UNIVERSITY OF CALIFORNIA, SAN DIEGO

Deployment Algorithms for Mobile Robots under Dynamic Constraints

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Sciences (Mechanical Engineering)

by

Andrew Nicholas Kwok

Committee in charge:

Professor Sonia Martínez, Chair
Professor Jorge Cortés
Professor J. William Helton
Professor Kenneth Kreutz-Delgado
Professor Miroslav Krstic

2011

The dissertation of Andrew Nicholas Kwok is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2011

EPIGRAPH

Then let us all do what is right, strive with all our might toward the unattainable, develop as fully as we can the gifts God has given us, and never stop learning.

—Ludwig van Beethoven

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I owe the world to my parents. Mom and Dad, without your emphasis on education in all forms I would not be here today. Thank you for all the times that you supported me, and let nothing get in the way of my educational goals. Thank you for all the times you reminded me to practice the piano growing up, it has become an immense source of sanity lately. Thank you also for keeping in touch and reminding me that life also is not just about school.

To my brother and sister Jeff and Jessica. Wow we have gone a long ways since we were young and living at home. You both have such promising and wonderful futures ahead of you, I can tell. Thank you for providing great times growing up and here's to many more good times in the future!

To Sonia, I hope this academic journey that I am concluding has been as enlightening for you as it was for me. Thank you for introducing me to the world of multi agent control, and guiding me through its intricacies in the beginning. Thank you also for developing my skills and for aiding me in achieving more independence in my research endeavors. Last but not least, thank you for reminding me every now and then not to stray too off course in my work.

Dear Professors Miroslav Krstic, Ken Kreutz-Delgado, Bill Helton, and Jorge Cortés, I thank you for your consideration and I am honored to have you all serve on my thesis committee. Thank you for furthering the academic tradition and maintaining the high standards and rigor that it should demand.

Bahman, we have only known each other for a little over one year, but our insightful conversations about math, geometry, and life have sufficiently brought us up to speed and in sync. Thank you for constantly showing me that there is still so much more in the world of mathematics that I can try to learn. I wish you the best in your career, you deserve it.

To my past and present colleagues in team Jorge-Sonia–Minghui, Yu, Mike, Cameron, Dean, Charlie, and Teymur–our group meetings are always insightful and informative. May there be many more group meetings and may they continue to be a source of ideas.

To my brothers in life: Oliver, Dave, and Mike. There are uncountably

many good times to reminisce about. Thank you for giving me support, advice and valued company from the wee days of kindergarten. It is because of you guys that I know the proper definition of awesome friends.

Nima, my graduation buddy, let this be the start of a fantastic future and the beginnings of successful careers. Thank you for offering your lab space for lunch breaks, microwaves, and other non-academic pastimes. For the sake of further self-incrimination, I will stop there.

Finally, thank you to Scripps TG. It is because of your brilliant event that I have something to look forward to at the end of every week. May many more TG's carry on into the future. Last but not least, thank you Filter Coffee shop in Hillcrest for providing the twenty-four hour office away from the office for my late night thesis writing escapades.

This work was funded by the following grants: NSF Career Award CMS-0643673, NSF IIS-0712746, and NSF-0930919.

VITA

2006	B. S. in Mechanical Engineering *with honors*, California Institute of Technology, Pasadena
2008	M. S. in Engineering Sciences (Mechanical Engineering), University of California, San Diego
2011	Ph. D. in Engineering Sciences (Mechanical Engineering), University of California, San Diego

PUBLICATIONS

Journal Publications

A. Kwok and S. Martinez, "Deployment algorithms for a power-constrained mobile sensor network," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 7, pp. 725–842, 2010.

A. Kwok and S. Martinez, "Unicycle coverage control via hybrid modeling," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 528–532, 2010.

A. Kwok and S. Martinez, "A Distributed Deterministic Annealing Algorithm for Limited-Range Sensor Coverage," *IEEE Transactions on Control Systems Technology*, In print, DOI: 10.1109/TCST.2010.2053036, Aug. 2010.

A. Kwok and S. Martinez, "Coverage maximization with autonomous agents in fast flow environments," *AIAA Journal of Guidance, Control, and Dynamics*, Under review, submitted Sep. 2010.

Conference Proceedings

A. Kwok and S. Martinez, "Energy-balancing cooperative strategies for sensor deployment," *IEEE International Conference on Decision and Control*, New Orleans, pp. 6136–6141, 2007.

A. Kwok and S. Martinez, "Deployment algorithms for a power-constrained mobile sensor network," *IEEE International Conference on Robotics and Automation*, Pasadena, pp. 140–145, 2008.

A. Kwok and S. Martinez, "Coverage control with unicycles via hybrid modeling," *American Control Conference*, Seattle, pp. 2672–2677, 2008.

A. Kwok and S. Martinez, "A distributed deterministic annealing algorithm for limited-range sensor coverage," *American Control Conference*, St. Louis, pp. 1448–1453, 2009.

A. Kwok and S. Martinez, "A coverage algorithm for drifters in a river environment," *American Control Conference*, Baltimore, pp. 6436–6441, 2010.

A. Kwok and S. Martinez, "Deployment of drifters in a piecewise-constant flow environment," *IEEE International Conference on Decision and Control*, Atlanta, 2010.

A. Kwok and R. Holsapple, "Approximate decentralized sensor fusion for Bayesian search of a moving target," *AIAA Infotech@Aerospace*, St. Louis, to appear, 2011.

ABSTRACT OF THE DISSERTATION

Deployment Algorithms for Mobile Robots under Dynamic Constraints

by

Andrew Nicholas Kwok

Doctor of Philosophy in Engineering Sciences (Mechanical Engineering)

University of California, San Diego, 2011

Professor Sonia Martínez, Chair

The use of unmanned vehicles in exploration and surveillance operations has become evermore pervasive in today's world. The development of cooperative motion strategies has been fueled by this increasing demand. However, many dynamical models for these autonomous vehicles remain simple and are not accurate representations of a vehicle where such cooperative motion strategies may be physically implemented. This dissertation will focus on the problem of cooperative deployment of autonomous vehicles subject to various constraints. We will enforce communication or sensing range limitations between agents via range-limited partitions of the coverage environment. Additional novel work includes the investigation of constraints due to remaining power supplies, nonholonomic dynamics, and constraints due to external environmental forces. In addition to these various

constraints, we develop a distributed deterministic annealing algorithm to address the separate problem of suboptimal convergence that is exhibited by many of the cooperative deployment problems studied in this work. Theoretical convergence results are obtained in each topic and these are verified in simulation as well as in a robotic test bed developed for this dissertation that we have at UCSD when applicable.

Chapter 1

Introduction

While we are quite far away from the apocalyptic world depicted in the hit series of *Terminator* movies, the best way to summarize the field of distributed motion control to anyone that you meet, whether on a bus, on a busy street, or in a cafe, is: "I am helping to build Skynet." If the reference is lost on said person, "I am studying how to get a group of autonomous agents to deploy over some region in order to maximize some measure of coverage while keeping physical constraints, like battery life, in mind," just does not provoke the same spark of interest in this work.[1] Of course, images of the end of civilization and murderous robots carries quite a grim outlook on this direction of research. Thus, after the *Terminator* reference and perhaps an awkward chuckle, one immediately follows up with the fact that such research has immediate application to improving the quality of human life.

Many applications of distributed motion control that hold promise to progress humanity rather than destroy it come to the immediate forefront. The prospect of autonomous cars on the roadways [10, 60, 23] holds promise to greatly reduce injuries and fatalities that result from traffic accidents. Such technology can also solve, perhaps temporarily, the issue of overcrowding on all the busy streets of this world. Along similar lines, the current method of managing air traffic and providing safety in the air cannot scale at the same rate as the increase in air traffic. Several studies have been done to distribute this crucial task in such a way

[1] Unless you happen to run into a fellow control engineer at such a location.

to insure safety and provide scalability, cf [83, 4, 86]. Yet another example application deals with the streamlining of distribution warehouses using autonomous agents to transport goods; i.e., [88].

The study of distributed motion control fits into the larger field of cooperative networks. Indeed, motion of agents in a network may be applied to the study of interesting behavior found in nature. Immediate examples range from the study of large insect hives to flocks of fish and birds; e.g., [49, 57, 18]. On a much smaller scale, motion coordination may hold the key to explaining certain aggregating behaviors of single-celled organisms such as in [31]. Even more exotic, such ideas may be applied to the modeling and control of nanomotors like those described in [33].

Such examples of mobile networks hold the promise to impact a large number of applications for exploration, environmental monitoring, safety, and recovery operations. A typical task in such applications involves deployment of cooperative agents within the operating environment in order to gather various measurements like temperature or chemical concentration. These sensors, endowed with the ability to move, provide a key advantage over static sensor networks. For example, suppose one wishes to monitor the spread of a oil slick. Ambient water currents may change, and if the network of sensors cannot dynamically adjust to these changes, then the quality sensor readings of the network may deteriorate. This results in more inaccurate estimation of the slick, which may have adverse effects if this information was used to coordinate cleanup efforts. The ability to move and track more important areas of the environment affords sensors networks greater resolution in their measurements.

Furthermore, suppose that one of these oil slick monitoring vehicles drops out due to hardware failure. In the case of a static network of buoys as sensors, there is now a large hole in the sensor network, which leads to higher uncertainty of the spread of the oil slick in the area of that malfunctioning sensor. With the ability to move, however, the other agents in the sensor network can work together to fill in the sensing hole. This capability to dynamically adjust to changing conditions is completely absent in a static network. Indeed, the study of novel schemes to

re-adapt to changing environmental conditions is ongoing, see [30, 75].

Clearly, the task of cooperative deployment for sensing purposes promises many useful applications. This work aims to further develop this idea and build upon existing results in the area of motion control. In the interest of coverage for sensing, one imagines that these autonomous agents wish to collectively maximize some objective function related to the quality of coverage, see for instance [16, 15, 65, 29, 5, 41, 74].

A concurrent interest in miniaturization of such autonomous vehicles brings to light problems that may not be major issues for larger vehicles, see for example [22, 64, 26, 69]. Naturally, smaller vehicles possess less power available for actuation, motion, and communication. Combined with the possibility of operating in harsh environments, simple dynamic models–such as those found in [16, 15]–may no longer be valid.

In essence, we would like to extend the coverage algorithms found within [16] and [15] with additional constraints on motion or additional vehicle dynamics. This dissertation will focus on such deployment and coverage situations where cooperative agents in the network are no longer free to move in certain directions or where cooperative agents have a finite energy supply (implying that they cannot move indefinitely). The problem that we study in this dissertation concerns a network of autonomous agents distributed suboptimally within an operating environment. These agents must reorganize themselves by using only locally sensed or communicated information into a better coverage configuration dictated by a cost/objective function. We depart from previous work by incorporating these additional dynamic constraints into the actual motion control algorithm.

1.1 Summary of results

In what follows, we give a brief overview of the contributions presented in this dissertation.

1.1.1 Distributed deployment that incorporates power consumption

We present a distributed coverage algorithm for mobile sensor networks where agents have limited power to move. Rather than making use of a constrained optimization technique, our approach accounts for power constraints by assigning non-homogeneously time-varying regions to each robot. This leads to novel partitions of the environment into limited-range, generalized Voronoi regions. The motion control algorithms are then designed to ascend the gradient of several types of Locational Optimization functions. In particular, the objective functions reflect the global energy available to the group and different coverage criteria. Under the gradient ascent algorithm, we prove convergence to specific configurations that depend on relative energy values between neighboring agents. Additionally, the control gain used to perform gradient ascent depends on energy content of an agent. This has the effect of limiting the velocity of an agent as it depletes its energy supply, which balances energy consumption across the network during deployment.

1.1.2 Applying hybrid systems analysis to nonholonomic deployment

We explore gradient-descent coverage algorithms for a group of nonholonomic vehicles. Similar to previous approaches, the deployment strategy relies on Locational Optimization techniques and algorithms are distributed in the sense of the Delaunay graph. In order to deal with unicycle dynamics and guarantee performance, we introduce several vehicle modes and integrate them in a hybrid system. We then analyze the algorithms with a recently introduced invariance principle for hybrid systems. Given the unicycle dynamics, and proper choice of switching rules, we prove convergence of the unicycle vehicles to centroidal Voronoi configurations. We also demonstrate via computer simulation that such algorithms successfully converge.

1.1.3 Coverage maximization in a fast flow environment

Here, we study the cooperative motion of a network of autonomous vehicles in a fast flow environment. The magnitude of the flow velocity is assumed to be always greater than the available actuation to each agent. This results in the inability of agents in such a field to choose any direction of travel it wants. Collectively, the agents wish to maximize total coverage area defined as the set of points reachable by any agent within T time. The reachable set of an agent in a fast flow is characterized using optimal control techniques. Specifically, this work addresses the complementary cases where the static flow field is smooth, and where the flow field is piecewise constant. The latter case arises as a proposed approximation of a smooth flow that remains analytically tractable. Furthermore, the techniques used in the piecewise constant flow case enables treatment for obstacles in the environment. In both cases, a gradient ascent method is derived to maximize the total coverage area in a distributed fashion. Through simulations, such a network is able to maximize the coverage area in a fast flow.

1.1.4 Improved deployment via deterministic annealing

We examine a distributed coverage algorithm for a network of mobile agents that leverage an existing deterministic annealing (DA) technique to achieve more optimal convergence values. Typical coverage algorithms employ a pure gradient based method to deploy. However, typical coverage objective functions contain many local minima which may result in sub-optimal final agent configurations. We replicate the results of the classical DA algorithm while imposing a limited-range constraint to sensors. As the temperature is decreased, phase changes lead to a regrouping of agents, which is decided through a distributed task allocation algorithm. While simple gradient descent algorithms are heavily dependent on initial conditions for such non-convex coverage objective functions, annealing techniques are generally less prone to this phenomena. The results of our simulations confirm this fact, as we show in the manuscript.

1.2 Outline

We present the organization of this dissertation in this section, along with a brief summary of each chapter's contents.

- *Chapter 2*: In this chapter, we review definitions and results pertaining to nonlinear systems, Locational Optimization, deterministic annealing, hybrid systems, and distributed algorithms.

- *Chapter 3*: We begin the study of dynamically constrained deployment with the case where agents have a finite energy supply. We develop a method to incorporate this constraint into the deployment problem that is both distributed and able to be computed on-the-fly.

- *Chapter 4*: We move on to case where vehicle dynamics prevent motion in certain directions. Specifically, we give convergence results for the deployment problem when the agents have unicycle dynamics or are Dubin's vehicles. These convergence results rely on the hybrid systems version of the LaSalle invariance principle.

- *Chapter 5*: This chapter considers the problem where external environmental effects substantially influence vehicle dynamics. In particular, we study the deployment problem for agents moving in a flow environment. Instead of treating the environment as a source of small-magnitude noise, we assign dynamics to the flow and those dynamics enter into the equations of motion of the agents.

- *Chapter 6*: Whereas the previous chapters deal with constraints on the agents' motion, this chapter focus on the problem of obtaining final agent configurations that are more robust to perturbations in the choice of initial positions of agents. We employ an annealing technique with the constraint that agents should only use local information.

- *Chapter 7*: While the work in Chapters 3-6 are verified via Matlab simulations, we test the performance of these deployment algorithms on a testbed

constructed at UCSD as part of this dissertation. We detail the equipment and architecture of the testbed and robots, and then present experimental results.

- *Chapter 8*: Finally, in this chapter we present some concluding remarks along with possible lines of future work pertaining to constrained multi agent deployment.

Chapter 2

Notation and Preliminaries

As stated previously, the goal of this work is to present motion control algorithms for a network of cooperative vehicles with additional constraints on dynamics. We first begin by specifying notation and definitions that will be common between the different aspects of cooperative coverage control found in this work. Then, we present some results from previous work that will be used in the subsequent chapters.

2.1 Basic notation

Let Q be a convex polytope in $\mathbb{R}^\Lambda$ including its interior, and let $\|\cdot\|$ denote the Euclidean norm. We will use $\mathbb{R}_{\geq 0}$ to denote the set of non-negative real numbers. A map $\phi\colon Q \to \mathbb{R}_{\geq 0}$, or a *distribution density function*, will represent a measure of *a priori* known information distinguishing zones of Q which are more important than others. Equivalently, we consider Q to be the bounded support of the function ϕ. We denote the interior of a set, S, as S°, its complement as S^C, and its boundary as ∂S. The cardinality of S is denoted as $|S|$.

Definition 1. *A partition of Q is a collection of sets, $\mathcal{A} = \{A_1, \ldots, A_n\}$, such that: (i) ${A_i}^\circ \cap {A_j}^\circ = \emptyset$ for all $i \neq j$ and, (ii) $\bigcup_{i=1}^n A_i = Q$.*

We are interested in developing motion algorithms for a network of N agents moving throughout Q. By convention, we let any quantity associated with agent

i be denoted by the superscript $^{(i)}$. Agent locations are denoted by $p^{(i)}$, $i \in \{1, \ldots, N\}$. We refer to $\mathbb{B}^{(i)}(R)$ to be the set of points within a distance R of $p^{(i)}$. That is, $\mathbb{B}^{(i)}(R) = \{q \in Q \mid \|q - p^{(i)}\| \leq R\}$. We use this shorthand notation since we are interested in these agents performing a global task using only local interactions. Such interactions will be constrained by, for example, a limitation on communication range based on distance. In this particular case, we say that if $p^{(j)} \in \mathbb{B}^{(i)}(R)$, then $p^{(i)}$ can communicate information with $p^{(j)}$. Additionally, we will make use of the *indicator function*, $\mathbf{1}_A \colon Q \to \{0, 1\}$ defined as

$$\mathbf{1}_A(q) = \begin{cases} 1, & q \in A\,, \\ 0, & q \notin A\,. \end{cases}$$

With this notation, we can equivalently say, for example, that agent i can communicate with agent j iff $\mathbf{1}_{[0,R]}(\|p^{(i)} - p^{(j)}\|) = 1$.

Finally, $\mathcal{L}_f X$ denotes the Lie derivative (directional derivative) of the vector field X along the flow f. Since we will be considering the case where f is a real-valued function with X defined over an open subset of $\mathbb{R}^m$, the Lie derivative evaluates to be

$$\mathcal{L}_f X = \sum_{k=1}^{m} \frac{\partial f}{\partial x_k} X_k\,,$$

where each x_k is the typical basis coordinate in $\mathbb{R}^m$. For a set-valued function F that maps an element in X to a subset of Y, we use the notation $F \colon X \rightrightarrows Y$.

2.2 Locational Optimization

The locational optimization problem has its origins in Operations Research. A typical problem to solve is the following: given a large set of customers, where should a business place distribution warehouses in order to minimize the mean delivery time from any particular warehouse? A concrete example of such a problem is the system of Netflix distribution centers that provide prompt delivery of DVDs to households. Now suppose these warehouse locations are mobile, and can react to changing demand in products. Such a hypothetical situation is the basis of the work presented in this dissertation. In this section, we review the work found

in [16, 15] relating to locational optimization and coverage. We will formally define the basic deployment and coverage problem with which we base this dissertation upon.

Let $\phi\colon Q \to \mathbb{R}_{\geq 0}$ be a scalar field with bounded support Q. Here, ϕ represents an *a priori* measure of information on Q–the higher the value of $\phi(q)$, for some $q \in Q$, the more attention that should be afforded to q. Let $P = (p^{(1)}, \ldots, p^{(N)})$ be the aggregation of the agent positions in Q. Take $\mathcal{W} = (W^{(1)}, \ldots, W^{(N)})$ to be any partition of Q. The locational optimization problem involves choosing the placement of sensor locations in order to minimize some cost of coverage. We assume that quality of coverage is a function of the distance between an agent at $p^{(i)}$ and a point $q \in Q$. Naturally, one would expect coverage to get poorer as distance increases. Let $f\colon \mathbb{R} \to \mathbb{R}$ be a monotonically increasing function relating the distance from $p^{(i)}$ to $q \in Q$ that expresses this notion of coverage. Then, the locational optimization problem may be posed as a minimization of

$$\mathcal{H}(P) = \int_Q \min_{p^{(i)}} f(\|p^{(i)} - q\|) dq \,. \tag{2.1}$$

For the purpose of distributed algorithms, it would be helpful to restate (2.1) in terms of the individual contribution that each agent in the network adds to $\mathcal{H}$. For example, if we wish to minimize the mean squared error between sensor placement and areas of interest, we take $f(x) = x^2$. It can be shown, see [56, 16], that such a decomposition of the cost function is possible for this particular choice of sensing function. By introducing the Voronoi partition, $\mathcal{V} = (V^{(1)}, \ldots, V^{(N)})$, where

$$V^{(i)} = \{q \in Q \mid \|q - p^{(i)}\| \leq \|q - p^{(i)}\|, \, \forall \, j \neq i\} \,, \tag{2.2}$$

one can show that the locational optimization cost may be rewritten as

$$\mathcal{H}(P, \mathcal{V}) = \sum_{i=1}^{N} \int_{V^{(i)}} \|q - p^{(i)}\|^2 \phi(q) dq \,. \tag{2.3}$$

In fact, a similar decomposition exists so long as f is non-decreasing and piecewise differentiable, see [15].

Associated with each Voronoi region is it's *mass* and *centroid*, respectively

$$M^{(i)} = \int_{V^{(i)}} \phi(q) dq \,, \qquad C^{(i)} = \int_{V^{(i)}} q\phi(q) dq \,. \tag{2.4}$$

It can be shown, see [56, 16], that if agents are in a *centroidal Voronoi configuration*, where $p^{(i)} = C^{(i)}$ for all i, then the cost function $\mathcal{H}$ is at a local minimum.

With the exception of the results presented in Chapter 6, we shall resort to a simple gradient descent of to minimize our proposed coverage cost functions. In order to compute the gradient, we employ a result from [15] that generalizes the notion of conservation of mass.

Proposition 1. *Let $\{\Omega(x) \subset \mathbb{R}^2 \mid x \in (a,b)\}$ be a piecewise smooth family such that $\Omega(x)$ is strictly star-shaped for all $x \in (a,b)$. Let the function $\phi\colon \mathbb{R}^2 \times (a,b) \to \mathbb{R}$ be continuous on $\mathbb{R}^2 \times (a,b)$, continuously differentiable with respect to its second argument for all $x \in (a,b)$ and almost all $q \in \Omega(x)$, and such that for each $x \in (a,b)$, the maps $q \mapsto \phi(q,x)$ and $q \mapsto \frac{\partial \phi}{\partial x}(q,x)$ are measurable, and integrable on $\Omega(x)$. Then, the function*

$$(a,b) \ni x \mapsto \int_{\Omega(x)} \phi(q,x) dq \tag{2.5}$$

is continuously differentiable and

$$\frac{d}{dx} \int_{\Omega x} \phi(q,x) dq = \int_{\Omega(x)} \frac{\partial \phi}{\partial x}(q,x) dq + \int_{\partial \Omega(x)} \phi(\gamma, x) \hat{\mathbf{n}}_{out}^{\mathsf{T}} \frac{\partial \gamma}{\partial x} d\gamma \,, \tag{2.6}$$

where $\hat{\mathbf{n}}_{out}\colon \partial\Omega(x) \to \mathbb{R}^2$, $q \mapsto \hat{\mathbf{n}}_{out}(q)$ denotes the unit outward pointing normal vector to $\partial\Omega(x)$ at $q \in \partial\Omega(x)$, and $\gamma\colon \mathbb{S}^1 \times (a,b) \to \mathbb{R}^2$ is a parametrization for the family $\{\Omega(x) \subset \mathbb{R}^2 \mid x \in (a,b)\}$.

In the following chapters, we will be basing our coverage cost functions upon these locational optimization cost functions presented here. We will also be incorporating a communication/sensing range constraint such that the algorithms developed in the following chapters will spatially distributed.

2.3 Centralized deterministic annealing

The locational optimization problem bears a striking resemblance to the *clustering* problem in signal processing. One method of solving this clustering problem is through annealing techniques. In Chapter 6, we adapt the Deterministic

Annealing (DA) algorithm proposed by Rose in [68] to make it more distributed. We now briefly describe the minimization process of the DA scheme as well as compare this with the method in [15].

In [68], the end goal is to minimize a *distortion function*,

$$D_0 = \int_Q \phi(q) \sum_{i=1}^{n} P(p_i|q) f_i(\|q - p_i\|) dq \,, \tag{2.7}$$

where $f_i \colon \mathbb{R}_{\geq 0} \to \mathbb{R}$ is a general metric (typically $f_i(x) = x^2$) and $P(p_i|q)$ is the probability of a point q being associated with an agent p_i. However, (2.7) is not directly minimized. The Shannon entropy function is introduced:

$$H_0 = -\int_Q \phi(q) \sum_{i=1}^{n} P(p_i|q) \log P(p_i|q) dq \,, \tag{2.8}$$

and the DA algorithm is a discrete-time algorithm that involves the minimization of the Lagrangian $F_0 = D_0 - TH_0$, where T is the temperature of the system. As temperature decreases, minimizing F_0 becomes more similar to minimizing D_0. The association probabilities $P(p_i|q)$ are derived from $P^*(p_i|q) = \operatorname{argmin}_{P(p_i|q)} F_0$. Then, the resulting $P^*(p_i|q)$ are substituted into F_0 to yield $\hat{F}_0$, and the optimal agent locations are given by $p_i^* = \operatorname{argmin}_{p_i} \hat{F}_0$.

In the case of the squared distance metric, the centralized DA scheme described in [68] gives

$$F_0 = -T \int_Q \phi(q) \log Z_0(q) dq \,, \qquad Z_0(q) = \sum_i \exp\left[-\frac{\|q - p_i\|^2}{T} \right] . \tag{2.9}$$

In contrast the objective in [15] was to minimize (2.7) with trivial association probabilities determined by a Voronoi partition of Q. That is, the probability of $q \in Q$ being associated to p_i is 1 iff q is in its generalized Voronoi region. The introduction of the Shannon entropy into the Lagrangian emphasizes the measure of uncertainty in the assignment of points q to p_i for a given temperature T. As in a physical process, higher temperatures should indicate a higher state of disorder (uncertainty), and this is captured here in the DA formulation. The goal of the DA algorithm is to track the minimum of $\hat{F}$ as temperature is decreased instead of trying to descend the gradient all at once in a typical gradient descent scheme.

Ultimately, as $T \to 0$, $\hat{F} \to D$, and we are minimizing the original distortion function.

As temperature decreases, the system undergoes *phase changes*. During a phase change, the number of agents participating in the optimization algorithm must increase. Roughly speaking, a phase change occurs when a small neighborhood about an equilibrium position p_i^* is no longer attractive in the presence of more than one agent. However, when there is only one agent in this small neighborhood, that agent is attracted to p_i^*. Rose in [68] provides a necessary and sufficient condition to detect phase changes, and we provide an analogous check in the limited-range case.

2.4 Hybrid systems review

As much as we would like to believe, the world that we live is not perfectly smooth and continuous. There are many phenomena that exhibit this sort of discontinuous behavior. For example, imagine a block sliding with friction on a ramp. Eventually, the deceleration from the friction force causes the block to come to a stop. However, such motion cannot be described a single continuous ODE. See [14] for further details on this fascinating topic. In order to extend models to include such phenomena, we resort to *hybrid systems theory*.

In addition to modeling everyday discontinuous behavior, the use of hybrid systems modeling can facilitate and describe the abrupt change in dynamics of a mobile robot. Simple maneuvers such as transitioning from straight motion to turning motion implies a change in a mode of operation. The dynamics of this vehicle depend on the mode of operation that it is in. Hybrid systems modeling is able to capture this, and we take advantage of this in designing and verifying the performance of coverage algorithms for nontrivial dynamics such as a wheeled vehicle.

We will build up to a version of the LaSalle invariance principle for hybrid systems at the end of this section. For continuous dynamical systems, that invariance principle is:

Theorem 2 (LaSalle, see [34], Theorem 4.4)**.** *Let $\Omega \subset D$ be a compact set that is positively invariant with respect to $\dot{x} = f(x)$. Let $V : D \to \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω. Let E be the set of all points in Ω where $\dot{V}(x) = 0$. Let M be the largest invariant set in E. Then every solution starting in Ω approaches M as $t \to \infty$.*

Here we gather some useful results on the modeling and the stability analysis of hybrid automata. The exposition is taken from [25, 72] and included here for completeness.

Definition 2 (Hybrid time domain)**.** *$D \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a* compact hybrid time domain *if*

$$D = \bigcup_{j=0}^{J-1} ([t_j, t_{j+1}], j) \ ,$$

for some finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \cdots \leq t_J$. It is a hybrid time domain *if for all $(T, J) \in D$, $D \cap ([0, T] \times \{0, 1, \ldots, J\})$ is a compact hybrid domain.*

Elements in hybrid time domains can be ordered: we say that $(t_i, j_i) \preceq (t_{i+1}, j_{i+1})$ iff $t_i + j_i \leq t_{i+1} + j_{i+1}$, $j \in \{1, \ldots, J\}$.

Definition 3 (Generalized solution)**.** *A* generalized solution *is a function $x(t, j) \in O$ defined on a hybrid time domain* $\operatorname{dom} x$ *such that:*

1. *on each interval $[t_j, t_{j+1}] \times \{j\} \subset \operatorname{dom} x$ of positive length (so that $t_j < t_{j+1}$) we have*

$$\dot{x}(t, j) \in F(x(t, j)),\ x(t, j) \in A \, ,$$

2. *for each $(t, j) \in \operatorname{dom} x$ such that $(t, j + 1) \in \operatorname{dom} x$, we have*

$$x(t, j + 1) \in G(x(t, j)),\ x(t, j) \in B \, .$$

The set-valued maps $F : O \rightrightarrows \mathbb{R}^N$ and $G : O \rightrightarrows \mathbb{R}^N$ are the flow map and jump map, respectively. The sets $A \subset O$ and $B \subset O$ denote where the state may flow in continuous time, and where the state may make a discontinuous jump, respectively. It is possible for $A \cap B \neq \emptyset$, and in this case, both flowing and jumping may occur. Together, F, G, A, B define a hybrid system, $S = (F, G, A, B)$.

Definition 4 (Complete solution)**.** *A solution* $x : \operatorname{dom} x \to \mathbb{R}^n$ *to a hybrid system* $\mathcal{S}$ *is* complete *if* $\operatorname{dom} x$ *is unbounded.*

As an example, a solution that is defined for all time, $(t, j) \in [0, \infty) \times \{0, 1, \ldots, J\}$ is complete. In addition a solution that exhibits an infinite number of switches in a finite time (a Zeno solution), $(t, j) \in [0, T] \times \mathbb{N}$ is also complete.

Definition 5 (Precompact solution)**.** *A solution* $x : \operatorname{dom} x \to \mathbb{R}^n$ *to a hybrid system* $\mathcal{S}$ *is* precompact *if it is complete and the closure of the range of* x, $\overline{\operatorname{rge} x} \subset O$.

Definition 6 (Weak invariance)**.** *For a hybrid system* $\mathcal{S}$*, the set* $\mathcal{M} \subset O$ *is said to be:*

1. weakly forward invariant *if for each* $x_0 \in \mathcal{M}$ *there exists at least one complete solution* x *with* $x(t, j) \in \mathcal{M}$ *for all* $(t, j) \in \operatorname{dom} x$*;*

2. weakly backward invariant *if for each* $q \in \mathcal{M}, N > 0$ *there exists* $x_0 \in \mathcal{M}$ *and at least one solution* x *such that for some* $(t^*, j^*) \in \operatorname{dom} x$*,* $t^* + j^* > N, x(t^*, j^*) = q$ *and* $x(t, j) \in \mathcal{M}$ *for all* $(t, j) \preceq (t^*, j^*), (t, j) \in \operatorname{dom} x$*;*

3. weakly invariant *if it is both weakly forward invariant and weakly backward invariant.*

Assumption 1 (Basic Conditions)**.** *A hybrid system* $\mathcal{S} = (F, G, A, B)$ *on a state space* $O \subset \mathbb{R}^N$ *satisfies the* Basic Conditions *if:*

(i) $O \subset \mathbb{R}^N$ *is an open set,*

(ii) A *and* B *are relative closed sets in* O*,*

(iii) F *is outer semicontinuous, locally bounded on* O*, and convex* $\forall\, x \in A$*,*

(iv) G *is outer semicontinuous, locally bounded on* O*, and satisfies* $G(x) \subset O$*,* $\forall\, x \in B$*.*

Theorem 3 (Hybrid invariance principle, [72] Corollary 4.3)**.** *Given a hybrid system* $\mathcal{S} = (F, G, A, B)$ *that satisfies the Basic Conditions, suppose that:*

- $V : O \to \mathbb{R}$ *is continuous on* O *and locally Lipschitz on a neighborhood of* A,
- $\mathcal{U} \subset O$ *is nonempty,*
- $u_A(x) = \max_{f \in F(x)} \mathcal{L}_f V(x) \leq 0$ *for all* $x \in A$,
- $u_B(x) = \max_{x^+ \in G(x)} \{V(x^+) - V(x)\} \leq 0$ *for all* $x \in B$.

Let x *be precompact with* $\overline{\operatorname{rge} x} \subset \mathcal{U}$. *Then, for some constant* $r \in V(\mathcal{U})$, x *approaches the largest weakly invariant set in* $V^{-1}(r) \cap \mathcal{U} \cap \left(\overline{u_A^{-1}(0)} \cup u_B^{-1}(0)\right)$.

2.5 Distributed algorithms

Here we make clear the notion of a *distributed algorithm*. Whereas it may be possible to complete a task given all the necessary information, the same task may be accomplished using a subset of that information. For example, it may not be necessary for any single agent in a mobile sensor network to be aware all events in the entire operating environment. The rhetorical question that describes this idea is: why should an individual agent care about what is happening to another agent halfway across the world?

To make the notion of a distributed algorithm more precise, we must start with a few definitions. See [45] and [8] for a complete treatment.

Definition 7 (Directed graph)**.** *A* directed graph *or a* digraph *is a pair* $G = (V, E)$ *where* V *is the* vertex set *and* E *is the* edge set *such that* $E \subseteq V \times V$. *A vertex* $v_i \in V$ *is connected to* $v_j \in V$ *if* $(v_i, v_j) \in E$.

Definition 8 (Undirected graph)**.** *An* undirected graph *or a* graph *is a digraph* $G = (V, E)$ *such that for all* $(v_i, v_j) \in E$, *we have* $(v_j, v_i) \in E$.

For the work presented in this thesis, we will be dealing mainly with networks where inter-agent communication links can be represented as an undirected graph. In other words, we assume that if a communication link exists between agents i and j, then information can flow in both directions.

Definition 9 (Neighbor)**.** *The* neighbor set *of agent i in the graph $G = (V, E)$ is the set of agents $\mathcal{N}^{(i)} = \{v^{(j)} \in V \mid (v^{(i)}, v^{(j)}) \in E\}$.*

The following definition precisely defines what a distributed algorithm is.

Definition 10 (Distributed algorithm)**.** *A* distributed algorithm *$\mathscr{A}$ for a network of agents consists of the sets*

1. *$\mathbb{A}$, the* communication alphabet*, whose elements are called* messages*, and such that $\mathbb{A}$ contains the* `Null` *message,*
2. *the* processor state sets *$W^{(i)}$, $i \in \{1, \ldots, N\}$, and*
3. *the sets of* allowable initial values*, $W_0^{(i)} \subseteq W^{(i)}$, $i \in \{1, \ldots, N\}$;*

and of the maps

1. *$\text{msg}^{(i)} \colon W^{(i)} \times \{1, \ldots, N\} \to \mathbb{A}$, the* message generating function *for $i \in \{1, \ldots, N\}$, and*
2. *$\text{stf}^{(i)} \colon W^{(i)} \times \mathbb{A}^N \to W^{(i)}$, the* state transition functions *for $i \in \{1, \ldots, N\}$.*

In essence, each round of a distributed algorithm consists of one execution of $\text{msg}^{(i)}$, transmission of the generated message, receipt of messages generated by other agents, and finally execution of $stf^{(i)}$ for each $i \in \{1, \ldots, N\}$. Let the set of messages received at each round of the above general algorithm be denoted as $Y^{(i)} \in \mathbb{A}^N$.

Definition 11 (Distributed algorithm over a graph)**.** *We say that an algorithm is* distributed over the graph *$\mathcal{G} = (V, E)$ if the received messages at each round has the following form for all $i, j \in \{1, \ldots, N\}$, $j \notin \mathcal{N}^{(i)}$ implies $Y_j^{(i)} =$* `Null`*.*

We now define some algorithmic primitives that we will employ in describing the algorithms we develop. Messages to a particular agent i can be sent with `send(`a`,` i`)`, where $a \in \mathbb{A}$ and $i \in \{1, \ldots, N\}$. Messages can also be sent to a set of agents, for example `send(`a`,` $\mathcal{N}^{(i)}$`)` sends a to all agents in the neighbor set of i. To receive all such messages we define `Listen` as the function that returns all the messages received during one round of an algorithm.

Let `flood(`a`)` to be an algorithm that floods a message, $a \in \mathbb{A}$, over the entire network. After completion of `flood(`a`)` by agent i, any agent j for which a path exists in the communication graph from i to j will have knowledge of a (possibly the `Null` message). We now formulate this algorithm with the framework presented previously. For proper termination of this algorithm, all agents need to know the total number of agents N. Suppose we are given the alphabet $\mathbb{A}$. Let $W^{(i)}$ consist of the following components: $\{Y^{(i)}, roundCtr\}$, where $Y^{(i)} \in \mathbb{A}^N$, and $roundCtr \in \mathbb{Z}$. Initially, $roundCtr = 0$. The message generating map is:

$$\text{msg}^{(i)} = \begin{cases} \texttt{Null}, & \text{if } Y_j^{(i)} = \texttt{Null},\ \forall j \in \mathcal{N}^{(i)}\,, \\ \texttt{Null}, & \text{if } roundCtr > N\,, \\ a, & \text{otherwise}\,. \end{cases}$$

The state transition map performs the following operations:

$$\text{stf}^{(i)} : \Big\{ roundCtr \leftarrow roundCtr + 1\,.$$

For the sake of brevity, we shall use a more compact form to describe our distributed algorithms in the remainder of this work. For example, the `flood(`a`)` algorithm can be equivalently stated in Algorithm 1. We also define `floodMax(`$x^{(i)}$`)` as a flooding method to determine $\max_{i\in\{1,\ldots,n\}} x^{(i)}$ over the entire network as in [45].

Algorithm 1: A shorter formulation of the `flood` algorithm.

Data: N, total number of agents

$roundCtr \leftarrow 0$

while $roundCtr \leq N$ **do**

  $Y^{(i)} \leftarrow$ `Listen` ()

  if $Y_j^{(i)} \neq$ `Null`, $\forall j$ **then**

    $a^{(i)} \leftarrow a$

  else

    $a^{(i)} \leftarrow$ `Null`

  end

  `Send` $(a^{(i)}, \mathcal{N}^{(i)})$

  $roundCtr \leftarrow roundCtr + 1$

end

Chapter 3

Sensor Deployment under Power Limitations

Unfortunately, a drawback to mobile networks is that of increased power consumption. Power-aware algorithms have been the subject of extensive research in static sensor networks and mobile middleware, see [66, 53]. However, limited work on power constraints and deployment has been done in the multi-vehicle sensor network field. The work of [27] and [85] utilize ordinary Voronoi diagrams and a discrete algorithm to show convergence through simulations. Energy considerations enter in their work as total distance traveled until convergence. Another related result from [52] considers a network of agents performing scan lines over a region of interest with energy and time constraints in mind. More involved vehicle energy dynamics are considered in that work, and they address the relevant problem of speed management as well as optimizing the number of agents necessary to provide adequate coverage in their deployment scheme.

We present an alternative approach to the distributed deployment problem of mobile sensor networks in which agents have limited energy budgets to move. Agents consider their energy level as well as those of their neighbors in determining their coverage ability. We accomplish this by limiting the coverage range of each agent to be proportional to the current energy of each agent. This leads to a novel partition of the environment into limited-range, generalized Voronoi regions that produces a more balanced region assignment. Since agents are continuously

moving, these limited-range coverage constraints are dynamically shrinking. We also adopt energy expenditure dynamics that assume the predominant force acting on the vehicles is drag. Our algorithms seek to maximize objective functions involving: (i) the quantity of area covered, and (ii) the quality of coverage as defined by standard Locational Optimization theory [56].

The new partitions become useful in order to obtain gradient algorithms that guarantee local maximization of the objective functions. The maximization of the objective functions will then require that agents tune their speed as prescribed by the gradient information. The analysis provided here extends and merges previous work in [15], where coverage algorithms for agents with homogeneous, static sensor ranges is studied, and in [36], where energy partitions for coverage are initially explored disregarding energy constraints on mobility. More precisely, the work in [15] is extended to heterogeneous sensing radii that change dynamically as agents spend energy. We include simulations of each algorithm that show that the corresponding objective functions are locally maximized.

In Section 3.1 we define the problem and present the objective functions that we would like to maximize. In Section 3.2, we present the partition necessary to implement the maximization of the functions in a distributed way with energy constraints. We analyze the objective functions in Section 3.3, and present their gradient directions. Section 6.2 introduces a common gradient ascent algorithm with guaranteed performance to apply to each case. In addition, we address some issues that may arise with such flows. We present simulation results in Section 3.5 and discuss the performance of the algorithms.

3.1 Problem definition

Let $P = (p^{(1)}, \ldots, p^{(N)}) \in Q^{\Lambda}$ be the *location of N sensors*, each moving in Q. We interchangeably refer to the elements of the network as sensors, agents, vehicles, or robots. The sensors have an associated energy content $E^{(i)}$ such that $0 \leq E^{(i)} \leq E_{\max}$, for all $i \in \{1, \ldots, N\}$. As agents move, their energy reserve will change. We propose the following simple agent dynamics in the augmented state

$(p^{(i)}, E^{(i)}) \in Q \times \mathbb{R}_{\geq 0}$:

$$\dot{p}^{(i)} = u^{(i)}\,, \qquad \dot{E}^{(i)} = -g^{(i)}(\|\dot{p}^{(i)}\|)\,, \tag{3.1}$$

where $\dot{p}^{(i)}$ denotes the velocity of agent i such that $\|\dot{p}^{(i)}\| \in [0, v_{\max}]$, $u^{(i)}$ is the control input, and $g^{(i)}\colon [0, v_{\max}] \to \mathbb{R}_{\geq 0}$ is any increasing function such that $g^{(i)}(x) = 0$ only at $x = 0$. Intuitively, $g^{(i)}(x)$ captures the fact that energy expenditure increases as velocity increases. We will suppose that $g^{(i)} = g$ for all $i \in \{1, \ldots, N\}$.

We wish to deploy the robots to maximize a performance metric that quantifies coverage and employs the travel ranges for agents. In the most general sense, and motivated by a Locational Optimization approach [56], we seek to maximize a general objective function

$$\mathcal{H}(P, W) = \int_Q \max_{i \in \{1,\ldots,N\}} f^{(i)}(d_{w^{(i)}}(q, p^{(i)}))\phi(q)dq\,, \tag{3.2}$$

where $f^{(i)}\colon \mathbb{R} \to \mathbb{R}$ is a non-increasing function associated with the sensing quality of agent i, and $d_{w^{(i)}}\colon \mathbb{R}^\Lambda \times \mathbb{R}^\Lambda \to \mathbb{R}$ is some metric function weighted by a scalar $w^{(i)} \in \mathbb{R}$, for all $i \in \{1, \ldots, N\}$. Together, $f^{(i)}(d_{w^{(i)}})$ describes the ability of a particular agent to sense points as a function of the generalized distance $d_{w^{(i)}}$. The scalars $w^{(i)}$ will be related to the travel ranges for each agent. Depending on the interpretation of coverage, $\mathcal{H}$ can be further specialized as we see in the following.

3.1.1 Energy-aware coverage by mobile sensors

In [16], the metric $d_{w^{(i)}}(q, p^{(i)})$ in (3.2) was taken to be the square of the Euclidean distance. Thus, $d_{w^{(i)}}(q, p^{(i)}) = \|q - p^{(i)}\|^2$ and there is no weight associated with this metric. We propose a natural extension of the results of [16] by considering a metric weighted by the energy content of each vehicle. As will be explained in a later section, we will choose

$$d_{E^{(i)},\mathrm{pow}}(q, p^{(i)}) = \|q - p^{(i)}\|^2 - (E^{(i)})^2\,, \tag{3.3}$$

$$d_{E^{(i)},\mathrm{mult}}(q, p^{(i)}) = \frac{1}{(E^{(i)})^2}\|q - p^{(i)}\|^2\,, \tag{3.4}$$

called, respectively, the *power-weighted metric* and the *multiplicatively-weighted metric*, see [56].

By modifying the Euclidean distance as in (3.3), (3.4), notice that a point q appears farther away if the energy level of an agent is lower. The associated Locational Optimization function for $f^{(i)}(x) = -x$, becomes:

$$\mathcal{H}_{\text{ea}}(P, W) = \int_Q \max_{i \in \{1,\ldots,N\}} \{-d_{w^{(i)}}(q, p^{(i)})\} \phi(q) dq \,. \tag{3.5}$$

As will be shown, the max function induces a partition of the entire space Q. Under energy-aware coverage, the energy content of each vehicle serves to determine (through $d_{w^{(i)}}$) the points $q \in Q$ that should belong to an agent. The energy-aware method, as its name implies, does not place any hard constraint on coverage, i.e.: all points $q \in Q$ are reachable by any agent, but there is one agent who is closest to q (in the sense of $d_{w^{(i)}}$) and who should be assigned to cover q.

3.1.2 Energy-limited coverage by mobile sensors

Now we consider the situation where energy content places a restriction on which points $q \in Q$ can be reached, in contrast with energy-aware coverage. If an agent has very little energy, it should not be feasible to reach far away points in Q, and we explicitly account for this through energy-limited coverage.

We now formulate the notion of *travel range*, the set of points that an agent can reach if it travels in a straight line at any fixed velocity $\|\dot{p}^{(i)}\| = v^{(i)} \in (0, v_{\max}]$ before running out of energy. Without loss of generality, assume $p^{(i)}(t_0) = 0$ and $E^{(i)}(t_0) > 0$ at some initial time t_0. We wish to find

$$R = \inf_{v^{(i)} \in (0, v_{\max}]} \|p^{(i)}(T)\| \,, \tag{3.6}$$

where $T > 0$ satisfies $E^{(i)}(T) = 0$. From (4.4), and for a constant velocity different from zero, we have that $E^{(i)}(t) - E^{(i)}(t_0) = -g(v^{(i)})(t - t_0)$, and so

$$T = \frac{E^{(i)}(t_0)}{g(v^{(i)})} + t_0 \,. \tag{3.7}$$

Integrating $\dot{p}^{(i)}$ from (4.4), we get $p^{(i)}(t) = (t - t_0)v^{(i)}$. From (3.6) and (3.7), we obtain

$$R = \inf_{v^{(i)} \in (0, v_{\max}]} \left\| \frac{E^{(i)}(t_0) v^{(i)}}{g(v^{(i)})} \right\| \,. \tag{3.8}$$

Note that if $g(x)$ is a polynomial satisfying $g(0) = 0$, then (3.8) is well-defined. In addition, for any other velocity profile $\tilde{v}_i(t)$ along a straight line path, the resulting travel range $\tilde{R}$ is such that $\tilde{R} \geq R$.

Suppose that the forces acting on any individual vehicle come from either the motor output $f^{(i)}$ or drag, $F_i = f^{(i)} - \beta \dot{p}^{(i)}$, where $\beta > 0$. If we consider a simple electric motor, the torque output is proportional to voltage supplied to the motor. Furthermore, the power consumed is proportional to voltage squared. Thus, there is a quadratic relationship between vehicle acceleration and power consumed.

For simplicity, we assume that drag is the predominant force acting on the vehicle throughout deployment. That is, the initial effect of accelerating from rest is small compared to the rest of the deployment process with vehicles traveling at constant speed. The following energy dynamics for each vehicle would then be:

$$\dot{E}^{(i)} = -\|\dot{p}^{(i)}\|^2 = -\|u^{(i)}\|^2 \,, \quad i \in \{1, \ldots, N\} \,, \tag{3.9}$$

or $g(x) = x^2$. For simplicity, let $v_{\max} = 1$, and with the assumed energy dynamics (3.9) along with (3.8), the travel range is $R = E^{(i)}(t_0)$.

Under energy-limited coverage, we propose that in order for an agent i to be able to cover a point $q \in Q$, agent i must be able to reach q with its current energy level. Under the previous assumptions, the range of an agent $i \in \{1, \ldots, N\}$ is then equal to the amount of energy $E^{(i)}$ that it has. Let $\mathbb{B}^{(i)}(E^{(i)})$ be a closed ball centered at $p^{(i)}$ with radius $E^{(i)}$ and let $S^{(i)} = \partial\mathbb{B}^{(i)}(E^{(i)})$ be a sphere centered at $p^{(i)}$ with radius $E^{(i)}$. We will let $\mathcal{R} = Q \cap \bigcup_{i=1}^{N} \mathbb{B}^{(i)}(E^{(i)})$ denote the set of all covered points by the group of agents. We now introduce various objective functions with the energy constraint in mind.

Area Coverage

The simplest problem to solve given the energy-limited constraint is to maximize area covered. Therefore, we can set $f^{(i)}(x) = 1_{[0,E^{(i)}]}(x)$ (i.e., the indicator function of the set $[0, E^{(i)}]$) and $d_{w^{(i)}}(q, p^{(i)}) = \|q - p^{(i)}\|$. Under these assumptions, the general objective function (3.2) becomes:

$$\mathcal{H}_{\text{area}}(P, E) = \int_Q \max_{i \in \{1,\ldots,N\}} 1_{[0,E^{(i)}]}(\|q - p^{(i)}\|)\phi(q)dq = \int_{\mathcal{R}} \phi(q)dq \,. \tag{3.10}$$

Centroidal Coverage

We can combine the energy-limited range with a typical objective function from Locational Optimization to obtain:

$$\mathcal{H}_{\text{cent}}(P,E) = \int_{\mathcal{R}} \max_{i\in\{1,\dots,N\}} \{-d_{E^{(i)}}(q,p^{(i)})\}\phi(q)dq\,. \tag{3.11}$$

This has the interpretation of minimizing the mean distance from a point q to an agent at $p^{(i)}$.

Mixed Coverage

We can combine (5.4) and (3.11) to strike a balance between quantity of coverage and quality of coverage. We introduce two weights, $\kappa_{\text{area}}, \kappa_{\text{cent}}$, to emphasize one over the other. The mixed coverage objective function is:

$$\mathcal{H}_{\text{mix}}(P,E) = \kappa_{\text{area}}\mathcal{H}_{\text{area}}(P,E) + \kappa_{\text{cent}}\mathcal{H}_{\text{cent}}(P,E)\,. \tag{3.12}$$

3.2 Limited-range, generalized Voronoi regions and associated proximity graphs

In order to devise local deployment rules for each agent, it is convenient to assign different regions of the space to them. Similarly as in [15, 36], the regions of dominance should reflect each agent's ability to cover an area. These assignments will also define the graphs that determine the degree of decentralization of the proposed algorithms. In this section we introduce novel partitions of $\mathcal{R}$, $\mathcal{D}_{\text{pow}} = \{D^{(1)}_{\text{pow}}, \dots, D^{(N)}_{\text{pow}}\}$ and $\mathcal{D}_{\text{mult}} = \{D^{(1)}_{\text{mult}}, \dots, D^{(N)}_{\text{mult}}\}$, based on energy motion constraints.

3.2.1 Global partitions determined from the intersection of spheres

Let us consider a configuration like the one shown in Figure 3.1 where in $\mathbb{R}^2$, every two spheres $S^{(i)}$, $S^{(j)}$ have an intersection $S^{(i)} \cap S^{(j)}$ at two points. Similar

configurations can be considered in higher dimensions, but the results that follow are general. A possibility is to define $D_{\text{pow}}^{(i)}$ as the region given by the intersection of $\mathbb{B}^{(i)}(E^{(i)})$ with halfplanes, $H(p^{(i)}, p^{(j)})$, $\forall\, i \neq j$. The halfplanes $H(p^{(i)}, p^{(j)})$ contain $p^{(i)}$ and have as a boundary the line passing through the points in $S^{(i)} \cap S^{(j)}$. Using a halfplane that contains all such points provides a computationally convenient method of assigning regions of dominance. The halfplanes that define boundaries between two regions can also be thought of as the convex hull of $S^{(i)} \cap S^{(j)}$ in the case where $|S^{(i)} \cap S^{(j)}| > 1$. This intuitive construction can be extended to cases where the spheres $S^{(i)}$, $S^{(j)}$ are tangent or have zero intersection through the observation provided in the following lemma.

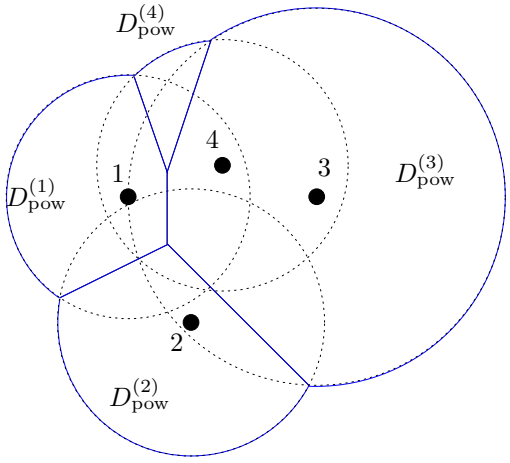


Figure 3.1: Proposed partition of $\mathcal{R}$. The individual spheres are shown in dotted lines, along with the boundaries of $D_{\text{pow}}^{(i)}$ in solid lines.

Lemma 4. *The intersection of halfplanes determined by the intersection of spheres $S^{(i)}$ generated by n agents with positions $p^{(i)}$ and energies $E^{(i)}$ for all $i \in \{1, \ldots, N\}$ induces a partition of $\mathbb{R}^\Lambda$ which is the power-weighted Voronoi diagram (PWVD), $\mathcal{V}_{pow} = (V_{pow}^{(1)}, \ldots, V_{pow}^{(n)})$,*

$$V_{pow}^{(i)} = \{q \in \mathbb{R}^\Lambda \mid \|q - p^{(i)}\|^2 - (E^{(i)})^2 \leq \|q - p^{(j)}\|^2 - (E^{(j)})^2\}\,, \tag{3.13}$$

for all $i \in \{1, \ldots, N\}$ found in [56].

