UNIVERSITY OF CALIFORNIA
SANTA CRUZ

INFORMATION-DRIVEN COOPERATIVE SAMPLING STRATEGIES FOR SPATIAL ESTIMATION BY ROBOTIC SENSOR NETWORKS

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

Doctor of Philosophy

in

APPLIED MATHEMATICS AND STATISTICS

by

Rishi Graham

June 2010

The Dissertation of Rishi Graham
is approved:

Professor Herbert Lee, Chair

Professor Jorge Cortés

Professor Bruno Sansó

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Information-driven Cooperative Sampling Strategies for Spatial Estimation by Robotic Sensor Networks

by

Rishi Graham

Networks of environmental sensors play an increasingly important role in scientific studies of the ocean, rivers, and the atmosphere. Robotic sensors can improve the efficiency of data collection, adapt to changes in the environment, and provide a robust response to individual failures. Ideally, online path planning algorithms should be statistically aware, driving the sensors towards those sampling locations which will provide the most information. At the same time, such algorithms need to be distributed and scalable to make robotic networks capable of operating in an autonomous and robust fashion. The combination of complex statistical modeling and distributed coordination presents difficult technical challenges: traditional statistical modeling and inference assume full availability of all measurements and central computation. While collecting sample values at a central location is certainly a desirable property, the paradigm for distributed motion coordination builds on partial, fragmented data. We present two alternative approaches to the problem of distributed optimal sampling design.

First, under a restricted class of spatio-temporal model, we consider the asymptotic regime of near independence between distinct sample locations. This study formally justifies the intuitive notion of space filling designs, thus transforming the statistical design problem into a geometric one. We provide distributed algorithms for optimal sampling under these conditions.

Second, for a more general family of Bayesian spatio-temporal models, we consider a gradient approach to sequential optimal design. We consider two well known optimality criteria: maximizing predictive entropy over potential sample locations, and minimizing average posterior variance over a predictive region. We introduce a hybrid

network of static computation and control nodes and dynamic sensing agents, and we develop approximations of these two objective functions which may be calculated in a cooperative way by this network. We detail a distributed gradient-based algorithm for obtaining local optima of the approximate objective function in a sequential setting.

for Anika Lorien Graham

who else?

Acknowledgments

I would like to thank Jorge Cortés for pushing me when I needed it and otherwise giving me reign to explore, and the faculty of the AMS department for their excellent tutelage, conversation, and inspiration. In particular, Herbie Lee and Bruno Sansó for help with spatial statistics. My thanks also to Gabe Elkaim for inspiring lectures and advice.

Thanks to Herbie Lee and Kevin Greenan for good advice, great companionship, and consistently catching my falls. Also, a shout out to all the AMS climbers and soccer folks for some good times, when I can get off my butt and show up.

Special thanks to my boyz and my entire family for all the love and support. Particularly to Lisa. No words for how wonderful she has been.

Chapter 1

Introduction

Scientific studies of environmental phenomena often involve a data collection stage. Samples are taken of a spatially distributed process of interest, such as a temperature field or chemical concentrations. Combining these samples with a model, the scientist may make predictions about the process at unmeasured locations, or inference about the quality and accuracy of the model. Any such prediction or inference should be associated with some measure of uncertainty, indicating the quality of the prediction. In a spatial context, where the model contains built-in dependencies on the locations of the samples, this uncertainty is driven in part by those locations. Optimal sampling design is the process of choosing where to take samples in order to maximize the information gained from them.

As robotic sensing technologies improve, the capability to make online adjustments to sampling locations has the potential to greatly increase the effectiveness of such data collection endeavors. The optimal design paradigm suggests using multiple sensing devices, particularly in spatial sampling, in which correlation structures allow accurate predictions from a small number of well-placed samples. Recent work in the field of distributed computation and control has enabled a new paradigm in environmental sampling. Teams of sensing and computing agents, capable of performing tasks in a robust and efficient manner, can navigate hazardous or remote terrain with little or no supervision. Distributed methods allow members of an autonomous team to take action based on local information, adding speed of response to the benefits of teamwork and autonomy. Cooperative control is the field of control theory devoted to the design and analysis of coordination algorithms to help swarms achieve some desired global goal. These algorithms usually build on simple local interaction rules because they scale well with the size of the network, and are robust to individual failures.

This thesis makes strides towards combining these two disciplines, optimal design for spatial statistics and cooperative control, developing strategies for networks of sensing robots to autonomously plan and follow optimal sampling paths in a robust and efficient manner. Obstacles to this goal arise both from the inherent difficulty of designing emergent behaviors and from the complexity and centralized nature of modern statistical methods (and particularly spatial statistics).

A typical distributed control algorithm is defined over a proximity graph, where individual agents interact only with others within a certain region of space, usually based on limited range wireless or line-of-sight communication. Thus all information known to each agent is either through direct interaction with the environment, or indirectly through their neighbors. The fact that many cooperative strategies run without an omniscient leader makes them robust to individual failures, scalable with the number of agents, and capable of easily adapting to changing conditions in the environment or in the commanded task. Due to the centralized nature of modern statistical methods and the complex dependency of optimal design on all sample locations, adapting these methods to a distributed context is an extremely difficult task. One common method in optimal design is to choose the best locations from a discrete set. Many important advances in Bayesian statistics make use of simulation methods such as Markov Chain Monte Carlo (MCMC), however such methods, at least in the current state of the technology, do not lend themselves to distributed implementation. Furthermore, the numerous choices of models for stochastic processes and different ways to define optimality provide a vast array of possible approaches. We use a Bayesian approach to spatio-temporal modeling, and treat the field as a Gaussian Process (GP). A GP is an infinite-dimensional model in which any vector of realizations from the field is treated as jointly normally distributed conditional on the space-time position. A prediction at any point in the field, or inference about model parameters takes the form of a posterior *distribution*, with uncertainty directly derived from the sampled data and the prior distribution. GP models are fully specified by mean and covariance functions, and provide powerful and flexible tools for modeling under uncertain conditions. Any measure of the utility of sample locations should be based on the uncertainty in the resulting posterior distribution. This presents a difficult challenge in a distributed setting, because the posterior uncertainty depends on all samples in a nontrivial way.

1.1 Literature review

There is a rich literature on the use of model uncertainty to drive the placement of sensing devices, e.g., [10, 67, 60, 51, 69]. Complex statistical techniques allow a detailed account of uncertainty in modeling physical phenomena. Of particular relevance to this work are [48, 29, 73, 13]. Most of this research has focused on choosing from discrete sets of hypothetical sampling locations, and until recently all of it has made use of centralized computational techniques. The work [28] examines the effect that adding and deleting measurement locations has on the kriging variance, and how this relates to optimal design. Under certain conditions on the covariance structure, data taken far from the prediction site have very little impact on the predictor [74]. When the random field does not have a covariance structure with finite spatial correlation, an approximation which does may be generated via covariance tapering [27].

In cooperative control, various works consider mobile sensor networks performing spatial estimation tasks. [82] introduces performance metrics for oceanographic surveys by autonomous underwater vehicles. [66] considers a robotic sensor network with centralized control estimating a static field from measurements with both sensing and localization error. Depending on the goal of the experiment, different types of information should be maximized [10, 16, 54, 39]. We focus on the predictive variance as a measure of the accuracy of prediction, and the predictive entropy as a measure of inferential uncertainty about model parameters. An alternative optimality criterion called mutual information [9, 46] is also effective for predictive purposes, but requires that samples and predictions are made on a discrete space (e.g., a grid). Using mutual information, the work [72] addresses the multiple robot path planning problem by greedily choosing way points from a discrete set of possible sensing locations. Given the difficulty of optimizing within the whole set of network trajectories, [49] restricts the optimization problem to a subset of possible paths described by a finite set of parameters. [83, 84] discuss the tracking of level curves in a noisy scalar field. [17] develops distributed estimation techniques for prediction of a spatiotemporal random field and its gradient. We make use of some of the tools developed in the latter paper for distributed calculations. In [61], a deterministic model is used, where the random elements come as unknown model parameters, and localization error is included. The work [12] uses a Gaussian process model where all information is globally available via all-to-all communication. Here instead we concentrate on aspects of the model and optimization which may be calculated via local information only. In [62, 58, 52, 22, 41, 57], the focus is on estimating deterministic fields with random measurement noise and, in some cases, stochastic evolution over discrete timesteps. Here we use measures of information which take into account uncertainty in the spatial process as well as its evolution over time. In between the specificity of deterministic modeling and the flexibility of fully probabilistic modeling there are other alternative methods such as distributed parameter systems [77], which introduce possibly correlated stochastic parameters into an otherwise deterministic model. In this paper, the process itself is treated as random. This allows for a simpler model and more prior uncertainty, and places the focus on estimation as opposed to inference. Complex dynamics and spatial variation are accounted for with space-time correlation instead of explicit partial differential equations. In part to cope with the additional burden of spatial uncertainty, we introduce a hybrid network of static computing nodes and mobile sensors. One alternative to the spatiotemporal Gaussian Process is the Kriged Kalman filter [3, 55] approach, which treats the process as a spatial GP with discrete temporal evolution governed by a Kalman filter. Instead, we use an integrated space-time model because it is more general, and because the treatment in this case is simpler.

