UNIVERSITY OF OTTAWA

OTTAWA-CARLETON INSTITUTE FOR MECHANICAL AND AEROSPACE ENGINEERING

Formation Control of Multi-agent Systems via Voronoi Tessellation and Kullback-Leiber Divergence

Author:
Ruiming Zheng

Supervisor:
Davide Spinello

A thesis submitted in partial fulfillment of the requirements for the Master of Applied Science degree in

Mechanical Engineering

Department of Mechanical Engineering
Faculty of Engineering
University of Ottawa

Formation Control of Multi-agent Systems via Voronoi Tessellation and Kullback-Leiber Divergence

Ruiming Zheng

Abstract

Recent advancements in communication technology, computational power, and control theory have led to the adoption of multi-agent systems in various engineering applications. By utilizing multiple agents, these systems can accomplish complex cooperative tasks that are difficult for a single agent to achieve, such as coordinated patrolling, surveillance, post-disaster search and rescue, and transportation logistics. Moreover, multi-agent systems have been employed to model and analyze flocking behaviour in both social and natural phenomena, including pedestrian flow, animal migration and hunting.

This thesis presents an algorithm to control the spatial distribution of kinematic multi-agent systems in two-dimensional workspaces. Leveraging on the coverage control framework, the problem is formulated as a multi-objective optimization with performance index composed of the area coverage metric and of the Kullback–Leibler (KL) divergence. The KL term drives the statistical spatial distribution of the agents to a desired, user-defined density in the workspace, whereas the coverage term drives the agents to a centroidal Voronoi configuration. The connection is the target distribution in the KL term, which is also the risk density in the area coverage term.

Since the system is non-autonomous due to the drift introduced by the evolving target density, the asymptotic stability is established by using Barbalat's lemma. The agents are proven to asymptotically converge to the trajectories of the time-varying stationary points of the multi-objective performance index, which monotonically minimizes the index, and drives the agents to evolve in a special centroidal Voronoi configuration of the same statistics as the target density. The collision avoidance of the agents between each other is guaranteed based on the inherent properties of the centroidal Voronoi tessellation. Furthermore, by designing the target density via its elliptical contour, the converged KL term drives the agents in a specified range, which allows the agents to accurately pass through confined environments, such as tunnels or pipelines. Theoretical predictions are illustrated in simulations.

Dedication

I would like to dedicate my thesis to my parents, Quanshan Zheng and Meiling Wang, for their unwavering love and support, and to my girlfriend, Yue Wan, for her constant encouragement and belief in me. Their strength and patience have been my pillars during this journey.

Acknowledgments

I would like to express my deepest gratitude to Dr. Davide Spinello, for his invaluable guidance, support, and encouragement throughout the course of my research. His expertise and insightful feedback have been instrumental in the completion of this thesis. I am truly grateful for the knowledge and opportunities he has provided, which have greatly enhanced my academic and research skills.

Contents

List of Figures

Nomenclature

This section defines the key terms and symbols that are frequently used in the main body of this thesis, particularly in Chapter 3 and Chapter 4. For each term, the formal definition and relevant details are provided in the respective sections where they are introduced and discussed.

$\mathbf{p}_i$	Position of i-th agent
$\mathbf{u}_i$	Velocity control input to i-th agent
$\mathbf{\Omega}$	Convex working space
$f(r_i)$	Sensing performance function of the i-th agent
$\mathbf{q}$	A point in the working space
ϕ_d	Target density (or desired density) weighting the workspace
$\boldsymbol{\mu}_d$	Mean vector of the target density ϕ_d
r_i	Euclidean distance between the i-th agent and any point $\mathbf{q} \in \mathbf{\Omega}$
$\mathbf{\Sigma}_d$	Covariance matrix of the target density ϕ_d
ϕ_P	Collective density of agents, determined by their spatial distribution; also referred to as agents' density or collective density for simplicity
$\boldsymbol{\mu}_P$	Mean vector of the agents' collective density ϕ_d
$\mathbf{\Sigma}_P$	Covariance matrix of the agents' collective density ϕ_d
H	Area coverage metric encoding sensing performance function $f(\cdot)$ and target density ϕ_d
D	KL divergence between the distribution of target density ϕ_d and agents' collective density ϕ_P

J	Total multi-objective performance index composed of weighted area coverage metric H and KL divergence D
α	Weighting coefficient for area coverage metric H
β	Weighting coefficient for KL divergence D
$\boldsymbol{\mathcal{V}}_i$	Voronoi cell corresponding to i-th agent
m_i^H	Generalized mass for i-th Voronoi cell
$\boldsymbol{c}_i^H$	Stationary point of area coverage metric H; also serves as generalized centroid of mass for i-th Voronoi cell
$\boldsymbol{c}_i^D$	Stationary point of KL divergence term D corresponding to i-th agent $\boldsymbol{p}_i$
$\tilde{\boldsymbol{c}}_i$	Stationary point of multi-objective performance index J corresponding to i-th agent $\boldsymbol{p}_i$
$\boldsymbol{W}_i$	Weight matrix for gradient $\partial J / \partial \mathbf{p}_i$
det	Determinant operator
tr	Trace operator
$\otimes$	Kronecker product

Chapter 1

Introduction

Portions of the content presented in this thesis are adapted from the research paper [1], on formation control for multi-agent systems via Voronoi tessellation and KL divergence with time-invariant Gaussian density functions, particularly in the sections regarding autonomous controls for time-invariant target densities (non evolving-environments). The paper has been accepted for publication in the 2024 IEEE Conference on Decision ad Control (CDC) that will be held in December 2024. This thesis includes advancements and extensions that build upon my work in [1], with the derivation of a control algorithm to navigate time-varying environments. The extended work is being prepared to be published in a technical journal.

The figures reused in this thesis comply with uOttawa's policy on the use of copyrighted material. Permissions have been obtained from the original authors or publishers when necessary, and all reused figures are properly cited in accordance with both university guidelines and copyright regulations.

1.1 Motivation

In recent years, multi-agent systems have emerged in various engineering applications, driven by advancements in communication technology, computational technology, and control theory. The deployment of multiple agents allows them to execute complex cooperative missions that are challenging for a single agent, such as harbour protection [2], coordinated patrolling and surveillance [3, 4], post-disaster search and rescue [5, 6], and transportation and logistics. Additionally, multi-agent systems have been employed to model and analyze collective behaviours observed in social and natural phenomena, such as pedestrian flow [7, 8, 9], animal migra-

tion and hunting [10, 11].

Area coverage provides an optimal control framework for multi-agent systems, where a finite set of mobile agents responds to a field defined over a given environment, aiming to achieve an optimal spatial distribution based on a coverage metric. This field, often referred to as risk density, quantifies properties such as relative importance or risk levels associated with different regions. The problem can be viewed as a resource allocation task, where areas with higher risks demand greater agent presence. For instance, in harbour protection scenarios, evolving risk densities can represent suspicious targets, allowing agents to track and monitor them effectively.

Building on previous works, this thesis proposes an optimal control algorithm to drive the formation of a group of kinematic agents via a user-defined distribution defined on the environment. The distribution models environmental signals and queues to which the agents are assumed to respond, organizing their formation according to optimality criteria dictated by a performance index that encodes the coverage of the area in which they operate, with clustering in regions with higher density. The converged multi-objective performance index drives the agents passing through specific regions in a specific spatial configuration (centroidal Voronoi tessellation), as long as the target density contour is designed accordingly. This capability can support multi-agent missions in confined environments, such as search-and-rescue operations in caves or tunnel detection missions. With the proper modifications, this can also be applied to biological and non-biological systems, as well as hybrid ones, using the idea of driving an inherently complex collective response via a low dimensional set of parameters, corresponding to the moments of a target distribution. The theoretical predictions are illustrated and verified in several sets of simulations.

1.2 Thesis Objectives and Contributions

The thesis focuses on the formation control for groups of mobile robots moving in an environment in which they are driven by stimuli described by distributed functions in the environment. The problem is approached by generalizing the standard area coverage control problem, by posing the control problem as a multi-objective optimization, with performance composed of the Kullback–Leibler (KL) divergence and of the coverage metric, respectively encoding the residual between the statistical distribution of the agents in the workspace and a target distribution,

and the weighted quantification of the performance of each agent with respect to points in the workspace, which induces a partition of the environment into regions that are "best served" by the agents to which they are assigned, resulting into the emergence of a Voronoi tessellation of the workspace. Simultaneously, the contribution of the KL divergence forces the statistical moments of the converged generalized centroidal formations to be as close as possible to the ones of the target density, where the proximity is dictated by the weights of the multi-objective function. By forcing the proximity of the moments of the agents' spatial distribution to the ones of a desired distribution via the KL divergence, the formation control is abstracted into a smaller dimensional space with respect to the one of the agents. Formation control of multi-robot systems via abstraction of their spatial statistical distribution has been proposed in [12] in the context of geometric formation control.

The contributions of this work are as follows:

- **Multi-objective Formulation** The problem is formulated as a multi-objective optimization, with the introduction of a novel performance index that directly uses the KL divergence to track the statistical moments of a desired spatial distribution in the workspace.
- **Stability Analysis** For the autonomous case, the largest invariant set is identified. For the non-autonomous case, the asymptotic trajectories of the agents are established using Barbalat's lemma.

1.3 Thesis Outline

The thesis is organized as follows:

In Chapter 2, a literature review pertinent to the work of the thesis is conducted. Background material is also briefly introduced when appropriate.

In Chapter 3, a preliminary of the area coverage control problem is presented in mathematical notations, including time-invariant and time-varying cases.

In Chapter 4, the multi-objective control problem is formulated with an analysis of the critical points, where the performance index is a convex combination of the coverage metric and the KL divergence between the agents' collective density and a desired one. In addition, the non-autonomous feedback law is proposed and the asymptotic trajectories are established with Barbalat's lemma. This implies the agents converge to the trajectories of their associated critical points under the

proposed control, and the formation of agents asymptotically minimizes the time-varying multi-objective performance index.

In Chapter 5, theoretical predictions are demonstrated and validated through several sets of simulations, including cases with time-invariant and time-varying densities. The control efforts for each sub-index are illustrated, and a design scenario involving an L-shaped tunnel is presented.

In Chapter 6, a summary of the work in the thesis is provided.

Chapter 2

Literature Review

This chapter provides an introduction to coordination control for networked multi-agent systems (Section 2.1), and an overview of the majority of techniques involved in this work, including Voronoi diagrams (Section 2.2), area coverage control for multi-agent systems (Section 2.3), and Kullback–Leibler divergence (Section 2.4.2).

2.1 Coordination Control for Networked Multi-agent Systems

Coordination control of networked systems of multiple kinematic agents has garnered extensive attention due to its potential application in various engineering fields. In the real world, many phenomena and applications can be simulated and studied using multi-agent systems, such as harbour protection [2], coordinated surveillance [3, 4], and search and rescue missions [5, 6], among others.

A networked multi-agent system typically refers to a collection of robotic devices that are interconnected through a network, allowing them to communicate, share information, coordinate their actions and execute complex cooperative missions that are challenging for a single agent [13, 14], as shown in Fig. 2.1. These agents can take various forms, including unmanned aerial vehicles (UAVs) [15, 16], autonomous underwater vehicles (AUVs) [17], autonomous ground vehicles (AGVs) [18], and other networked robotic systems or a combination thereof [19].

The communication network among the agents enables them to perform more complex tasks, and also brings new challenges, such as communication delay, topology changes, packet loss, noise and interference. The communication topology of networked systems is described by graph theory [20]. In this thesis, the

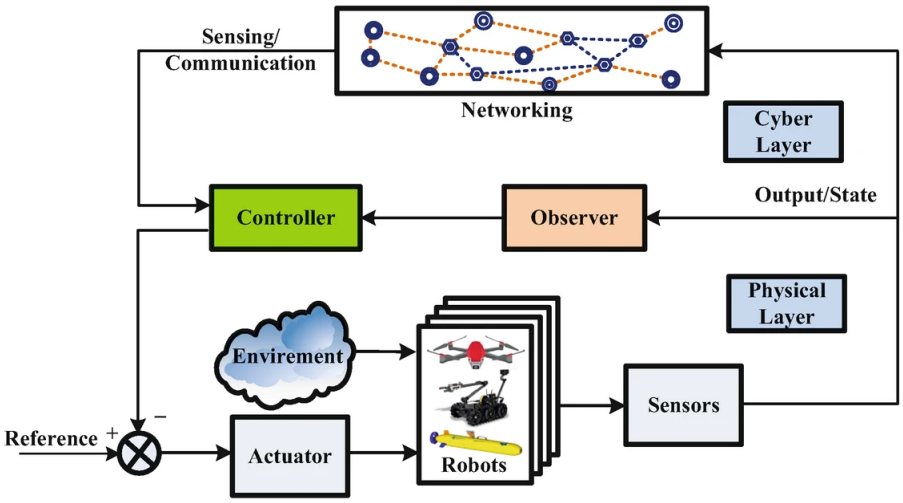


Figure 2.1: Structure and features of the networked robotic systems. [13]

communication network is assumed to be connected and instantaneous to focus on the formulation of the multi-objective formation control problem.

2.2 Voronoi Diagrams

2.2.1 Introduction

The concept of the Voronoi diagram dates back to the 17th century [21], when R. Descartes claimed that the solar system consists of vortices [22]. He believed that the fixed stars decomposed space into convex regions, each containing matter attracted to and revolving around them, as illustrated in Fig. 2.2. Although Descartes has not explicitly defined the boundaries of these regions, the underlying idea is consistent with the latter Voronoi diagram: let a space $\boldsymbol{M}$, and a set $\boldsymbol{P}$ of sites (fixed stars) $\boldsymbol{p}$. Each site $\boldsymbol{p}$ experts influence on the points $\boldsymbol{q}$ (matter) in the space $\boldsymbol{M}$. Then the region of $\boldsymbol{p}$ consists of all the points $\boldsymbol{q}$ which the influence of $\boldsymbol{p}$ is the strongest. The concept was formalized by mathematicians Dirichlet [23] and Voronoi [24], and the resulting structure became known as the Dirichlet tessellations or, more commonly today, the Voronoi diagrams.

2.2.2 Formal Definition

The definition of Voronoi diagrams depends on a metric which is often determined by the application. An ordinary type is the Euclidean-based Voronoi diagram.

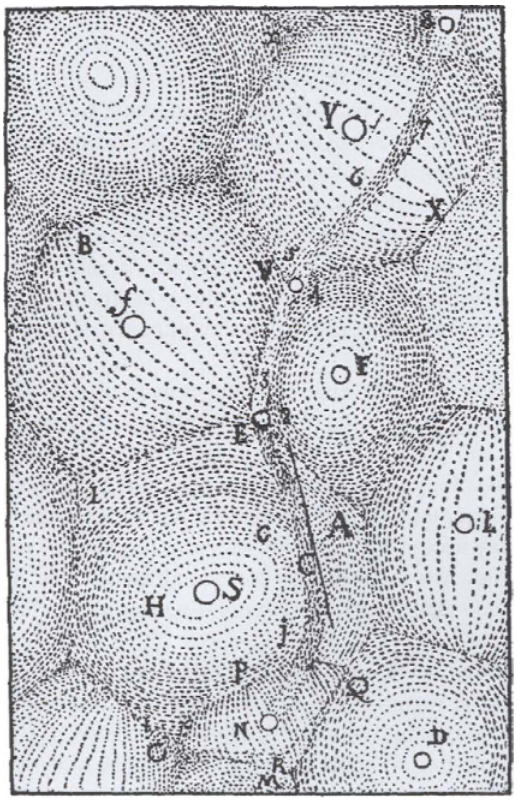


Figure 2.2: Descartes' decomposition of space into vortices. [21]

Let $\boldsymbol{\Omega}$ be the 2-dimensional space to be partitioned, and let $\boldsymbol{q} = (x, y)^\top \in \boldsymbol{\Omega}$ be a point in $\boldsymbol{\Omega}$. Let $d(\boldsymbol{a}, \boldsymbol{b})$ be the distance between point $\boldsymbol{a} = (a_1, a_2)$ and point $\boldsymbol{b} = (b_1, b_2)$. Then, the Voronoi partition $\boldsymbol{\mathcal{V}}(\boldsymbol{P}) = \{\boldsymbol{\mathcal{V}}_1, \ldots, \boldsymbol{\mathcal{V}}_n\}$ of $\boldsymbol{\Omega}$ generated by a set of Voronoi seeds $\boldsymbol{P} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\}$ can be formally defined as [25]

$$\boldsymbol{\mathcal{V}}_i = \left\{ \boldsymbol{q} \in \boldsymbol{\Omega} \mid d\left(\boldsymbol{q}, \boldsymbol{p}_i\right) \leq d\left(\boldsymbol{q}, \boldsymbol{p}_j\right), \forall j \neq i \right\} \tag{2.1}$$

where a Voronoi cell is associated to one Voronoi seed in an order-1 partition. Higher order partitions that associte k-clusters to each cell are possible, but not considered in this thesis. A common choice of the distance function $d(\cdot)$ is the Euclidean distance (or 2-norm), that is

$$d(\boldsymbol{a}, \boldsymbol{b}) = \|\boldsymbol{a} - \boldsymbol{b}\| = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2} \tag{2.2}$$

An Euclidean-based Voronoi diagram (2.1) partitions a plane $\boldsymbol{\Omega}$ into regions based on the distance to a specified set of points $\boldsymbol{P}$, known as Voronoi generators, Voronoi sites, or Voronoi seeds. Each of these regions $\boldsymbol{\mathcal{V}}_i$, called Voronoi cells, include the points in the region that are closest to the cell's generator, according

to a specific metric. The boundaries between the cells are called Voronoi edges, which are defined as an intersection between two Voronoi cells

$$\boldsymbol{l}_{ij} = \boldsymbol{\mathcal{V}}_i \cap \boldsymbol{\mathcal{V}}_j \tag{2.3}$$

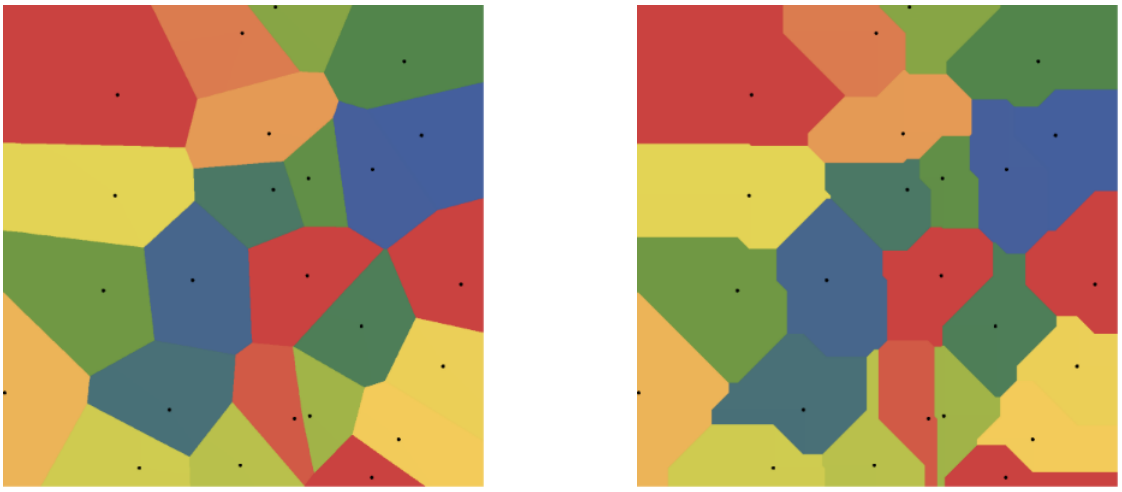
It has been proven using the Euler formula [26] that the average number of edges in the boundary of a Voronoi region is less than six in Ref. [21]. The intersection of Voronoi edges is called Voronoi vertices, which belongs to three or more Voronoi cells. What's more, When the two Voronoi regions $\boldsymbol{\mathcal{V}}_i$ and $\boldsymbol{\mathcal{V}}_j$ are adjacent (i.e., they share an edge), $\boldsymbol{\mathcal{V}}_i$ is called a Voronoi neighbor of $\boldsymbol{\mathcal{V}}_j$, and vice versa [27, 28].

The metric (2.1) of a Voronoi diagram can be generalized into different forms based on the application scenarios [25]. For instance, in analyzing customer preferences for shops, assuming all other factors are equal, customers are likely to choose the shop closest to them. In cities designed with a rectangular grid of streets, such as Manhattan, pedestrians typically walk along the grid, making the Euclidean distance, which measures the straight-line distance between two points, less applicable. Instead, the Manhattan distance, named after the grid layout of Manhattan's streets, is commonly used in urban scenarios. The Manhattan distance is defined to be the sum of the absolute differences of their respective Cartesian coordinates as follows [29]

$$d(\boldsymbol{a}, \boldsymbol{b}) = |a_1 - b_1| + |a_2 - b_2| \tag{2.4}$$

Figure 2.3 shows a comparison between an Euclidean-distance-based Voronoi diagram and a Manhattan-distance-based Voronoi diagram, both generated by the same set of Voronoi seeds.

In area coverage control problems, the metric used to generate Voronoi partitions is typically a sensor performance function, which models the radial degradation of agents' sensing from their location. Therefore, it is modelled by a monotonic function of the 2-norm. In this work, we assume homogeneous agents, meaning all agents have equal sensing capabilities. Therefore, comparing their sensing performance is equivalent to comparing their Euclidean distances. As a result, the Voronoi diagram for area coverage will resemble the standard Euclidean-based diagram.



(a) Voronoi diagram based on Euclidean distance.

(b) Voronoi diagram based on Manhattan distance.

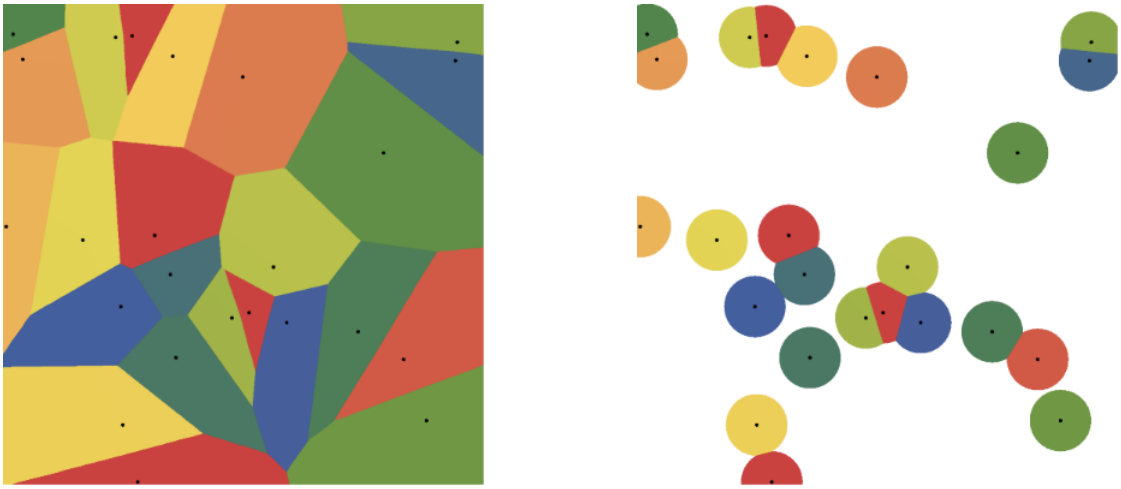
Figure 2.3: A comparison between the Euclidean-distance-based Voronoi diagram and the Manhattan-distance-based Voronoi diagram generated by a same set of Voronoi seeds. Adapted from [30]

2.2.3 Applications

Voronoi diagrams, also called Voronoi partitions or Voronoi tessellations, are fundamental geometric structures widely applied in various fields such as computational geometry, spatial analysis, and computer graphics. A Voronoi diagram partitions a plane into regions based on the distance to a specified set of points, known as Voronoi generators, Voronoi sites, or Voronoi seeds. Each of these regions, called Voronoi cells, include the points in the region that are closest to the cell's generator, according to a specific metric. The boundaries of the cells are called Voronoi edges, and the points where the edges intersect are called Voronoi vertices.

Voronoi diagrams and their variants are extensively used in various fields in different ways. The proximity-based property makes Voronoi diagrams an efficient data structure for proximity queries, which significantly improves computational efficiency in large-scale crowd simulations. For example, Ref.[31] apply truncated Voronoi diagrams in pedestrian crowd simulations, where the individuals in the crowd serve as the seeds of the Voronoi cells. As a special type of Voronoi diagram, the truncated Voronoi diagram limits the range of regions assigned to the Voronoi seeds, resulting in smaller circular Voronoi cells, which, together with GPU computation, further improves the computation speed. The visual difference between a Voronoi diagram and a truncated one generated from a same set of seeds is shown in figure 2.4. A similar application of truncated Voronoi diagrams is pre-

sented in [32], where agents are divided into sub-groups and assigned to Voronoi cells. By incorporating carefully designed collision-response rules based on the truncated cells, their algorithm generates realistic group behaviors in large crowds in real time.