Proof. While the following proof is done in the case of $\mathbb{R}^2$, the extension to $\mathbb{R}^3$ is straightforward, as is the extension to $\mathbb{R}^\Lambda$. Now we examine two intersecting

spheres in order to formulate some expression for the boundary points from the Lemma statement,

$$\mathrm{co}(S^{(i)} \cap S^{(j)}) = \{q \in \mathbb{R}^{\Lambda} \mid \|q - p^{(i)}\|^2 - (E^{(i)})^2 = \|q - p^{(j)}\|^2 - (E^{(j)})^2,\, \forall\, i \neq j\}\,.$$

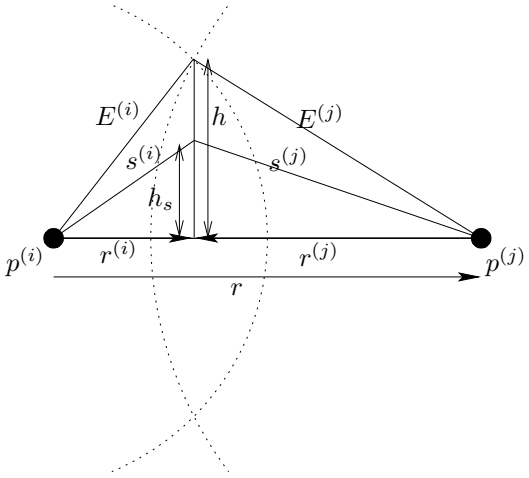


Figure 3.2: Diagram for the derivation of the Voronoi boundary location.

Examining Figure 3.2, we can see that:

$$\begin{cases} r^{(i)} - r^{(j)} = r\,, \\ (E^{(i)})^2 - \|r^{(i)}\|^2 = (E^{(j)})^2 - \|r^{(j)}\|^2\,. \end{cases}$$

Note that $r^{(i)}, r^{(j)}$ and r are vectors, so if $E^{(j)} > \|r\|$ in Figure 3.2, then the above relationships still hold. We solve for $r^{(j)}$ in the first equation and substitute into the second:

$$(E^{(i)})^2 - \|r^{(i)}\|^2 = (E^{(j)})^2 - \|r^{(i)} - r\|^2 = (E^{(j)})^2 - \|r^{(i)}\|^2 + 2r^{\mathsf{T}} r^{(i)} - \|r\|^2\,.$$

Since r and $r^{(i)}$ point in the same direction, their inner product is the product of their individual magnitudes,

$$r^{\mathsf{T}} r^{(i)} = \|r\|\|r^{(i)}\| = \frac{(E^{(i)})^2 - (E^{(j)})^2 + \|r\|^2}{2}$$

$$\Rightarrow \quad r^{(i)} = \frac{(E^{(i)})^2 - (E^{(j)})^2 + \|r\|^2}{2\|r\|} \frac{r}{\|r\|}\,.$$

Note that even though we considered the intersection as in Figure 3.2, we can arrive at a similar conclusion with a different type of intersection (i.e., one leading to an obtuse triangle.)

We now return to the case illustrated in Figure 3.2. Points along each boundary $\mathrm{co}(S^{(i)} \cap S^{(j)})$ satisfy

$$\begin{cases} \|r^{(i)}\|^2 + \|h_s\|^2 = \|s^{(i)}\|^2 , \\ \|r^{(j)}\|^2 + \|h_s\|^2 = \|s^{(j)}\|^2 . \end{cases}$$

Subtracting the two equations, we get

$$\begin{aligned} \|s^{(i)}\|^2 - \|s^{(j)}\|^2 &= \|r^{(i)}\|^2 - \|r^{(j)}\|^2 = \|r^{(i)}\|^2 - \|r^{(i)} - r\|^2 \\ &= \|r^{(i)}\|^2 - \|r^{(i)}\|^2 + 2{r^{(i)}}^{\mathsf{T}} r - \|r\|^2 \\ &= 2\frac{(E^{(i)})^2 - (E^{(j)})^2 + \|r\|^2}{2\|r\|}\|r\| - \|r\|^2 = (E^{(i)})^2 - (E^{(j)})^2 . \end{aligned}$$

This gives our final result,

$$\|s^{(i)}\|^2 - (E^{(i)})^2 = \|s^{(j)}\|^2 - (E^{(j)})^2 .$$

In other words, points $q \in \mathrm{co}(S^{(i)} \cap S^{(j)})$ satisfy $\|q - p^{(i)}\|^2 - (E^{(i)})^2 = \|q - p^{(j)}\|^2 - (E^{(j)})^2$. Note that a set of points, $q \in \mathbb{R}^\Lambda$ that satisfy this property, exists regardless of whether or not $S^{(i)}, S^{(j)}$ intersect. In fact, this boundary requirement is found in [56] as the defining property of the *power-weighted Voronoi partition*, with a weighting factor of $(E^{(i)})^2$ for each generating point $p^{(i)}$. □

It can be seen [56] that the boundary of a PWVD region is made of straight lines in two dimensions, or (hyper-) planes in higher dimensions. Thus, each of the $V^{(i)}_{\mathrm{pow}}$ is convex. By construction, this indeed creates a partition of $\mathbb{R}^\Lambda$. According to [56], generator points $p^{(i)}$ may fall outside their corresponding region $V^{(i)}_{\mathrm{pow}}$. See Figure 3.3 for an illustration of the PWVD, $\mathcal{V}_{\mathrm{pow}}$.

Recall from Section 3.1 that if $v^{(i)} = v^{(j)} = v_{\max}$, then two agents must spend all of their energy to reach a point at the intersection of the energy spheres $S^{(i)} \cap S^{(j)}$. However, both agents do not spend a proportionately equal amount of energy to reach points along the interior of the boundary segments $\Delta^e_{ij} \cap \mathbb{B}^{(i)}(E^{(i)})$. For this to be the case, the property that needs to be satisfied is $\frac{1}{(E^{(i)})^2}\|q - p^{(i)}\|^2 =$

$\frac{1}{E^{(j)}}\|q - p^{(j)}\|^2$. In fact, this corresponds to the *multiplicatively-weighted Voronoi diagram* (MWVD), $\mathcal{V}_{\text{mult}} = \{V^{(1)}_{\text{mult}}, \ldots, V^{(n)}_{\text{mult}}\}$, such that:

$$V^{(i)}_{\text{mult}} = \left\{ q \in \mathbb{R}^{\Lambda} \mid \frac{1}{(E^{(i)})^2}\|q - p^{(i)}\|^2 \leq \frac{1}{(E^{(j)})^2}\|q - p^{(j)}\|^2 \right\}, \quad i \in \{1, \ldots, N\}. \tag{3.14}$$

Thus, given $v^{(i)} = v^{(j)} = v_{\max}$, agents spend proportionately equal amounts of energy to reach boundary points, $\Delta^{(ij)}_{\text{mult}} = V^{(i)}_{\text{mult}} \cap V^{(j)}_{\text{mult}}$. According to [56], for this type of partition, generator points $p^{(i)}$ lie in their regions, which may not be convex, may have holes and be disconnected. The boundaries of these regions are composed of circular arcs.

Figure 3.3 compares the PWVD with the MWVD, when intersected with a convex polytope Q. Notice that the intersected regions of $\mathcal{V}_{\text{pow}}$ are convex whereas the ones associated with $\mathcal{V}_{\text{mult}}$ are not.

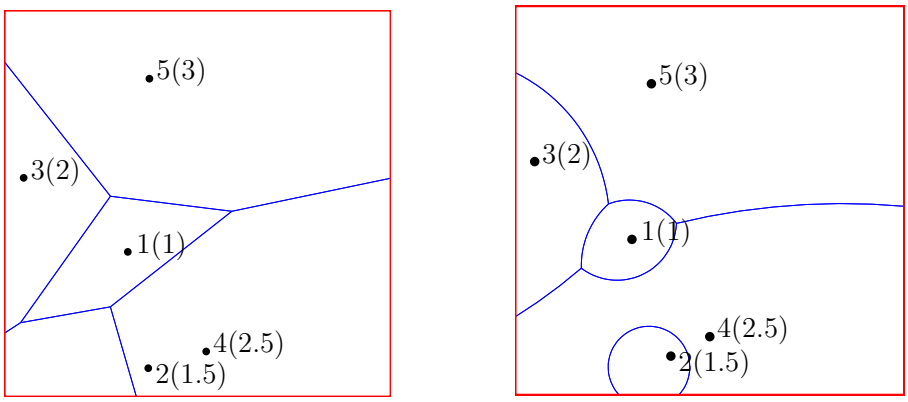


Figure 3.3: Comparison of the PWVD (left) and MWVD (right). Energy contents are shown in parentheses. Observe that agent 2 is outside its region in the power-weighted case.

For the remainder of this chapter, we will focus primarily on results using the MWVD. When, applicable, we state the analogous results using the PWVD without proof.

3.2.2 Limited-range partitions

In order to partition $\mathcal{R} = Q \cap \bigcup_{i=1}^{N} \mathbb{B}^{(i)}(E^{(i)})$, we propose that each element of $\mathcal{D}_{\text{mult}} = \{D_{\text{mult}}^{(1)}, \ldots, D_{\text{mult}}^{(N)}\}$ be defined as $D_{\text{mult}}^{(i)} = Q \cap \mathbb{B}^{(i)}(E^{(i)}) \cap V_{\text{mult}}^{(i)}$, $i \in \{1, \ldots, N\}$. Agent regions of dominance, $D_{\text{mult}}^{(i)}$, will have boundaries that consist of Voronoi face segments $\Delta_{\text{mult}}^{(ij)} \cap \mathbb{B}^{(i)}(E^{(i)}) \cap Q$, boundary segments $\partial Q \cap \mathbb{B}^{(i)}(E^{(i)}) \cap V_{\text{mult}}^{(i)}$, and energy radius arcs. We will refer specifically to those energy arcs in $\partial D_{\text{mult}}^{(i)}$ as $\text{Arcs}(D_{\text{mult}}^{(i)})$.

In addition, the quantities $M_{\text{mult}}^{(i)}$ and $C_{\text{mult}}^{(i)}$ will denote the *mass* and *centroid* of either $V_{\text{mult}}^{(i)}$ or $D_{\text{mult}}^{(i)}$. For example,

$$M_{\text{mult}}^{(i)} = \int_{D_{\text{mult}}^{(i)}} \phi(q) dq \,, \qquad C_{\text{mult}}^{(i)} = \frac{1}{M_{\text{mult}}^{(i)}} \int_{D_{\text{mult}}^{(i)}} q\phi(q) dq \,.$$

It should be clear from the context whether $M_{\text{mult}}^{(i)}$ and $C_{\text{mult}}^{(i)}$ refer to the mass and centroid of $V_{\text{mult}}^{(i)}$ or $D_{\text{mult}}^{(i)}$. In addition, we also define the *moment of inertia* of $V_{\text{mult}}^{(i)}$ or $D_{\text{mult}}^{(i)}$ as,

$$I_{\text{mult}}^{(i)} = \int_{D_{\text{mult}}^{(i)}} \|q - p^{(i)}\|^2 \phi(q) dq \,.$$

As defined, it is not immediately clear that the collections $\mathcal{D}_{\text{mult}}$ are partitions of $\mathcal{R}$. This is proved in the next theorem.

Theorem 5. *Let $\mathcal{D}_{mult} = \{D_{mult}^{(1)}, \ldots, D_{mult}^{(n)}\}$ with $D_{mult}^{(i)} = \mathbb{B}^{(i)}(E^{(i)}) \cap V_{mult}^{(i)} \cap Q$. Then, $\mathcal{D}_{mult}$ are partitions of $\mathcal{R} = Q \cap \bigcup_{i=1}^{N} \mathbb{B}^{(i)}(E^{(i)})$.*

Proof. We prove the result for $\mathcal{D}_{\text{mult}}$, being the proof for $\mathcal{D}_{\text{pow}}$ is analogous. Since the MWVD is a partition of $\mathbb{R}^{\Lambda}$, we have that ${V_{\text{mult}}^{(i)}}^{\circ} \cap {V_{\text{mult}}^{(j)}}^{\circ} = \emptyset$, for all $i \neq j$. Since $D_{\text{mult}}^{(i)} = \mathbb{B}^{(i)}(E^{(i)}) \cap V_{\text{mult}}^{(i)} \cap Q$, then

$${D_{\text{mult}}^{(i)}}^{\circ} = \mathbb{B}^{(i)}(E^{(i)}) \cap {V_{\text{mult}}^{(i)}}^{\circ} = \mathbb{B}^{(i)}(E^{(i)})^{\circ} \cap {V_{\text{mult}}^{(i)}}^{\circ} \cap Q^{\circ} \,.$$

This implies that

$${D_{\text{mult}}^{(i)}}^{\circ} \cap {D_{\text{mult}}^{(j)}}^{\circ} = \mathbb{B}^{(i)}(E^{(i)})^{\circ} \cap {V_{\text{mult}}^{(i)}}^{\circ} \cap \mathbb{B}^{(j)}(E^{(j)})^{\circ} \cap {V_{\text{mult}}^{(j)}}^{\circ} \cap Q = \emptyset \,.$$

Thus we have proved the first defining condition of a partition.

Now we must show $\bigcup_{i=1}^{N} D_{\text{mult}}^{(i)} = Q \cap \bigcup_{i=1}^{N} \mathbb{B}^{(i)}(E^{(i)})$. Expanding the left-hand side,

$$\bigcup_{i=1}^{N} D_{\text{mult}}^{(i)} = \bigcup_{i=1}^{N} (V_{\text{mult}}^{(i)} \cap \mathbb{B}^{(i)}(E^{(i)}) \cap Q) = Q \cap \bigcup_{i=1}^{N} \mathbb{B}^{(i)}(E^{(i)})\,.$$

Thus it is sufficient to show $\bigcup_{i=1}^{N}(V_{\text{mult}}^{(i)} \cap \mathbb{B}^{(i)}(E^{(i)})) = \bigcup_{i=1}^{N} \mathbb{B}^{(i)}(E^{(i)})$. Note also that

$$\begin{aligned}\mathbb{B}^{(i)}(E^{(i)}) &= \mathbb{B}^{(i)}(E^{(i)}) \cap \left(V_{\text{mult}}^{(i)} \cup (V_{\text{mult}}^{(i)})^C\right) \\ &= (\mathbb{B}^{(i)}(E^{(i)}) \cap V_{\text{mult}}^{(i)}) \cup \left(\mathbb{B}^{(i)}(E^{(i)}) \cap (V_{\text{mult}}^{(i)})^C\right)\,.\end{aligned}$$

Taking the union over all i,

$$\bigcup_{i=1}^{N} \mathbb{B}^{(i)}(E^{(i)}) = \left(\bigcup_{i=1}^{N} (\mathbb{B}^{(i)}(E^{(i)}) \cap V_{\text{mult}}^{(i)})\right) \cup \left(\bigcup_{i=1}^{N} \left(\mathbb{B}^{(i)}(E^{(i)}) \cap (V_{\text{mult}}^{(i)})^C\right)\right)\,.$$

If we show that $\left(\mathbb{B}^{(i)}(E^{(i)}) \cap (V_{\text{mult}}^{(i)})^C\right) \subset \bigcup_{j\neq i} D_{\text{mult}}^{(j)}$, then we will have proved the requirement. Let $A^{(i)} = \mathbb{B}^{(i)}(E^{(i)}) \cap (V_{\text{mult}}^{(i)})^C$, and note that $A^{(i)}$ may not be connected, however $\partial A^{(i)} = \bigcup_{j\in\mathcal{N}_{\text{mult}}^{(i)}} (S^{(i)} \cap (V_{\text{mult}}^{(i)})^C) \cup (\mathbb{B}^{(i)}(E^{(i)}) \cap \Delta_{\text{mult}}^{(ij)})$.

Consider a point $q \in A^{(i)}$. Because $\mathcal{V}_{\text{mult}}$ is a partition, there exists a $j_q \neq i$ such that,

$$\frac{1}{(E^{(j)})^2}\|q - p^{(j)}\|^2 \geq \frac{1}{(E^{(j_q)})^2}\|q - p^{(j_q)}\|^2\,, \qquad \forall j \neq j_q\,.$$

On the other hand $q \in A^{(i)}$ implies $\|q - p^{(i)}\|^2 \leq (E^{(i)})^2$. Taking the first equation for $j = i$ and applying the stated inequality, we get

$$\|q - p^{(j_q)}\|^2 \leq 1 \cdot (E^{(j_q)})^2 \leq (E^{(j_q)})^2\,.$$

This implies that $q \in D_{\text{mult}}^{(j_q)}$. Since this argument is valid for any $q \in A^{(i)}$, we have that $A^{(i)} \subseteq \bigcup_{j\neq i} D_{\text{mult}}^{(j)}$. □

The two partitions of $\mathcal{R}$ yield similar results as can be seen in Figure 3.4. Generally speaking, the power-weighted partition $\mathcal{D}_{\text{pow}}$ is a good approximation to the multiplicatively-weighted partition $\mathcal{D}_{\text{mult}}$ if the agents are spaced far enough apart, or if the energy contents of neighbors are similar.

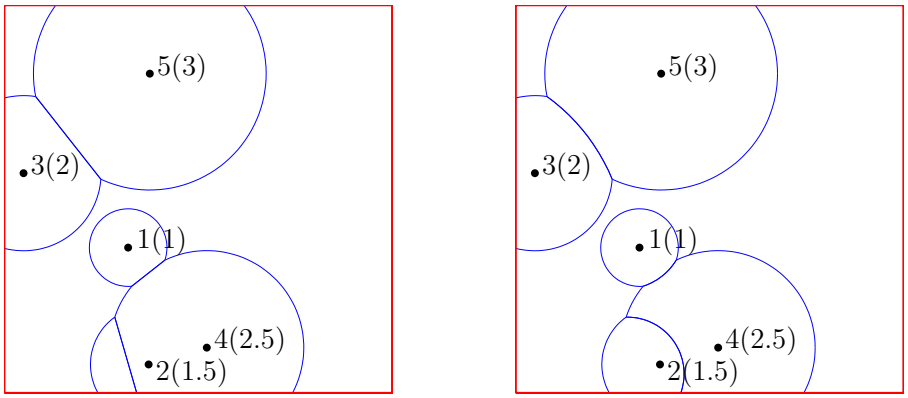


Figure 3.4: Comparison of the limited range power-weighted (left) and multiplicatively-weighted (right) partitions, for the same set of points from Figure 3.3. Energy contents are shown in parentheses.

3.2.3 Communication graphs

As from [56] and [15], these previously described generalized Voronoi regions also have a dual graph. It is over this graph that communication between agents must occur. In what follows, we exactly define the notion of the neighbor of agent i, which is related to the dual graph.

From now on, we adopt the following nomenclature. When two Voronoi regions $V_{\text{mult}}^{(i)}$ and $V_{\text{mult}}^{(j)}$ are adjacent (i.e., they share an edge), $p^{(i)}$ is called a (multiplicatively-weighted) *Voronoi neighbor* of $p^{(j)}$. The set of indices of the MW Voronoi neighbors of $p^{(i)}$ is denoted by $\mathcal{N}_{\text{mult}}^{(i)}$. We define the (i,j) face as $\Delta_{\text{mult}}^{(ij)} = V_{\text{mult}}^{(i)} \cap V_{\text{mult}}^{(j)}$.

Note also that a definition for $\mathcal{N}_{\text{mult}}^{(j)}$ may be obtained from the dual of the MWVD, the MW Delaunay graph, $\mathcal{G}_{\text{D, mult}}$. The graph $(P,E) \to \mathcal{G}_{\text{D, mult}} = (P, \mathcal{E}_{\text{D, mult}}(P,E))$ is a type of proximity graph (see [15]) consisting of the vertices P and the edges $\mathcal{E}_{\text{D, mult}}(P,E)$ such that

$$\mathcal{E}_{\text{D, mult}}(P,E) = \{(p^{(i)}, p^{(j)}) \in P \times P \setminus \text{diag}(P \times P) \mid V_{\text{mult}}^{(i)} \cap V_{\text{mult}}^{(j)} \neq \emptyset\}\,.$$

In this way, we can define the neighbors of $p^{(i)}$ in $\mathcal{G}_{\text{D, mult}}$ as:

$$\mathcal{N}_{\text{mult}}^{(i)} = \{p^{(j)} \in P \mid (p^{(i)}, p^{(j)}) \in \mathcal{E}_{\text{D, mult}}(P,E)\}\,. \tag{3.15}$$

The limited-range counterparts of the MWVD will also have similar dual graphs. These are the energy-limited MW graph $\mathcal{G}_{\text{LD,mult}} = (P, \mathcal{E}_{\text{LD,mult}}(P,E))$. The edge set is defined as:

$$\mathcal{E}_{\text{LD,mult}}(P,E) = \{(p^{(i)}, p^{(j)}) \in P \times P \setminus \text{diag}(P \times P) \mid D^{(i)}_{\text{mult}} \cap D^{(j)}_{\text{mult}} \neq \emptyset\,, \\ \|p^{(i)} - p^{(j)}\| \leq E^{(i)} + E^{(j)}\}\,.$$

This allows the definition of the set of neighbors:

$$\mathcal{N}^{(i)}_{\text{LD,mult}} = \{p^{(j)} \in P \,|\, (p^{(i)}, p^{(j)}) \in \mathcal{E}_{\text{LD,mult}}(P,E)\} = \{p^{(j)} \in P \,|\, D^{(i)}_{\text{mult}} \cap D^{(j)}_{\text{mult}} \neq \emptyset\}\,.$$

3.3 Objective function gradient characterization

In this section we derive the gradient direction for each of the objective functions that we have introduced previously. These gradients define the proper direction of flow in order to optimize coverage. Before we begin, we restate the objective functions from Section 3.1 in a form that facilitates analysis.

The metric that generates the MWVD is determined from the metric (3.4). We present the new forms of (3.5), (5.4), (3.11) (upon which the remaining objective function (3.12) is based) using the MWVD here:

$$\begin{aligned}
\mathcal{H}_{\text{ea}}(P,E) &= \int_Q \max_{i\in\{1,\ldots,N\}} \{-d_{E^{(i)},\text{mult}}(q,p^{(i)})\}\phi(q)dq \\
&= \sum_{i=1}^{N} \int_{V^{(i)}_{\text{mult}}} -\frac{1}{(E^{(i)})^2}\|q-p^{(i)}\|^2\phi(q)dq\,, \qquad (3.16)\\
\mathcal{H}_{\text{area}}(P,E) &= \int_{\mathcal{R}} \phi(q)dq = \sum_{i=1}^{N} \int_{D^{(i)}_{\text{mult}}} \phi(q)dq\,, \qquad (3.17)\\
\mathcal{H}_{\text{cent}}(P,E) &= \int_{\mathcal{R}} \max_{i\in\{1,\ldots,N\}} \{-d_{E^{(i)},\text{mult}}(q,p^{(i)})\}\phi(q)dq \\
&= \sum_{i=1}^{N} \int_{D^{(i)}_{\text{mult}}} -\frac{1}{(E^{(i)})^2}\|q-p^{(i)}\|^2\phi(q)dq\,. \qquad (3.18)
\end{aligned}$$

Before computing the gradients we would like to note the following result.

Lemma 6. *The objective functions* (3.16)–(3.18) *are continuously differentiable with respect to* $p^{(i)}$ *and* $E^{(i)}$.

Proof. This is a result of Proposition 1.6 in [15]. Even though regions of MWVD may not be star-shaped, each individual region $V^{(i)}_{\text{mult}} \cap Q$ and $D^{(i)}_{\text{mult}}$ can be composed of a finite union of star-shaped sets, which makes such regions fall under the scope of the result from [15]. Additionally, the function $\mathcal{H}$ is piecewise smooth since it is smooth over each $V^{(i)}_{\text{mult}}$, a further requirement in [15]. We also present ϕ to be the bounded support of Q, thus it is integrable over Q. This satisfies all the requirements of Proposition 1.6, so then the objective functions (3.16)–(3.18) are are continuously differentiable. □

We now state the gradient expressions for these functions.

Proposition 7. *Consider the objective functions $\mathcal{H}_{area}$, $\mathcal{H}_{cent}$, and $\mathcal{H}_{mix}$ based on* (3.16), (3.17), *and* (3.18) *using the partition $\mathcal{D}_{mult}$ and metric* (3.4). *Let $X = (X_1, \ldots, X_N)$ be a general vector field where $X_i = (X_{p^{(i)}}, X_{E^{(i)}}) : Q \times \mathbb{R} \to \mathbb{R}^\Lambda \times \mathbb{R}$ for all $i \in \{1, \ldots, N\}$. Then, the Lie derivatives of $\mathcal{H}_{ea}$, $\mathcal{H}_{area}$, $\mathcal{H}_{cent}$, and $\mathcal{H}_{mix}$ along the flow X are given by:*

$$\mathcal{L}_X \mathcal{H}_{ea} = \sum_{i=1}^{N} \frac{2M^{(i)}_{mult}}{(E^{(i)})^2}(C^{(i)}_{mult} - p^{(i)})^\mathsf{T} X_{p^{(i)}} + \frac{2I^{(i)}_{mult}}{(E^{(i)})^3} X_{E^{(i)}}\,, \tag{3.19}$$

$$\begin{aligned}\mathcal{L}_X \mathcal{H}_{area} = &\sum_{i=1}^{N} \left(\int_{\text{Arcs}(D^{(i)}_{mult})} \phi(\gamma^{(i)})[\hat{\mathbf{n}}^\mathsf{T}_{out}(\gamma^{(i)})]^\mathsf{T} d\gamma^{(i)} \right) X_{p^{(i)}} \\ &+ \left(\int_{\text{Arcs}(D^{(i)}_{mult})} \phi(\gamma^{(i)}) d\gamma^{(i)} \right) X_{E^{(i)}}\,,\end{aligned} \tag{3.20}$$

$$\mathcal{L}_X \mathcal{H}_{cent} = \sum_{i=1}^{N} \left(\frac{2M^{(i)}_{mult}}{(E^{(i)})^2}(C^{(i)}_{mult} - p^{(i)})^\mathsf{T} - \frac{\partial \mathcal{H}_{area}}{\partial p^{(i)}} \right) X_{p^{(i)}} \tag{3.21}$$

$$+ \left(\frac{2I^{(i)}_{mult}}{(E^{(i)})^3} - \frac{\partial \mathcal{H}_{area}}{\partial E^{(i)}} \right) X_{E^{(i)}}\,, \tag{3.22}$$

$$\mathcal{L}_X \mathcal{H}_{mix} = \kappa_{area} \mathcal{L}_X \mathcal{H}_{area} + \kappa_{cent} \mathcal{L}_X \mathcal{H}_{cent}\,, \tag{3.23}$$

Proof. We will use the conservation of mass law in [15]. We will perform analysis of the objective functions (3.16), (3.17), and (3.18), and note that the gradient formulations for the remaining function $\mathcal{H}_{\text{mix}}$ follows from those results.

Energy-aware (3.16)

We have that:

$$\begin{aligned}
\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} &= \sum_{j=1}^{N} \frac{\partial}{\partial p^{(i)}} \int_{V^{(j)}_{\text{mult}}} -\frac{1}{(E^{(j)})^2} \|q - p^{(j)}\|^2 \phi(q) dq \\
&= \sum_{j=1}^{N} \int_{V^{(j)}_{\text{mult}}} -\frac{1}{(E^{(j)})^2} \frac{\partial}{\partial p^{(i)}} \left[\|q - p^{(j)}\|^2\right] \phi(q) dq \\
&\quad + \int_{\partial V^{(j)}_{\text{mult}}} -\frac{\|\gamma^{(j)} - p^{(j)}\|^2}{(E^{(j)})^2} \phi(\gamma^{(j)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)}) \frac{\partial \gamma^{(j)}}{\partial p^{(i)}} d\gamma^{(j)} \\
&= 2 \int_{V^{(i)}_{\text{mult}}} \frac{1}{(E^{(i)})^2} (q - p^{(i)})^{\mathsf{T}} \phi(q) dq \\
&\quad + \int_{\partial V^{(i)}_{\text{mult}}} -\frac{(E^{(i)})^2}{(E^{(i)})^2} \phi(\gamma^{(i)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)}) \frac{\partial \gamma^{(i)}}{\partial p^{(i)}} d\gamma^{(i)} \\
&\quad + \sum_{j \in \mathcal{N}_i} \int_{\partial V^{(j)}_{\text{mult}}} -\frac{(E^{(j)})^2}{(E^{(j)})^2} \phi(\gamma^{(j)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)}) \frac{\partial \gamma^{(j)}}{\partial p^{(i)}} d\gamma^{(j)} \,.
\end{aligned}$$

Since we take the derivative with respect to $p^{(i)}$, certainly the boundary $V^{(i)}_{\text{mult}}$ changes. These boundaries are shared by neighboring Voronoi regions $V^{(j)}_{\text{mult}}$, $j \in \mathcal{N}^{(i)}_{\text{mult}}$. Therefore, along the boundaries $\partial V^{(j)}_{\text{mult}}$ that each region $V^{(j)}_{\text{mult}}$ has in common with $V^{(i)}_{\text{mult}}$, there is a change as well. However, the parts of the boundaries $\partial V^{(j)}_{\text{mult}}$, $j \in \mathcal{N}^{(i)}_{\text{mult}}$ that are not shared with region $V^{(i)}_{\text{mult}}$ are not affected by a change in $p^{(i)}$, so the derivative $\frac{\partial \gamma^{(j)}}{\partial p^{(i)}} = 0$ along those sections. For the sections of $\partial V^{(j)}_{\text{mult}}$ that are shared with $V^{(i)}_{\text{mult}}$, the derivatives are equal: $\frac{\partial \gamma^{(j)}}{\partial p^{(i)}} = \frac{\partial \gamma^{(i)}}{\partial p^{(i)}}$. Additionally, the metric that defines the partition are equal at the boundaries $d_{E^{(i)},\text{mult}} = d_{E^{(j)},\text{mult}}$. The only difference is the direction of the outwards-pointing normal vector, and we have $\hat{\mathbf{n}}_{\text{out}}(\gamma^{(i)}) = -\hat{\mathbf{n}}_{\text{out}}(\gamma^{(j)})$ along the shared boundary for all $j \in \mathcal{N}^{(i)}_{\text{mult}}$. Therefore, the integrals along shared boundaries vanish, leaving the final result:

$$\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} = 2 \int_{V^{(i)}_{\text{mult}}} \frac{1}{(E^{(i)})^2} (q - p^{(i)})^{\mathsf{T}} \phi(q) dq \,.$$

A similar procedure leads to the energy derivative:

$$\begin{aligned}
\frac{\partial \mathcal{H}_{\text{ea}}}{\partial E^{(i)}} &= \sum_{j=1}^{N} \frac{\partial}{\partial E^{(i)}} \int_{V^{(j)}_{\text{mult}}} -\frac{1}{(E^{(j)})^2}\|q - p^{(j)}\|^2 \phi(q) dq \\
&= \sum_{j=1}^{N} \int_{V^{(j)}_{\text{mult}}} -\|q - p^{(j)}\|^2 \phi(q) \frac{\partial}{\partial E^{(i)}} \left[\frac{1}{(E^{(j)})^2}\right] dq \\
&\quad + \int_{\partial V^{(j)}_{\text{mult}}} -\frac{\|\gamma^{(j)} - p^{(j)}\|^2}{(E^{(j)})^2} \phi(\gamma^{(j)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)}) \frac{\partial \gamma^{(j)}}{\partial E^{(i)}} d\gamma^{(j)} \\
&= 2 \int_{V^{(i)}_{\text{mult}}} \frac{1}{(E^{(i)})^3}\|q - p^{(i)}\|^2 \phi(q) dq \\
&\quad + \sum_{j \in \mathcal{N}_i} \int_{\partial V^{(j)}_{\text{mult}}} -\frac{(E^{(j)})^2}{(E^{(j)})^2} \phi(\gamma^{(j)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)}) \frac{\partial \gamma^{(j)}}{\partial E^{(i)}} d\gamma^{(j)} \,.
\end{aligned}$$

Again, the integral along the boundaries vanish, giving the final result:

$$\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} = \frac{2M^{(i)}_{\text{mult}}}{(E^{(i)})^2}(C^{(i)}_{\text{mult}} - p^{(i)})^{\mathsf{T}}, \quad \frac{\partial \mathcal{H}_{\text{ea}}}{\partial E^{(i)}} = \frac{2I^{(i)}_{\text{mult}}}{(E^{(i)})^3} \,.$$

Area coverage (3.17)

We must calculate the expression for $\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}$ and $\frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}}$. Applying the conservation of mass law once again, we have

$$\begin{aligned}
\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} &= \frac{\partial}{\partial p^{(i)}} \sum_{j=1}^{N} \int_{D^{(j)}_{\text{mult}}} \phi(q) dq \\
&= \sum_{j=1}^{N} \int_{D_j} \frac{\partial}{\partial p^{(i)}} \phi(q) dq + \sum_{j=1}^{N} \int_{\partial D^{(j)}_{\text{mult}}} \phi(\gamma^{(j)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)}) \frac{\partial \gamma^{(j)}}{\partial p^{(i)}} d\gamma^{(j)} \\
&= \int_{\partial D^{(i)}_{\text{mult}}} \phi(\gamma^{(i)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)}) \frac{\partial \gamma^{(i)}}{\partial p^{(i)}} d\gamma^{(i)} \\
&\quad + \sum_{j \in \mathcal{N}^{(i)}_{\text{LD,mult}}} \int_{\partial D^{(j)}_{\text{mult}}} \phi(\gamma^{(j)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)}) \frac{\partial \gamma^{(j)}}{\partial p^{(i)}} d\gamma^{(j)} \,.
\end{aligned}$$

Each region may have boundaries composed of circular energy arcs or MW Voronoi faces. Over the set of shared faces $\Delta^{(ij)}_{\text{mult}}$, for all $j \in \mathcal{N}^{(i)}_{\text{LD,mult}}$, note that the normal vectors $\hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)}) = -\hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)})$. Therefore, the integrals along shared boundaries vanish, leaving the integration over the energy arcs of $D^{(i)}_{\text{mult}}$,

$$\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} = \int_{\text{Arcs}(D^{(i)}_{\text{mult}})} \phi(\gamma^{(i)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)}) \frac{\partial \gamma^{(i)}}{\partial p^{(i)}} d\gamma^{(i)} \,. \tag{3.24}$$

From the definition of $D^{(i)}_{\text{mult}}$, $\text{Arcs}(D^{(i)}_{\text{mult}})$ are a fixed distance with respect to the generating point, $p^{(i)}$. Then, the change in position of the arcs is equal to the change in position of the point $p^{(i)}$. Therefore, the derivative $\frac{\partial \gamma^{(i)}}{\partial p^{(i)}} = I$. From (3.24) we then have,

$$\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} = \int_{\text{Arcs}(D^{(i)}_{\text{mult}})} \phi(\gamma^{(i)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)}) d\gamma^{(i)} \,. \tag{3.25}$$

Similarly, we compute the derivative with respect to $E^{(i)}$:

$$\frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}} = \int_{\text{Arcs}(D^{(i)}_{\text{mult}})} \phi(\gamma^{(i)}) \frac{\partial \gamma^{(i)}}{\partial E^{(i)}} \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)}) d\gamma^{(i)} \,.$$

Note that the boundary, $\gamma^{(i)}$, grows and shrinks proportional to the normal as $E^{(i)}$ changes. Therefore, $\frac{\partial \gamma^{(i)}}{\partial E^{(i)}} = \hat{\mathbf{n}}_{\text{out}}(\gamma^{(i)})$ and

$$\frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}} = \int_{\text{Arcs}(D^{(i)}_{\text{mult}})} \phi(\gamma^{(i)}) \|\hat{\mathbf{n}}_{\text{out}}(\gamma^{(i)})\|^2 d\gamma^{(i)} = \int_{\text{Arcs}(D^{(i)}_{\text{mult}})} \phi(\gamma^{(i)}) d\gamma^{(i)} \,. \tag{3.26}$$

Centroidal coverage (3.18)

First take the derivative wrt $p^{(i)}$ using conservation of mass law:

$$\begin{aligned}
\frac{\partial \mathcal{H}_{\text{cent}}}{\partial p^{(i)}} &= \sum_{j=1}^{N} \frac{\partial}{\partial p^{(i)}} \int_{D^{(j)}_{\text{mult}}} -\frac{1}{(E^{(j)})^2} \|q - p^{(j)}\|^2 \phi(q) dq \\
&= \sum_{j=1}^{N} \int_{D^{(j)}_{\text{mult}}} -\frac{1}{(E^{(j)})^2} \frac{\partial}{\partial p^{(i)}} \left[\|q - p^{(j)}\|^2 \right] \phi(q) dq \\
&\quad + \int_{\partial D^{(j)}_{\text{mult}}} -\frac{\|\gamma^{(j)} - p^{(j)}\|^2}{(E^{(j)})^2} \phi(\gamma^{(j)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)}) \frac{\partial \gamma^{(j)}}{\partial p^{(i)}} d\gamma^{(j)} \\
&= 2 \int_{D^{(i)}_{\text{mult}}} \frac{1}{(E^{(i)})^2} (q - p^{(i)})^{\mathsf{T}} \phi(q) dq \\
&\quad + \int_{\partial D^{(j)}_{\text{mult}}} -\frac{\|\gamma^{(j)} - p^{(j)}\|^2}{(E^{(j)})^2} \phi(\gamma^{(j)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)}) \frac{\partial \gamma^{(j)}}{\partial p^{(i)}} d\gamma^{(j)} \\
&\quad + \sum_{j \in \mathcal{N}_i} \int_{\partial D^{(j)}_{\text{mult}}} -\frac{(E^{(j)})^2}{(E^{(j)})^2} \phi(\gamma^{(j)}) \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)}) \frac{\partial \gamma^{(j)}}{\partial p^{(i)}} d\gamma^{(j)} \,.
\end{aligned}$$

For similar reasons as in the energy-aware case, the integrals along the shared Voronoi faces vanish, leaving only the integral along the energy radius arcs centered

at $p^{(i)}$. We then have

$$\begin{aligned}\frac{\partial \mathcal{H}_{\text{cent}}}{\partial p^{(i)}} &= 2\int_{D^{(i)}_{\text{mult}}} \frac{1}{(E^{(i)})^2}(q-p^{(i)})^{\mathsf{T}}\phi(q)dq - \int_{\text{Arcs}(D^{(i)}_{\text{mult}})} \hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)})\phi(q)d\gamma^{(i)} \\ &= \frac{2M^{(i)}_{\text{mult}}}{(E^{(i)})^2}(C^{(i)}_{\text{mult}}-p^{(i)})^{\mathsf{T}} - \frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\,.\end{aligned}$$

The energy derivative follows similarly,

$$\begin{aligned}\frac{\partial \mathcal{H}_{\text{cent}}}{\partial E^{(i)}} &= \sum_{j=1}^{N}\frac{\partial}{\partial E^{(i)}}\int_{D^{(j)}_{\text{mult}}} -\frac{1}{(E^{(j)})^2}\|q-p^{(j)}\|^2\phi(q)dq \\ &= \sum_{j=1}^{N}\int_{D^{(j)}_{\text{mult}}} -\|q-p^{(j)}\|^2\phi(q)\frac{\partial}{\partial E^{(i)}}\left[\frac{1}{(E^{(j)})^2}\right]dq \\ &\quad + \int_{\partial D^{(j)}_{\text{mult}}} -\frac{\|\gamma^{(j)}-p^{(j)}\|^2}{(E^{(j)})^2}\phi(\gamma^{(j)})\hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(j)})\frac{\partial \gamma^{(j)}}{\partial E^{(i)}}d\gamma^{(j)} \\ &= \frac{2I^{(i)}_{\text{mult}}}{(E^{(i)})^3} - \int_{\text{Arcs}(D^{(i)}_{\text{mult}})}\|\hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)})\|^2\phi(\gamma^{(i)})d\gamma^{(i)} = \frac{2I^{(i)}_{\text{mult}}}{(E^{(i)})^3} - \frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}}\,.\end{aligned}$$

□

Remark 8. The area coverage result (3.20) and its PWVD analogue can be seen as a generalization of a limited sensing coverage problem in [15] for a network of heterogeneous sensors with different and constant sensing ranges R_i. The corresponding sensing range would satisfy $R_i = (E^{(i)})^2$. However, since this sensing range is fixed, the Lie derivative will not have a second term involving energy. We would simply have

$$\frac{d\mathcal{H}_{\text{area}}}{dt} = \sum_{i=1}^{N}\left(\int_{\text{Arcs}(D^{(i)}_{\text{mult}})}\phi(\gamma^{(i)})[\hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)})]^{\mathsf{T}}d\gamma^{(i)}\right)\dot{p}^{(i)}\,,$$

which extends the result found in [15]. •

Remark 9. In the area-maximizing case, $\mathcal{H}_{\text{area}}$, the gradient $\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}$ points outwards in the direction that is the most "open". However, for the "centroidal" case, $\mathcal{H}_{\text{cent}}$, the opposite term appears. For most choices of ϕ, this has a detrimental effect on coverage as seen through simulations. Instead of deploying out over the domain Q, agents tend to remain motionless or collect together. However, the mixed case, $\mathcal{H}_{\text{mix}}$ with $\kappa_{\text{area}} = \kappa_{\text{cent}}$ negates this effect, and the gradient direction points exactly

towards the centroids, $C^{(i)}_{\text{mult}}$, for all i. It seems that a more natural extension of the energy-aware case to the limited-range MWVD is $\mathcal{H}_{\text{mix}}$ instead of $\mathcal{H}_{\text{cent}}$. •

The following is the analogue of Proposition 7 for the PWVD case. Take caution, that the functions $\mathcal{H}$ in what follows are not the same as the $\mathcal{H}$ found in Proposition 7.

Proposition 10. *Given the objective functions $\mathcal{H}_{ea}$, $\mathcal{H}_{area}$, $\mathcal{H}_{cent}$, and $\mathcal{H}_{mix}$, and a general vector field $X = (X_1, \ldots, X_n)$ where $X_i = (X_{p^{(i)}}, X_{E^{(i)}}) : Q \times \mathbb{R} \to \mathbb{R}^\Lambda \times \mathbb{R}$ for all $i \in \{1, \ldots, N\}$, the Lie derivatives of $\mathcal{H}_{ea}$, $\mathcal{H}_{area}$, $\mathcal{H}_{cent}$, and $\mathcal{H}_{mix}$ are*

$$\mathcal{L}_X \mathcal{H}_{ea} = \sum_{i=1}^{N} \left(2M^{(i)}_{pow}(C^{(i)}_{pow} - p^{(i)})^\mathsf{T}\right) X_{p^{(i)}} + 2E^{(i)} M^{(i)}_{pow} X_{E^{(i)}} \,, \tag{3.27}$$

$$\begin{aligned}\mathcal{L}_X \mathcal{H}_{area} = &\sum_{i=1}^{N} \left(\int_{\text{Arcs}(D^{(i)}_{pow})} \phi(\gamma^{(i)}) [\hat{\mathbf{n}}^\mathsf{T}_{out}(\gamma^{(i)})]^\mathsf{T} d\gamma^{(i)} \right) X_{p^{(i)}} \\ &+ \left(\int_{\text{Arcs}(D^{(i)}_{pow})} \phi(\gamma^{(i)}) d\gamma^{(i)} \right) X_{E^{(i)}} \,,\end{aligned} \tag{3.28}$$

$$\mathcal{L}_X \mathcal{H}_{cent} = \sum_{i=1}^{N} \left(2M^{(i)}_{pow}(C^{(i)}_{pow} - p^{(i)})^\mathsf{T}\right) X_{p^{(i)}} + \left(2E^{(i)} M^{(i)}_{pow}\right) X_{E^{(i)}} \,, \tag{3.29}$$

$$\mathcal{L}_X \mathcal{H}_{mix} = \kappa_{area} \mathcal{L}_X \mathcal{H}_{area} + \kappa_{cent} \mathcal{L}_X \mathcal{H}_{cent} \,, \,. \tag{3.30}$$

We refer the reader to the proof of Proposition 7, as it is similar. ■

3.4 Gradient-ascent deployment algorithms

Once we have computed the gradient directions for each objective function, we will apply a gradient-ascent control algorithm for each case. The resulting control algorithms are extensions of Lloyd's algorithm for quantization [43], and are distributed in the sense of a (limited) Delaunay graph. Consider (4.4) with

$$\begin{aligned}\dot{p}^{(i)} &= k(p^{(i)}, E^{(i)}) \,\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \,, \\ \dot{E}^{(i)} &= -\|\dot{p}^{(i)}\|^2 \,,\end{aligned} \tag{3.31}$$

for all $i \in \{1, \ldots, N\}$, where the saturation function is

$$\text{sat}(v) = \begin{cases} v & , \|v\| \leq 1 \,, \\ \frac{v}{\|v\|} & , \|v\| > 1 \,. \end{cases}$$

Here the control gain $k(p^{(i)}, E^{(i)}) \geq 0$ serves to modulate the velocity of each agent along its gradient climbing path.

Let $\mathcal{H}$ be any one of the objective functions analyzed in the previous section, with the exception of $\mathcal{H}_{\text{cent}}$ as noted in Remark 9. Now we analyze the time evolution of the corresponding objective function $\mathcal{H}$ with respect to (3.31). We adopt the shorthand notation $k^{(i)} = k(p^{(i)}, E^{(i)})$. Combining the gradient direction with the time derivatives above, we get the following time derivative:

$$\begin{aligned} \frac{d\mathcal{H}}{dt} &= \sum_{i=1}^{N} \frac{\partial \mathcal{H}}{\partial p^{(i)}} \dot{p}^{(i)} + \frac{\partial \mathcal{H}}{\partial E^{(i)}} \dot{E}^{(i)} \\ &= \sum_{i=1}^{N} k^{(i)} \frac{\partial \mathcal{H}}{\partial p^{(i)}} \cdot \text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) - (k^{(i)})^2 \frac{\partial \mathcal{H}}{\partial E^{(i)}} \left\| \text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \right\|^2 \\ &= \sum_{i=1}^{N} k^{(i)} \, \text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \cdot \left(\frac{\partial \mathcal{H}}{\partial p^{(i)}} - k^{(i)} \frac{\partial \mathcal{H}}{\partial E^{(i)}} \, \text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right)\right) \,. \end{aligned} \tag{3.32}$$

Remark 11. Non-smooth dynamics are also possible:

$$\begin{aligned} \dot{p}^{(i)} &= k(p^{(i)}, E^{(i)}) \frac{\frac{\partial \mathcal{H}}{\partial p^{(i)}}}{\left\| \frac{\partial \mathcal{H}}{\partial p^{(i)}} \right\|} \,, \\ \dot{E}^{(i)} &= -\|\dot{p}^{(i)}\|^2 = -k^2(p^{(i)}, E^{(i)}) \,. \end{aligned}$$

Doing so would require the non-smooth analysis techniques found in [13]. We would, however, arrive at the same convergence conclusions found in the next subsections. •

Remark 12. The properties of the gradient of $\mathcal{H}$ as in Proposition 7 make the associated law (3.31) distributed in the sense of the corresponding Delaunay graph. The information that an agent needs to implement (3.31) is only the position and energies of neighbors in the associated Delaunay communication graph. Moreover, in the case of the energy-limited algorithm, it is spatially distributed due to the travel range constraint.

With this information, an agent can correctly construct its region, $D^{(i)}_{\text{mult}}$. A sufficient condition to achieve this is if $p^{(j)}$ can transmit to $p^{(i)}$ when $\|p^{(i)} - p^{(j)}\| \leq 2E_{\max}$ for all $j \neq i$. With this communication requirement, the control law described in (3.31) is spatially distributed over the graph $\mathcal{G}_{\text{LD,mult}}$. •

3.4.1 Optimal gain selection

We wish that $\frac{d\mathcal{H}}{dt} \geq 0$ since we are maximizing the objective function. We now derive a sufficient condition for k and also present an optimal choice for k.

Lemma 13. *Given the model* (4.4), (3.31)*, and an objective function* $\mathcal{H}$*, the latter is maximized if*

$$0 \leq k(p^{(i)}, E^{(i)}) \leq \frac{\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \cdot \frac{\partial \mathcal{H}}{\partial p^{(i)}}}{\left\|\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right)\right\|^2 \frac{\partial \mathcal{H}}{\partial E^{(i)}}}\,, \tag{3.33}$$

for all $i \in \{1, \ldots, N\}$*. An optimal choice of* $k(p^{(i)}, E^{(i)})$ *is then*

$$k^*(p^{(i)}, E^{(i)}) = \frac{1}{2} \frac{\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \cdot \frac{\partial \mathcal{H}}{\partial p^{(i)}}}{\left\|\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right)\right\|^2 \frac{\partial \mathcal{H}}{\partial E^{(i)}}}\,. \tag{3.34}$$

Proof. In order for $\frac{d\mathcal{H}}{dt} \geq 0$ we require that each summand of (3.32) be positive. Since $k(p^{(i)}, E^{(i)}) \geq 0$, we must have

$$\frac{\partial \mathcal{H}}{\partial p^{(i)}} \cdot \text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) - k^{(i)} \frac{\partial \mathcal{H}}{\partial E^{(i)}} \left\|\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right)\right\|^2 \geq 0\,.$$

Since $\frac{\partial \mathcal{H}}{\partial E^{(i)}} \geq 0$ in Propositions 10 and 7 (except for (3.22)), the formula in (3.33) follows.

We are free to choose k subject to (3.33). In particular, we would like to maximize each summand of (3.32) for each i. Taking the derivative with respect to k, we have

$$\frac{\partial \mathcal{H}}{\partial p^{(i)}} \cdot \text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) - 2k^{(i)*} \frac{\partial \mathcal{H}}{\partial E^{(i)}} \left\|\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right)\right\|^2 = 0\,,$$

and the critical point condition for $k^{(i)}$ follows. □

Remark 14. Different energy dynamics can be considered. That is, the consideration of $g^{(i)}(x)$ different from x^2 is possible as long as $g^{(i)}(0) = 0$ and the $g^{(i)}$ are sufficiently smooth. •

Although using (3.34) provides the quickest rate of convergence, it may not be the best. Consider the situation shown in Figure 3.5, for the case where the objective function is (5.4). Agent 4 has a small arc component compared to its entire boundary. However with the area coverage gradient (3.28), the optimal gain (3.34) remains constant since $\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}, \frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}} \to 0$ at the same rate. For this reason, we would like $k^{(i)}$ to be chosen by the following constrained optimization way: maximize each summand of (3.32) subject to $k^{(i)} \leq \frac{\partial \mathcal{H}}{\partial E^{(i)}}$. Notice that this quantity is of the form $f(k^{(i)}) = k^{(i)}(c_1 - k^{(i)}c_2)$, a concave parabola. With this constraint, the optimum ${k^{(i)}}^*$ is then

$$ {k^{(i)}}^* = \min \left\{ \frac{1}{2} \frac{\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \cdot \frac{\partial \mathcal{H}}{\partial p^{(i)}}}{\left\| \text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \right\|^2 \frac{\partial \mathcal{H}}{\partial E^{(i)}}}, \frac{\partial \mathcal{H}}{\partial E^{(i)}} \right\} . \tag{3.35} $$

A simulation in Section 3.5 further discusses this choice. Another possibility is ${k^{(i)}}^*$,

$$ {k^{(i)}}^* = \min \left\{ \frac{1}{2} \frac{\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \cdot \frac{\partial \mathcal{H}}{\partial p^{(i)}}}{\left\| \text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \right\|^2 \frac{\partial \mathcal{H}}{\partial E^{(i)}}}, \frac{E^{(i)}}{E_{\max}} \right\} . \tag{3.36} $$

This choice of $\frac{E^{(i)}}{E_{\max}}$ is motivated from [36]. With ${k^{(i)}}^* \leq \frac{E^{(i)}}{E_{\max}}$, it can be shown that an individual agent will not run out of energy in finite time. For the MWVD case, this factor appears naturally in the optimal gain (3.34). We can impose a similar constraint for the base return flow such that if $\|p^{(i)} - q_0\| \leq E^{(i)}$, then the vehicle at $p^{(i)}$ will always be able to return to q_0. We can choose

$$ {k^{(i)}}^* = \min \left\{ \frac{1}{2} \frac{\text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \cdot \frac{\partial \mathcal{H}}{\partial p^{(i)}}}{\left\| \text{sat}\left(\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right) \right\|^2 \frac{\partial \mathcal{H}}{\partial E^{(i)}}}, 1 \right\} . \tag{3.37} $$

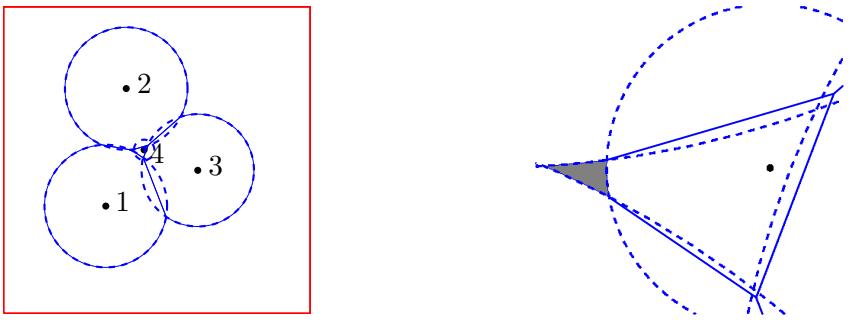
This is because when ${k^{(i)}}^* = 1$, $\|\dot{p}^{(i)}\| \leq 1$, which insures that a vehicle can reach any point in its travel range given in Section 3.1.

Using the method of Lagrange multipliers (e.g., in [46]) we have the following condition for extremum points along the boundary $k^{(i)} = \frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}}$,

$$\nabla \left[k(p^{(i)}, E^{(i)}) \left\| \frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} \right\| - k^2(p^{(i)}, E^{(i)}) \frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}} \right] = \lambda \nabla \left[k^{(i)} - \frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}} \right],$$

$$\left\| \frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} \right\| - 2k \frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}} = \lambda,$$

$$\lambda = \left\| \frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} \right\| - 2 \left(\frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}} \right)^2.$$



(a) (b)

Figure 3.5: Motivation for modified gain. Agent number 4 in (a), has a small arc component compared to its total boundary mass. In (b) we show a magnified region about agent 4. In this magnified picture, the dotted lines are the energy radii of the four respective agents, and the solid lines are the Voronoi faces. The shaded region is not in the region of any agent.

With $k^{(i)}(p^{(i)}, E^{(i)})$ chosen as in (3.35), we have two possible flow cases:

$$\dot{p}^{(i)} = \frac{\frac{\partial \mathcal{H}}{\partial p^{(i)}}}{2 \frac{\partial \mathcal{H}}{\partial E^{(i)}}}, \qquad \dot{p}^{(i)} = \frac{\partial \mathcal{H}}{\partial E^{(i)}} \frac{\frac{\partial \mathcal{H}}{\partial p^{(i)}}}{\left\| \frac{\partial \mathcal{H}}{\partial p^{(i)}} \right\|}. \tag{3.38}$$

The former quantity may pose some problems as $\frac{\partial \mathcal{H}}{\partial E^{(i)}} \to 0$. However, if this is the case, then the choice of $k^{(i)*}$ results in the latter flow case, which is bounded. In fact, it can be shown that in the area coverage (5.4) and mixed coverage (3.12) cases, the original optimal gain, (3.34) is upper bounded by $\frac{1}{2}$ and 1, respectively.