Throughout this work, we make use of the technical foundations for cooperative robot control laid out in [7]. We have also found particularly useful the works [4] for everything linear algebra, and [63] for all things Voronoi.

1.2 Contributions

Here we summarize the specific contributions of this thesis according to chapter.

In Chapter 3, we consider two performance metrics for optimal placement of sensor networks based on simple kriging. We first characterize the continuity properties of the predictive variance of the estimator as a function of the network configuration. In the case of zero measurement error, this is not trivial. Previous results in the optimal design literature have avoided this problem by optimizing over a discrete set of possible configurations. We consider the continuous space of all agent locations within the region, and instead make restrictions on the form of the covariance function. Next, we define our first optimality criterion, the maximum predictive variance of the estimator as a function of network configuration. This gives a measure of the worst-case estimate over the region. We study its critical points asymptotically, in the limit of near independence. We define a second optimality criterion, the extended prediction variance of the estimator, as a novel form of D-optimality, where we introduce a method for applying this criterion to a bounded region. We study the critical points of this function within the same asymptotic framework as the first. Our main results show that circumcenter, respectively incenter, Voronoi configurations are asymptotically optimal for the maximum predictive variance over the environment, respectively the extended prediction variance. In general, these objective functions pose nonconvex and high-dimensional optimization problems. In addition, the first criterion is nonsmooth. For these reasons, it is difficult to obtain exactly the configurations that optimize them. Our results are relevant to the extent that they guarantee that, for scenarios with small enough correlation between distinct points, circumcenter and incenter Voronoi configurations are optimal for appropriate measures of uncertainty. The network can achieve these desirable configurations by executing simple distributed dynamical systems.

In Chapter 4, we consider the problem of minimizing the maximum prediction variance over agent trajectories (in time as well as space). We introduce a weighted distance metric called the correlation distance and define a novel generalized disk-covering function based on it. We show that minimizing this function is equivalent to minimizing the maximum prediction variance in the limit of near-independence, thus turning the optimization problem into a geometric one. Our next contributions all pertain to the solution of this geometric problem. We first introduce a form of generalized Voronoi partition based on the maximal correlation between a given predictive location and the samples. Assuming a fixed network trajectory, we show that this partition minimizes the maximal correlation distance over all partitions of the predictive space. We next define multicircumcenter trajectories, which minimize the maximal correlation distance over all trajectories for a fixed partition. The combination of these two results gives rise to the optimal trajectories for the correlation distance disk-covering problem. The final stage of our solution is to define an extension of the maximal correlation partition which takes into account the positions of consecutive samples taken by the same robotic agent. Over this extended set, we define a notion of centering which ensures that the distance between such consecutive samples does not exceed a maximum distance. We show that these constrained multicenter trajectories optimize the correlation distance disk-covering problem over the set of distance-constrained trajectories. Finally, using the duality between optimal trajectory and optimal partition, we design a Lloyd-type algorithm which enables the network to arrive at globally optimal trajectories. At any step of the experiment, our strategy is capable of optimizing the remainder of the trajectories as new information arrives.

In Chapter 5, we develop an approximate predictive variance which may be calculated efficiently in a sequential and distributed manner. This includes introducing a scheduled update of the estimated covariance parameter based on uncorrelated clusters of samples. Our second contribution is the characterization of the smoothness properties of the objective function and the computation of its gradient. Using consensus and distributed Jacobi overrelaxation algorithms, we show how the objective function and its gradient can be computed in a distributed way across a network composed of robotic agents and static nodes. This hybrid network architecture is motivated in part by the heavier computational capabilities of static agents and in part by the spatial structure of the problem. Our third contribution is the design of a coordination algorithm based on projected gradient descent which guarantees one-step-ahead locally optimal data collection. Due to the nature of the solution, optimality here takes into account both the unknown parameter in the covariance and the (conditional) uncertainty in the prediction. Finally, our fourth contribution is the characterization of the communication, time, and space complexities of the proposed algorithm. For reference, we compare these complexities against the ones of a centralized algorithm in which all sample information is broadcast throughout the network at each step of the optimization.

In Chapter 6, we develop an aggregate objective function based on entropy maximization. We show that the proposed objective function is a second-order approximation of the posterior predictive entropy, characterize its smoothness properties, and describe a distributed method to compute it. We employ average consensus and distributed Jacobi overrelaxation algorithms to compute the objective function and its gradient in a distributed way across a network composed of robotic agents and static nodes. Finally, we synthesize a distributed motion coordination for adaptive sampling based on one-step-ahead local optimization of data collection. We conclude illustrating the performance of the algorithm in simulation.

1.3 Organization

Our solutions take shape in two fundamentally different types of approach, which motivates the following organization for this work. In Chapter 2, we define concepts and tools used throughout the work, and introduce the overall GP model and related optimality criteria. In Part I we take the approach of restricting the class of model and the paradigm under which we consider a design "optimal". Under an assumption of near independence between samples, it is possible to formally justify the notion of a space-filling design, thus transforming the problem into one of geometric optimization. This is initially done in Chapter 3 for the case of a single set of measurements taken in a static field, and optimal results are given for the maximum predictive variance and a novel reformulation of the generalized variance. In Chapter 4, a similar method is extended to *trajectories* of multiple samples over time, optimizing for the maximum predictive variance. In both cases, the geometric optimization problems may be solved with simple distributed algorithms. In Part II, we allow for more general classes of model and tackle the distributed aspect of the problem through a series of approximations and recently developed tools for distributed computation. In this approach we introduce a hybrid network of static and mobile agents, and provide gradient based algorithms for optimization in a sequential, "adaptive design" setting. In Chapter 5 we introduce the solution using the average predictive variance as optimality criterion. This is followed in Chapter 6 by application to the predictive entropy criterion.

The models, assumptions, and optimality criteria, as well as the technical approaches differ considerably between the two methods mentioned here. For this reason, we preface each part with an introduction of its own. Chapter 7 brings it all back together with conclusions drawn from the various approaches and directions for future work.

Chapter 2

Background and preliminaries

2.1 Notation

Before we get into the technical discussion, we should first introduce some preliminary notions. In this section we outline common notational conventions and useful concepts from the literature. For ease of exposition, we break the discussion of notation into the following categories.