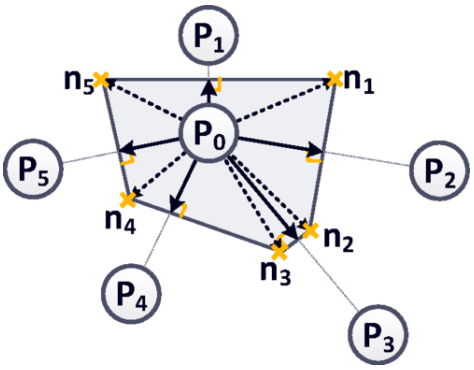


(a) Voronoi diagram (b) Truncated Voronoi diagram

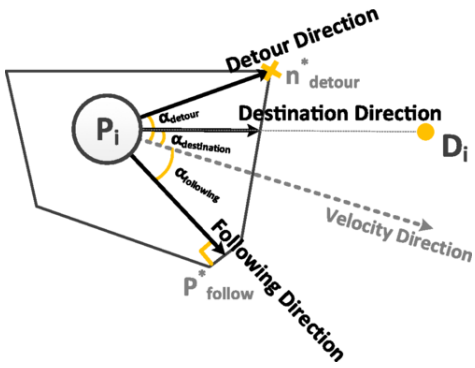
Figure 2.4: A comparison between a Voronoi diagram and a truncated Voronoi diagram generated from a same set of seeds. Adapted from [30]

Voronoi diagrams are also widely applied in studies and simulations of crowd density, where each pedestrian is represented as a Voronoi seed, occupying a corresponding Voronoi cell [8, 33]. In this context, smaller Voronoi cells indicate higher crowd density, aiding the visualization and analysis of pedestrian traffic flow experiments, as demonstrated in [8]. Based on the fact that the shape of a Voronoi cell reflects the relative positions of a pedestrian (Voronoi seed) and its neighbours, Xiao et al. [33] built an approach to simulate uni- and bi-directional pedestrian flows, where the logic of individuals' choices of directions depends on the shape of their respective Voronoi cells, target directions and current velocity directions, imitating the decision-making process for a pedestrian in a crowd flow, as shown in figure 2.5.

A different approach to utilize Voronoi diagrams for navigation can be found in reference [34], in which Sud et al. [34] proposed a Multi-agent Navigation Graph (MaNG) method to simulate a group of dynamic agents. In their work, each agent treats other agents as obstacles, who serve as the generators for the Voronoi diagram, such that the Voronoi edges provide the shortest path with maximal clearance, facilitating collision avoidance, as shown in figure 2.6. Beyond navigation,



(a) A Voronoi cell for a pedestrian



(b) Voronoi diagram based direction choices

Figure 2.5: Analyzation of potential directions of a pedestrian P_0 in crowd neighboured with 5 pedestrians[33].

Voronoi diagrams find applications in image compression, quadrature, finite difference methods, resource allocation, and cellular biology, as explored in [35].

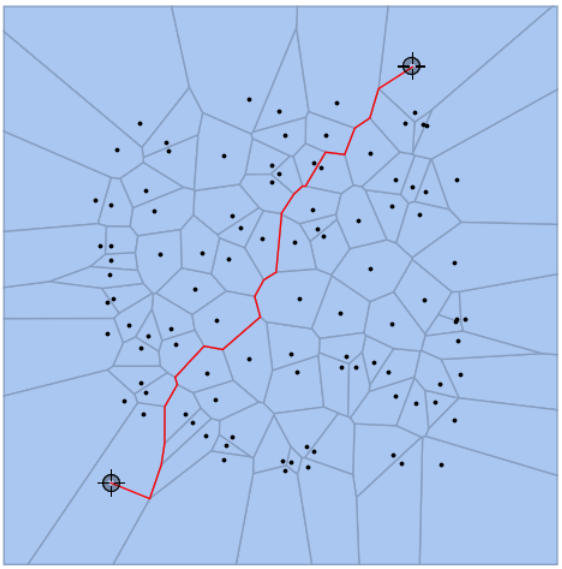
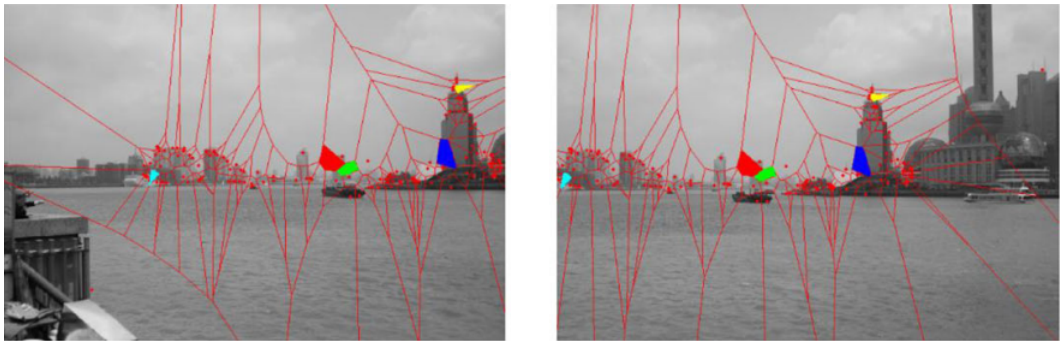


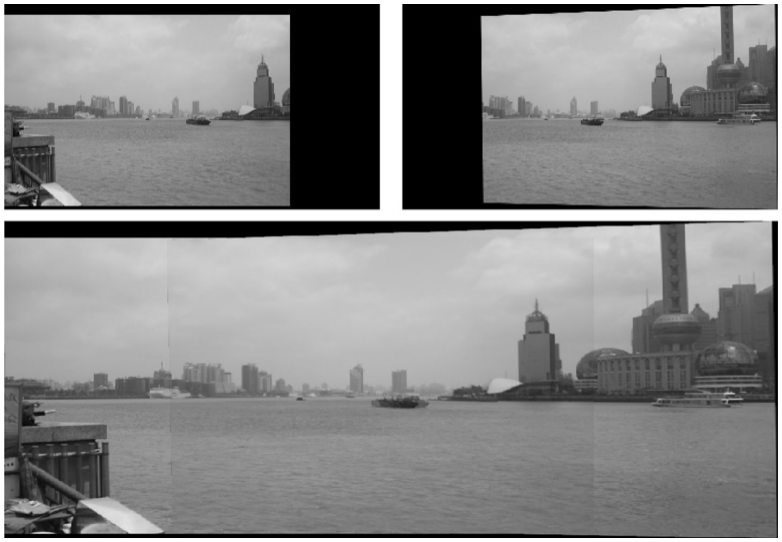
Figure 2.6: A schematic diagram of path planning by using Voronoi edges. Adapted from [36]

Laraqui et al.[37] presented an image stitching approach based on Voronoi diagrams. The process begins by identifying two sets of distinctive points from images taken from different angles, utilizing Scale-Invariant Feature Transform (SIFT) descriptors to facilitate information extraction. These distinctive points are then used as Voronoi seeds to partition the images into corresponding Voronoi diagrams. Based on the most similar Voronoi cells, the images are transformed and

stitched into a single composite image, as shown in figure 2.7.



(a) Matching Voronoi cells with the best similarity.



(b) Transforming and stitching images.

Figure 2.7: Stitching of two images[37]

Another important class of application of Voronoi diagrams is the area coverage control, which will be further elaborated in Section 2.3 given its relevance for the present work.

2.2.4 Centroidal Voronoi Tessellations

A centroidal Voronoi tessellation (CVT) is a special form of Voronoi tessellation where the Voronoi generators coincide with the centroids of their respective cells. Consider the workspace $\boldsymbol{\Omega}$ weighted by a strictly positive density field function $\Phi(\boldsymbol{q})$ and the set of Voronoi seeds $\boldsymbol{P}$ inducing the voronoi partition $\mathcal{V}(\boldsymbol{P}$. Then, the

mass m_i and the centroid $\boldsymbol{c}_i$ of the i-th Voronoi cell are given by

$$m_i(\boldsymbol{P}) = \int_{\mathcal{V}(\boldsymbol{P}_i} \Phi(\boldsymbol{q}) d\boldsymbol{\Omega} \tag{2.5a}$$

$$\boldsymbol{c}_i(\boldsymbol{P}) = \frac{1}{m_i(\boldsymbol{P})} \int_{\mathcal{V}(\boldsymbol{P}_i)} \boldsymbol{q}\Phi(\boldsymbol{q}) d\boldsymbol{\Omega} \tag{2.5b}$$

One of the approaches to generating centroidal Voronoi tessellation is Lloyd's algorithm [38], in which the Voronoi generators iteratively move to the positions of their corresponding Voronoi centroids and partition the workspace based on the updated configuration. The algorithm repeatedly executes the steps described in the following, as shown in figure 2.8:

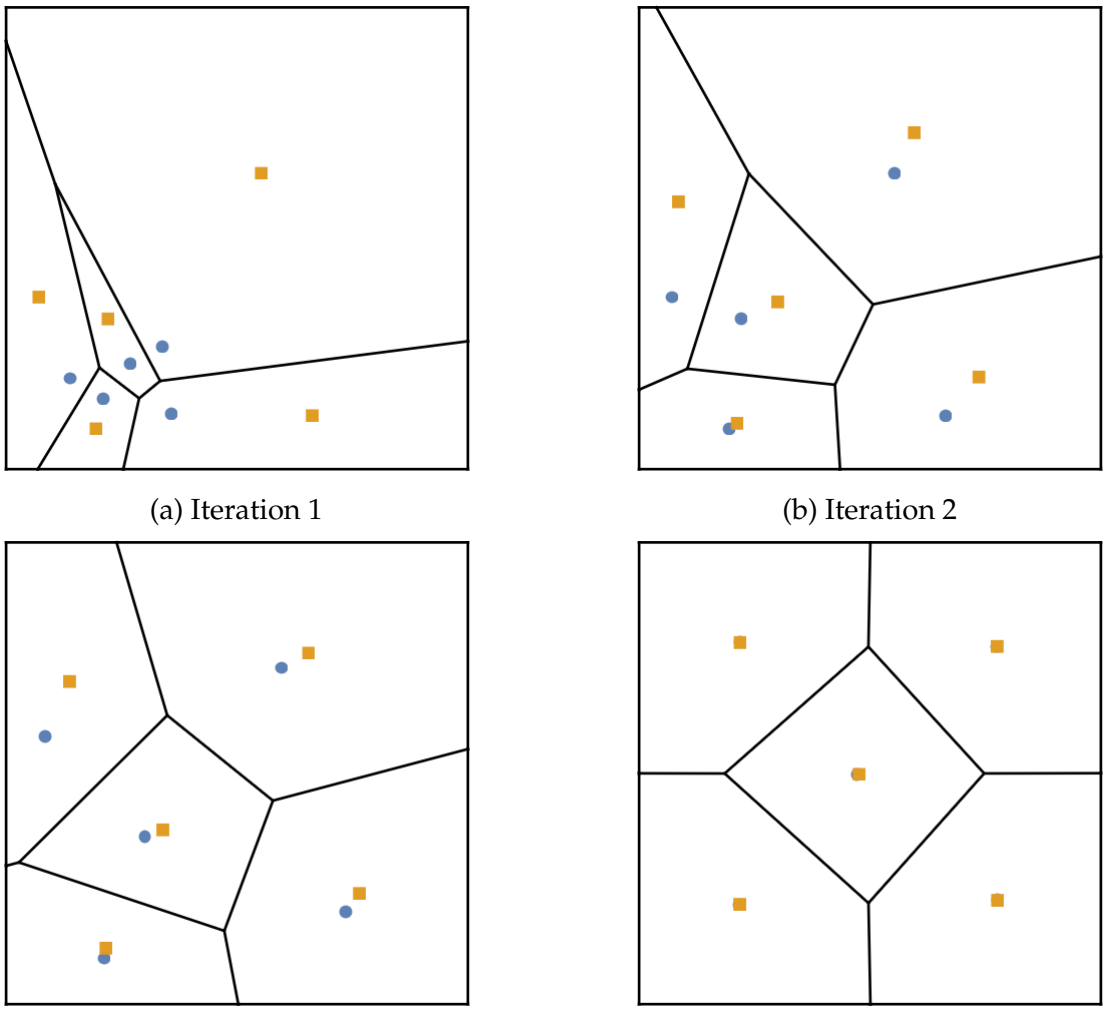
Lloyd's Algorithm Description

- **Initialization:** A set of n points are placed in the convex domain $\boldsymbol{\Omega}$ as the initial Voronoi generators.
- **Voronoi partition:** Based on the configuration of the Vornoi generators, the Voronoi diagram is computed.
- **Centroids computation:** Calculate the mass and centroid of each Voronoi cell.
- **Voronoi generators update:** Move the Voronoi generators to the centroids of their corresponding Voronoi cells.
- **Iteration:** Repeat step 2 to step 4 until convergence (i.e., the generators no longer vary).

Du et al. [35] proposed a generalized definition of centroidal Voronoi tessellation for a non-uniform region weighted by a density function, that is

$$\int_{\mathcal{V}} f(d(\boldsymbol{q}, \boldsymbol{p}^*))\Phi(\boldsymbol{q}) d\boldsymbol{\Omega} = \inf_{\boldsymbol{p}\in\mathcal{V}^*} \int_{\mathcal{V}} f(d(\boldsymbol{q}, \boldsymbol{p}))\Phi(\boldsymbol{q}) d\boldsymbol{\Omega} \tag{2.6}$$

The function $f(d(\boldsymbol{q}, \boldsymbol{p}))$ is convex in the distance, hence the definition of the optimal points $\boldsymbol{p}_i$ (i.e. the centroidal Voronoi tessellation) is taken by infimum. Based



(a) Iteration 1

(b) Iteration 2

(c) Iteration 3

(d) Iteration 20

Figure 2.8: An example of Lloyd's algorithm for 20 iterations. The Voronoi diagrams of the current positions of Voronoi sites (blue dots) are shown, and the yellow square denotes the corresponding centroids of the Voronoi cells.

on this, they interpreted Lloyd's algorithm as a fixed point iteration, that is, iteratively submit the mass centroids $\boldsymbol{c}$ into the Voronoi seeds $\boldsymbol{p}$ in equation (2.5) until convergence. When using the gradient descent method to optimize cost functions such as

$$\sum_{i=1}^{n} \int_{\mathcal{V}_i} f(d(\boldsymbol{q}, \boldsymbol{p}_i)) \Phi(\boldsymbol{q}) d\boldsymbol{\Omega} \tag{2.7}$$

they prove that stationary points of the cost function are the fixed points of the Voronoi mapping defined in equation (2.5). This framework addresses a class of

resource allocation problems and serves as the foundation of the Voronoi-based area coverage control for mobile sensing networks.

2.3 Area Coverage Control

Area coverage is an optimal control framework in which a finite set of mobile agents is optimally deployed in a given environment, where optimality is evaluated via a coverage metric that encodes the agents' sensing performance in the environment, weighted by a field defined on the same environment, often called risk density, that quantifies a distributed property. This could be for example the relative importance associated with different regions, so that area with higher risk demand more resources allocated to them, in the form of denser distribution of agents (mobile robots, sensors), as shown in figure 2.9. The agents serve as the seeds for the Voronoi diagram, and typically converge into a centroidal Voronoi configuration when the coverage metric is optimized, where the agents occupy the positions of the mass centroid of their respective Voronoi cells. The inherent characteristics of the centroidal Voronoi tessellation not only guarantee the collision avoidance of the agents between each other, but also have many potential applications in military and civilian fields, such as harbour patrolling, surveillance missions, and cooperative search and rescue, among others. Area coverage control algorithms in nonuniform environments have been presented in [27, 39, 40, 41, 42, 43, 44, 28, 45, 46]. Cortes et al. [27] formalized the properties of non-uniform area coverage control and its connection to Lloyd's algorithm [38], which is a gradient-descent search for the fixed points of the coverage metric, also can be referred to as a continuous form of Lloyd's algorithm.

Lee and Egerstedt [39] addressed the problem of generalizing area coverage control to time-varying environments, where the risk density function is time-varying, and the resulting control problem becomes non-autonomous due to the drift induced by the evolving environment. Lloyd's algorithm does not apply in the original form, since the problem is not an optimal regulator one in the search of fixed points of the autonomous coverage metric, but it is rather an optimal tracking problem, in which the centroids of optimal tessellations of the environment evolve due to the time dependency of the risk. In their work, optimality is guaranteed as long as the initial configuration of the agents is optimal; in other words, if the agents can be carried to an optimal centroidal configuration, then they can track the centroids' trajectories, maintaining optimality. An advancement was proposed

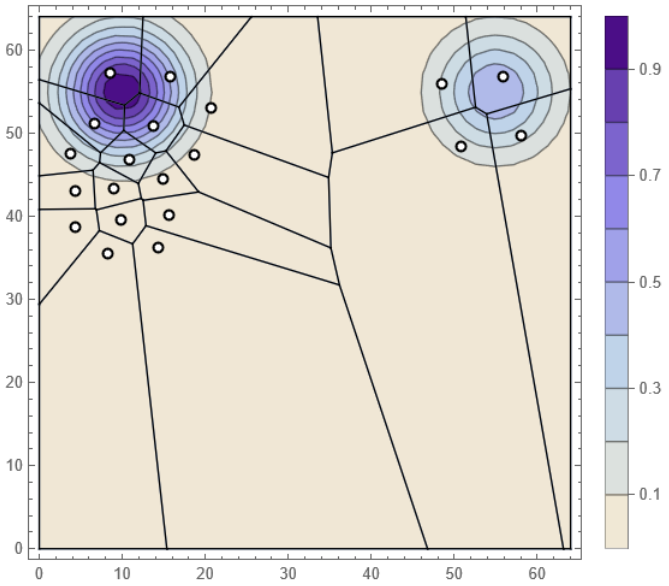


Figure 2.9: An example of the converged centroidal configuration of 20 agents for a mixture of 2 Gaussian densities, where the denser area attracts more agents.

by Miah et al. in reference [40], with a generalized Lloyd's algorithm that unifies the transient and the steady-state optimal tacking of time-varying centroids, allowing the agents to configure themselves asymptotically to optimize a related non-autonomous coverage metric. An application of [40] in a diffusive evolving environment can be found in [47].

A different development of Voronoi-based coverage control can be found in [48], where the authors introduce a modified Voronoi tessellation called Obstacle-Aware-Voronoi-Cell (OVAC), which is a modified Voronoi tessellation that dynamically weights the boundaries between agents and obstacles such that an agent's OAVC is tangent but never intersecting the obstacle, as shown in figure 2.10. Based on the OVAC, Abdulghafoor and Bakolas [43] introduced the idea of collective density of agents modelled as a Gaussian Mixture and designed a control law to track a time-varying environmental density while simultaneously driving the agents' density to the environmental density. Since the control itself could not guarantee the tracking of the reference density, they proposed an interpolation method for the feed-forward part of the control.

A detailed formulation with classic control solutions of area coverage control will be presented as a preliminary in Chapter 3.

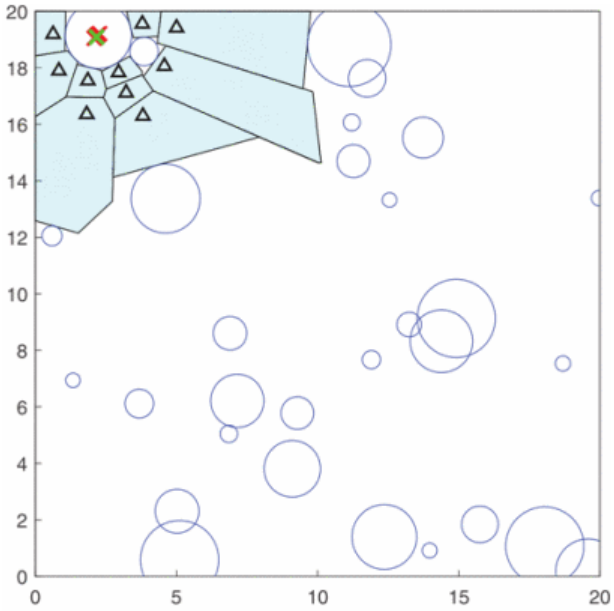


Figure 2.10: An example of Obstacle-Aware-Voronoi-Cells. [48]

2.4 Kullback-Leibler Divergence

Kullback-Leibler (KL) divergence (relative entropy, I-divergence) is a statistical distance to measure the difference between two probability density functions, with applications in several fields such as anomaly detection and diagnosis [49, 50, 51], layer selection for convolutional neural networks [52], and hypertension prediction [53] among others.

2.4.1 Preliminary: Information Entropy

Claude Shannon [54] first introduced the concept of information entropy in his groundbreaking paper "A Mathematical Theory of Communication," published in 1948, which laid the foundation for modern digital technology. The core idea behind quantifying information is that the less possible is the occurrence of an event, the more information it contains. Consider a discrete random variable X mapping to a finite dimensional event space $\{x_1, \ldots, x_n\}$, with a corresponding probability mass function $P(X) = \{p_1, \ldots, p_n\}$, where $\sum_{i=1}^{n} p_i = 1$ and $0 \leq p_i \leq 1$. Then, for a specific event x_i, the information content can be defined as a logarithm function to its probability as

$$I(x_i) = -\log P(x_i) \tag{2.8}$$

and the information entropy is defined as the expectation of the information content as

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{2.9}$$

where, if the logarithm is base 2, the unit of information is "bits" (or “shannons”), which introduces the well-known smallest computer storage unit. Just as entropy in thermodynamics precludes the possibility of perpetual motion, entropy in information theory reveals the limit of information compression.

2.4.2 Kullback-Leibler Divergence

Building on the concept of information entropy, Kullback and Leibler [55] introduced a statistical measure known as Kullback-Leibler divergence (KL divergence) or relative entropy. This measure quantifies the distance between a given probability density function and a reference one. Formally, given discrete probability distributions P and Q defined on the same sample space $\mathcal{X}$, the KL divergence from Q to P is denoted by $D_{KL}(P||Q)$, and it is defined as the expected value of logarithmic difference between the probability distributions P and Q, where the expectation is taken over the distribution P, that is

$$\begin{aligned} D_{KL}(P||Q) &= \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \\ &= \mathbb{E}_P \left[\log \left(\frac{P(x)}{Q(x)} \right) \right] \end{aligned} \tag{2.10}$$

The KL divergence is a non-negative real number, with zero value if and only if the distributions P and Q are identical. It is worth noting that the KL divergence is not a metric in the sense of statistics because of its asymmetry, that is $D_{KL}(P||Q) \neq D_{KL}(Q||P)$. Therefore, KL divergence is often referred to as a statistical "distance" rather than a metric. For the continuous distributions P and Q, the KL divergence is defined by the integral

$$D_{KL}(P||Q) = \int_{\mathcal{X}} p(x) \log \left(\frac{p(x)}{q(x)} \right) \tag{2.11}$$

where $p(x)$ and $q(x)$ are respectively the probability density functions of P and Q.

2.4.3 Closed-form of KL Divergence between Gaussians

The Gaussian distribution, also known as the normal distribution, is pivotal across numerous fields due to its mathematical properties and the natural phenomena it models. The probability function related to a d-dimensional Gaussian distribution is given by

$$f(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) \tag{2.12}$$

Here, $\boldsymbol{\mu} \in \mathbb{R}^d$ denotes the d-dimensional mean vector, and $\boldsymbol{\Sigma}$ represents the covariance matrix, which is a $d \times d$ symmetric positive definite matrix. Specifically, for two different d-dimensional multivariate Gaussian distributions $P : x \sim \mathcal{N}_d(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1})$ and $Q : x \sim \mathcal{N}_d(\boldsymbol{\mu_2}, \boldsymbol{\Sigma_2})$, the KL divergence has the following closed form [56]

$$D\left(P \| Q\right) = \frac{1}{2}\left(\left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)^\top \boldsymbol{\Sigma}_2^{-1}\left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right) + \operatorname{tr}\left(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\right) - \log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} - d\right) \tag{2.13}$$

where tr is the trace operator. A detailed derivation can be found in [57].

2.5 Summary

This chapter reviews the works and techniques relevant to and that inspire this thesis, including Voronoi diagrams, area coverage control and KL divergence. The area coverage control utilizes the Voronoi diagrams to find the stationary points of the coverage metric by gradient descent. When the field defined in the environment is time-invariant, this process can be viewed as a continuous form of Lloyd's algorithm. The converged optimal configuration is called centroidal Voronoi tessellation, where the agents occupy the centroid of the associated Voronoi cells. When the field is allowed to evolve, the system becomes non-autonomous due to the drift introduced by the time-varying field, thus the control becomes tracking the trajectories of the time-varying centroids.

Inspired by this, this thesis attempts to drive a set of mobile agents using a user-defined field. To achieve this, we introduce a collective density with moments depending on their spatial configuration in a two-dimensional normal distribution, and minimizethe distance between the agents' density and a target one. A similar idea can be found in [43], where the authors modeled the agents' distribution as

a Gaussian Mixture and designed a control law to track a reference environmental density while simultaneously driving the agents' density to the environmental density. However, since the control itself could not guarantee the tracking of the reference density, they proposed an interpolation method for the feed-forward part of the control.