3.4.2 Convergence analysis

We now replace the general objective, $\mathcal{H}$, with the functions developed in Section 3.3. The proof of the following theorem relies on the LaSalle invariance principle; see [34].

We omit the results of using $\mathcal{H}_{\text{cent}}$ (3.18) for reasons stated in Remark 9.

Theorem 15 (Critical configurations and convergence, energy-aware MWVD)**.** *The critical points of a gradient ascent flow characterized by* (3.31) *and appropriate choice of k using an objective function $\mathcal{H}_{ea}$ are configurations where each agent is either:*

1. *located at the centroid, $p^{(i)} = C^{(i)}_{mult}$,*
2. *has no energy, $E^{(i)} = 0$.*

Agents approach these critical configurations as $t \to \infty$.

Proof. From Lemma 13, $\frac{d\mathcal{H}}{dt} \geq 0$ using $\mathcal{H} = \mathcal{H}_{\text{ea}}$. Thus we need to characterize the critical points where $\frac{d\mathcal{H}_{\text{ea}}}{dt} = 0$. From (3.32), this is the case if for all $i \in \{1, \ldots, N\}$, either:

1.
$$k^{(i)} = \frac{1}{2} \frac{\text{sat}\left(\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\right) \cdot \frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}}{\left\|\text{sat}\left(\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\right)\right\|^2 \frac{\partial \mathcal{H}_{\text{ea}}}{\partial E^{(i)}}} = 0\,, \tag{3.39}$$

2.
$$\text{sat}\left(\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\right) = \text{sat}\left(\frac{2M^{(i)}_{\text{mult}}}{(E^{(i)})^2}(C^{(i)}_{\text{mult}} - p^{(i)})^{\mathsf{T}}\right) = 0\,, \tag{3.40}$$

3. or
$$\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} - k^{(i)} \frac{\partial \mathcal{H}_{\text{ea}}}{\partial E^{(i)}} \text{sat}\left(\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\right) = 0\,. \tag{3.41}$$

Case (3.39) implies either $\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} = 0$ or $E^{(i)} = 0$. Case (3.40) also implies that $\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} = 0$, which occurs if either $M^{(i)}_{\text{mult}} = 0$ or $p^{(i)} = C^{(i)}_{\text{mult}}$. The case where $M^{(i)}_{\text{mult}} = 0$ implies that the region $V^{(i)}_{\text{mult}} \cap Q = \emptyset$.

To analyze case (3.41), we consider a situation when $\left\|\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\right\| \leq 1$ and when $\left\|\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\right\| > 1$. For the former case,

$$\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\left[1 - \frac{1}{2\frac{\partial \mathcal{H}_{\text{ea}}}{\partial E^{(i)}}}\frac{\partial \mathcal{H}_{\text{ea}}}{\partial E^{(i)}}\right] = 0\,,$$
$$\frac{1}{2}\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} = 0\,.$$

This implies that $\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} = 0$, a case addressed in (3.40). The latter case, when $\left\|\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\right\| > 1$ leads to:

$$\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\left[1 - \frac{\left\|\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\right\|}{2\frac{\partial \mathcal{H}_{\text{ea}}}{\partial E^{(i)}}}\frac{\frac{\partial \mathcal{H}_{\text{ea}}}{\partial E^{(i)}}}{\left\|\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}}\right\|}\right] = 0\,,$$
$$\frac{1}{2}\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} = 0\,.$$

This again implies that $\frac{\partial \mathcal{H}_{\text{ea}}}{\partial p^{(i)}} = 0$, addressed earlier.

We now characterize the invariant configurations, when $\dot{p}^{(i)} = 0$ for all $i \in \{1, \ldots, N\}$. This corresponds to the cases (3.39) and (3.40). The largest invariant set contained in $S = \{(p_1, \ldots, p_N) \mid \dot{\mathcal{H}}_{ea} = 0\}$ is S itself. By the LaSalle invariance principle, system configurations will asymptotically approach S. □

Theorem 16 (Critical configurations and convergence, energy-limited MWVD)**.** *The critical points of a gradient ascent flow characterized by* (3.31) *and appropriate choice of k using an objective function $\mathcal{H} \in \{\mathcal{H}_{area}, \mathcal{H}_{mix}\}$ are configurations where each agent either satisfies $\frac{\partial \mathcal{H}}{\partial p^{(i)}} = 0$, or has no energy, $E^{(i)} = 0$. The statement $\frac{\partial \mathcal{H}}{\partial p^{(i)}} = 0$ has the following meanings:*

1. *the vehicle cannot further locally increase its coverage area when using $\mathcal{H}_{area}$,*
2. *the vehicle has reached the centroid of $D^{(i)}_{mult}$ when using $\mathcal{H}_{mix}$.*

Agents approach these critical configurations as $t \to \infty$.

Proof. The results for each deployment objective function will follow from the LaSalle invariance principle; see [34]. We will present the proof for the Area

coverage objective function (3.17) and note that the proofs for the remaining cases are similar.

The region Q is positively invariant since agents cannot leave it. Also, using ${k^{(i)}}^*$ from (3.34) into (3.32) results in $\dot{\mathcal{H}}_{\text{area}} \geq 0$ in Q. We now compute the critical points where $\dot{\mathcal{H}}_{\text{area}} = 0$. From (3.32), this occurs when either ${k^{(i)}}^* = 0$, or $\text{sat}\left(\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right) = 0$, or when

$$\left(\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} - {k^{(i)}}^* \frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}} \,\text{sat}\left(\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right)\right) = 0\,, \tag{3.42}$$

for all $i \in \{1, \ldots, N\}$.

If ${k^{(i)}}^* = 0$, we divide the problem into the case where $\left\|\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right\| \leq 1$ or where $\left\|\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right\| > 1$. If the former is true, then

$${k^{(i)}}^* = \frac{1}{2\frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}}} = 0\,.$$

However, since $\frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}}$ is bounded, this is not possible. If $\left\|\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right\| > 1$,

$${k^{(i)}}^* = \frac{\left\|\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right\|}{2\frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}}} = 0\,,$$

and $\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} = 0$. This implies that

$$\int_{\text{Arcs}(D^{(i)}_{\text{mult}})} \phi(\gamma^{(i)})[\hat{\mathbf{n}}^{\mathsf{T}}_{\text{out}}(\gamma^{(i)})]^{\mathsf{T}} d\gamma^{(i)} = 0\,.$$

Assuming that $\phi > 0$ on Q, the first equation implies that either $\text{Arcs}(D^{(i)}_{\text{mult}}) = \emptyset$ or that the integral over the boundary is balanced in all directions. The latter equation also implies that $\text{Arcs}(D^{(i)}_{\text{mult}}) = \emptyset$. This occurs if either $E^{(i)} = 0$, or $D^{(i)}_{\text{mult}}$ is composed entirely of Voronoi edges (the boundary of Q counts as a Voronoi edge), or $D^{(i)}_{\text{mult}} = \emptyset$.

When $\text{sat}\left(\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right) = 0$, then $\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} = 0$, and we have addressed this situation.

Now suppose that (3.42) is true. We substitute (3.34) into (3.42) and consider the case where $\left\|\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right\| \leq 1$ or where $\left\|\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right\| > 1$. When the former is

true,

$$\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} - \frac{1}{2\frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}}} \frac{\partial \mathcal{H}_{\text{area}}}{\partial E^{(i)}} \left(\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} \right) = 0\,,$$

$$\frac{1}{2} \frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} = 0\,,$$

which implies $\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} = 0$. Similarly $\left\| \frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} \right\| > 1$, implies $\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} = 0$. For both cases we conclude that $\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}} = 0$, a situation that we have addressed earlier.

We now characterize the set of invariant configurations. The system is positively invariant if $\dot{p}^{(i)} = 0$ for all $i \in \{1, \ldots, N\}$. From (3.31), this occurs when ${k^{(i)}}^* = 0$ or $\text{sat}\left(\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}\right) = 0$ for all $i \in \{1, \ldots, N\}$. Therefore, the invariant configurations are exactly these where $\dot{\mathcal{H}}_{\text{area}} = 0$, which have been described. By LaSalle's invariance principle [34], the agents will asymptotically approach this set of configurations. □

We now state the PWVD analogues to the convergence theorems stated previously.

Theorem 17 (Critical configurations and convergence, energy-aware PWVD)**.** *The critical points of a gradient ascent flow characterized by* (3.31) *and appropriate choice of k using an objective function $\mathcal{H}_{ea}$ are configurations where each agent is either:*

1. *located at the centroid, $p^{(i)} = C^{(i)}_{pow}$,*
2. *has an empty region, $V^{(i)}_{pow} \cap Q = \emptyset$,*
3. *has no energy, $E^{(i)} = 0$.*

Agents approach these critical configurations as $t \to \infty$. We refer to the proof of Theorem 15 as it is similar. ■

Theorem 18 (Critical configurations and convergence, energy-limited PWVD)**.** *The critical points of a gradient ascent flow characterized by* (3.31) *and appropriate choice of k using an objective function $\mathcal{H} \in \{\mathcal{H}_{area}, \mathcal{H}_{cent}, \mathcal{H}_{mix}\}$ are configurations where each agent either satisfies $\frac{\partial \mathcal{H}}{\partial p^{(i)}} = 0$, or its region $D^{(i)}_{pow} = \emptyset$. The statement $\frac{\partial \mathcal{H}}{\partial p^{(i)}} = 0$ has the following meanings:*

1. *the vehicle cannot further locally increase its coverage area when using* $\mathcal{H}_{area}$,

2. *vehicle* i *is located at the centroid of* $D_{pow}^{(i)}$ *when using* $\mathcal{H}_{cent}$, *and*

3. *the vehicle has reached a balance between maximizing area covered and remaining close to the centroid of* $D_{pow}^{(i)}$ *when using* $\mathcal{H}_{mix}$.

Agents approach these critical configurations as $t \to \infty$. *We refer to the proof of Theorem 16 as it is similar.* ■

3.4.3 An algorithmic formulation for deployment

Here we outline the necessary functions that an individual agent must perform in order to distributively complete the deployment task. We make the following assumptions about the agents:

1. each agent i can communicate with other agents in the ball $\mathbb{B}^{(i)}(2E_{\max})$, where $E_{\max} = \max_i\{E^{(i)}\}$,

2. communication is done synchronously, and

3. each agent has knowledge of $\phi(q)$ for $q \in Q$.

Remark 19. Results from [84] and, more relevantly [16], address the issue of asynchronous communication and convergence of gradient algorithms similar to the one presented here. An extension of these results is possible for the algorithms demonstrated in this manuscript. •

Each vehicle must be capable of performing computations regarding general Voronoi regions and their limited-range versions. We denote by `computeRegion()` the function that takes in neighbor positions and energies in $\mathcal{N}_{\text{mult}}^{(i)}$ (resp. $\mathcal{N}_{\text{LD,mult}}^{(i)}$) and computes the regions $V_{\text{mult}}^{(i)}$ (resp. $D_{\text{mult}}^{(i)}$). We then define `computeGradient()` to be the function that takes the result of `computeRegion()` and returns the gradient direction (3.31). Algorithm 2 outlines the deployment process.

Algorithm 2: Distributed deployment algorithm

Data: $\phi(q)$ for $q \in Q$

while true do `// this repeats indefinitely`

send$((p^{(i)}, E^{(i)}), \mathcal{N}^{(i)}_{\text{mult}})$

$\{p^{(j)}\}, \{E^{(j)}\} \leftarrow$ listen()

$myRegion \leftarrow$ computeRegion$(\{p^{(j)}\}, \{E^{(j)}\})$

$\dot{p}^{(i)}$, $\dot{E}^{(i)} \leftarrow$ computeGradient$(myRegion, \phi)$

end

3.5 Simulations

In this section, we present simulation results for the three coverage objectives. First, however, we will address the motivation for choosing ${k^{(i)}}^*$ from (3.35) over (3.34). In this simulation, $n = 8$ agents were initialized at random initial positions with $E^{(i)} = 10$ for $i \in \{1, \ldots, 8\}$. The agents remain in the invariant domain $Q = [0, 15] \times [0, 15] \subset \mathbb{R}^2$ with $\phi(x, y) = 1 + 10 \exp[-\frac{1}{9}((x-10)^2 + (y-10)^2)]$. The agents maximized the area coverage objective function (5.4).

The use of ${k^{(i)}}^*$ from (3.35) demonstrates some advantages over (3.34) in the simulation of Figure 3.6. In Figure 3.6(a), agent 4 finishes with almost no energy, while the same agent has significantly more energy in Figure 3.6(b). In addition, all 8 agents were deployed in Figure 3.6(a) while only 5 agents left the starting location in Figure 3.6(b).

In this section, we present simulations for the MWVD-based deployment algorithms. We also present and compare these simulations with the PWVD-based deployment algorithms for completeness. We additionally demonstrate how a carefully chosen objective function can result in other behavior, such as remaining close to a starting point.

We would like to note the following general observations regarding deployment algorithms based on the PWVD versus the MWVD. When the task is to maximize coverage quality as in (3.5) and (3.12), agents using the MWVD tend to expend energy in a more balanced fashion. In other words, the final energy values of each agent have lower variance at convergence under the MWVD-based

deployment than under the PWVD-based deployment. This phenomena is not observed under area-maximizing coverage (5.4). Additional observations regarding the PWVD are addressed in the following subsections.

Additionally, when maximizing coverage quality, agent final positions are clustered around the parts of Q where ϕ has large value. This is not the case in the area-maximizing case since the associated metric does not place any weight on the importance of distance to points inside an agent's coverage region.

3.5.1 Energy-aware coverage case

Here, $n = 12$ agents remain in the invariant domain $Q = [0, 10] \times [0, 10] \subset \mathbb{R}^2$ with a density function ϕ composed of 4 Gaussian distributions (see Figure 3.7). The density function used was

$$\phi(q) = 1 + 10 \left[e^{-\frac{\|q-r_1\|^2}{9}} + e^{-\frac{\|q-r_2\|^2}{2}} + e^{-\frac{\|q-r_3\|^2}{2}} + e^{-\|q-r_4\|^2} \right],$$

where $r_1 = (8, 8)$, $r_2 = (8, 2)$, $r_3 = (8, 4)$ and $r_4 = (3, 7)$. Agents started at random positions in the lower-left corner, with $E^{(i)} = 5$ for $i \in \{1, \ldots, 12\}$. The agents followed the gradient ascent control law in (3.31), and used (3.34). Energy consumption profiles of the deployment process are shown in Figure 3.8.

Remark 20. Using the PWVD, it is possible for an agent i to be outside of its region of dominance $V^{(i)}_{\text{pow}}$, i.e.: $p^{(i)} \notin V^{(i)}_{\text{pow}}$. It is oftentimes the case where $V^{(i)}_{\text{pow}} \cap Q = \emptyset$ when $p^{(i)}$ is close to the boundary of Q, and $E^{(i)} < E^{(j)}$ for $j \in \mathcal{N}^{(i)}_{\text{pow}}$ This phenomena can be observed in the simulation of Figure 3.7 (left). From an energy perspective, the use of these sensors is wasted, since they are pushed away and do not play any role in the coverage task. This raises the question of how to characterize the number of sensors needed to cover a given region under the PWVD-based algorithms. •

The undesirable behavior as mentioned in Remark 20 of the PWVD energy-aware algorithm is avoided when using the MWVD based algorithm, since it is a property of the MWVD that $p^{(i)} \in V^{(i)}_{\text{mult}}$. In this regard, the MWVD forces the participation of all sensors in the coverage task. The use of the PWVD would require an assessment of how many sensors are enough to solve the coverage task.

We ran 20 simulations each of the PWVD and MWVD energy-aware algorithms. Histograms for objective function value at the end of each simulation and energy spent by each agent are compared in Figures 3.9 and 3.10. Because these are gradient algorithms, they converge to many local minima of the objective function, as can be seen in Figure 3.10.

3.5.2 Area coverage case

We now examine the area coverage case, (5.4) between the PWVD and MWVD partitions. The system of agents was initialized identically to the energy-aware simulations, and we compare the performance between the PWVD-based deployment and the MWVD-based deployment. Agent trajectories are shown in Figure 5.26 and energy consumption profiles of the deployment process are shown in Figure 3.12.

Here we do not see the undesirable effect in the PWVD case (Figure 5.26, left) of agents lying along the boundary of Q without a region of dominance. This is because the gradient, $\frac{\partial \mathcal{H}_{\text{area}}}{\partial p^{(i)}}$, depends only on the presence of energy arcs for each vehicle. As agents move closer to the boundary ∂Q, the arcs tend to disappear since the distance from $p^{(i)}$ to the boundary is less than $E^{(i)}$. This prevents agents from getting squeezed into the boundary. Both algorithms perform similarly, deploying to cover almost all of Q in both cases.

We then performed 50 simulations of both the MWVD and PWVD area coverage algorithms, starting from random initial positions. Histograms of energy consumption as well as objective function value at the end of each simulation are compared in Figures 3.13 and 3.14.

3.5.3 Mixed coverage case

The third simulation presents the mixed coverage case (3.12) between the PWVD and MWVD with identical initial conditions as before. The area and centroidal components carried equal weight, $\kappa_{\text{area}} = \kappa_{\text{cent}} = 1$ from (3.12). Agent trajectories are shown in Figure 3.19 and energy consumption profiles of the de-

ployment process are shown in Figure 3.16.

We again notice the return of the same undesirable phenomena mentioned in the energy-aware simulations for the energy-limited PWVD simulation in Figure 3.19, left. Despite this, the mixed coverage algorithm resulted in agent positions that are more collocated with dense regions of ϕ for both the PWVD and MWVD cases, as compared to Figure 5.26.

We performed 50 simulations from random initial positions and compare the statistics in Figures 3.17 and 3.18. Note that while both deployment cases maximize the function (3.12), the PWVD uses the metric (3.3) and the MWVD uses the metric (3.4), which accounts for the different ranges in final cost values.

3.5.4 Base return coverage case

Here we explore a new objective function. Rather than repeating the gradient analysis and convergence result, we only provide the cost function:

$$\mathcal{H}_{\text{base}}(P, E) = \mathcal{H}_{\text{area}}(P, E) - \sum_{i=1}^{N} \rho(q_0, p^{(i)}, E^{(i)}) \,, \tag{3.43}$$

with $\rho(q_0, p^{(i)}, E^{(i)}) = -\exp[\|q_0 - p^{(i)}\|^2 - (E^{(i)})^2]$ and $q_0 = (0, 0)$. The effect of the additional term ρ acts to heavily penalize agents if they venture too far from the fixed point q_0.

We now provide a simulation using both the PWVD and MWVD partitions. Agents remain in $Q = [-5, 5] \times [-5, 5] \subset \mathbb{R}^2$ and $\phi(x, y) = 1$. We initialized $n = 8$ agents randomly around the origin with $E^{(i)} = 3$ for all $i \in \{1, \ldots, 8\}$.

The simulations have almost identical results, with the exception of different partitioning schemes. The agents manage to stay close enough to the origin when fully deployed. As expected, agents maximized coverage in the beginning. However as energy levels decreased and distance from the origin increased, the penalty term of the gradient dominates.

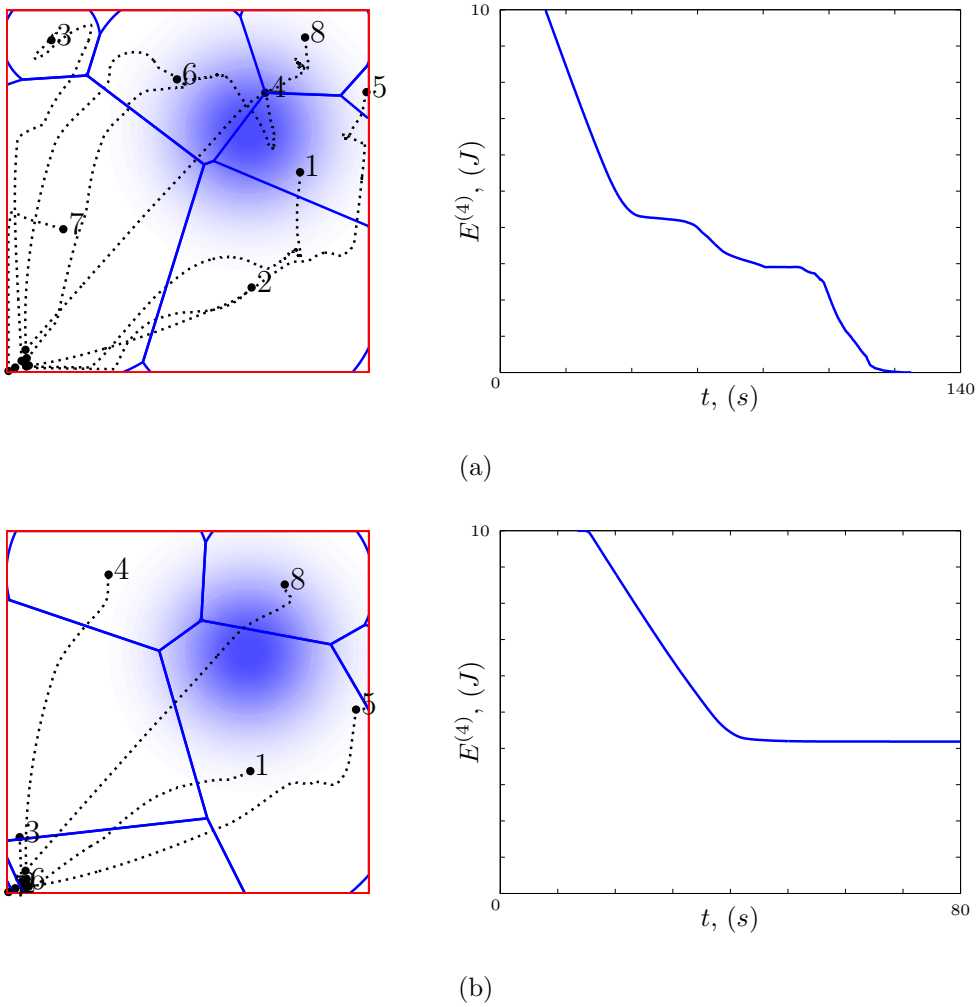


Figure 3.6: Comparison between the performance of $k^{(i)^*}$ from (3.34), (a); and from (3.35), (b). Agent paths and final configurations are shown at left, and the energy level of agent 4 is plotted at right. Shaded regions indicate a high value of ϕ.

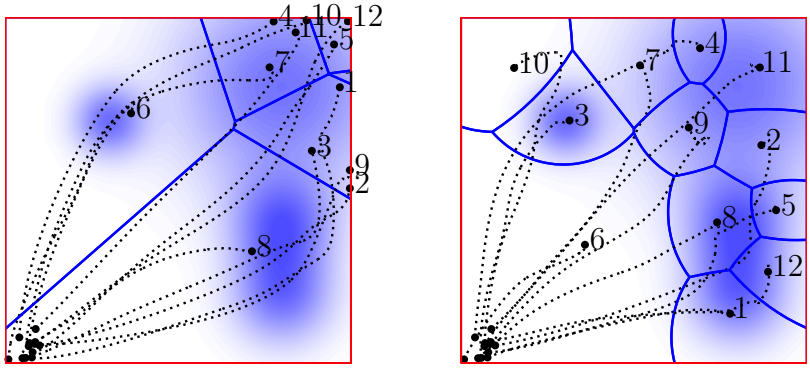


Figure 3.7: Energy-aware coverage simulation results. Shaded regions indicate a high value of ϕ. The path lines and final configurations are shown for energy-aware PWVD, left, and energy-aware MWVD coverage, right.

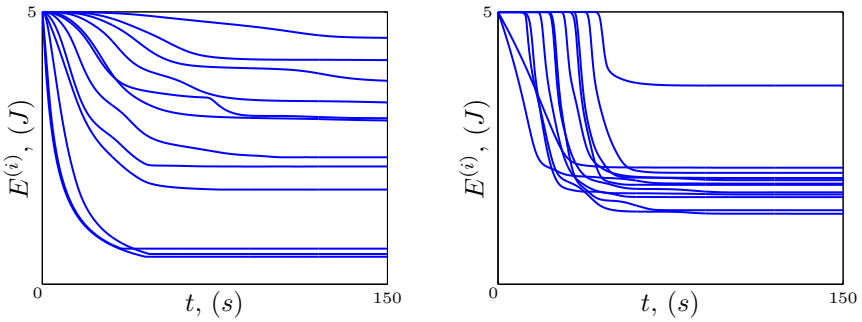


Figure 3.8: Energy-aware coverage energy consumption results. Energy profiles for the energy-aware PWVD simulation are at left, and MWVD coverage at right.

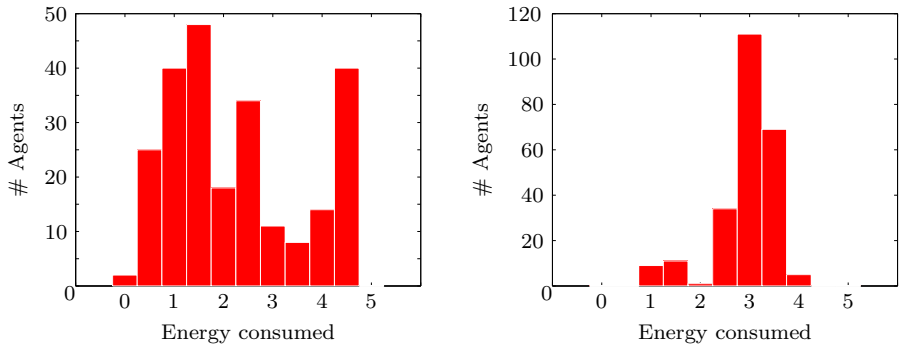


Figure 3.9: Energy consumption histograms for energy-aware coverage using the PWVD (left) and the MWVD (right).

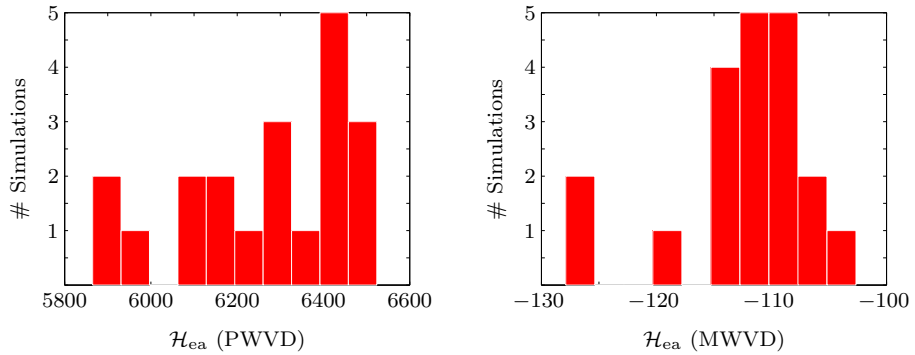


Figure 3.10: Objective function values at the end of the simulations for the energy-aware coverage using the PWVD (left) and the MWVD (right). Note that the PWVD objective function is from (3.5) using the metric (3.3) while the MWVD uses the metric (3.4).

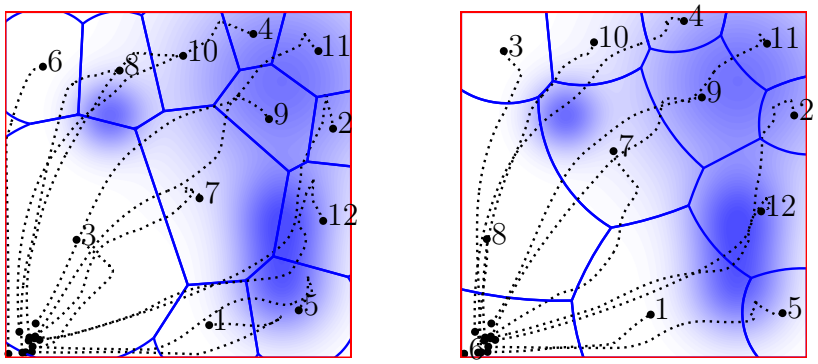


Figure 3.11: Area coverage simulation results. The path lines and final configurations are shown for limited-range PWVD area coverage, left, and limited-range MWVD area coverage, right.

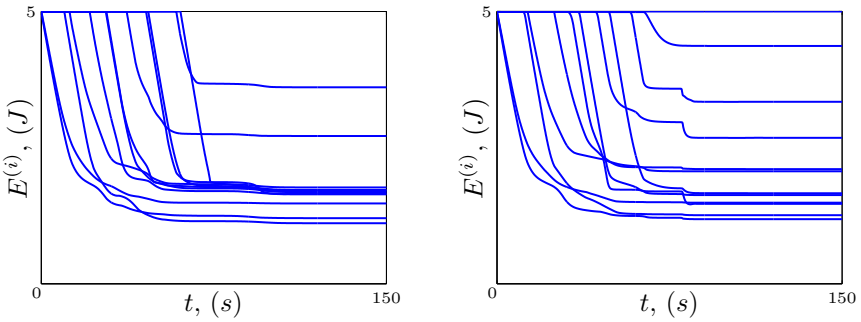


Figure 3.12: Area coverage simulation energy consumption results. Energy profiles for the area PWVD simulation are at left, and area MWVD coverage at right.

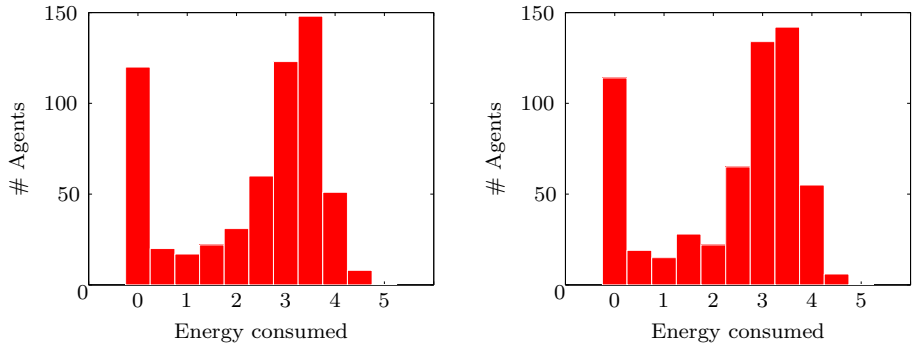


Figure 3.13: Energy consumption histograms for area coverage using the PWVD (left) and the MWVD (right).

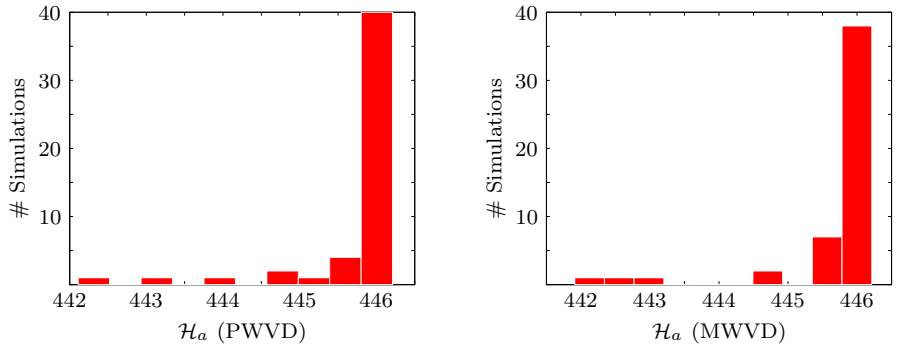


Figure 3.14: Objective function values at the end of the simulations for area coverage using the PWVD (left) and the MWVD (right).

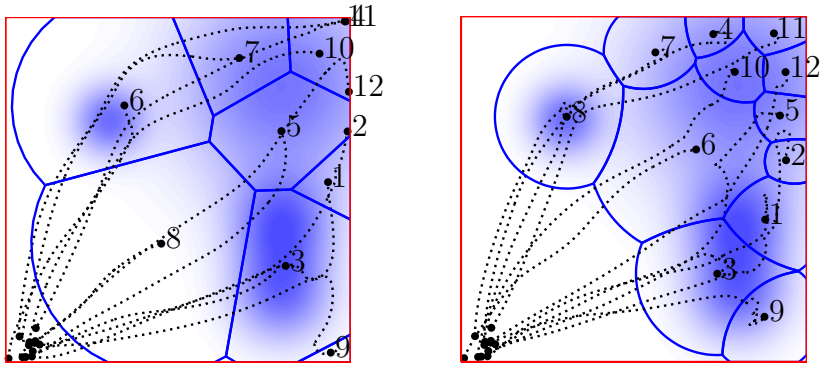


Figure 3.15: Mixed coverage simulation results. The path lines and final configurations are shown for limited-range PWVD mixed coverage, left, and limited-range MWVD mixed coverage, right.

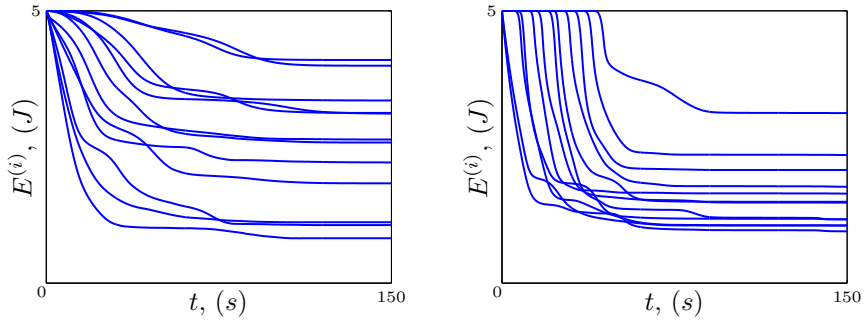


Figure 3.16: Mixed coverage simulation energy consumption results. Energy profiles for the mixed PWVD simulation are at left, and the mixed MWVD coverage at right.

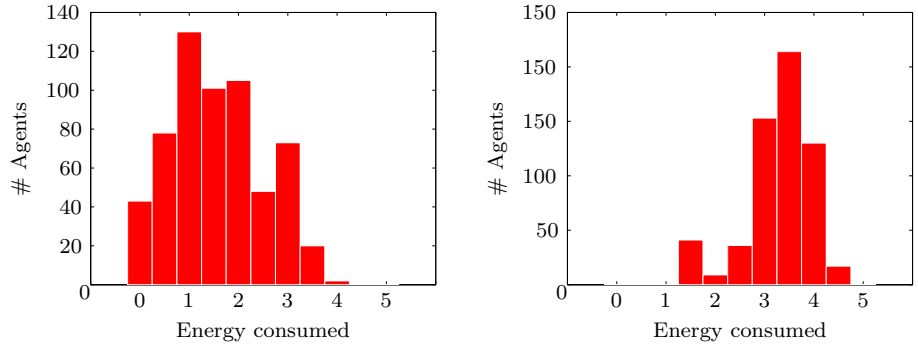


Figure 3.17: Energy consumption histograms for mixed coverage using the PWVD (left) and the MWVD (right).

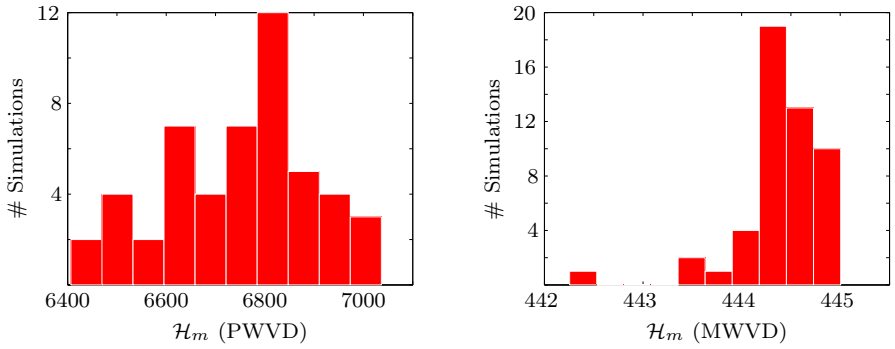


Figure 3.18: Objective function values at the end of the simulations for the mixed coverage using the PWVD (left) and the MWVD (right). Note that the PWVD objective function uses the metric (3.3) while the MWVD objective function uses the metric (3.4).

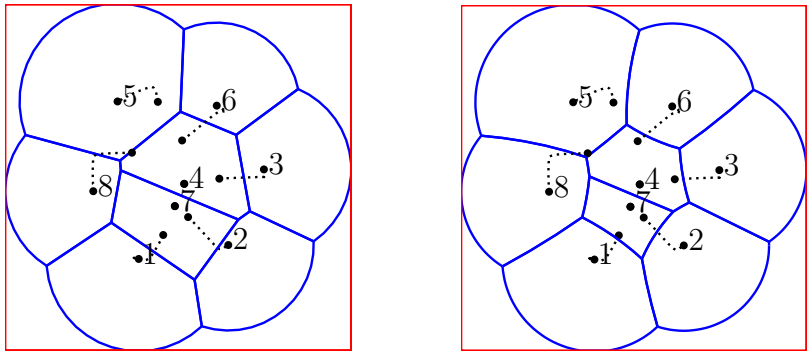


Figure 3.19: Base-return coverage simulation results. The path lines and final configurations are shown for limited-range PWVD case, left, and limited-range MWVD case, right.

Chapter 4

Unicycle Coverage via Hybrid Systems Analysis

Although each agent in a network may be controllable and the interaction among them can even be fixed, the consideration of nontrivial vehicle dynamics needs special treatment to avoid destabilizing effects. This has motivated a large number of papers on coordination algorithms for multi-agent systems with fixed interaction topologies; see e.g., [17, 90, 40, 21] on formation stabilization and synchronization. In particular, the stability analysis of these algorithms can be approached via Lyapunov methods and the classical LaSalle invariance principle as from [34]. However, when the inter-vehicle interaction topology is not fixed, even the consideration of first-order integrator dynamics may require hybrid-systems or switched-systems techniques.

Using multiple Lyapunov functions has been a predominant method for proving stability of a hybrid system, see [6, 42] and references therein. For multi-agent systems, previous work ([16, 15]) relied on LaSalle's invariance principle instead. The work [44] provides an extension of LaSalle's invariance principle to hybrid systems, while [54] makes use of set-valued Lyapunov theory to analyze the stability of multi-agent systems. More recently, [72, 25] revisits the notion of hybrid (time) trajectories and develops an invariance principle based on graphical convergence of set-valued maps. We choose the latter framework to analyze our system.

This chapter contributes to previous work on nonholonomic vehicle networks by applying Locational Optimization techniques from e.g. [16, 63] while assuming nonholonomic vehicle dynamics. References involving such dynamics include collision avoidance [61], cyclic pursuit [47, 12], and path-planning for Dubins vehicles [73]. In [16], convergence of a coverage control problem using unicycle-type dynamics was analyzed so that vehicles converged to a fixed target point as in [1], which was updated at discrete-time intervals. We lift this simplification, allowing target points (Voronoi centroids) to vary continuously with time.

The chapter is organized as follows. In Section 4.1, we introduce the two nonholonomic vehicle dynamics that we consider. We then apply the hybrid systems results found in Chapter 2 to each system of wheeled vehicles in Sections 4.2 and 4.3. Within those sections, we verify that the network of vehicles falls within the framework described in Chapter 2. We then prove convergence to centroidal configurations for each case of vehicle dynamics. We also provide simulations of this hybrid system and show that the algorithms perform as intended.

4.1 Problem setup and notation

We again perform coverage in the closed and convex region Q as from before, however, we additionally require that the boundary ∂Q be smooth such that the *inwards* pointing normal vector $\hat{\mathbf{n}}_{\text{in}}$ is well defined. We wish to perform the same minimization of the function (4.11) while incorporating various types of nonholonomic dynamics. That is, given a set of N agents with nonholonomic dynamics, deploy throughout Q and arrive at centroidal Voronoi configurations.

4.1.1 Nonholonomic vehicle dynamics

The use of omni-directional vehicles in [16] allows the minimization of (4.11) via a Lloyd-like gradient descent control law. This control law forces individual agents to move directly towards the centroid of their Voronoi regions and is distributed in the sense of the Delaunay graph. We wish to develop control algorithms that propel nonholonomic vehicles to these centroidal Voronoi configurations. We

utilize a hybrid systems approach and we refer the reader to the hybrid systems framework developed in [72, 25] and presented in Chapter 2 for the analysis tools employed in this work.

Referencing Figure 4.1, each vehicle has configuration variables $(p^{(i)}, \theta^{(i)}) \in \mathrm{SE}_Q(2)$, and a body coordinate frame with basis $e_1^{(i)} = (\cos\theta^{(i)}, \sin\theta^{(i)})$ and $e_2^{(i)} = (-\sin\theta^{(i)}, \cos\theta^{(i)})$. We denote $d^{(i)} = C^{(i)} - p^{(i)}$ as in Figure 4.1 and define the angle $\Omega^{(i)} \in [-\pi, \pi]$ to be the angle between $e_1^{(i)}$ and $d^{(i)}$. We assume the vehicles have bounded velocity and turning rate, $|v^{(i)}| \leq v_{\max}$ and $|\omega^{(i)}| \leq \omega_{\max}$ respectively. Next, we introduce two types of nonholonomic vehicles that we shall consider.

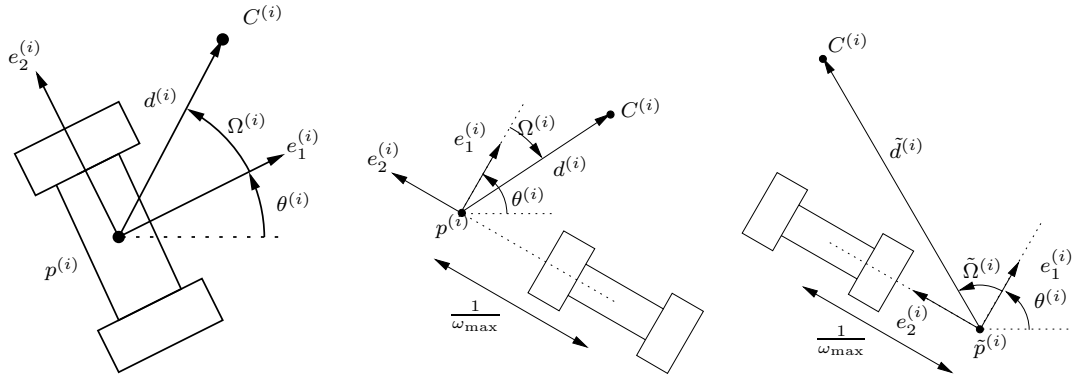


Figure 4.1: Vehicle with wheeled mobile dynamics (left). The variables are redefined for a vehicle with fixed forward velocity currently and a left virtual center (middle). The non-active virtual center quantities are shown with a tilde, ˜ (right).

Variable forward velocity

The first vehicle we consider has control over both forward speed and turning rate. Such a kinematic model is common for many mobile robot applications. It can perform one of three maneuvers: (i) move forward and steer, (ii) rotate in place, or (iii) be at a full stop. An additional discrete variable, $l \in \{1, 2, 3\}$, will be used to describe which of the three modes (forward, rotation, and rest) a vehicle is in. Each agent can then be described by a state variable, $x^{(i)} \in \mathrm{SE}_Q(2) \times \{1, 2, 3\}$. The multi-agent system state is denoted by $x = (x^{(1)}, \ldots, x^{(N)}) \in \mathbb{R}^{4N}$. In forward

motion, each vehicle flows according to

$$\begin{aligned} \dot{p}_1^{(i)} &= v^{(i)} \cos\theta^{(i)} \,, \\ \dot{p}_2^{(i)} &= v^{(i)} \sin\theta^{(i)} \,, \\ \dot{\theta}^{(i)} &= \omega^{(i)} \,, \end{aligned} \tag{4.1}$$

where $(\omega^{(i)}, v^{(i)})$ are the control inputs. Note that the definition of $(\theta^{(i)}, v^{(i)})$ is unique up to the discrete action $(\theta^{(i)}, v^{(i)}) \mapsto (\theta^{(i)} + \pi, -v^{(i)})$. A possibility is to use this symmetry to require $e_1^{(i)} \cdot d^{(i)} \geq 0$ for all $i \in \{1, \ldots, N\}$. Should the equality be violated at some time $t = t_0$, i.e., the centroid is behind the vehicle, then we could redefine $\theta^{(i)}(t^+) = \theta^{(i)}(t_0) + \pi$. The vehicle would instantaneously reverse directions leading to a fast adjustment. However, to provide a technical proof of correctness, we modify these proposed dynamics in many cases by a rotation in place.

A rotation in place introduces the new set of dynamics:

$$\begin{aligned} \dot{p}^{(i)} &= 0 \,, \\ \dot{\theta}^{(i)} &= \omega^{(i)} \,, \end{aligned} \tag{4.2}$$

where $\omega^{(i)}$ is the only control input.

In order to stabilize the vehicle to the target $C^{(i)}$, we will employ a discontinuous stabilizing law similar to that of [1]. This law relies on the angle $\Omega^{(i)}$ to both stabilize the position and orientation of the unicycle; see Figure 4.1. However, as the vehicle approaches $C^{(i)}$, the angle $\Omega^{(i)}$ will become ill-defined. To avoid this problem, we will make vehicles switch their dynamics to rest when they are within an ϵ-neighborhood of their targets. That is, the dynamics will be:

$$\begin{aligned} \dot{p}^{(i)} &= 0 \,, \\ \dot{\theta}^{(i)} &= 0 \,. \end{aligned} \tag{4.3}$$

Vehicles with fixed forward velocity

The second vehicle we consider has a fixed forward velocity, $v^{(i)} = 1$ for all $i \in \{1, \ldots, N\}$. Path planning for UAVs, underwater gliders, and other vehicles that must maintain a minimum forward velocity motivate this model. For path

planning purposes, a simplified UAV model as in [20] assumes constant velocity, constant altitude, and a minimum turning radius like the model we present here. We also define the vehicle *virtual center* as its center of rotation when the turning input is $\pm\omega_{\max}$. These centers can be on either side of the vehicle, and we will introduce a strategy to switch virtual center locations, if desired.

The objective is to steer the virtual center of each vehicle to a desired centroid target. This target will be the centroid of the vehicle's Voronoi region calculated using the virtual centers of all vehicles in the network. Once the virtual center has arrived at the centroid, the vehicle will *hover* about the centroid by maintaining the maximum steering input $\pm\omega_{\max}$. We construct the dynamics of the virtual center by first assuming dynamics of the form

$$\dot{p}^{(i)} = \left(\cos\theta^{(i)}, \quad \sin\theta^{(i)}\right)^{\mathsf{T}}, \dot{\theta}^{(i)} = \omega_i \,,$$

where ω_i is the only input. Then the virtual center is located in the local frame at $\left(0\,, \quad \pm\frac{1}{\omega_{\max}}\right)^{\mathsf{T}}$. We then transform this into the global frame and obtain

$$p^{(i)\prime} = p^{(i)} \pm \frac{1}{\omega_{\max}} \left(-\sin\theta^{(i)}, \quad \cos\theta^{(i)}\right)^{\mathsf{T}} . \tag{4.4}$$

The time derivative is

$$\begin{aligned} \dot{p}^{(i)\prime} &= \dot{p}^{(i)} \pm \frac{1}{\omega_{\max}} \left(-(\cos\theta^{(i)})\dot{\theta}^{(i)}, \quad -(\sin\theta^{(i)})\dot{\theta}^{(i)}\right)^{\mathsf{T}} \\ &= \left(1 \mp \frac{\omega^{(i)}}{\omega_{\max}}\right) \left(\cos\theta^{(i)}, \quad \sin\theta^{(i)}\right)^{\mathsf{T}} . \end{aligned} \tag{4.5}$$

Indeed, with $\omega^{(i)} = \pm\omega_{\max}$, the vehicle is hovering since the virtual center remains fixed, $\dot{p}^{(i)\prime} = 0$.

The virtual center may be located on either side of a vehicle's direction of travel. Also, each vehicle can either be in forward motion or hovering motion. This results in four possible modes of operation for each vehicle: forward-left, hover-left, forward-right, and hover-right. We enumerate each mode with the state $l^{(i)} \in \{1, 2, 3, 4\}$.

4.2 Vehicles with variable forward velocity

We now address the case of a vehicle that can actuate its forward speed as well as turning rate.

4.2.1 Hybrid modeling

Here we formally define the first hybrid system sketched in Section 4.1.1, so that it satisfies the Basic Conditions in [72] in order to apply the invariance principle found therein. We will take the state-space of the entire system to be $O = \mathbb{R}^{4N}$, so that $x \in (\mathrm{SE}_Q(2) \times \{1,2,3\})^N \subset O$ describes the position and orientation of *all* agents. We now define the hybrid system that models the nonholonomic vehicles, $\mathcal{S} = (F, G, A, B)$. In Section 4.1.1, we described three different types of dynamics. Here we specify the relatively-closed set $A \subseteq O$, where continuous-time evolution occurs.

To begin, we examine the configurations when a particular agent can flow, $A^{(i)} = A_1^{(i)} \cup A_2^{(i)} \cup A_3^{(i)}$:

1. an individual agent can be in $A_1^{(i)}$ if the centroid is in front of the agent, and if the centroid is too far from the agent,

2. an agent can be in $A_2^{(i)}$ if the centroid is too far from the agent, and either (i) the centroid is behind the agent or (ii) the agent is on the boundary ∂Q and pointing outwards, or

3. an agent can be in $A_3^{(i)}$ if the centroid is sufficiently close to the agent.

We present the precise set definitions with $0 < \underline{\epsilon} < \epsilon < \overline{\epsilon}$ and $\overline{\epsilon}$ is arbitrarily small, respectively:

$$
\begin{aligned}
A_1^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{1\},\, e_1^{(i)} \cdot d^{(i)} \geq \underline{\epsilon},\, \|d^{(i)}\| \geq \epsilon\}\,, \\
A_2^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{2\},\, \|d^{(i)}\| \geq \epsilon,\, e_1^{(i)} \cdot d^{(i)} \leq \epsilon\} \\
&\qquad \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_{\partial Q}(2) \times \{2\},\, \|d^{(i)}\| \geq \epsilon,\, e_1^{(i)} \cdot \hat{\mathbf{n}}_{\mathrm{in}} \leq 0\}\,, \\
A_3^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{3\}, \|d^{(i)}\| \leq \overline{\epsilon}\}\,.
\end{aligned}
$$

Extending this to apply for all agents, we have

$$A = \bigcap_{i=1}^{N} A^{(i)} \,. \tag{4.6}$$

Since each $A_k^{(i)} \subset A^{(i)}$ is relatively closed in O, A is also relatively closed in O, satisfying one of the Basic Conditions 1.

In forward motion, we propose a turning control gain $k_{\theta^{(i)}} < \infty$ proportional to the angular separation between the orientation of the vehicle and the target, $\Omega^{(i)}$. Additionally, we will have a control gain $k_{p^{(i)}} < \infty$ that is proportional to the distance to the target. In rotation, we consider a constant turning rate of $\pm\omega_{\max}$. We propose the following for each $i \in \{1, \ldots, N\}$:

$$\begin{aligned}
F_1^{(i)}(x) &= \left(\dot{p}_1^{(i)},\, \dot{p}_2^{(i)},\, \dot{\theta}^{(i)},\, \dot{l}^{(i)}\right)^{\mathsf{T}} = \left(k_{p^{(i)}} \cos\theta^{(i)},\, k_{p^{(i)}} \sin\theta^{(i)},\, k_{\theta^{(i)}}\Omega^{(i)},\, 0\right)^{\mathsf{T}} , \\
F_2^{(i)}(x) &= \left(0,\, 0,\, \omega_{\max}\operatorname{sgn}(\Omega^{(i)}),\, 0\right)^{\mathsf{T}} , \\
F_3^{(i)}(x) &= \mathbf{0} \,.
\end{aligned}$$

The control gains $k_{p^{(i)}}$ and $k_{\theta^{(i)}}$ will be chosen such that $0 \leq k_{p^{(i)}} \leq v_m$ and $0 \leq k_{\theta^{(i)}} \leq \frac{2\omega_{\max}}{\pi}$. From here, we can define the flow map $F : O \rightrightarrows O$. When $x \notin A$, $F(x) = \emptyset$, and when $x \in A$,

$$\begin{aligned}
&F(x) = (F_1(x), \ldots, F_N(x))^{\mathsf{T}}, \\
&F_i(x) = F_i^k(x) \iff l^{(i)} = k \in \{1,2,3\} \,. F(x) = \begin{pmatrix} F_1(x) \\ \vdots \\ F_n(x) \end{pmatrix} , \\
&F_i(x) = \begin{cases} F_1^{(i)}(x) & \text{if } l^{(i)} = 1 \,, \\ F_2^{(i)}(x) & \text{if } l^{(i)} = 2 \,, \\ F_3^{(i)}(x) & \text{if } l^{(i)} = 3 \,. \end{cases}
\end{aligned} \tag{4.7}$$

We now define the sets of configurations, $B^{(i)}$, $i \in \{1, \ldots, N\}$, where transitions from flowing to jumping may occur for an agent. An agent can jump if any of the following occur:

1. the Voronoi centroid is behind an agent,

2. the centroid is almost perpendicular to the direction of travel or when the agent is on the boundary,

3. the centroid transitions to be in front of an agent while the agent is in the rotation mode,

4. the centroid passes sufficiently close to an agent while it is performing either forward motion or rotation in place,

5. the centroid moves too far away from an agent while it is resting.