2.1.1 Geometry

Let $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{R}_{\geq 0}$ denote the set of reals, positive reals and nonnegative reals, respectively. Similarly, let $\mathbb{Z}$, $\mathbb{Z}_{>0}$, and $\mathbb{Z}_{\geq 0}$ denote the set of integers, positive integers and nonnegative integers, respectively. We denote by $\lfloor x \rfloor$, respectively $\lceil x \rceil$ the floor, respectively ceiling of $x \in \mathbb{R}$. We consider a convex region $\mathcal{D} \subset \mathbb{R}^d$, $d \in \mathbb{Z}_{>0}$. Let $\mathcal{D}_e = \mathcal{D} \times \mathbb{R}$ denote the space of points over $\mathcal{D}$ and time. For $p \in \mathbb{R}^d$ and $r \in \mathbb{R}_{>0}$, let $\overline{B}(p, r)$ denote the *closed ball* of radius r centered at p. For $p, q \in \mathbb{R}^d$, we let $]p, q[= \{\lambda p + (1 - \lambda)q \mid \lambda \in]0, 1[\}$ denote the *open segment* with extreme points p and q. Given $U = (u_1, \dots, u_a)^T$, $a \in \mathbb{Z}_{>0}$, and $V = (v_1, \dots, v_b)^T$, $b \in \mathbb{Z}_{>0}$, we denote by (U, V) the concatenation $(U, V) = (u_1, \dots, u_a, v_1, \dots, v_b)^T$.

2.1.2 Sets

For a set Ω, we denote by $|\Omega|$, $\mathrm{bnd}(\Omega)$, $\mathrm{int}(\Omega)$, and $\mathrm{co}(\Omega)$ its cardinality, the boundary, the interior, and the convex hull, respectively. A set $\Omega \subset \mathbb{R}^d$ is *convex*, respectively *strictly convex* if, for every $s_1, s_2 \in \Omega$ and $\alpha \in (0,1)$, we have $\alpha s_1 + (1-\alpha)s_2 \in \Omega$, respectively, $\alpha s_1 + (1-\alpha)s_2 \in \mathrm{int}(\Omega)$. Figure 2.1 illustrates the distinction between nonconvex, convex and strictly convex sets. Let $\mathrm{proj}_\Omega : \mathbb{R}^m \to \Omega$ denote the orthogonal projection onto the set Ω, $\mathrm{proj}_\Omega(s) = \mathrm{argmin}_{y\in\Omega} \|s-y\|$. Let $i_{\mathbb{F}} : (\mathbb{R}^d)^n \to \mathbb{F}(\mathbb{R}^d)$ be the natural immersion, i.e., $i_{\mathbb{F}}(P)$ contains only the distinct points in $P = (p_1, \dots, p_n)$. Note that $i_{\mathbb{F}}$ is invariant under permutations of its arguments and that the cardinality of $i_{\mathbb{F}}(p_1, \dots, p_n)$ is in general less than or equal to n. We denote by $\mathrm{d}(q,\Omega)$ the *minimum* Euclidean distance from point q to set Ω, i.e., $\mathrm{d}(q,\Omega) = \min_{s\in\Omega} \|q - s\|$. Let $\mathrm{mds} : \mathbb{R}^d \times \mathfrak{P}(\mathcal{D}) \to \mathfrak{P}(\mathcal{D})$ be the *minimum distance set* map, $\mathrm{mds}(s,\Omega) = \{s' \in \Omega \mid \|s - s'\| = \mathrm{d}(s,\Omega)\}$. Figure 2.2 illustrates the notions of orthogonal projection and minimum distance set. For a vector $P \in \mathcal{D}^n$, we will use the slight abuse of notation $\mathrm{mds}(s,P) = \mathrm{mds}(s, i_{\mathbb{F}}(P))$. The ϵ-*contraction* of a set Ω, with $\epsilon > 0$, is the set $\Omega_{\mathrm{ctn}:\epsilon} = \{q \in \Omega \mid \mathrm{d}(q, \mathrm{bnd}(\Omega)) \geq \epsilon\}$. With a slight abuse of notation, for two convex sets, Ω_1, Ω_2, we will write the minimum distance between points in the two sets as, $\mathrm{d}(\Omega_1,\Omega_2) = \min_{p\in\Omega_1} \mathrm{d}(p,\Omega_2)$. Let $\mathrm{d}_{\max} : \mathbb{R}^d \times \mathfrak{P}(\mathbb{R}^d) \to \mathbb{R}$ denote the maximum distance between a point and set, i.e., $\mathrm{d}_{\max}(s,\Omega) = \sup_{s\in\Omega}\{\|s - s\|\}$. We denote by $\mathbb{F}(\Omega)$ the collection of finite subsets of Ω. For a bounded set $\Omega \subset \mathbb{R}^d$, we let $\mathrm{IC}(\Omega)$ denote

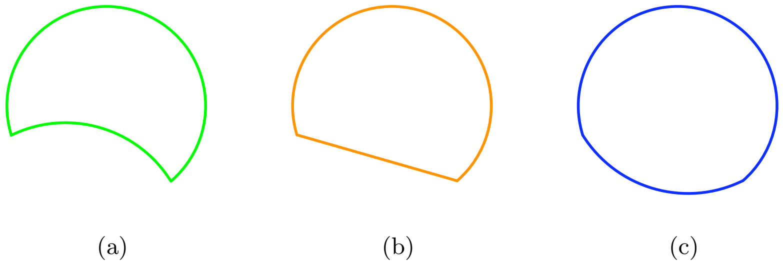
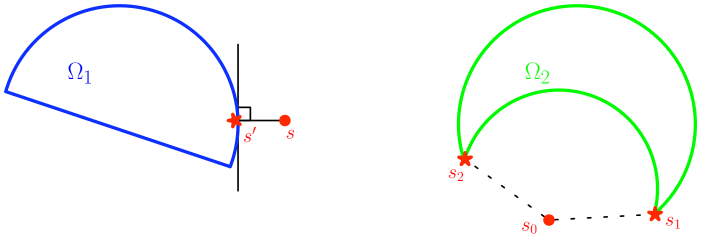


Figure 2.1: Examples of (a) a non-convex set (b) a convex set which is not strictly convex and (c) a strictly convex set.



(a) (b)

Figure 2.2: Examples of (a) the orthogonal projection, $s' = \mathrm{proj}_{\Omega_1}(s)$, (b) and the minimum distance set, $\{s_1, s_2\} = \mathrm{mds}(s_0, \Omega_2)$.

the *incenter* of Ω, that is, the center of the largest-radius d-sphere contained within Ω. We let $\mathrm{CC}(\Omega)$, respectively $\mathrm{CR}(\Omega)$ denote the *circumcenter*, respectively *circumradius* of Ω, that is, the center and radius of the smallest-radius d-sphere enclosing Ω.

Table 2.1 summarizes the notation introduced in this section.

Notation	Description
$\mathbb{R}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$	Reals, positive reals, and nonnegative reals
$\mathbb{Z}$, $\mathbb{Z}_{>0}$, $\mathbb{Z}_{\geq 0}$	Integers, positive integers, and nonnegative integers
$\lfloor a \rfloor$, $\lceil a \rceil$	Floor and ceiling of a
d	Spatial dimension of experiment region
$\mathcal{D}$	Convex spatial region where the experiment takes place
$\mathcal{D}_e$	Space-time domain of $\mathcal{D}$ over the entire experiment
$\overline{B}(p, r)$	Closed d-dimensional ball of radius r centered at p
$]p, q[$	Open segment between p and q
$\lvert\Omega\rvert$,$\mathrm{bnd}(\Omega)$,$\mathrm{int}(\Omega)$,$\mathrm{co}(\Omega)$	Cardinality, boundary, interior, and convex hull of set Ω
$\mathrm{proj}_{\Omega}(s)$	Orthogonal projection of s onto Ω
$i_{\mathbb{F}}(P)$	Natural immersion of vector P (set of distinct points)
$\mathrm{d}(p, \Omega), \mathrm{d}(\Omega_1, \Omega_2)$	Minimum point-to-set and set-to-set distance
$\mathrm{d}_{\max}(s, \Omega)$	Maximum point-to-set distance
$\mathrm{mds}(s, \Omega)$	Subset of Ω at minimum (Euclidean) distance from s
$\mathbb{F}(\Omega)$	Collection of finite subsets of Ω
$\Omega_{\mathrm{ctn}:\epsilon}$	The ϵ-contraction of Ω
$\mathrm{CC}(\Omega)$, $\mathrm{CR}(\Omega)$	Center and radius of *smallest d-sphere containing* Ω

Table 2.1: Notational conventions.