To explicitly include the tracking of a reference density, this thesis proposes to use the KL divergence between the agents' collective density and a target one as a part of the multi-objective optimization metric. The minimization of the KL divergence enforces convergence in the sense of the agents' spatial distribution. The idea of using environmental cues to drive collective behaviors is quite mature, but the direct use of KL divergence as the motion control metric for multi-agent systems is novel.

Chapter 3

Preliminary: Area Coverage Control based on Voronoi diagrams

3.1 Formulation of Area Coverage Control Problems with Time-invariant Environments

Problem Description

Area coverage problems focus on the optimal deployment of a finite set of networked mobile agents within a bounded, two-dimensional convex region $\mathbf{\Omega} \subset \mathbb{R}^2$. A field is defined over the same workspace, quantifying a distributed property, such as relative importance or risk associated with different regions. This field is typically modelled by a smooth, or at least twice differentiable mass density function [40], such as a two-dimensional multi-variate Gaussian density or a Gaussian mixture.

Naturally, an agent's ability to sense, measure, or detect the risk density diminishes radially from its position. This decay is typically modelled by a monotonic function based on $r_i = \|\mathbf{p}_i - \mathbf{q}\|$, representing the Euclidean distance between a point $\mathbf{q} \in \mathbf{\Omega}$ and the position of the i-th agent $\mathbf{p}_i$, often referred to as the sensing performance function. Both monotonically increasing and decreasing functions can be employed, as long as the underlying meaning stands for the radial attenuation of the sensing performance. Here, we present the control algorithm in Ref. [27], in which the sensing performance function is defined by

$$f(r_i) = r_i^2 \tag{3.1}$$

In the context of area coverage control, the Voronoi partition of the workspace $\mathbf{\Omega}$ can be defined based on the sensing performance functions. Note that the sensing performance function in equation (3.1) is increasing with respect to r_i, so a smaller value indicates better sensing performance. Therefore, the Voronoi cell associated with the i-th agent is given by

$$\boldsymbol{\mathcal{V}}_i = \{\mathbf{q} \in \mathbf{\Omega} \mid f(r_i) \leq f(r_i), \forall j \neq i\} \tag{3.2}$$

The sensing performance of the agents can also be modelled by a monotonically decreasing function as in[41, 42, 47]

$$f(r_i) = \exp\left(-\frac{r_i^2}{2\sigma^2}\right) \tag{3.3}$$

In this case, the definition of the i-th Voronoi cell becomes

$$\boldsymbol{\mathcal{V}}_i = \{\mathbf{q} \in \mathbf{\Omega} \mid f(r_i) \geq f(r_i), \forall j \neq i\} \tag{3.4}$$

In Ref. [27], the communication network among the interested set of agents is typically assumed to be fully connected and synchronous, which means the agents are capable of broadcasting their spatial coordinates to all the other agents in the group without delay, such that distributed computations based on the knowledge of the state of the group are possible. A thorough discussion of network topology compatible with Voronoi tessellation computations can be found in [27]. In addition, Miah et al. [42] addressed the non-uniform area coverage in time-varying stochastically intermittent communication topology.

The agents are described as purely kinematics as single integrator [28]

$$\dot{\mathbf{p}}_i(t) = \mathbf{u}_i(t) \tag{3.5}$$

where $\mathbf{p}_i(t) = [x_i(t), y_i(t)]^\top$ is the position of the i-th agent, and $\mathbf{u}_i(t) = \left[u_{xi}(t), u_{yi}(t)\right]^\top$ is the velocity input at time t, both referred to a rectangular Cartesian system spanning the workspace. The collection of the agents' positions is $\mathbf{P} = [\mathbf{p}_1(t)^\top, ..., \mathbf{p}_n(t)^\top]^\top$, and $\mathbf{U} = [\mathbf{u}_1(t)^\top, ..., \mathbf{u}_n(t)^\top]^\top$ is the set of corresponding control inputs.

The optimality of the coverage is evaluated by time-invariant a coverage metric encoding the sensing performance functions $f(\cdot)$ of the agents and the risk density

$\phi(\mathbf{q})$ defined on the workspace $\mathbf{\Omega}$

$$H(\mathbf{P}, \boldsymbol{\mathcal{V}}) = \sum_{i=1}^{n} \int_{\mathcal{V}_i} f(r_i)\, \phi(\mathbf{q})\, d\mathbf{q} \tag{3.6}$$

Formally, the coverage control problem can be stated as an optimization problem as follows: [47]

Given the coverage metric H in (3.6), determine an optimal configuration of the agents, that is

$$\lim_{t\to\infty} \min_{\mathbf{P}\in\mathbf{\Omega}} H \text{ subject to (3.5)} \tag{3.7}$$

The converged centroidal Voronoi configuration is not necessarily unique [27], representing a local minimum of the locational optimization metric H.

Control Solutions

Area coverage control in time invariant environments, corresponding to time-invariant density functions, can be interpolated as a continuous-time Lloyd algorithm [27], where the agents move along the gradient descent flow of the coverage metric H. Unlike the classic Lloyd's algorithm (Voronoi partition updates after the agents coincide with the centroids), in area coverage control, the Voronoi partition is dynamically updated in response to the evolution of the agents' configurations, until the system converges to a centroidal Voronoi configuration where the agents are the seeds. The gradient of the coverage metric H with respect to $\mathbf{p}_i$ is given by [40]

$$\frac{\partial H}{\partial \mathbf{p}_i} = \int_{\mathcal{V}_i} -2\frac{\partial f(r_i)}{\partial {r_i}^2} (\mathbf{q} - \mathbf{p}_i)\, \phi(\mathbf{q})\, d\mathbf{q} \tag{3.8}$$

For a C^2 sensing performance function $f(\cdot)$, following the definition in [40], a generalized density is defined by

$$\phi_i^H(\mathbf{q}) = -2\frac{\partial f(r_i)}{\partial {r_i}^2}\phi(\mathbf{q}) \tag{3.9}$$

with the corresponding generalized mass and centroid

$$m_i^H(\mathbf{p}) = \int_{\mathcal{V}_i} \phi^H(\mathbf{q}) d\mathbf{\Omega} \tag{3.10a}$$

$$\mathbf{c}_i^H(\mathbf{p}) = \frac{1}{m_i^H(\mathbf{p})} \int_{\mathcal{V}_i} \mathbf{q}\phi^H(\mathbf{q}) d\mathbf{\Omega} \tag{3.10b}$$

Thus the gradient of H can be written as

$$\frac{\partial H}{\partial \mathbf{p}_i} = m_i^H(\mathbf{c}_i^H - \mathbf{p}_i) \tag{3.11}$$

Assuming the set of agents has full knowledge of the time-invariant density $\phi(\mathbf{q})$, the gradient descent feedback law is given by

$$\mathbf{u}_i = -k\frac{\partial H}{\partial \mathbf{p}_i} = -km_i^H(\mathbf{c}_i^H - \mathbf{p}_i) \tag{3.12}$$

where $k > 0$ is the control gain, driving the system along the gradient descent direction. Based on LaSalle's principle [58], the agents' positions converge to the largest invariant set $\mathbf{c}_i^H = \mathbf{p}_i$, which is the centroidal Voronoi tessellation, as proven in [28].

3.2 Area Coverage Control with Time-Variant Density Function

An important advancement in area coverage problem is the extension to time-varying environments, which make the system non-autonomous. Then, the problem becomes a tracking problem of the trajectories of a set of generalized centroids, and LaSalle's principle needs to be extended via Barbalat's lemma in order to prove convergence of the agents to such trajectories. The environment is time-varying because the risk density function ϕ in equation (3.6) is time-dependent, resulting in a non-autonomous optimization control problem, in which the optimal configuration evolves over time.

Cortes et al. (2002) proposed a solution of tracking in time-varying environments in [28] that drives the agents to form a time-varying centroidal Voronoi tessellation and to remain optimal, under a strict assumption that time derivative of $I_{\mathcal{V},\mathbf{p}_i}$, which is the moment of inertia of the agents regarding the associated Voronoi cells, to be zero. As mentioned above, the agents occupy the centroids of their respective Voronoi cells in centroidal Voronoi tessellation. According to the parallel axis theorem, the polar moments of inertia of each cell regarding the agents' positions $I_{\mathcal{V}_i,\mathbf{p}_\Omega}$ and the the total moments of inertia of the environment $\mathbf{\Omega}$ with respect

to its center of mass $\mathbf{c}_\Omega$ have the following connection [59]

$$\mathbf{c}_\Omega = \frac{\sum_{i=1}^{n} m_i \mathbf{p}_i}{\sum_{i=1}^{n} m_i} \tag{3.13a}$$

$$I_{\mathcal{V}_i,\mathbf{p}_i} = \int_{\mathcal{V}_i} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi(\mathbf{q}) d\mathbf{q} \tag{3.13b}$$

$$\begin{aligned} I_{\Omega,\mathbf{c}_\Omega} &= \sum_{i=1}^{n} \left(I_{\mathcal{V},\mathbf{p}_i} + m_i \|\mathbf{c}_\Omega - \mathbf{p}_i\|^2 \right) \\ &= \int_{\mathcal{V}_i} \|\mathbf{q} - \mathbf{c}_i\|^2 \phi(\mathbf{q}) d\mathbf{q} \end{aligned} \tag{3.13c}$$

When the agents are distributed in the centroidal Voronoi tessellation, the summation of $I_{\mathcal{V}_i,\mathbf{p}_\Omega}$ equals the total moment of inertia of the environment. However, as the density is allowed to be dependent on time, the moment of inertia of the workspace $I_{\Omega,\mathbf{c}_\Omega}$ is not likely to remain constant, , as it is dependent on the density in eq.(3.13c). Therefore, the assumption on the time derivative of $I_{\Omega,\mathbf{c}_\Omega}$ equal to zero is difficult to establish.

To address this, Kennedy et al. [45] proposed a generalized algorithm that guarantees practical stability under relaxed technical assumptions. By introducing an additional term in the control law to account for the varying density and assuming bounds on the rate of change of the density, the algorithm ensures convergence to a set.

Miah et al. [40] advanced the generalized non-autonomous area coverage control by proposing the feedback law

$$\mathbf{u}_i = \frac{\partial H/\partial \mathbf{p}_i}{\|\partial H/\partial \mathbf{p}_i\|^2} \left(-k \left\| \mathbf{c}_i^H - \mathbf{p}_i \right\|^2 - \frac{\partial H_i}{\partial t} \right) \tag{3.14}$$

where $\frac{\partial H_i}{\partial t}$ denotes the partial derivative of the i-th component of coverage metric H with respect of time. It is shown in [40] that this control law monotonically increases the coverage metric. Furthermore, asymptotic tracking of time-varying centroidal configurations $\mathbf{c}^H$ are established by using Barbalat's lemma [60]. Therefore, the algorithm asymptotically stabilizes the agents' states on the trajectories of time varying centroids, while increasing the coverage metric. This algorithmic framework finds potential applications in both military [41, 44] and civilian fields

[47].

In [41], this control law was applied to the optimal deployment of multiple mobile agents to protect high-value units from multiple attacks in a harbor-like environment. By incorporating an extended Kalman filter and an optimal linear quadratic control technique, the approach minimizes the actuator energy required by each agent to intercept threats within their respective Voronoi cells. In reference [44], a dynamic task assignment mechanism was introduced to increase the possibility of successful detecting and intercepting threats, which reformulated the problem into a multi-objective optimization problem with respect to the total travel cost of all agents. This approach allows agents to switch to uniform coverage of the harbor when no threat is detected or after the threat has been neutralized.

In the civilian domain, [47] advanced the work in [40] by introducing diffusive density driven by time-varying boundary conditions, which could model realistic natural phenomena affected by penetration of substances governed by diffusion mechanisms, such as oil leakage in a marine environment, benefiting environmental monitoring and intervention tasks.

Chapter 4

Non-autonomous Formation Control via Voronoi Tessellation and Kullback-Leiber Divergence

This present an algorithm to control the spatial distribution of kinematic multi-agent systems in two-dimensional workspace. Leveraging on the coverage control framework, the problem is formulated as a multi-objective optimization with a performance index composed of the area coverage metric and of the Kullback–Leibler (KL) divergence [1].

4.1 Problem Statement for Multi-objective Optimization

This section presents the formal problem formulation. The control problem is framed as a multi-objective optimization, in which the performance index is defined as a convex combination of area coverage metric and KL divergence. Inspired by [12, 43], we introduce an abstraction to estimate the agents' collective density depending on their spatial configuration in a bi-variate normal distribution, which allows us to incorporate KL divergence as part of the optimization index.

The coverage metric encodes an environmental density and a weighted quantification of the performance of each agent with respect to points in the workspace (often referred to as sensing performance in mobile sensing networks), which allows the partition of the environment into regions that are "best served" by the agents to which they are assigned, resulting in the emergence of a Voronoi tessel-

lation of the workspace. Meanwhile, the KL divergence measures the residual between the statistical distribution of the agents in the workspace and a user-defined target distribution. The connection is that the density in the coverage metric is the probability density function (PDF) of the target distribution in the KL term.

The optimization of the coverage metric results in a centroidal Voronoi tessellation, which guarantees collision avoidance performance between the agents. Simultaneously, the minimization of the KL divergence forces the converged Voronoi centroidal configuration to have the same spatial statistical moments as the target density, which may allow users to intuitively command the collective positioning of the group by designing the target Gaussian density via its elliptical contour.

4.1.1 Agents' Dynamics and Collective Density

Formal Definition

We consider a set of networked mobile agents deployed in a bounded 2-dimensional convex spatial region $\mathbf{\Omega} \in \mathbb{R}^2$. The agents are capable of sensing/measuring distributed quantities in the workspace, and broadcasting their spatial coordinates to all the other agents in the group. Here, we assume a fully connected, synchronous network topology, so that distributed computations based on the knowledge of the state of the group are possible. The relaxation of this hypothesis is beyond the scope of this work; a thorough discussion of network topology compatible with Voronoi tessellation computations can be found in [27]. Each agent is modelled as a simple kinematic integrator

$$\dot{\mathbf{p}}_i(t) = \mathbf{u}_i(t) \tag{4.1}$$

where $\mathbf{p}_i(t) = [x_i(t), y_i(t)]^\top$ is the position of the i-th agent, and $\mathbf{u}_i(t) = \left[u_{xi}(t), u_{\mathrm{yi}}(t)\right]^\top$ is the velocity input at time t, both referred to a rectangular Cartesian system spanning the workspace. The collection of the agents' positions is $\mathbf{P} = [\mathbf{p}_1(t)^\top, ..., \mathbf{p}_n(t)^\top]^\top$, and $\mathbf{U} = [\mathbf{u}_1(t)^\top, ..., \mathbf{u}_n(t)^\top]^\top$ is the set of corresponding control inputs.

Inspired by [12], the agents' formation is abstracted into a two-dimensional multivariate Gaussian distribution whose moments are generated by their spatial coordinates. The corresponding density is given by

$$\phi_P(\boldsymbol{\mu}_P, \mathbf{\Sigma}_P; \mathbf{q}) = \frac{\exp\left(-\frac{1}{2}\left(\mathbf{q} - \boldsymbol{\mu}_P\right)^\top {\mathbf{\Sigma}_P}^{-1}\left(\mathbf{q} - \boldsymbol{\mu}_P\right)\right)}{2\pi\sqrt{\det \mathbf{\Sigma}_P}} \tag{4.2}$$

where $\mathbf{q} = [x, y]^\top \in \mathbf{\Omega}$ represents a point in the workspace, $\boldsymbol{\mu}_P \in \mathbb{R}^2$ and $\mathbf{\Sigma}_P \in \mathbb{R}^{2\times 2}$ denote the mean and the unbiased estimation of covariance matrix respectively, defined by

$$\boldsymbol{\mu}_P = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i \tag{4.3a}$$

$$\mathbf{\Sigma}_P = \frac{1}{n-1}\sum_{i=1}^{n} (\mathbf{p}_i - \boldsymbol{\mu}_P)(\mathbf{p}_i - \boldsymbol{\mu}_P)^\top \tag{4.3b}$$

where $\mathbf{\Sigma}_P$ is a symmetric positive-definite matrix, with entries given by $\Sigma_{P_{11}} = \sigma_{Px}^2$, $\Sigma_{P_{22}} = \sigma_{Py}^2$ and $\Sigma_{P_{12}} = \Sigma_{P_{21}} = \rho_P \sigma_{Px}\sigma_{Py}$, where σ_{Px} and σ_{Py} are the standard deviations along the reference frame axes, and $\rho_P \in [-1, 1]$ is the correlation coefficient.

Bound of Region Occupied by the Agents: Confidence Ellipse Contour

It is notable that when agents are modelled by equation (4.2), the region they occupy can be represented by an ellipse contour of the Gaussian density at a constant confidence level. Specifically, the contour ellipse of the probability density for a 2-dimensional multivariate Gaussian distribution is given by [12]

$$(\mathbf{q} - \boldsymbol{\mu}_P)^\top \mathbf{\Sigma}_P^{-1} (\mathbf{q} - \boldsymbol{\mu}_P) = -2\ln(1 - c) \tag{4.4}$$

where $0 \leq c \leq 1$ denotes the confidence level. In statistics, confidence level represents the degree of certainty that a sample accurately reflects the population from which it was drawn. In this work, it is used to manipulate the boundary of the region occupied by the agents. A greater confidence level leads to a bigger ellipse and a more conservative estimation of the agents' boundary. A typical choice of c is 95% or 99%.

As the covariance matrix $\mathbf{\Sigma}_P$ is positive definite, it can be written into a multiplication of scale matrix and rotational function by using eigendecomposition as follows [61, 62]

$$\mathbf{\Sigma}_P = \boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^\top \tag{4.5}$$

where the scale matrix $\mathbf{\Lambda}$ and the rotational matrix $\boldsymbol{R}$ are given by

$$\mathbf{\Lambda} = \begin{bmatrix} s_x^2 & 0 \\ 0 & s_y^2 \end{bmatrix} \tag{4.6a}$$

$$\boldsymbol{R} = [\boldsymbol{\lambda}_x | \boldsymbol{\lambda}_y] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{4.6b}$$

here, s_x^2 and s_y^2 denote the eigenvalues, and $\boldsymbol{\lambda}_y$ stand for the normalized eigenvectors for the covariance matrix $\boldsymbol{\Sigma}_P$. When applied to the ellipse of a constant confidence level c in equation 4.4, it represents an ellipse centered at $\boldsymbol{\mu}_P$, with axes length of $\sqrt{-2\ln(1-c)s_x}$ and $\sqrt{-2\ln(1-c)s_y}$, and rotated counter-clockwise by θ degrees.

This property can be applied in controlling the collective position and motion of the agents, for example designing target density according to the ellipse contour to direct the agents through a path (tunnel, corridor) without collision against the walls. A simulation of tunnel scenario is shown in subsection 5.2.2.

4.1.2 Time-varying Target Density

In area coverage problems, there is typically a reference density defined in $\boldsymbol{\Omega}$ weighting the points $\mathbf{q}$ in the workspace. This density is referred to as risk density, quantifying the relative importance of different regions, and it is assumed to be twice differentiable [40]. A common choice of function for modelling the risk density is two-dimensional Gaussian density. However, various density functions can be applied in Voronoi-based formation control algorithms [27]. In Ref. [47], the coverage metric is coupled to the evolution of a diffusive density, modelling scenarios such as environmental dispersion. In this work, a bi-variate Gaussian density depending on time through its statistical moments is adopted to model the desired reference density, that is

$$\phi_d(\mathbf{q}, t) = \frac{\exp\left(-\frac{1}{2}\left(\mathbf{q} - \boldsymbol{\mu}_d(t)\right)^\top \boldsymbol{\Sigma}_d(t)^{-1}\left(\mathbf{q} - \boldsymbol{\mu}_d(t)\right)\right)}{2\pi\sqrt{\det \boldsymbol{\Sigma}_d(t)}} \tag{4.7}$$

where $\boldsymbol{\mu}_d(t) \in \mathbb{R}^2$ and $\boldsymbol{\Sigma}_d(t) \in \mathbb{R}^{2\times 2}$ are respectively the mean and the covariance matrix.

4.1.3 Multi-objective performance index

Coverage Metric H

The coverage metric H is introduced to drive the agents into the centroidal Voronoi configuration, whose underlying properties also guarantee collision avoidance be-

tween the agents as long as the generated Voronoi partitions are non-singular, that is, no cell can degenrate into a set of zero measure (point or line) [1]. Formally, it is defined by [40]

$$H(\mathbf{p}, \mathcal{V}, t) = \sum_{i=1}^{n} \int_{\mathcal{V}_i} f(r_i)\, \phi_d(\mathbf{q}, t)\, d\mathbf{q} \tag{4.8}$$

where the region $\mathcal{V}_i$ represents the Voronoi cell associated with the i-th agent, which is generated based on the sensing performance function $f(r_i)$. Differently than in the classical formulation of area coverage control, we consider ϕ_d to be a user-defined, desired time-varying density in $\mathbf{\Omega}$, as defined in equation (4.7). In other words, optimizing this metric will lead the agents to form a centroidal Voronoi configuration weighted by the specified density ϕ_d.

Voronoi tessellations emerge in area coverage optimization problems with mobile sensor networks, where the agents serve as the generators or seeds that partition the monitored working area $\mathbf{\Omega}$ into a set of Voronoi cells $\boldsymbol{\mathcal{V}} = [\mathcal{V}_1, ..., \mathcal{V}_n]^\top$ [27]. In this work, the monotonic increasing sensing performance function in (3.1) and the compatible generalized Voronoi cell partition metric in (3.2) are employed.

KL Divergence D

The KL-Divergence is a non-symmetric measure of the difference between two probability distributions as disscussed in 2.4.2. In this work, it is used to encode a second objective into the system's performance metric, attempting to minimize the error between the moments of the target distribution and of the agents' distribution. The KL Divergence between two bi-variate Gaussian distributions can be derived in closed form in terms of the moments [57, 63]

$$D(\phi_d \| \phi_P) = \frac{1}{2}\left((\boldsymbol{\mu}_d - \boldsymbol{\mu}_P)^\mathrm{T} \boldsymbol{\Sigma}_d^{-1} (\boldsymbol{\mu}_d - \boldsymbol{\mu}_P) + \mathrm{tr}\left(\boldsymbol{\Sigma}_d^{-1}\boldsymbol{\Sigma}_P\right) - \ln \frac{|\boldsymbol{\Sigma}_P|}{|\boldsymbol{\Sigma}_d|} - 2 \right) \tag{4.9}$$

where $D \in \mathbb{R}^+$, ϕ_d and ϕ_P are defined in (4.7) and (4.2) respectively, and tr is the trace operator. A lower value of the KL divergence (4.9) indicates closer similarity between the densities. Particularly, (4.9) reaches zero when the two densities are identical in terms of the first two moments.

Based on the quantities defined in this section, the optimal control problem can be formulated as follows: consider a team of mobile agents subjected to the dynamics (4.1), and determine a control that drives the system to an optimal spatial

configuration such that (4.11) is minimized, that is

$$\lim_{t\to\infty} \min_{\mathbf{P}\in\mathbf{\Omega}} J \text{ subject to (3.5)} \tag{4.10}$$

To simultaneously pursue the locational optimization objective and the minimization of the distance between the agents' spatial distribution ϕ_p and the target distribution ϕ_d, we adopt the multi-objective system's performance metric

$$J = \alpha H + \beta D \tag{4.11}$$

where H denotes the coverage metric and D represents the KL divergence, each weighted by positive scalars α and β respectively.

4.1.4 Gradients and Stationary Points

A standard way to approach optimal control problems is to generate inputs along the gradient of the performance index, so that the trajectories climb or descend the performance toward locally optimal stationary points [64].

Gradients of Coverage Metric

For the gradient of the coverage metric H with respect of $\mathbf{p}_i$, we express it as a compact form by introducing modified mass and centroid, generalizing the formulation in [40, 47]. The gradient of the coverage metric H with respect to the agents' states $\mathbf{p}_i$ is

$$\frac{\partial H}{\partial \mathbf{p}_i} = \int_{\mathcal{V}_i} -2\frac{\partial f(r_i)}{\partial {r_i}^2}(\mathbf{q}-\mathbf{p}_i)\,\phi_d(\mathbf{q},t)\,d\mathbf{q} \tag{4.12}$$

The generalized density is defined as

$$\phi_i^H = -2\frac{\partial f(r_i)}{\partial {r_i}^2}\phi_d(\mathbf{q},t) \tag{4.13}$$

then the generalized mass and centroid of the i-th Voronoi cell are given by

$$m_i^H = \int_{\mathcal{V}_i} \phi_i^H\,d\mathbf{q} \tag{4.14a}$$

$$\mathbf{c}_i^H = \int_{\mathcal{V}_i} \phi_i^H \, d\mathbf{q} / m_i^H \tag{4.14b}$$

Therefore, the gradient of H with respect to $\mathbf{p}_i$ in equation (4.12) can be written as

$$\frac{\partial H}{\partial \mathbf{p}_i} = m_i^H (\mathbf{c}_i^H - \mathbf{p}_i) \tag{4.15}$$

Gradients of KL Divergence

By using the chain rule, the KL divergence between ϕ_d to ϕ_P can be expressed as

$$\frac{\partial D}{\partial \mathbf{p}_i} = \frac{\partial \boldsymbol{\mu}_p}{\partial \mathbf{p}_i} \frac{\partial D}{\partial \boldsymbol{\mu}_p} + \frac{\partial D}{\partial \boldsymbol{\Sigma}_p} \frac{\partial \boldsymbol{\Sigma}_p}{\partial \mathbf{p}_i} \tag{4.16}$$

where the operations on the right-hand side result in one-dimensional tensors that are consistent with those on the left-hand side. The following properties are employed to calculated the derivatives[65]

$$\frac{\partial}{\partial a} \det \boldsymbol{A} = \operatorname{tr}\left(\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial a} \det \boldsymbol{A} \right) \det \boldsymbol{A} \tag{4.17a}$$

$$\frac{\partial \boldsymbol{A}^{-1}}{\partial a} = -\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial a} \boldsymbol{A}^{-1} \tag{4.17b}$$

$$\operatorname{tr}(\boldsymbol{A}\boldsymbol{B}) = \operatorname{tr}(\boldsymbol{B}\boldsymbol{A}) \tag{4.17c}$$

for matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and the scalar a.