The precise set definitions for the above scenarios are, respectively:

$$\begin{aligned}
B_1^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{1\}, e_1^{(i)} \cdot d^{(i)} \leq -\epsilon\}, \\
B_2^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_{\partial Q}(2) \times \{1\}, e_1^{(i)} \cdot \hat{\mathbf{n}}_{\text{in}} \leq -\epsilon\} \\
&\qquad \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{1\}, -\epsilon \leq e_1^{(i)} \cdot d^{(i)} \leq \underline{\epsilon}\}, \\
B_3^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{2\}, e_1^{(i)} \cdot d^{(i)} \geq \epsilon, e_1^{(i)} \cdot \hat{\mathbf{n}}_{\text{in}} \geq 0\}, \\
B_4^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{1,2\}, \|d^{(i)}\| \leq \epsilon\}, \\
B_5^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{3\}, \|d^{(i)}\| \geq \overline{\epsilon}\}.
\end{aligned}$$

Then $B^{(i)} = \bigcup_{k=1}^{5} B_k^{(i)}$ and

$$B = \bigcup_{i=1}^{N} B^{(i)}. \tag{4.8}$$

It is not difficult to see that each $B_k^{(i)}$ is relatively closed in O, and so B is also relatively closed in O, satisfying another Basic Condition 1. The functions describing where x might jump to are, respectively:

$$\begin{aligned}
g_1^{(i)}(x) &= (p^{(i)}, \theta^{(i)} + \pi, 1), \\
g_2^{(i)}(x) &= (p^{(i)}, \theta^{(i)}, 2), \\
g_3^{(i)}(x) &= (p^{(i)}, \theta^{(i)}, 1), \\
g_4^{(i)}(x) &= (p^{(i)}, \theta^{(i)}, 3), \\
g_5^{(i)}(x) &= (p^{(i)}, \theta^{(i)}, 1).
\end{aligned}$$

For each vehicle, we define $G^{(i)}(x) = \{(x^{(1)}, \ldots, g_k^{(i)}(x), \ldots, x^{(N)}) \mid x \in B_k^{(i)}, \forall\, k \in \{1, \ldots, 5\}\}$. The overall jump map $G : O \rightrightarrows O$ is

$$G(x) = \begin{cases} \emptyset, & x \notin B\,, \\ \bigcup_{i=1}^{N} G^{(i)}(x), & \text{otherwise}\,. \end{cases} \tag{4.9}$$

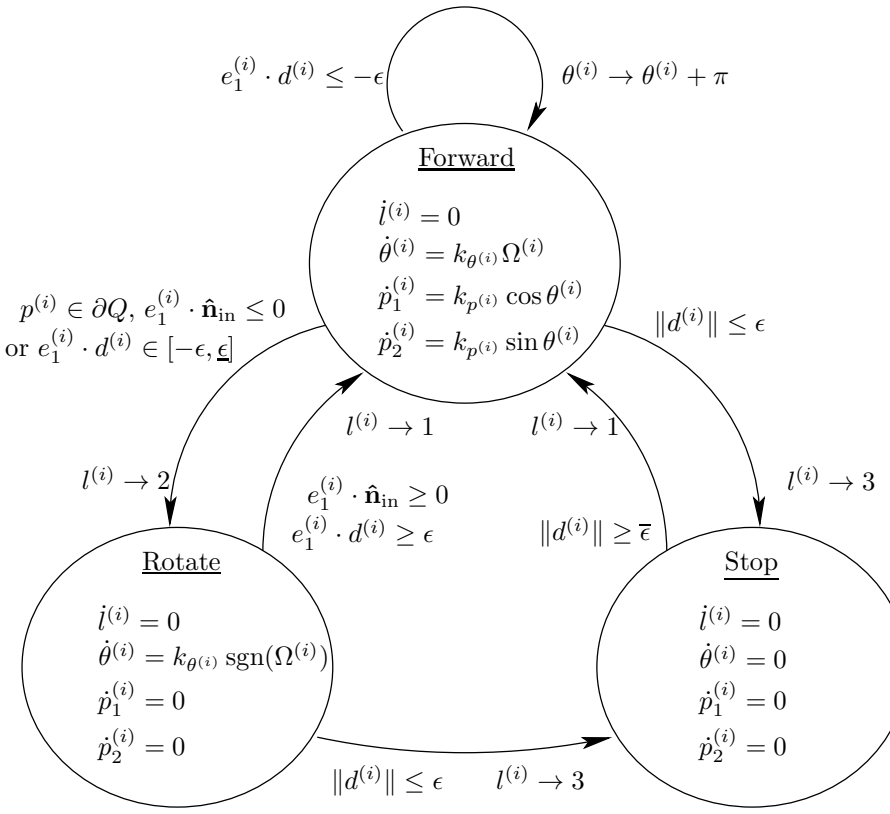


Figure 4.2: State transition diagram for each vehicle in the network.

Remark 21. *The jump map G takes the state $x(t, j) \in B_k^{(i)}$ to another set, $x(t, j+1) \in A \cup B$. The following are all the possibilities:*

1. *if $k = 1$ then $G(x) \in A_1^{(i)} \cup B_2^{(i)} \cup B_4^{(i)}$,*
2. *if $k = 2$ then $G(x) \in A_2^{(i)} \cup B_4^{(i)}$,*
3. *if $k = 3$ then $G(x) \in A_1^{(i)} \cup B_4^{(i)}$,*
4. *if $k = 4$ then $G(x) \in A_3^{(i)}$, and*

5. *if* $k = 5$ *then* $G(x) \in A_1^{(i)} \cup \ B_1^{(i)} \cup B_2^{(i)}$.

The state may also be in more than one jump set, such as $x \in B_2^{(i)} \cup B_4^{(i)}$. *Then, a jump can occur using* $g_2^{(i)}(x)$ *or* $g_4^{(i)}(x)$, *making this process non-deterministic.* •

Remark 22. *If we only implemented direction flipping, there exists a trajectory such that when* $e_1^{(i)} \cdot d^{(i)} = 0$, *the hybrid time domain* (t, j) *grows unbounded in* j *for fixed* t. *This occurs since after flipping directions, a vehicle is still in the switching set defined when* $e_1^{(i)} \cdot d^{(i)} = 0$. *Thus, it is possible for that vehicle to switch directions infinitely often in finite time. We include* $\underline{\epsilon}, \epsilon, \overline{\epsilon}$, *and the careful definition of* A *and* B *to prevent this and other similar situations. Other choices of the* A, B *sets are possible.* •

Proposition 23. *The hybrid system for variable forward velocity vehicles defined above in* (4.6)–(4.9) *satisfies the Basic Conditions of [25], Section-VI.*

Proof. By construction, O is an open set, so Basic Condition (i) is true. In addition, each $A_k^{(i)}$, $k = \{1, 2, 3\}$, is closed since Q is a closed set, and the inequalities are continuous and closed. A is then relatively closed in O since $A^{(i)}$ is the union of three closed sets, and A is the intersection of all $A^{(i)}$ for $i \in \{1, \ldots, N\}$. For similar reasons, the set $B^{(i)}$ for each $i \in \{1, \ldots, N\}$ is closed and this implies that B is relatively closed since it is a finite union of $B^{(i)}$, for $i \in \{1, \ldots, N\}$. Thus, Basic Condition (ii) is true.

We now check Basic Condition (iii). The flow map F can map to a single point or to the empty set, both of which are convex. In addition, F is locally bounded because $F_1^{(i)}, F_2^{(i)}, F_3^{(i)}$, $i \in \{1, \ldots, N\}$, are bounded over $\mathbb{R}^{4N}$ given that $k_{\theta^{(i)}}$ and $k_{p^{(i)}}$ are bounded.

Outer semicontinuity of F part 1, $x \in A$

It is important to note that each $A^{(i)}$, $i \in \{1, \ldots, N\}$ is the disjoint union, $A_1^{(i)} \cup A_2^{(i)} \cup A_3^{(i)}$. Suppose $x \in A_1^{(i)}$ for all $i \in \{1, \ldots, N\}$, the other cases where x is in one of $A_2^{(i)}, A_3^{(i)}$ are analogous. Consider now the convergent sequences $x_m \to x$ and $y_{i,m} \to y_i$ with $y_{i,m} = F_1^{(i)}(x_m)$ for all i.

(1) Suppose there exists an M such that $x_m \in A_1^{(i)}$ for all $m \geq M$. By continuity of $F_1^{(i)}$, $F_1^{(i)}(x_m)$ converges to $F_1^{(i)}(x)$. By unicity of limits, we have that $y_i = F_1^{(i)}(x)$ for all i.

(2) Suppose that for all $M \geq 0$ there exists $m_k \geq M$ such that $x_{m_k} \notin A_1^{(i)}$. Note that $\{x_{m_k}\} \subseteq \{x_m\}$. This implies that $x_{m_k} \in A_2^{(i)} \cup A_3^{(i)} \cup (O \setminus A^{(i)})$. We can assume that x_{m_k} are all in one of these sets without loss of generality.

   (a) Let $x_{m_k} \in A_2^{(i)}$ for all k. Since $A_2^{(i)}$ is closed, the limit of x_{m_k} is in $A_2^{(i)}$. This implies $x \in A_2^{(i)}$, a contradiction.

   (b) Let $x_{m_k} \in A_3^{(i)}$ for all k. We will reach a similar contradiction.

   (c) Let $x_{m_k} \in (O \setminus A)$ for all k. Then $y_{m_k} = F(x_{m_k}) = \emptyset$ for all k. Since the empty set is closed, $y_{m_k} \to y \in \emptyset$. Note that $\emptyset \subset F(x^{(i)})$ and the result follows.

Outer semicontinuity of F part 2, $x \notin A$

If $x \notin A$, then $F(x) = \emptyset$. Suppose also that there exists convergent sequences $x_m \to x$ and $y_m \to y$ such that $y_m = F(x_m)$ for all m.

1. Assume that $x_m \notin A$ for all $m \geq M$. Then $F(x_m) = \emptyset$ and $y_m = \emptyset$ for all $m \geq M$. Since $\emptyset$ is closed, $y_m \to y \in \emptyset$.

2. Suppose there exists an infinite subsequence $\{x_{m_k}\} \subseteq \{x_m\}$ with $x_{m_k} \in A$ for all k. Since A is closed, $x_{m_k} \to x \in A$, a contradiction.

Finally, we prove Basic Condition (iv). The map G is strictly set-valued since a particular $x^{(i)}$ can jump to multiple configurations, see (4.9). To prove local boundedness, consider an $x \in B$. We have to find a neighborhood $\mathcal{N} \subset O$ of x such that $\bigcup_{\bar{x} \in \mathcal{N}} G(\bar{x})$ is bounded. Observe that,

$$\begin{aligned} \bigcup_{\bar{x} \in \mathcal{N}} G(\bar{x}) &= \bigcup_{i, \bar{x} \in \mathcal{N}} G^{(i)}(\bar{x}) \\ &= \bigcup_{i=1}^{N} \{(x^{(1)}, \ldots, g_k^{(i)}(\bar{x}), \ldots, x_n) \mid \bar{x} \in \mathcal{N} \cap B_k^{(i)}\}. \end{aligned}$$

Each $g_k^{(i)}(x)$ is clearly locally bounded, so then we can find a $\mathcal{N}_0 \subset O$ of x such that $\bigcup_{\bar{x}\in\mathcal{N}_0} G(\bar{x})$ is a bounded set. Therefore, G is locally bounded. We now prove outer semicontinuity of G. Suppose there exist two convergent sequences $x_m \to x$ and $y_m \to y$ such that $y_m \in G(x_m)$. We must prove that $y \in G(x)$.

Outer semicontinuity of G part 1

Suppose that for all $K \geq 0$ there exists $m_k \geq K$ such that $x_{m_k} \notin B$. Then $G(x_{m_k}) = \emptyset$ and $y_{m_k} \in \emptyset$. Since the empty set is closed, $y_{m_k} \to y \in \emptyset$, and it is true that $\emptyset \subseteq G(x)$, for any $x \in O$.

Outer semicontinuity of G, part 2

Suppose that for all $K \geq 0$ there exists $m_k \geq K$ such that $x_{m_k} \in B$. Since B is closed, this implies $x_{m_k} \to x \in B$. If $x_{m_k} \in B$, this implies there exist fixed i_0 and k_0 such that $x_{m_k} \in B_{i_0}^{k_0}$ for an infinite number of m_k. Without loss of generality denote $\{x_{m_k}\}$ as $\{x_m\}$. Since each $B_k^{(i)}$ are closed, then $x_m \in B_{i_0}^{k_0}$ and $x_m \to x \in B_{i_0}^{k_0}$. Now, $y_m = (x^{(1)}, \ldots, g_{i_0}^{k_0}(x_m), \ldots, x_n) \in G(x_m)$. Since $g_{i_0}^{k_0}(x_m)$ is continuous, $g_{i_0}^{k_0}(x_m) \to g_{i_0}^{k_0}(x)$. By unicity of limits, $y_m \to y \in G(x)$. □

4.2.2 Asymptotic convergence

With Proposition 23, we can apply the hybrid LaSalle invariance principle in [72]:

Theorem 24. *Let $\mathcal{U} = O$. Given the hybrid system for vehicles with variable forward velocities above, any precompact trajectory $x(t, j)$, with* $\operatorname{rge} x \in \mathcal{U}$*, will approach the set of points*

$$\mathcal{M} = \left\{ x \in O \mid x \in \bigcap_{i=1}^{N} A_3^{(i)}, \right\} . \tag{4.10}$$

Proof. We will analyze the evolution of the cost (4.11). It can be shown that (4.11) is locally Lipschitz on O [16], for all x in A, $u_A(x) = \mathcal{L}_F\mathcal{H}$. We now compute the

derivative (see [16]).

$$\mathcal{L}_F\mathcal{H} = \sum_{i=1}^{N}\left[\frac{\partial\mathcal{H}}{\partial p^{(i)}}\dot{p}^{(i)} + \frac{\partial\mathcal{H}}{\partial\theta^{(i)}}\dot{\theta}^{(i)} + \frac{\partial\mathcal{H}}{\partial l^{(i)}}\dot{l}^{(i)}\right] = \sum_{i=1}^{N}\frac{\partial\mathcal{H}}{\partial p^{(i)}}\dot{p}^{(i)}$$
$$= \sum_{i=1}^{N} 2M^{(i)}(p^{(i)} - C^{(i)})^{\mathsf{T}}\dot{p}^{(i)}\,.$$

When an agent is in a rotating or rest mode, $x^{(i)} \in A_2^{(i)} \cup A_3^{(i)}$ and $\dot{p}^{(i)} = 0$. When $x^{(i)} \in A_1^{(i)}$, we have

$$\frac{\partial\mathcal{H}}{\partial p^{(i)}}\dot{p}^{(i)} = 2M^{(i)}(p^{(i)} - C^{(i)})^{\mathsf{T}}\begin{pmatrix} k_{p^{(i)}}\cos\theta^{(i)} \\ k_{p^{(i)}}\sin\theta^{(i)}\end{pmatrix} = 2k_{p^{(i)}}M^{(i)}(p^{(i)} - C^{(i)})\cdot e_1^{(i)}\,.$$

Recall from the $A_1^{(i)}$ definition that $e_1^{(i)}\cdot(C^{(i)} - p^{(i)}) \geq \underline{\epsilon}$, then $\frac{\partial\mathcal{H}}{\partial p^{(i)}}\dot{p}^{(i)} < 0$. Thus, $u_A(x) \leq 0$, $\forall\, x \in A$.

Since G is set-valued, $u_B(x) = \max_{x^+\in G(x)}\{\mathcal{H}(x^+) - \mathcal{H}(x)\}$. The cost function (4.11) does not have any dependence on $l^{(i)}$ or $\theta^{(i)}$. In addition, the jump map (4.9) does not create discontinuities in position. Thus, $\mathcal{H}$ does not change in value over jumps, and $u_B(x) = 0$, $\forall\, x \in B$.

The conditions of the hybrid LaSalle invariance principle are satisfied. Thus, the precompact trajectories x will approach the largest weakly invariant set in

$$L = \mathcal{H}^{-1}(r)\cap\mathcal{U}\cap\left(\overline{u_A^{-1}(0)}\cup u_B^{-1}(0)\right) = \mathcal{H}^{-1}(r)\cap\left(\overline{u_A^{-1}(0)}\cup B\right)\,,$$

for some $r \in \mathcal{H}(\mathcal{U})$. Note that $\mathcal{H}^{-1}(r)$ represents some level set of the cost function (4.11). Now we must identify the largest weakly invariant set, $\mathcal{M}$ in L. Since our system is autonomous, the largest weakly forward invariant set is also the largest weakly invariant set.

We now check for weakly invariant trajectories. We do this by assuming that one vehicle is in a switching state, and show that it must switch to a flowing state, and remain there for a non-zero amount of time. Then we show that the only flowing state which remains in a level set for all time is the stationary state, $x \in A_3^{(i)}$ for all $i \in \{1,\ldots,N\}$.

Suppose there exists a trajectory $\tilde{x}(t,j)$ with $\mathcal{H}(\tilde{x}) = r$ for all $(t,j) \in \mathbb{R}_{\geq 0}\times\mathbb{N}$ such that $\tilde{x}(t_0,j_0) \in B$. This implies that there exists i^* and k^* such

that $\tilde{x}(t_0, j_0) \in B_{k^*}^{(i^*)}$. Referencing Remark 21, all jumps eventually terminate with $\tilde{x}(t_0, j) \in A^{(i)} = A_1^{(i)} \cup A_2^{(i)} \cup A_3^{(i)}$ for some $j > j_0$. Furthermore, this configuration $\tilde{x}(t_0, j)$ remains in $A^{(i)}$ for a nontrivial amount of time. We have shown that all configurations $x \in B$ return to flowing states. Now we examine the case where $\tilde{x}(t, j) \in A$ to arrive at the final result.

Suppose there exists a trajectory $\tilde{x}(t, j)$ with $\mathcal{H}(\tilde{x}) = r$ for all $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$ and $\tilde{x}(t_0, j_0) \in A$ for some $t_0 + j_0 \geq 0$. Since $\frac{\partial \mathcal{H}}{\partial p^{(i)}} \dot{p}^{(i)} < 0$ for any $x \in A_1^{(i)}$, this implies that $\tilde{x}(t_0, j_0) \in A_2^{(i)} \cup A_3^{(i)}$ for all $i \in \{1, \ldots, N\}$. If this is true, then $\dot{p}^{(i)} = 0$ for all $i \in \{1, \ldots, N\}$. Suppose there exists an i^* such that $\tilde{x}(t_0, j_0) \in A_2^{(i^*)}$. Because $\dot{p}^{(i)} = 0$, $C^{(i)}$ is constant for all $i \in \{1, \ldots, N\}$, and under the flow $F_2^{(i)}$, $|\Omega^{(i^*)}|$ decreases. Then, for some t_1, such that $t_0 \leq t_1 < \infty$, $\tilde{x}(t_1, j_0) \in B_3^{(i^*)}$ where a jumped is forced so that $\tilde{x}(t_1, j_0 + 1) \in agentAi^*{}_1$. This implies that $u_A(\tilde{x}) < 0$, and the trajectory $\tilde{x}(t, j)$ leaves the level set $\mathcal{H}^{-1}(r)$.

Therefore, in order to remain in the level set $\mathcal{H}^{-1}(r)$, trajectories $x(t, j)$ must satisfy $x \in A_3^{(i)}$ for all $i \in \{1, \ldots, N\}$. This also satisfies $x \in \overline{u_A^{-1}(0)}$. □

4.3 Vehicles with fixed forward velocity

We now address the case of a vehicle that can only actuate its turning rate.

4.3.1 Virtual center switching

Taking the Locational Optimization cost function

$$\mathcal{H}(P, \mathcal{W}) = \sum_{i=1}^{N} \int_{W^{(i)}} \|q - p^{(i)}\|^2 \phi(q) dq \,, \tag{4.11}$$

where $P = (p^{(1)}, \ldots, p^{(N)})$ is the tuple of vehicle locations, and $\mathcal{W} = (W^{(1)}, \ldots, W^{(N)})$ is any partition of Q formed from P. Let $\mathcal{V}$ be the Voronoi partition formed from P. It has been shown that for any partition $\mathcal{W}$,

$$\mathcal{H}(C, \mathcal{W}) \leq \mathcal{H}(P, \mathcal{W}) \,, \tag{4.12}$$

$$\mathcal{H}(P, \mathcal{V}) \leq \mathcal{H}(P, \mathcal{W}) \,, \tag{4.13}$$

where $C = (C^{(1)}, \ldots, C^{(N)})$ is the tuple containing the centroids of each region $\mathcal{W}$, see [15].

Lemma 25. *Let $P = (p^{(1)}, \ldots, p^{(i)}, \ldots, p^{(N)})$, and let $\tilde{P} = (p^{(1)}, \ldots, \tilde{p}^{(i)}, \ldots, p^{(N)})$ where $\tilde{p}^{(i)}$ is closer to the centroid $C^{(i)}$, $\|\tilde{p}^{(i)} - C^{(i)}\| \leq \|p^{(i)} - C^{(i)}\|$. Let $\mathcal{V}(P)$ be the Voronoi partition of Q associated with P. Then,*

$$\mathcal{H}(\tilde{P}, \mathcal{V}(P)) \leq \mathcal{H}(P, \mathcal{V}(P))\,, \tag{4.14}$$

and

$$\Delta\mathcal{H} \triangleq \mathcal{H}(\tilde{P}, \mathcal{V}(P)) - \mathcal{H}(P, \mathcal{V}(P)) = M^{(i)}(\|\tilde{p}^{(i)} - C^{(i)}\|^2 - \|p^{(i)} - C^{(i)}\|^2) \leq 0\,. \tag{4.15}$$

Additionally, $\mathcal{H}(\tilde{P}, \mathcal{V}(\tilde{P})) - \mathcal{H}(P, \mathcal{V}(P)) \leq \Delta\mathcal{H}$.

Proof. At first, we have

$$\mathcal{H}(P, \mathcal{W}) = \sum_{i=1}^{N} \int_{W^{(i)}} \|q - p^{(i)}\|^2 \phi(q) dq\,.$$

Note that the above expression is the sum of the moment of inertias of each region about the positions $p^{(i)}$ of the vehicles. Recalling the Parallel Axis Theorem, the moment of inertia of an object about any axis of rotation parallel to an axis passing through the center of mass can be written as

$$I = I_{CM} + MR^2\,,$$

where I_{CM} is the moment of inertia about the axis through the center of mass, M is the mass of the object, and R is the perpendicular distance between the new axis and the axis trough the center of mass. Therefore, we can rewrite the cost function as

$$\mathcal{H}(P, \mathcal{W}) = \sum_{i=1}^{N} I^{(i)} = \sum_{i=1}^{N} I^{(i)}_{\mathrm{CM}} + M^{(i)} \|p^{(i)} - C^{(i)}\|^2\,.$$

and then,

$$\mathcal{H}(\tilde{P}, \mathcal{W}) = \mathcal{H}(P, \mathcal{W}) - M^{(i)}(\|p^{(i)} - C^{(i)}\|^2 - \|\tilde{p}^{(i)} - C^{(i)}\|^2)\,.$$

Replacing the arbitrary partition $\mathcal{W}$ with $\mathcal{V}(P)$ and noting that $\|\tilde{p}^{(i)} - C^{(i)}\| \leq \|p^{(i)} - C^{(i)}\|$ gives the result (4.15).

However, using the relations (4.12), (4.13), we can conclude that

$$\mathcal{H}(\tilde{P}, \mathcal{V}(\tilde{P})) \leq \mathcal{H}(\tilde{P}, \mathcal{V}(P)) \leq \mathcal{H}(P, \mathcal{V}(P))\,, \tag{4.16}$$

and so the difference $\mathcal{H}(\tilde{P}, \mathcal{V}(\tilde{P}) - \mathcal{H}(P, \mathcal{V}(P))$ is upper bounded by (4.15). □

We propose that each vehicle can switch the position of its virtual center only if the improvement given by Lemma 25 is better than some threshold β, which implies that the actual improvement in cost is $\mathcal{H}(P, \mathcal{V}(P)) - \mathcal{H}(\tilde{P}, \mathcal{V}(\tilde{P})) \geq \beta$.

4.3.2 Hybrid modeling

To simplify notation, let the current virtual center be $p^{(i)}$, and the opposite virtual center be $\tilde{p}^{(i)} = p^{(i)} \pm \frac{2}{\omega_{\max}} e_2^{(i)}$. Let $\tilde{d}_i = C^{(i)} - \tilde{p}^{(i)}$ and let $\tilde{\Omega}_i$ denote the angle between $e_1^{(i)}$ and $\tilde{d}_i$, see Figure 4.1.

As in the previous section, we define the state-space $O = \mathbb{R}^{4N}$. Let $A_1^{(i)}$, $\ldots$, $A_4^{(i)}$ be the set of points in O where a vehicle i can flow continuously in each of the four modes. We present a brief description of these sets followed by a precise set definition.

(1) An individual vehicle can be in $A_1^{(i)}$ (resp. $A_3^{(i)}$) if the centroid is in front of the left (resp. right) virtual center at $p^{(i)}$, and if $p^{(i)}$ is not sufficiently close to $C^{(i)}$. Additionally, the improvement from switching between forward-left to forward-right (resp. vice-versa) given by Lemma 25 must be better than β. However, if the opposite virtual center $\tilde{p}^{(i)}$ is not in Q, then the vehicle may maintain its current virtual center despite violating the improvement threshold β.

$$\begin{aligned}
A_1^{(i)} = \{&x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{1\},\, e_1^{(i)} \cdot d^{(i)} \geq \underline{\epsilon},\\
&M^{(i)}\|d^{(i)}\|^2 - M^{(i)}\|\tilde{d}_i\|^2 \leq \beta,\, \|d^{(i)}\| \geq \epsilon\}\\
\cup \{&x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{1\},\, e_1^{(i)} \cdot d^{(i)} \geq \underline{\epsilon},\, \tilde{p}^{(i)} \in \overline{Q^c},\, \|d^{(i)}\| \geq \epsilon\},\\
A_3^{(i)} = \{&x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{3\},\, e_1^{(i)} \cdot d^{(i)} \geq \epsilon,\\
&M^{(i)}\|d^{(i)}\|^2 - M^{(i)}\|\tilde{d}_i\|^2 \leq \beta,\, \|d^{(i)}\| \geq \epsilon\}\\
\cup \{&x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{3\},\, e_1^{(i)} \cdot d^{(i)} \geq \underline{\epsilon},\, \tilde{p}^{(i)} \in \overline{Q^c},\, \|d^{(i)}\| \geq \epsilon\},
\end{aligned}$$

(2) A vehicle can be in $A_2^{(i)}$ (resp. $A_4^{(i)}$) if $C^{(i)}$ is behind the left (resp. right) virtual center $p^{(i)}$, or if $p^{(i)}$ is on the boundary Q and heading outwards, or if $p^{(i)}$ is sufficiently close to $C^{(i)}$.

$$\begin{aligned}
A_2^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{2\},\, e_1^{(i)} \cdot d^{(i)} \leq \epsilon,\, \|d^{(i)}\| \geq \epsilon\} \\
&\quad \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_{\partial Q}(2) \times \{2\},\, e_1^{(i)} \cdot \hat{\mathbf{n}}_{\mathrm{in}} \leq 0\} \\
&\quad \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{2\},\, \|d^{(i)}\| \leq \overline{\epsilon}\}\,, \\
A_4^{(i)} &= \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{4\},\, e_1^{(i)} \cdot d^{(i)} \leq \epsilon,\, \|d^{(i)}\| \geq \epsilon\} \\
&\quad \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_{\partial Q}(2) \times \{4\},\, e_1^{(i)} \cdot \hat{\mathbf{n}}_{\mathrm{in}} \leq 0\} \\
&\quad \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{4\},\, \|d^{(i)}\| \leq \overline{\epsilon}\}\,.
\end{aligned}$$

The hysteresis variables $0 < \underline{\epsilon} < \epsilon < \overline{\epsilon}$ serve to insure that Zeno effects do not occur. Combining these sets together, the entire system can flow if $x \in A$ where

$$A = \bigcap_{i=1}^{N} \left(A_1^{(i)} \cup A_2^{(i)} \cup A_3^{(i)} \cup A_4^{(i)} \right)\,. \tag{4.17}$$

When the system configuration $x \in A$, the state evolves under the map F. We present flow maps for individual vehicles and then compose them to form F. Let $F^{(i)}(x) = (\dot{p}_1^{(i)},\, \dot{p}_2^{(i)},\, \dot{\theta}^{(i)},\, \dot{l}^{(i)})^{\mathsf{T}}$ with:

$$\begin{aligned}
F_1^{(i)}(x) &= (\cos\theta^{(i)},\, \sin\theta^{(i)},\, \frac{2\Omega^{(i)}\omega_{\max}}{\pi},\, 0)^{\mathsf{T}}\,, \\
F_2^{(i)}(x) &= (\cos\theta^{(i)},\, \sin\theta^{(i)},\, \omega_{\max},\, 0)^{\mathsf{T}}\,, \\
F_3^{(i)}(x) &= (\cos\theta^{(i)},\, \sin\theta^{(i)},\, \frac{2\Omega^{(i)}\omega_{\max}}{\pi},\, 0)^{\mathsf{T}}\,, \\
F_4^{(i)}(x) &= (\cos\theta^{(i)},\, \sin\theta^{(i)},\, -\omega_{\max},\, 0)^{\mathsf{T}}\,.
\end{aligned}$$

Then, for any $x \in A$,

$$\begin{aligned}
F(x) &= (F^{(1)}(x),\, \cdots,\, F^{(N)}(x))^{\mathsf{T}}\,, \\
F^{(i)}(x) &= F_k^{(i)}(x) \iff l^{(i)} = k \in \{1,2,3,4\}.
\end{aligned} \tag{4.18}$$

We now describe the subset of O where discrete jumps can occur. We will consider:

1. switching from forward-left to forward-right, $B_1^{(i)} = \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{1\},\, e_1^{(i)} \cdot d^{(i)} \geq \underline{\epsilon},\, M^{(i)}(\|d^{(i)}\|^2 - \|\tilde{d}_i\|^2) \geq \beta,\, \tilde{p}^{(i)} \in Q\}\,,$

2. switching from forward-right to forward-left, $B_2^{(i)} = \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{3\},\ e_1^{(i)} \cdot d^{(i)} \geq \underline{\epsilon},\ M^{(i)}(\|d^{(i)}\|^2 - \|\tilde{d}_i\|^2) \geq \beta,\ \tilde{p}^{(i)} \in Q\}$,

3. switching from forward-left to hover-left, $B_3^{(i)} = \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{1\},\ e_1^{(i)} \cdot d^{(i)} \leq \underline{\epsilon}\} \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_{\partial Q}(2) \times \{1\}, e_1^{(i)} \cdot \hat{\mathbf{n}}_{\text{in}} \leq -\epsilon\} \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{1\},\ \|d^{(i)}\| \leq \epsilon\}$,

4. switching from hover-left to forward-left, $B_4^{(i)} = \{x \in O \mid x^{(i)} \in SE_Q(2) \times \{2\},\ e_1^{(i)} \cdot d^{(i)} \geq \epsilon,\ e_1^{(i)} \cdot \hat{\mathbf{n}}_{\text{in}} \geq 0,\ \|d^{(i)}\| \geq \overline{\epsilon}\}$

5. switching from forward-right to hover-right, $B_5^{(i)} = \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{3\},\ e_1^{(i)} \cdot d^{(i)} \leq \underline{\epsilon}\} \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_{\partial Q}(2) \times \{3\}, e_1^{(i)} \cdot \hat{\mathbf{n}}_{\text{in}} \leq -\epsilon\} \cup \{x \in O \mid x^{(i)} \in \mathrm{SE}_Q(2) \times \{3\},\ \|d^{(i)}\| \leq \epsilon\}$, and

6. switching from hover-right to forward-right. $B_6^{(i)} = \{x \in O \mid x^{(i)} \in SE_Q(2) \times \{4\},\ e_1^{(i)} \cdot d^{(i)} \geq \epsilon,\ e_1^{(i)} \cdot \hat{\mathbf{n}}_{\text{in}} \geq 0,\ \|d^{(i)}\| \geq \overline{\epsilon}\}$

Similar to the previous section, we define

$$B = \bigcup_{i=1}^{N} \bigcup_{k=1}^{6} B_k^{(i)}. \tag{4.19}$$

With the switching domain defined, we present the jump map G. Let $g_1^{(i)}(x), \ldots, g_6^{(i)}(x)$ be the maps for an individual vehicle i. These maps are:

$$\begin{aligned}
g_1^{(i)}(x) &= (3, p^{(i)} - \frac{2}{\omega_{\max}} e_2^{(i)}, \theta^{(i)}), \\
g_2^{(i)}(x) &= (1, p^{(i)} + \frac{2}{\omega_{\max}} e_2^{(i)}, \theta^{(i)}), \\
g_3^{(i)}(x) &= (2, p^{(i)}, \theta^{(i)}), \\
g_4^{(i)}(x) &= (1, p^{(i)}, \theta^{(i)}), \\
g_5^{(i)}(x) &= (4, p^{(i)}, \theta^{(i)}), \\
g_6^{(i)}(x) &= (3, p^{(i)}, \theta^{(i)}).
\end{aligned}$$

We combine the above functions for each vehicle and obtain

$$G^{(i)}(x) = \left\{(x^{(1)}, \ldots, g_k^{(i)}(x), \ldots, x^{(N)}) \mid x \in \bigcup_{k=1}^{6} B_k^{(i)}\right\}.$$

The complete set-valued jump map is then

$$G(x) = \bigcup_{i=1}^{N} G^{(i)}(x)\,. \tag{4.20}$$

Proposition 26. *The hybrid system described in* (4.17)–(4.20) *satisfy the Basic Conditions of [25], Section-VI. The proof for this is similar to that of Proposition 23.* ■

4.3.3 Convergence analysis

We now apply the hybrid LaSalle principle.

Theorem 27. *Let $\mathcal{U} = O$. Given the hybrid system for fixed forward velocity vehicles defined above and with virtual center dynamics* (4.5)*, any precompact trajectory $x(t, j)$ with* $\operatorname{rge} x \in \mathcal{U}$*, will approach the set of points*

$$\mathcal{M} = \{x \in O \mid \|C^{(i)} - p^{(i)}\| \leq \overline{\epsilon},\, \forall\, i \in \{1, \ldots, N\}\}\,. \tag{4.21}$$

Proof. We choose to analyze the system using the cost function $\mathcal{H}(P, \mathcal{V})$ from (4.11) with P being the locations of the virtual centers. It can be shown that $\mathcal{H}$ is locally Lipschitz on O [16].

For all $x \in A$, $u_A = \mathcal{L}_F\mathcal{H}(x)$. It can be seen that

$$\mathcal{L}_F\mathcal{H}(x) = \sum_{i=1}^{N} 2M^{(i)}(p^{(i)} - C^{(i)})^{\mathsf{T}} \cdot \left(1 \mp \frac{\Omega^{(i)}}{\omega_{\max}}\right) \begin{pmatrix} \cos\theta^{(i)} \\ \sin\theta^{(i)} \end{pmatrix}\,.$$

When all agents are in a hovering mode, $A_2^{(i)}, A_4^{(i)}$, $\omega_i = \pm\omega_{\max}$ and the virtual center remains stationary, therefore $\dot{\mathcal{H}} = 0$. When all agents are in forward mode, we have

$$\dot{\mathcal{H}} = \sum_{i=1}^{N} 2M^{(i)} \left(1 \mp \frac{2\Omega^{(i)}}{\pi}\right) (p^{(i)} - C^{(i)})^{\mathsf{T}} \begin{pmatrix} \cos\theta^{(i)} \\ \sin\theta^{(i)} \end{pmatrix}\,.$$

Additionally, $(p^{(i)} - C^{(i)}) \cdot \left(\cos\theta^{(i)},\ \sin\theta^{(i)}\right) = -d^{(i)} \cdot e_1^{(i)} = -\cos\Omega^{(i)}$. Thus, $\dot{\mathcal{H}} = \sum_{i=1}^{N} -2M^{(i)} \left(1 \mp \frac{2\Omega^{(i)}}{\pi}\right) \cos\Omega^{(i)}$. A vehicle can only be in forward mode if $d^{(i)} \cdot e_1^{(i)} \geq \underline{\epsilon}$, or equivalently, if $\Omega^{(i)} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$ (see the definition of $A_1^{(i)}$ and $A_3^{(i)}$

in (4.17)). Therefore $\cos\Omega^{(i)} \in [0,1)$ and $\frac{2\Omega^{(i)}}{\pi} \in (-1,1)$ and $\dot{\mathcal{H}} \leq 0$ for all $x \in A$, and the inequality is strict if there is at least one vehicle in forward motion.

Since G is set-valued, $u_B(x) = \max_{x^+ \in G(x)}\{\mathcal{H}(x^+) - \mathcal{H}(x)\}$. The cost function (4.11), does not have any dependence on $l^{(i)}$ or $\theta^{(i)}$. Thus, $\mathcal{H}$ changes only if virtual center positions change. However, when an agent switches from $l^{(i)} = 1$ to $l^{(i)} = 3$ or vice-versa, Lemma 25 insures that the difference $\mathcal{H}(\tilde{P}, \tilde{\mathcal{V}}) - \mathcal{H}(P, \mathcal{V}) \leq -\beta$. Therefore, for all discrete jumps with $x \in B$, $u_B(x) = \max_{x^+ \in G(x)}\{\mathcal{H}(x^+) - \mathcal{H}(x)\} \leq 0$.

All conditions of the hybrid LaSalle invariance principle have been satisfied. The precompact trajectories x will approach the largest weakly invariant set in

$$\begin{aligned} L &= \mathcal{H}^{-1}(r) \cap \mathcal{U} \cap \left(\overline{u_A^{-1}(0)} \cup u_B^{-1}(0)\right) \\ &= \mathcal{H}^{-1}(r) \cap \left[\bigcap_{i=1}^{N}(A_2^{(i)} \cup A_4^{(i)})\right] \cup \left[\bigcup_{i=1}^{N}(B_3^{(i)} \cup \cdots \cup B_6^{(i)})\right]. \end{aligned}$$

for some $r \in \mathcal{H}(\mathcal{U})$. Note that $\mathcal{H}^{-1}(r)$ represents some level set of the cost function (4.11). We now check for weakly invariant trajectories that remain in L. We do this by assuming that one vehicle is in a switching state, and show that it must switch to a flowing state, and remain there for a non-zero amount of time. Then we show that the only flowing state which remains in a level set for all time is the hovering state with $\|p^{(i)} - C^{(i)}\| \leq \overline{\epsilon}$ for all $i \in \{1, \ldots, N\}$.

Suppose there exists a trajectory $\tilde{x}(t,j)$ with $\mathcal{H}(\tilde{x}) = r$ for all $(t,j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$ such that $\tilde{x}(t_0, j_0) \in B$. This implies that there exists i^* and k^* such that $\tilde{x}(t_0, j_0) \in B_{k^*}^{(i^*)}$. We confine our analysis to the cases where $\tilde{x}(t_0, j_0) \in (B_3^{(i)} \cup \cdots \cup B_6^{(i)})$. Let $\tilde{x}^+$ denote the state that $\tilde{x}(t_0, j_0)$ jumps to. The following transitions are possible: (1) $\tilde{x} \in B_3^{(i)} \mapsto \tilde{x}^+ \in A_2^{(i)}$, (2) $\tilde{x} \in B_4^{(i)} \mapsto \tilde{x}^+ \in A_1^{(i)} \cup B_1^{(i)}$, (3) $\tilde{x} \in B_5^{(i)} \mapsto \tilde{x}^+ \in A_4^{(i)}$, (4) $\tilde{x} \in B_6^{(i)} \mapsto \tilde{x}^+ \in A_3^{(i)} \cup B_2^{(i)}$.

We note that for the jumps that could result with $x^+ \in B_1^{(i)} \cup B_2^{(i)}$ (cases 2 and 4), the system must jump again, but this jump decreases the cost function. In other words, these jumps take the system outside of the set L. The remaining possibilities result with $x^+ \in A^{(i)}$. The only possible trajectories that remain in the set L are those that jump to flowing states, $x^+ \in A$. Specifically, $x^+ \in A_2^{(i)} \cup A_4^{(i)}$ for all $i \in \{1, \ldots, N\}$. Now we examine this case to arrive at the final result.

Suppose that there exists a trajectory $\tilde{x}(t,j)$ with $\mathcal{H}(\tilde{x}) = r$ for all $(t,j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$. such that $\tilde{x}(t_0,j_0) \in A_2^{(i)} \cup A_4^{(i)}$ for all $i \in \{1,\ldots,N\}$. An agent in $A_2^{(i)}$ (resp. $A_4^{(i)}$) can only jump to forward motion by being in $B_4^{(i)}$ (resp. $B_6^{(i)}$). Since all agents are rotating about their virtual centers, the locations of the centroids, $C^{(i)}$ for all $i \in \{1,\ldots,N\}$, remain fixed. This implies that $d^{(i)}$, $i \in \{1,\ldots,N\}$, also remain fixed. If there exists one agent i^* such that $\|d^{(i^*)}\| \geq \overline{\epsilon}$, then a jump eventually occurs since all vehicles rotate about their virtual centers with constant angular velocity. The system configuration will be $\tilde{x}(t_1,j_0) \in B_4^{(i^*)}$ (resp. $\tilde{x}(t_1,j_0) \in B_6^{(i^*)}$) for some $t_1 \geq t_0$. The resulting jump necessarily results in $\tilde{x}^+ \in A_1^{(i)} \cup A_3^{(i)}$, and the trajectory $\tilde{x}$ leaves the level set $\mathcal{H}(\tilde{x}) = r$.

Thus, the only weakly invariant set in L is exactly that of (4.21). □

4.4 Simulations

We simulate $N = 8$ unicycles in $Q \subset \mathbb{R}^2 = [0,10] \times [0,10]$. The density function, ϕ, is composed of 3 Gaussian distributions $\phi(q) = 0.05 + 3\Big[e^{-\frac{\|q-r_1\|^2}{2}} + e^{-\frac{\|q-r_2\|^2}{2}} + e^{-\|q-r_3\|^2}\Big]$, where $r_1 = (8,2)$, $r_2 = (8,4)$ and $r_3 = (3,7)$. The agent positions and orientations were randomly distributed in the bottom left corner, $l^{(i)} = 1$ for all $i \in \{1,\ldots,N\}$. We chose the control gains to be $k_{\theta^{(i)}} = 5$ and $k_{p^{(i)}} = \text{sat}\,\|C^{(i)} - p^{(i)}\|$. Note that any positive $k_{\theta^{(i)}}$ and $k_{p^{(i)}}$ will work. Figure 5.25 shows that the wheeled vehicles with variable forward velocity do in fact converge to near-centroidal configurations, and that the corresponding cost function $\mathcal{H}$ is nonincreasing.

We present the case where vehicles have a fixed forward velocity in Figure 4.4. All vehicles began with random positions and orientations in the lower left corner. All agents started with a left virtual center, but agent 5 switches to a right virtual center early in the simulation. Similar to the case of variable forward velocity vehicles, the fixed forward velocity vehicles navigate their virtual centers to the centroids of their Voronoi cells.

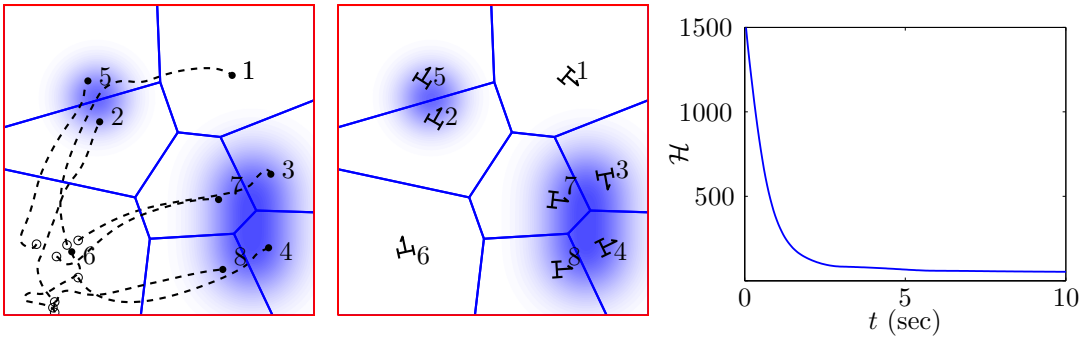


Figure 4.3: Wheeled vehicle deployment simulation. The agents start in the lower left corner, denoted by the 'o'. Path lines are shown in the left figure, and final positions and orientations in the right figure.

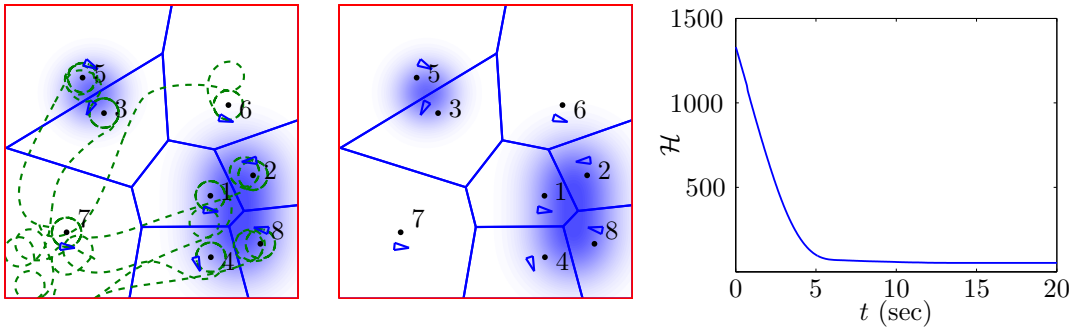


Figure 4.4: Fixed forward velocity deployment simulation. The agents start in the lower left corner and path lines are shown in the left figure with final positions and orientations shown in the right figure. Virtual center locations are denoted by a dot.

Chapter 5

Deployment with Environmental Constraints

In this chapter, we consider the case where the environment prevents an agent from maintaining a constant position, or even revisiting previous locations. Specifically, we will assume that the agents move in an *a priori* known flow field, but the field is faster than the absolute maximum speed of an agent. We wish to enable a group of cooperative vehicles to move such that the area of the reachable set of points in the flow environment is maximized. We will draw upon previous work on path planning and optimal control and apply it to cooperative motion control.

The problem of path planning in a flow field has its roots in the classical optimal control problem known as Zermelo's navigation problem [91], whose solution relies on use of the Pontryagin minimum principle as in [7]. Motion planning for single vehicles in a flow has been lately studied in e.g. [67, 11]. More recently, [51, 70, 82] characterizes the optimal trajectories for a Dubins-like vehicle in the presence of constant flows, while [50] presents a robust control strategy for a Dubins' vehicle in non-constant flows. The paper [71] provides a generalization of time-optimal course changes for a Dubins vehicle moving across a heterogeneous terrain, and across boundaries where the vehicle dynamics change.

Regarding cooperative motion of multiple vehicles in a flow environment, the authors in [59] have demonstrated stable formation control in the presence of

an external time-invariant flow field. However, that work assumes that individual vehicles have full controllability. Similarly, [40] addresses the case of maintaining a stable formation for sampling in an ocean environment with exogenous currents, while [81] combines time-optimal trajectories in a constant flow field with formation stabilization.

With the exception of [70], previous work has considered the motion control problem of vehicles in a slow flow. That is, the vehicles had sufficient actuation to move against the flow. We utilize a simple kinematic model as in [59] to consider flows which are both spatially varying and fast. Furthermore, much previous work, such as [70], consider constant flow fields. In the interest of obtaining analytic solutions, we consider piecewise constant flow fields as an approximation of smooth spatially varying flows. Such a consideration introduces several interesting effects regarding time-optimal trajectories over discontinuous flows. The work of [71] considers a similar situation, except that the effect of the environment only imposes a change in maximum velocity of a vehicle. This is not the case for the work in this chapter due to the directionality of flows in the environment.

This chapter expands on the results found in [37] and [39] by lifting technical assumptions related to the flow environment. Whereas the results in [39] for piecewise constant flows relied on assumptions relating how neighboring regions of different constant flows are joined, this chapter removes those assumptions and we incorporate obstacles in the flow as well. We study the different cases of time-optimal trajectories that may arise in a piecewise constant flow and use these cases as a basis of a gradient-based coverage motion algorithm.

This chapter is organized as follows. The next section details the multi-vehicle coverage problem and formally establishes definitions used throughout the chapter. We provide a synopsis of [37] in Section 5.2 for the case of smooth flow environments. We then study time-optimal trajectories in a piecewise constant flow in the next two sections. This allows for a definition of the partitioning scheme and a solution for the area coverage maximization problem presented in Section 5.5. Finally, we demonstrate the area maximizing algorithm in simulations found in Section 5.6.

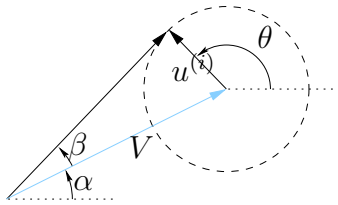


Figure 5.1: This diagram shows the different angular quantities for calculating $\beta(x)$, the maximum possible angle away from $\alpha(x)$ that results in a time-optimal trajectory in constant flow. The dashed circle represents the set of inputs $u^{(i)}$ with magnitude 1.

5.1 Problem statement and definitions

We assume the following kinematic model:

$$\dot{p}^{(i)} = u^{(i)} + V(p^{(i)}) \,, \tag{5.1}$$

where $u^{(i)}(t)$ is piecewise smooth and $\|u^{(i)}\| \leq 1$. Since we are concerned with time-optimal trajectories, $u^{(i)} = (\cos\theta^{(i)}, \sin\theta^{(i)})^{\mathsf{T}}$, see [7]. Assume that $\|V\| > 1$.

We define two quantities related to the flow field and motion of the agents, see Figure 5.1. The first is the flow direction:

$$\alpha(x) = \operatorname{atan2}\left(V_2(x), V_1(x)\right) \,. \tag{5.2}$$

Next, $\beta : X \to \left(0, \frac{\pi}{2}\right)$ denotes the maximum angle away from $\alpha(x)$ that results in a time-optimal trajectory in constant flow. More precisely, given $V(x)$ and $\|u^{(i)}\| = 1$, the endpoint of an agent's velocity vector (5.1) lies on the dashed circle of Figure 5.1. Using trigonometry, one can show that

$$\beta(x) = \arcsin\left(\frac{1}{\|V(x)\|}\right) \,. \tag{5.3}$$

Additionally, the choice of headings θ that result in these maximal travel directions are $\theta = \alpha(x) \pm (\beta(x) + \frac{\pi}{2})$. The feasible set of directions that an agent at $p^{(i)}$ would be able to travel in is then the interval $[\alpha(p^{(i)}) - \beta(p^{(i)}), \alpha(p^{(i)}) + \beta(p^{(i)})]$.

Definition 12 (Reachable set)**.** *We define the* reachable set*, $\mathcal{R}(p^{(i)})$, of an agent at position $p^{(i)}$ to be the set of points $x \in X$ that an agent can reach in finite time starting from the initial position $p^{(i)}$ and using a piecewise smooth control*

input $u^{(i)}(t)$ *with* $\|u^{(i)}\| \leq 1$. *The* T-limited reachable set, $\mathcal{R}_T(p^{(i)})$, *of an agent at position* $p^{(i)}$, *is the set of points that an agent can reach within time* T *using a piecewise smooth control input* $u^{(i)}(t)$ *with* $\|u^{(i)}\| \leq 1$.

Thus, $\mathcal{R}\colon X \rightrightarrows \mathbb{R}^2$ and $\mathcal{R}_T\colon X \rightrightarrows \mathbb{R}^2$ are set-valued maps.

Concerning solutions of a continuous ODE, we have the following result detailing the "closeness" of solutions to an ODE with initial conditions that are "close" to each other.

Theorem 28 ([34], Theorem 3.5)**.** *Let* $f(t, x, \lambda)$ *be continuous in* (t, x, λ) *and locally Lipschitz in* x *(uniformly in* t *and* λ*) on* $[t_0, t_1] \times D \times \{\|\lambda - \lambda_0\| < c\}$, *where* $D \subset \mathbb{R}^n$ *is an open connected set. Let* $y(t, \lambda_0)$ *be a solution of* $\dot{x} = f(t, x, \lambda_0)$ *with* $y(t_0, \lambda_0) = y_0 \in D$. *Suppose* $y(t, \lambda_0)$ *is defined and belongs to* D *for all* $t \in [t_0, t_1]$. *Then, given* $\epsilon > 0$, *there is* $\delta > 0$ *such that if*

$$\|z_0 - y_0\| < \delta \quad \textit{and} \quad \|\lambda - \lambda_0\| < \delta$$

then there is a unique solution $z(t, \lambda)$ *of* $\dot{x} = f(t, x, \lambda)$ *defined on* $[t_0, t_1]$, *with* $z(t_0, \lambda) = z_0$, *and* $z(t, \lambda)$ *satisfies*

$$\|z(t, \lambda - y(t, \lambda_0)\| < \epsilon\,, \quad \forall\, t \in [t_0, t_1]\,.$$

To keep the problem analytically tractable, we restrict our attention to a time invariant flow case and examine two distinct but complementary flow environments. First, we characterize the reachability sets for vehicles in affine flows and we use these to describe a vehicle deployment strategy in constant and affine flow fields. Then, we look at piecewise constant flow environments, where the optimal trajectories found for the constant flow case are adjusted as they traverse multiple flow regions.

Our objective will be to define a distributed algorithm for the deployment of agents in a flow environment. Specifically, we wish to maximize the area coverage metric:

$$\mathcal{H}(p^{(1)}, \ldots, p^{(n)}) = \int_{\bigcup \mathcal{R}_T(p^{(i)})} 1 dx\,. \tag{5.4}$$

This must be done while taking into account the flow environment and how it affects the dynamics of each agent. In order to do this, we will follow the next

steps: (a) determine minimum time trajectories in the flow environment, and (b) use knowledge of the properties of these optimal trajectories to compute a gradient direction. We will present the above steps for the two flow scenarios described previously.

5.2 Affine flows

In this section we characterize the reachable set of an agent in an affine flow field. We will analyze some of the topological properties of these sets.

For such flows, we have the following standing assumption:

Assumption 2. *The coverage domain X is simply connected (has no holes) and the vector field V describing the flow is:*

1. *time invariant and affine over X, $V(x) = Ax + b$, and*
2. *$\|V(x)\| > 1$ for all $x \in X$.*

5.2.1 Properties of the reachable set

Shown previously, an agent cannot turn more than $\beta(x)$ radians away from the flow direction $\alpha(x)$ for any $x \in X$. Thus, by choosing $u^{(i)}$ such that an agent is always heading $\pm\beta$ away from α, we can compute the boundary of the reachable set.

To facilitate the characterization of the reachable sets, we consider the class of control inputs:

$$u^{(i)}(p^{(i)}, K) = \begin{bmatrix} \cos(\alpha(p^{(i)}) + K\left(\beta(p^{(i)}) + \frac{\pi}{2}\right)) \\ \sin(\alpha(p^{(i)}) + K\left(\beta(p^{(i)}) + \frac{\pi}{2}\right)) \end{bmatrix}, \tag{5.5}$$

for $K \in [-1, 1]$. We define the trajectory $\gamma(t, K)$ to be the solution of (5.1) using (5.5) for a fixed K with initial position $p^{(i)}$. Then, the two trajectories $\gamma(t, -1)$ and $\gamma(t, 1)$ define the boundary of the reachable set $\mathcal{R}(p^{(i)})$.

We say that a basis of vectors $\{a, b\}$ in the plane is *positively* (resp. *negatively*) *oriented* if the vector cross product in $\mathbb{R}^3$, $a \times b$, yields a vector pointing in

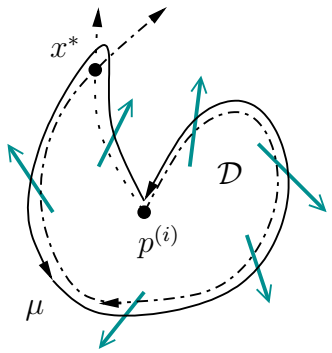


Figure 5.2: Cartoon to aid in proof of Proposition 29. The boundary $\gamma(t, 1)$ is shown as the dotted path and $\gamma(t, -1)$ is the dot-dashed path. The flow field is shown as thick gray arrows. If the two trajectories intersect at a point other than $p^{(i)}$ then the intersection must be oriented as shown at x^*.

a positive (resp. negative) direction out of the plane. The next two results address properties of the reachable set.

Proposition 29. *Under Assumption 2, the boundaries* $\gamma(t, 1)$ *and* $\gamma(t, -1)$ *intersect only at* $\gamma(0, 1) = \gamma(0, -1) = p^{(i)}$.