2.2 Nonsmooth analysis

Here we present some useful notions from nonsmooth analysis following [15]. For a vector, $S = (s_1, \dots, s_n)^T \in (\mathbb{R}^d)^n$, let $\pi_k : (\mathbb{R}^d)^n \to \mathbb{R}^d$ denote the canonical projection onto the kth factor, i.e. $\pi_k(S) = s_k$. For a function $f : \mathbb{R}^d \to \mathbb{R}$ and $c \in \mathbb{R}$, let

$$S_{\text{lvl}}(f, c) = \left\{ s \in \mathbb{R}^d \mid f(s) = c \right\}, \quad S_{\text{sublvl}}(f, c) = \left\{ s \in \mathbb{R}^d \mid f(s) \leq c \right\},$$

denote the c-level and c-sublevel sets of f, respectively. For a given closed, convex set $\Omega \subset \mathbb{R}^d$, let $N_\Omega : \Omega \to \mathfrak{P}(\mathbb{R}^d)$ and $T_\Omega : \Omega \to \mathfrak{P}(\mathbb{R}^d)$ map locations in Ω to the normal cone and the tangent cone of Ω, respectively. Specifically, we have

$$\begin{aligned} N_\Omega(x) &= \left\{ y \in \mathbb{R}^d \mid y^T(x - z) \geq 0, \ \forall z \in \Omega \right\} \\ T_\Omega(x) &= \left\{ y \in \mathbb{R}^d \mid y^T z \leq 0, \ \forall z \in N_\Omega(x) \right\}. \end{aligned}$$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is *locally Lipschitz at* $s \in \mathbb{R}^d$ if there exist positive constants L_s and ϵ such that $|f(y) - f(y')| \leq L_s \|y - y'\|$ for all $y, y' \in \overline{B}(s, \epsilon)$. The function f is *locally Lipschitz on* $\Omega \subseteq \mathbb{R}^d$ if it is locally Lipschitz at s, for all $s \in \Omega$. A function $f : \mathbb{R}^d \to \mathbb{R}$ is *regular at* $s \in \mathbb{R}^d$ if for all $v \in \mathbb{R}^d$, the right and generalized directional derivatives of f at s in the direction of v, coincide. The interested reader is referred to [15] for the precise definition of these directional derivatives. The *generalized gradient* of a locally Lipschitz function f is

$$\partial f(s) = \text{co} \left\{ \lim_{i \to +\infty} df(s_i) \mid s_i \to s \, , \ s_i \notin \Omega \cup \Omega_f \right\},$$

where $\Omega_f \subset \mathbb{R}^d$ denotes the set of points at which f fails to be differentiable, and Ω denotes any other set of measure zero. Note that this definition coincides with $df(s)$ if f is continuously differentiable at s. A point $s \in \mathbb{R}^d$ which satisfies that $0 \in \partial f(s)$ is called a *critical point of* f. The following results correspond to [15, Propositions 2.3.12,2.3.13] and a special case of [15, Theorem 2.3.9].

Proposition 2.2.1 (Generalized gradient of point-wise maxima) *Let $f_k : \mathbb{R}^d \to \mathbb{R}$, $k \in \{1, \dots, m\}$ be locally Lipschitz functions at $s \in \mathbb{R}^d$ and let $f : \mathbb{R}^d \to \mathbb{R}$ denote the maximum, $f(s') = \max\{f_k(s') \mid k \in \{1, \dots, m\}\}$. Then,*

(i) f is locally Lipschitz at s,

(ii) if $I(s')$ denotes the set of indices k for which $f_k(s') = f(s')$, we have

$$\partial f(s) \subseteq \operatorname{co}\{\partial f_i(s) \mid i \in I(s)\}, \tag{2.1}$$

and if f_i, $i \in I(s)$, is regular at s, then equality holds and f is regular at s.

Proposition 2.2.2 (Generalized gradient product rule) *Let $f_1, f_2 : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be Lipschitz and regular near $s \in \mathbb{R}^d$. Then the product $f_1 f_2$ is Lipschitz and regular near s, and the generalized gradient admits the form,*

$$\partial(f_1 f_2)(s) = f_2(s)\partial f_1(s) + f_1(s)\partial f_2(s).$$

Theorem 2.2.3 (Generalized gradient chain rule special case) *Let $f_1 : \mathbb{R}^d \to \mathbb{R}$ be Lipschitz and regular near s, let $f_2 : \mathbb{R} \to \mathbb{R}$ be continuously differentiable near $f_1(s)$, and let $f : \mathbb{R}^d \to \mathbb{R}$ be the composition, $f(s) = f_2(f_1(s))$. Then $f(s)$ is locally Lipschitz and regular and its generalized gradient takes the form*

$$\partial f(s) = f_2'(f_1(s))\partial f_1(s).$$

The notation introduced in this section is summarized in Table 2.2.

Notation	Description
$\pi_k(S)$	Canonical projection onto the kth factor (s_k)
$S_{\mathrm{lvl}}(f, c)$, $S_{\mathrm{lvl}}(f, c)$	c-level and -sublevel sets of f for constant
$N_\Omega(x)$,$T_\Omega(x)$	Normal and tangent cones to set Ω at point $x \in \Omega$
$\partial f(s)$	Generalized gradient of f at s

Table 2.2: Notational conventions.

2.3 Multicenter Voronoi configurations

Here we present some relevant concepts on Voronoi partitions [63, 20]. The *Voronoi partition* $\mathcal{V}(S) = (V_1(S), \dots, V_n(S))$ of $\mathcal{D}$ generated by points $S = (s_1, \dots, s_n)$ is defined by $V_i(S) = \{q \in \mathcal{D} \mid \|q - s_i\| \le \|q - s_j\|, \, \forall j \neq i\}$. Each $V_i(S)$ is called a *Voronoi cell*. Two points s_i and s_j are *Voronoi neighbors* if their Voronoi cells share a boundary. We say that P is a *circumcenter Voronoi configuration* if $p_i = \mathrm{CC}(V_i(P))$, for all $i \in \{1, \dots, n\}$, and that P is an *incenter Voronoi configuration* if $p_i \in \mathrm{IC}(V_i(P))$, for all $i \in \{1, \dots, n\}$. Figure 2.3 shows examples of these configurations.

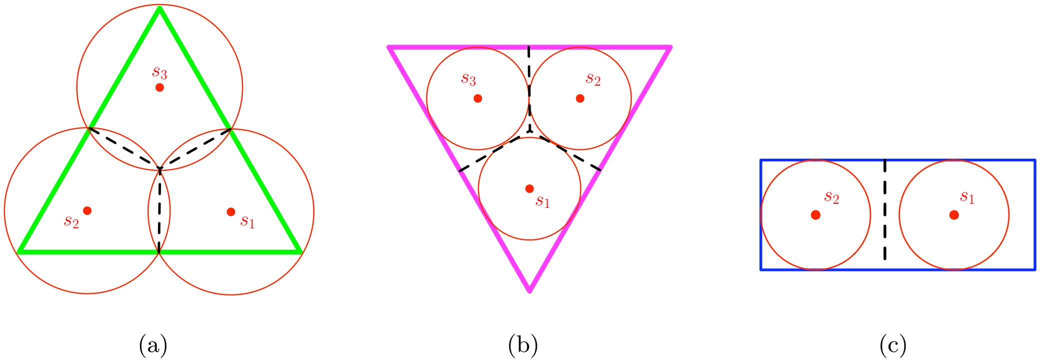


Figure 2.3: (a) A multi-circumcenter Voronoi configuration with circumcircle shown around each cell, (b) an isolated multi-incenter Voronoi configuration with inscribed circles, and (c) a multi-incenter configuration which is not isolated. In each case, the dashed lines depict boundaries between Voronoi cells.