From (4.9)

$$\begin{aligned} \frac{\partial D}{\partial \boldsymbol{\mu}_p} &= \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\mu}_p} \left(\left(\boldsymbol{\mu}_d - \boldsymbol{\mu}_p \right) \boldsymbol{\Sigma}_d^{-1} \left(\boldsymbol{\mu}_d - \boldsymbol{\mu}_p \right) \right) \\ &= -\boldsymbol{\Sigma}_d^{-1} \left(\boldsymbol{\mu}_d - \boldsymbol{\mu}_p \right) \end{aligned} \tag{4.18}$$

Moreover

$$\frac{\partial D}{\partial \boldsymbol{\Sigma}_p} = \frac{1}{2} \frac{\partial}{\partial \boldsymbol{\Sigma}_p} \left(\operatorname{tr}\left(\boldsymbol{\Sigma}_d^{-1} \boldsymbol{\Sigma}_p \right) + \ln \left(\frac{\det \boldsymbol{\Sigma}_d}{\det \boldsymbol{\Sigma}_p} \right) \right) \tag{4.19}$$

By operating component-wise as $\partial \operatorname{tr}\left(\boldsymbol{\Sigma}_d^{-1} \boldsymbol{\Sigma}_p \right) / \partial \Sigma_{p_{ij}}, i, j = 1, 2$, we have $\partial \operatorname{tr}\left(\boldsymbol{\Sigma}_d^{-1} \boldsymbol{\Sigma}_p \right) / \partial \boldsymbol{\Sigma}_p =$

$\boldsymbol{\Sigma}_d^{-1}$. For the second term, we have

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_p} \ln\left(\frac{\det \boldsymbol{\Sigma}_d}{\det \boldsymbol{\Sigma}_p}\right) = -\frac{1}{\det \boldsymbol{\Sigma}_p} \frac{\partial}{\partial \boldsymbol{\Sigma}_p} \det \boldsymbol{\Sigma}_p = -\boldsymbol{\Sigma}_p \tag{4.20}$$

after applying (4.17a) component-wise with $a \equiv \Sigma_{p_{ij}}$, $i, j = 1, 2$. To summarize

$$\frac{\partial D}{\partial \mathbf{p}_i} = -\boldsymbol{\Sigma}_d^{-1}\left(\boldsymbol{\mu}_d - \boldsymbol{\mu}_p\right) \frac{\partial \boldsymbol{\mu}_p}{\partial \mathbf{p}_i} + \frac{1}{2}\left(\boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_p^{-1}\right) \frac{\partial \boldsymbol{\Sigma}_p}{\partial \mathbf{p}_i} \tag{4.21}$$

To complete the analysis, we use the definitions (4.3), to obtain $\partial \boldsymbol{\mu}_p / \partial \mathbf{p}_i = \boldsymbol{I}_2 / n$ for all i, where $\boldsymbol{I}_2$ is the identity matrix of order 2. For the covariance

$$\begin{aligned}
\frac{\partial \boldsymbol{\Sigma}_p}{\partial \mathbf{p}_i} &= \frac{1}{n-1} \sum_j \frac{\partial}{\partial \mathbf{p}_i} \left(\mathbf{p}_j - \boldsymbol{\mu}_p\right) \left(\mathbf{p}_j - \boldsymbol{\mu}_p\right)^\top \\
&= \frac{2}{n-1} \sum_j \left(\mathbf{p}_j - \boldsymbol{\mu}_p\right) \frac{\partial}{\partial \mathbf{p}_i} \left(\mathbf{p}_j - \boldsymbol{\mu}_p\right)^\top \\
&= \frac{2}{n-1} \sum_j \left(\mathbf{p}_j - \boldsymbol{\mu}_p\right) \left(\delta_{ij} - \frac{1}{n}\right) \boldsymbol{I}_2 \\
&= \frac{2}{n-1} \left(\mathbf{p}_i - \boldsymbol{\mu}_p\right)
\end{aligned} \tag{4.22}$$

where δ_{ij} is the Kronecker delta. By putting together all the expressions we have

$$\frac{\partial D}{\partial \mathbf{p}_i} = \frac{1}{n-1}\left(\boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_P^{-1}\right)\left(\mathbf{p}_i - \boldsymbol{\mu}_P\right) - \frac{1}{n}\boldsymbol{\Sigma}_d^{-1}\left(\boldsymbol{\mu}_d - \boldsymbol{\mu}_P\right) \tag{4.23}$$

It is worth noting that the expression of the gradient of KL divergence with respect to agent $\mathbf{p}_i$ includes error terms for the moments of the distributions, corresponding to the action of driving the agents' density to the target one in the sense of moments. To obtain the stationary points of the KL divergence D, we introduce

$$\mathbf{c}_i^D = \frac{n-1}{n}\boldsymbol{\mu}_d + \frac{1}{n}\boldsymbol{\mu}_P - \boldsymbol{\Sigma}_d\boldsymbol{\Sigma}_P^{-1}\boldsymbol{\mu}_P + \boldsymbol{\Sigma}_d\boldsymbol{\Sigma}_P^{-1}\mathbf{p}_i \tag{4.24}$$

the expression equation (4.23) can be written as

$$\frac{\partial D}{\partial \mathbf{p}_i} = -\frac{1}{n-1}\boldsymbol{\Sigma}_d^{-1}(\mathbf{c}_i^D - \mathbf{p}_i) \tag{4.25}$$

which has the samilar structure as equation (4.15).

Based on this, the gradient of the performance index defined in equation (4.11) is written as

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{p}_i} &= \alpha\frac{\partial H}{\partial \mathbf{p}_i} + \beta\frac{\partial D}{\partial \mathbf{p}_i} \\ &= \alpha m_i^H\left(\mathbf{c}_i^H - \mathbf{p}_i\right) - \frac{\beta\boldsymbol{\Sigma}_d^{-1}}{n-1}\left(\mathbf{c}_i^D - \mathbf{p}_i\right) \\ &= \boldsymbol{W}_i\left(\boldsymbol{W}_i^{-1}\mathbf{c}_i^{HD} - \mathbf{p}_i\right) \\ &= \boldsymbol{W}_i\left(\tilde{\mathbf{c}}_i - \mathbf{p}_i\right) \end{aligned} \tag{4.26}$$

where the weight matrix $\boldsymbol{W}_i$ and the quantities $\mathbf{c}_i^{HD}$ and $\tilde{\mathbf{c}}_i$ are given by

$$\boldsymbol{W}_i = \alpha m_i^H\mathbf{I}_2 - \beta\left(\frac{1}{n-1}\right)\boldsymbol{\Sigma}_d^{-1} \tag{4.27a}$$

$$\mathbf{c}_i^{HD} = \alpha m_i^H\mathbf{I}_2\mathbf{c}_i^H - \beta\left(\frac{1}{n-1}\right)\boldsymbol{\Sigma}_d^{-1}\mathbf{c}_i^D \tag{4.27b}$$

$$\tilde{\mathbf{c}}_i = \boldsymbol{W}_i^{-1}\mathbf{c}_i^{HD} \tag{4.27c}$$

Note that the point $\tilde{\mathbf{c}}_i$ includes the weights α and β.

4.1.5 Gradient Descent Law for Time-invariant Target Density

This subsection presents our previous work in [1] for a static target density ϕ_d. We propose the following dissipative controls derived from the gradients of J with respect to the agents' trajectories to generate trajectories that drive the formation to a minimum of the performance index (4.11),

$$\mathbf{u}_i = -k\frac{\partial J}{\partial \mathbf{p}_i} = -k\boldsymbol{W}_i\left(\tilde{\mathbf{c}}_i - \mathbf{p}_i\right) \tag{4.28}$$

with $k > 0$. Note that if we set $\alpha > 0, \beta = 0$, controller (4.28) reduces to the area coverage control in Lloyd's algorithm. On the other hand, if $\alpha = 0, \beta > 0$, (4.28) becomes a controller that only minimizes the KL-divergence metric in (4.9) by driving the agents' density moments to the desired ones. The intermediate cases with dominant α vs β induce stationary points closer to the centroidal Voronoi configuration $\boldsymbol{c}^H$, whereas dominant β results in stationary points closed to $\boldsymbol{c}^D$,

with the main effect being the minimization of the errors in the moments of the distributions ϕ_P and ϕ_d.

The following Lyapunov argument with Lyapunov function $V = J$ is employed to show that the autonomous control law (4.28) drives the system to a configuration that minimizes the multi-objective performance index J. The minimizing agents' formations are weighted combinations of centroidal Voronoi configurations generating spatial distributions with statistics that tend to the desired, driving one. The result is formalized in the following

Proposition 1. *Consider the optimal control problem* (4.11) *with kinematic agents* (4.1). *Assume that each agent knows the state of the respective spatial neighbours, so that the Voronoi tessellation* $\mathcal{V}_i$ *in* (3.2) *and the moments* $\boldsymbol{\mu}_P$ *and* $\boldsymbol{\Sigma}_P$ *in* (4.3) *can be computed. Then the trajectories generated by the controls* (4.28) *asymptotically drive the agents to the set of critical points* $\mathbf{p}_i = \tilde{\mathbf{c}}_i$, $i = 1, \dots, n$.

Proof. The time derivative along the trajectories generated by controls (4.28) gives

$$\dot{V} = \dot{J} = \sum_{i=1}^{n} \frac{\partial J}{\partial \mathbf{p}_i} \cdot \dot{\mathbf{p}}_i = -k \sum_{i=1}^{n} \left\| \boldsymbol{W}_i \left(\tilde{\mathbf{c}}_i - \mathbf{p}_i \right) \right\|^2 \tag{4.29}$$

Then $\dot{V} \leq 0$. In addition, the weight matrix (4.27a) is negative definite because the generalized mass $m_i^H < 0$ and $\boldsymbol{\Sigma}_d^{-1}$ is positive definite. Therefore, by virtue of LaSalle's invariance principle [58], the trajectories converge to the largest invariant set of $\dot{V} = \left\| \boldsymbol{W}_i \left(\tilde{\mathbf{c}}_i - \mathbf{p}_i \right) \right\|^2$ which is $\mathbf{p}_i = \tilde{\mathbf{c}}_i$, $i = 1, \dots, n$. Since these are invariant points of the controls (4.28), the agents asymptotically converge to the set of critical points $\mathbf{p}_i = \tilde{\mathbf{c}}_i$. □

Based on the definition of $\tilde{\mathbf{c}}_i$ in (4.27c), the invariant set admits different configurations that combine centroidal Voronoi configurations and stationary points of the KL divergence in terms of the moments of the the agents' distribution, weighted by α and β in the performance index J in (4.11).

4.1.6 Non-autonomous State-feedback Law

In this section, the statistical moments of the target density are allowed to be time-varying, hence the control problem becomes non-autonomous due to the drift introduced by the time-varying target density.

We propose a distributed non-autonomous feedback law that asymptotically drives the agents to the trajectories of their respective time-varying stationary points

$\tilde{\mathbf{c}}_i$ while minimizing J. The structure is the same as the one proposed in [40] in the context of non-autonomous area coverage, and it is extended here to the case of multi-objective optimization. The velocity control input for the i-th kinematic agent is given by

$$\mathbf{u}_i = \frac{\partial J/\partial \mathbf{p}_i}{\|\partial J/\partial \mathbf{p}_i\|^2}\left(-k\,\|\tilde{\mathbf{c}}_i - \mathbf{p}_i\|^2 - \lambda_i \frac{\partial J}{\partial t}\right) \tag{4.30}$$

where $k > 0$ is a control gain, and the time partial derivative of J compensates the drift introduced by the evolution of the target density. The scalar λ_i scales the drift term according to

$$\lambda_i = \frac{\|\partial J/\partial \mathbf{p}_i\|^2}{\sum_{i=1}^n \|\partial J/\partial \mathbf{p}_i\|^2} = \frac{\|\boldsymbol{W}_i\,(\tilde{\mathbf{c}}_i - \mathbf{p}_i)\|^2}{\sum_{i=1}^n \|\boldsymbol{W}_i\,(\tilde{\mathbf{c}}_i - \mathbf{p}_i)\|^2} \tag{4.31}$$

Using the chain rule, the time partial $\frac{\partial J}{\partial t}$ can be written as

$$\frac{\partial J}{\partial t} = \alpha \frac{\partial H}{\partial t} + \beta \frac{\partial D}{\partial t} \tag{4.32a}$$

$$\begin{aligned}\frac{\partial H}{\partial t} = \sum_{i=1}^n \int_{\mathcal{V}_i} f(r_i)\phi_d \Bigg(\boldsymbol{\Sigma}_d^{-1}(\mathbf{q}-\boldsymbol{\mu}_d)\cdot\frac{\partial \boldsymbol{\mu}_d}{\partial t} \\ +\frac{1}{2}\,\mathrm{tr}\left(\boldsymbol{\Sigma}_d^{-1}(\mathbf{q}-\boldsymbol{\mu}_d)(\mathbf{q}-\boldsymbol{\mu}_d)^\top\boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_d^{-1}\right)\frac{\partial \boldsymbol{\Sigma}_d}{\partial t}\Bigg)\,d\mathbf{q}\end{aligned} \tag{4.32b}$$

$$\begin{aligned}\frac{\partial D}{\partial t} = \mathrm{tr}\Bigg(\left(\boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_d^{-1}(\mathbf{q}-\boldsymbol{\mu}_d)(\mathbf{q}-\boldsymbol{\mu}_d)^\top\boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_d^{-1}\boldsymbol{\Sigma}_P\boldsymbol{\Sigma}_d^{-1}\right)\frac{\partial \boldsymbol{\Sigma}_d}{\partial t}\Bigg) \\ +\boldsymbol{\Sigma}_d^{-1}(\boldsymbol{\mu}_d - \boldsymbol{\mu}_P)\cdot\frac{\partial \boldsymbol{\mu}_d}{\partial t}\end{aligned} \tag{4.32c}$$

Here we show the derivation of equation (4.32b) and (4.32c). As the coverage metric H depends on time t only through the time-varying density ϕ, $\partial H/\partial t$ is given by

$$\frac{\partial H}{\partial t} = \sum_{i=1}^n \int_{\mathcal{V}_i} f\,(r_i)\,\frac{\partial}{\partial t}\phi_d(\mathbf{q}, t)\,d\mathbf{q} \tag{4.33}$$

where, the time-partial derivative of the target density $\frac{\partial}{\partial t}\phi_d(\mathbf{q},t)$ is

$$\frac{\partial}{\partial t}\phi_d(\mathbf{q},t) = \frac{\partial \phi_d}{\partial \boldsymbol{\mu}_d}\cdot\frac{\partial \boldsymbol{\mu}_d}{\partial t} + \operatorname{tr}\left((\frac{\partial \phi_d}{\partial \boldsymbol{\Sigma}_d})^\top \frac{\partial \boldsymbol{\Sigma}_d}{\partial t}\right) \tag{4.34}$$

where, because ϕ_d is defined as a Gaussian density in an exponential form, the partial derivative of ϕ_d with respect to the moments yields

$$\frac{\partial \phi_d}{\partial \boldsymbol{\mu}_d} = \phi_d \boldsymbol{\Sigma}_d^{-1}(\mathbf{q}-\boldsymbol{\mu}_d) \tag{4.35a}$$

$$\frac{\partial \phi_d}{\partial \boldsymbol{\Sigma}_d} = \frac{1}{2}\phi_d \operatorname{tr}\left(\boldsymbol{\Sigma}_d^{-1}(\mathbf{q}-\boldsymbol{\mu}_d)(\mathbf{q}-\boldsymbol{\mu}_d)^\top\boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_d^{-1}\right) \tag{4.35b}$$

Note that the covariance matrix is symmetric, hence $(\frac{\partial \phi_d}{\partial \boldsymbol{\Sigma}_d})^\top = \frac{\partial \phi_d}{\partial \boldsymbol{\Sigma}_d}$. Therefore, the substitution of equation (4.35) back into equation (4.35) is the time partial derivative of the coverage metric $\frac{\partial H}{\partial t}$ in equation (4.32b).

For the partial derivative of KL divergence with respect to time $\partial D/\partial t$, the chain rule gives

$$\frac{\partial D}{\partial t} = \frac{\partial D}{\partial \boldsymbol{\mu}_d}\cdot\frac{\partial \boldsymbol{\mu}_d}{\partial t} + \operatorname{tr}\left((\frac{\partial D}{\partial \boldsymbol{\Sigma}_d})^\top \frac{\partial \boldsymbol{\Sigma}_d}{\partial t}\right) \tag{4.36}$$

similar to the derivation for the gradient of KL divergence to an agent's position in section 4.1.4, we have

$$\begin{aligned}\frac{\partial D}{\partial \boldsymbol{\mu}_d} = \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\mu}_d}\left(\left(\boldsymbol{\mu}_d-\boldsymbol{\mu}_p\right)\boldsymbol{\Sigma}_d^{-1}\left(\boldsymbol{\mu}_d-\boldsymbol{\mu}_p\right)\right) \\ = \boldsymbol{\Sigma}_d^{-1}(\boldsymbol{\mu}_d-\boldsymbol{\mu}_P)\end{aligned} \tag{4.37a}$$

$$\frac{\partial D}{\partial \boldsymbol{\Sigma}_d} = \boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_d^{-1}(\mathbf{q}-\boldsymbol{\mu}_d)(\mathbf{q}-\boldsymbol{\mu}_d)^\top\boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_d^{-1}\boldsymbol{\Sigma}_P\boldsymbol{\Sigma}_d^{-1} \tag{4.37b}$$

The ratio of α and β determines the spatial distribution of the time-varying stationary points $\tilde{\mathbf{c}}$ closer to the centroidal Voronoi configuration or the distribution of $\mathbf{c}^D$ that minimize the difference between the statistical moments of ϕ_P and ϕ_d. If we set $\alpha = 0, \beta > 0$, then $\tilde{\mathbf{c}}_i = \mathbf{c}_i^D$, and equation (4.30) reduces into a controller that only minimizes the KL-Divergence metric. On the contrary, when $\alpha > 0, \beta = 0$, equation (4.30) becomes the generalized non-autonomous coverage controller

proposed in [40].

As the system is non-autonomous, the agents converge to the trajectories generated by the evolution of the stationary point $\tilde{\mathbf{c}}_i$ rather than a set of static points. The formal proof of asymptotic stability is presented in the following

Lemma 1. *Consider the dynamics of agents equation* (4.1) *and the sensing performance function equation* (3.1)*, the non-autonomous state-feedback law equation* (4.30) *minimizes the multi-objective mixed metric defined in equation* (4.11)*.*

Proof. Take the derivative of equation (4.11) along time, we have

$$\dot{J} = \sum_{i=1}^{n} \frac{\partial J}{\partial \mathbf{p}_i} \cdot \dot{\mathbf{p}}_i + \frac{\partial J}{\partial t} \tag{4.38}$$

substituting equation (4.30) into equation (4.38) yields

$$\dot{J} = -k \sum_{i=1}^{n} \|\tilde{\mathbf{c}}_i - \mathbf{p}_i\|^2 \leq 0 \tag{4.39}$$

therefore, the non-autonomous mixed metric J monotonically decreases under the controls (4.30). □

Lemma 1 guarantees that the distributed feedback law in equation (4.30) minimizes the multi-objective index (4.11). In addition, we have to prove the agents asymptotically converge to the trajectories generated by their respective time-varying generalized centroids $\tilde{\mathbf{c}}_i$. Given the non-autonomous structure of the problem, we use Barbalat's lemma

Lemma 2 (Barbalat [60])**.** *If there exists a non-autonomous Lyapunov function $V(\mathbf{P}, t)$ satisfies the following properties:*

1. *$V(\mathbf{P}, t)$ is lower bounded,*
2. *$\dot{V}(\mathbf{P}, t)$ is negative semi-definite,*
3. *$\dot{V}(\mathbf{P}, t)$ is uniformly continuous in time.*

then $\lim_{t\to\infty} \dot{V}(\mathbf{P}, t) = 0$.

Now, we can state the convergence result.

Proposition 2. *Consider the set of mobile agents equation (4.1) under the action of the non-autonomous feedback laws defined in equation (4.30). Assume that the communication network among the agents is fully connected so that the states $\mathbf{p}_i$ of all the agents are shared in the group and the collective density equation (4.7) can be estimated. Then the trajectories of agents asymptotically converge to the trajectories of the points $\tilde{\mathbf{c}}_i$, i.e.* $\lim_{t\to\infty} \|\tilde{\mathbf{c}}_i - \mathbf{p}_i\| = 0$.

Proof. Consider the candidate Lyapunov function

$$V = J \tag{4.40}$$

Proposition 2 holds if equation (4.40) satisfies all the conditions in Lemma 2. To this end, we note that

1. The coverage metric H is positive because both the target density function equation (4.7) and the sensing performance function equation (3.1) are strictly positive. Moreover, the KL-Divergence D is non-negative . Hence $V(\mathbf{P}, t) = J > 0$.

2. $\dot{V}(\mathbf{P}, t) \leq 0$ from the calculation of $\dot{J}$ in equation (4.39) is negative semi-definite.