Proof. We show this by contradiction. Suppose $\gamma(t, 1)$ and $\gamma(t, -1)$ intersect at some point x^* and time t^* other than the initial condition $p^{(i)}$. Since $\beta \in \left(0, \frac{\pi}{2}\right)$, the angle between the velocity vectors $\dot{\gamma}(t^*, -1)$ and $\dot{\gamma}(t^*, 1)$ is less than π. Furthermore, the velocity vectors $\{\dot{\gamma}(t^*, -1), \dot{\gamma}(t^*, 1)\}$ form a positively oriented basis by definition, see Figure 5.2.

Consider the oriented closed curve μ starting from $p^{(i)}$, following $\gamma(t, 1)$ until x^*, and returning to $p^{(i)}$ along $\gamma(t, -1)$. By the definition given in (5.5), we have that $\{\dot{\gamma}(t, -1), V(\dot{\gamma}(t, -1))\}$ and $\{V(\dot{\gamma}(t, 1)), \dot{\gamma}(t, 1)\}$ are positively oriented bases; see Figure 5.2. Thus, along μ the flow field V must always be pointing outwards.

Let $\mathcal{D}$ be the connected and compact region enclosed by C. Since X has no holes, $\mathcal{D} \subset X$. Let $\tilde{V} = -V$, then $\mathcal{D}$ is a positively invariant set under the flow $\dot{x} = \tilde{V}(x)$ since $\tilde{V}$ points inwards for all $x \in \partial\mathcal{D} = \mu$. Thus by a simple extension of the Brouwer Fixed point theorem [3], there exists an $x \in \mathcal{D}$ such that $\tilde{V}(x) = 0$. This implies that $V(x) = 0$, a contradiction under Assumption 2. □

Theorem 30. *Under Assumption 2, the set of reachable points using* (5.5) *for* $K \in [-1, 1]$ *has no holes, and is equal to the set* $\mathcal{R}(p^{(i)})$.

Proof. We first prove the absence of holes. Suppose that there is a hole, $\mathcal{D} \subset X$, in the reachable set. The set $\mathcal{D}$ is bounded and connected, but there does not exist a control input $u^{(i)}$ such that the solution of (5.1) enters $\mathcal{D}$. However, given $\epsilon > 0$, there exists a $r \in \mathcal{R}(p^{(i)})$ such that $\inf_{x \in \mathcal{D}} \|r - x\| < \epsilon$. That is, we can get arbitrarily close to $\mathcal{D}$.

Consider the closure $\bar{\mathcal{D}}$. Along $\partial\bar{\mathcal{D}}$, the flow field V must always point outwards from (or tangent to) $\mathcal{D}$ at $\partial\mathcal{D}$. If not, then solutions of (5.1) can enter $\mathcal{D}$ for $u^{(i)} = 0$. If we consider the reverse flow $\tilde{V} = -V$, then $\bar{\mathcal{D}}$ is an invariant set. Similar to the proof of the previous result, this implies that there exists a point $x^* \in \mathcal{D}$ such that $V(x^*) = 0$, a contradiction of Assumption 2. Thus, no holes in the reachable set exist.

In conjunction with Proposition 29, that $\mathcal{R}(p^{(i)})$ has no holes implies that the entire space between the curves $\gamma(t, -1) = \gamma(t, -1)$ and $\gamma(t, 1) = \gamma(t, 1)$ is the reachable set $\mathcal{R}(p^{(i)})$. Now we show that the set of solutions $\gamma(t, K)$ for fixed $K \in [-1, 1]$ can reach any point in the reachable set.

Let $\tilde{\mathcal{R}}(p^{(i)})$ be the set of points corresponding to solutions of $\gamma(t, K)$. Suppose that $\mathcal{D}$ is a hole in this restricted reachable set. That is, $\mathcal{D} \subset \mathcal{R}(p^{(i)})$, but $\mathcal{D} \cap \tilde{\mathcal{R}}(p^{(i)}) = \emptyset$. Then it cannot be that $\gamma(t, -1)$ or $\gamma(t, 1)$ reach a point in $\partial\mathcal{D}$ for some t. Otherwise, suppose that there exists $x_0 \in \partial\mathcal{D}$ such that $\gamma(t_0, -1) = x_0$ for some t_0. Take $T > t_0$ then, by Theorem 3.5 in the standard reference [34], for every $\epsilon > 0$ there is a $\delta > 0$ such that for every $|K - (-1)| < \delta$, it must be that $\|\gamma(t, -1) - \gamma(t, K)\| < \epsilon$ for all $t \leq T$. In particular, $\|x_0 - \gamma(t_0, K)\| < \epsilon$.

Consider one such solution $\gamma(t, K_1)$ satisfying $0 < |K_1 - (-1)| < \delta$. By continuity of solutions and for sufficiently large T, in order for $\gamma(t, K_1)$ to stay outside of $\mathcal{D}$, there must be some t' such that $\gamma(t', K_1) = x_0$. However, this is a contradiction because $u(p^{(i)}, -1)$ is the unique control input to reach points along the boundary of the reachable set.

In other words, we can guarantee that there is a $K_0 \in (-1, 1)$ sufficiently close to -1 such that the set of solutions for $K \in [-1, K_0]$ does not have any

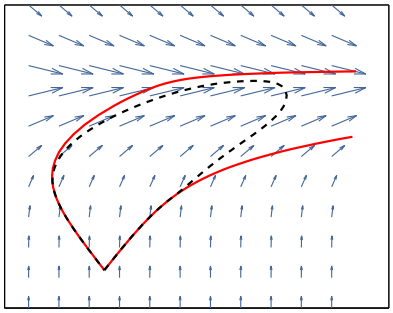


Figure 5.3: An example of the reachable set $\mathcal{R}(p^{(i)})$ with solid line and the T-reachable set in the dashed line.

holes in it. Now, it is easy to see that $\{\gamma(K,t)\,|\,K \in [-1,K_0]\}$ is homotopically equivalent to the set $\{\gamma(K,t)\,|\,K \in [-1,1]\}$ by using a continuous deformation from $[-1,K_0]$ to $[-1,1]$. Since the property of being simply connected is a homotopic invariant, the set $\{\gamma(K,t)\,|\,K \in [-1,1]\}$ can not have any holes in it. □

The fact that there are no holes in the reachable set greatly simplifies the analysis of area coverage for differentiable flows. This insures that the boundaries of the reachable set come only from the extremal paths $\gamma(t,\pm 1)$, and intersections of $\mathcal{R}(p^{(i)})$ with the flow domain boundary ∂X.

5.2.2 T-limited reachable set

In order to find $\mathcal{R}_T(p^{(i)})$, one must then solve the following minimum-time optimal control problem:

$$\begin{aligned} \text{minimize:} \quad & J = \int_0^{t_f} 1dt\,, \\ \text{subject to:} \quad & \dot{p}^{(i)} = u^{(i)} + V(p^{(i)})\,, \|u^{(i)}\| \leq 1\,, \\ & p^{(i)}(0) \text{ and } p^{(i)}(t_f) \text{ given}\,. \end{aligned} \tag{5.6}$$

For a smooth flow field V, this is known as Zermelo's problem, and a solution can be found [7]. The optimal solution is to consider a control input of the form

$$\begin{aligned} u^{(i)} &= \begin{pmatrix} \cos\theta^{(i)} \\ \sin\theta^{(i)} \end{pmatrix}, \\ \dot{\theta}^{(i)} &= \sin^2\theta^{(i)} \frac{\partial V_2}{\partial x_1} + \sin\theta^{(i)} \cos\theta^{(i)} \left(\frac{\partial V_1}{\partial x_1} - \frac{\partial V_2}{\partial x_2} \right) \\ &\quad - \cos^2\theta^{(i)} \frac{\partial V_1}{\partial x_2}. \end{aligned} \tag{5.7}$$

The minimum-time trajectories are obtained by using this input in combination with (5.1).

To define the T-limited boundary, one can integrate (5.1) using (5.7) to time T starting at the agent location $p^{(i)}$ and initial heading $\theta^{(i)}(0) \in [\alpha-\beta-\frac{\pi}{2}, \alpha+\beta+\frac{\pi}{2}]$. The solutions for various initial headings at time T are then recorded and combined with the solutions $\gamma(t,-1)$ and $\gamma(t,1)$ for $t \in [0,T]$ to give the boundary of $\mathcal{R}_T(p^{(i)})$. See Figure 5.3 for an example of such a set. Finally, we define $\tau\colon X \times X \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ to be the minimum travel time between two points in X. The minimum travel time $\tau(x,y)$ is from the solution of the optimal control problem (5.6) for an agent starting at x and ending at y. Due to the flow environment, in general, $\tau(x,y) \neq \tau(y,x)$.

Remark 31. We restrict ourselves to affine flows due to the fact that the Pontryagin Minimum principle is a necessary condition for optimality. Solutions satisfying (5.7) may be maximum-time solutions, or may no longer be time-optimal after a point along a solution. For the Zermelo problem with fast flows, there will be a set of initial heading choices that result in maximum-time solutions. For any feasible direction of travel in a fast flow, there are two possible initial headings that result in motion along this feasible direction. One corresponds to moving with the flow, and the other moves against the flow. The initial headings that move against the flow correspond to $\theta_0 \notin [\alpha-(\beta+\frac{\pi}{2}), \alpha+(\beta+\frac{\pi}{2})]$. See Figure 5.4 for a graphical explanation of this.

The case of a time-optimal solution that ceases to be optimal is what is known in the optimal control literature as a *conjugate point*. Roughly speaking,

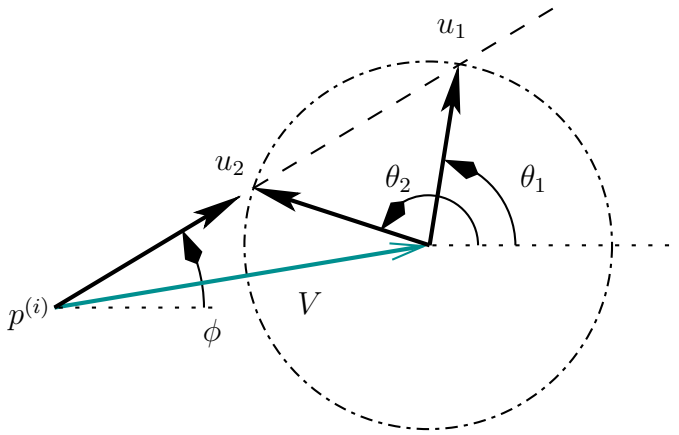


Figure 5.4: Diagram for time optimality in a fast flow. Suppose agent i wishes to travel in the feasible direction ϕ. Using vector addition and trigonometric arguments, one can show that there exist two possible control inputs u_1, u_2 with headings of θ_1, θ_2 that achieve motion in the direction of ϕ. However, clearly one choice results in faster motion along ϕ, and hence, that choice is time-minimizing.

a conjugate point along an optimal trajectory (or geodesic) occurs wherever that geodesic intersects another geodesic. However, due to the result in [76], for the case of affine flow fields, optimal trajectories do not have conjugate points. In this way, T can be arbitrarily large without risk of incorrectly approximating the T-limited reachable set by joining the endpoints of solutions of the Pontryagin minimum principle, (5.6) and (5.7). •

We say that agents i and j are *reachable set neighbors* if $\mathcal{R}_T(p^{(i)}) \cap \mathcal{R}_T(p^{(j)}) \neq \emptyset$. Thus, agent i is trivially a neighbor of itself. Additionally, the *neighbor set* of i, is

$$\mathcal{N}^{(i)}_{\text{flow}} = \{j \in \{1, \ldots, n\} \mid \mathcal{R}_T(p^{(i)}) \cap \mathcal{R}_T(p^{(j)}) \neq \emptyset\}\,.$$

5.2.3 Special flow examples

Following Assumption 2, we consider affine flows of the form

$$V(x) = Ax + b\,, \tag{5.8}$$

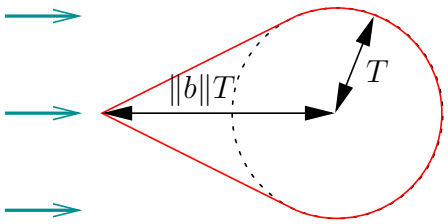


Figure 5.5: The T-limited reachable set $\mathcal{R}_T(p^{(i)})$ when the flow field is a constant. The T-limited boundary connecting $\gamma(t,-1)$ and $\gamma(t,1)$ is a circular arc centered $\|b\|T$ units downstream with radius T.

where $A \in \mathbb{R}^{2\times 2}$ and $b \in \mathbb{R}^2$ are arbitrary constants. The dynamics (5.1) take the form

$$\dot{p}^{(i)} = Ap^{(i)} + b + u^{(i)} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} p^{(i)} + \begin{pmatrix} b_1 + \cos\theta^{(i)} \\ b_2 + \sin\theta^{(i)} \end{pmatrix}, \quad i \in \{1,\dots,N\}. \tag{5.9}$$

The time-optimal steering input given by (5.7) with the affine flow parameters is

$$\dot{\theta^{(i)}} = a_{21}\sin^2\theta^{(i)} + (a_{11} - a_{22})\sin\theta^{(i)}\cos\theta^{(i)} - a_{12}\cos^2\theta^{(i)}, \quad i \in \{1,\dots,N\}.$$

Note that the above differential equation is separable, and one can obtain an explicit solution for $\theta^{(i)}(t)$ up to quadratures and inversion. The system described by (5.9) is linear in $p^{(i)}$, and so the time-optimal trajectories in an affine flow are given by

$$p^{(i)}(t) = e^{At}p^{(i)}(0) + \int_0^t e^{A(t-s)}(b + u^{(i)}(s))ds, \quad i \in \{1,\dots,N\}. \tag{5.10}$$

For the constant flow case, $A = 0$ and $\|b\| > 1$, the set $\mathcal{R}_T(p^{(i)})$ resembles Figure 5.5. The curves $\gamma^{(i)}(t,-1)$ and $\gamma^{(i)}(t,1)$ are straight line segments at angles $\pm\beta$ away from the flow direction. They are joined by a circular arc with radius T centered at a distance $\|b\|T$ downstream.

Figure 5.6 depicts some typical variations on flows of the form (5.8). In areas of faster flow, the reachable sets are narrower and in regions of flow where $\|V(x)\|$ is only slightly larger than 1, the reachable sets tend towards a circular appearance.

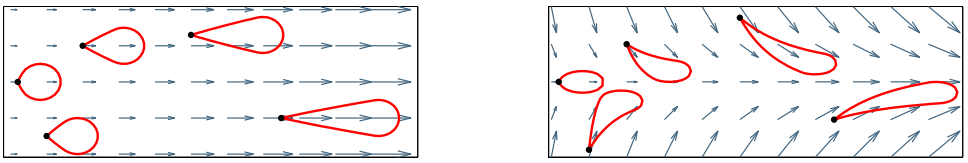


Figure 5.6: Examples of reachable set regions $\mathcal{R}_T(p^{(i)})$ with flows of the form (5.8). Agents have a maximum speed of 1, and $T = 5$. Top: $a_{11} = 0.05$, $b_1 = 1$, $a_{22} = b_2 = 0$. Bottom: $a_{11} = 0.05$, $b_1 = 1$, $a_{22} = -0.25$ and $b_2 = 0$.

5.3 Piecewise constant flows – simple optimal trajectories

This section and the following section detail the various types of optimal trajectories that may occur in a piecewise constant flow. We will work now under the following assumption:

Assumption 3. *The flow environment X may have obstacles and:*

1. *the flow V is piecewise constant. That is, we can partition X into m regions, $X = \bigcup_{k=1}^{m} X_k$ such that $V_{|X_k}$ is constant and satisfies $\|V_{|X_k}\| > 1$ for all k,*

2. *the regions X_k, $k \in \{0, \ldots, m\}$ are separated by piecewise differentiable curves. Let $\psi_{k,\ell} \colon X_k \to \mathbb{R}$ be piecewise differentiable and the curve defined by $\psi_{k,\ell}(x) = 0$ separates the flow regions k and $\ell \in \{1, \ldots, m\}$, and*

3. *along the interface between two flows k and ℓ, the flow velocity is in the convex hull of the two respective velocities. That is, $V(x) \in \text{co}\{V_{|X_k}, V_{|X_\ell}\}$ for $\{x \mid \psi_{k,\ell}(x) = 0\}$.* •

Under this assumption, the optimal paths in the interior of each region X_k will be straight lines since the flow in each region is constant. Suppose an agent begins with an initial position of p_0 and heading θ_0. Then, there will be a sequence of optimal course changes corresponding to each time an agent travels into a new flow region. We enumerate these with a subscript ℓ. Also, $\bar{t}_\ell$ and $\bar{x}_\ell$ will be the time and location of a trajectory's crossing from the $\ell - 1$th region to the ℓth region,

with the convention that $\bar{t}_0 = 0$ and $\bar{x}_0 = p_0$. When necessary, we denote the first and second components of $\bar{x}_\ell$ as $\bar{x}_{\ell,1}$ and $\bar{x}_{\ell,2}$, respectively.

5.3.1 Catalog of optimal trajectories

Before launching into the detailed description of each type of trajectory and the manner in which they occur, we provide a brief overview of the various types of optimal trajectories in a piecewise constant flow. This section and the next will detail what can occur if trajectories intersect either the flow domain boundaries, ∂X, or the boundaries between two neighboring flows, $\partial X_k \setminus \partial X$.

In the case that a trajectory intersects the flow domain boundary ∂X, one side of the boundary is where an agent may perform motion. The other side represents an impassable region, either a constraint in allowable positions imposed by the user or an obstacle in the flow field. A trajectory can either intersect ∂X transversely or tangentially. In the case of a transverse intersection, a collision effectively occurs and the trajectory does not continue. However, interesting results can occur for tangential intersections, and this will be detailed in the next main section.

For the case that a trajectory intersects a boundary between two flows, defined as a level set where $\psi_{k,\ell}(x) = 0$, this intersection can again occur either transversely or tangentially. The former case will be addressed in this section while the latter will be examined in the next section.

5.3.2 Simple trajectories

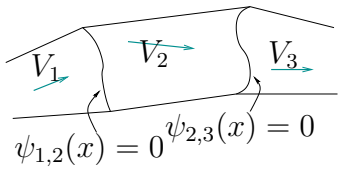


Figure 5.7: A general piecewise flow case

To determine optimal trajectories for a simple two-region switched system, we determine the direction change that the optimal paths undergo at the interface

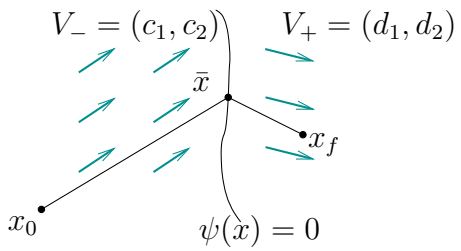


Figure 5.8: Diagram for deriving the relation between incoming and outgoing headings at the interface between two regions of different, but constant, flows.

.

between two regions. The following analysis is detailed further in [7]. Optimal trajectories in this section's formulation will be called *simple trajectories*.

Referencing Figure 5.8, the vehicle begins at x_0, the interface between two regions X_1, X_2, occurs at the smooth level set of $\psi(x) = 0$, and the two constant flows are given by $V_- = (c_1, c_2)^\mathsf{T}$ and $V_+ = (d_1, d_2)^\mathsf{T}$. We wish to solve for the minimum-time trajectory from x_0 to a point x_f after crossing the interface between the two flows at some point $\bar{x}$. For minimum time problems, the cost function will be:

$$J = \int_0^{t_f} L(x, u, t)dt = \int_0^{t_f} 1dt = t_f \,.$$

The vehicle has dynamics $\dot{x} = F_-(x, \theta)$ or $\dot{x} = F_+(x, \theta)$ depending on which side of the flow interface it is on, where:

$$F_-(x_1, x_2, \theta) = \begin{pmatrix} \cos\theta + c_1 \\ \sin\theta + c_2 \end{pmatrix}, \quad \text{if } x \in X_1,$$

$$F_+(x_1, x_2, \theta) = \begin{pmatrix} \cos\theta + d_1 \\ \sin\theta + d_2 \end{pmatrix}, \quad \text{if } x \in X_2 \,.$$

We also have the interior point boundary condition that for some time $0 < \bar{t} < t_f$, $\psi(x(\bar{t})) = 0$. One can now define the Hamiltonians for each set of dynamics as:

$$H_1(x, \theta, \lambda) = 1 + \lambda_1(\cos\theta + c_1) + \lambda_2(\sin\theta + c_2)\,,$$
$$H_2(x, \theta, \lambda) = 1 + \lambda_1(\cos\theta + d_1) + \lambda_2(\sin\theta + d_2)\,,$$

where $\lambda = (\lambda_1, \lambda_2)^\mathsf{T}$ is the costate vector. The interior boundary condition can be viewed as an additional constraint occurring at time $\bar{t}$. In this way, we adjoin the

interior point constraint to the costate and Hamiltonian with a Lagrange multiplier $\nu \in \mathbb{R}$. Let $\bar{t}_-$ and $\bar{t}_+$ denote the time immediately before and after $\bar{t}$, respectively. Then,

$$\lambda^{\mathsf{T}}(\bar{t}_-) = \lambda^{\mathsf{T}}(\bar{t}_+) + \nu \frac{\partial \psi}{\partial x}\,,$$
$$H_1(x(\bar{t}_-), \theta(\bar{t}_-), \lambda(\bar{t}_-)) = H_2(x(\bar{t}_+), \theta(\bar{t}_+), \lambda(\bar{t}_+)) + \nu \frac{\partial \psi}{\partial t}.$$

We have $H_1(x(\bar{t}_-), \theta(\bar{t}_-), \lambda(\bar{t}_-)) = H_2(x(\bar{t}_+), \theta(\bar{t}_+), \lambda(\bar{t}_+))$ because there is no time dependence in ψ. Furthermore, for minimum-time optimal control problems with time-invariant dynamics, the Hamiltonian is constant and equal to zero [7]. The costate jump at $\bar{t}$ is described by

$$\lambda_1(\bar{t}_-) = \lambda_1(\bar{t}_+) + \nu \frac{\partial \psi}{\partial x_1}\,, \qquad \lambda_2(\bar{t}_-) = \lambda_2(\bar{t}_+) + \nu \frac{\partial \psi}{\partial x_2}\,.$$

Using optimality with respect to the control input θ, we derive an additional set of relations:

$$\frac{\partial H_1}{\partial \theta} = -\lambda_1(\bar{t}_-)\sin\theta_- + \lambda_2(\bar{t}_-)\cos\theta_- = 0 \quad \Rightarrow \tan\theta_- = \frac{\lambda_2(\bar{t}_-)}{\lambda_1(\bar{t}_-)}\,,$$

And similarly $\tan\theta_+ = \dfrac{\lambda_2(\bar{t}_+)}{\lambda_1(\bar{t}_+)}$.

Currently, there is enough information to develop a relation between the constant control inputs θ_- and θ_+.

Proposition 32. *Let $V_- = (c_1, c_2)^{\mathsf{T}}$ and $V_+ = (d_1, d_2)^{\mathsf{T}}$ be the flows in two neighboring regions, and α_1, α_2 be their respective flow orientations. Let ξ be the orientation of the normal vector of the smooth curve $\psi(x) = 0$ at the point where the optimal trajectory crosses into the second flow region. A necessary condition for an optimal trajectory across the interface of the two flow regions requires that:*

$$\frac{1 + \|V_-\|\cos(\theta_- - \alpha_1)}{\sin(\theta_- - \xi)} = \frac{1 + \|V_+\|\cos(\theta_+ - \alpha_2)}{\sin(\theta_+ - \xi)}. \tag{5.11}$$

Given (5.11)*, and a fixed heading θ_-, the final heading satisfies*

$$\sin\theta_+ = \frac{B \pm C\sqrt{B^2 + C^2 - 1}}{B^2 + C^2}\,, \tag{5.12}$$

where

$$B = \frac{1 + \|V_-\| \cos(\theta_- - \alpha_1)}{\sin(\theta_- - \xi)} \cos\xi - d_2\,,$$
$$C = \frac{1 + \|V_-\| \cos(\theta_- - \alpha_1)}{\sin(\theta_- - \xi)} \sin\xi + d_1\,.$$

Proof. The above analysis leads to the following six equations:

$$1 + \lambda_1(\bar{t}_-)(\cos\theta_- + c_1) + \lambda_2(\bar{t}_-)(\sin\theta_- + c_2) = 0\,,$$
$$1 + \lambda_1(\bar{t}_+)(\cos\theta_+ + d_1) + \lambda_2(\bar{t}_+)(\sin\theta_+ + d_2) = 0\,,$$
$$\lambda_1(\bar{t}_-) = \lambda_1(\bar{t}_+) + \nu\frac{\partial\psi}{\partial x_1}\,,$$
$$\lambda_2(\bar{t}_-) = \lambda_2(\bar{t}_+) + \nu\frac{\partial\psi}{\partial x_2}\,,$$
$$\tan\theta_- = \frac{\lambda_2(\bar{t}_-)}{\lambda_1(\bar{t}_-)}\,,$$
$$\tan\theta_+ = \frac{\lambda_2(\bar{t}_+)}{\lambda_1(\bar{t}_+)}\,,$$

with the six unknowns: ν, θ_+, $\lambda_1(\bar{t}_-)$, $\lambda_2(\bar{t}_-)$, $\lambda_1(\bar{t}_+)$, and $\lambda_2(\bar{t}_+)$. Eliminating all but θ_+, we obtain the relation

$$\left(\frac{\partial\psi}{\partial x_1}\tan\theta_+ - \frac{\partial\psi}{\partial x_2}\right)(1 + c_1\cos\theta_- + c_2\sin\theta_-)\sec\theta_-$$
$$= -\left(\frac{\partial\psi}{\partial x_2} - \frac{\partial\psi}{\partial x_1}\tan\theta_-\right)(1 + d_1\cos\theta_+ + d_2\sin\theta_+)\sec\theta_+\,.$$

Substitute $(c_1, c_2) = (\|V_-\|\cos\alpha_1,\ \|V_-\|\sin\alpha_1)$ and $(d_1, d_2) = (\|V_+\|\cos\alpha_2,$ $\|V_+\|\sin\alpha_2)$. The relation becomes

$$\frac{1 + \|V_-\|\cos(\theta_- - \alpha_1)}{\frac{\partial\psi}{\partial x_2}\cos\theta_- - \frac{\partial\psi}{\partial x_1}\sin\theta_-} = -\frac{1 + \|V_+\|\cos(\theta_+ - \alpha_2)}{\frac{\partial\psi}{\partial x_1}\sin\theta_+ - \frac{\partial\psi}{\partial x_2}\cos\theta_+}\,.$$

Let ξ be the angle of the gradient vector $\nabla\psi$. That is, $\xi = \arctan\left(\frac{\frac{\partial\psi}{\partial x_2}}{\frac{\partial\psi}{\partial x_1}}\right)$. This direction is normal to the level set $\psi(x_1, x_2) = 0$. Then, the substitution $\left(\frac{\partial\psi}{\partial x_1}, \frac{\partial\psi}{\partial x_2}\right) = (\|\nabla\psi\|\cos\xi,\ \|\nabla\psi\|\sin\xi)$, leads to (5.11).

The cosine functions in the numerators of (5.11) can be expanded. Then performing the substitution $\cos\theta_+ = \sqrt{1 - \sin^2\theta_+}$ yields the equation

$$(A^2 + B^2)\sin^2\theta_+ + 2A\sin\theta_+ + 1 - B^2 = 0\,.$$

This is quadratic in $\sin\theta_+$, which results in (5.12). ∎

Remark 33. This is a direct analogue to Snell's law as it describes the relation between incident and refracted angles for a beam of light, disregarding the light beam's origin. Finally, the extension to multiple region flows is straightforward; see Figure 5.9 for a graph of the reachable set. •

Remark 34. For certain choices of initial headings θ_-, the system of equations in the proof of Proposition 32 may have no solution. These special angles correspond to the choice of extreme headings $\theta_- = \alpha_1 \pm \left(\beta_1 + \frac{\pi}{2}\right)$. However, the solution of θ_+ as a function of θ_- has a well-defined limit as $\theta_- \to \alpha_1 \pm \left(\beta_1 + \frac{\pi}{2}\right)$

Additionally, it may be that there is no solution for θ_+ in (5.12). This may occur when $B^2 + C^2 - 1 < 0$. However, since (5.12) is a necessary condition for optimality, the fact that there is no real-valued θ_+ satisfying (5.12) implies that there is no optimal continuation of a particular trajectory that intersects the boundary between two flows. An example of this occurs in Figure 5.18 and we will study this case in a later section. •

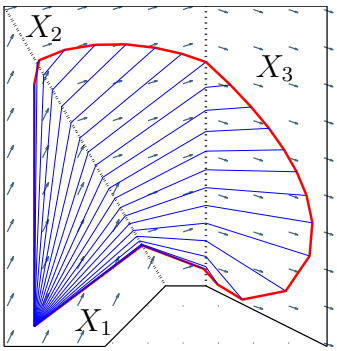


Figure 5.9: Trajectories of solutions for the case of a three region flow with $V_- = (1, 2)^\mathsf{T}$, $V_+ = (1.5, 0.5)^\mathsf{T}$ and $V_3 = (1.5, -0.5)^\mathsf{T}$. The boundaries of the regions are straight lines with angles $\xi_{1,2} = \frac{\pi}{6}$ and $\xi_{2,3} = 0$.The initial position of the agent is at $(1.5, 1)$ and $T = 6$.

5.4 Piecewise constant flows–optimal non-simple trajectories

We now study how consideration of flow boundaries, obstacles, or boundaries between two neighboring flow regions affect the optimal trajectories. We refer to these optimal trajectories as *non-simple trajectories.*

5.4.1 Obstacles

By obstacles in the flow, we refer to both holes in the flow environment and boundaries marking impassable regions. Such phenomena may impede a simple trajectory. We define the *simply reachable set* $\mathcal{S}(p_0)$ to be the set of points reachable with a simple trajectory, and the *simply unreachable set* to be $\mathcal{U}(p_0) = \mathcal{R}(p_0) \setminus \mathcal{S}(p_0)$. See Figure 5.15 for such an example.

Before introducing the next lemma, we define the path concatenation operator, $*$. Given two parametrized paths $\rho_1, \rho_2 \colon [0,1] \to \mathbb{R}^2$, such that $\rho_1(1) = \rho_2(0)$, the concatenation of the two paths, $\rho = \rho_1 * \rho_2$ is such that $\rho(0) = \rho_1(0)$ and $\rho(1) = \rho_2(1)$.

Lemma 35. *Let ρ be a feasible trajectory in a constant flow environment without obstacles from $\rho_0 = \rho(0)$ to $\rho_f = \rho(1)$. Choose $s_1, s_2 \in [0,1]$ with $s_1 < s_2$ and let ρ_1 be the section of $\rho(s)$ for $0 \leq s < s_1$ and ρ_2 be the section of $\rho(s)$ for $s_2 \leq s \leq 1$. Let the function J take a parametrized path ρ and return the time required to traverse it in the constant flow. Suppose μ is the straight line path from $\rho(s_1)$ to $\rho(s_2)$. Then the concatenation $\tilde{\rho} = \rho_1 * \mu * \rho_2$ has the property that $J(\tilde{\rho}) \leq J(\rho)$.*

Proof. Since there are no obstacles, the point $\rho(s_2)$ is reachable from $\rho(s_1)$ via a straight line. Finally, straight line paths in a constant flow environment are time-optimal, so by replacing the middle section of ρ with a straight line segment in $\tilde{\rho}$, we have decreased the travel time along $\tilde{\rho}$ compared to ρ. □

By replacing a section of a trajectory with a straight line path, the time to traverse that path is shortened, see Figure 5.10 for a graphical interpretation.

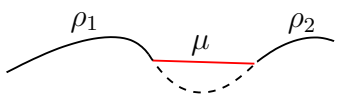


Figure 5.10: A graphical interpretation of Lemma 35. The original path ρ is the concatenation of ρ_1, the dashed path, and ρ_2. Replacing the dashed path with μ (straight line) results in a path that takes less time to traverse.

Next, we consider the situation when an optimal trajectory intersects an obstacle (or flow environment boundary). We present a definition of convexity with respect to the flow environment.

Definition 13. *Let ζ be a parametrized curve in the flow environment. Then, ζ is* locally convex *at $\zeta(\bar{s})$ relative to the flow domain X if there exists a $\delta > 0$ such that for all $s \in (\bar{s}-\delta, \bar{s}+\delta)$, $s \neq \bar{s}$, and for all $t \in [0,1]$, we have $\zeta(\bar{s}) + t(\zeta(s) - \zeta(\bar{s})) \in X$. Similarly, ζ is* locally concave *at $\zeta(\bar{s})$ relative to X if for $t \in (0,1)$, $\zeta(\bar{s}) + t(\zeta(s) - \zeta(\bar{s})) \notin X$.*

The convex (resp. concave) curve ζ is strictly convex *(resp.* strictly concave*) if for all $t \in (0,1)$, $\zeta(\bar{s}) + t(\zeta(s) - \zeta(\bar{s})) \neq \zeta(\tilde{s})$ for some $\tilde{s} \in (\bar{s}-\delta, \bar{s}+\delta)$.*

Figure 5.11 offers a graphical depiction of the above definition. With these definitions we can state the following result regarding non-simple paths.

Figure 5.11: Diagram illulstrating convexity relative to the flow environment. At left, an example of a locally strictly convex boundary with a dashed tangent line. At right is a locally strictly concave curve with a dashed tangent line.

Theorem 36. *Let ζ be a parametrization of the boundary of an obstacle. If ζ is locally strictly concave at the point $\zeta(s)$, then time-optimal trajectories that are tangent at $\zeta(s)$ may result in non-simple paths. These paths are formed by following the boundary ζ and then tangentially leaving as a straight line path into the interior of X.*

Proof. Let ρ be a time-optimal trajectory originating from p_0 that is tangent to ζ at $\zeta(s)$. We note that if a time-optimal trajectory intersects ζ transversely, then that trajectory must stop. First we show that ζ must be locally concave about the intersection point $\zeta(s)$ in order to admit non-simple paths. Then for the case that ζ is locally concave at $\zeta(s)$, we will construct non-simple (optimal) paths.

Suppose that ζ is locally strictly convex at $\zeta(s)$ and that ρ is tangent to ζ at $\zeta(s)$. The time-optimal trajectory ρ must be feasible, thus it must remain in X at all times. For ρ to continue past $\zeta(s)$, it must continue to follow the curve ζ or return into the interior of the flow region. Following the boundary is not optimal since ζ is locally convex, so there exists a neighborhood N about $\zeta(s)$ such that there is a straight line path joining $\zeta(s)$ to another point along ζ through the interior of X that also lies in N, see Figure 5.12. If ζ is locally convex but not strictly convex about $\zeta(s)$, then ζ is a straight line through $\zeta(s)$. Thus, straight line trajectories may continue, and ρ remains a simple trajectory.

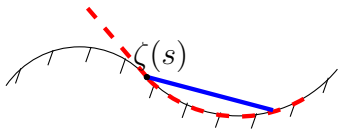


Figure 5.12: Convex boundary counterexample. The thick dashed path that follows the boundary of the obstacle is not time-optimal since there exists a straight line path from $\zeta(s)$ to any point along the convex boundary.

For the case that ζ is locally strictly concave at $\zeta(s)$, then the boundary of the obstacle occludes a portion of the flow domain X. Non-simple trajectories may result after this tangential intersection if it is feasible to steer along the path ζ under the constant flow. Let N be a small neighborhood about the tangential intersection point $\zeta(s)$. We first show that a trajectory that follows the boundary ζ is time-optimal. Then we combine this with the fact that within the interior of X, time-optimal paths must be straight lines in order to complete the construction of these optimal non-simple trajectories.

Consider a trajectory ρ that connects $\zeta(s_1)$ to $\zeta(s_2)$ on the boundary of a locally strictly concave curve ζ. Suppose ρ follows the boundary ζ, but is not time-optimal. Then there exists a trajectory ρ^* connecting $\zeta(s_1)$ and $\zeta(s_2)$ with

a shorter travel time. Furthermore, ρ^* must leave the boundary ζ at some point and then re-enter it in order to finish at $\zeta(s_2)$. However, since ζ is locally strictly concave, such a maneuver requires a course change within the interior of the flow region, which implies that ρ^* is not time-optimal, see Figure 5.13.

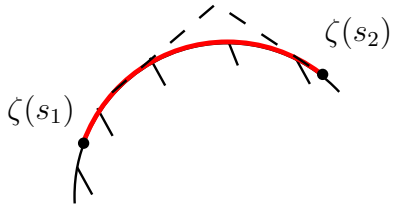


Figure 5.13: A diagram demonstrating optimality of a path the follows the boundary of an obstacle. Since ζ is locally strictly concave, any other path connecting $\zeta(s_1)$ to $\zeta(s_2)$ must undergo a path change within the interior of a flow region (i.e. the dotted line). This violates the optimality condition that straight-line paths within the interior of a flow region are time-optimal.

At any point along the boundary, the time-optimal trajectory ρ may depart tangentially. This continuation of the time-optimal path satisfies Bellman's necessary principle for optimality. Furthermore, it is the only path that does so. Suppose that at some point $\zeta(s^*) \in N$, $s^* > s$, the departing trajectory is not tangential. Then it either transversely crosses ζ, which is not feasible, or it exits back into the flow domain X. Let μ be this non-tangential line. Since ζ is locally concave in N, for some small $\epsilon > 0$, there is a straight line path from an earlier point along the boundary, $\zeta(s^* - \epsilon)$ that necessarily intersects this non-tangential departing line, see Figure 5.14. By Lemma 35, the tangentially departing path results in a faster arrival time to the intersection point. □

Figure 5.14: Cartoon for the proof of 36. At left, an example of a non-simple (time-optimal) path around an obstacle. At right, we provide a counterexample showing that a tangentially departing path is optimal.

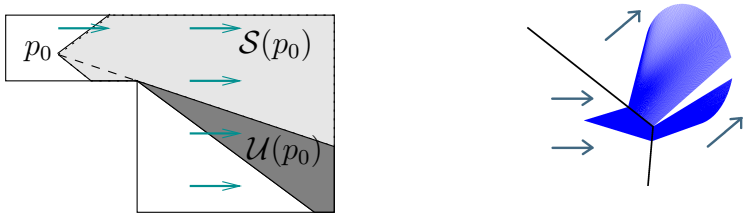


Figure 5.15: Illustrations of scenarios where the simply-reachable set is not equal to the entire reachable set. At left, the simply-reachable set $\mathcal{S}(p_0)$ is denoted by the light gray region, and the simply-unreachable set is denoted by the dark gray region $\mathcal{U}(p_0)$. Optimal trajectories to reach points in $\mathcal{U}(p_0)$ start from p_0 and follow the black dashed line to $\partial\mathcal{U}(p_0)$, then are straight line paths from the corner to any point inside $\mathcal{U}(p_0)$. At right, the solid region denotes $\mathcal{S}(p_0)$. Optimal trajectories to reach points inside $\mathcal{U}(p_0)$ must pass through the corner of the flow interface.

Remark 37. Cases where ∂X is not smooth can be seen as a limiting case of Theorem 36. Similarly, nonsmooth points along $\partial X_k \setminus \partial X$ can be seen as a limiting case of Proposition 32. Both scenarios are illustrated in Figure 5.15. •

Definition 14. *We define a* course change *to be either a discontinuous change in heading as described in* (5.11) *or the sequence of navigating around an obstacle described in Theorem 36.*

5.4.2 Intersecting trajectories

It could happen that multiple simple or non-simple trajectories intersect. This may occur in two possible ways. The first instance involves trajectories that intersect at the same time, and the second instance involves trajectories that intersect at different times. It is important to note that in both cases, the individual trajectories are optimal up until the point where they intersect. Then, a decision must be made to determine which trajectory (if any) to continue based on travel time.

Such scenarios highlight the fact that the Pontryagin minimum principle along with the Bellman principle of optimality are necessary conditions. Thus, while a particular trajectory $\rho_1(t)$ originating from p_0 and passing through $x \in X$ may satisfy Proposition 32 and Theorem 36, it does not imply that $\rho_1(t)$ is the

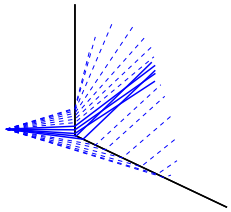


Figure 5.16: An illustration of a scenario where simple trajectories (solid paths) intersect each other.

only trajectory satisfying those results that passes through x.

Proposition 38. *Let $\rho_1(t)$ and $\rho_2(t)$ be two distinct trajectories originating from p_0. Furthermore, let $\rho_1(t)$ and $\rho_2(t)$ satisfy Proposition 32 and Theorem 36. Suppose that there exist $t_1 > 0$ and $t_2 > 0$ such that $\rho_1(t_1) = \rho_2(t_2)$. Then,*

1. *if (without loss of generality) $t_1 < t_2$, then $\rho_1(t)$ can be considered as the optimal trajectory to reach points in $\rho_1(t)$ for all $t > 0$, while $\rho_2(t)$ is restricted to be optimal for points in $\rho_2(t)$ reachable for $t \in [0, t_2)$.*

2. *if $t_1 = t_2$, then the optimality of both trajectories is restricted to only points in $\rho_1(t)$, $\rho_2(t)$ for $t \in [0, t_1]$.*

Proof. Suppose that ρ_1 and ρ_2 do not intersect any other trajectory originating from p_0. Then, since ρ_1 and ρ_2 are the only trajectories satisfying the necessary conditions for optimality, they must be optimal at least up to times t_1 and t_2, respectively.

Consider the case $\rho_1(t_1) = \rho_2(t_2) \triangleq x$ and $t_1 < t_2$. Clearly ρ_2 is not the time-optimal path to reach x. For ρ_2, optimality is only true for points $\rho(t)$ with $0 < t < t_2$, while the optimality of ρ_1 can be used to optimally reach any point in $\rho_1(t)$, for any t.

Suppose instead that $\rho_1(t_1) = \rho_2(t_2) \triangleq x$, with $t_1 = t_2$ and x inside the interior of a region X_k. Suppose that ρ_1 continues to be the time-optimal path to go from p_0 to $y = \rho_1(t)$ for $t > t_1$. Then the trajectory composed of $\rho_2(t)$ until x and $\rho_1(t)$ until y would also be time-optimal (it takes the same time to reach y

from p_0). This is a contradiction with the fact that inside the region X_k, optimal trajectories must be straight lines. □

The first case of Proposition 38 involves simple trajectories that intersect at different times. This can occur with flow around an obstacle as in Figure 5.17. It may be that flow on one side of the obstacle is faster or shorter. Then it is possible for a trajectory that takes the faster path to intersect a trajectory that travels the slower path. Intersecting trajectories can also occur when an agent travels along a flow interface as in Figure 5.18. Upon crossing a flow interface, the new course change may result in an agent that travels along the interface. Then, some trajectories are blocked by the trajectory flowing along the interface.

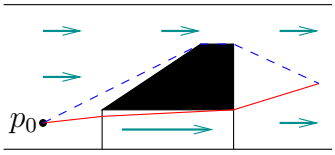


Figure 5.17: Example of one path flowing around an obstacle faster than the other. The dashed path must terminate upon intersecting the solid path.

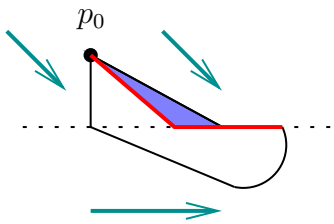


Figure 5.18: Example of trajectories terminating at a flow interface. Any simple trajectory starting from p_0 traversing through the shaded region must terminate at the flow interface (dotted line). There is one simple trajectory (thick line) that flows along the interface after switching into the lower region. The T-reachable set $\mathcal{R}_T(p_0)$ is outlined.

The second case of Proposition 38 introduces the case of simple trajectories intersecting at the same time. This may occur when there is a flow interface that is convex relative to the incoming flow as in Figure 5.16.

5.4.3 Trajectories along flow interfaces

The flow case shown in Figure 5.18 motivates additional study of trajectories that flow along the interface of two constant regions. Application of (5.11) can result in a trajectory that travels along the boundary between two different flows. Use of the same result can also give a way to compute a heading back into the first region. This leads to trajectories as in Figure 5.19.

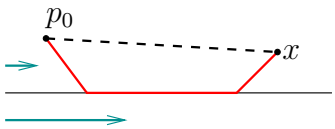


Figure 5.19: Trajectories along flow interfaces. In certain scenarios, it may be faster to follow the solid line path to reach x rather than the simple trajectory (dotted line). This is due to the faster flow velocity in the bottom region.

We now show that the existence of an initial heading θ_- that results in an outgoing trajectory that is tangent with the flow interface. For simplicity, suppose we have a two-region flow with parameters:

$$\alpha_1 = \alpha_2 = 0; \quad 1 < \|V_-\| < \|V_+\|; \quad \xi = \frac{\pi}{2}.$$

This corresponds to the cartoon shown in Figure 5.19. We seek a solution to θ_- in (5.11) that results in $\theta_+ = 0$. This implies that the trajectory after switching moves along the flow interface.

$$\begin{aligned} \frac{1 + \|V_-\| \cos\theta_-}{-\cos\theta_-} &= \frac{1 + \|V_+\|}{-1} \\ 1 + \|V_-\| \cos\theta_- &= \cos\theta_- + \|V_+\| \cos\theta_- \\ \cos\theta_- &= \frac{1}{1 + \|V_+\| - \|V_-\|}. \end{aligned}$$

Note here that the assumption that $\|V_+\| > \|V_-\|$ is important in order to have a well-defined solution.

Now we wish to solve the subsequent problem. Once we are flowing along the boundary in the faster flow region, what is the optimal heading change back

into the slower region?

$$\alpha_1 = \alpha_2 = 0; \quad 1 < \|V_+\| < \|V_-\|; \quad \xi = \frac{\pi}{2}\,.$$

We set $\theta_- = 0$ and solve for θ_+.

$$\begin{aligned}
\frac{1 + \|V_-\|}{-1} &= \frac{1 + \|V_+\| \cos\theta_+}{-\cos\theta_+} \\
\cos\theta_+ + \|V_-\| \cos\theta_+ &= 1 + \|V_+\| \cos\theta_+ \\
\cos\theta_+ &= \frac{1}{1 + \|V_-\| - \|V_+\|}\,.
\end{aligned}$$

Again, since $\|V_-\| > \|V_+\|$ the solution for θ_+ is well-defined.

If instead we had $\|V_-\| > \|V_+\|$, then flow along the boundary results only if $1 + \|V_+\| - \|V_-\| \leq -1$. This implies that $\|V_-\| \geq \|V_+\| + 2$. We also know that the feasible range of optimal initial headings is $\theta_- \in [-\beta_1 - \frac{\pi}{2}, \beta_1 + \frac{\pi}{2}]$ where $\beta_1 = \arcsin(\frac{1}{\|V_-\|})$. Letting $\theta_- = \beta_1 + \frac{\pi}{2}$ and taking cos of both sides gives:

$$\begin{aligned}
\cos\left[\arcsin\left(\frac{1}{\|V_-\|}\right) + \frac{\pi}{2}\right] &= \cos\left(\arcsin\frac{1}{\|V_-\|}\right)\cos\frac{\pi}{2} - \sin\left(\arcsin\frac{1}{\|V_-\|}\right)\sin\frac{\pi}{2} \\
&= -\frac{1}{\|V_-\|}\,.
\end{aligned}$$

The same quantity results if we consider $\theta_1 = -\beta_1 - \frac{\pi}{2}$. Thus far, we have:

$$\cos\theta_- = \frac{1}{1 + \|V_+\| - \|V_-\|}, \quad \cos(\beta_1 + \frac{\pi}{2}) = -\frac{1}{\|V_-\|}\,.$$

Since $\|V_-\| > \|V_+\|$, $1 + \|V_+\| - \|V_-\| \leq -1$, and $\|V_-\|, \|V_+\| > 1$, we can say that

$$\frac{1}{1 + \|V_+\| - \|V_-\|} < -\frac{1}{\|V_-\|}\,.$$

This implies that $\cos\theta_- < \cos\left(\beta_1 + \frac{\pi}{2}\right)$ since for the example we are considering, θ_- and $\beta_1 + \frac{\pi}{2}$ are in the second or third quadrants. This implies that for any choice of $\|V_-\| > \|V_+\|$, the resulting initial heading that gives $\theta_+ = 0$ is not a feasible initial optimal heading, and time-optimal motion along the boundary after switching is not possible.

Although we have considered a simple example for how a trajectory can flow along a boundary, this may happen for other choices of α_1, α_2. When an

agent is moving along a flow boundary, and it is possible to switch back into the first region, the agent may choose to switch back at any time, making this process indeterminate. However, the result above dictates that there is only one possible outgoing heading back into the first flow region. Furthermore, it is possible for these trajectories that flow along boundaries and later return to intersect other trajectories that remained in the original flow region. See Figure 5.20 for an example of this. The treatment of these intersecting trajectories has been discussed in the previous subsection.

For completeness, the following result summarizes the above analysis for general flow parameters.

Proposition 39. *Assume two flow regions defined by the parameters* $\|V_-\|, \alpha_1$ *and* $\|V_+\|, \alpha_2$*, respectively, separated by an interface whose normal angle is* ξ*. If it is possible for an agent to flow along the boundary under the second flow, then* θ_+ *satisfies*

$$\theta_+ \in \left\{ \xi \pm \arccos\left[-\|V_+\| \sin(\alpha_2 + \xi)\right] , -\xi \pm \arccos\left[\|V_+\| \sin(\alpha_2 + \xi)\right] \right\}. \quad (5.13)$$

Let

$$D = \frac{1 + \|V_+\| \cos(\theta_+ - \alpha_2)}{\sin(\theta_+ - \xi)} .$$

Then, the incoming heading resulting in flow along the boundary, if it exists, satisfies

$$\theta_- = \arctan\left[\frac{\|V_-\| \sin\alpha_1 - D\cos\xi}{\|V_-\| \cos\alpha_1 + D\sin\xi}\right] \pm \arccos\left(\frac{-1}{\sqrt{(\|V_-\| \sin\alpha_1 - D\cos\xi)^2 + (D\sin\xi + \|V_-\| \cos\alpha_1)^2}}\right) . \quad (5.14)$$

Proof. We begin by stating the useful identity

$$a\cos\theta + b\sin\theta = \sqrt{a^2 + b^2}\cos(\theta - \phi) ,$$
$$\phi = \arctan\left(\frac{b}{a}\right) .$$

Given that the outgoing trajectory flows along the flow interface, we have the following condition

$$\tan\left(\xi \pm \frac{\pi}{2}\right) = \frac{\sin\left(\xi \pm \frac{\pi}{2}\right)}{\cos\left(\xi \pm \frac{\pi}{2}\right)} = \frac{\sin\theta_+ + \|V_+\| \sin\alpha_2}{\cos\theta_+ + \|V_+\| \cos\alpha_2} .$$

The left hand side denotes the tangent direction at the interface crossing while the right hand side denotes the resultant velocity of an agent. We now solve for the outgoing vehicle heading θ_+. Throughout we note that we keep track of the plus/minus signs so that they remain consistent.

$$\begin{aligned}
\frac{\pm\cos\xi}{\mp\sin\xi} &= \frac{\sin\theta_+ + \|V_+\|\sin\alpha_2}{\cos\theta_+ + \|V_+\|\cos\alpha_2} \\
\pm\cos\xi\cos\theta_+ \pm \|V_+\|\cos\alpha_2\cos\xi &= \mp\sin\xi\sin\theta_+ \mp \|V_2\|\sin\alpha_2\sin\xi \\
\cos\xi\cos\theta_+ \pm \sin\xi\sin\theta_+ &= \mp\|V_+\|\sin(\alpha_2+\xi)
\end{aligned}$$

At this point, we apply the trigonometric identity for addition of sin and cos.

$$\begin{aligned}
\cos\left[\theta_+ - \arctan\left(\frac{\pm\sin\xi}{\pm\cos\xi}\right)\right] &= \mp\|V_+\|\sin(\alpha_2+\xi) \\
\cos(\theta_+ \mp \xi) &= \mp\|V_2\|\sin(\alpha_2+\xi)
\end{aligned}$$

At this point we take the inverse cosine of both sides. However, due to the multiplicity of solutions for arccos, we will again have two possible solutions. Before proceeding with arccos, we split the two existing solutions we already have to obtain–in the end–four total possible solutions.

$$\begin{cases}
\theta_+ - \xi = \pm\arccos[-\|V_+\|\sin(\alpha_2+\xi)]\,, \\
\theta_+ + \xi = \pm\arccos[\|V_+\|\sin(\alpha_2+\xi)]\,,
\end{cases}$$

$$\begin{cases}
\theta_+ = \xi \pm \arccos[-\|V_+\|\sin(\alpha_2+\xi)]\,, \\
\theta_+ = xi \pm \arccos[\|V_+\|\sin(\alpha_2+\xi)]\,.
\end{cases}$$

Thus, we have achieved the first result of this proposition.

Now we assume that θ_+ is known and satisfies the first result. We wish to solve for the incoming heading θ_- that achieves an outgoing heading of θ_+. To simplify some clutter, we let

$$D = \frac{1 + \|V_+\|\cos(\theta_+ - \alpha_2)}{\sin(\theta_+ - \xi)}\,.$$

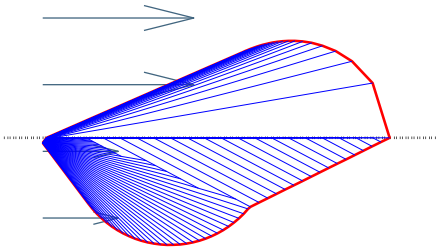


Figure 5.20: Example of trajectories that move along a flow interface and return into the original flow region and its effect on the reachable set of an agent. Notice that some of the returning trajectories intersect simple trajectories and result in termination based on travel time.

Now we solve for θ_-:

$$\frac{1 + \|V_-\| \cos(\theta_- - \alpha_1)}{\sin(\theta_- + \xi)} = D$$

$$1 + \|V_-\|(\cos\theta_- \cos\alpha_1 + \sin\theta_- \sin\alpha_1) = D(\sin\theta_- \cos\xi - \cos\theta_- \sin\xi)$$

$$(\|V_-\cos\alpha_1 + D\sin\xi)\cos\theta_- + (\|V_-\|\sin\alpha_1 - D\cos\xi)\sin\theta_- = -1$$

We apply the trigonometric identity for addition of sin and cos.

$$\sqrt{(\|V_-\|\cos\alpha_1 + D\sin\xi)^2 + (\|V_-\|\sin\alpha_1 - D\cos\xi)^2}$$
$$\cdot \cos\left(\theta_- - \arctan\left[\frac{\|V_-\|\sin\alpha_1 - D\cos\xi}{\|V_-\|\cos\alpha_1 + D\sin\xi}\right]\right) = -1\,.$$

Applying arccos with multiplicity of solutions in mind and solving for θ_- gives the final result of this proposition. □

5.4.4 Flow along a general boundary

In the previous subsection, we demonstrated the existence of optimal paths that flow along the interface between different flow regions. We motivate the following discussion with Figure 5.21, where a slower flow is surrounded by a fast flow. In this scenario, it may be faster to avoid entering the slower flow region in order to reach points on the other side. In this case, the slower region may

be treated as an obstacle in the flow, and a candidate trajectory may be formed using previous techniques. Then, apply the result of Proposition 38 to resolve the conflict of multiple trajectories meeting at the same point.