An incenter Voronoi configuration is *isolated* if it has a neighborhood in $\mathcal{D}^n$ which does not contain any other incenter Voronoi configuration. Consider the *disk-covering* and *sphere-packing multicenter* functions defined by

$$\mathcal{H}_{\mathrm{DC}}(P) = \max_{s \in \mathcal{D}} \big\{ \mathrm{d}(s, i_{\mathbb{F}}(P)) \big\}, \; \mathcal{H}_{\mathrm{SP}}(P) = \min_{i \neq j \in \{1,\dots,n\}} \big\{ \frac{1}{2}\|p_i - p_j\|, \mathrm{d}(p_i, \mathrm{bnd}(\mathcal{D})) \big\}.$$

We are interested in the configurations that optimize these multicenter functions. The minimization of $\mathcal{H}_{\mathrm{DC}}$ corresponds to minimizing the largest possible distance of any point in $\mathcal{D}$ to one of the agents' locations given by $p_1, \dots, p_n$. We refer to it as the as the *multi-circumcenter problem*. The maximization of $\mathcal{H}_{\mathrm{SP}}$ corresponds to the situation where we are interested in maximizing the coverage of the area $\mathcal{D}$ in such a way that the radius of the generators do not overlap (in order not to interfere with each other) or leave the environment. We refer to it as the *multi-incenter problem*. It is useful to define the *index function* $N : \mathcal{D}^n \to \mathbb{Z}_{>0}$ as

$$N(P) = \left| \underset{p_i \neq p_j}{\operatorname{argmin}} \left\{ \frac{1}{2}\|p_i - p_j\|, \mathrm{d}(p_i, \mathrm{bnd}(\mathcal{D})) \right\} \right|.$$

The notation introduced in this section is summarized in Table 2.3.

Notation	Description
$\mathcal{V}(S)$	Voronoi partition generated by S
$V_i(S)$	The ith cell in the partition
$\mathcal{H}_{\mathrm{DC}}(P)$	Disk covering function
$\mathcal{H}_{\mathrm{SP}}(P)$	Sphere packing function
$N(P)$	Index function (cardinality of maximum spheres in $\mathcal{H}_{\mathrm{SP}}$)

Table 2.3: Notational conventions.

2.4 Network architecture

In the context of environmental sampling, the term "sensor network" may describe anything from a small number of fixed position rainfall monitors in the forest to a complex group of static flotation devices and mobile robots in the ocean. The literature on stochastic spatial modeling has traditionally dealt with sensors whose location is fixed in space. However, the ability to move about the field and take samples at desired locations has obvious benefits. A network in this context is a group of agents connected by wired or wireless communication paths. For our purposes, we consider networks comprised of two types of agents: static and mobile. The term "mobile agents" describes robots with the ability to move, take samples of the spatial process, and possibly sense their immediate physical environment. Their storage and computational capabilities are assumed to be minimal. By "static agents", we refer to fixed position computational devices which may or may not take samples. Because they are static and do not require energy to move around, they may carry more equipment and thus perform more in the way of computation and storage tasks. In some contexts, slower moving large vehicles may be considered static as compared to the faster mobile agents. Some limited range communication is also assumed for both types of agents. Networks may be divided into static, mobile, and hybrid, based on the classification of agents as static nodes, mobile robots, or both static and mobile. In all of these cases, the term *network* refers to the combination of the agents (static or mobile) and the communication links between them. Distributed solutions to global problems are therefore defined on the *communication graph* of the system. Sensing technology may also vary in different scenarios. Agents may have the ability to take point measurements or broader area measurements, with large or small error margins. In the case of area sensors, the measurement error may itself be a distribution as opposed to a number. Here we assume that the samples come in the form of point measurements. We will call the mobile robots $\{\mathrm{R}_1, \dots, \mathrm{R}_n\},\ n \in \mathbb{Z}_{>0}$, and denote their locations at time t by $P = P(t) = (p_1(t), \dots, p_n(t))^T \in \mathcal{D}^n$. Where static nodes are mentioned, we will call them $\{\mathrm{N}_1, \dots, \mathrm{N}_m\},\ m \in \mathbb{Z}_{>0}$ at locations $Q = (q_1, \dots, q_m)^T \in \mathcal{D}^m$.

The robots take samples of the spatio-temporal field at discrete instants of time in $\mathbb{Z}_{\geq 0}$. For simplicity, we assume that the sampling is synchronous. Our results below are independent of the particular robot dynamics, so long as each agent is able to move up to a distance $u_{\max} \in \mathbb{R}_{>0}$ between consecutive sampling times. We assume that robots have perfect information about their location. Table 2.4 summarizes notational conventions introduced in this section.

Notation	Description
R_i	The ith (mobile) robotic agent
P	Vector of locations of robotic agents
N_j	The jth static node
Q	Vector of locations of static nodes
$u_{\max}$	maximum movement between discrete sample times

Table 2.4: Notational conventions.

2.5 Bayesian modeling of space-time processes

Physical process models may be roughly divided into two categories: deterministic and stochastic. Deterministic models are often coupled with a stochastic measurement error term, e.g., [52, 22, 41, 57], but require that model parameters and initial conditions be known to a high degree of accuracy [48]. When this cannot be guaranteed, or when the parameter space of the deterministic model has high dimension, it may be desirable to treat the process itself as in some degree unknown, using a stochastic process model. A classic example is a fair coin toss. It is clear that under extremely strict monitoring of the initial conditions and model parameters, the interested physicist could exactly model the entire trajectory of the coin, culminating in its final resting position (for some interesting work in this direction, see, e.g. [24]). The model which is usually used, however, is to assign a simple probability to each outcome. In this context, it is easy to allow for the possibility that the coin is not "fair". We toss the coin a few times, collect the data, and the results give us information about the model (or about future coin tosses). For this reason, stochastic modeling is sometimes called *data driven*, as opposed to the *model driven* deterministic modeling. We focus on data driven models, and particularly their explicit representations of uncertainty.

Let z denote a random process taking values on the space-time domain, $\mathcal{D}_e$. Let $Y = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ be $n \in \mathbb{Z}_{>0}$ measurements taken from z at corresponding space-time coordinates $X = (x_1, \ldots, x_n)^T \in \mathcal{D}_e^n$, with $x_i = (s_i, t_i),\ i \in \{1, \ldots, n\}$. To formally introduce the various optimality criteria, particularly in the sequential setting considered in Part II, we make a distinction between predictive realizations of the field, sampled data, and unsampled hypothetical realizations at potential sample locations. Given the data Y and a particular stochastic model, the goal of the experiment might be to make predictions of z at any point in $\mathcal{D}_e$, or to make inference about the model itself. Optimal design is the process of choosing where to take measurements in order to reduce the uncertainty of the resulting prediction or inference. Since uncertainty drives the problem, it should be modeled as accurately as possible.

In a Bayesian setting, the prediction takes the form of a distribution, called the posterior predictive [50]. One advantage of a Bayesian approach is that parameters such as the mean and variance of the field may be treated as random variables, carrying forward uncertainty which informs the predictive distribution. We assume that z is a Gaussian Process. This means that any vector of realizations is treated as jointly normally distributed conditional on unknown parameters, with mean vector and covariance matrix dictated by the mean and covariance functions of the field. Thus a prediction made after samples have been taken is the result of conditioning the posterior predictive distribution on the sampled data. When this conditional posterior distribution is analytically tractable, as in the case of the models presented here, the resulting approach provides two powerful advantages to the design process. First, there is a direct and (under certain technical conditions) continuous map from the space-time coordinates of realizations (data and prediction) to the predictive uncertainty. Second, the joint distribution of predictions and samples allow conditioning on subsets of samples to see the effect, for instance, of optimizing over a single timestep.