3. To verify that the last condition of Lemma 2 is satisfied, we need to prove that $\ddot{V}$ is finite. Substituting from (4.39), we have

$$\begin{aligned}\ddot{V} = &-\sum_{i=1}^{n}\sum_{j=1}^{n}\left(2k\left(\tilde{\mathbf{c}}_i - \mathbf{p}_i\right)^\top\left(\frac{\partial\tilde{\mathbf{c}}_i}{\partial\mathbf{p}_j} - \mathbf{I}_2\delta_{ij}\right)\mathbf{u}_j\right) \\ &-\sum_{i=1}^{n}\left(2k\left(\tilde{\mathbf{c}}_i - \mathbf{p}_i\right)^\top\frac{\partial\tilde{\mathbf{c}}_i}{\partial t}\right)\end{aligned} \tag{4.41}$$

   where $\mathbf{I}_2$ is the 2- identity matrix and δ_{ij} is the Kronecker delta. Equation (4.30) implies that the term $\mathbf{u}_j$ is finite as long as the performance index J in (4.11) is finite, which is true because both the agents and the time-varying density are distributed in a bounded region $\boldsymbol{\Omega}$, hence H and D are finite [28]. Following the same argument, the term $(\tilde{\mathbf{c}}_i - \mathbf{p}_j)$ is finite [28]. Expanding the

term $\frac{\partial \mathbf{c}_i^D}{\partial \mathbf{p}_j}$ we obtain

$$\begin{aligned}\frac{\partial \mathbf{c}_i^D}{\partial \mathbf{p}_j} = -\alpha \left(\boldsymbol{W}_i^{-1} \left(\tilde{\mathbf{c}}_i \frac{\partial m_i^H}{\partial \mathbf{p}_j^\top} - \frac{\partial (m_i^H \mathbf{c}_i^H)}{\partial \mathbf{p}_j^\top} \right) \right) \\ -\frac{\beta}{n-1} \Bigg(\left(\mathbf{p}_j - \boldsymbol{\mu}_P\right)^\top \boldsymbol{\Sigma}_P^{-1} \left(\mathbf{p}_j - \boldsymbol{\mu}_P\right) \otimes \left(\boldsymbol{\Sigma}_d^{-1} \boldsymbol{\Sigma}_P^{-1} \right) \\ + \frac{1}{n} \boldsymbol{\Sigma}_d^{-1} + \frac{n-1}{n} \boldsymbol{\Sigma}_d^{-1} \boldsymbol{\Sigma}_P^{-1} \Bigg)\end{aligned} \tag{4.42}$$

where $\otimes$ is the Kronecker product. The terms $(\mathbf{p}_j - \boldsymbol{\mu}_P)$ and $\boldsymbol{\Sigma}_p^{-1}$ are finite as the workspace is bounded, and $\boldsymbol{\Sigma}_d$ is also finite as long as the target time-varying density ϕ_d evolves with finite speed in a bounded region. Moreover, $\frac{\partial m_i^H}{\partial \mathbf{p}_j^\top}$ and $\frac{\partial (m_i^H \mathbf{c}_i^H)}{\partial \mathbf{p}_j^\top}$ are finite as established by [39]. To complete the analysis, the partial time derivative $\frac{\partial \tilde{\mathbf{c}}_i}{\partial t}$ yields

$$\frac{\partial \tilde{\mathbf{c}}_i}{\partial t} = \frac{\partial \boldsymbol{W}_i^{-1}}{\partial t} \mathbf{c}_i^{HD} + \boldsymbol{W}_i^{-1} \frac{\partial \mathbf{c}_i^{HD}}{\partial t} \tag{4.43}$$

By using the chain rule, we have

$$\frac{\partial \boldsymbol{W}_i^{-1}}{\partial t} = \alpha \frac{\partial m_i^H}{\partial t} + \frac{\beta}{n-1} \boldsymbol{\Sigma}_d^{-1} \frac{\partial \boldsymbol{\Sigma}_d}{\partial t} \boldsymbol{\Sigma}_d^{-1} \boldsymbol{\mu}_d \tag{4.44a}$$

$$\begin{aligned}\frac{\partial \mathbf{c}_i^{HD}}{\partial t} = \alpha \frac{\partial m_i^H \mathbf{c}_i^H}{\partial t} - \frac{\beta}{n-1} \Bigg(\frac{n-1}{n} \boldsymbol{\Sigma}_d^{-1} \frac{\partial \boldsymbol{\Sigma}_d}{\partial t} \boldsymbol{\Sigma}_d^{-1} \boldsymbol{\mu}_d \\ + \frac{n-1}{n} \boldsymbol{\Sigma}_d^{-1} \frac{\partial \boldsymbol{\mu}_d}{\partial t} + \frac{1}{n} \boldsymbol{\Sigma}_d^{-1} \frac{\partial \boldsymbol{\Sigma}_d}{\partial t} \boldsymbol{\Sigma}_d^{-1} \boldsymbol{\mu}_P \Bigg)\end{aligned} \tag{4.44b}$$

where the time partial derivative in equation (4.44a) and equation (4.44b) of the terms related to the Voronoi mass and centroids are expressed as

$$\frac{\partial m_i^H}{\partial t} = \int_{\mathcal{V}_i} \frac{\partial \phi_i^H}{\partial t} \, d\mathbf{q} = -2 \int_{\mathcal{V}_i} \frac{\partial \phi_d}{\partial t} \, d\mathbf{q} \tag{4.45a}$$

$$\frac{\partial (m_i^H \mathbf{c}_i^H)}{\partial t} = \int_{\mathcal{V}_i} q \frac{\partial \phi_i^H}{\partial t} \, d\mathbf{q} = -2 \int_{\mathcal{V}_i} q \frac{\partial \phi_d}{\partial t} \, d\mathbf{q} \tag{4.45b}$$

where the time partial derivative of ϕ_d is given by

$$\frac{\partial \phi_d}{\partial t} = \phi_d \left(\boldsymbol{\Sigma}_d^{-1} (\mathbf{q} - \boldsymbol{\mu}_d) \cdot \frac{\partial \boldsymbol{\mu}_d}{\partial t} + \frac{1}{2} \text{tr} \left(\left(\boldsymbol{\Sigma}_d^{-1} (\mathbf{q} - \boldsymbol{\mu}_d) (\mathbf{q} - \boldsymbol{\mu}_d)^\top \boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_d^{-1} \right) \frac{\partial \boldsymbol{\Sigma}_d}{\partial t} \right) \right) \tag{4.45c}$$

where the time partial derivative in equation (4.44a) and equation (4.44b) of the terms related to the Voronoi mass and centroids are expressed as

$$\frac{\partial m_i^H}{\partial t} = \int_{\mathcal{V}_i} \frac{\partial \phi_i^H}{\partial t} \, d\mathbf{q} = -2 \int_{\mathcal{V}_i} \frac{\partial \phi_d}{\partial t} \, d\mathbf{q} \tag{4.46a}$$

$$\frac{\partial (m_i^H \mathbf{c}_i^H)}{\partial t} = \int_{\mathcal{V}_i} q \frac{\partial \phi_i^H}{\partial t} \, d\mathbf{q} = -2 \int_{\mathcal{V}_i} q \frac{\partial \phi_d}{\partial t} \, d\mathbf{q} \tag{4.46b}$$

where the time partial derivative of ϕ_d is given by

$$\frac{\partial \phi_d}{\partial t} = \phi_d \left(\boldsymbol{\Sigma}_d^{-1} (\mathbf{q} - \boldsymbol{\mu}_d) \cdot \frac{\partial \boldsymbol{\mu}_d}{\partial t} + \frac{1}{2} \text{tr} \left(\left(\boldsymbol{\Sigma}_d^{-1} (\mathbf{q} - \boldsymbol{\mu}_d) (\mathbf{q} - \boldsymbol{\mu}_d)^\top \boldsymbol{\Sigma}_d^{-1} - \boldsymbol{\Sigma}_d^{-1} \right) \frac{\partial \boldsymbol{\Sigma}_d}{\partial t} \right) \right) \tag{4.46c}$$

It can be seen that the finiteness of $\frac{\partial \tilde{\mathbf{c}}_i}{\partial t}$ is determined by the rates $\frac{\partial \boldsymbol{\mu}_d}{\partial t}$ and $\frac{\partial \boldsymbol{\Sigma}_d}{\partial t}$ of the statistical moments of the target time-varying density, which are finite in this work. Therefore, $\frac{\partial \tilde{\mathbf{c}}_i}{\partial t}$ is finite.

From this analysis it follows that all terms in (4.41) are finite, and therefore $\ddot{V}$ is bounded and the third condition of Lemma 2 is satisfied. Since the candidate Lyapunov function V satisfies all the conditions of Barbalat's lemma, it follows that

$$\lim_{t \to \infty} \dot{V}(\mathbf{P}, t) = - \lim_{t \to \infty} k \sum_{i=1}^{n} \| \tilde{\mathbf{c}}_i - \mathbf{p}_i \|^2 = 0 \tag{4.47}$$

which implies that the agents asymptotically converge to the trajectories of $\tilde{\mathbf{c}}_i$. □

4.2 Summary

This chapter provides the formal problem formulation, control solution and stability analysis. The control problem is formulated as a multi-objective optimization, where the performance index is defined as a convex combination of area coverage metric H and KL divergence$D$. In the autonomous case with time-invariant target density, the proposed control (4.28) is proven to minimize the multi-objective performance index J and drive the agents to the largest invariant set of stationary points $\tilde{c}$. When the target density is allowed to be time-varying, the control problem becomes non-autonomous. The non-autonomous control (4.28) minimizes the time-varying performance index, and the agents asymptotically converge to the trajectories of $\tilde{c}$

Chapter 5

Simulation Results and Discussion

In this chapter, the theoretical results of the feedback control laws (4.28) and (4.30) are validated via simulations implemented in Mathematica. The code is attached in the Appendix. Simulations are performed by considering a square-shaped planar area $\mathbf{\Omega}$ with sides of length 150 in a coherent system of units. In all the simulated scenarios, we consider a group of eight agents with initial states (33,17), (45,35), (54,10), (56,24), (23,30), (56,19), (49,13), (16,48), generating the initial statistical parameters of the agents' spatial density $\boldsymbol{\mu}_{P0} = (41.5, 24.5)$, $\sigma_{Px0} = 15.6$, $\sigma_{Py0} = 12.7$ and $\rho_{P0} = -0.68$, as shown in Fig. 5.1.

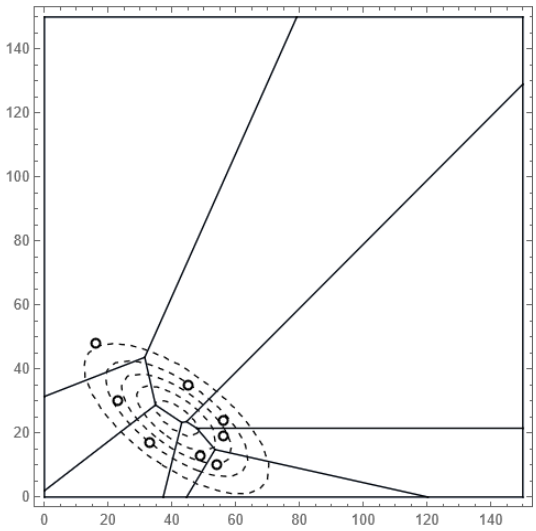


Figure 5.1: Initial states of the agents with the corresponding Voronoi tessellation and collective density.

5.1 Autonomous Gradient Descent Control for Time-invariant Density

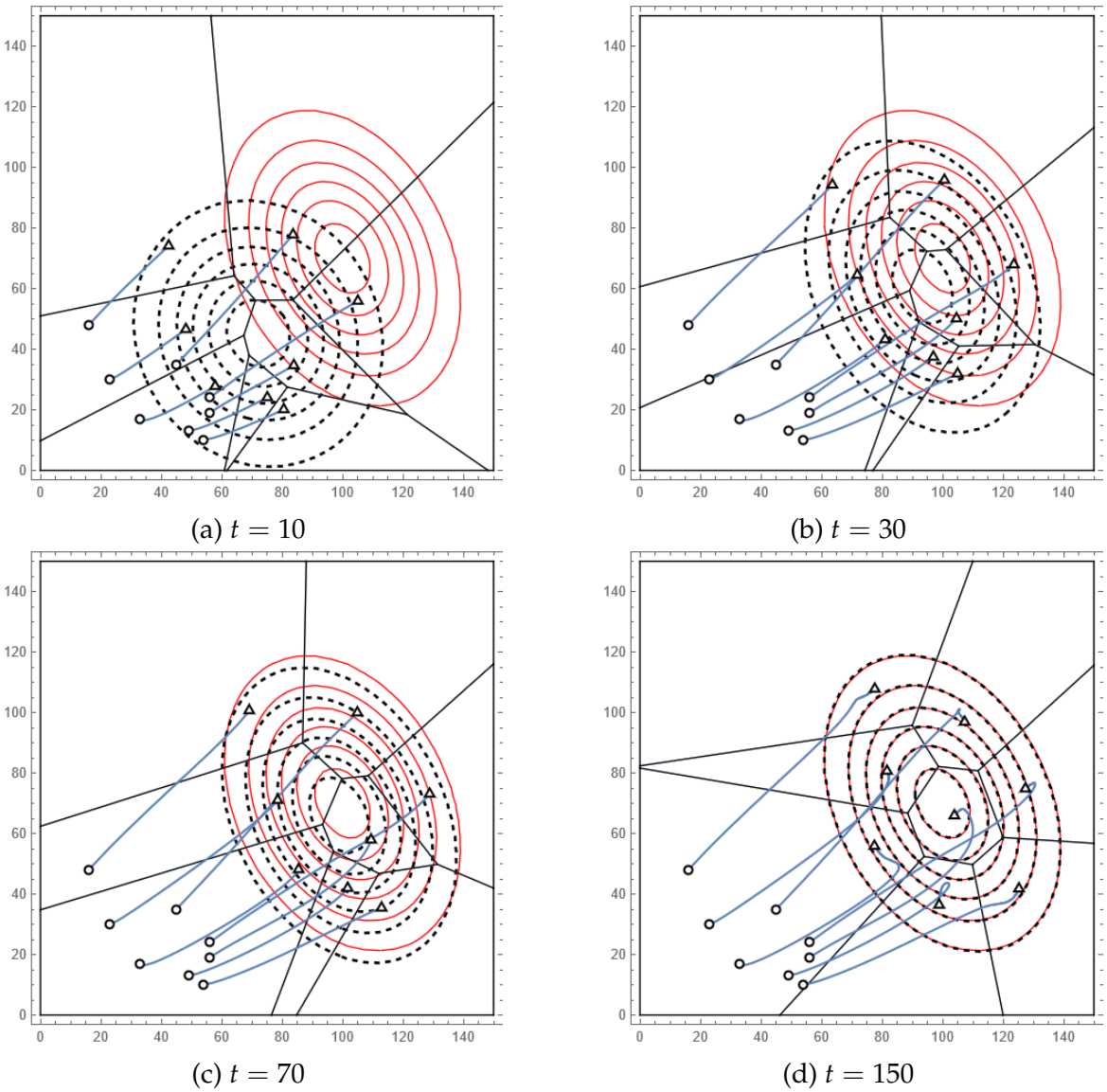
This section illustrates results for the autonomous case with control law in (4.28). The target density is stationary with mean $\boldsymbol{\mu}_d = (100, 70)$ and covariance matrix $\boldsymbol{\Sigma}_d = \begin{pmatrix} 400 & -150 \\ -150 & 625 \end{pmatrix}$. The control gain is $k = 0.15$, and the weights of the performance index J are $\alpha = 2.5$, $\beta = 5000$.

Fig. 5.2 depicts the agents' trajectories at different time instants, where the black hollow dots represent the initial states, and the red contours denotes the target density ϕ_d. The triangles indicate the current positions of the agents, with the corresponding density ϕ_P realized by the agents denoted by black dashed contours, with the corresponding Voronoi partition. It can be noted that the agents' formation reflect the combined control actions of tracking the target density while deploying into a centroidal Voronoi configuration. In addition, under the same choice of confidence interval, the contours of the realized density ϕ_P and of the desired density ϕ_d virtually overlap, as shown in figure 5.2d illustrating the tracking in the sense of statistical moments.

The performance indexes are shown in Fig. 5.3 in logarithmic scale, where the red curve is the total multi-objective index J, the blue curve is the locational index H and the black curve is the KL-divergence D, each scaled by their respective weights α and β. The three quantities are minimized with preference on the KL divergence due to the choice of the weights $\alpha << \beta$.

The tracking performance in the sense of density functions is illustrated in figure 5.4. It can be observed that the agents' statistical moments converge to the target ones in figures 5.4a and 5.4b. In addition, the Euclidean distance $\|\boldsymbol{\mu}_d - \boldsymbol{\mu}_P\|$ and Frobenius norm $\|\boldsymbol{\Sigma}_d - \boldsymbol{\Sigma}_P\|_F$ between the moments are shown in Fig. 5.4c in log scale. As expected, the feedback law (4.28) does not drive these errors to zero since the control actions are the weighted combination of (4.15) and (4.23), which combine the attractions towards centroidal Voronoi tessellations and towards distributions with the same moments as ϕ_d.

Fig. 5.5 shows the Euclidean distances between the agents and their respective critical points. It can be observed that the distances of the agents from the Voronoi centroids $\boldsymbol{c}^H$ in Fig. 5.5a are further than that from the critical points $\boldsymbol{c}^D$ in Fig. 5.5b(b). Under the prescribed large ratio β/α, the agents converge to a modified Voronoi configuration $\tilde{\boldsymbol{c}}$ that is biased towards the minimization of the errors



(a) $t = 10$ (b) $t = 30$

(c) $t = 70$ (d) $t = 150$

Figure 5.2: Snapshots of the autonomous controller (4.28) for $\alpha = 2.5$, $\beta = 5000$ and control gain $k = 0.15$. The target density is represented by red contours. The hollow dots denote the initial position of the agents. The triangles denote the agents' positions at each time instant, along with the corresponding collective density (dashed contours) and Voronoi diagram.

in the statistical moments as in Fig. 5.5c.

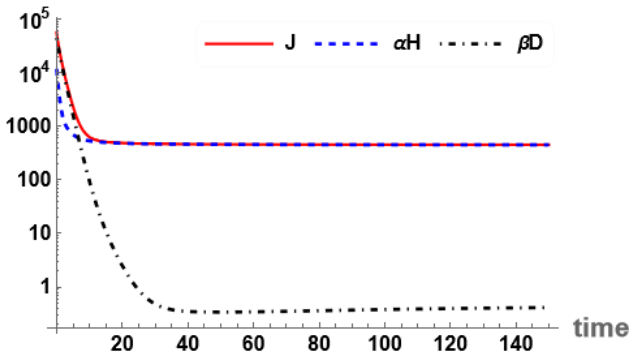
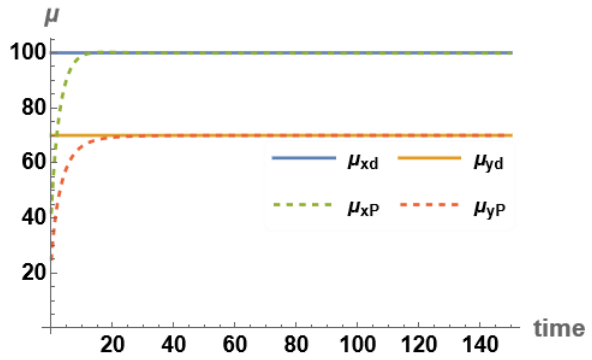
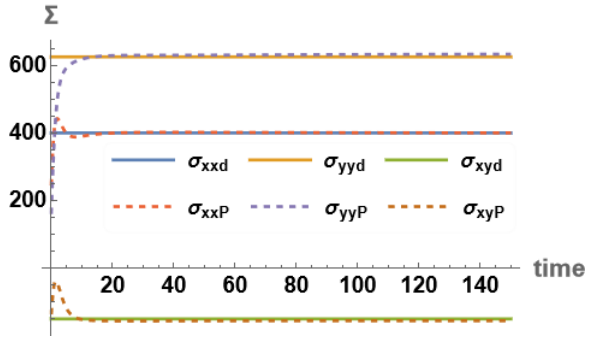


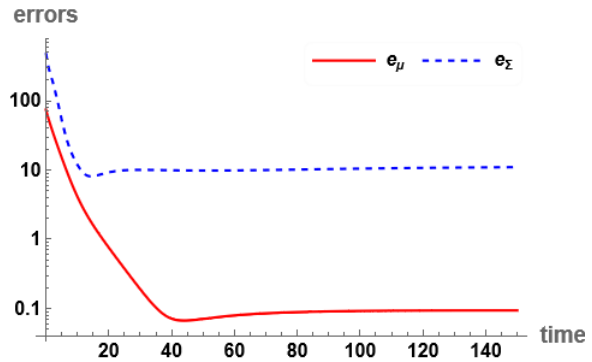
Figure 5.3: Performance indices of the autonomous optimal control 4.28 are shown in logarithmic scale, each sub-indices are scaled by their respective weights.



(a) Evolution of the mean μ_P of the agents' collective density, and reference μ_d.

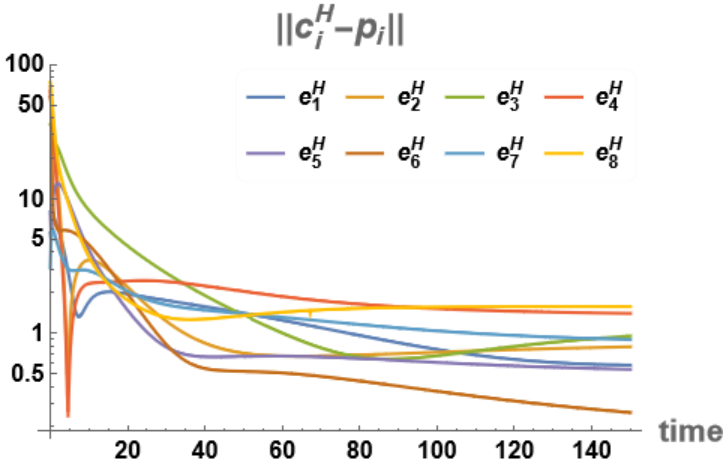


(b) Evolution of the components of the covariance matrix Σ_P of the agents' collective density, including standard deviation σ_{xxP}, σ_{yyP} and covariance σ_{xyP}, along with the conponents of the reference Σ_d

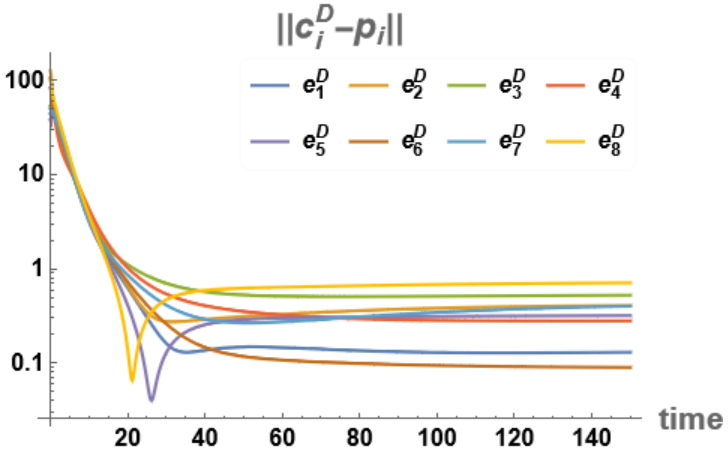


(c) Tracking errors in log scale, where $e_\mu = \|\mu_d - \mu_P\|$, $e_\Sigma = \|\Sigma_d - \mu_P\|_F$.

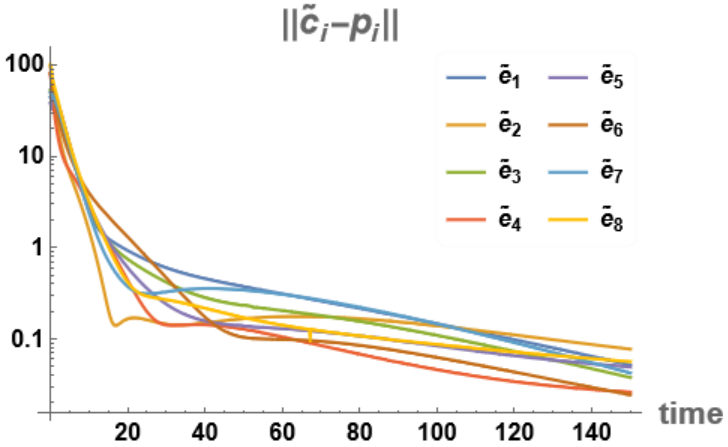
Figure 5.4: Tracking performance in the sense of statistical moments.



(a) Errors between the agents and their respective Voronoi centroids $\mathbf{c}^H$



(b) Errors between the agents and their respective stationary points of KL divergence $\mathbf{c}^D$



(c) Errors between the agents and their respective stationary points of total multi-objective performance index $\tilde{\mathbf{c}}$

Figure 5.5: Euclidean distance between the agents and the stationary points of H in eq.(4.8), D in eq.(4.9) and J in eq.(4.11) for $\alpha = 2.5$, $\beta = 5000$.

5.2 Non-autonomous Formation Control in time-varying environments

This section shows the simulation results for the proposed non-autonomous multi-objective formation control law (4.30), with a comparative simulation analysis of the impact of the sub-indices H and D, and an application of the non-autonomous law in a bounded region.

5.2.1 Impact of coverage metrics H and KL divergence D on Formation Performance: A Comparative Simulation Analysis

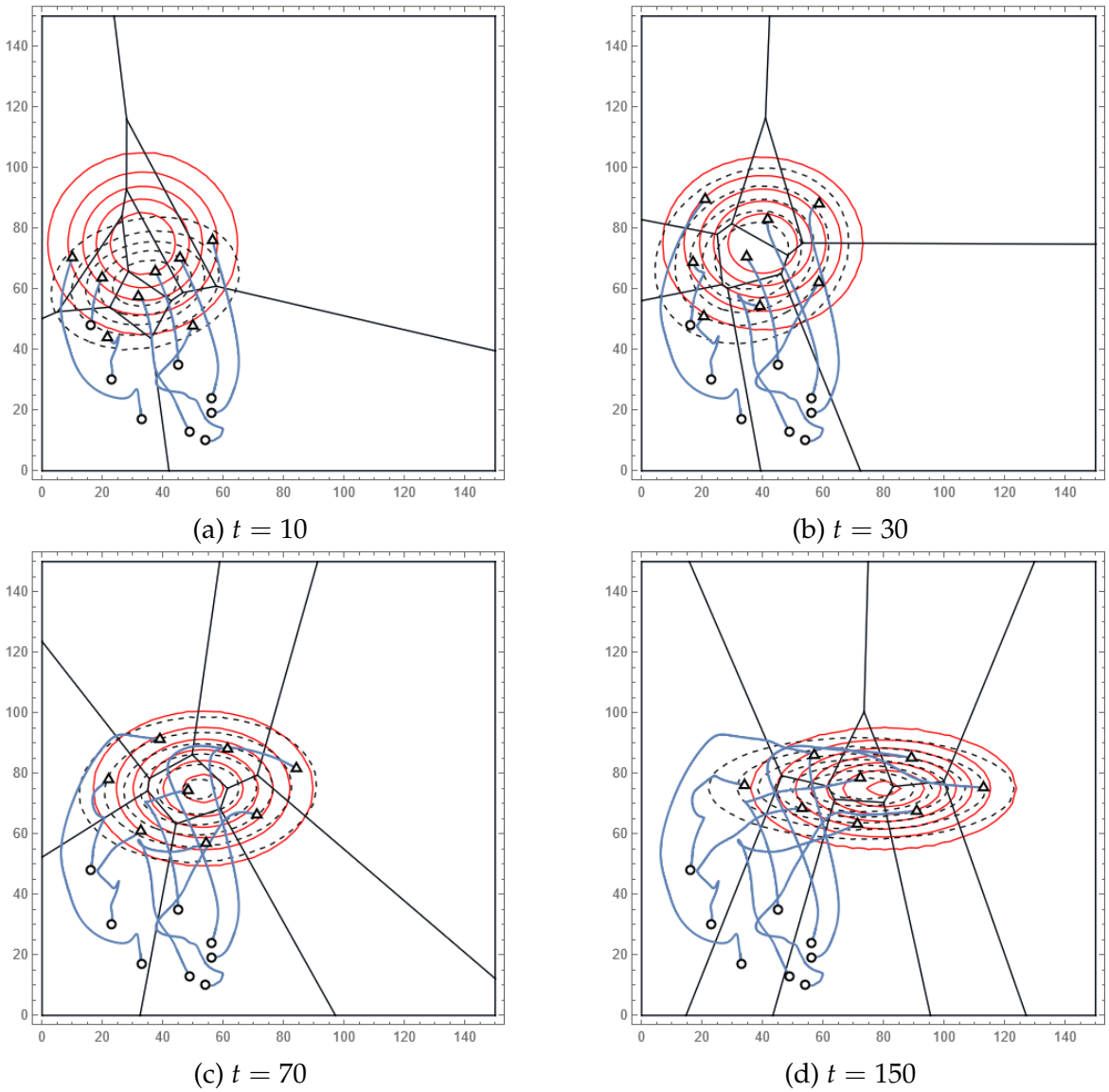
This subsection demonstrates the impact of H and D by setting their corresponding weights to zero in (4.11). Three simulations are performed with the proposed non-autonomous state feedback control law (4.30). For convenience, the setting $\alpha = 10, \beta = 0$ is referred to as coverage control, $\alpha = 0, \beta = 2 \times 10^4$ as KL control, and $\alpha = 10, \beta = 2 \times 10^4$ as multi-objective control.

