[5] Nazlı Demir, Utku Eren, and Behçet Açıkmeşe. Decentralized probabilistic density control of autonomous swarms with safety constraints. *Autonomous Robots*, 39:537–554, 2015.

[6] Cristina Dimidov, Giuseppe Oriolo, and Vito Trianni. Random walks in swarm robotics: an experiment with kilobots. In *International conference on swarm intelligence*, pages 185–196. Springer, 2016.

[7] Karthik Elamvazhuthi and Spring Berman. Optimal control of stochastic coverage strategies for robotic swarms. In *2015 ieee international conference on robotics and automation (icra)*, pages 1822–1829. IEEE, 2015.

[8] Arthur Fleig, Lars Grüne, and Roberto Guglielmi. Some results on model predictive control for the fokker-planck equation. In *MTNS 2014*, pages 1203–1206, 2014.

[9] Arthur Fleig and Roberto Guglielmi. Optimal control of the fokker–planck equation with space-dependent controls. *Journal of Optimization Theory and Applications*, 174:408–427, 2017.

[10] Crispin Gardiner. *The Fokker-Planck Equation*, pages 113–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[11] Roland Herzog and Karl Kunisch. Algorithms for pde-constrained optimization. *GAMM-Mitteilungen*, 33(2):163–176, 2010.

[12] Hanjun Li, Chunhan Feng, Henry Ehrhard, Yijun Shen, Bernardo Cobos, Fangbo Zhang, Karthik Elamvazhuthi, Spring Berman, Matt Haberland, and Andrea L Bertozzi. Decentralized stochastic control of robotic swarm density: Theory, simulation, and experiment. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4341–4347. IEEE, 2017.

[13] Siddharth Mayya, Pietro Pierpaoli, and Magnus Egerstedt. Voluntary retreat for decentralized interference reduction in robot swarms. In *2019 international conference on robotics and automation (ICRA)*, pages 9667–9673. IEEE, 2019.

[14] Gennaro Notomista, Siddharth Mayya, Anirban Mazumdar, Seth Hutchinson, and Magnus Egerstedt. A study of a class of vibration-driven robots: Modeling, analysis, control and design of the brushbot. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5101–5106. IEEE, 2019.

[15] Motoya Ohnishi, Li Wang, Gennaro Notomista, and Magnus Egerstedt. Barrier-certified adaptive reinforcement learning with applications to brushbot navigation. *IEEE Transactions on robotics*, 35(5):1186–1205, 2019.

[16] R-E Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, 2006.

[17] Daniele Proverbio, Luca Gallo, Barbara Passalacqua, Marco Destefanis, Marco Maggiora, and Jacopo Pellegrino. Assessing the robustness of decentralized gathering: a multi-agent approach on micro-biological systems. *Swarm Intelligence*, 14(4):313–331, 2020.

[18] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE international conference on robotics and automation*, pages 3293–3298. IEEE, 2012.

[19] Melanie Schranz, Martina Umlauft, Micha Sende, and Wilfried Elmenreich. Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, 7:36, 2020.

[20] Carlo Sinigaglia, Andrea Manzoni, and Francesco Braghin. Density control of large-scale particles swarm through pde-constrained optimization. *IEEE Transactions on Robotics*, 38(6):3530–3549, 2022.

[21] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, Mass., 2005.

[22] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.

[23] Zhiyu Zhang, Wenchong Tian, and Zhenliang Liao. Towards coordinated and robust real-time control: a decentralized approach for combined sewer overflow and urban flooding reduction based on multi-agent reinforcement learning. *Water Research*, 229:119498, 2023.