If the field is modeled as a Gaussian Process with known covariance, the posterior predictive mean corresponds to the *Best Linear Unbiased Predictor*, and its variance corresponds to the mean-squared prediction error. Predictive modeling in this context is often referred to in geostatistics as *simple kriging* if the mean is also known, or *universal kriging* if the mean is treated as an unknown linear combination of known basis functions. If the covariance of the field is not known, however, few analytical results exist which take the full uncertainty (i.e., uncertainty in the field and in the parameters) into account. We present here a model [43, 29] which allows for uncertainty in the covariance and still produces an analytical posterior predictive distribution. We will call this the Kitanidis model after the author of the first derivation. We assume that the measurements take the n-variate normal distribution,

$$Y \sim \mathrm{N}_n\left(\mu(X), \sigma^2 \mathbf{K}\right), \text{ with } \mu(X) = \mathbf{F}^T \beta. \tag{2.2}$$

Here $\beta \in \mathbb{R}^p$ is a vector of unknown regression parameters, $\sigma^2 \in \mathbb{R}_{>0}$ is the unknown variance parameter, and $\mathbf{K}$ is a correlation matrix whose (i,j)th element is $\mathbf{K}_{ij} = \mathrm{Cor}[y_i, y_j]$ (discussed in more detail in Section 2.5.1). We will sometimes refer to the covariance matrix, $\sigma^2\mathbf{K}$, but mostly deal with correlation. Note that $\mathbf{K}$ is symmetric, positive definite, with 1's on the diagonal. The matrix $\mathbf{F}$ is determined by a set of $p \in \mathbb{Z}_{>0}$ known basis functions $f_i : \mathcal{D}_e \to \mathbb{R}$ evaluated at X, i.e.,

$$\mathbf{F} = \begin{bmatrix} f_1(x_1) & \dots & f_1(x_n) \\ \vdots & \ddots & \vdots \\ f_p(x_1) & \dots & f_p(x_n) \end{bmatrix}.$$

We will also use $\mathbf{f}(x) = (f_1(x), \dots, f_p(x))^T \in \mathbb{R}^p$ to denote the vector of basis functions evaluated at a single point in $\mathcal{D}_e$. It should be pointed out here that the standard approach in kriging is to use basis functions which are static with respect to time, however the related practice of "objective analysis" used in atmospheric sciences [79] does commonly use time-dependent basis functions [30]. We include the possibility of space-time basis functions for completeness. To ensure an analytical form for the posterior predictive distribution, we assume conjugate prior distributions for the parameters,

$$\beta|\sigma^2 \sim \mathrm{N}_p\left(\beta_0, \sigma^2\mathbf{K}_0\right), \tag{2.3a}$$

$$\sigma^2 \sim \Gamma^{-1}\left(\frac{\nu}{2}, \frac{q\nu}{2}\right). \tag{2.3b}$$

Here $\mathbf{K}_0 \in \mathbb{R}^{p\times p}$, $\beta_0 \in \mathbb{R}^p$, and $q, \nu \in \mathbb{R}_{>0}$ are constants, known as *tuning parameters* for the model, and $\Gamma^{-1}(a, b)$ denotes the inverse gamma distribution with shape parameter a and scale parameter b (see, e.g. [68]). It should be noted that $\mathbf{K}_0$ must be positive definite. A common practice is to use $\mathbf{K}_0$ proportional to the identity matrix. We consider two classes of optimality criteria, described in detail after a note on correlation.

2.5.1 Correlation

The spatial correlation structure is an important part of specification for any GP model. In particular the notions of stationarity, isotropy, and compact support play important roles. We will make use of these assumptions in different contexts below.

A random process, δ, taking place on $\mathcal{D}$ is second order stationary if it has constant mean and its correlation function may be written as $\mathrm{Cor}[\delta(s_1), \delta(s_2)] = C(s_1, s_2)$, where $C : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_{\geq 0}$ is a positive definite function which only depends on the difference $s_1 - s_2$. A common way to include stationarity in a spatial random process which does not have a constant mean is to write it as the sum of a deterministic mean function and a zero mean second order stationary process. A second common restriction on correlation functions is isotropy. The function C above is isotropic if it depends on s_1 and s_2 only through the distance, $\|s_1 - s_2\|$. Note that if a process has constant mean, an isotropic correlation function necessarily implies second order stationarity. Yet a third assumption sometimes made on correlation functions is that of compact support in the form of a range beyond which the function is identically zero.

When dealing with fields which change over time, it is common practice to make assumptions on the interaction between the spatial and temporal aspects of the correlation. One way to deal with time is to treat the correlation as "separable", meaning that it can be written as the product of spatial and temporal correlation functions, i.e., $\mathrm{Cor}[z(s_i, t_i), z(s_j, t_j)] = C_t(t_i, t_j)C_s(s_i, s_j)$.

In the kriging models (in which we assume the constant covariance multiplier, σ_0^2), it is also sometimes the practice to introduce an independent and identically distributed (i.i.d.) measurement error term, so that the samples are written,

$$y(x_i) = z(x_i) + \epsilon_i, \text{ with } \epsilon_i \sim \mathrm{N}(0, \tau^2),$$

for some $\tau \in \mathbb{R}$. This has the effect of adding $\tau^2 \boldsymbol{I}$ to the covariance matrix, or equivalently reformulating it as $\sigma_\tau^2 \mathbf{K}_\tau$, where $\sigma_\tau^2 = \sigma_0^2 + \tau^2$, and $\mathbf{K}_\tau$ is the matrix $\mathbf{K}$, with the off-diagonal elements multiplied by $\frac{\sigma_0^2}{\sigma_0^2 + \tau^2}$. This is mathematically equivalent to using a covariance with a "nugget" effect (see, e.g. [19]), but the sampling error term is more common in the controls literature.

In the work that follows, we will make use of the above assumptions where necessary. In order to focus on the dependence on spatial location, we will use the slight abuse of notation, $\mathrm{Cor}[x_i, x_j] = \mathrm{Cor}[y_i, y_j]$. Where we distinguish between two different vectors of sample locations, we will use the functional notation, $\mathbf{K}(X) = \mathrm{Cor}[X, X] \in \mathbb{R}^{n\times n}$, and $\mathbf{k}(s, X) = \mathrm{Cor}[X, s] \in \mathbb{R}^n$.

2.5.2 Predictive Variance

The first class of optimality criterion arises when the goal of the experiment is to make predictions of the value of z. In this case, we consider minimizing the variance of predictions made over the region. In Chapters 3, 4 we focus on worst-case mitigation: minimizing the maximum predictive variance, while Chapter 5 considers minimizing the average variance over predictions made. These correspond to the notions of G-optimality (maximum variance minimization) and A-optimality (average variance minimization) from optimal design [19, 67]. The following proposition gives the posterior predictive distribution of z. The explicit forms of optimality criteria based on this distribution will be introduced in the aforementioned chapters.

Proposition 2.5.1 (Posterior predictive distribution) *Under the prior assumptions in Equations (2.2) and (2.3), the posterior predictive at $x \in \mathcal{D}_e$ given data Y is a shifted Students t distribution (see, e.g. [68]) with $\nu + n$ degrees of freedom, with probability density function, for $z = z(x)$,*

$$p(z|Y, X) \propto \mathrm{Var}[z|Y, X]^{-\frac{1}{2}} \left(1 + \frac{(z - \mathrm{E}[z|Y, X])^2}{(\nu + n - 2)\,\mathrm{Var}[z|Y, X]}\right)^{-\frac{\nu+n+1}{2}}.$$

Here, the expectation is given by

$$\begin{aligned}\mathrm{E}[z|Y, X] &= \left(\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k}\right)^T \beta^\dagger + \mathbf{k}^T\mathbf{K}^{-1}Y,\\ \beta^\dagger &= (\mathbf{E} + \mathbf{K}_0^{-1})^{-1}\left(\mathbf{F}\mathbf{K}^{-1}Y + \mathbf{K}_0^{-1}\beta_0\right),\end{aligned}$$

where $\mathbf{E} = \mathbf{F}\mathbf{K}^{-1}\mathbf{F}^T$ and $\mathbf{k} = \mathrm{Cor}[Y, z] \in \mathbb{R}^n$. The variance is given by

$$\begin{aligned}&\mathrm{Var}[z|Y, X] = \varphi(Y, X)\phi(x; X),\\ &\quad\phi(x; X) = \mathrm{Cor}[z, z] - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k} + (\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k})^T(\mathbf{K}_0^{-1} + \mathbf{E})^{-1}(\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k})\\ &\quad\varphi(Y, X) = \frac{1}{\nu + n - 2}\left(q\nu + (Y - \mathbf{F}^T\beta_0)^T\left(\mathbf{K} + \mathbf{F}^T\mathbf{K}_0\mathbf{F}\right)^{-1}(Y - \mathbf{F}^T\beta_0)\right),\end{aligned}$$

The proof of the proposition follows from the application of Bayes Theorem to the model given by (2.3). Alternatively, it can also be derived from results in [29] and [43] using a technique similar to the one used in the proof of Proposition B.0.3 in Appendix B. Note that since $\mathbf{K}_0$ and $\mathbf{K}$ are positive definite, the quantities $\phi(x;X)$ and $\varphi(Y,X)$ are well posed.