Identical time-varying target density ϕ_d are employed for the three sets of simulations. Specifically, ϕ_d evolves according to its dynamic statistical properties $\boldsymbol{\mu}_d$ and $\boldsymbol{\Sigma}_d$. In a total simulation time $T = 150$, the mean position of the target density $\boldsymbol{\mu}_d$ shifts from $(30, 75)$ to $(80, 75)$ at a constant velocity, at the meantime, the standard deviations σ_x and σ_y evolves from 16 to 22 and 16 to 10 respectively at a constant rate, with a static correlation coefficient $\rho_d = 0$. This setting implies that the target multivariate Gaussian density is stretched along the x direction and compressed in the y direction, while its center moves uniformly along the positive direction of the x-axis.

The agents' trajectories for the three sets of simulations at different times are shown in figure 5.6, figure 5.9 and figure 5.12. The figures also present the time-varying target density in red contours, the agents' collective density in dashed contours and the Voronoi partition.

Figure 5.6 illustrates the scenario where only the coverage component is active ($\alpha = 10$, $\beta = 0$). In this case, the agents converge to a centroidal Voronoi configuration; however, this configuration does not guarantee tracking in the sense of statistical moments.

In contrast, figure 5.9 displays the performance of the KL term of the controller, where the agents' collective density (dashed contours) converges to the target time-varying density (red contours), but the configuration of agents does not converge



(a) $t = 10$ (b) $t = 30$

(c) $t = 70$ (d) $t = 150$

Figure 5.6: Snapshots of the area coverage performance for $\alpha = 10$, $\beta = 0$ and control gain $k = 0.5$. The hollow dots denote the initial position of the agents. The triangles denote the agents' positions at each time instant, along with the corresponding collective density (dashed contours) and Voronoi diagram.

to the centroidal Voronoi tessellation.

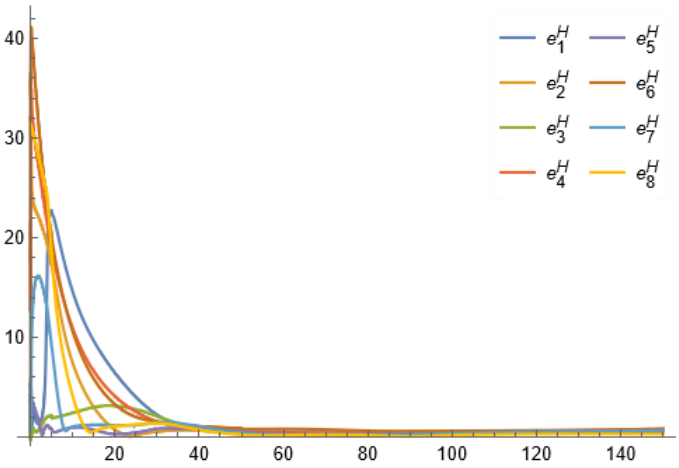
The combined configuration from the multi-objective controller is shown in Figure 5.12, where the agents form a centroidal Voronoi configuration and the collective density of agents converges to the target one. At the beginning of the simulation, the trajectories of the multi-objective controller in figures 5.12a and 5.12b resemble those of the KL controller in figures 5.9a and 5.9b. As the agents move into the region with higher target density, the agents converge to a centroidal Voronoi

tessellation, thus the trajectories generated by the multi-objective controller are more alike the one of the coverage controller, see the comparison of figure 5.12c 5.12d and figure 5.6c 5.6d. This is attributed to the increased mass m in the weight matrix $\boldsymbol{W}$ when the agents approach the target density, as outlined in equation (4.27a).

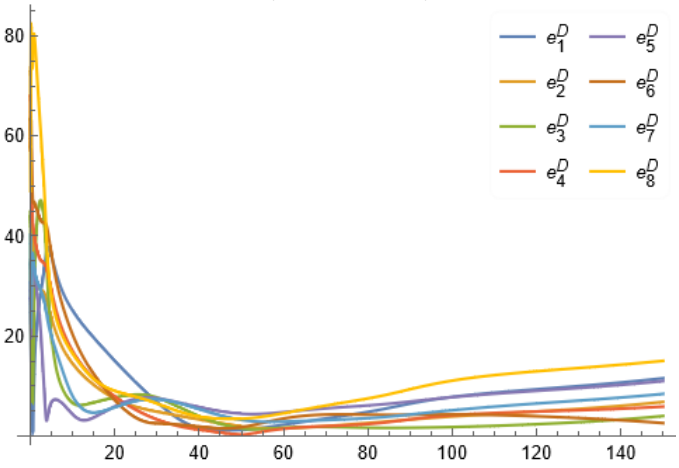
Figures 5.7, 5.10 and 5.13 show the errors between the agents' positions and the trajectories of $\mathbf{c}^H$, $\mathbf{c}^D$ and $\tilde{\mathbf{c}}$, associated with the coverage metric H, KL divergence index D and the total multi-objective index J respectively. The coverage controller drives the agents to the centroids $\mathbf{c}^H$ of the Voronoi cells as shown in figure 5.7, while the KL controller drives the agents to the stationary points $\mathbf{c}^D$ of the KL index, see figure 5.10. Furthermore, figure 5.13 reveals that the agents converge to the trajectories of the points $\tilde{\mathbf{c}}$ of J under the multi-objective control, which drives the agents to a configuration of the stationary points $\tilde{c}$, minimizing the coverage metrics H and KL divergence D as much as possible. As this is a multi-objective scenario, both sub-index H and J are sub-optimized. However, in the sense of overall performance under specific weights α and β, $\tilde{\mathbf{c}}$ represents a trade-off that incorporates the characteristics of centroidal Voronoi tessellation and tracking of spatial statistical distribution.

Figures 5.8, 5.11 and 5.14 further illustrate the trade-off between coverage and statistical tracking performance from the perspective of performance indices. Figure 5.8 displays the performance indices when only the coverage control function is active, where the coverage metric H converges to an optimal value of 88, but the KL divergence in this scenario does not converge. In figure 5.11, under the effect of the KL controller, the KL divergence between the agents' collective density and the target time-varying density converges to zero; however, the coverage metric only reaches a non-optimal value of 142. The agents' trajectories generated by the multi-objective controller minimize both coverage metrics H and the KL divergence D, converging to the optimal values close to the ones when the sub-controllers operate independently.

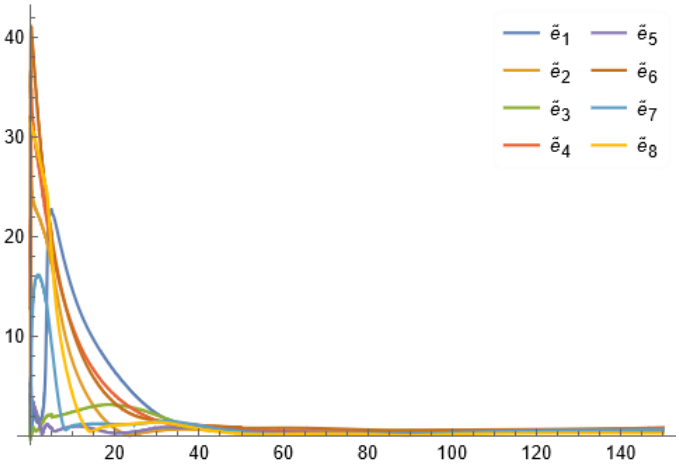
In summary, this subsection discusses the effect of the sub-indices H and D. Optimizing the coverage metric H drives the agents to a centroidal Voronoi tessellation. Simultaneously, the minimization of the KL-Divergence forces the converged Voronoi centroidal configuration to have the same spatial statistical moments as the target density.



(a) $e_i^H = \|\mathbf{p}_i - \mathbf{c}_i^H\|$

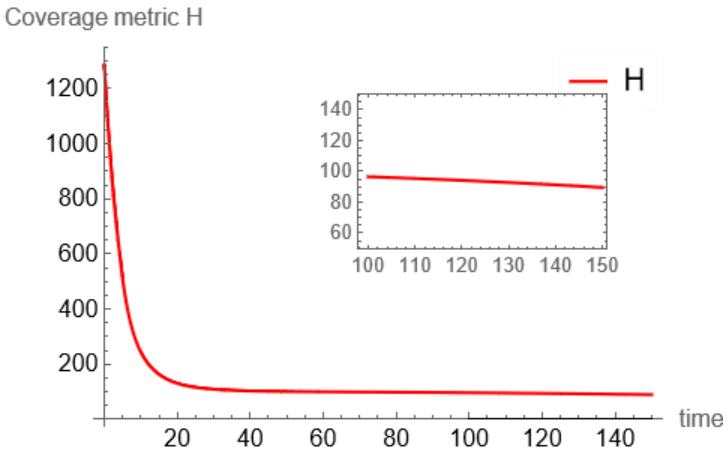


(b) $e_i^D = \|\mathbf{p}_i - \mathbf{c}_i^D\|$

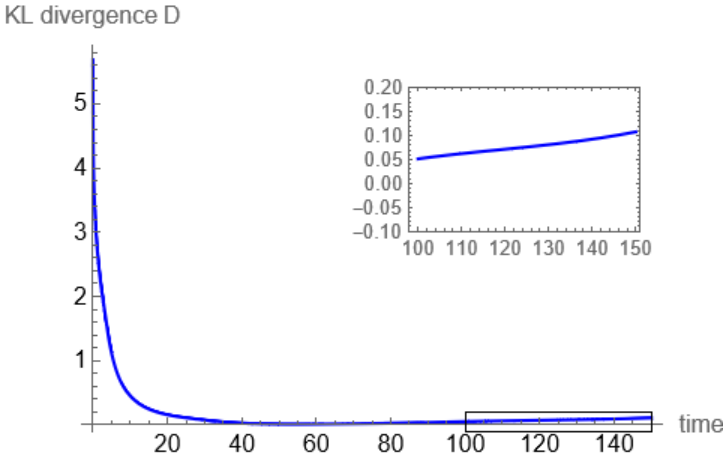


(c) $\tilde{e}_i = \|\tilde{\mathbf{p}}_i - \tilde{\mathbf{c}}_i\|$

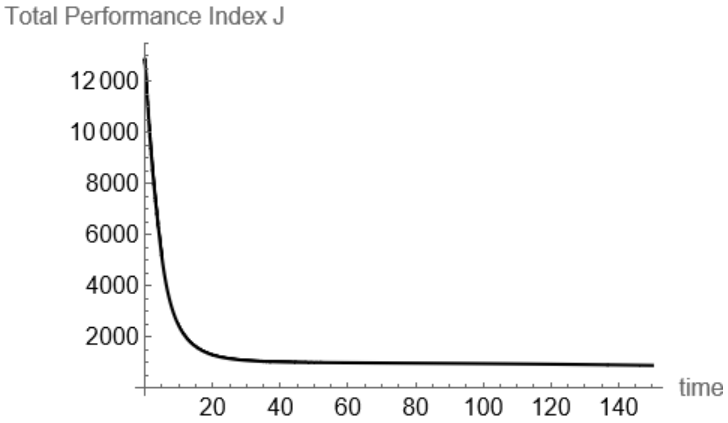
Figure 5.7: Euclidean distance between the agents and their corresponding stationary points of H, D and J for $\alpha = 10$, $\beta = 0$.



(a) Coverage metric defined in (4.8)

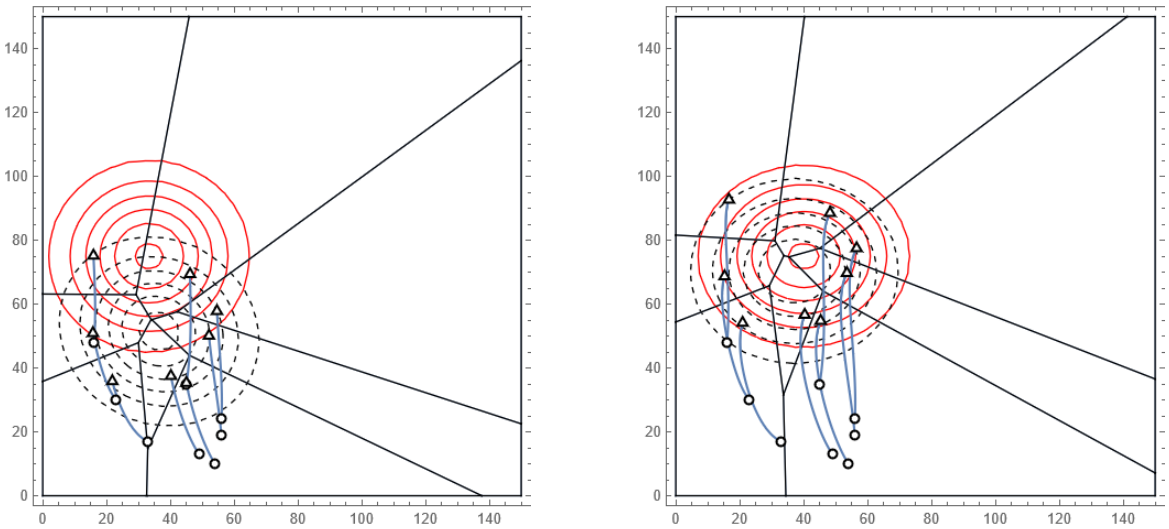


(b) KL divergence index defined in (4.9)



(c) $J = \alpha H + \beta D$

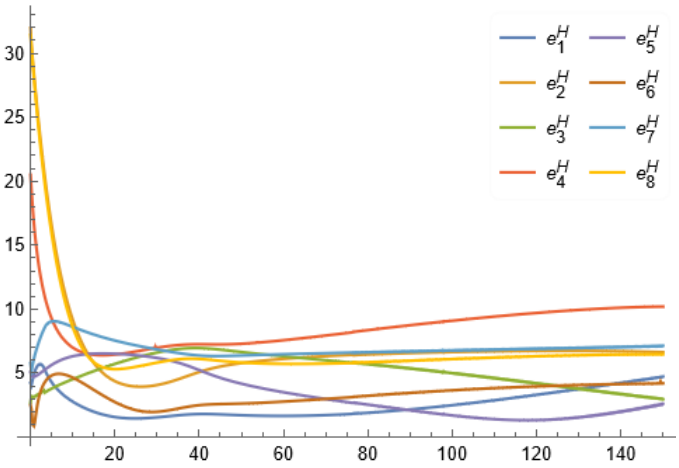
Figure 5.8: Performance indices with coverage control $\alpha = 10$, $\beta = 0$.



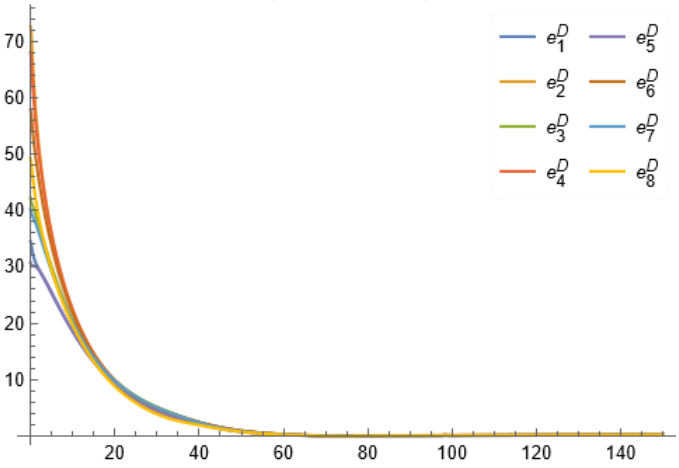
(a) $t = 10$ (b) $t = 30$

(c) $t = 70$ (d) $t = 150$

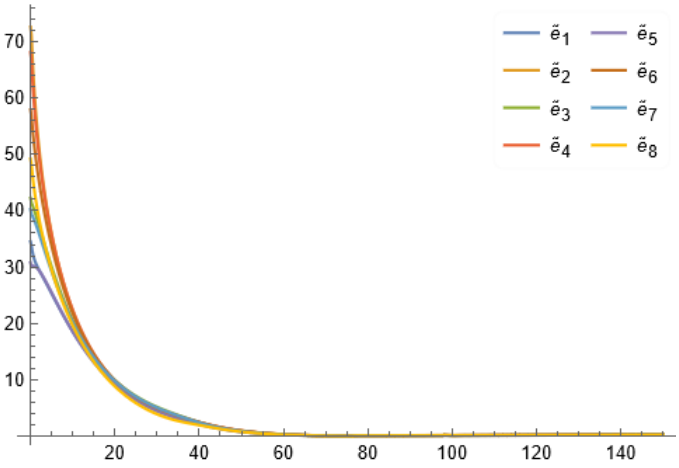
Figure 5.9: Snapshots of the area coverage performance for $\alpha = 0$, $\beta = 2 \times 10^4$ with control gain $k = 1$. The hollow dots denote the initial position of the agents. The triangles denote the agents' positions at each time instant, along with the corresponding collective density (dashed contours) and Voronoi diagram.



(a) $e_i^H = \|\mathbf{p}_i - \mathbf{c}_i^H\|$

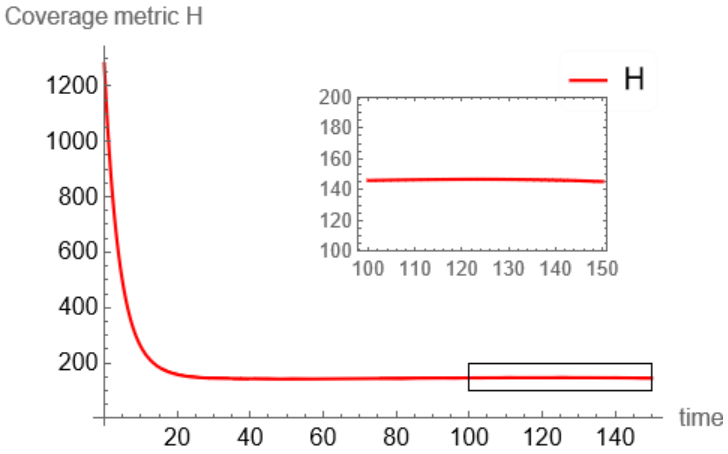


(b) $e_i^D = \|\mathbf{p}_i - \mathbf{c}_i^D\|$

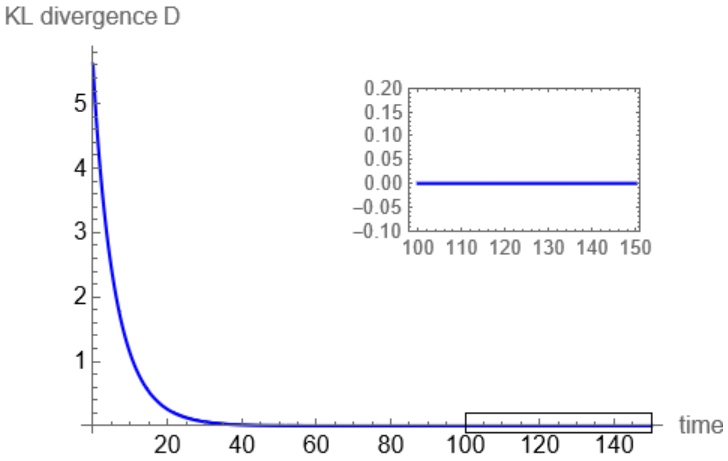


(c) $\tilde{e}_i = \|\mathbf{p}_i - \tilde{\mathbf{c}}_i\|$

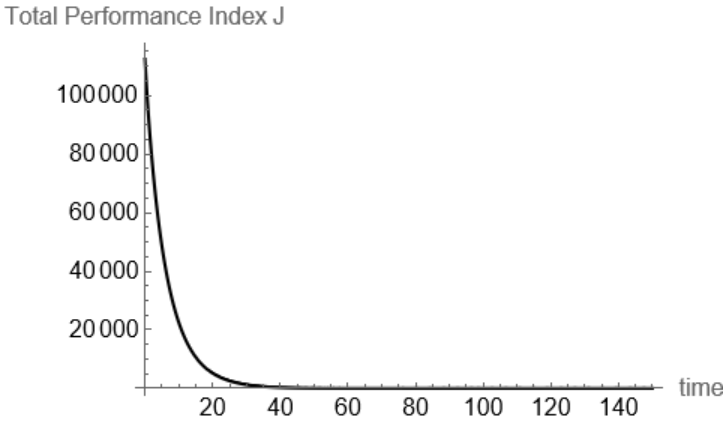
Figure 5.10: Euclidean distance between the agents and their corresponding stationary points of H, D and J for $\alpha = 0$, $\beta = 2 \times 10^4$.



(a) Coverage metric defined in (4.8)

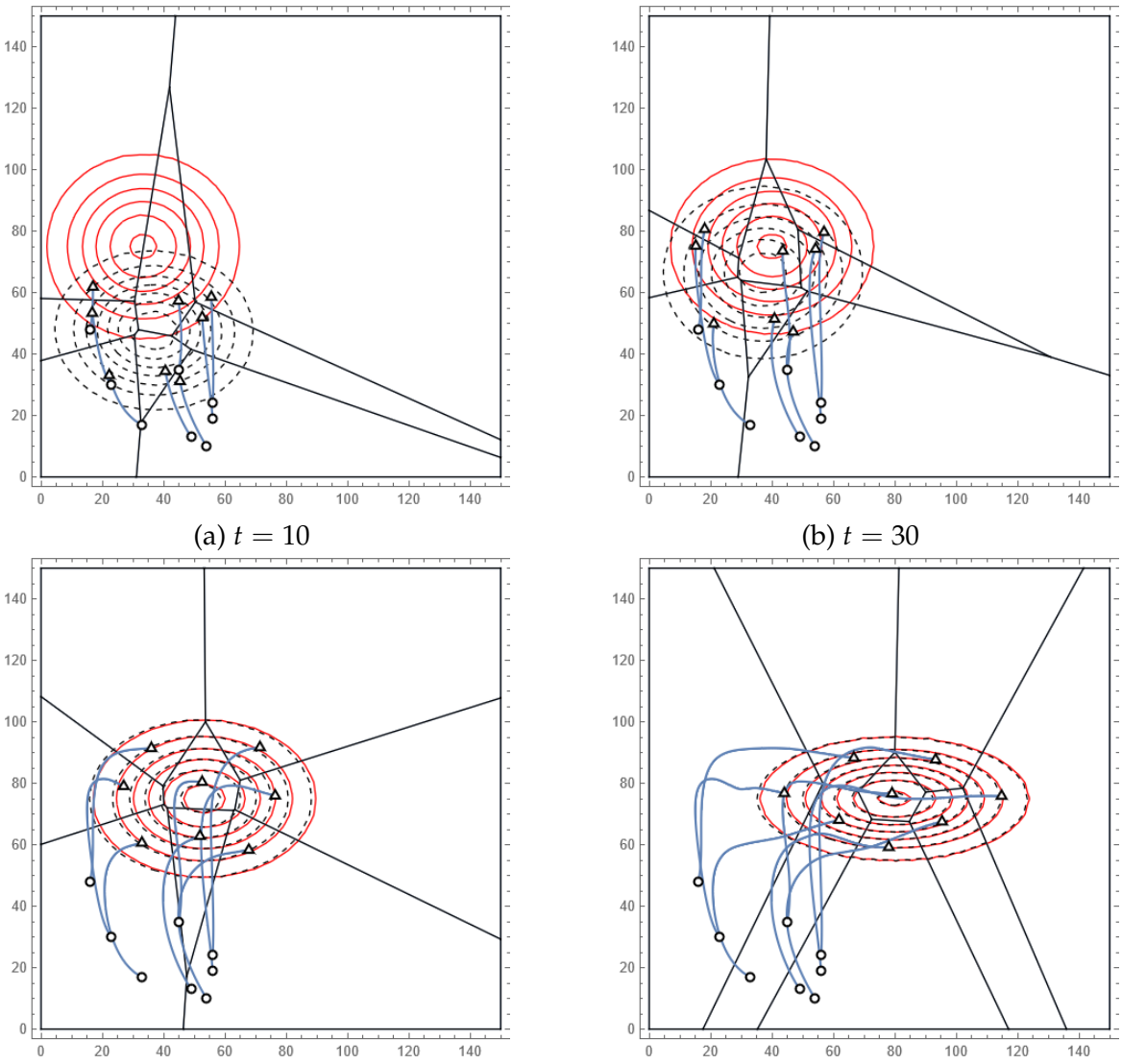


(b) KL divergence index defined in (4.9)



(c) $J = \alpha H + \beta D$

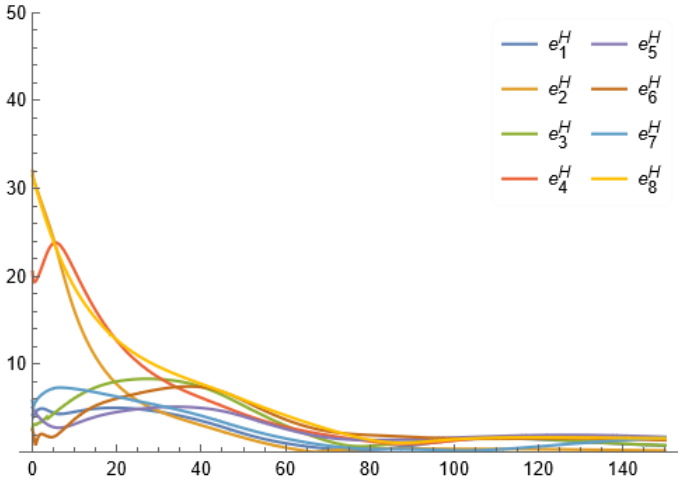
Figure 5.11: Performance indices with KL control $\alpha = 0$, $\beta = 2 \times 10^4$.