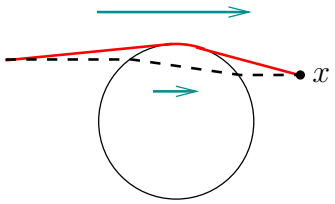


Figure 5.21: Flow along a general boundary. An example situation where it may be faster to follow the solid path than the dotted path. This is because the flow is much faster outside of the circular region.

5.4.5 Nested non-simple trajectories

In the previous subsections, we described scenarios where either the reachable set is larger than the set of all simple trajectories or where simple trajectories intersect. In the case where there is a set $\mathcal{U}(p_0)$ that is unreachable by simple trajectories as in Figure 5.15, it is possible to have nested non-simply reachable sets. This occurs as in Figure 5.22. For such scenarios, self-similarity provides a solution for an optimal path from p_0 to a point within a nested non-simply reachable set.

For example, in Figure 5.22, we can view the problem of reaching a point in the region $\mathcal{U}_2(p_0)$ in minimum time as a minimum time problem starting from the point x_1^*. By repeatedly applying the result from the previous subsection an optimal path from p_0 to a point in $\mathcal{U}_2(p_0)$ can be obtained. By the principle of optimality, this path is correct since all subpaths are optimal trajectories.

5.5 Area coverage

In this section we present coverage algorithms that aim to maximize the area coverage metric of (5.4). The algorithms are distributed over the reachable set graph $\mathcal{G}_{\text{reach}}$.

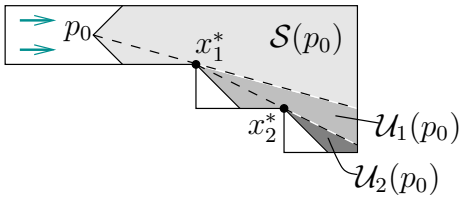


Figure 5.22: A scenario where there is a nested non-simply-reachable set $\mathcal{U}_2(p_0)$ (dark gray) that cannot be reached by a simple trajectory from the corner marked x_1^*. The simply-reachable set is denoted by the light gray region, and the non-simply-reachable set $\mathcal{U}_1(p_0)$ is shown as the medium-gray region. Together, $\mathcal{U}(p_0) = \mathcal{U}_1(p_0) \cup \mathcal{U}_2(p_0)$.

5.5.1 Gradient of the area objective function

Since the goal is to maximize $\mathcal{H}$, we begin by taking the gradient of $\mathcal{H}$ with respect to $p^{(i)}$ in order to obtain a set directions each agent must travel in.

Proposition 40. *Given the area objective* (5.4)*, let*

$$\mathcal{A}^{(i)} = \partial\mathcal{R}_T(p^{(i)}) \cap \left(\bigcup_{j \in \mathcal{N}_{flow}^{(i)}} \mathcal{R}_T(p^{(j)}) \right)^c \cap X \,, \tag{5.15}$$

the set of points in $\partial\mathcal{R}_T(p^{(i)})$ not in the interior of neighboring reachable sets. Then the gradient with respect to $p^{(i)}$ is:

$$\frac{\partial \mathcal{H}}{\partial p^{(i)}} = \int_{\mathcal{A}^{(i)}} \hat{\mathbf{n}}_{out}^{\mathsf{T}}(\zeta^{(i)}) \frac{\partial \zeta^{(i)}}{\partial p^{(i)}} d\zeta^{(i)} \,, \tag{5.16}$$

where $\zeta^{(i)} : \mathbb{S} \to \mathbb{R}^2$ is a parametrization of $\partial\mathcal{R}_T(p^{(i)})$, and $\hat{\mathbf{n}}_{out} \colon \mathbb{R}^2 \to \mathbb{R}^2$ is the unit outward-pointing normal vector at $\zeta^{(i)}$.

Proof. We utilize a conservation of mass result found in the fluid mechanics literature, and detailed also in [15, 24], to compute this derivative. According to [15], the integrand, 1, must be continuous, and the region $\bigcup \mathcal{R}_T(p^{(i)})$ must be a union of star-shaped sets. Both these criteria are satisfied, so the derivative of $\mathcal{H}$ with respect to $p^{(i)}$ is

$$\frac{\partial \mathcal{H}}{\partial p^{(i)}} = \int_{\bigcup \mathcal{R}_T(p^{(i)})} \frac{\partial}{\partial p^{(i)}}[1] dq + \int_{\partial\left(\bigcup \mathcal{R}_T(p^{(i)})\right)} \hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\zeta) \frac{\partial \zeta}{\partial p^{(i)}} d\zeta \,,$$

where ζ is a parametrization of $\partial\left(\bigcup \mathcal{R}_T(p^{(i)})\right)$.

Based on Theorem 30, since there are no holes in $\mathcal{R}_T(p^{(i)})$, the boundary $\partial\left(\bigcup \mathcal{R}_T(p^{(i)})\right)$ is only composed of curves from the following: solutions $\gamma^{(i)}(t,-1)$, $\gamma^{(i)}(t,1)$, $\gamma^{(i)}(T,K)$, and sections of ∂X for $i \in \{1,\ldots,n\}$. Note that any point along those solutions is only a function of the initial position $p^{(i)}$, and not any other agent's initial position. For this reason, we conclude that $\frac{\partial \zeta}{\partial p^{(i)}} = 0$ if $\zeta \notin \mathcal{R}_T(p^{(i)})$.

Let $\mathcal{A}^{(i)} = \partial\mathcal{R}_T(p^{(i)}) \cap \partial\left(\bigcup_{j=1}^N \mathcal{R}_T(p^{(j)})\right) \cap X$. Then, $\mathcal{A}^{(i)}$ is the set of points along $\partial\mathcal{R}_T(p^{(i)})$ that is not in the interior of $\bigcup_{j=1}^N \mathcal{R}_T(p^{(j)})$. The integral expression above then simplifies to

$$\frac{\partial \mathcal{H}}{\partial p^{(i)}} = \int_{\mathcal{A}^{(i)}} \hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\zeta)\frac{\partial \zeta}{\partial p^{(i)}} d\zeta\,.$$

Instead of using the parametrization for the entire boundary of $\bigcup_{i=1}^N \mathcal{R}_T(p^{(i)})$, we can just use a parametrization for the boundary $\partial\mathcal{R}_T(p^{(i)})$. Let $\zeta^{(i)}$ be one such parametrization. Then, we obtain the result (6.14). □

The general strategy now is to follow the gradient direction in order to maximize coverage area. We begin by examining the time-evolution of the objective function $\mathcal{H}$.

Proposition 41. *The rate of area increase is* locally *maximized if each agent utilizes the control law*

$$u^{(i)} = \frac{\frac{\partial \mathcal{H}}{\partial p^{(i)}}}{\left\|\frac{\partial \mathcal{H}}{\partial p^{(i)}}\right\|}\,. \tag{5.17}$$

Proof. We consider the time evolution of the cost function by applying the chain rule:

$$\dot{\mathcal{H}} = \sum_{i=1}^N \frac{\partial \mathcal{H}}{\partial p^{(i)}}\dot{p}^{(i)} = \sum_{i=1}^N \frac{\partial \mathcal{H}}{\partial p^{(i)}}\left(u^{(i)} + V\right)\,.$$

Each agent's contribution to the rate of change of $\mathcal{H}$ is then given by $\frac{\partial \mathcal{H}}{\partial p^{(i)}}(u^{(i)}+V) = \frac{\partial \mathcal{H}}{\partial p^{(i)}}u^{(i)} + \frac{\partial \mathcal{H}}{\partial p^{(i)}}V$. However, there is the constraint that $\|u^{(i)}\| \leq 1$. The choice of $u^{(i)}$ that maximizes each individual term of the sum is the vector that points in the same direction as $\frac{\partial \mathcal{H}}{\partial p^{(i)}}$, but with magnitude 1. □

Remark 42. Depending on the direction of the gradient $\frac{\partial \mathcal{H}}{\partial p^{(i)}}$, an individual agent may not be able to maximize its own area covered. This is because $\|u^{(i)}\| < \|V\|$, so it cannot be guaranteed that each quantity in the sum can be positive due to the term $\frac{\partial \mathcal{H}}{\partial p^{(i)}} V$. Nevertheless, we can still choose $u^{(i)}$ such that the quantity $\frac{\partial \mathcal{H}}{\partial p^{(i)}}(u^{(i)} + V)$ is maximized for all i. •

5.5.2 Affine flow cases

In general the gradient is difficult to analytically compute due to the $\frac{\partial \zeta^{(i)}}{\partial p^{(i)}}$ term inside the integral. This term is a 2×2 matrix that represents the variation of a point along $\partial \mathcal{R}_T(p^{(i)})$ as the agent position $p^{(i)}$ varies. An analytical result would require an explicit solution to (5.1) using (5.7). However, there are certain cases where such a solution is possible.

Proposition 43. *If the flow field is of the form* (5.8)*, then*

$$\frac{\partial \zeta^{(i)}}{\partial p^{(i)}} = e^{A\tau(p^{(i)}, \zeta^{(i)})} \,. \tag{5.18}$$

Proof. For this particular flow, the time-optimal trajectories are given by (5.10). Taking the gradient of $p^{(i)}(t)$ with respect to initial position $p^{(i)}(0)$ results in

$$\frac{\partial p^{(i)}(t)}{\partial p^{(i)}(0)} = e^{At} \,.$$

To connect the above gradient back to the problem of finding $\frac{\partial \zeta^{(i)}}{\partial p^{(i)}}$, recall that $\zeta^{(i)}$ is a parametrization of $\partial \mathcal{R}_T(p^{(i)})$. Thus points along $\zeta^{(i)}$ are solutions of the optimal control problem at a particular (fixed) time. The result (5.18) follows immediately. □

5.5.3 Constant flows

We can further analyze the algorithms proposed above for the special case of a single constant flow field. For a constant flow, $A = 0$ in (5.18) and the gradient according to (6.14) becomes

$$\frac{\partial \mathcal{H}}{\partial p^{(i)}} = \int_{\mathcal{A}^{(i)}} \hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\zeta^{(i)}) d\zeta^{(i)} \,. \tag{5.19}$$

The result of (5.19) has an intuitive interpretation. In order to maximize area covered, agents move towards locations that are not occupied by other agents' reachable sets.

Lemma 44. *For a constant flow field, $V = c$, if no regions intersect the boundaries ∂X, then $\mathcal{H}$ is non-decreasing if agents use the control law* (5.17).

Proof. To show this, we will prove that

$$\sum_{i=1}^{N} \int_{\mathcal{A}^{(i)}} \hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\zeta^{(i)}) d\zeta^{(i)} = 0\,,$$

using the divergence theorem in 2 dimensions:

$$\int_{\mathcal{D}} (\nabla \cdot F) dq = \int_{\partial \mathcal{D}} (F \cdot \hat{\mathbf{n}}_{\text{out}}) ds\,,$$

where $\mathcal{D} \subset \mathbb{R}^2$ is compact and $F : \mathcal{D} \to \mathbb{R}^2$ is continuously differentiable. If $F = 1$, then $0 = \int_{\partial \mathcal{D}} \hat{\mathbf{n}}_{\text{out}}\, ds$.

From the definition of $\mathcal{A}^{(i)}$ (5.15), since no regions intersect the flow boundary,

$$\mathcal{A}^{(i)} = \partial \mathcal{R}_T(p^{(i)}) \cap \partial \left(\bigcup_{j=1}^{N} \mathcal{R}_T(p^{(j)}) \right).$$

Then, note that each $\mathcal{A}^{(i)}$ is disjoint except at the isolated points where two regions $\mathcal{R}_T(p^{(i)})$ and $\mathcal{R}_T(p^{(j)})$ meet. Furthermore, $\partial \left(\bigcup_{i=1}^{N} \mathcal{R}_T(p^{(i)}) \right) = \bigcup_{i=1}^{N} \mathcal{A}^{(i)}$.

Since $\bigcup_{i=1}^{N} \mathcal{R}_T(p^{(i)})$ is compact, then the divergence theorem states that: $0 = \int_{\partial \left(\bigcup_{i=1}^{N} \mathcal{R}_T(p^{(i)}) \right)} \hat{\mathbf{n}}_{\text{out}}\, ds = \int_{\partial \left(\bigcup_{i=1}^{N} \mathcal{A}^{(i)} \right)} \hat{\mathbf{n}}_{\text{out}}\, ds = \sum_{i=1}^{N} \int_{\mathcal{A}^{(i)}} \hat{\mathbf{n}}_{\text{out}} ds$. In other words, $\sum_{i=1}^{N} \frac{\partial \mathcal{H}}{\partial p^{(i)}} = 0$.

Revisiting the time evolution of $\mathcal{H}$, we have

$$\dot{\mathcal{H}} = \sum_{i=1}^{N} \frac{\partial \mathcal{H}}{\partial p^{(i)}} (u^{(i)} + c) = \frac{\partial \mathcal{H}}{\partial p^{(i)}} u^{(i)} + \frac{\partial \mathcal{H}}{\partial p^{(i)}} c = \sum_{i=1}^{N} \left\| \frac{\partial \mathcal{H}}{\partial p^{(i)}} \right\| + \frac{\partial \mathcal{H}}{\partial p^{(i)}} c\,.$$

Since c is a constant, the second term in the summation is zero. Thus, $\dot{\mathcal{H}} = \sum_{i=1}^{N} \left\| \frac{\partial \mathcal{H}}{\partial p^{(i)}} \right\| \geq 0$. □

Lemma 45. *For constant flows, if the flow boundaries are parallel with the flow direction and X is unbounded (the flow domain is an infinitely long strip), then $\mathcal{H}$ is maximized by* (5.17).

Proof. Let $u^{(i)} = 0$ for all $i \in \{1, \ldots, n\}$. Then $\dot{\mathcal{H}} = \sum_{i=1}^{N} \frac{\partial \mathcal{H}}{\partial p^{(i)}} c$. Suppose an agent i has a region that intersects the boundary of X. Since $u^{(i)} = 0$, agent i drifts parallel to the shore. Furthermore, other agents drift at the same rate. Thus, all agents maintain their relative positions and so total area remains constant. Thus $\sum_{i=1}^{N} \frac{\partial \mathcal{H}}{\partial p^{(i)}} c = 0$. Now, for $u^{(i)}$ as in (5.17), $\dot{\mathcal{H}} = \sum_{i=1}^{N} \left\| \frac{\partial \mathcal{H}}{\partial p^{(i)}} \right\| \geq 0$. □

5.5.4 Piecewise constant flows

The gradient expression for $\mathcal{H}$ with respect to agent positions $p^{(i)}$ is identical to the result for affine flows in (6.14); the main difference lies in the term $\frac{\partial \zeta^{(i)}}{\partial p^{(i)}}$. First we present a result dealing with simple trajectories, and then modify it to include non-simple trajectories.

We seek a formula for the position of a trajectory at terminal time T with an initial heading θ_0 at $t = 0$, and passing through a total of $q + 1$ flow regions. Thus, there will be q optimal course changes. Once the initial heading θ_0 is chosen, the sequence of optimal headings to follow, $\{\theta_1, \ldots, \theta_q\}$, within each region can be solved.

Points on the boundary $\partial \mathcal{A}^{(i)}$ can be viewed as endpoints of simple trajectories starting at p_0. Suppose that a simple trajectory passes through a total of $q + 1$ flow regions. Then, the final position at time T is given by

$$p_T = \bar{x}_q + (T - \bar{t}_q)\dot{p}_q \,. \tag{5.20}$$

From (5.20) given some arbitrary initial heading, we compute $\frac{\partial \zeta^{(i)}}{\partial p^{(i)}}$ with the following results.

Proposition 46. *Let* $\bar{x}_0 = p_0$ *and*

$$D_\ell = \left(\frac{\partial \bar{x}_\ell}{\partial \bar{x}_{\ell-1}} \right) \left(\frac{\partial \bar{x}_{\ell-1}}{\partial \bar{x}_{\ell-2}} \right) \cdots \left(\frac{\partial \bar{x}_1}{\partial \bar{x}_0} \right) .$$

Then the derivative of (5.20)*, the endpoint of a simple trajectory, with respect to initial position* p_0 *is*

$$\frac{\partial p_T}{\partial p_0} = D_q - \dot{p}_q \sum_{\ell=1}^{q} \frac{\partial \bar{t}_\ell}{\partial \bar{x}_{\ell-1}} D_{\ell-1} \,. \tag{5.21}$$

Proof. We can take the result (5.20) and take the derivative with respect to initial position p_0 by applying the chain rule. Before launching into the computation, we make a couple observations. The sequence of interface crossing positions $\{\bar{x}_1, \ldots, \bar{x}_q\}$ depends only on the initial heading choice θ_0. In fact, each interface crossing position $\bar{x}_\ell$ depends only on the previous interface crossing $\bar{x}_{\ell-1}$ and the (fixed) choice of direction $\theta_{\ell-1}$ determined by (5.11). On the other hand, we can express the interface crossing time $\bar{t}_\ell$ as

$$\bar{t}_\ell = \bar{t}_{\ell-1} + \frac{\|\bar{x}_\ell - \bar{x}_{\ell-1}\|}{\|\dot{p}_{\ell-1}\|}\,.$$

Thus, $\bar{t}_\ell$ depends on both the previous interface crossing time and the previous crossing position.

Applying the chain rule to (5.20), we have

$$\begin{aligned}\frac{\partial p_T}{\partial p_0} &= \frac{\partial \bar{x}_q}{\partial p_0} - \dot{p}_q \frac{\partial \bar{t}_q}{\partial p_0} \\ &= \frac{\partial \bar{x}_q}{\partial \bar{x}_{q-1}}\frac{\partial \bar{x}_{q-1}}{\partial p_0} - \dot{p}_q \left[\frac{\partial \bar{t}_q}{\partial \bar{x}_{q-1}}\frac{\partial \bar{x}_{q-1}}{\partial p_0} + \frac{\partial \bar{t}_q}{\partial \bar{t}_{q-1}}\frac{\partial \bar{t}_{q-1}}{\partial p_0}\right].\end{aligned}$$

Note that

$$\frac{\partial \bar{x}_q}{\partial \bar{x}_0} = \left(\frac{\partial \bar{x}_q}{\partial \bar{x}_{q-1}}\right)\left(\frac{\partial \bar{x}_{q-1}}{\partial \bar{x}_{q-2}}\right)\cdots\left(\frac{\partial \bar{x}_1}{\partial \bar{x}_0}\right) \triangleq D_q\,.$$

Then,

$$\frac{\partial p_T}{\partial p_0} = D_q - \dot{p}_q\left[\frac{\partial \bar{t}_q}{\partial \bar{x}_{q-1}} D_{q-1} + \frac{\partial \bar{t}_{q-1}}{\partial p_0}\right].$$

Repeated application of the chain rule on the last term gives the result. □

We now detail how to compute each term in (5.21).

Proposition 47. *Let an interface crossing position and time be $\bar{x}$ and $\bar{t}$, and suppose that there is only one such crossing and that the interface is differentiable at $\bar{x}$. Let $\eta = (\cos\xi, \sin\xi)^\mathsf{T}$ be the unit normal vector to the interface at $\bar{x}$. Then, for a fixed choice of heading $\dot{p}$, the Jacobian of the crossing position with respect to change in initial position is given by*

$$\frac{\partial \bar{x}}{\partial p_0} = I_2 - \frac{\left(\dot{p}\eta^\mathsf{T}\right)}{\dot{p}^\mathsf{T}\eta}\,, \tag{5.22}$$

where I_2 is the identity matrix of dimension 2. The Jacobian of the crossing time with respect to change in initial position is given by

$$\frac{\partial \bar{t}}{\partial p_0} = -\frac{\eta^{\mathsf{T}}}{\dot{p}^{\mathsf{T}}\eta}\,. \tag{5.23}$$

Proof. Suppose there is a parametrization $\zeta\colon \mathbb{R} \to \mathbb{R}^2$ of the curve described by $\psi(x) = 0$. Let $\zeta(0)$ correspond to the interface crossing position $\bar{x}$. We wish to find the variation of the crossing position $\bar{x}$ with respect to a change in initial position p_0. For example, in Figure 5.23, for a horizontal displacement δx, we wish to find

$$\lim_{\|\delta x\|\to 0} \frac{\zeta(\tilde{t}) - \zeta(0)}{\|\delta x\|}\,, \tag{5.24}$$

where $\zeta(\tilde{t})$ is the point along the curve ζ that intersects the displaced trajectory. We also wish to find a similar quantity for vertical displacements δy.

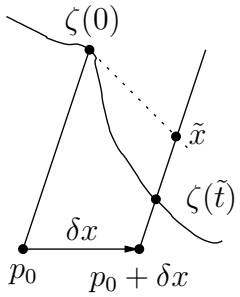


Figure 5.23: Diagram for the proof of Proposition 47.

Since $\bar{x}$ is constrained to lie on the curve $\zeta(t)$, the derivatives of $\bar{x}$ with respect to the horizontal and vertical variations in initial position must lie along the tangent direction $\zeta'(0)$. One can show that computing

$$\lim_{\|\delta x\|\to 0} \frac{\tilde{x} - \zeta(0)}{\|\delta x\|}\,, \tag{5.25}$$

where $\tilde{x}$ is the point along the tangent line from $\zeta(0)$ that intersects the displaced trajectory, is equivalent to computing (5.24) as $\|\delta x\| \to 0$. The limit (5.25) is much easier to compute since $\tilde{x}$ can be solved for explicitly.

In other words, we use the linear approximation of $\zeta(t)$ about $\zeta(0) = \bar{x}$ to compute the Jacobian $\frac{\partial \bar{x}}{\partial p_0}$ at the point $\bar{x}$. Let η be the unit normal vector to ζ at

$\zeta(0)$. Then, given a constant velocity $\dot{p}$, the intersection point $\tilde{x}$ of a trajectory starting from p_0 to the tangent line through $\zeta(0)$ is given by

$$\tilde{x} = p_0 + \frac{(\zeta(0) - p_0)^\mathsf{T}\eta}{\dot{p}^\mathsf{T}\eta}\dot{p}\,.$$

The quotient of inner products represents the travel time from p_0 to the tangent line, since the inner products project the travel distance and travel velocity onto the normal direction. Expanding p_0, $\dot{p}$, and η into components and differentiating with respect to horizontal and vertical changes of p_0 yields the final result of (5.22). In performing the above differentiation, we also differentiate the crossing time with respect to initial position to get (5.23). □

For non-simple trajectories, the expression for the derivative of the endpoint with respect to the initial position still follows from (5.21) by treating non-simple course changes as one of the $\bar{x}_\ell$ in (5.21). However, the expressions (5.22) and (5.23) do not apply for course changes around obstacles or through flow interfaces that are not differentiable. We present the following result to address these cases.

Proposition 48. *Let p_T be the endpoint of a non-simple trajectory that undergoes q course changes. Let $\ell^* \in \{1, \ldots, q\}$ denote the* first *non-simple course change described in Theorem 36 or the Remark 37. Then, the Jacobian of the endpoint position with respect to initial position is*

$$\frac{\partial p_T}{\partial p_0} = -\dot{p}_q \sum_{\ell=1}^{\ell^*} \frac{\partial \bar{t}_\ell}{\partial \bar{x}_{\ell-1}} D_{\ell-1}\,, \tag{5.26}$$

with

$$\frac{\partial \bar{t}_{\ell^*}}{\partial \bar{x}_{\ell^*-1}} = \frac{1}{\|\dot{p}_{\ell^*-1}\|} \frac{(\bar{x}_{\ell^*-1} - \bar{x}_{\ell^*})^\mathsf{T}}{\|\bar{x}_{\ell^*} - \bar{x}_{\ell^*-1}\|} \tag{5.27}$$

Proof. We assume a non-simple trajectory such as that described in Theorem 36. Then, there is a specific point $x^* = \bar{x}_{\ell^*}$ that a trajectory originating from p_0 must pass through to reach points in $\mathcal{U}(p_0)$. This point is *fixed* in space, so the quantity $\frac{\partial \bar{x}_{\ell^*}}{\partial p_0} = 0$. Thus, the expression for the Jacobian (5.21) becomes (5.26).

The remaining task is to compute the special term $\frac{\partial \bar{t}_{\ell^*}}{\partial \bar{x}_{\ell^*-1}}$ corresponding to the special crossing point $\bar{x}_{\ell^*}$. For $\ell < \ell^*$, the derivative is given by (5.23). We

begin with

$$\bar{t}_{\ell^*} = \bar{t}_{\ell^*-1} + \frac{\|\bar{x}_{\ell^*} - \bar{x}_{\ell^*-1}\|}{\|\dot{p}_{\ell^*-1}\|}.$$

Then, differentiating with respect to the previous crossing location $\bar{x}_{\ell^*-1}$, we obtain the result (5.27). □

Remark 49. The only impediment of considering piecewise affine flows as opposed to piecewise continuous flows comes from the necessity of analytically computing the term $\frac{\partial \bar{t}}{\partial \bar{x}}$. An expression for $\bar{t}_\ell$ in terms of $\bar{t}_{\ell-1}$, $\bar{x}_\ell$, and $\bar{x}_{\ell-1}$ as in the proofs of Propositions 46 and 48 cannot be easily stated. •

5.6 Simulations

The following simulation portrays a typical execution of the area coverage algorithm for the set of special flow cases described in Sections 5.2-5.2.3. Here, 8 agents deploy in a flow environment of length $100m$ and width $50m$. We demonstrate a case with flow parameters $a_{11} = 0.025s^{-1}$, $b_1 = 1m/s$, $a_{22} = -0.1s^{-1}$, and $b_2 = 0$. The simulation results are shown in Figure 5.24.

Next, we present a simulation result for a piecewise constant flow scenario. The flow environment that we choose features a slower, wide flow that becomes narrower and faster. The initial flow velocity has is $1.75m/s$ to the right, and the final flow velocity is $3.5m/s$ to the right. We wish to maximize area covered using 8 agents with maximum speed of $1m/s$ and a limited-time range of $2s$.

Figure 5.25 displays snapshots of the agents as they deploy and flow across the environment. We note that the algorithm does not monotonically maximize area coverage due to the varying flow environment. Figure 5.26 displays the objective function (5.4) for the simulation. The gradient ascent achieves maximization of coverage area when all agents are in the same flow region. This corresponds to $t = 21s$ in the simulation, as all agents are in the final flow region. In other words, the agents minimize the rate of area loss.

In the next simulation, we demonstrate a piecewise constant flow domain with an obstacle. The flow environment features three parallel strips with the middle strip having a faster flow. Additionally, the obstacle is located in the middle

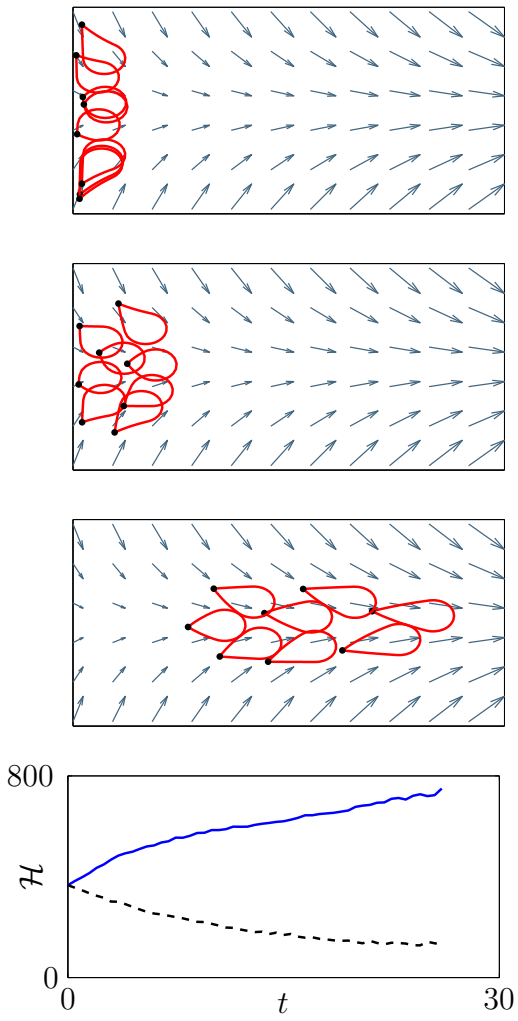


Figure 5.24: Different snapshots of the area deployment algorithm in an affine flow for 8 agents. Times for the three snapshots are first $t = 0s$, second $t = 5s$, and third $t = 26s$. Afterward, the agents leave the flow environment. A plot of the objective function is shown in the bottom graph (solid line) and a plot of the objective function for $u^{(i)} = 0$ for all i is shown in the dashed line for comparison.

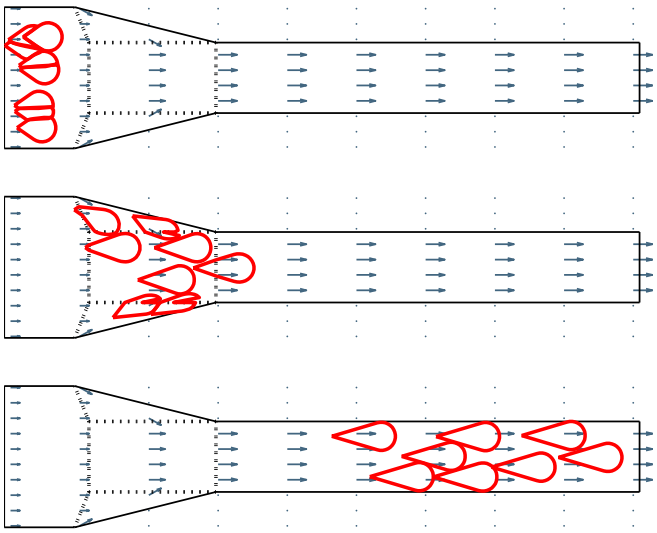


Figure 5.25: Deployment in a piecewise constant flow environment by 8 agents. Simulation times are $t = 0s$ (top), $t = 10s$ (middle), and $t = 20s$ (bottom).

flow region. Figure 5.27 gives three four snapshots of the coverage maximization as time progresses. The agents successfully navigate around the obstacle and move away from each other in order to maximize area covered. Figure 5.28 provides a plot of the total area covered as a function of time.

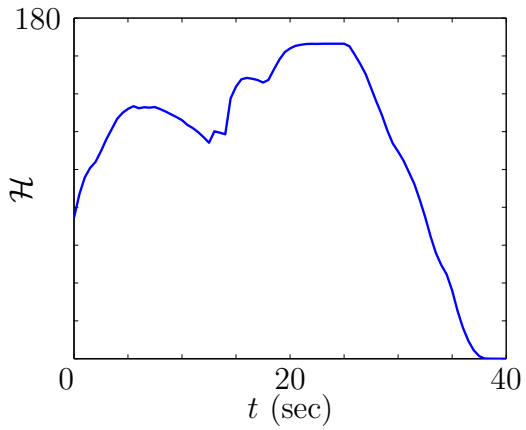


Figure 5.26: Coverage area plot of the simulation in Figure 5.25. Agents begin to enter the final flow region after $t = 10s$. Then, once all agents are in the final flow region, the area is maximized after $t = 20s$.

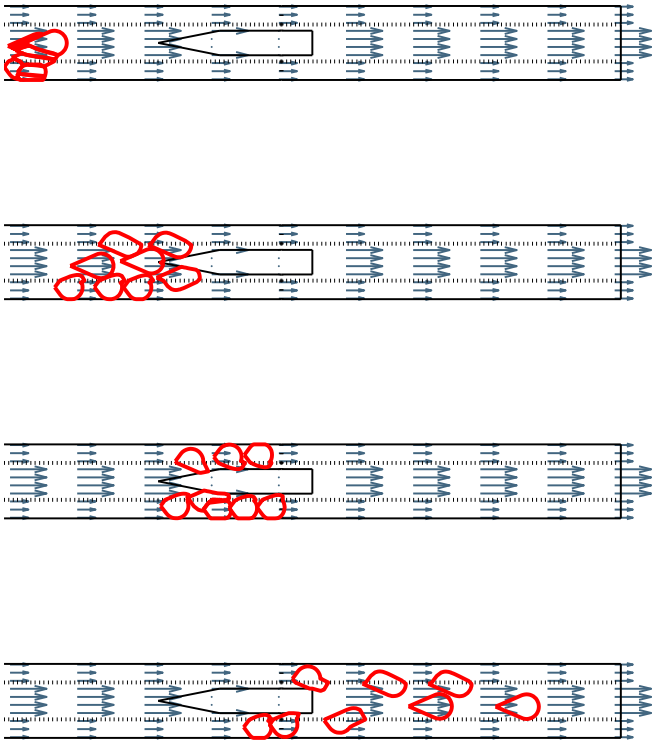


Figure 5.27: Deployment in a flow environment with an island obstacle by 8 agents. The flow is also faster in the middle strip than in the regions close to the top and bottom boundaries. Simulation times from top to bottom are: $t = 0s$, $t = 10s$, $t = 20s$, and $t = 40s$.

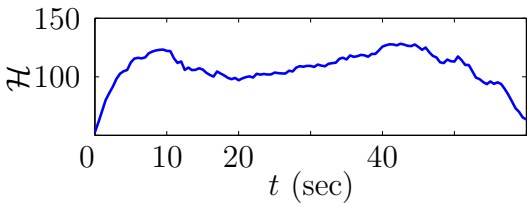


Figure 5.28: Coverage area plot of the simulation in Figure 5.27.

Chapter 6

Distributed Deterministic Annealing

Most current methods for deployment, i.e. [15, 29], rely on gradient techniques to converge to an extremum of a cost function that is generally non-convex. As a result, the final value of the cost function may not be the globally optimal one. Many annealing techniques exist to find a better final value of a cost function. Of these techniques, there are simulated annealing (SA) algorithms [35], as well as a more recent development, deterministic annealing (DA) [68]. Unfortunately, these are centralized algorithms requiring global knowledge of the total state of the system.

Annealing algorithms differ from standard gradient algorithms through the addition of a temperature state. The goal, as in physical annealing, is to gradually lower this temperature, so that the internal configuration of the system is always at or near the lowest energy state. The SA and DA techniques feature *phase changes* as the temperatures are lowered past certain critical values, and we quantify these transitions for the distributed algorithm version.

A closely related work is that of Sharma et. al. [77]. The resulting algorithm discards information of other agents and resources that are far from a given agent. However, the algorithm still requires knowledge of all agents involved in the optimization to determine the information to discard.

In [89], SA was used to solve the clustering and formation control prob-

lems. That work also considered limited-range interactions, however, punctual long-range communication between agents was required. A cell decomposition of the environment had to be done a priori.

In this chapter, we extend the DA algorithm of [68]. Here, we take that discrete DA algorithm to make it continuous in both space and time as well as spatially distributed. We strictly enforce that an individual agent can only sense the presence of other agents within a fixed radius. To do so, we introduce a spatial partition of the environment, and use this to develop a distributed local check of phase changes. Additionally, we introduce a task assignment algorithm to reassign vehicles according to phase changes. With the limited-range constraint, we achieve very similar results as in [68, 77]. Additionally, as this sensing radius increases, the algorithm recovers the original DA algorithm.

The chapter is organized as follows. In section 6.1, we introduce the limited range coverage problem, as well as provide an overview of the DA algorithm. In section 6.2 we derive the gradient direction for a limited-range DA algorithm, and continue in Section 6.3 to provide a sufficient condition to distributively check for phase changes. We merge the two results in Section 6.4 by describing an algorithm for a network of autonomous agents to implement that includes a task allocation subroutine. We provide a simulation in Section 6.5 as a proof of concept.

6.1 Problem formulation

Let N be the total number of agents performing the proposed limited-range distributed DA algorithm. There will be $n \leq N$ formations of agents, with leaders at $p^{(1)}, \ldots, p^{(n)}$ and $p^{(i)} \in Q$ for all i, that split during phase changes. The algorithm finishes with N formations of single vehicles at positions $p^{(1)}, \ldots, p^{(N)}$. All agents have a limited coverage sensing radius $R^{(i)}$, and they have a communication range limited to $2 \max_i R^{(i)}$.

As in [15], we choose to analyze the distributed DA coverage problem via general metrics $f^{(i)} \colon \mathbb{R}_{\geq 0} \to \mathbb{R}$ such that $f^{(i)}$ is Lipschitz and non-decreasing. Let $0 = R_0^{(i)} < R_1^{(i)} < \cdots < R_{m^{(i)}}^{(i)} = R^{(i)}$ be a finite sequence of radii. We assume that

each $f^{(i)}$ is of the form

$$f^{(i)}(x) = \sum_{\alpha=1}^{m^{(i)}} f_\alpha^{(i)}(x) 1_{[R_{\alpha-1}^{(i)}, R_\alpha^{(i)})}\,, \tag{6.1}$$

such that each $f_\alpha^{(i)}$ is differentiable and non-decreasing over $[R_{\alpha-1}^{(i)}, R_\alpha^{(i)})$. Additionally, we have for all α, $f_\alpha^{(i)}(R_\alpha^{(i)}) = f_{\alpha+1}^{(i)}(R_\alpha^{(i)})$, which enforces continuity of $f^{(i)}$.

In what follows we will consider the limited-range heterogeneous analogues of the area-maximizing and centroidal sensing metrics found in [15]. The sensing functions, denoted with the superscript a and m respectively, are

$$f_{\text{area}}^{(i)}(x) = \left[\left(\frac{x}{R^{(i)}}\right)^c - 1\right] 1_{[0,R^{(i)})}(x)\,, \tag{6.2}$$

$$f_{\text{cent}}^{(i)}(x) = \left[x^2 - (R^{(i)})^2\right] 1_{[0,R^{(i)})}(x)\,, \tag{6.3}$$

where $c > 2$.

6.2 Limited-range DA lagrangian gradient

In order to obtain a continuous-time version of the DA algorithm adapted to our coverage problem, we compute the gradient of the Lagrangian F with sensing functions (6.1) in this section. To do so, we first start with a derivation of the association probabilities, and then introduce a partition of Q that takes advantage of the limited-range nature of agent sensors.

6.2.1 Limited-range association probabilities

Similar to the original DA algorithm, we consider each point $q \in Q$ to have some probability of being associated with an agent at $p^{(i)}$. The probabilities, $P(p^{(i)}|q)$ $i \in \{1, \ldots, n\}$, satisfy the following constraint for all $q \in Q$:

$$\sum_{i=1}^{n} P(p^{(i)}|q) = 1\,. \tag{6.4}$$

Lemma 50. *The association probability distribution that minimizes $F = D - TH$ and satisfies* (6.4) *is the Gibbs distribution*

$$P(p^{(i)}|q) = \frac{\exp\left[-\frac{f^{(i)}(\|q-p^{(i)}\|)}{T}\right]}{Z(q)}\,, \quad i \in \{1,\ldots,n\}\,, \tag{6.5}$$

where the normalizing factor is:

$$Z(q) = \sum_{i=1}^{n} \exp\left[-\frac{f^{(i)}(\|q-p^{(i)}\|)}{T}\right]\,. \tag{6.6}$$

Proof. Following the DA derivation, we seek to minimize $F = D - TH$ first with respect to $P(p^{(i)}|q)$ subject to (6.4). We employ the conservation of mass formula found in [15] to compute derivatives. Starting with (2.7),

$$\begin{aligned}\frac{\partial D}{\partial P(p^{(i)}|q)} &= \int_Q \phi(q) f^{(i)}(\|q-p^{(i)}\|)dq \\ &\quad + \int_{\partial Q} \phi(\gamma) \sum_i P(p^{(i)}|\gamma) f^{(i)}(\|\gamma - p^{(i)}\|)\hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\gamma)\frac{\partial \gamma}{\partial P(p^{(i)}|q)}^{0} d\gamma \\ &= \int_Q \phi(q) f(\|q-p^{(i)}\|)dq\,.\end{aligned}$$

In the above, $\gamma\colon \mathbb{S} \to Q$ is a parametrization of the boundary of Q. The outwards pointing unit normal column vector along γ is denoted by $\hat{\mathbf{n}}_{\text{out}}(\gamma)$.

Performing the same differentiation on (2.8),

$$\frac{\partial H}{\partial P(p^{(i)}|q)} = -\int_Q \phi(q)[\log P(p^{(i)}|q) + 1]dq\,.$$

To solve the constrained minimization problem, we use the Lagrange multipliers technique. Let $G = \sum_{i=1}^{n} P(p^{(i)}|q) - 1$. In this way, $\frac{\partial G}{\partial P(p^{(i)}|q)} = 1$. We then solve for

$$\nabla F = \lambda \nabla G\,, \tag{6.7}$$

$$G = 0\,. \tag{6.8}$$

Let $A = \int_Q \phi(q)dq$, then $\lambda(\nabla G)_i = \int_Q \frac{1}{A}\phi(q)\lambda dq$. Starting with (6.7), we have

$$\int_Q \phi(q)\left[f^{(i)}(\|q-p^{(i)}\|) + T\log P(p^{(i)}|q) + T - \frac{\lambda}{A}\right]dq = 0\,.$$

The above is true if for all $q \in Q$,

$$0 = f^{(i)}(\|q - p^{(i)}\|) + T \log P(p^{(i)}|q) + T - \frac{\lambda}{A}$$
$$P(p^{(i)}|q) = \exp\left[\frac{\lambda}{AT} - 1 - \frac{f^{(i)}(\|q - p^{(i)}\|)}{T}\right].$$

Substituting the above into (6.8) results in

$$\exp\left[\frac{\lambda}{AT} - 1\right] = \frac{1}{\sum_{i=1}^{n} \exp\left[-\frac{f^{(i)}(\|q-p^{(i)}\|)}{T}\right]},$$

and we can extract the results (6.5) and (6.6). □

Remark 51. *The function $Z(q)$ is continuous since each $f^{(i)}$ is Lipschitz. This observation proves to be important for simplifying the analysis in future sections.*•

We can take the result (6.5) and substitute it back into F:

$$\hat{F} = \int_Q \phi(q) \Bigg[\sum_i P(p^{(i)}|q) f^{(i)}(\|q - p^{(i)}\|) \tag{6.9}$$
$$+ TP(p^{(i)}|q) \Bigg(- \frac{f^{(i)}(\|q - p^{(i)}\|)}{T} - \log Z(q) \Bigg) \Bigg] dq$$
$$= -T \int_Q \phi(q) \log Z(q) dq \,, \tag{6.10}$$

where we use the fact that $\sum_{i=1}^{n} P(p^{(i)}|q) = 1$.

6.2.2 Limited-range partition

For further analysis, it is advantageous to partition Q such that $Z(q)$ is differentiable over each region in this partition. We start by assuming that each agent's sensing function $f^{(i)}$ is described by equation (6.1). We define the annulus centered at $p^{(i)}$ with inner radius $R_{\alpha-1}^{(i)}$ and outer radius $R_{\alpha}^{(i)}$ as

$$A_{\alpha}^{(i)} = \{q \in Q \parallel R_{\alpha-1}^{(i)} \leq \|q - p^{(i)}\| < R_{\alpha}^{(i)}\}. \tag{6.11}$$

There are $M = \sum_{i=1}^{n} m^{(i)}$ of these sets, so we can equivalently enumerate the $A_{\alpha}^{(i)}$ as $\mathcal{A}_s$ for $s \in \{1, \ldots, M\}$. Additionally, let β be the set of all possible binary sequences with length M, i.e.: each $b_k \in \beta$, $k \in \{1, \ldots, 2^M\}$ is a finite sequence of zeros and ones. Furthermore, the s^{th} element of b_k is denoted by $b_{k,s}$.

Proposition 52. *Let $\{D_k\}$ be a collection of sets such that for each $b_k \in \beta$,*

$$D_k = \bigcap_{s=1}^{M} \{\mathcal{A}_s \, if b_{k,s} = 1 \, ; \mathcal{A}_s^C \, if b_{k,s} = 0\} \, . \tag{6.12}$$

Then, $\{D_k\}$ forms a partition of Q and $Z(q)$ is continuously differentiable in each D_k.

Proof. We show that $\{D_k\}$ forms a partition by verifying that: (i) $\bigcup_k D_k = Q$, and (ii) $D_k \cap D_\ell = \emptyset$ for $k \neq \ell$.

For the first criterion, by definition of the sets $\mathcal{A}_s$, for any $q \in Q$ it is true that $q \in \mathcal{A}_{s^*}$ for some $s^* \in S^* \subseteq \{1, \ldots, M\}$. Then consider the binary sequence b_k such that $b_{k,s} = 1$ for each $s \in S^*$. Then by definition of the regions (6.12), $q \in D_k$. Since q is arbitrary, every point $q \in Q$ lies in some D_k, and so $Q = \bigcup_k D_k$.

For the next criterion, take two different regions D_k and D_ℓ. They are formed from the intersections

$$D_k = \bigcap_{s=1}^{M} \{\mathcal{A}_s \, \text{if} \, b_{k,s} = 1 \, ; \mathcal{A}_s^C \, \text{if} \, b_{k,s} = 0\} \, , \quad D_\ell = \bigcap_{s=1}^{M} \{\mathcal{A}_s \, \text{if} \, b_{\ell,s} = 1 \, ; \mathcal{A}_s^C \, \text{if} \, b_{\ell,s} = 0\} \, ,$$

respectively. Because $k \neq \ell$, the sequences $b_k \neq b_\ell$. Thus, for some index s^*, $b_{k,s^*} \neq b_{\ell,s^*}$. Without loss of generality, suppose $b_{k,s^*} = 1$ and $b_{\ell,s^*} = 0$. Then we have

$$\begin{aligned}
D_k \cap D_\ell &= \left[\bigcap_{s \neq s^*} \{\mathcal{A}_s \, \text{if} \, b_{k,s} = 1 \, ; \mathcal{A}_s^C \, \text{if} \, b_{k,s} = 0\} \cap \mathcal{A}_{s^*} \right] \\
&\quad \cap \left[\bigcap_{s \neq s^*} \{\mathcal{A}_s \, \text{if} \, b_{\ell,s} = 1 \, ; \mathcal{A}_s^C \, \text{if} \, b_{\ell,s} = 0\} \cap \mathcal{A}_{s^*}^C \right] \\
&= \left[\bigcap_{s \neq s^*} \{\mathcal{A}_s \, \text{if} \, b_{k,s} = 1 \, ; \mathcal{A}_s^C \, \text{if} \, b_{k,s} = 0\} \cap \{\mathcal{A}_s \, \text{if} \, b_{\ell,s} = 1 \, ; \mathcal{A}_s^C \, \text{if} \, b_{\ell,s} = 0\} \right] \\
&\quad \cap \left(\mathcal{A}_{s^*} \cap \mathcal{A}_{s^*}^C \right) \\
&= \emptyset \, .
\end{aligned}$$

We have verified both properties, therefore $\{D_k\}$ is a partition of Q.

Next we show that the function $Z(q)$ is continuously differentiable over each D_k. From the definition (6.6), and assumption of the form of $f^{(i)}$ in (6.1), we can

write

$$Z(q) = \sum_{i=1}^{n} \sum_{\alpha=1}^{m^{(i)}} \exp\left[-\frac{f_\alpha^{(i)}(\|q - p^{(i)}\|) 1_{[R_{\alpha-1}^{(i)}, R_\alpha^{(i)})}(\|q - p^{(i)}\|)}{T} \right] . \tag{6.13}$$

Additionally, each of the $f_\alpha^{(i)}$ are differentiable over the annulus centered at $p^{(i)}$ with inner and outer radii of $R_{\alpha-1}^{(i)}$ and $R_\alpha^{(i)}$, respectively. Because the D_k are defined as the intersection of a subset of these annuli, $Z(q)$ is the sum of the same set of continuously differentiable $f_\alpha^{(i)}$ over each region D_k. Thus, $Z(q)$ is continuously differentiable over each D_k. □

In the next section, we will use G_k to refer to the indices of the points $p^{(i)}$ which form the region D_k. That is,

$$G_k = \{ i \in \{1, \ldots, n\} \mid \|q - p^{(i)}\| < R^{(i)}, \forall\, q \in {D_k}^\circ \} .$$

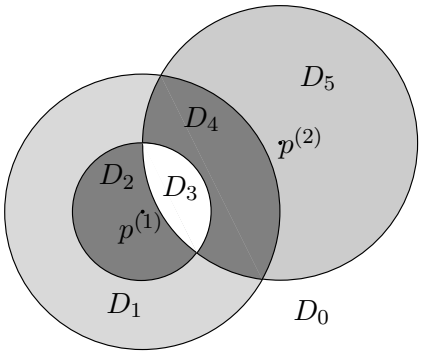
The regions D_k also have a convenient relation to each $\mathbb{B}^{(i)}$.

Proposition 53. *Let $A_\alpha^{(i)}$ be the annulus centered at $p^{(i)}$ with inner radius $R_{\alpha-1}^{(i)}$ and outer radius $R_\alpha^{(i)}$. Each annulus $A_\alpha^{(i)}$ is* exactly *covered by a subcollection of $\{D_k\}$. We denote the indices of this subcollection as $C_\alpha^{(i)}$ such that $A_\alpha^{(i)} = \bigcup_{k \in C_\alpha^{(i)}} D_k$.*

Proof. Since $\{D_k\}$ is a partition of Q, there exists a subcollection $\{D_k\}_{k \in I}$ such that $\mathcal{A}_s \subseteq \bigcup_{k \in I} D_k$. Let I be the smallest index set such that this is true. Then by definition we have $D_k \cap \mathcal{A}_s \neq \emptyset$ for each $k \in I$. Additionally, for each point $p^{(i)}$, the set of all annuli centered at $p^{(i)}$ covers Q, see (6.11) and (6.1).

Now suppose there exists a D_k such that $\mathcal{A}_s \cap D_k$ and $D_k \setminus \mathcal{A}_s$ are both nonempty. Referencing the definition of D_k in (6.12), this D_k must be formed from the intersection of $\mathcal{A}_s$ and another annulus centered at $p^{(i)}$, but with different radii $R_{\alpha-1}^{(i)}, R_\alpha^{(i)}$. This intersection, however, is empty, and such a D_k cannot exist. □

Corollary 54. *Each ball $\mathbb{B}^{(i)}$ of radius $R^{(i)}$ centered at $p^{(i)}$ is* exactly *covered by a subcollection of $\{D_k\}$. We denote the set of indices corresponding to this subcollection as $C^{(i)}$ such that $\mathbb{B}^{(i)} = \bigcup_{k \in C^{(i)}} D_k$.* •



Variable	Description
$A_2^{(1)}$	$D_1 \cup D_4$
G_1	$\{1\}$
G_4	$\{1, 2\}$
$C^{(1)}$	$\{1, 2, 3, 4\}$
$C_1^{(1)}$	$\{2, 3\}$
$C_3^{(1)}$	$\{0, 5\}$
$C^{(2)}$	$\{3, 4, 5\}$
$C_1^{(2)}$	$\{3, 4, 5\}$

Figure 6.1: Graphical description of notation for DA indices and sets.

We now introduce notation that will facilitate the derivation of the gradient direction and the critical temperature check.

Definition 15 (Arcs)**.** *Let* $\text{Arcs}\colon \mathbb{N} \times \mathbb{N} \rightrightarrows Q$ *be the function that takes an agent number,* i*, and a region index* k *and produces the set of circular arcs centered at* $p^{(i)}$ *that are found in* ∂D_k*. That is,* $\text{Arcs}(i, k) = \bigcup_\alpha \partial A_\alpha^{(i)} \cap D_k$*.*

6.2.3 Gradient formulation

The next step in the DA derivation is to optimize the Lagrangian $\hat{F}$ with respect to sensor positions $p^{(i)}$. Each agent in the network will use this result in order to compute its gradient direction.

Proposition 55. *Given the Lagrangian* (6.10)*, and sensing metrics of the form* (6.1)*, the gradient of* (6.10) *is:*

$$\frac{\partial \hat{F}}{\partial p^{(i)}} = -T \sum_{k \in C^{(i)}} \int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial p^{(i)}} dq \,. \tag{6.14}$$

Proof. We begin by taking the following derivative (via the conservation of mass formula in [15]):

$$\frac{\partial \hat{F}}{\partial p^{(i)}} = -T \sum_k \left[\int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial p^{(i)}} dq + \int_{\partial D_k} \phi(\gamma_k) \log Z(\gamma_k) \hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\gamma_k) \frac{\partial \gamma_k}{\partial p^{(i)}} d\gamma_k \right].$$

In the above, $\gamma_k \colon \mathbb{S} \to Q$ is a parametrization of the boundary of region D_k. The outwards pointing unit normal column vector along γ_k is denoted by $\hat{\mathbf{n}}_{\text{out}}(\gamma_k)$.