Remark 2.5.2 (Terms in the posterior predictive variance) The posterior predictive variance in Proposition 2.5.1 is a product of two terms. The first, $\varphi(Y,X)$, is the posterior mean of σ^2, given the sampled data. We refer to it as the *sigma mean.* The second term, $\phi(x;X)$, can be thought of as the scaled posterior predictive variance conditioned on σ^2. We refer to it as the *sigma-conditional variance.* •

The sigma-conditional variance is very close to what the predictive variance would look like if σ^2 were known, as we show next. The following results may be derived by applying Bayes Theorem to the model specified by Equations (2.2), with the appropriate alterations of the priors.

Proposition 2.5.3 (Kriging variance) *If the variance parameter is known, σ_0^2, the result is the* universal kriging predictor*, with posterior predictive variance,*

$$\mathrm{Var}_{UK}[z|Y,X] = \sigma_0^2 \left(\mathrm{Cor}[z,z] - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k} + (\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k})^T\mathbf{E}^{-1}(\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k})\right).$$

If, in addition, the mean function, $\mu(X)$, is known, the result is the simple kriging predictor*, and the posterior predictive variance is,*

$$\mathrm{Var}_{SK}[z|Y,X] = \sigma_0^2 \left(\mathrm{Cor}[z,z] - \mathbf{k}^T\mathbf{K}^{-1}\mathbf{k}\right).$$

Remark 2.5.4 (Notation for kriging variance) Note that in the kriging models the posterior predictive variance depends on the locations of the samples, X, but not on their values. For this reason, we will use the shorthand $\mathrm{Var}_{\mathrm{SK}}[s|X] = \mathrm{Var}_{\mathrm{SK}}[z|Y,X]$ and $\mathrm{Var}_{\mathrm{UK}}[s|X] = \mathrm{Var}_{\mathrm{UK}}[z|Y,X]$. •

2.5.3 Predictive Entropy

The second class of criterion is inferential. If we are more interested in making inference about the quality or accuracy of the model itself, an appropriate measure of uncertainty is the predictive entropy of a vector of potential sample locations. A related measure is the generalized variance, which focuses more in theory on the predictive aspect of the problem, but amounts to the same optimization criterion. Because of the similarities between entropy and generalized variance, we will use the same symbol, $\mathcal{E}$ for both metrics. In Chapters 3 and 6 we consider inferential uncertainty.

The entropy of an arbitrary continuous distribution with pdf $p(y)$ can be written as $\mathcal{E} = -\operatorname{E}\left[\log \frac{p(y)}{h(y)}\right]$, where $h(y)$ is a reference measure chosen to ensure invariance under affine transformations of y. When the data come from a multivariate Student t distribution, $Y \sim \mathrm{t}_n(\mu, \Psi, \delta)$, then the entropy is [48],

$$\mathcal{E} = \frac{1}{2} \log \det \left((\delta - n + 1)\Psi\right). \tag{2.4}$$

Consider the situation at timestep k, in which n samples have already been taken at each of $k-1$ previous timesteps, and we wish to choose the best locations for the next set of measurements. The number of *unsampled* measurements is n, while the number of *sampled* measurements is $n_s = n(k-1)$. The sample vector, Y, may then be partitioned as $Y = (Y_s, Y_u)$, with corresponding partitions, $X = (X_u, X_s)$, basis function matrix, $\mathbf{F} = (\mathbf{F}_u, \mathbf{F}_s)$, and correlation matrix,

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_u & \mathbf{K}_{us} \\ \mathbf{K}_{su} & \mathbf{K}_s \end{bmatrix}.$$

Using Lemma B.0.1 in Appendix B, we may write the posterior predictive variance at the unsampled locations, conditional on the sampled ones as,

$$\operatorname{Var}[Y_u | Y_s, X] = \varphi(Y_s, X_s)\phi(X_u; X_s),$$

where, with a slight abuse of notation, we have used $\phi(X_u; X_s)$ to denote the following multivariate extensions of ϕ and φ,

$$\begin{aligned}\phi(X_u; X_s) = {} & \mathbf{K}_u - \mathbf{K}_{us}\mathbf{K}_s{}^{-1}\mathbf{K}_{su} + \\ & + \left(\mathbf{F}_u - \mathbf{F}_s\mathbf{K}_s{}^{-1}\mathbf{K}_{su}\right)^T \left(\mathbf{K}_0^{-1} + \mathbf{F}_s\mathbf{K}_s{}^{-1}\mathbf{F}_s{}^T\right)^{-1} \left(\mathbf{F}_u - \mathbf{F}_s\mathbf{K}_s{}^{-1}\mathbf{K}_{su}\right), \\ \varphi(Y_s, X_s) = {} & \frac{1}{\nu + ns - 2}\Big(q\nu + \left(Y_s - \mathbf{F}_s{}^T\beta_0\right)^T \left(\mathbf{K}_s + \mathbf{F}^T\mathbf{K}_0\mathbf{F}\right)^{-1} \left(Y_s - \mathbf{F}_s{}^T\beta_0\right)\Big).\end{aligned}$$

Identifying terms in (2.4), the posterior predictive entropy at unsampled locations is,

$$\mathcal{E}_u = \frac{1}{2}\log\det\left(\phi(X_u; X_s)\right) + \mathcal{M}(Y_s, X_s), \tag{2.5}$$

where $\mathcal{M}(Y_s, X_s)$ does not depend on the locations or values of the new samples. Given past measurements at locations X_s, it is desirable to take the next measurements at locations X_u which maximize $\log\det\left(\phi(X_u; X_s)\right)$. This function, which we call the *conditional entropy*, is invariant under permutations of X_u, so we are free to choose any ordering to facilitate computation. Notation summarized in Table 2.5.

Notation	Description
S	Vector of spatial locations of samples (potential or realized)
X, x_i	Space-time coordinate vector, element of that vector
Y, y_i	Vector of sample values, element of sample vector
$\mathrm{E}[A], \mathrm{Var}[A]$	Expectation and variance of random vector A
$\mathrm{Cor}[a, b]$	Correlation between random variables (or vectors) a and b
z	Random space-time process of interest
$\mathbf{K}$	Sample correlation matrix
$\mathbf{k}$	Samples to prediction correlation vector
$\mathbf{F}$	Matrix of basis functions evaluated at sample locations
$\mathbf{f}$	Basis vector evaluated at predictive location
β	Unknown mean regression parameters
$\beta_0, \mathbf{K}_0$	Prior mean vector and correlation matrix of β
σ^2	Unknown variance scalar parameter
q, ν	Tuning parameters of prior distribution for σ^2
ϕ	Predictive variance conditional on σ^2 ("conditional variance")
φ	Posterior mean of σ^2 ("sigma mean")
$\mathcal{E}$	Entropy

Table 2.5: Statistical notation.