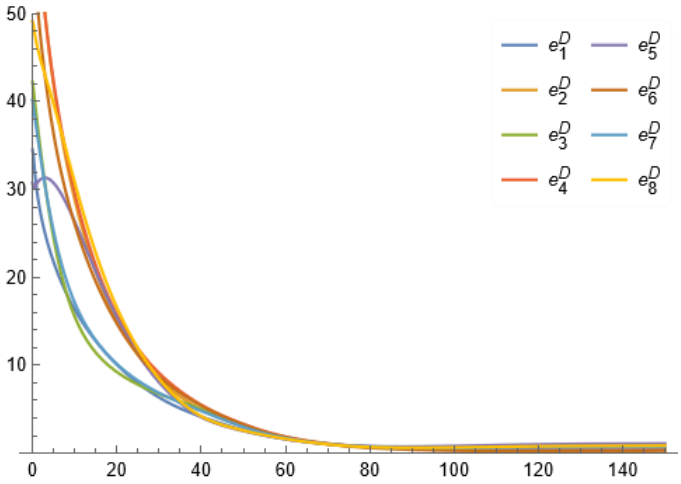
(a) $t = 10$ (b) $t = 30$

(c) $t = 70$ (d) $t = 150$

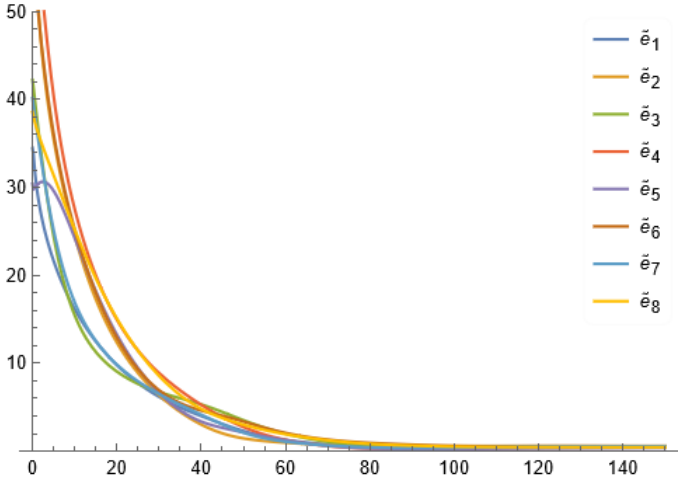
Figure 5.12: Snapshots of the area coverage performance for $\alpha = 10$, $\beta = 2 \times 10^4$ and control gain $k = 1$.



(a) $e_i^H = \|\mathbf{p}_i - \mathbf{c}_i^H\|$

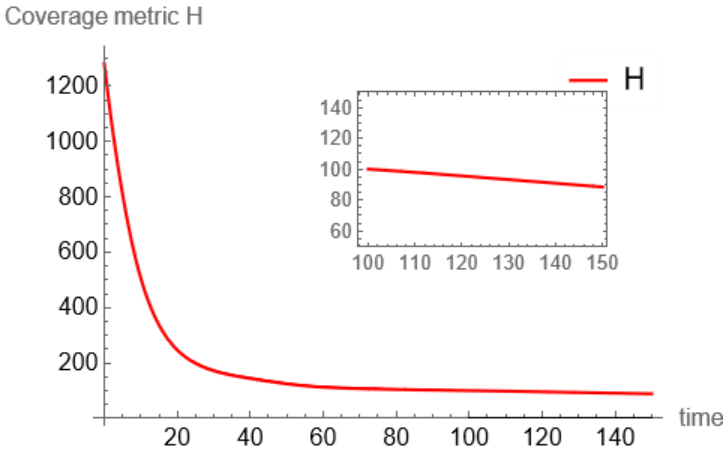


(b) $e_i^D = \|\mathbf{p}_i - \mathbf{c}_i^D\|$

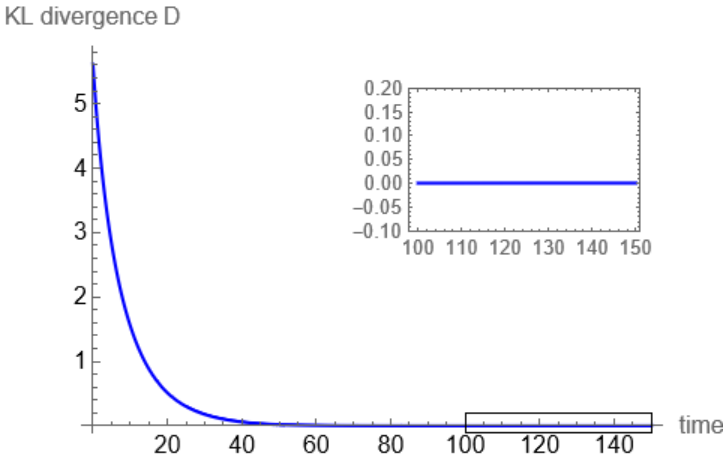


(c) $\tilde{e}_i = \|\mathbf{p}_i - \tilde{\mathbf{c}}_i\|$

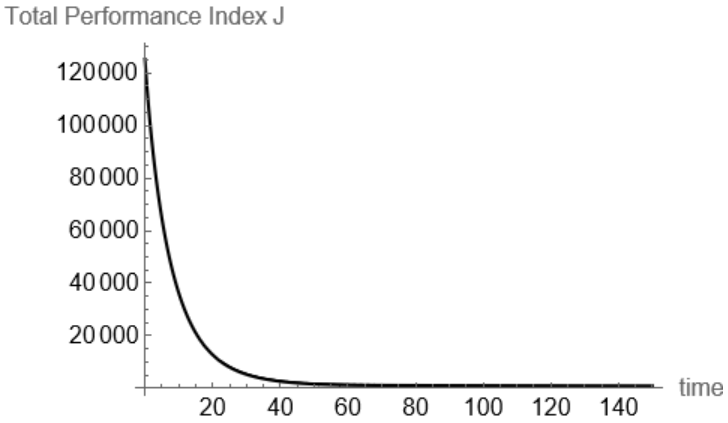
Figure 5.13: Euclidean distance between the agents and their corresponding stationary points of H, D and J for $\alpha = 10$, $\beta = 2 \times 10^4$.



(a) Coverage metric defined in (4.8)



(b) KL divergence index defined in (4.9)



(c) $J = \alpha H + \beta D$

Figure 5.14: Performance indices with multi-objective control $\alpha = 10$, $\beta = 2 \times 10^4$.

5.2.2 Desired Density Design via Elliptical Contour: An Application in Confined Environments

In this subsection, the proposed control law is applied to drive the agents through an L-shaped tunnel. As discussed in Subsection 4.1.1, the bound of the region occupied by the agents can be represented by an elliptical contour of a constant confidence level. The contribution of the KL-divergence term ensures that the collective density of the agents is driven to the target time-varying density. Consequently, as long as the elliptical contour of the target density is contained within the tunnel, the agents will avoid collisions with the tunnel walls once the algorithm converges.

The tunnel has a uniform width of 20 units and consists of two straight-line sections connected by a 90-degree arc. The centerlines of the two straight sections span from (40,60) to (40,90) and (80,130) to (120,130) respectively, which are connected by a 90-degree arc centred at (80,90) with a radius of 40. In this scenario, the target time-varying density is designed based on its 95% confidence elliptical contour remaining tangent to the tunnel walls. This design ensures that the converged KL term guarantees collision avoidance for the agents within the tunnel. The 95% confidence ellipse of the target density defined in equation (4.7) is given by

$$(\mathbf{q}-\boldsymbol{\mu}_d)^\top \boldsymbol{\Sigma}_d^{-1} (\mathbf{q}-\boldsymbol{\mu}_d) = -2\ln(1-c_d) \tag{5.1}$$

with $c_d = 0.95$. The covariance matrix $\boldsymbol{\Sigma}_d$ can be decomposed as the product of scale matrix $\boldsymbol{\Lambda}$ and rotational matrix $\boldsymbol{R}$ as follows

$$\boldsymbol{\Sigma}_d = \boldsymbol{R}\boldsymbol{\Lambda}\boldsymbol{R}^\top \tag{5.2a}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} s_x^2 & 0 \\ 0 & s_y^2 \end{bmatrix} \tag{5.2b}$$

$$\boldsymbol{R} = [\boldsymbol{\lambda}_x | \boldsymbol{\lambda}_y] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{5.2c}$$

where $\boldsymbol{\Lambda}$ determines the length of axes of the elliptical contour for certain c_d, which

are given by

$$l_x = \sqrt{-2\ln(1-c_d)s_x} \tag{5.3a}$$

$$l_y = \sqrt{-2\ln(1-c_d)s_y} \tag{5.3b}$$

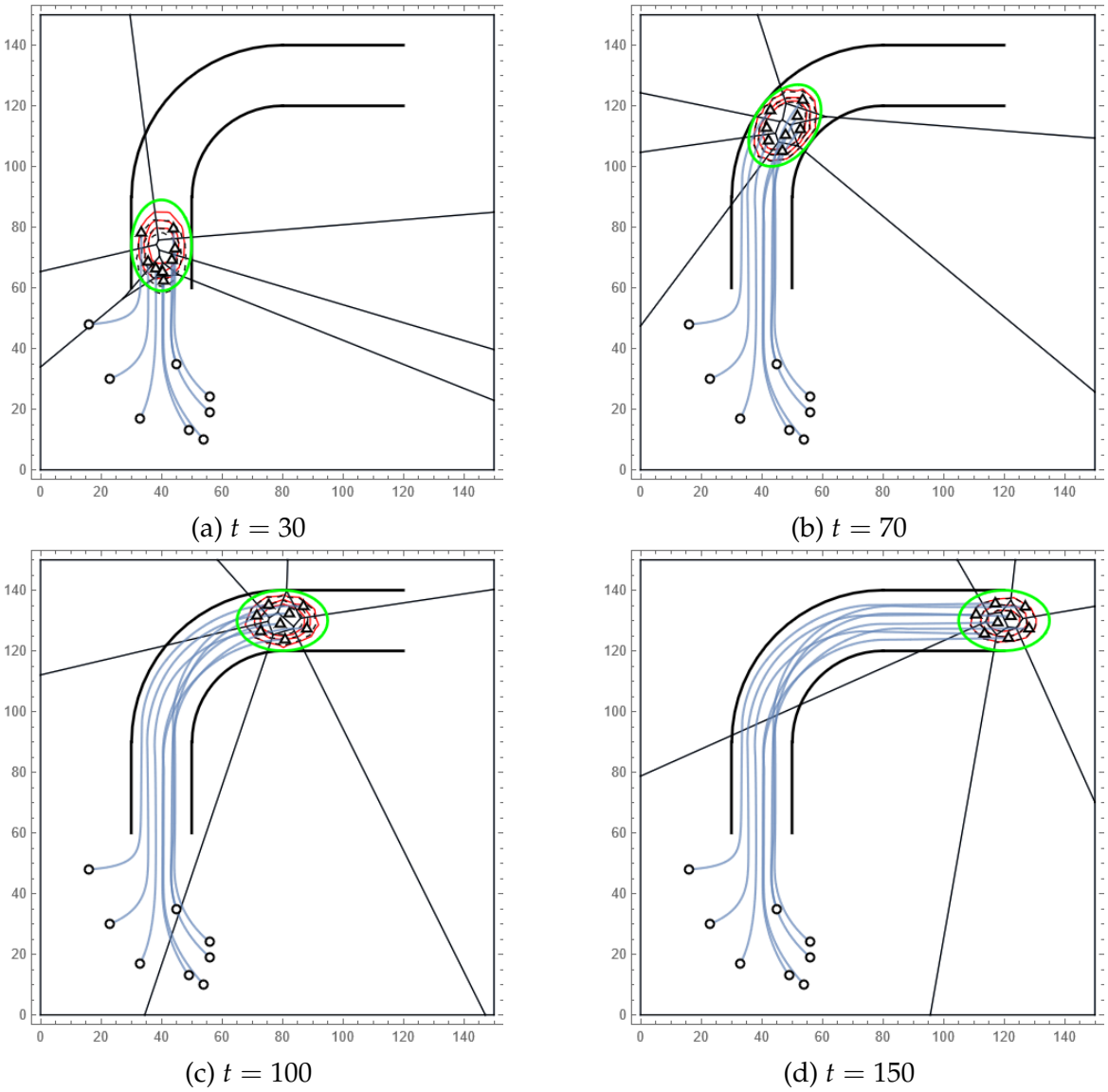
and matrix $\boldsymbol{R}$ determines the angel rotated counter-clockwise.

To control the collective position of the agents for passing through the tunnel without collision, the elliptical contour of the target density ϕ_d with $c_d = 0.95$ is designed such that it is always tangent to the tunnel inner wall to "slide" through the tunnel. One set of simulation results is presented below, with weights factor $\alpha = 10$, $\beta = 3 \times 10^4$, and control gain $k = 10$.

Figure 5.15 depicts the agents' trajectories at different time instants. The green ellipse denotes the 95% confidence contour for the target Gaussian density. After the KL term convergence, the agents are contained within the green ellipse, ensuring collision avoidance with the tunnel walls. At the same time, the converged coverage metric drives the agents into a centroidal Voronoi tessellation.

Figure 5.16 shows the tracking in the sense of density function. In figures 5.16a and 5.16b, it can be observed that the mean $\boldsymbol{\mu}_P$ and components of covariance matrix $\boldsymbol{\Sigma}_P$ for the agents' collective density ϕ_P (dashed curves) converge to and track the ones of the time-varying target density ϕ_d. Figure 5.16c shows the Euclidean distance between $\boldsymbol{\mu}_P$ and $\boldsymbol{\mu}_d$, as well as the Frobenius norm of the error between $\boldsymbol{\sigma}_P$ and $\boldsymbol{\sigma}_d$, further demonstrating of the tracking in the sense of statistical moments.

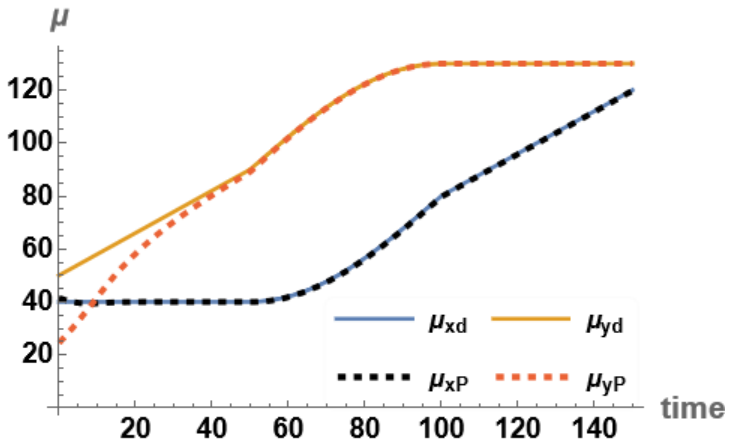
This capability of driving a group of agents accurately through a tunnel in the centroidal Voronoi configuration implied an application of the formation in emergency scenarios, for example, environmental cues, such as visual and auditory signals, can be modelled as distributed signals to guide a crowd safely towards an emergency exit. In addition, this formation can also benefit multi-agent cooperative missions in confined environments, such as searching and rescue missions in mine caves or tunnel detection missions.



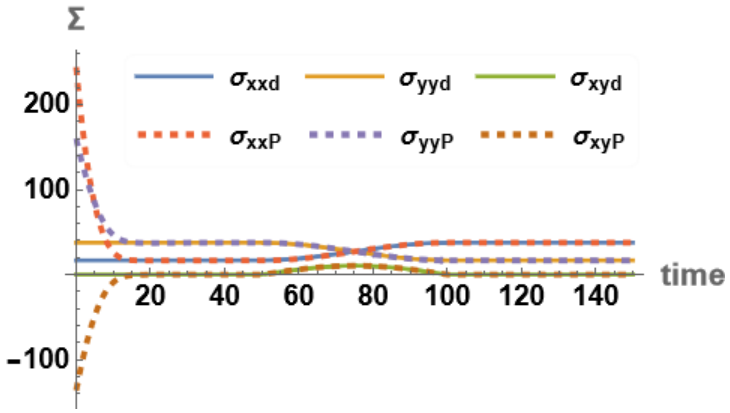
(a) $t = 30$ (b) $t = 70$

(c) $t = 100$ (d) $t = 150$

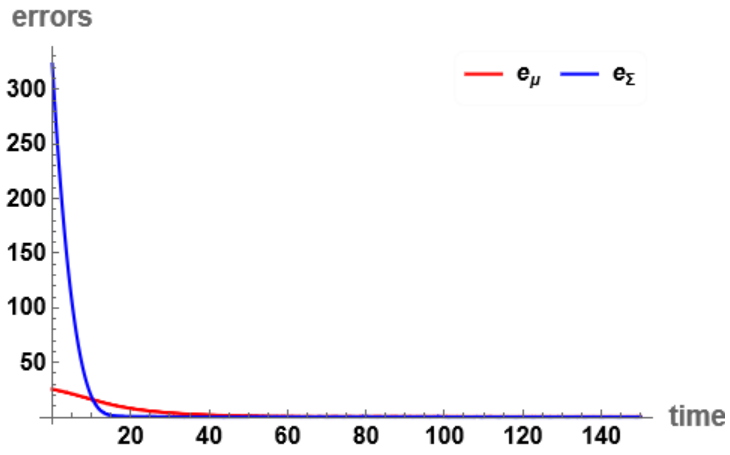
Figure 5.15: Snapshots of the tunnel scenario simulation with $\alpha = 10$, $\beta = 3 \times 10^4$ and control gain $k = 10$. The hollow dots denote the initial position of the agents. The triangles denote the agents' positions at each time instant, along with the corresponding collective density (dashed contours) and Voronoi diagram. The green ellipse is the 95% confidence ellipse for the time-varying target density, which is defined to be always within the tunnel.



(a) The mean μ_P of the agents' collective density tracks the target value in x and y directions.



(b) The components of the covariance matrix $\mathbf{\Sigma}_P$ of the agents' collective density, including standard deviation σ_{xxP}, σ_{yyP} and covariance σ_{xyP} track the target values in the desired variance-covariance matrix $\mathbf{\Sigma}_P$



(c) Errors between distributions' values displayed in log scale, where $e_\mu = \|\mu_d - \mu_P\|$, $e_\Sigma = \|\Sigma_d - \mu_P\|_F$.

Figure 5.16: Tracking performance in the sense of statistical moments.

Chapter 6

Conclusion and Future Work

This thesis has proposed a state feedback law for the formation control of kinematic multi-agent systems, to optimize a multi-objective performance index composed of the weighted area coverage metric and the KL-Divergence between the agents' spatial density and a desired density, coinciding with the risk density of the area coverage.

The stability of the control law is established by using Barbalat's lemma, as the target density is allowed to be time-varying. The control law is proved to minimize the performance index monotonically, and to drive the agents to the trajectories of the time-varying stationary points of the non-autonomous multi-objective index, which can be interpreted as a modified type of centroidal Voronoi tessellation that has the same statistical moments as the target density. It should be noted that both the coverage metric and the KL divergence are sub-optimal due to the trade-off between the sub-indices in multi-objective optimization, as discussed in Section 5.2.1. The coverage term drives the configuration of the agents into a centroidal Voronoi tessellation, ensuring the collision avoidance performance among the agents and maximizing the coverage to the target density, meanwhile, the KL term forces the converged centroidal configuration into the same statistical distribution as the target density. What's more, by designing the target density via its elliptical contour of a certain confidence level, the converged KL term contains the agents in an ellipse region, which can be utilized in the scenarios of confined environments, such as passing through tunnels, corridors, or pipelines while maintaining centroidal Voronoi configuration, potentially benefiting the search and rescue missions in confined environments, as elaborated in the second simulation in Section 5.2.2.

The restriction of this work is that the communication network has been assumed to be fully connected and synchronous. In the future, more general network

topologies can be considered for more realistic scenarios. In addition, higher-order can be considered, in which more than one agent can be assigned to each Voronoi cell, enhancing redundancy and robustness.