We now show the integrals over the boundaries ∂D_k vanish when summed over all k. Each γ_k that parametrizes the boundary of D_k is composed of circular arcs centered at various $p^{(i)}$. For a particular $p^{(i)}$, the derivative $\frac{\partial \gamma_k}{\partial p^{(i)}}$ is nonzero only when the γ_k parametrizes an arc centered at $p^{(i)}$. Since each $\text{Arcs}(i,k)$ is a fixed radius from $p^{(i)}$,

$$\frac{\partial \gamma_k}{\partial p^{(i)}} = \begin{cases} I_\Lambda \,, & \gamma_k \in \text{Arcs}(i,k) \,, \\ 0 \,, & \text{otherwise} \,, \end{cases}$$

where I_Λ is the square identity matrix of size Λ. Thus, only the integral along the boundaries $\text{Arcs}(i,k)$ needs to be considered. The derivative is now simplified to

$$\frac{\partial \hat{F}}{\partial p^{(i)}} = -T \sum_k \left[\int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial p^{(i)}} dq + \int_{\text{Arcs}(i,k)} \phi(\gamma_k) \log Z(\gamma_k) \hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\gamma_k) d\gamma_k \right].$$

Since each arc in $\text{Arcs}(i,k)$ is part of the boundary ∂D_k, Each arc in $\text{Arcs}(i,k)$ is shared between two regions D_k and D_ℓ. Thus, there will be two integrals over each $\text{Arcs}(i,k)$: one from D_k and one from D_ℓ. For these two integrals, the normal vector $\hat{\mathbf{n}}_{\text{out}}(\gamma_k)$ will be equal and opposite. Additionally the function $Z(q)$ is continuous over Q, so the sum of these two integrals will be zero. The derivative is simplified to

$$\frac{\partial \hat{F}}{\partial p^{(i)}} = -T \sum_k \int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial p^{(i)}} dq \,.$$

We now show that the derivative $\frac{\partial Z}{\partial p^{(i)}}$ is zero if $q \notin \mathbb{B}^{(i)}$. Recall from the limited-range assumption that each sensing function $f_\alpha^{(i)}(x)$ is a constant if $x \geq R^{(i)}$. Therefore Z as in (6.13) has no dependence on $p^{(i)}$ if $\|q - p^{(i)}\| \geq R^{(i)}$. With this realization, we obtain the result (6.14). □

Remark 56. *For the area-maximizing case* (6.2)*, we begin by computing the*

derivative $\frac{\partial Z}{\partial p^{(i)}}$.

$$\begin{aligned}\frac{\partial Z}{\partial p^{(i)}} &= \sum_{j=1}^{n} \frac{\partial}{\partial p^{(i)}} \exp\left[-\frac{f^{(i)}_{area}(\|q-p^{(j)}\|)}{T}\right] \\ &= -\frac{1}{T}\frac{\partial}{\partial p^{(i)}}[f^{(i)}_{area}(\|q-p^{(i)}\|)] \exp\left[-\frac{f^{(i)}_{area}(\|q-p^{(i)}\|)}{T}\right].\end{aligned}$$

As an aside, we have the following result:

$$\begin{aligned}\frac{\partial}{\partial p^{(i)}}[\|q-p^{(i)}\|^c] &= \frac{\partial}{\partial p^{(i)}}\left[\left((q-p^{(i)})^{\mathsf{T}}(q-p^{(i)})\right)^{c/2}\right] \\ &= \frac{c}{2}\left(\|q-p^{(i)}\|^2\right)^{c/2-1}\frac{\partial}{\partial p^{(i)}}\left[\|q\|^2 - 2q^{\mathsf{T}}p^{(i)} + \|p^{(i)}\|^2\right] \\ &= c\|q-p^{(i)}\|^{c-2}(p^{(i)}-q)^{\mathsf{T}}.\end{aligned}$$

Then using the above result when taking the derivative of (6.2)*, we obtain the following gradient expression*

$$\begin{aligned}\frac{\partial \hat{F}}{\partial p^{(i)}} &= -T\sum_{k\in C^{(i)}}\int_{D_k}\phi(q)\frac{c\|q-p^{(i)}\|^{c-2}}{(R^{(i)})^c T}(q-p^{(i)})^{\mathsf{T}}P(p^{(i)}|q)dq \\ &= -\frac{c}{(R^{(i)})^c}\sum_{k\in C^{(i)}}\int_{D_k}\phi(q)\|q-p^{(i)}\|^{c-2}(q-p^{(i)})^{\mathsf{T}}P(p^{(i)}|q)dq\,. \qquad (6.15)\end{aligned}$$

Here we see that the weight $\|q-p^{(i)}\|^{c-2}$ *serves to amplify the density* ϕ *for points close to the boundary of* $\mathbb{B}^{(i)}$ *while neglecting the value of* ϕ *close to* $p^{(i)}$ *for large* c*. This achieves the area-maximizing effect that we seek.* •

Remark 57. *We can compute the derivative* $\frac{\partial Z}{\partial p^{(i)}}$ *using the sensing function* (6.3) *in a similar manner. We then obtain the gradient* (6.14) *to be:*

$$\frac{\partial \hat{F}}{\partial p^{(i)}} = -2\sum_{k\in C^{(i)}}\int_{D_k}\phi(q)(q-p^{(i)})^{\mathsf{T}}P(p^{(i)}|q)dq\,. \qquad (6.16)$$

This is similar to the gradient expression for the mixed coverage case in [15], with the addition of the association probabilities $P(p^{(i)}|q)$ *as an extra weighting factor.*•

6.2.4 Constant factor approximation

Thus far, we have developed a gradient direction for agents to follow in order to minimize (6.10), which for low temperatures, minimizes the underlying

distortion function (2.7). We now relate how minimizing (6.10) for $\max_i R^{(i)} < \text{diam}(Q)$ bears some relation to minimization of (6.10) for when $R^{(i)} > \text{diam}(Q)$ for all i. In other words, we demonstrate the relation between the gradient-descent of the limited-range DA algorithm with the gradient descent of the original DA algorithm.

We compare the mixed case (6.3) as this is most similar to the distance metric found in [68] and referenced as (2.9). This proposition makes it clear that minimization of (6.10) is equivalent to minimization of the analogous function in the original DA case as sensing radii increase.

Proposition 58. *Let F_0 be defined as in* (2.9)*. Additionally, let $\hat{F}$ be defined as from* (6.10)*. Then, the following is true:*

$$\hat{F} + \min_i (R^{(i)})^2 \leq F_0 \leq \hat{F} + \text{diam}^2(Q)\,. \tag{6.17}$$

Proof. Let $f^{(i)}(x) = x^2$ and let $f^{(i)}_{\text{cent}}(x)$ be defined as in (6.3). Additionally, let $\alpha = \min_i R^{(i)}$ and $d = \text{diam}(Q)$. Then it is true that $-f^{(i)}_{\text{cent}}(x) - d^2 \leq -f^{(i)}(x) \leq -f^{(i)}_{\text{cent}}(x) - \alpha^2$. Since the exponential and logarithmic functions are monotonic increasing, the following statements hold:

$$\begin{aligned}
\sum_i \exp\left[-\frac{f^{(i)}_{\text{cent}}(x) + d^2}{T}\right] \leq \sum_i \exp\left[-\frac{f^{(i)}(x)}{T}\right] \leq \sum_i \exp\left[-\frac{f^{(i)}_{\text{cent}}(x) + \alpha^2}{T}\right],\\
-\frac{d^2}{T} + \log \sum_i \exp\left[-\frac{f^{(i)}_{\text{cent}}(x)}{T}\right] \leq \log \sum_i \exp\left[-\frac{f^{(i)}(x)}{T}\right]\\
\leq -\frac{\alpha}{T} + \log \sum_i \exp\left[-\frac{f^{(i)}_{\text{cent}}(x)}{T}\right].
\end{aligned}$$

Substitute each of these into the expression for F to obtain:

$$\begin{aligned}
-T \int_Q \phi(q) \left[-\frac{\alpha^2}{T} + \log Z(q)\right] dq \leq -T \int_Q \phi(q) \log Z_0(q) dq\\
\leq -T \int_Q \phi(q) \left[-\frac{d^2}{T} + \log Z(q)\right] dq\,.
\end{aligned}$$

The result (6.17) follows since ϕ can be normalized over Q. ∎

6.3 Limited-range DA phase changes

As temperature decreases, the equilibrium points of $\hat{F}$ under the evolution of (6.14) become unstable in a special sense. The set of equilibrium agent positions $\{p^{(i)^*}\}$ for a given temperature T and number of agents n may not be the same set of equilibrium positions for temperature T and $n+1$ agents. When this happens a phase change occurs and we say that we have reached a critical temperature. We present a sufficient condition for agents to individually check if they have reached a critical temperature value under both area-maximizing and mixed centroidal-area coverage.

The introduction of virtual agents proves to be a distributed way to check the special sense of instability that occurs at phase changes. Using a similar argument as in [68], we enlarge the group of leaders $\{p^{(1)}, \ldots, p^{(n)}\}$ with a set of virtual agents $\{p^{(n+1)}, \ldots, p^{(l)}\}$ so that for all $j \in \{n+1, \ldots, l\}$, $p^{(j)} = p^{(i)}$ for some $i \in \{1, \ldots, n\}$. Then we introduce perturbations $\Psi = (\psi^{(1)}, \ldots, \psi^{(l)}) \in \mathbb{R}^{2l}$. Given a scaling factor ϵ, consider the perturbed agent locations, $x^{(i)} = p^{(i)} + \epsilon\psi^{(i)}$, for $i \in \{1, \ldots, l\}$. Critical points of $\hat{F}$ correspond to configurations where $\frac{d\hat{F}(x^{(1)},\ldots,x^{(l)})}{d\epsilon}\Big|_{\epsilon=0} = 0$. However, those configurations fail to be a minimum when the second derivative $\frac{d^2\hat{F}}{d\epsilon^2}\Big|_{\epsilon=0} \leq 0$.

We now find the second derivative. Consider the partition $\{D_k\}$ associated with the $\{x^{(i)}\}$, $i \in \{1, \ldots, l\}$. The first derivative of the Lagrangian (6.10) with respect to ϵ is

$$\frac{d\hat{F}}{d\epsilon} = -T \sum_k \left[\int_{D_k} \phi(q) \frac{1}{Z(q)} \frac{\partial Z}{\partial \epsilon} dq + \int_{\partial D_k} \phi(\gamma_k) \log Z(\gamma_k) \hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\gamma_k) \frac{\partial \gamma_k}{\partial \epsilon} d\gamma_k \right].$$

Using the same reasoning as before when computing the gradient (6.14), the integrals over the boundaries ∂D_k cancel when summed over all k. Taking another derivative with respect to ϵ,

$$\begin{aligned} \frac{d^2\hat{F}}{d\epsilon^2} = -T \sum_k \Bigg\{ & \int_{D_k} \phi(q) \left[-\frac{1}{Z^2(q)} \left(\frac{\partial Z}{\partial \epsilon} \right)^2 + \frac{1}{Z(q)} \frac{\partial^2 Z}{\partial \epsilon^2} \right] dq \\ & + \int_{\partial D_k} \phi(\gamma_k) \frac{1}{Z(\gamma_k)} \frac{\partial Z}{\partial \epsilon} \hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\gamma_k) \frac{\partial \gamma_k}{\partial \epsilon} d\gamma_k \Bigg\}. \end{aligned} \tag{6.18}$$

Unfortunately, the same convenient cancellation of the boundary terms may not occur here. Since each $f^{(i)}(x)$ is only Lipschitz, and $Z(q)$ is composed of a sum of exponentials of $f^{(i)}$, $Z(q)$ is also only Lipschitz. The derivative of Z evaluated at one side of the boundary ∂D_k may not be the same as it is evaluated on the other side.

Let $y^{(i)} = q - x^{(i)}$ to reduce the amount of notation. The derivative $\frac{dZ}{d\epsilon}$ is computed as follows:

$$\frac{dZ}{d\epsilon} = \sum_{i=1}^{l} \sum_{\alpha=1}^{m^{(i)}} -\frac{1}{T} \frac{\partial f_\alpha^{(i)}}{\partial \epsilon} \exp\left[-\frac{f_{i,\alpha}(\|y^{(i)}\|) 1_{[R_{\alpha-1}^{(i)}, R_{i,\alpha})}(\|y^{(i)}\|)}{T} \right] 1_{[R_{\alpha-1}^{(i)}, R_{i,\alpha})}(\|y^{(i)}\|) \,.$$

Again note that this derivative may not be continuous if $\frac{\partial f_{i,\alpha}}{\partial \epsilon}$ is not continuous. In a particular region D_k, the above simplifies to

$$\frac{dZ}{d\epsilon} = \sum_{j \in G_k} -\frac{1}{T} \frac{\partial f_{j,\alpha}}{\partial \epsilon} \exp\left[-\frac{f_\alpha^{(j)}(\|y^{(j)}\|)}{T} \right] . \tag{6.19}$$

This is because the indicator function evaluates to zero if $\|y^{(i)}\| = \|q - x^{(i)}\| \leq R^{(i)}$, and the index set G_k captures all such $x^{(i)}$ that satisfy this condition. It should be noted that the α in $f_\alpha^{(i)}$ may vary. This α indexes the sensing metric associated with an annulus centered at an agent $x^{(j)}$. Since the region D_k is fixed, the associated annulus index corresponding to the region D_k may change for various $j \in G_k$.

Continuing, the second derivative is:

$$\frac{d^2 Z}{d\epsilon^2} = \sum_{j \in G_k} \left\{ \left[\left(\frac{1}{T} \frac{\partial f_\alpha^{(j)}}{\partial \epsilon} \right)^2 - \frac{1}{T} \frac{\partial^2 f_\alpha^{(j)}}{\partial \epsilon^2} \right] \exp\left[-\frac{f_\alpha^{(j)}(\|y^{(j)}\|)}{T} \right] \right\} . \tag{6.20}$$

Since $y^{(i)} = q - p^{(i)} - \epsilon \psi^{(i)}$, using the chain rule, $\frac{\partial f_\alpha^{(i)}}{\partial \epsilon} = -{\psi^{(i)}}^\mathsf{T} \frac{\partial f_\alpha^{(i)}}{\partial y^{(i)}}$ and $\frac{\partial^2 f_\alpha^{(i)}}{\partial \epsilon^2} = {\psi^{(i)}}^\mathsf{T} \frac{\partial^2 f_\alpha^{(i)}}{\partial {y^{(i)}}^2} \psi^{(i)}$.

We substitute the results (6.19) and (6.20) into (6.18). Additionally, since

$\frac{1}{Z}\exp\left[-\frac{f_\alpha^{(i)}}{T}\right] = P(x^{(i)}|q)$ we get:

$$\frac{d^2\hat{F}}{d\epsilon^2} = -T\sum_k \Bigg\{ \int_{D_k} \phi(q)\Bigg(-\frac{1}{T^2}\left(\sum_{j\in G_k}\frac{\partial f_\alpha^{(j)}}{\partial\epsilon}P(x^{(j)}|q)\right)^2 + \sum_{j\in G_k}\left[\left(\frac{1}{T}\frac{\partial f_\alpha^{(j)}}{\partial\epsilon}\right)^2 - \frac{1}{T}\frac{\partial^2 f_\alpha^{(j)}}{\partial\epsilon^2}\right]P(x^{(j)}|q)\Bigg)dq - \frac{1}{T}\int_{\partial D_k}\phi(\gamma_k)\sum_{j\in G_k}\left(\frac{\partial f_\alpha^{(j)}}{\partial\epsilon}P(p^{(j)}|\gamma_k)\right)\hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\gamma_k)\frac{\partial\gamma_k}{\partial\epsilon}d\gamma_k\Bigg\}. \tag{6.21}$$

The check for critical temperature is to numerically compute $\frac{d^2\hat{F}}{d\epsilon^2}\Big|_{\epsilon=0}$ at an equilibrium configuration. The equilibrium configurations occurs when $\frac{\partial\hat{F}}{\partial p^{(i)}} = 0$ for all i, or equivalently, when $\frac{d\hat{F}}{d\epsilon}\Big|_{\epsilon=0} = 0$. If the second derivative is negative, then the equilibrium configuration is unstable, and that signifies that we are below a critical temperature value. To simplify the critical temperature check and make it spatially distributed, we consider the following perturbation. Let $S^{(i)} \subseteq \{1,\ldots,m\}$ be such that $j \in S^{(i)}$ implies $p^{(j)} = p^{(i)}$. We define $\Psi^{(i)}$ to be

$$\Psi^{(i)} = \Big\{(\psi^{(1)},\ldots,\psi^{(m)}) \mid \psi^{(j)} = 0,\, \forall\, j \notin S^{(i)};\, \sum_{j\in S^{(i)}}\psi^{(j)} = 0\Big\}. \tag{6.22}$$

If the critical temperature has not yet been reached, then these coincident agents (i.e., leaders and virtual agents) will remain together. Otherwise, the coincident agents are at an unstable equilibrium point, and any perturbation will force them apart. By using this particular perturbation, we will obtain a sufficient condition for critical temperature. We will now take the above results and consider the metric functions (6.2) and (6.3).

6.3.1 Area metric

For the area-maximizing metric, we present the following check for critical temperature.

Proposition 59. *Critical temperature for the area-maximizing DA algorithm has been reached if for $i \in \{1,\ldots,n\}$, any of the following matrices $\mathcal{F}^{(i)}$ are non-*

positive definite:

$$\mathcal{F}^{(i)} = \sum_{k \in C^{(i)}} \int_{D_k} \phi(q) P(p^{(i)}|q) \Bigg(\left[(c-2) - \frac{c\|q-p^{(i)}\|^c}{T(R^{(i)})^c} \right] (q-p^{(i)})(q-p^{(i)})^{\mathsf{T}} + \|q-p^{(i)}\|^2 I \Bigg) dq\,, \quad (6.23)$$

and $\dot{p}^{(i)} = 0$ *for all* $i \in \{1, \ldots, n\}$.

Proof. With this choice of perturbations, consider the case using the area metric $f^{(i)}_{\text{area}}$ from (6.2). We fix an i such that the perturbation (6.22) is true and compute $\frac{d^2\hat{F}}{d\epsilon^2}\Big|_{\epsilon=0}$ using (6.21).

The derivatives $\frac{\partial f^{(i)}_{\text{area}}}{\partial y^{(i)}}$ and $\frac{\partial^2 f^{(i)}_{\text{area}}}{\partial {y^{(i)}}^2}$ are:

$$\begin{aligned}
\frac{\partial f^{(i)}_{\text{area}}}{\partial y^{(i)}} &= \frac{1}{(R^{(i)})^c} \frac{c}{2} (\|y^{(i)}\|^2)^{c/2-1} (2{y^{(i)}}^{\mathsf{T}}) = \frac{c}{(R^{(i)})^c} \|y^{(i)}\|^{c-2} {y^{(i)}}^{\mathsf{T}}\,, \\
\frac{\partial^2 f^{(i)}_{\text{area}}}{\partial {y^{(i)}}^2} &= \frac{c}{(R^{(i)})^c} \left(\frac{c}{2} - 1 \right) (\|y^{(i)}\|^2)^{c/2-2} (2y^{(i)}) {y^{(i)}}^{\mathsf{T}} + \frac{c}{(R^{(i)})^c} \|y^{(i)}\|^{c-2} I \\
&= \frac{c}{(R^{(i)})^c} \|y^{(i)}\|^{c-4} \left[(c-2) y^{(i)} {y^{(i)}}^{\mathsf{T}} + \|y^{(i)}\|^2 I \right].
\end{aligned}$$

Since $y^{(i)} = q - p^{(i)} - \epsilon\psi^{(i)}$, when $\epsilon = 0$, $y^{(j)} = q - p^{(j)}$ for all $j \in S^{(i)}$. However, since $p^{(j)} = p^{(i)}$, $y^{(j)} = q - p^{(i)}$ for all $j \in S^{(i)}$. Similarly, the association probabilities $P(x^{(j)}|q) = P(p^{(i)}|q)$ for all $j \in S^{(i)}$.

When considering the perturbations (6.22) on (6.21), we note the following simplification. In (6.21), there is a sum over $j \in G_k$ involving the first and second derivatives, $\frac{\partial f^{(j)}_\alpha}{\partial \epsilon} = \psi^{(j)} \frac{\partial f^{(j)}_\alpha}{\partial y^{(j)}}$ and $\frac{\partial^2 f^{(j)}_\alpha}{\partial \epsilon^2} = {\psi^{(j)}}^{\mathsf{T}} \frac{\partial^2 f^{(j)}_\alpha}{\partial {y^{(j)}}^2} \psi^{(j)}$ respectively. However, if $j \notin S^{(i)}$ then $\psi^{(j)} = 0$. Therefore instead of summing over $j \in G_k$, we can equivalently sum over $j \in S^{(i)}$.

With the above results, the second derivative (6.21) evaluated at $\epsilon = 0$ can

be simplified as follows:

$$\begin{aligned}
\left.\frac{d^2\hat{F}}{d\epsilon^2}\right|_{\epsilon=0} &= -T\sum_{k\in C^{(i)}}\Bigg\{\int_{D_k}\phi(q)\Bigg(-\frac{1}{T^2}\Bigg(-\frac{c\|q-p^{(i)}\|^{c-2}}{(R^{(i)})^c}(q-p^{(i)})^{\mathsf{T}} \\
&\quad \cdot P(p^{(i)}|q)\cancelto{0}{\sum_{j\in S^{(i)}}}\psi^{(j)}\Bigg)^2+\sum_{j\in S^{(i)}}\left[\left(\frac{1}{T}\frac{\partial f_\alpha^{(j)}}{\partial\epsilon}\right)^2-\frac{1}{T}\frac{\partial^2 f_\alpha^{(j)}}{\partial\epsilon^2}\right]P(p^{(i)}|q)\Bigg)dq \\
&\quad -\frac{1}{T}\int_{\partial D_k}\phi(\gamma_k)\Bigg(-\frac{c\|\gamma_k-p^{(i)}\|^{c-2}}{(R^{(i)})^c}(\gamma_k-p^{(i)})^{\mathsf{T}}P(p^{(i)}|\gamma_k)\cancelto{0}{\sum_{j\in S^{(i)}}}\psi^{(j)}\Bigg) \\
&\quad \cdot\hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\gamma_k)\frac{\partial\gamma_k}{\partial\epsilon}d\gamma_k\Bigg\} \\
&= -T\sum_{k\in C^{(i)}}\Bigg\{\int_{D_k}\phi(q)\left(\sum_{j\in S^{(i)}}\left[\left(\frac{1}{T}\frac{\partial f_\alpha^{(j)}}{\partial\epsilon}\right)^2-\frac{1}{T}\frac{\partial^2 f_\alpha^{(j)}}{\partial\epsilon^2}\right]P(p^{(i)}|q)\right)dq \\
&= -\sum_{j\in S^{(i)}}\sum_{k\in C^{(i)}}\int_{D_k}\phi(q)P(p^{(i)}|q)\Bigg[\frac{1}{T}\left(-\frac{c\|q-p^{(i)}\|^{c-2}}{(R^{(i)})^c}{\psi^{(j)}}^{\mathsf{T}}(q-p^{(i)})\right)^2 \\
&\quad -{\psi^{(j)}}^{\mathsf{T}}\frac{c}{(R^{(i)})^c}\|q-p^{(i)}\|^{c-4}\left[(c-2)(q-p^{(i)})(q-p^{(i)})^{\mathsf{T}}\right. \\
&\quad \left.+\|q-p^{(i)}\|^2 I\right]\psi^{(j)}\Bigg]dq\,.
\end{aligned}$$

Factoring out $\psi^{(j)}$ from the left and right sides and using the substitution (6.23), the second derivative evaluated at $\epsilon = 0$ is:

$$\left.\frac{d^2\hat{F}}{d\epsilon^2}\right|_{\epsilon=0} = \frac{c\|q-p^{(i)}\|^{c-4}}{(R^{(i)})^c}\sum_{j\in S^{(i)}}{\psi^{(j)}}^{\mathsf{T}}\mathcal{F}^{(i)}\psi^{(j)}\,.$$

It is clear now that in order for an equilibrium configuration to be stable in the area-maximizing case, the matrix quantity in (6.23) must be positive definite. □

6.3.2 Mixed metric

We perform the same analysis for the mixed centroidal-area coverage sensing metric (6.3). This metric is most similar to that found in [68] and [77].

Proposition 60. *Critical temperature for the centroidal-area DA algorithm has*

been reached if for $i \in \{1, \ldots, n\}$*, any of the following matrices* $\mathcal{F}^{(i)}$ *are non-positive definite:*

$$\mathcal{F}^{(i)} = \sum_{k \in C^{(i)}} \int_{D_k} \phi(q) P(p^{(i)}|q) \left[I - \frac{2}{T}(q - p^{(i)})(q - p^{(i)})^{\mathsf{T}} \right] dq \,, \tag{6.24}$$

and $\dot{p}^{(i)} = 0$ *for all* $i \in \{1, \ldots, n\}$.

Proof. The derivatives $\frac{\partial f_{\text{cent}}^{(i)}}{\partial y^{(i)}}$ and $\frac{\partial^2 f_{\text{cent}}^{(i)}}{\partial {y^{(i)}}^2}$, when $\|y^{(i)}\| \leq R^{(i)}$, are:

$$\frac{\partial f_{\text{cent}}^{(i)}}{\partial y^{(i)}} = 2{y^{(i)}}^{\mathsf{T}}$$
$$\frac{\partial^2 f_{\text{area}}^{(i)}}{\partial {y^{(i)}}^2} = 2I \,.$$

Since $y^{(i)} = q - p^{(i)} - \epsilon\psi^{(i)}$, when $\epsilon = 0$, $y^{(j)} = q - p^{(i)}$ for all $j \in S^{(i)}$. Similarly, the association probabilities $P(x^{(j)}|q) = P(p^{(i)}|q)$ for all $j \in S^{(i)}$. Therefore, with the perturbations (6.22) and the mixed metric (6.3), the second derivative (6.21) evaluated at $\epsilon = 0$ can be simplified as follows:

$$\begin{aligned}
\left.\frac{d^2\hat{F}}{d\epsilon^2}\right|_{\epsilon=0} &= -T \sum_{k \in C^{(i)}} \Bigg\{ \int_{D_k} \phi(q) \Bigg(-\frac{1}{T^2} \left(-2(q - p^{(i)})^{\mathsf{T}} P(p^{(i)}|q) \cancelto{0}{\sum_{j \in S^{(i)}} \psi^{(j)}} \right)^2 \\
&\quad + \sum_{j \in S^{(i)}} \left[\left(-\frac{2}{T} {\psi^{(j)}}^{\mathsf{T}} (q - p^{(i)}) \right)^2 - \frac{2}{T} {\psi^{(j)}}^{\mathsf{T}} I \psi^{(j)} \right] P(p^{(i)}|q) \Bigg) dq \\
&\quad - \frac{1}{T} \int_{\partial D_k} \phi(\gamma_k) \left(-2P(p^{(i)}|\gamma_k)(\gamma_k - p^{(i)})^{\mathsf{T}} \cancelto{0}{\sum_{j \in S^{(i)}} \psi^{(j)}} \right) \hat{\mathbf{n}}_{\text{out}}^{\mathsf{T}}(\gamma_k) \frac{\partial \gamma_k}{\partial \epsilon} d\gamma_k \Bigg\} \\
&= -\sum_{k \in C^{(i)}} \Bigg\{ \int_{D_k} \phi(q) P(p^{(i)}|q) \\
&\quad \cdot \left[\frac{4}{T} \sum_{j \in S^{(i)}} \left(-{\psi^{(j)}}^{\mathsf{T}} (q - p^{(i)}) \right)^2 - 2{\psi^{(j)}}^{\mathsf{T}} I \psi^{(j)} \right] dq \Bigg\} \\
&= 2 \sum_{j \in S^{(i)}} \sum_{k \in C^{(i)}} \int_{D_k} \phi(q) P(p^{(i)}|q) \left[{\psi^{(j)}}^{\mathsf{T}} I \psi^{(j)} - \frac{2}{T} \left({\psi^{(j)}}^{\mathsf{T}} (q - p^{(i)}) \right)^2 \right] dq \,.
\end{aligned}$$

Factoring out $\psi^{(j)}$ from the left and right sides and using the substitu-

tion (6.24), the second derivative evaluated at $\epsilon = 0$ is:

$$\left.\frac{d^2\hat{F}}{d\epsilon^2}\right|_{\epsilon=0} = 2\sum_{j\in S^{(i)}} {\psi^{(j)}}^{\mathsf{T}} \mathcal{F}^{(i)} \psi^{(j)} .$$

It is clear now that in order for an equilibrium configuration to be stable in the area-maximizing case, the matrix quantity in (6.24) must be positive definite. □

6.4 Distributed implementation

We have so far demonstrated how a network of agents can descend the gradient and check for phase changes in a distributed DA algorithm. However, we still must provide a distributed method for implementing these phase changes.

6.4.1 Algorithm descriptions

The DA algorithm begins with one active agent, and the other agents moving in formation with it. A formation will split in two if its critical temperature is reached. The agents following in formation are divided evenly between the current formation leader and a new formation leader. After the first phase change, it is possible that future phase changes occur at an agent who is by itself. Therefore, this agent must communicate its desire for an additional companion, and the network of agents must distributively assign an inactive agent to this task. We propose a task-assignment algorithm to accomplish this.

We provide a possible scheme under the following assumptions: (1) Agents have knowledge of the total number of formations n and the total number of agents N, (2) The communication graph between all active agents is connected, (3) Each active agent knows the number of inactive agents traveling with it, and (4) All agents have knowledge of the initial temperature, and the cooling factor κ.

Connectivity of the communication graph is important because both the temperature and the total number of active agents must be constant across all agents. We assume that if the graph is connected, the agents can agree on the current temperature, and determine through a flooding algorithm (see [45]) the

Algorithm 3: Distributed DA algorithm for each agent

$T \leftarrow$ initial temperature
while $T > T_{min}$ **or** $n < N$ **do**
  while *floodMax*($\|\dot{p}^{(i)}\|$) $> \epsilon$) **do**
    $\dot{p}^{(i)} \leftarrow -$computeGradient()
  end
  if *checkSplit()* == *true* **then** flood("T_c reached")
  if *received "T_c reached"* **then**
    doTaskAssign() $N - n$ times
  end
  $T \leftarrow \kappa T$
end
doNormalCoverage()

number of active agents n at any point in time. Additionally, agents must wait for the flooding algorithms to terminate; the worst case is proportional to the diameter of the communication graph.

First we define some algorithmic primitives specific to this application. We let `computeGradient()` be the function that computes (6.14), and we let `checkSplit()` be the function that determines if a critical temperature has been reached as in (6.23) and (6.24). Finally, we introduce `doNormalCoverage()` to mean to perform limited-range coverage as from [15].

The distributed DA algorithm can informally be described as follows, see Algorithm 3. Starting with a single formation, and a high initial temperature, formations descend the gradient (6.14). When all agents agree they are stationary, they individually check for phase changes and, if necessary, implement Algorithm 4 $N - n$ times to guarantee the assignment of all companion requests. The temperature is lowered, regardless of whether or not there was a phase change, and the gradient descent is continued. This process repeats until the system temperature is below a minimum temperature threshold T_{min} or if $n = N$. Once this happens, the agents perform the normal coverage algorithm described in [15], as this

is equivalent to having $T = 0$.

6.4.2 Algorithm complexity

We first characterize the complexity of the task assignment Algorithm 4 in terms of worst-case number of messages passed. Then we address the time complexity of the overall Algorithm 3 in terms of the time complexity for a typical gradient descent.

Algorithm 4 outlines the task assignment algorithm for agents who are in need of a companion to split. Roughly speaking, there are three rounds of communication where an agent broadcasts its need for a companion, other agents reply if they can help, and finally a handshake is formed with the agent transfer. In this algorithm, n is incremented for every new formation, and this command is flooded over the network. This algorithm has a finite termination time upper bounded by $3n + n(N - n)$ messages passed.

While work has been done addressing the convergence rate for Lloyd's algorithm in one dimension, cf. [48, 19], the convergence rate for such gradient descent algorithms in higher dimensions remains open. Additionally, with a dynamic gradient descent as in Algorithm 3, we assume that motion of agents is slow compared to the delays involved in wireless communication or gradient computation. Nevertheless, one can compare the performance of the DA algorithm to the typical Lloyd gradient descent.

Referring to Algorithm 3, a dynamic gradient descent using the gradient (6.14) is performed for $n \leq N$ agents until a phase change is handled. At a phase change resolution, it is possible for more than one agent to undergo a phase change, but the worst-case scenario would be for the entire process to undergo $N - 1$ phase changes before performing `doNormalCoverage()`. After each phase change, a new gradient descent is performed. Thus, compared to a normal gradient descent, the DA algorithm could perform N gradient descents. In general, this represents a trade off between overall execution time and optimality of the final cost function.

Remark 61. Alternatively, we can imagine a scenario where a total of N agents wish to deploy in an environment from a base station. First, a single agent departs and descends the DA gradient. At the time of a phase change, this agent radios back to the base asking for help, and the next agent joins the deployment task. This process repeats until all N agents have left the base. Thus, we would achieve better coverage than typical gradient descent with mostly local communication, punctuated with $N-1$ long-distance communication instances back to the base station. •

6.5 Simulations

We present a simulation of the limited-range DA algorithm using the area-maximizing sensing function (6.2) in Figure 6.2. The total number of agents is $N = 8$ and the square region Q has length 10 per side. Each agent has a sensing radius of $R = 2$. The initial temperature is $T = 10$ and the annealing factor $\kappa = 0.9$ in this simulation.

As the temperature decreases, we see the agents split and continue to increase coverage area. Once the minimum temperature is reached, the coverage algorithm of [15] is conducted, and Figure 6.2 (f) shows the limited-range Voronoi partitions separating each agent.

To compare this algorithm with the normal limited-range (area) coverage algorithm in [15], we conducted 100 simulations of the normal coverage algorithm and compared the cost function (2.7) for $T = 0$. Eight agents with sensing radius $R = 2$ were uniformly and randomly distributed over Q. Over the 100 runs, the minimum area covered was 93.5% of the total area, the maximum was 96.4% and the mean was 95.9%. The limited-range DA algorithm had a final coverage area of 96.1%. For the particular ϕ in Figure 6.2, the normal (area) coverage simulations show how the final cost has dependence on initial conditions. However, the distributed DA algorithm converged to the same cost over many random initial conditions.

Next, we present a simulation of the limited-range DA algorithm using

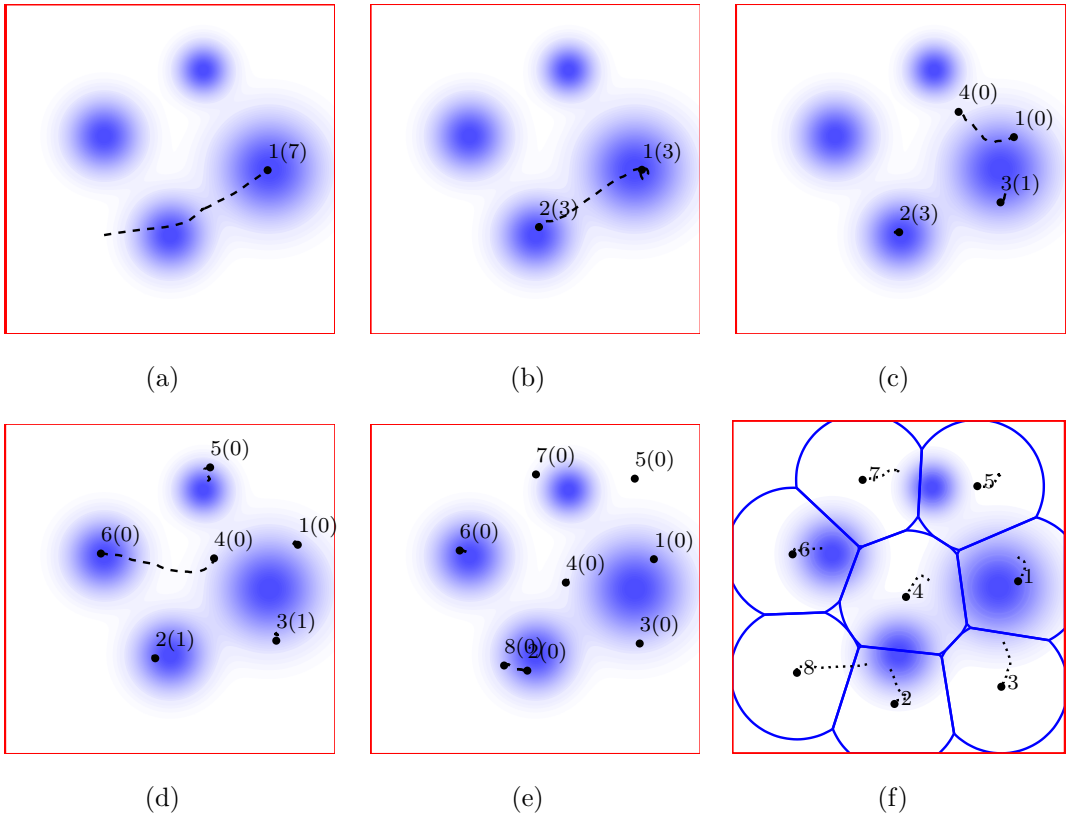


Figure 6.2: A typical run of the limited-range DA algorithm. Agents are labeled in (a)-(f) and the number of agents in formation are contained in parenthesis in (a)-(e). Agent positions at each time step are shown as the small circles. The temperatures begin at $T = 10$ (a), then decrease: $T = 0.15$ (b), $T = 0.12$ (c), $T = 0.1$ (d), $T = 0.01$ (e). The final panel (f) shows the result of running normal coverage starting from the configuration of (e).

the mixed centroidal-area sensing function (6.3). The total number of agents is $N = 6$ and the square region Q has length 10 per side. We will demonstrate the performance of the DA algorithm versus a normal Lloyd-type gradient descent found in [15] as sensing radius decreases.

We first consider the limiting case where all agents can sense the entire region Q. We ran the DA algorithm once, since the initial condition does not influence the outcome of the convergence. For comparison, we ran 50 simulations starting from random initial conditions of the Lloyd-like algorithm. The DA algorithm converges to the optimal cost of 10.64 *regardless of initial condition* in Figure 6.3. On the other hand, the Lloyd-like gradient descent achieved this final

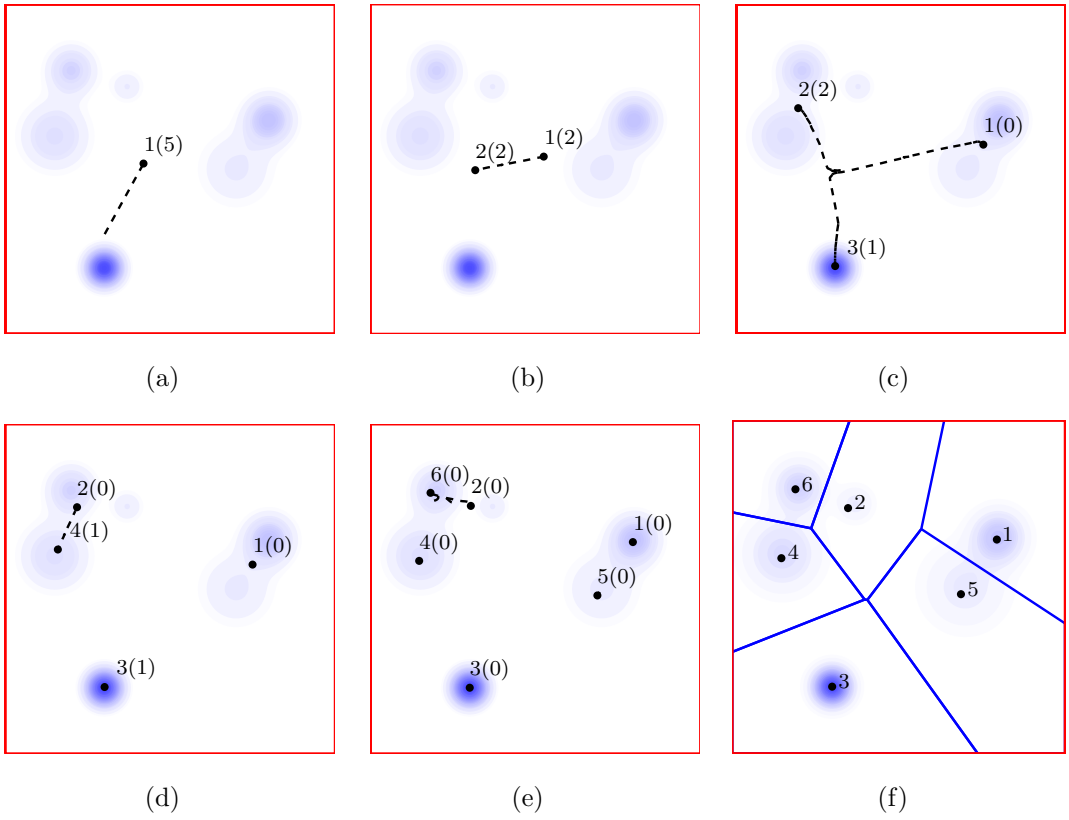
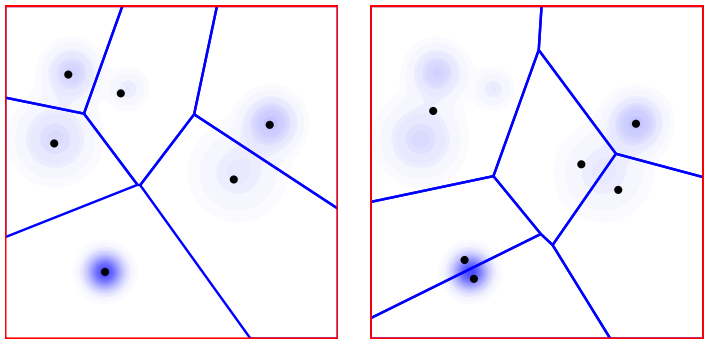


Figure 6.3: A typical run of the limited-range DA algorithm. Agents are labeled in (a)-(f) and the number of agents in formation are contained in parenthesis in (a)-(e). Agent positions at each time step are shown as the small circles. The temperature begins at $T = 20$, then decrease: $T = 13.1$ (a), $T = 12.5$ (b), $T = 2.4$ (c), $T = 2.1$ (d), $T = 0.5$ (e). The final panel (f) shows the result of running the normal gradient-descent coverage starting from the configuration of (e).

configuration only 4 of the 50 tries. The worst simulation converged to a final cost of 18.19, see Figure 6.4. The average final cost of the 50 simulations was 15.4. Comparing the running time, the DA algorithm converged to the final configuration in $133s$ of simulated time, while the normal gradient descent took on average only $18s$ of simulated time to finish.

Now we consider smaller sensing radii to see how the DA algorithm performs. A sensing range of $R = 4$ for each agent was sufficiently large for the DA algorithm to converge to the final configuration shown in Figure 6.5 regardless of initial position. Next, 50 trials of the Lloyd-like gradient descent were computed. The DA algorithm converges to the minimum cost of -277.5 regardless of initial



(a) (b)

Figure 6.4: A comparison between the best-case performance of the Lloyd-like gradient decent (left) with a cost of 10.64, to the worst-case (right) with a cost of 18.19.

position whereas only 3 of the 50 trials of the Lloyd-like gradient descent achieve that cost. The worst case converged to a cost of -270.3 with a mean of -272.3. The cost values are negative since the metric (6.3) has the $-(R^{(i)})^2$ term. When inserted into the distortion function (2.7), this creates large negative quantities. The run time of the DA algorithm was $79s$ compared to the normal gradient descent average run time of $17s$.

We further reduce the sensing radius to $R = 3$, and perform similar trials. Due to the smaller sensing radius, initial conditions begin to influence the outcome of the DA algorithm. For this particular choice of ϕ, we have the two possible outcomes shown in Figure 6.7. The better outcome of the DA attains a final cost of -151.5, while the worse outcome reaches a final cost of -110.5. Over the 50 random Lloyd-like gradient descent simulations, only 2 reach the configuration shown in 6.8, which is also 6.7 (c). The worst case of all 50 trials was a cost of -103.5, with an average of -135.3. The average run time of the DA algorithms was $70s$ compared to the average run time of the normal gradient descent of $17s$.

Further analysis of this scenario, however, demonstrates that the limited-range DA algorithm still has an advantage over a normal gradient descent algorithm. Figure 6.9 shows the set of initial conditions for which the limited-range DA algorithm converges to the best solution. This figure was generated using a find grid of initial positions, and recording the outcome. Note that over half of the

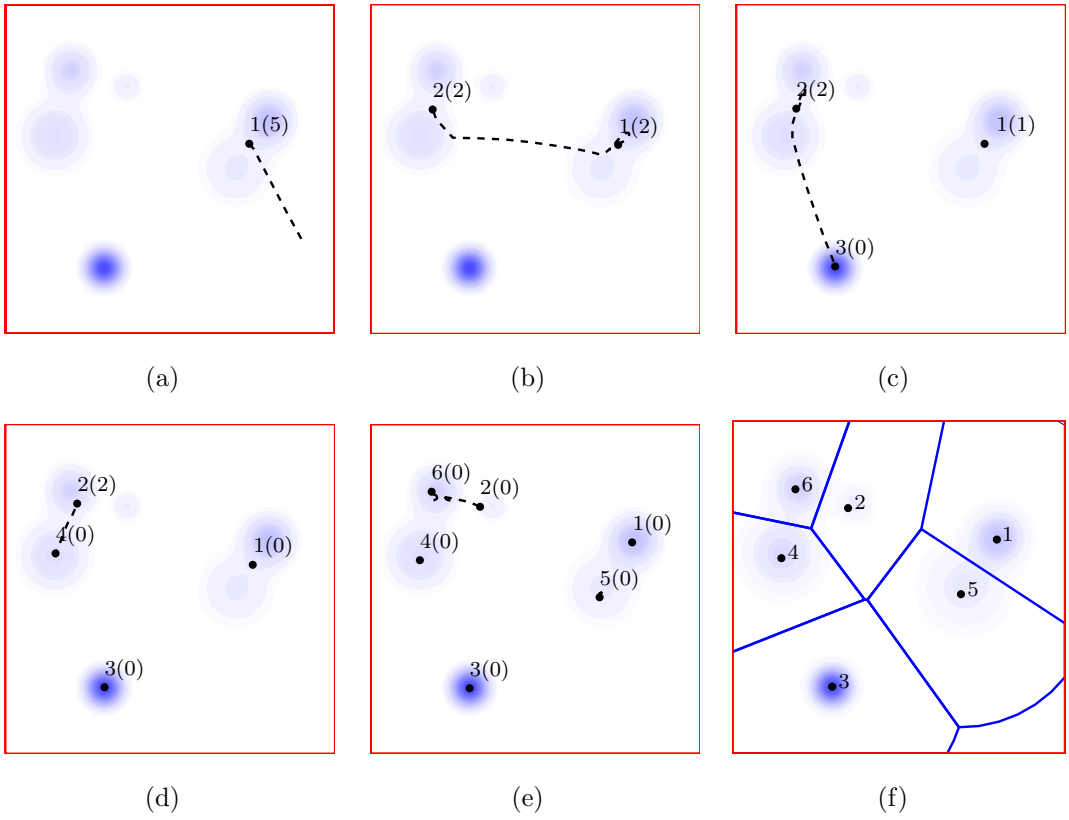
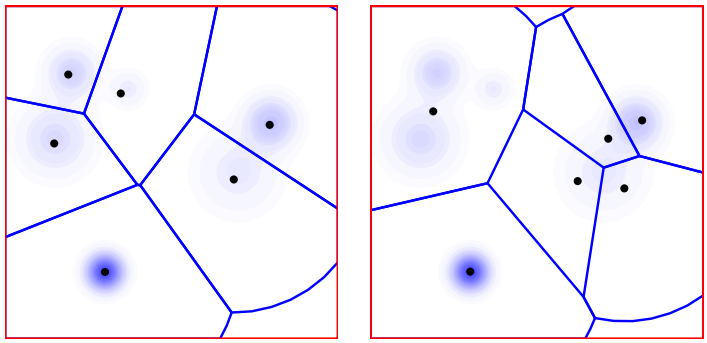


Figure 6.5: A typical run of the limited-range DA algorithm with $R = 4$. Agents are labeled in (a)-(f) and the number of agents in formation are contained in parenthesis in (a)-(e). Agent positions at each time step are shown as the small circles. The temperature begins at $T = 20$, then decrease: $T = 2.2$ (a), $T = 2.1$ (b), $T = 2.0$ (c), $T = 1.9$ (d), $T = 0.1$ (e). The final panel (f) shows the result of running the normal gradient-descent coverage starting from the configuration of (e).

possible initial condition locations leads to the optimal solution while only 4% of the Lloyd-like gradient descent simulations achieved the same final cost.

The limited-range DA algorithm may have decreased performance versus a normal gradient-descent algorithm. If sensing range is not large enough, as was observed in the previous example, the DA algorithm may fall into a local minimum. Consider the distribution shown in Figure 6.10, where there are two equal Gaussians symmetrically placed at opposite corners of Q. Almost every simulation of the limited-range DA algorithm results in a final configurations like 6.10(a), or its mirror image. This occurs because the DA algorithm begins with only one



(a) (b)

Figure 6.6: A comparison between the best-case performance of the Lloyd-like gradient decent (left), to the worst-case (right). The sensing radius for all agents is $R = 4$.

agent, and this agent moves towards the nearest Gaussian that it senses and stays there. Then, future phase changes result in only adding more agents around the same Gaussian.

On the other hand, over 50 trials of the Lloyd-like gradient descent with similar initial conditions as before, we see an improved statistic. Only 18 of the 50 simulations fell into the worst-case minima of Figure 6.10(a). However, none of the simulations were able to converge to the best configuration, which is having 5 agents located around each Gaussian.

A possible way to address this problem of the limited-range DA algorithm is to consider a heating and cooling cycle. Agents can deploy over Q using an area-maximizing technique. Thus, agents will tend to move away from each other and cover all of Q, as shown in Figure 6.11(a). Then, the limited-range DA algorithm is run with a high temperature. This forces agents to collect together about denser parts of Q, shown in Figure 6.11(b)–(c). Finally, the usual limited-range DA coverage is run, causing agents to split evenly over the important areas of Q, as in Figure 6.11(d)–(f). Note, however, that the communication connectivity requirement must be modified so that an agent can communicate with any other agent in Q for this solution to work.

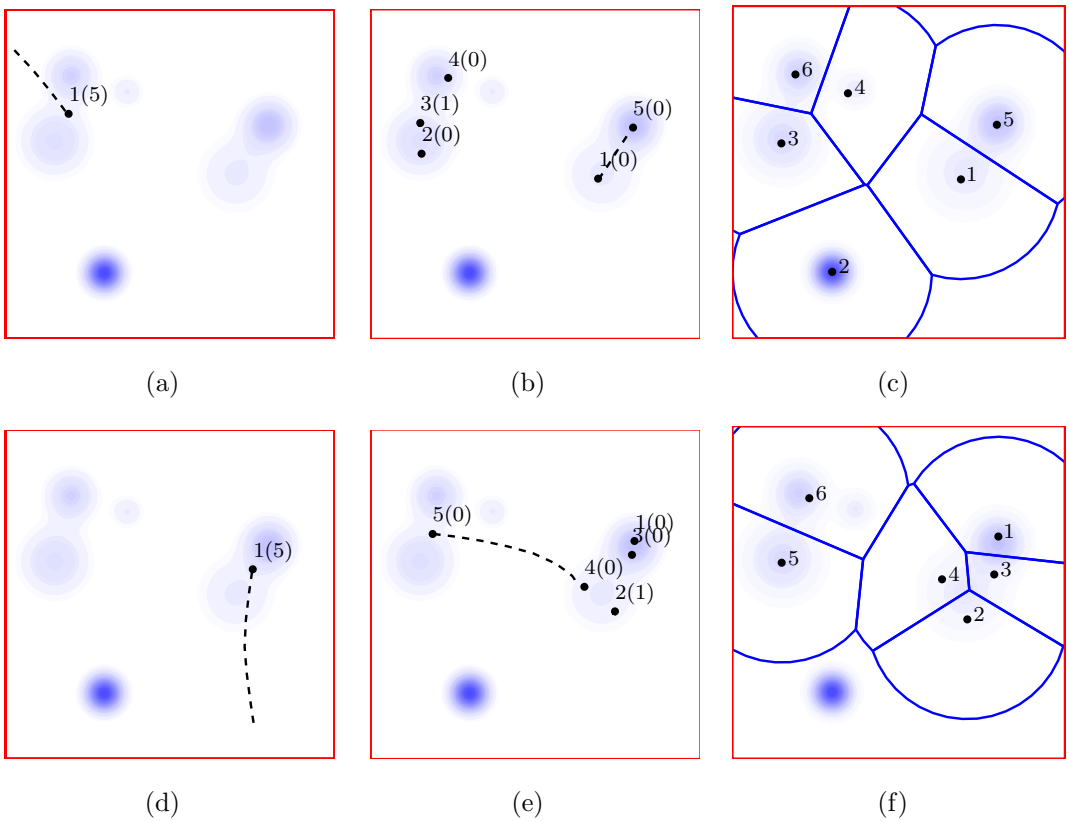


Figure 6.7: Two runs of the limited-range DA algorithm with $R = 3$. In (a)–(c), the temperature begins at $T = 20$ and decreases: $T = 2.2$ (a), $T = 0.8$ (b), with a final configuration in (c). Similarly in (d)–(f), the temperature begins at $T = 20$ and decreases: $T = 2.0$ (d), $T = 0.5$ (e), with a final configuration in (f). Agent positions at each time step are shown as the small circles.

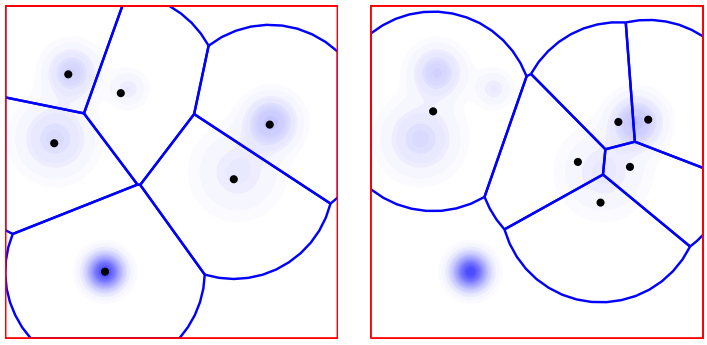


Figure 6.8: A comparison between the best-case performance of the Lloyd-like gradient decent (left), to the worst-case (right). The sensing radius for all agents is $R = 3$.

Algorithm 4: Task assignment algorithm for each agent

$a^{(i)} \leftarrow$ number of agents in formation
if *checkSplit()* == *true* **then**
  if $a^{(i)} == 0$ **then**
    flood("need companion at $p^{(i)}$")
    $M \leftarrow$ positions $p^{(j)}$ of replies for help
    if $m^{(i)} == null, \forall\, m^{(i)} \in M$ **then**
      return
    else
      $J \leftarrow \text{sortAscending}(\{\|p^{(i)} - p^{(j)}\|\}, j \in M)$
      $j^* \leftarrow \text{removeFirst}(J)$
      send("request companion", j^*)
      flood("increment n by 1")
    end
  else
    split formation evenly
    flood("increment n by 1")
  end
else `// no splitting at` $p^{(i)}$
  $M \leftarrow$ received companion requests $p^{(j)}$
  $J \leftarrow \text{sortAscending}(\{\|p^{(i)} - p^{(j)}\|\}, j \in M)$
  if $a^{(i)} == 0$ **then**
    send(null, $\forall\, j \in J$)
  else
    while *length(J)* > 0 **and** $a^{(i)} > 0$ **do**
      $j^* \leftarrow \text{removeFirst}(J)$
      send("help available from $p^{(i)}$", j^*)
      $a^{(i)} \leftarrow a^{(i)} - 1$
    end
  end
end

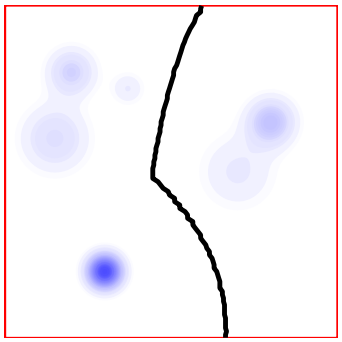
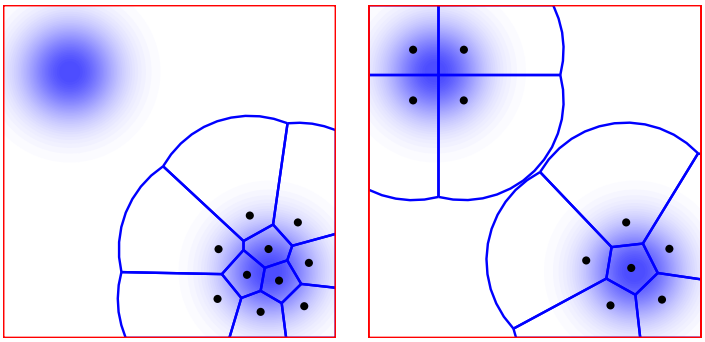


Figure 6.9: DA sensitivity to initial conditions. For initial conditions of the limited-range DA algorithm that start to the left of the thick black line, simulations converge to the optimal solution.