Appendix A

Mathematica Code for the Multi-objective Controller in 5.2.1

Appendix A shows the Mathematica code for the multi-objective control with $\alpha = 10, \beta = 2 \times 10^4$ presented in subsection 5.2.1.

```
(*Define 150*150 environment*)

X = 150;
Y = 150;
ff[i_, j_] = {i, j};
q = Array[ff, {X, Y}];
Off[General::munfl]
Off[NIntegrate::ncvbr]


(*Define n variables (from p1 to pn)*)

p0 = {{33, 17}, {45, 35}, {54, 10}, {56, 24}, {23, 30}, {56, 19}, {49,
    13}, {16, 48}};
p = p0;
n = Length[p];

{pxl, pxr} =
 Transpose@
  ToExpression[{StringJoin["px", #, "_"], StringJoin["px", #]} & /@
    Array[ToString, n]];

{pyl, pyr} =
 Transpose@
```

```
  ToExpression[{StringJoin["py", #, "_"], StringJoin["py", #]} & /@
    Array[ToString, n]];

(*Desired density μxd, μyd, σxd, σyd, ρd*)

μx0 = 30;
μy0 = 75;
σx0 = 16;
σy0 = 16;
ρ0 = 0;

(*Final density μxd, μyd, σxd, σyd, ρd*)

μxf = 80;
μyf = 75;
σxxf = 10^2;
σyyf = 12^2;
σxyf = 10 12 0.2;

μxf = 80;
μyf = 75;
σxf = 22;
σyf = 10;
ρf = 0;

T = 150;
μxd[t_] = μx0 + (μxf - μx0)/T t;
μyd[t_] = μy0 + (μyf - μy0)/T t;
σxd[t_] = σx0 + (σxf - σx0)/T t;
σyd[t_] = σy0 + (σyf - σy0)/T t;
ρd[t_] = ρ0 + (ρf - ρ0)/T t;
σd[
   t_] = {{σxd[t]^2, ρd[t] σxd[t] σyd[t]}, {ρd[
      t] σxd[t] σyd[t], σyd[t]^2}};
μd[t_] = {μxd[t], μyd[t]};

μP[pxl, pyl] = {μx[pxr], μy[pyr]};
σP[pxl,
   pyl] = {{σxx[pxr], σxy[pxr, pyr]}, {σxy[pxr,
     pyr], σyy[pyr]}};

ϕd[{x_, y_}, t_] =
  Exp[-(1/(2 (1 - ρd[t]^2))) (((x - μxd[t])/σxd[t])^2 -
```

```
       2 ρd[t] ((x − μxd[t])/σxd[t]) ((
         y − μyd[t])/σyd[t]) + ((
        y − μyd[t])/σyd[t])^2)]/(2 \[Pi] σxd[t] σyd[
      t] Sqrt[1 − ρd[t]^2]);

(*Sensing performance*)

γ = 1;
f[{x_, y_}, {pxi_, pyi_}] = γ ((x − pxi)^2 + (y − pyi)^2);

fp[{x_, y_}, {pxi_, pyi_}] = D[f[{x, y}, {pxi, pyi}], {{pxi, pyi}}];
ϕPp[{x_, y_}, pxl, pyl] =
  Table[D[ϕP[{x, y}, pxr, pyr], {{pxr[[i]], pyr[[i]]}}], {i, n}];

μdt[t_] = D[μd[t], t];
Σdt[t_] = D[Σd[t], t];

ϕdt[{x_, y_}, t_] = D[ϕd[{x, y}, t], t];

dfr2[{x_, y_}, {pxi_, pyi_}] = γ;

(*Voronoi calculation*)
vor[p_] :=
  Module[{r2, fr2, MaxFr2, Vdis, reg},
   r2 = Table[EuclideanDistance[q[[i]][[j]], p[[k]]], {i, X}, {j, Y},
        {k, n}];
   fr2 = r2^2;
   MaxFr2 = Table[Min[fr2[[i]][[j]]], {i, X}, {j, Y}];
   Vdis = Table[{}, {n}];
   Table[If[fr2[[i]][[j]][[k]] == MaxFr2[[i]][[j]],
     AppendTo[Vdis[[k]], q[[i]][[j]]]], {i, X}, {j, Y}, {k, n}];
   reg = Table[ConvexHullRegion[Vdis[[k]]], {k, n}];
   reg];

(*KL divergence*)
DKL[pxl, pyl, t_] =
  1/2 (Tr[Inverse[Σd[t]] . ΣP[pxr, pyr]] +
     Transpose[(μd[t] − μP[pxr, pyr])] .
      Inverse[Σd[t]] . (μd[t] − μP[pxr, pyr]) +
     Log[Det[Σd[t]]/Det[ΣP[pxr, pyr]]] − 2);

(*control gain and weights*)
k = 1;
```

```
A = 10;
B = 20000;

(*initialization*)
$HistoryLength = 1;
reg = vor[p0];
mH0 = Parallelize[
   Table[NIntegrate[-2 dfr2[{x, y}, p0[[i]]] ϕd[{x, y}, 0], {x,
       y} ∈ reg[[i]], AccuracyGoal -> 5], {i, n}]];
cH0 = Parallelize[
   Table[NIntegrate[-2 dfr2[{x, y}, p0[[i]]] ϕd[{x, y}, 0] {x,
        y}, {x, y} ∈ reg[[i]], AccuracyGoal -> 5]/
     mH0[[i]], {i, n}]];
Hpi0 = Table[mH0[[i]] (cH0[[i]] - p0[[i]]), {i, n}];
Hti0 = Parallelize[
   Table[NIntegrate[
     f[{x, y}, p[[i]]] ϕdt[{x, y}, 0], {x, y} ∈
      reg[[i]], AccuracyGoal -> 5], {i, n}]];
Dklpi0 = Parallelize[
   Table[( -(1/n) Inverse[Σd[
          0]] . (μd[0] - μP[px, py]) + (1/(n -
          1)) Transpose[(Inverse[Σd[
            0]] - (1/
             Det[ΣP[px,
               py]]) Adjugate[ΣP[px, py]])] . (p0[[
          i]] - μP[px, py])), {i, n}]];
Dklt0 = Transpose[
    Inverse[Σd[
       0]] . (μd[tt] - μP[px, py])] . μdt[0](*μdt*)+
   Tr[(1/2) (-Inverse[Σd[0]] . ΣP[px,
          py] . Inverse[Σd[0]] -
       Inverse[Σd[0]] .
        Transpose[{μd[0] - μP[px, py]}] . {μd[
           0] - μP[px, py]} . Inverse[Σd[0]] +
       Adjugate[Σd[0]]/
        Det[Σd[0]]) . Σdt[0]];
cD0 = Table[((n - 1)/n) μd[0] + (1/n) μP[px,
     py] - Σd[0] .
    Inverse[ΣP[px, py]] . μP[px,
     py] + Σd[0] . Inverse[ΣP[px, py]] .
    p[[i]], {i, n}];
ctilde0 =
  Table[(Inverse[
```

```
      A mH0[[i]] IdentityMatrix[2] -
       B (1/(n - 1)) Inverse[Σd[0]]]) . (A mH0[[
        i]] cH0[[i]] - (1/(n - 1)) B Inverse[Σd[tt]] .
        cD0[[i]]), {i, n}];
λ0 =
  Table[Norm[(A Hpi0[[j]] + B Dklpi0[[j]])]^2/
    Total[Table[Norm[(A Hpi0[[i]] + B Dklpi0[[i]])]^2, {i, n}]], {j,
    n}];
H0 = Total[
   Parallelize[
    Table[NIntegrate[
      f[{x, y}, p0[[i]]] ϕd[{x, y}, 0], {x, y} ∈
       reg[[i]], AccuracyGoal -> 5], {i, n}]]];
J0 = A H0 + B DKL[px, py, 0];

(*Data storage*)
Hs = {H0};
ps = {p};
us = {Table[{0, 0}, {n}]};
Js = {J0};
DKLs = {DKL[px, py, 0]};
cds = {cD0};
chs = {cH0};
ctildes = {ctilde0};
ps = {p0};
ts = {0};
tt = 0;
Vdis = Table[{}, {n}];
Δt = 0.2;

(*loop*)
While[
  tt < T,
  (*Substitute positions of agents for control input*)
  Module[{vor, Hpi, Hti, Dklpi, Dklt, uH, uD, mH, cH, cD, ctilde, u,
    J, H, λ, reg, pardt, pt, ci, mti,
    cti},
   vor = VoronoiMesh[p, {{0, X}, {0, Y}}];
   reg = Transpose[
      Table[Select[MeshPrimitives[vor, 2],
        RegionMember[#, p[[i]]] &], {i, n}]][[1]];
   mH = Parallelize[
     Table[NIntegrate[-2 dfr2[{x, y}, p[[i]]] ϕd[{x, y}, tt], {x,
```

```
       y} ∈ reg[[i]], AccuracyGoal -> 5], {i, n}]];
cH = Parallelize[
  Table[NIntegrate[-2 dfr2[{x, y}, p[[i]]] ϕd[{x, y}, tt] {x,
       y}, {x, y} ∈ reg[[i]], AccuracyGoal -> 5]/
    mH[[i]], {i, n}]];
Hpi = Table[mH[[i]] (cH[[i]] - p[[i]]), {i, n}];
Hti = Parallelize[
  Table[NIntegrate[
    f[{x, y}, p[[i]]] ϕdt[{x, y}, tt], {x, y} ∈
     reg[[i]], AccuracyGoal -> 5], {i, n}]];
Dklpi =
 Parallelize[
  Table[( -(1/n) Inverse[Σd[
         tt]] . (μd[tt] - μP[px, py]) + (1/(n -
          1)) Transpose[(Inverse[Σd[
            tt]] - (1/
             Det[ΣP[px,
               py]]) Adjugate[ΣP[px, py]])] . (p[[
          i]] - μP[px, py])), {i, n}]];
Dklt =
 Transpose[
    Inverse[Σd[
       tt]] . (μd[tt] - μP[px, py])] . μdt[
    tt](*μdt*)+
  Tr[(1/2) (-Inverse[Σd[tt]] . ΣP[px,
         py] . Inverse[Σd[tt]] -
       Inverse[Σd[tt]] .
        Transpose[{μd[tt] - μP[px, py]}] . {μd[
           tt] - μP[px, py]} . Inverse[Σd[tt]] +
       Adjugate[Σd[tt]]/
        Det[Σd[tt]]) . Σdt[tt]];
cD = Table[((n - 1)/n) μd[tt] + (1/n) μP[px,
     py] - Σd[tt] .
    Inverse[ΣP[px, py]] . μP[px,
     py] + Σd[tt] .
    Inverse[ΣP[px, py]] . p[[i]], {i, n}];
ctilde =
 Table[(Inverse[
     A mH[[i]] IdentityMatrix[2] -
      B (1/(n - 1)) Inverse[Σd[tt]]]) . (A mH[[
      i]] cH[[i]] - (1/(n - 1)) B Inverse[Σd[tt]] .
      cD[[i]]), {i, n}];
λ =
```

```
 Table[Norm[(A Hpi[[j]] + B Dklpi[[j]])]^2/
   Total[Table[Norm[(A Hpi[[i]] + B Dklpi[[i]])]^2, {i, n}]], {j,
   n}];
u = Table[((A Hpi[[i]] + B Dklpi[[i]])/
    Norm[A Hpi[[i]] + B Dklpi[[i]]]^2) (-k Norm[
      ctilde[[i]] - p[[i]]]^2 - λ[[
      i]] (A Total[Hti] + B Dklt)), {i, n}];
p = p + u Δt;
H = Total[
  Parallelize[
   Table[NIntegrate[
     f[{x, y}, p[[i]]] ϕd[{x, y}, tt], {x, y} ∈
      reg[[i]], AccuracyGoal -> 5], {i, n}]]];
J = A H + B DKL[px, py, tt];
ClearSystemCache;
Unprotect[Out];
Clear[Out];
tt = tt + Δt;
AppendTo[ps, p];
AppendTo[Js, J];
AppendTo[DKLs, DKL[px, py, tt]];
AppendTo[Hs, H];
AppendTo[cds, cD];
AppendTo[chs, cH];
AppendTo[ctildes, ctilde];
AppendTo[ts, tt]]];
```

Bibliography

[1] Ruiming Zheng and Davide Spinello. "Formation Control of Multi-agent Systems via Voronoi Tessellation and Kullback-Leibler Divergence". In: *Proceedings of the 63rd IEEE Conference on Decision and Control*. Accepted for publication. 2024.

[2] E. Simetti et al. "Towards the Use of a Team of USVs for Civilian Harbour Protection: USV Interception of Detected Menaces". In: *IFAC Proceedings Volumes* 43.16 (2010), pp. 145–150. DOI: 10.3182/20100906-3-IT-2019.00027.

[3] Luciano C.A. Pimenta et al. "Decentralized Controllers for Perimeter Surveillance with Teams of Aerial Robots". In: *Advanced Robotics* 27.9 (June 2013), pp. 697–709. DOI: 10.1080/01691864.2013.778942.

[4] Annalisa Milella et al. "Active Surveillance of Dynamic Environments Using a Multi-Agent System". In: *IFAC Proceedings Volumes* 43.16 (2010), pp. 13–18. DOI: 10.3182/20100906-3-IT-2019.00005.

[5] Jinwen Hu et al. "Multiagent Information Fusion and Cooperative Control in Target Search". In: *IEEE Transactions on Control Systems Technology* 21.4 (July 2013), pp. 1223–1235. DOI: 10.1109/TCST.2012.2198650.

[6] Daniel S. Drew. "Multi-Agent Systems for Search and Rescue Applications". eng. In: *Current Robotics Reports* 2.2 (2021), pp. 189–200. ISSN: 2662-4087.

[7] Yao Xiao et al. "A Pedestrian Flow Model Considering the Impact of Local Density: Voronoi Diagram Based Heuristics Approach". In: *Transportation Research Part C: Emerging Technologies* 68 (July 2016), pp. 566–580. DOI: 10.1016/j.trc.2016.05.012.

[8] Peng Wang, Shuchao Cao, and Ming Yao. "Fundamental Diagrams for Pedestrian Traffic Flow in Controlled Experiments". In: *Physica A: Statistical Mechanics and its Applications* 525 (July 2019), pp. 266–277. DOI: 10.1016/j.physa.2019.03.057.

[9] Yunchao Qu et al. "Analyzing Pedestrian Individual and Interaction Collision Avoidance Dynamics in Traditional Scenarios". In: *Transportation Research Part C: Emerging Technologies* 133 (Dec. 2021), p. 103445. DOI: 10.1016/j.trc.2021.103445.

[10] Gaia Dell'Ariccia et al. "Flock flying improves pigeons' homing: GPS track analysis of individual flyers versus small groups". eng. In: *Animal behaviour* 76.4 (2008), pp. 1165–1172. ISSN: 0003-3472.

[11] M. Ballerini et al. "Interaction Ruling Animal Collective Behavior Depends on Topological Rather than Metric Distance: Evidence from a Field Study". eng. In: *Proceedings of the National Academy of Sciences - PNAS* 105.4 (2008), pp. 1232–1237. ISSN: 0027-8424.

[12] C. Belta and V. Kumar. "Abstraction and Control for Groups of Robots". In: *IEEE Transactions on Robotics* 20.5 (Oct. 2004), pp. 865–875. DOI: 10.1109/TRO.2004.829498.

[13] Xiaolei Li et al. *Secure Coordination Control of Networked Robotic Systems: From a Control Theory Perspective*. Singapore: Springer Nature Singapore, 2024. ISBN: 978-981-9993-58-1 978-981-9993-59-8. DOI: 10.1007/978-981-99-9359-8.

[14] Takeshi Hatanaka et al. *Passivity-based control and estimation in networked robotics*. eng. Communications and control engineering. Cham: Springer, 2015 - 2015. ISBN: 9783319151717.

[15] Hailong Huang and Andrey V. Savkin. "An Algorithm of Reactive Collision Free 3-D Deployment of Networked Unmanned Aerial Vehicles for Surveillance and Monitoring". eng. In: *IEEE transactions on industrial informatics* 16.1 (2020), pp. 132–140. ISSN: 1551-3203.

[16] Ziquan Yu et al. "Distributed adaptive fractional-order fault-tolerant cooperative control of networked unmanned aerial vehicles via fuzzy neural networks". eng. In: *IET control theory & applications* 13.17 (2019), pp. 2917–2929. ISSN: 1751-8644.

[17] Yingjie Deng, Tao Liu, and Dingxuan Zhao. "Event-triggered output-feedback adaptive tracking control of autonomous underwater vehicles using reinforcement learning". eng. In: *Applied ocean research* 113 (2021), pp. 1–. ISSN: 0141-1187.

[18] Farid Sharifi et al. "A Distributed Deployment Strategy for a Network of Cooperative Autonomous Vehicles". eng. In: *IEEE transactions on control systems technology* 23.2 (2015), pp. 737–745. ISSN: 1063-6536.

[19] Khashayar Asadi et al. "An integrated UGV-UAV system for construction site data collection". eng. In: *Automation in construction* 112 (2020), pp. 103068–. ISSN: 0926-5805.

[20] Lowell W. Beineke and Robin J. Wilson. *Topics in algebraic graph theory*. eng. Encyclopedia of mathematics and its applications ; 102. Cambridge, UK ; Cambridge University Press, 2004. ISBN: 0521801974.

[21] Franz Aurenhammer and Rolf Klein. "Voronoi Diagrams Partially Supported by the Deutsche Forschungsgemeinschaft". In: *Handbook of Computational Geometry*. Elsevier, 2000, pp. 201–290. ISBN: 978-0-444-82537-7. DOI: 10.1016/B978-044482537-7/50006-1.

[22] René Descartes and René Descartes. *Principia philosophiae*. lat. Amstelodami: Apud Danielem Elzevirium, 1664.

[23] "Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen". ger. In: *Journal für die reine und angewandte Mathematik* 40 (1850), pp. 209–227. ISSN: 0075-4102.

[24] Georges Voronoi. "Nouvelles applications des paramètres continus à théorie des formes quadratiques. Deuxième Mémoire. Recherches sur les parallélo\`edres primitifs". ger. In: *Journal für die reine und angewandte Mathematik* 1909.136 (1909), pp. 67–182. ISSN: 0075-4102.

[25] Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. "Nearest neighbourhood operations with generalized Voronoi diagrams: a review". eng. In: *International journal of geographical information systems* 8.1 (1994), pp. 43–71. ISSN: 0269-3798.

[26] Alan (Alan M.) Gibbons. *Algorithmic graph theory*. eng. Cambridge [Cambridgeshire] ; Cambridge University Press, 1985. ISBN: 0521246598.

[27] J. Cortes et al. "Coverage Control for Mobile Sensing Networks". In: *IEEE Transactions on Robotics and Automation* 20.2 (Apr. 2004), pp. 243–255. DOI: 10.1109/TRA.2004.824698.

[28] Jorge Cortes et al. "COVERAGE CONTROL FOR MOBILE SENSING NETWORKS: VARIATIONS ON A THEME". In: *10th Mediterranean Conference on Control and Automation* (2002). URL: https://www.clear.rice.edu/comp651/papers/CoverageControl-CortesEtAl-MED2002.pdf.

[29] Paul E. Black. *Manhattan distance*. 2019. URL: https://xlinux.nist.gov/dads/HTML/manhattanDistance.html.

[30] Erik Mahieu. *Wolfram Demonstrations Project*. URL: https://demonstrations.wolfram.com/ConstructingVoronoiDiagramsWithExpandingCells/.

[31] Oriam De Gyves, Leonel Toledo, and Isaac Rudomín. "Proximity Queries for Crowd Simulation Using Truncated Voronoi Diagrams". eng. In: *Proceedings of Motion on Games*. New York, NY, USA: ACM, 2013, pp. 87–92. ISBN: 1450325467.

[32] J. Champagne and W. Tang. "Real-time Simulation of Crowds Using Voronoi Diagrams". In: *EG UK Theory and Practice of Computer Graphics*. Ed. by Louise M. Lever and Mary McDerby. The Eurographics Association, 2005. ISBN: 3-905673-56-8. DOI: /10.2312/LocalChapterEvents/TPCG/TPCGUK05/195-201.

[33] Yao Xiao et al. "Investigation of Voronoi Diagram Based Direction Choices Using Uni- and Bi-Directional Trajectory Data". In: *Physical Review E* 97.5 (May 18, 2018), p. 052127. DOI: 10.1103/PhysRevE.97.052127.

[34] Avneesh Sud et al. "Real-Time Path Planning in Dynamic Virtual Environments Using Multiagent Navigation Graphs". In: *IEEE Transactions on Visualization and Computer Graphics* 14.3 (2008), pp. 526–538. DOI: 10.1109/TVCG.2008.27.

[35] Qiang Du, Vance Faber, and Max Gunzburger. "Centroidal Voronoi Tessellations: Applications and Algorithms". In: *SIAM Review* 41.4 (Jan. 1999), pp. 637–676. DOI: 10.1137/S0036144599352836.

[36] Wolfram Research. *VoronoiMesh*. 2022. URL: https://reference.wolfram.com/language/ref/VoronoiMesh.html.

[37] Abderrahmane Laraqui et al. "Image mosaicing using voronoi diagram". In: *Multimedia Tools and Applications* 76 (2016), pp. 8803–8829. URL: https://api.semanticscholar.org/CorpusID:12251758.

[38] S. Lloyd. "Least Squares Quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[39] Sung. G. Lee and Magnus Egerstedt. "Controlled Coverage Using Time-Varying Density Functions". In: *IFAC Proceedings Volumes* 46.27 (2013), pp. 220–226. DOI: 10.3182/20130925-2-DE-4044.00030.

[40] Suruz Miah et al. "Generalized Non-Autonomous Metric Optimization for Area Coverage Problems with Mobile Autonomous Agents". In: *Automatica* 80 (June 2017), pp. 295–299. DOI: 10.1016/j.automatica.2017.02.044.

[41] Suruz Miah et al. "Nonuniform Deployment of Autonomous Agents in Harbor-Like Environments". eng. In: *Unmanned systems (Singapore)* 2.4 (2014), pp. 377–389. ISSN: 2301-3850.

[42] Suruz Miah et al. "Nonuniform Coverage Control With Stochastic Intermittent Communication". eng. In: *IEEE transactions on automatic control* 60.7 (2015), pp. 1981–1986. ISSN: 0018-9286.

[43] A. Abdulghafoor and E. Bakolas. "Distributed Coverage Control of Multi-Agent Networks with Guaranteed Collision Avoidance in Cluttered Environments". In: *IFAC-PapersOnLine* 54.20 (2021), pp. 771–776. DOI: 10.1016/j.ifacol.2021.11.265.

[44] Kai Luo et al. "Distributed Coordination of Multi-Agent Systems for Neutralizing Unknown Threats Based on a Mixed Coverage-Tracking Metric". In: *Journal of the Franklin Institute* 357.17 (Nov. 2020), pp. 12700–12723. DOI: 10.1016/j.jfranklin.2020.08.039.

[45] James Kennedy, Airlie Chapman, and Peter M. Dower. "Generalized Coverage Control for Time-Varying Density Functions". In: *2019 18th European Control Conference (ECC)*. 2019 18th European Control Conference (ECC). Naples, Italy: IEEE, June 2019, pp. 71–76. ISBN: 978-3-907144-00-8. DOI: 10.23919/ECC.2019.8796308.

[46] Adekunle A Adepegba, Suruz Miah, and Davide Spinello. "Multi-agent area coverage control using reinforcement learning". In: *The twenty-ninth international flairs conference*. 2016.

[47] Suruz Miah, Mostafa M. H. Fallah, and Davide Spinello. "Non-Autonomous Coverage Control With Diffusive Evolving Density". In: *IEEE Transactions on Automatic Control* 62.10 (Oct. 2017), pp. 5262–5268. DOI: 10.1109/TAC.2016.2633789.

[48] Alyssa Pierson and Daniela Rus. "Distributed Target Tracking in Cluttered Environments with Guaranteed Collision Avoidance". In: *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. 2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). Los Angeles, CA: IEEE, Dec. 2017, pp. 83–89. ISBN: 978-1-5090-6309-3. DOI: 10.1109/MRS.2017.8250935.

[49] Ruikun Zhou, Wail Gueaieb, and Davide Spinello. "A Model-Free Kullback–Leibler Divergence Filter for Anomaly Detection in Noisy Data Series". In: *Journal of Dynamic Systems, Measurement, and Control* 145.2 (Feb. 1, 2023), p. 024501. DOI: 10.1115/1.4056105.

[50] Mostafa Afgani, Sinan Sinanovic, and Harald Haas. "Anomaly Detection Using the Kullback-Leibler Divergence Metric". In: *2008 First International Symposium on Applied Sciences on Biomedical and Communication Technologies*. 2008 First International Symposium on Applied Sciences on Biomedical and Communication Technologies (ISABEL). Aalborg, Denmark: IEEE, Oct. 2008, pp. 1–5. ISBN: 978-1-4244-2647-8. DOI: 10.1109/ISABEL.2008.4712573.

[51] Andrea Giantomassi et al. "Electric Motor Fault Detection and Diagnosis by Kernel Density Estimation and Kullback–Leibler Divergence Based on Stator Current Measurements". In: *IEEE Transactions on Industrial Electronics* 62.3 (Mar. 2015), pp. 1770–1780. DOI: 10.1109/TIE.2014.2370936.

[52] Raphael Ngigi Wanjiku, Lawrence Nderu, and Michael Kimwele. "Dynamic Fine-tuning Layer Selection Using Kullback–Leibler Divergence". In: *Engineering reports (Hoboken, N.J.)* 5.5 (2023).

[53] Antonio Clim, Răzvan Daniel Zota, and Grigore TinicĂ. "The Kullback-Leibler Divergence Used in Machine Learning Algorithms for Health Care Applications and Hypertension Prediction: A Literature Review". In: *Procedia Computer Science* 141 (2018), pp. 448–453. DOI: 10.1016/j.procs.2018.10.144.

[54] Claude Elwood Shannon. *A Mathematical Theory of Communication*. New York: American Telephone and Telegraph Company, 1948.

[55] Solomon Kullback and R. A. Leibler. "On Information and Sufficiency". In: *Annals of Mathematical Statistics* 22 (1951), pp. 79–86. URL: https://api.semanticscholar.org/CorpusID:120349231.

[56] Yufeng Zhang et al. *On the Properties of Kullback-Leibler Divergence Between Multivariate Gaussian Distributions*. Jan. 22, 2023. arXiv: 2102.05485 [cs, math]. Pre-published.

[57] John C. Duchi. "Derivations for Linear Algebra and Optimization". In: 2016. URL: https://api.semanticscholar.org/CorpusID:237012311.

[58] Joseph P. LaSalle and Solomon Lefschetz. *Stability by Liapunov's direct method with applications*. eng. Mathematics in science and engineering ; v. 4. New York: Academic Press, 1961.

[59] Elisabeth (Libby) Osgood, Gayla Cameron, and Christensen. Emma. *Engineering Mechanics : Statics*. eng. Charlottetown, Prince Edward Island: Robertson Library Pressbooks, 2024.

[60] Frank L. Lewis, Draguna L. Vrabie, and Vassilis L. Syrmos. *Optimal Control*. 3rd ed. Hoboken: Wiley, 2012. ISBN: 1-118-12272-0.

[61] Gene H. (Gene Howard) Golub and Charles F. Van Loan. *Matrix computations*. eng. 2nd ed. Johns Hopkins series in the mathematical sciences ; 3. Baltimore: Johns Hopkins University Press, 1989. ISBN: 0801837723.

[62] Alan. Jeffrey. *Advanced engineering mathematics*. eng. San Diego: Harcourt Academic Press, 2002. ISBN: 012382592X.

[63] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge, U.K: Cambridge University Press, 2003. ISBN: 978-0-521-64298-9.

[64] Frank L. Lewis, Draguna L. Vrabie, and Vassilis L. Syrmos. *Optimal Control*. 3rd ed. Hoboken: Wiley, 2012. ISBN: 1-118-12272-0.

[65] Xianda Zhang. *Matrix Analysis and Applications*. Cambridge: Cambridge University Press, 2017. ISBN: 978-1-108-27758-7.