(a) (b)

Figure 6.10: A comparison between the best-case performance of the Lloyd-like gradient decent (right), to the worst-case (left) for the symmetric Gaussian scenario. The sensing radius for all agents is $R = 3$.

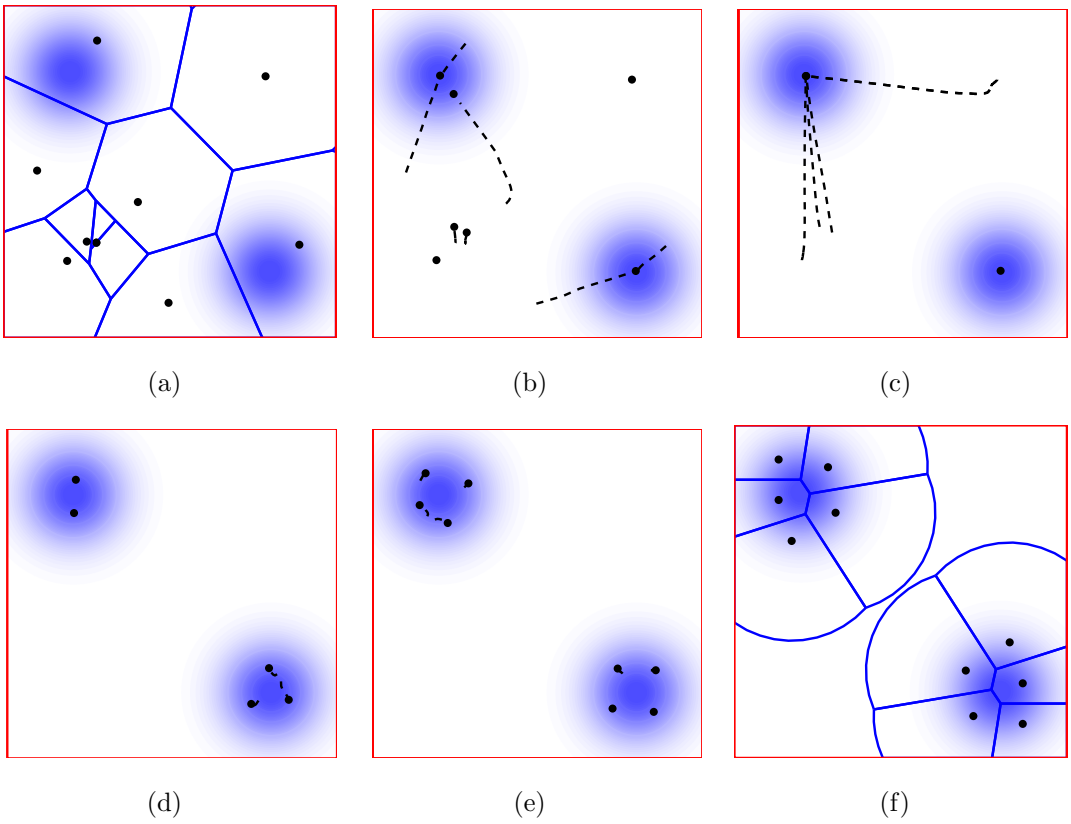


Figure 6.11: A demonstration of a heating and cooling cycle with $R = 3$. Agents perform area-maximizing coverage (a), then agents run the limited-range DA algorithm for high temperature, $T = 20$, (b)–(c), and the trajectories towards the upper-left and lower-right corners are shown as a series of dots. Finally, agents perform the usual limited-range DA algorithm in (d)–(f).

Chapter 7

Robotic Testbed

Thus far, we have outlined several different aspects of cooperative coverage control for mobile vehicles. Each aspect has focused on a particular constraint that may arise in a physical application. To demonstrate the efficacy of those results, we implement some coverage control algorithms in a robotic testbed at the University of California at San Diego. In general, we developed our testbed with with cost-effectiveness and ease of development in mind.

We first describe the organization and interplay between various elements of the experimental set up. This includes the construction of the individual robotic vehicles, the overhead vision system, and the central server that logs data and handles communication between agents. Finally, we conduct an experiment of coverage in the laboratory and present its results.

7.1 Central server

In its current implementation, the testbed is indoors. This scenario has the following implications: (a) individual vehicles cannot rely on GPS for position and orientation information, and (b) due to the small confines of the lab space, inter-vehicle wireless communication will be essentially all-to-all. We utilize a central server to address those issues by simulating a GPS-enabled environment, and also by handling communication between agents. In this way, we are able to simulate deployment with limited-range interactions. We now go in detail to describe the

functions of the central server, a Dell Optiplex 360 running with a 2.53 GHz Intel Core 2 Duo, 2GB RAM, and the Ubuntu 10.04 32-bit operating system.

7.1.1 Overhead vision

We utilize an overhead camera to address the need for accurate position and orientation measurements. The setup consists of a Logitech QuickCam Pro 9000 webcam mounted above and connected to the central server via USB. The server performs image processing on the video feed using Java and computes the position and orientation of all active robots at a data rate of 10Hz. The camera resolution is 640×480 pixels, and that covers a floorspace of $3.6m \times 2.5m$. This results in pixels that correspond roughly to a $5mm \times 5mm$ patch of the lab floor.

To further smooth the data from the overhead camera, we employ a discrete-time Kalman filter.

Assumption 4. *We assume that the model of these robots for the Kalman filter implementation satisfies:*

1. *the amount of time that passes in the lab from one discrete time step to the next is constant, and*

2. *over the span of one time step, the robot will undergo circular motion with constant speed and turning rate.*

In the following, we will develop a linear time-varying model for an agent and use this to create a Kalman filter that handles noisy data from the image processing stage as well as the occasional dropped frame. We will let $p \in \mathbb{R}^2$ denote the position of the agent and $\theta \in \mathbb{S}^1$ denote its orientation. The state of the agent at time step k can be described with the pair (p_k, θ_k).

Under the assumptions stated previously, the position and orientation of the agent at the next time step can be calculated using the current and previous position and orientation. Referencing Figure 7.1, we can see that if a vehicle is constrained to move on a circle with fixed speed, then consecutive positions of the vehicle are equally spaced apart. Furthermore, the location of p_{k+1} is the reflection

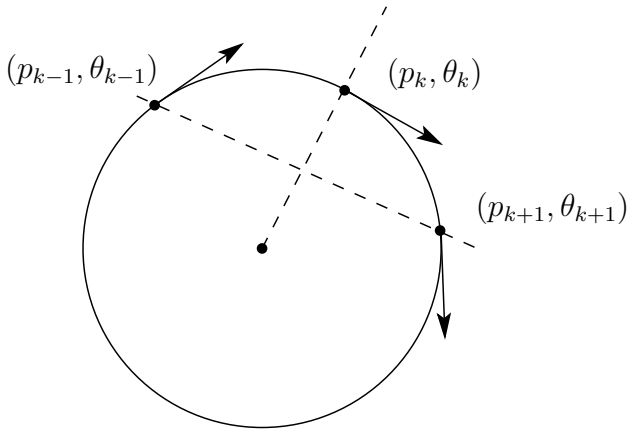


Figure 7.1: Diagram to aid in the derivation of the smoothing Kalman filter.

of p_{k-1} over the axis passing through p_k with an angle of $\theta_k + \frac{\pi}{2}$. To solve for the next position and orientation, we will utilize the rotation matrix

$$R_k = \begin{pmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{pmatrix},$$

along with the reflection matrix

$$T = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The rotation matrix R_k performs a counter-clockwise rotation by an angle of θ_k while T performs a reflection across the vertical axis. The location of the point p_{k+1} is then

$$p_{k+1} = p_k + R_k T R_k^{-1}(p_{k-1} - p_k).$$

Since R_k is orthonormal, $R_k^{-1} = R_k^{\mathsf{T}}$ and

$$p_{k+1} = (I_2 - R_k T R_k^{\mathsf{T}})p_k + R_k T R_k^{\mathsf{T}} p_{k-1}. \tag{7.1}$$

Additionally, since the turning rate is assumed to be constant,

$$\theta_{k+1} = 2\theta_k - \theta_{k-1}. \tag{7.2}$$

Let $\xi_k = (p_k, \theta_k, p_{k-1}, \theta_{k-1})$. Combining (7.1) and (7.2), the discrete-time state

update equation is

$$\underbrace{\begin{bmatrix} p_{k+1} \\ \theta_{k+1} \\ p_k \\ \theta_k \end{bmatrix}}_{\xi_{k+1}} = \underbrace{\begin{bmatrix} I_2 - R_k T R_k^{\mathsf{T}} & 0 & R_k T R_k^{\mathsf{T}} & 0 \\ 0 & 2 & 0 & -1 \\ I_2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{\mathscr{F}_k} \underbrace{\begin{bmatrix} p_k \\ \theta_k \\ p_{k-1} \\ \theta_{k-1} \end{bmatrix}}_{\xi_k} . \tag{7.3}$$

Now we introduce noise into the model. Let y_k be the measured position of an agent, and w_k, v_k be zero-mean Gaussian noise with covariances $\bar{Q}_k$, $\bar{R}_k$, respectively. We assume that there is no cross-correlation between any of the position and orientation components of the sensing and plant noise. Additionally, since the overall state vector ξ_k is contains information from the previous state, there is no additional noise added to the (p_{k-1}, θ_{k-1}) components. The covariance matrices have the following structure:

$$\bar{Q}_k = \begin{bmatrix} \bar{Q}_{p,k} & 0 & 0 \\ 0 & \bar{Q}_{\theta,k} & 0 \\ 0 & 0 & 0 \end{bmatrix} , \quad \bar{R}_k = \begin{bmatrix} \bar{R}_{p,k} & 0 & 0 \\ 0 & \bar{R}_{\theta,k} & 0 \\ 0 & 0 & 0 \end{bmatrix} .$$

We assume that the plant noise is constant in time, $\bar{Q}_k = \bar{Q}$. To account for a dropped frame at the kth time step in the video capturing process, we let $\bar{R}_k$ have arbitrarily large values along its diagonal.

Following the formulation for a Kalman filter, as in [78], we have the following one-step smoothing Kalman filter:

$$\begin{aligned} \hat{\xi}_k &= \mathscr{F}_{k-1}\hat{\xi}_{k-1} + K_k(y_k - \mathscr{F}_{k-1}\hat{\xi}_{k-1}) , \\ P_k &= (I_6 - K_k)(\mathscr{F}_{k-1} P_{k-1} \mathscr{F}_{k-1}^{\mathsf{T}} + \bar{Q}) , \\ K_k &= (\mathscr{F}_{k-1} P_{k-1} \mathscr{F}_{k-1}^{\mathsf{T}} + \bar{Q})(\mathscr{F}_{k-1} P_{k-1} \mathscr{F}_{k-1}^{\mathsf{T}} + \bar{Q} + \bar{R}_k)^{-1} . \end{aligned} \tag{7.4}$$

We implement this for each active robot on the server side in Java to obtain a better estimate of the position and orientation of each robot.

7.1.2 Communication

Communication between robots is handled via the central server. In this way, we can impose range-based communication constraints using information from

the overhead vision system. The actual wireless communication is done over the UCSD wireless network using Java's implementation of TCP/IP sockets. While the TCP/IP link has its own protocol for redundancy and error checking, the actual messages passed via TCP/IP are done in an asynchronous manner.

We choose an asynchronous communication architecture for robust-ness against dropped packets and agent failures. For example, in a synchronous setting if an agent does not respond that it has received a message, the server may wait indefinitely and this will cause the entire cooperative deployment process to halt. Received messages are treated as incoming events that trigger a sequence of actions to be performed locally. Likewise, requests that are sent out over the network do not necessarily result in immediate responses. Locally, an agent will have an event handler in the case of receiving a response at a later time.

We impose communication constraints by having agents send their own communication range to the server. The server referees requests from agents for their neighbors by determining via the vision system if any other agents are in range. If we were to implement an adhoc wireless network, the small size of the lab space would result in having all-to-all communication at all times. The overall testbed architecture is summarized in Figure 7.2.

7.2 Robot description

In this section we will detail the construction and components of the five mobile robots in our testbed. Afterward, we will describe the motion controller that is implemented on each robot.

7.2.1 Components and construction

Mobile robot choices range from taking a complete do-it-yourself standpoint all the way to purchasing a ready-built documented robot with a development kit such as the Khepera III [32]. Those two options represent the two extremes of the developmental ease scale. However, the cost scale is, in general, inversely proportional to ease of development.

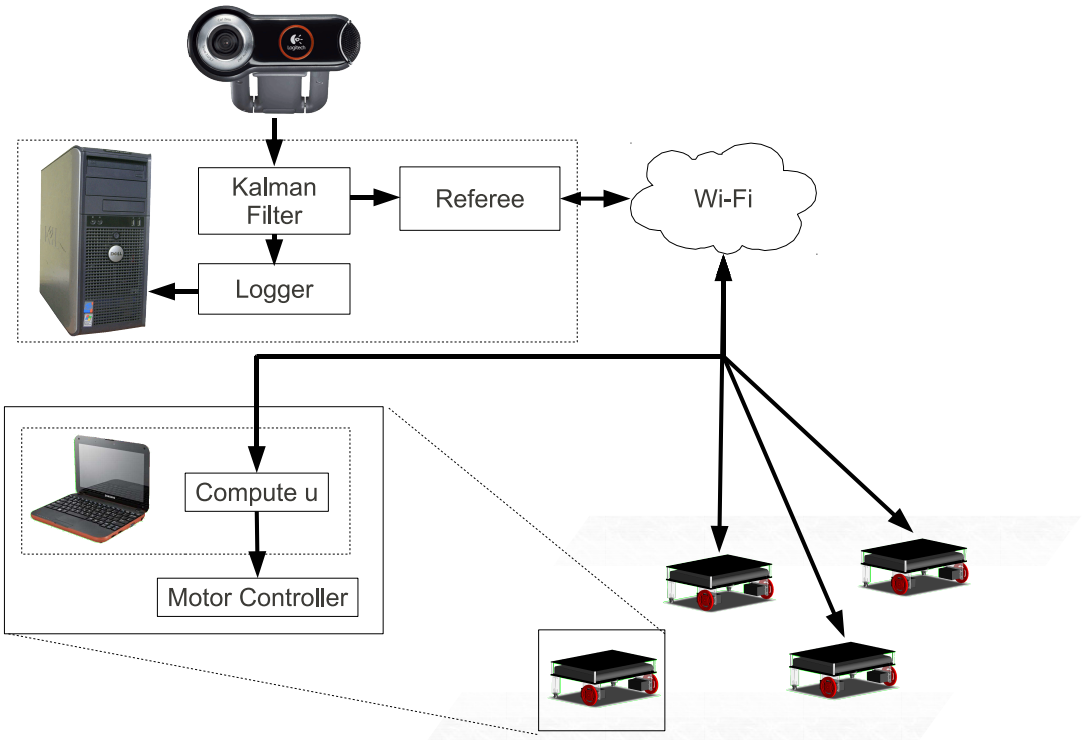


Figure 7.2: Overall testbed architecture and organization.

We proposed a compromise between the two extremes, affording each of our robots significant computational power with a lower cost at the expense of some developmental ease and compactness in size. At the heart of each robot is a Samsung N150 netbook, equipped with an Intel 1.66 GHz N450 atom processor running Windows 7. This netbook provides a relatively compact solution for wireless communication and motion planning that can be done on board.

For motion, we utilize two Hitec HS-755Hb 1/4 Scale servos that are modified for continuous rotation. These two servos are controlled via a Micro Maestro 6-Channel USB Servo Controller from Pololu Robotics and Electronics. While we could have used a pair of DC motors for motion, the required electronics for controlling the motors would have either greatly increased cost or added much development time to the robot construction process.

The complete bill of materials, priced in May 2010 without shipping costs

and tax, is shown in Table 7.1. Certain items such as the black acrylic and the LEDs needed to be purchased in larger quantities. The prices shown in Table 7.1 reflect the cost of just the material used in construction of a single robot.

The larger acrylic pieces were cut down to size using the Lasercamm available in EBUII, Room 315. The rest of the components were wired and assembled in our own lab. The LEDs are used for identification and tracking purposes in conjunction with the overhead camera. Figure 7.4 details the LED arrangement, allowing for overhead position and orientation tracking as well as a unique identification number for the robot ranging from 0 to 26. A Solidworks model of an assembled robot is shown in Figure 7.3 without the LED indicator lights along with a photograph of a robot in the lab.

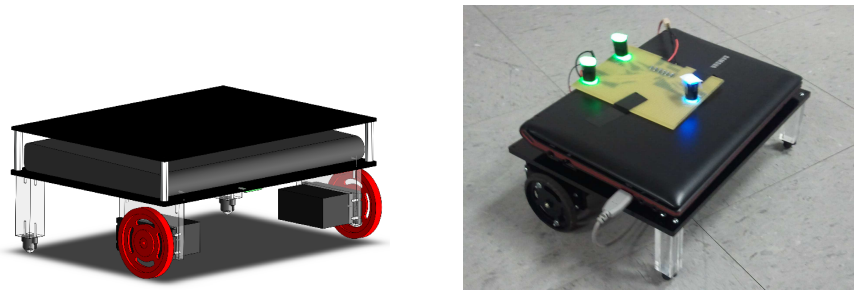


Figure 7.3: A drawing of the robot in Solidworks without the LED indicator lights for the overhead vision system at left and a photograph of a robot in the lab with LED lights.

Using a netbook allows for ease of expanding the capabilities of each robot. The USB interface allows for the addition of any compatible device or sensor. For example, if we wish to perform deployment outdoors, then one can add on a USB GPS device and obtain position and orientation information. One can also have the possibility of placing range finders and other obstacle-avoiding sensors on the robot and interfacing via a A-to-D USB converter.

The only limiting factor for the current implementation is the processor speed of the netbook. However, if that becomes an issue, then obtaining a more powerful laptop in the future will result in very little time needed to transition the software to the new laptop due to the ubiquity of the Java programming language. The current implementation of the robots can be thought of as a concept model for

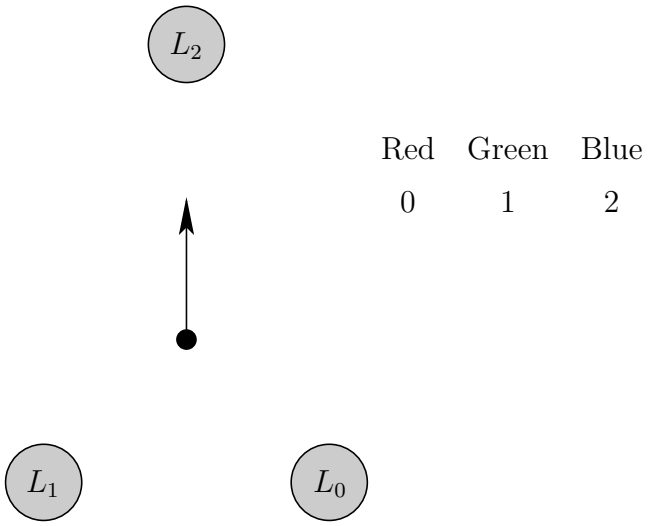


Figure 7.4: Schematic of the LED identification tag that each robot displays. The triangular shape of the three lights gives position and orientation information, indicated by the dot and arrow. The different colors–red, green, and blue–of the LEDs correspond to $0, 1, 2$, respectively. Combined as $L_0 \cdot 3^0 + L_1 \cdot 3^1 + L_2 \cdot 3^2$, this gives each robot a unique identification number ranging from 0 to 26.

distributed deployment. Indeed, with some modifications, these robots may be able to operate outside without the need for simulated GPS or simulated communication constraints.

7.2.2 Motion controller

The dynamics of each robot are best approximated by the equations of motion of (4.1), repeated here:

$$\begin{aligned} \dot{p}_1^{(i)} &= v^{(i)} \cos\theta^{(i)} \,, \\ \dot{p}_2^{(i)} &= v^{(i)} \sin\theta^{(i)} \,, \\ \dot{\theta}^{(i)} &= \omega^{(i)} \,. \end{aligned}$$

Therefore, in order to navigate to any given waypoint, $C^{(i)}$, we implemented the motion controller described in equations (4.6)–(4.9). Specifically, if the desired

waypoint $C^{(i)}$ is in front of agent i, then that agent will use the dynamics given by

$$\begin{aligned} v^{(i)} &= k_p \operatorname{sat}(\|C^{(i)} - p^{(i)}\|)\,, \\ \omega^{(i)} &= \frac{2\omega_{\max}}{\pi}\Omega^{(i)}\,, \end{aligned} \tag{7.5}$$

where $\Omega^{(i)} = \angle(C^{(i)} - p^{(i)}) - \theta^{(i)}$ and $\Omega^{(i)} \in [-\pi, \pi]$. If $C^{(i)}$ is behind the vehicle, then $|\Omega^{(i)}| > \frac{\pi}{2}$ and the robot will turn in place according to

$$\begin{aligned} v^{(i)} &= 0\,, \\ \omega^{(i)} &= \frac{2\omega_{\max}}{\pi}\operatorname{sgn}(\Omega^{(i)})\,. \end{aligned} \tag{7.6}$$

We allow the agent to come to a complete stop when it is within 5 cm of the desired waypoint.

7.3 Experimental results

In this section, we demonstrate various deployment scenarios with the robots in the testbed. The first one is an application of the nonholonomic coverage algorithm described in Chapter 4 for vehicles with variable forward velocities. Next, we will demonstrate power-limited deployment.

We will demonstrate two examples of nonholonomic deployment. For the first example, we wish to have the robots deploy into a circular formation. To achieve this, we choose the following density function

$$\phi(q) = 5\exp\left[-\frac{1}{\sigma^2}\left(\|q - q_0\| - r\right)^2\right]\,, \tag{7.7}$$

with $r = 150$, $q_0 = (240, 320)^{\mathsf{T}}$, and $\sigma = 25$. The agents employ the motion controller described in the previous section and use the positions of themselves and their neighbors to compute the centroid of their Voronoi regions. The deployment process was logged by the central server and the data is presented in Figure 7.5. The agents successfully deploy to centroidal configurations and they form a circle. The corresponding time evolution of cost function is also shown in the figure and it is indeed minimized. A picture of the robots in the final configuration is shown in Figure 7.6.

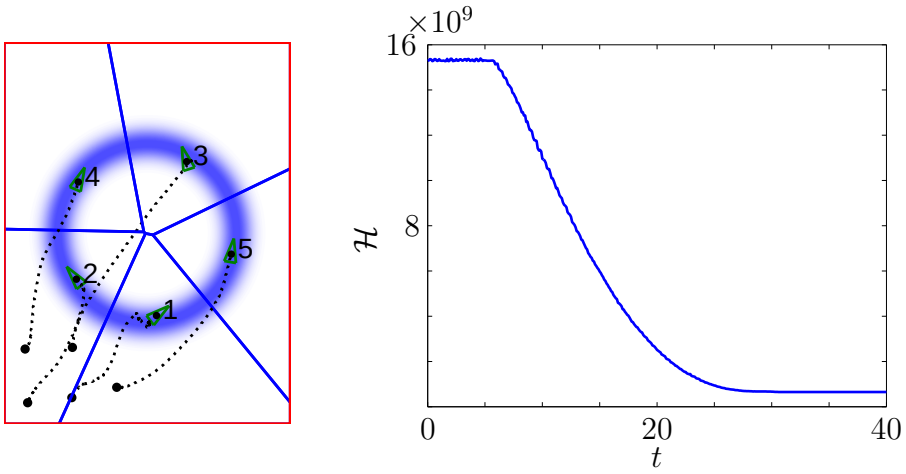


Figure 7.5: An experimental run with a circular density function. Agent initial positions are denoted as the black dots in the left figure, and their final positions and orientations are shown as the green triangles. The data was taken from the overhead vision system and the corresponding cost function is shown at right.

In the second example of nonholonomic deployment, we choose the density function ϕ to be a sum of four Gaussian peaks with different amplitudes and variances. We also enforce a sensing range of 200 pixels, which is approximately $1m$. Experimental results are shown in Figure 7.7 along with the objective function evolution. Note that in this case, as in Chapter 3, we are maximizing the objective function. A picture of the robots in the final configuration is shown in Figure 7.8.

Next, we test the power-limited deployment algorithm for the MWVD described in Chapter 3 using the same ϕ as the previous experiment. We initialize the agents with $E^{(i)} = 500$ for all $i \in until5$. Energy use is simulated locally at each robot by subtracting an amount of energy proportional to the commanded servo actuation speed at each time step. The results are shown in Figure 7.9 along with a picture of the final deployment configuration. Final energy values were $E^{(1)} = 222$, $E^{(2)} = 293$, $E^{(3)} = 143$, $E^{(4)} = 216$, and $E^{(5)} = 293$. Unfortunately, due to the asynchronous nature of the testbed communication, we cannot display a plot of the cost function as this relies on the knowledge of agent energies at all times.

Finally, we demonstrate a DA deployment for the five robots. We choose

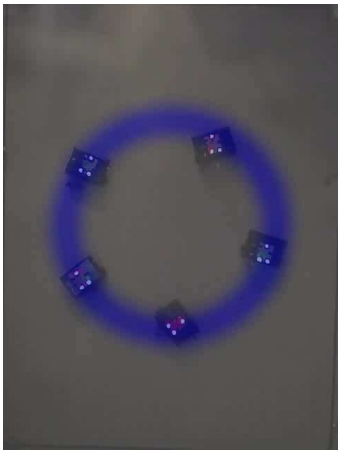


Figure 7.6: Screen capture of the limited-range nonholonomic deployment with the density function overlaid on top.

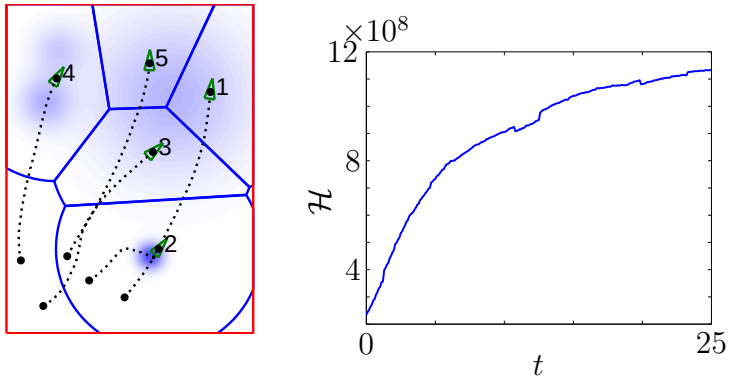


Figure 7.7: Limited-range nonholonomic deployment for a collection of Gaussian peaks. Note that the objective function is maximized in this case.

ϕ to be the sum of five Gaussian peaks with varying amplitudes and variances. Rather than moving in formation, we implement the version of distributed DA deployment described in Remark 61. Figure 7.10 displays experimental data from the entire run along with a picture of the final agent configurations in the testbed. Each plot details the paths of all active robots, along with the positions and orientations of the inactive robots located at the bottom area of each plot. There were a total of five plots corresponding to the four phase changes that occurred for this particular choice of ϕ.

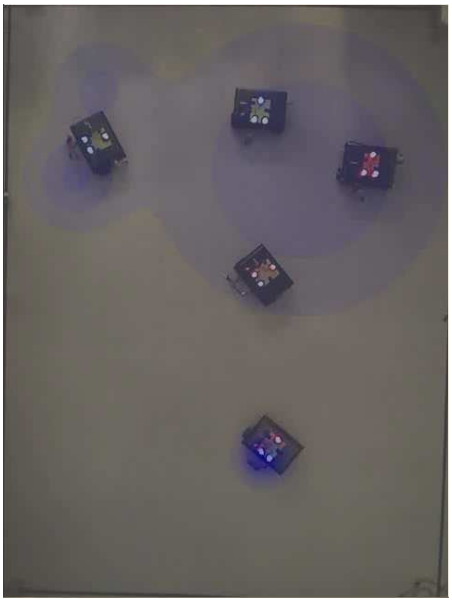


Figure 7.8: Screen capture of the limited-range nonholonomic deployment with the density function overlaid on top.

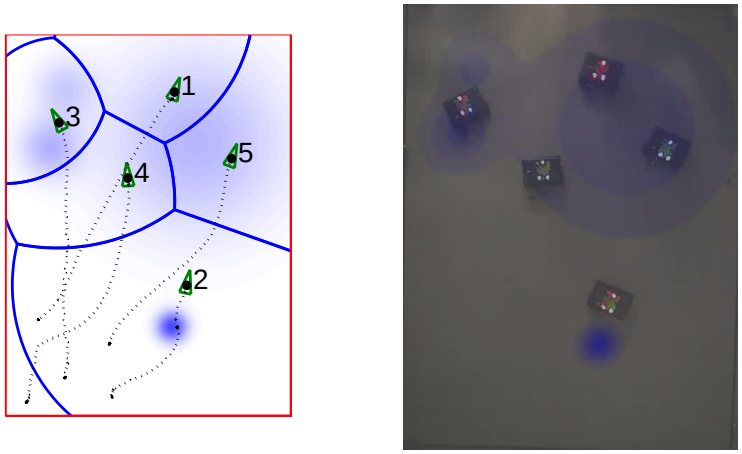


Figure 7.9: Power-limited MW deployment in the testbed. Data and path information from the vision system is shown at left and a screen capture of the final configuration is shown with the density overlaid on the right. Initial energy levels were $E^{(i)} = 500$ for all $i \in \{1, \ldots, 5\}$. Final energy values were $E^{(1)} = 222$, $E^{(2)} = 293$, $E^{(3)} = 143$, $E^{(4)} = 216$, and $E^{(5)} = 293$.

Table 7.1: Robot bill of materials, May 2010

Amazon	
Samsung N150 Netbook	279.99
McMaster-Carr	
11" x 8" x 0.236" Black Acrylic	5.75
11" x 8" x 0.117" Black Acrylic	4.08
12" x 0.75" x 0.75" Acrylic	3.04
4 x 1.5" Standoffs	2.60
6 x 0.25" Standoffs	2.52
Pololu	
2 5/8" tires	3.50
Ball Casters	5.98
Servo Controller	24.95
Fry's Electronics	
3-Pin Molex male to female cable	2.99
2-Pin male to 2-pin female cable	1.49
6 x 470 ohm resistors	0.48
5V voltage regulator	1.79
4" x 5" prototyping board	4.99
Digikey	
6 x LED	1.58
RC Planet	
2 x Hitec HS-755Hb 1/4 Scale Servo	57.98
onlybatterypacks.com	
7.2V Team Losi battery pack	21.50
UCSD Bookstore	
1ft Mini USB cable	2.99
Radioshack	
SPDT Switch	3.79
Total	$431.99

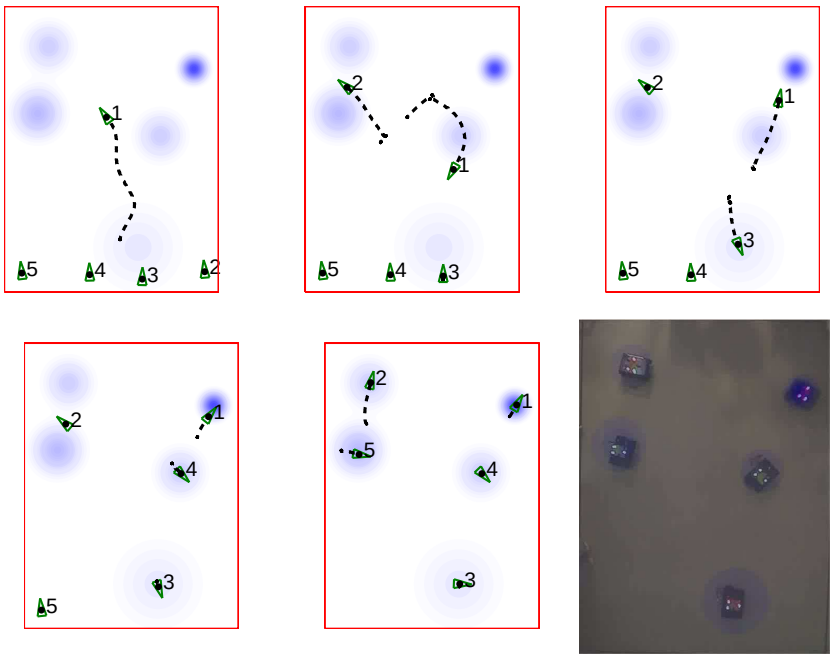


Figure 7.10: DA deployment in the testbed

Chapter 8

Closing remarks

We now offer some concluding remarks and thoughts for future extensions that may be taken from this dissertation. The topic of cooperative motion control is a natural consideration with the increasing use of single autonomous vehicles. This dissertation has addressed a small subset of problems that must be considered for certain real-world applications. Still, these issues and many others must be taking into account before a reliable implementation of a cooperative multi agent system can occur. The ideas contained within this dissertation may have employed relatively simple models of vehicles, but the successful results using these models shows promise for applications involving much more complex vehicle models.

8.1 Review of results

In Chapter 3, we presented a novel set of spatially distributed coverage control algorithms that incorporate energy dynamics in order to obtain regions of guaranteed reachability. This led to the use of limited-range generalized Voronoi partitions instead of the typical Voronoi partitions. We designed four objective functions to demonstrate the usage of these generalized Voronoi partitions in various coverage tasks. We have shown through simulation and through experimental results that the cases we developed perform as intended.

Next, Chapter 4 introduced two systems of wheeled vehicles with unicycle dynamics undergoing deployment. We have also shown that the nonholonomic

vehicles converge to centroidal configurations via a hybrid invariance principle. The hybrid systems framework has the potential to allow for more modes of operation. For example, additional vehicle modes may also be introduced to handle obstacle or collision avoidance as well. Even though we performed our analysis using regular Voronoi regions, a very similar convergence result should apply for different spatial tessellations. Other partitions, such as the range-limited partitions from [15] or more general Voronoi partitions like those found in [38], should not affect the convergence of wheeled vehicles.

Chapter 5 addressed the problem of area coverage maximization for agents moving in a fast flow environment. The inability to move against the flow presented many interesting challenges, and the resulting set of possible time-optimal trajectories were studied. We limited the set of flow environments to affine flow fields and piecewise constant flow fields in order to have analytic results. However, the methods presented in this chapter may be applied to more general flow fields using numerical methods to compute optimal trajectories and gradient directions. Another open question is in the appropriate choice of the coverage horizon T. This choice is largely dependent on the communication power of the agents, however, it may be useful from an energy-saving perspective to consider a smaller coverage horizon. Possible directions of future work include consideration of objective functions other than the total reachable area. This may be more suitable for sampling a plume or chemical compound present in the flow environment. In this scenario, it may be more advantageous for agents to move toward areas of higher chemical concentrations, or to plan paths that maximize the amount of information gained from sampling in order to construct an accurate estimate of the chemical field.

Rather than focusing on dynamic constraints as in the previous three chapters, Chapter 6 introduced a limited-range and distributed implementation of the DA algorithm developed by Rose, and applied it to the coverage problem. We developed limited-range results that extend those in [68] and [77]. When the sensing radius is as large as the diameter of Q, this algorithm becomes the normal DA algorithm of Rose. While the limited-range DA algorithm is able to outperform a Lloyd-like gradient descent algorithm in many cases, the algorithm has its limita-

tions. As noted in simulations, final configurations may not be optimal as sensing range decreases. Consideration of a heating and cooling cycle produces improved results, but it is still an ad hoc solution to the underlying problem. Additionally, the running time of the DA algorithm is longer than that of normal gradient descent. This is because a gradient descent must be repeatedly performed after each phase change until all N agents are active. Compare this to normal gradient descent algorithms where all N agents are active from the beginning and only one round of gradient descent is performed. Thus usage of the DA algorithm is a trade off between fast convergence and improved optimality of final configurations and improved insensitivity to initial positions.

Finally, Chapter 7 describes a robotic testbed constructed at UCSD that demonstrates many of the deployment and coverage algorithms described in this dissertation. We described the construction process and the overall architecture of the various systems that comprise the testbed: the central server, the overhead vision, and the individual robotic agents. Experimental data representative of the power-aware, unicycle, and deterministic annealing coverage algorithms were presented. The robotic agents are nonholonomic by construction, so we have shown via implementation that algorithms from Chapter 3 as well as from Chapter 6 can be implemented with such dynamics without loss of convergence results. Due to the unique nature of coverage in a flow, we could not implement the results from Chapter 5 in our testbed.

8.2 Future directions

With respect to dynamically constrained coverage, several questions and interesting extensions remain. While the deployment algorithms presented in this dissertation have focused on individual dynamic constraints, the question remains if it is true that a combination of the results presented still converge to e.g. centroidal configurations. Experimental results from the testbed seem to indicate this fact. It should be straightforward to verify this desired convergence result using the methods described in this work. In what follows, however, we will describe

other less obvious extensions motivated by the dynamically constrained deployment problems addressed in this dissertation.

8.2.1 Efficient computation of Voronoi diagrams in a flow

The flow coverage problem addressed in Chapter 5 presents a method to maximize the area of reachable points in the flow environment. A crucial step in doing this is computation of the reachable set of points for each agent. A related problem is computing the so-called Zermelo-Voronoi diagram (ZVD) for the group of agents in the flow environment. For a particular agent i in the flow environment, the Zermelo-Voronoi region of agent i is defined as the set of points that i can reach sooner than any other agent j in the flow environment. Much work has been done describing these regions as well as developing efficient methods for their computation, cf. [79, 55].

A notable property of the ZVD for slow flows–that is when the available vehicle actuation exceeds the maximum flow speed–is that there exists a one-to-one correspondence between the ZVD and the normal Voronoi partition. This has motivated the work of Bakolas and Tsiotras [2]. However, the assumption that the flow is slow is crucial in the formulation of the map from the ordinary Voronoi partition to the ZVD and vice-versa. A similar map should be possible for the case of a spatially constant fast flow field, as in the ones studied in Chapter 5. However, the ZVD for a fast flow should map to something that is different from the normal Voronoi diagram. This is due to the fact that there are directions of travel in the fast flow that are "redundant" or sub-optimal. Thus, exclusion of these directions in the transformed flow-less frame should result in a similar bijective map to the one described in [2].

8.2.2 Coverage on general manifolds

Whereas the topics described in this dissertation deal with constraints that enter through the dynamics of vehicles, a relevant variation involves constraints entering in the topology of the vehicle's operating environment. Specifically, the

agents could be constrained to move along the surface of a sphere, $p^{(i)} \in \mathbb{S}^2$ for all i. Several issues stemming from the topology of the sphere include proper choice of coordinate frame and the notion of centroidal coverage. Interesting applications for a result regarding spherical coverage involve the use of UAVs or UUVs over a large portion of the Earth. Thus, curvature of the Earth becomes a factor, and it should be verified that coverage still works properly in this topology.

There have been many promising results regarding spherical motion control and Voronoi partitioning on a sphere. Regarding cooperative motion on a spherical topology, we have results such as [58, 28, 87]. However, the particular problem of deployment remains to be addressed, even though it may seem obvious. A promising result for the coverage problem is presented by Buss and Fillmore in [9] where they address the problem of determining the weighted average of discrete points on a sphere. This weighted average point is also constrained to lie on the sphere, and Buss and Fillmore introduce an efficient method to compute this particular average.

What remains to be done is to extend the result of Buss and Fillmore to continuous distributions over the sphere. They demonstrate that the spherical averaging problem can be posed as a (convex) optimization problem whose cost function appears very similar to the coverage cost functions found throughout this dissertation. One should be able to show the analogous results for a continuous distribution on a sphere. With this information in hand, along with results regarding Voronoi diagrams on a sphere [56], there remains the question of a proper motion control algorithm for agents on a sphere.

8.2.3 Coverage on a graph

Perhaps an even more extreme case of topological constraint enters when agents wish to perform deployment on a discrete graph. The connection between this particular constraint and real-time coverage is motivated by Tabuada and his work regarding the decomposition of the state space into a graph. Transitions between nodes of this graph are determined by the underlying dynamics of the continuous-time system [80]. The idea of coverage on a graph can also be used in

the context of general optimization as well. A recent problem has been introduced by Patterson [62] regarding optimal placement of leader nodes in a network of nodes performing noisy average consensus. In this preliminary work, some methods have been introduced to solve this hard combinatorial problem. However, it may be possible to approach this from a coverage viewpoint.

In both cases of deployment over a graph, several important issues must be solved. A notion of distance is a central assumption in the deployment algorithms described in this dissertation. However, on a discrete graph, one must come up with a similar notion, for example hop count. Additionally, due to the discrete nature of the graph, the convergence results established for continuous deployment may not hold anymore. It may be possible for agents to follow a "gradient descent" algorithm over a graph and end up in a cycle of configurations instead of converging to a fixed set of nodes on the graph. These issues, and many more, present nontrivial challenges to this particular problem. However, deriving results in with this particular constraint would indeed be useful.

Bibliography

[1] A. Astolfi, "Exponential stabilization of a wheeled mobile robot via discontinuous control," *ASME Journal on Dynamic Systems, Measurement, and Control*, vol. 121, no. 1, pp. 121–127, 1999.

[2] E. Bakolas and P. Tsiotras, "The zermelovoronoi diagram: A dynamic partition problem," *Automatica*, vol. 46, no. 12, pp. 2059–2067, Dec 2010.

[3] W. Basener, B. P. Brooks, and D. Ross, "The Brouwer Fixed Point Theorem applied to rumour transmission," *Applied Mathematics Letters*, vol. 19, no. 8, pp. 841–842, August 2006.

[4] R. W. Beard and T. W. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *IEEE Conf. on Decision and Control*, Maui, Hawaii, Dec. 2003, pp. 25–30.

[5] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 865–875, 2004.

[6] M. S. Branicky, "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 475–482, 1998.

[7] A. E. Bryson and Y. Ho, *Applied Optimal Control*. Blaisdell Publishing Company, 1969.

[8] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks*, ser. Applied Mathematics Series. Princeton University Press, 2009, available at http://www.coordinationbook.info.

[9] S. Buss and J. Fillmore, "Spherical averages and applications to spherical splines and interpolation," *ACM Transactions on Graphics*, vol. 20, no. 2, pp. 95–126, Apr 2001.

[10] M. Campbell, M. Egersdedt, J. How, and R. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," *Philosophical*

Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 368, no. 1928, pp. 4649–4672, Oct 2010.

[11] N. Ceccarelli, J. Enright, E. Frazzoli, S. Rasmussen, and C. Schumacher, "Micro uav path planning for reconnaissance in wind," in *American Control Conference*, July 2007, pp. 5310–5315.

[12] N. Ceccarelli, M. D. Marco, A. Garulli, and A. Giannitrapani, "Collective circular motion of multi-vehicle systems with sensory limitations," in *44th IEEE Conference on Decision and Control, and European Control Conference*, 2005, pp. 740–745.

[13] J. Cortés, "Finite-time convergent gradient flows with applications to network consensus," *Automatica*, vol. 42, no. 11, pp. 1993–2000, November 2006.

[14] J. Cortés, "Discontinuous dynamical systems – a tutorial on solutions, nonsmooth analysis, and stability," *IEEE Control Systems Magazine*, vol. 28, no. 3, pp. 36–73, 2008.

[15] J. Cortés, S. Martínez, and F. Bullo, "Spatially-distributed coverage optimization and control with limited-range interactions," *ESAIM: Control, Optimisation & Calculus of Variations*, vol. 11, pp. 691–719, 2005.

[16] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[17] J. Desai, J. P. Ostrowski, and V. J. Kumar, "Control of formations for multiple robots," in *IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 2864–2869.

[18] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, Nov 2006.

[19] Q. Du, M. Emelianenko, and L. Ju, "Convergence of the lloyd algorithm for computing centroidal voronoi tessellations," *SIAM Journal on Numerical Analysis*, vol. 44, no. 1, pp. 102–119, 2006.

[20] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.

[21] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[22] D. Estrin, D. Culler, K. Pister, and G. Sukhatme, "Connecting the physical world with pervasive networks," *IEEE Pervasive Computing*, vol. 1, no. 1, pp. 59–69, 2002.

[23] A. Fagiolini, M. Pellinacci, G. Valenti, G. Dini, and A. Bicchi, "Consensus-based distributed intrusion detection for multi-robot systems," in *IEEE International Conference on Robotics and Automation*, Pasadena, CA, May 2008, pp. 120–127.

[24] H. Flanders, "Differentiation under the integral sign," *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, June 1973.

[25] R. Goebel, J. P. Hespanha, A. R. Teel, C. Cai, and R. G. Sanfelice, "Hybrid systems: generalized solutions and robust stability," in *IFAC Symposium on Nonlinear Control Systems*, Stuttgart, Germany, 2004, pp. 1–12.

[26] H. Hamilton, "Navigation and control problems for classes of micro air vehicles," in *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, Keystone, CO, Aug 2006.

[27] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 35, no. 1, pp. 78–92, January 2005.

[28] S. Hernandez and D. A. Paley, "Stabalization of collective motion in a time-invariant flow field on a rotating sphere," in *American Control Conference*, St. Louis, MO, June 2009, pp. 623–628.

[29] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem," in *International Conference on Distributed Autonomous Robotic Systems*, Fukuoka, Japan, Jun. 2002, pp. 299–308.

[30] I. I. Hussein and D. M. Stipanovič, "Effective coverage control for mobile sensor networks with guaranteed collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, 2007.

[31] T. Hfera, J. Sherrattb, and P. Mainia, "Cellular pattern formation during dictyostelium aggregation," *Physica D: Nonlinear Phenomena*, vol. 85, no. 3, pp. 425–444, Aug 1995.

[32] (2010, Nov) Khepera iii. K-Team. Website. [Online]. Available: http://www.k-team.com/mobile-robotics-products/khepera-iii

[33] D. Kagan, P. Calvo-Marzal, S. Balasubramanian, S. Sattayasamitsathit, K. Manesh, G. Flechsig, and J. Wang, "Chemical sensing based on catalytic

nanomotors: Motion-based detection of trace silver," *Journal of the American Chemical Society*, vol. 131, no. 34, pp. 12 082–12 083, Aug 2009.

[34] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[35] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[36] A. Kwok and S. Martínez, "Energy-balancing cooperative strategies for sensor deployment," in *IEEE International Conference on Decision and Control*, New Orleans, USA, December 2007.

[37] ——, "A coverage algorithm for drifters in a river environment," in *2010 American Control Conference*, 2010, 6436–6441.

[38] ——, "Deployment algorithms for a power-constrained mobile sensor network," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 7, pp. 725–842, 2010.

[39] ——, "Deployment of drifters in a piecewise-constant flow environment," in *IEEE International Conference on Decision and Control*, 2010, to appear.

[40] N. E. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. M. Fratantoni, and R. Davis, "Collective motion, sensor networks and ocean sampling," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, 2007.

[41] W. Li and C. G. Cassandras, "Distributed cooperative coverage control of sensor networks," in *IEEE International Conference on Decision and Control*, December 2005, pp. 2542–2547.

[42] D. Liberzon, *Switching in Systems and Control*, ser. Systems & Control: Foundations & Applications. Birkhäuser, 2003.

[43] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982, presented as Bell Laboratory Technical Memorandum at a 1957 Institute for Mathematical Statistics meeting.

[44] J. Lygeros, K. H. Johansson, S. N. Simić, J. Zhang, and S. S. Sastry, "Dynamical properties of hybrid automata," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 2–17, 2003.

[45] N. A. Lynch, *Distributed Algorithms*. Morgan Kaufmann, 1997.

[46] J. Marsden and A. Tromba, *Vector Calculus*, 5th ed. W. H. Freeman, 2003.

[47] J. A. Marshall, M. E. Broucke, and B. A. Francis, "Formations of vehicles in cyclic pursuit," *IEEE Transactions on Automatic Control*, vol. 49, no. 11, pp. 1963–1974, 2004.

[48] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli, "On synchronous robotic networks – Part II: Time complexity of rendezvous and deployment algorithms," *IEEE Transactions on Automatic Control*, vol. 52, no. 12, pp. 2214–2226, 2007.

[49] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems Magazine*, vol. 27, no. 4, pp. 75–88, Aug 2007.

[50] T. G. McGee, , and J. K. Hendrick, "Path planning and control for multiple point surveillance by an unmanned aircraft in wind," in *American Control Conference*, June 2006, pp. 4261–4266.

[51] T. G. McGee, S. Spry, and J. K. Hendrick, "Optimal path planning in a constant wind with a bounded turning rate," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2005, pp. 1–11.

[52] Y. Mei, Y. Lu, Y. Hu, and C. Lee, "Deployment of mobile robots with energy and timing constraints," *IEEE Transactions on Robotics and Automation*, vol. 22, no. 3, pp. 507–522, June 2006.

[53] S. Mohapatra, N. Dutt, A. Nicolau, and N. Venkatasubramanian, "DYNAMO: A cross-layer framework for end-to-end QoS and Energy Optimization in Mobile Handheld Devices," *IEEE J. Selected Areas in Communication*, May 2007.

[54] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[55] T. Nishida and K. Sugihara, "Approximation of the boat-sail voronoi diagram and its application," in *Int'l Conf. on Computational Science and Its Applications*, Assisi, Italy, May 2004, pp. 227–236.

[56] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed., ser. Wiley Series in Probability and Statistics. John Wiley, 2000.

[57] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[58] D. Paley, "Stabilization of collective motion on a sphere," *Automatica*, vol. 45, no. 1, pp. 212–216, Jan 2009.

[59] D. A. Paley and C. Peterson, "Stabilization of collective motion in a time-invariant flowfield," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 32, no. 3, pp. 771–779, 2009.

[60] L. Pallottino, V. G. Scordio, E. Frazzoli, and A. Bicchi, "Decentralized cooperative policy for conflict resolution in multi-vehicle systems," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1170–1183, 2007.

[61] L. Pallottino, V. Scordio, A. Bicchi, and E. Frazzoli, "Decentralized cooperative policy for conflict resolution in multivehicle systems," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1170–1183, December 2007.

[62] S. Patterson and B. Bamieh, "Leader selection for optimal formation coherence," in *IEEE International Conference on Decision and Control*, Atlanta, GA, Dec 2010.

[63] L. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. Mesquita, and G. Pereira, "Simultaneous coverage and tracking (SCAT) of moving targets with robot networks," in *Workshop on Algorithmic Foundations of Robotics*, December 2008.

[64] D. Pines and F. Bohorquez, "Challenges facing future micro-air-vehicle development," *AIAA Journal of Aircraft*, vol. 43, no. 2, pp. 290–305, Mar 2006.

[65] S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, May 2004, pp. 165–172.

[66] V. Raghunathan, C. Pereira, M. Srivastava, and R. Gupta, "Energy-aware wireless systems with adaptive power-fidelity tradeoffs," *IEEE Trans. Very Large Scale Integration Systems*, vol. 13, no. 2, pp. 211–225, February 2005.

[67] J. Reif and Z. Sun, "Movement planning in the presence of flows," *Algorithmica*, vol. 39, no. 2, pp. 127–153, June 2004.

[68] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 80, no. 11, pp. 2210–2239, 1998.

[69] D. Rudnick, R. Davis, C. Eriksen, D. Fratantoni, and M. Perry, "Underwater gliders for ocean research," *Marine Technology Society Journal*, vol. 38, no. 2, pp. 73–84, 2004.

[70] R. Rysdyk, "Course and heading changes in significant wind," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 30, no. 4, pp. 1168–1171, 2007.

[71] R. G. Sanfelice and E. Frazzoli, "On the optimality of Dubins paths across heterogeneous terrain," in *Hybrid systems: Computation and Control*, 2008, pp. 457–470.

[72] R. G. Sanfelice, R. Goebel, and A. R. Teel, "Results on convergence in hybrid systems via detectability and an invariance principle," in *American Control Conference*, 2005, pp. 551–556.

[73] K. Savla, E. Frazzoli, and F. Bullo, "Traveling salesperson problems for the Dubins vehicle," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1378–1391, jULY 2008.

[74] M. Schwager, J. McLurkin, and D. Rus, "Distributed coverage control with sensory feedback for networked robots," in *Proceedings of Robotics: Science and Systems*, August 2006.

[75] M. Schwager, D. Rus, and J. Slotine, "Decentralized, adaptive coverage control for networked robots," *International Journal of Robotics Research*, 2009, submitted.

[76] U. Serres, "On the curvature of two-dimensional optimal control systems and zermelos navigation problem," *Journal of Mathematical Sciences*, vol. 135, no. 4, pp. 3224–3243, 2006.

[77] P. Sharma, S. Salapaka, and C. Beck, "A scalable deterministic annealing algorithm for resource allocation problems," in *American Control Conference*, June 2006, pp. 3092–3097.

[78] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience, 2006.

[79] K. Sugihara, "Voronoi diagrams in a river," Purdue University, Tech. Rep. CSD-TR-1052, Dec 1990.

[80] P. Tabuada, "An approximate simulation approach to symbolic control," *IEEE Transactions on Automatic Control*, vol. 53, no. 6, pp. 1406–1418, Jul 2008.

[81] L. Techy, D. Smale, and C. Woolsey, "Coordinated aerobiological sampling of a plant pathogen in the lower atmosphere using two autonomous unmanned aerial vehicles," *Journal of Field Robotics*, vol. 27, no. 3, pp. 335–343, 2010.

[82] L. Techy and C. A. Woolsey, "Minimum-time path planning for unmanned aerial vehicles in steady uniform winds," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 32, no. 6, pp. 1736–1746, 2009.

[83] C. Tomlin, G. J. Pappas, and S. S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 509–21, 1998.

[84] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministc and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, September 1986.

[85] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, pp. 640–652, June 2006.

[86] S. Waslander, R. Raffard, and C. Tomlin, "Market-based air traffic flow control with competing airlines," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 1, pp. 148–161, 2008.

[87] W. Wu and F. Zhang, "Curvature based cooperative exploration of three dimensional scalar fields," in *American Control Conference*, Baltimore, MD, Jun 2010, pp. 2909–2914.

[88] P. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Magazine*, vol. 29, no. 1, pp. 9–19, 2008.

[89] W. Xi, X. Tan, and J. S. Baras, "Gibbs sampler-based coordination of autonomous swarms," *Automatica*, vol. 42, no. 7, pp. 1107–1119, July 2006.

[90] H. Yamaguchi and J. W. Burdick, "Time-varying feedback control for nonholonomic mobile robots forming group formations," in *IEEE International Conference on Decision and Control*, vol. 4, 1998, pp. 4156–4163.

[91] E. Zermelo, "Über das navigationproble bei ruhender oder veranderlicher windverteilung," *Z. Angrew. Math. und. Mech*, vol. 11, 1931.