2.6 Basic linear algebraic facts

Here we present some basic facts from linear algebra [4] that will be useful throughout the paper. We use $\lambda_{\min}(A)$, respectively $\lambda_{\max}(A)$ to denote the smallest, respectively largest eigenvalue of square matrix A, and $\mathrm{sprad}(A)$ to denote its spectral radius. Note that $\lambda_{\max}(A - \boldsymbol{I}) = \lambda_{\max}(A) - 1$. Let $\det(A)$ denote the determinant of matrix A. We denote by $[A]_{ij}$ the (i, j)th element of the matrix A, and by $\mathrm{row}_i(A)$, respectively $\mathrm{col}_i(A)$, its ith row, respectively column. Let $\mathbf{0}_{i\times j}$ denote the $i \times j$ zero matrix. If the dimensions are clear from the context we may omit the subscripts and use $\mathbf{0}$. A useful consequence of the Gershgorin Circle Theorem (see, e.g., [4, Fact 4.10.13]) for positive definite matrices yields the following bound on the largest eigenvalue,

$$\lambda_{\max}(A) \leq \max_{i\in\{1,\ldots,n\}} \{\sum_{j=1}^{n} [A]_{ij}\}. \tag{2.6}$$

Given a partitioned matrix, $A = \begin{smallmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{smallmatrix}$, we denote by $(A_{11} \,|A)$, respectively $(A_{22} \,|A)$ the Schur complement of A_{11}, respectively A_{22} in A, i.e.,

$$(A_{11} \,|A) = A_{22} - A_{21}A_{11}^{-1}A_{12} \qquad \text{and} \qquad (A_{22} \,|A) = A_{11} - A_{12}A_{22}^{-1}A_{21}.$$

Using the Schur complement, we can write the determinant,

$$\det(A) = \det(A_{11})\det(A_{11} \,|A). \tag{2.7}$$

A matrix A is nonnegative if $[A]_{ij} \geq 0$ for all i, j. A matrix norm, $\|\cdot\|$, is said to be *submultiplicative* if it satisfies the inequality $\|AB\| \leq \|A\|\|B\|$, and said to be *normalized* if it satisfies $\|\boldsymbol{I}\| = 1$. The map $A \to \lambda_{\max}(A)$ is a normalized, submultiplicative norm on the set of nonnegative Hermitian matrices.

Let e^A denote the standard matrix exponential of A. For a positive definite matrix B, there exists a (not necessarily unique) matrix A of the same dimensions such that $e^A = B$, in which case $\log\det(B) = \mathrm{tr}(A)$. If B satisfies,

$$\|B - \boldsymbol{I}\|_S < 1, \tag{2.8}$$

where $\| \cdot \|_S$ is a normalized submultiplicative norm, then one representation of the matrix logarithm is the Taylor series,

$$\log(B) = \sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i} (B - \boldsymbol{I})^i. \tag{2.9}$$

Let $M : \mathbb{R} \to \mathbb{R}^{n \times n}$ denote a mapping of some real value to an invertible $n \times n$ matrix. Then we may write [56, 21]

$$\frac{d}{da} \log \det \left(M(a)\right) = \operatorname{tr}\Big(M(a)^{-1} \frac{d}{da} M(a)\Big), \tag{2.10}$$

where the derivative of the matrix is taken component-wise.

Table 2.6 reviews notation introduced in this section.

Notation	Description
$\lambda_{\min}(A)$, $\lambda_{\max}(A)$	Extremal eigenvalues of square matrix, A
$\mathrm{sprad}(A)$	Spectral radius of A
$\det(A)$	Determinant of A
$\mathrm{row}_i(A)$, $\mathrm{col}_j(A)$	Row i and column j of matrix A
$\mathbf{0}_{i \times j}$	$i \times j$ dimensional zero vector
$[A]_{ij}$	Element i, j of matrix A
$(B\,\vert A)$	Schur complement of submatrix B in matrix A
e^A	Matrix exponential of A
$\log(A)$	Taylor series matrix logarithm (for appropriate A)

Table 2.6: Notational conventions.

Part I

Geometric solutions under near independence

We begin by considering the case in which the mean and covariance of the random field are known, using the simple kriging model. Due to the nontrivial interactions between correlated samples, exactly characterizing the optimal sampling trajectories even for this simplest of models is a difficult problem. Simply choosing a fixed number of sampling locations from a discrete set has been shown to be NP-hard [44]. In our technical approach, we have been inspired by [42], which considers the problem of minimizing the maximum uncertainty over a discrete space and shows that minimax configurations are asymptotically optimal as the correlation between any two distinct points vanishes. Minimax configurations minimize the maximum distance to the nearest agent from any point in space. In Chapter 3 we build on this setup to characterize the optimal network configurations in continuous sampling spaces at a fixed instant in time (i.e. a single snapshot of the field) and established the connection with Voronoi partitions [63] and geometric optimization [2, 25]. The work [18] defines circumcenter and incenter Voronoi configurations and proposes coordination algorithms which steer the network to these configurations. Subsequently, in Chapter 4, we extend this notion to the scenario in which the agents take multiple of samples at sequential timesteps.

Chapter 3

Optimal configurations for sampling static fields under near independence

Here we summarize the work published in the conference paper [31], and the follow-up technical report [33]. In it, we consider the problem of where to place the agents in the case that a single measurement is to be taken by each, all at a single instant of time. Addressing this problem is an initial step towards the more ambitious goal of characterizing optimal coordinated agent trajectories when multiple measurements are possible.

3.1 Model assumptions

As we are concerned with a purely spatial problem in this chapter, we consider only the spatial correlation function in the simple kriging equations. Furthermore, since the agents will only take a single sample, we will use P in place of X for sample locations. We assume that the correlation function is isotropic and that the samples may be corrupted with measurement error, so that the covariance matrix is given by $\sigma_\tau^2 \mathbf{K}_\tau$, with $\tau \in \mathbb{R}$. We denote by $g_s : \mathbb{R}_{\geq 0} \to \mathbb{R}_{>0}$ the scaled isotropic correlation function, so that $[\mathbf{K}_\tau]_{ij} = g_s(\|s_i - s_j\|)$ for $i \neq j \in \{1, \ldots, n\}$.

3.2 Objective functions

Here, we consider two objective functions inspired by the notions of G- and D-optimality from optimal design [19, 67]. The *maximum predictive variance* is

$$\mathcal{M}(p_1, \ldots, p_n) = \max_{s \in \mathcal{D}} \mathrm{Var}_{\mathrm{SK}}[z(s); p_1, \ldots, p_n] = \sigma_0^2 - \sigma_\tau^2 \min_{s \in \mathcal{D}} \{\mathbf{k}\mathbf{K}_\tau^{-1}\mathbf{k}\}. \tag{3.1a}$$

Let us make an important observation about the well-posedness of $\mathcal{M}$. Under noisy measurements, i.e., $\tau^2 > 0$, the functional map $s \mapsto \mathrm{Var}_{\mathrm{SK}}[z(s); P]$ is well-defined for any $s \in \mathcal{D}$ and $P \in \mathcal{D}^n$. Indeed, the dependence of $\mathrm{Var}_{\mathrm{SK}}$ on the network configuration is continuous, and hence, $\mathcal{M}$ is also well-defined. However, when no measurement noise is present, i.e., $\tau^2 = 0$, then the matrix $\mathbf{K}_\tau$ is not invertible for configurations in which one or more samples are collocated, and therefore, it is not clear what the value of $\mathrm{Var}_{\mathrm{SK}}$ is. This problem is carefully addressed in Proposition 3.3.2, where it is shown that, in the no measurement noise case, $\mathrm{Var}_{\mathrm{SK}}$ is a continuous function of the configuration under suitable technical conditions on the covariance structure of the spatial field.

Our second objective function requires some background. The generalized variance [81] of the simple kriging predictor is defined as $\det\left((\sigma_\tau^2 \mathbf{K}_\tau)^{-1}\right)$. Minimizing the generalized variance is equivalent to minimizing $-\det(\mathbf{K}_\tau)$. For discrete state spaces, it can be shown [42] that configurations which maximize the minimum distance between agents asymptotically minimize $-\det(\mathbf{K}_\tau)$ in the limit of near independence. This tends to place agents on the boundary of $\mathcal{D}$. Since we are only interested in predictions over $\mathcal{D}$, we would like a notion of optimality which penalizes agents too close to the boundary as it does agents too close to each other. To this end, let $\gamma : \mathcal{D} \to \mathbb{R}^d$ map a point in $\mathcal{D}$ to its mirror image reflected across the nearest boundary of $\mathcal{D}$. Formally,

$$\gamma(s) \in s + 2\Big(\operatorname*{argmin}_{s^* \in \mathrm{bnd}(\mathcal{D})} \{\|s^* - s\|\} - s \Big).$$

Note that $\gamma(s)$ is in general not unique, and is not a smooth function of s. However, $\|s - \gamma(s)\|$ is smooth, and is the same for all values of $\gamma(s)$. Now consider minimizing the negative determinant of the estimator which would result if we had data from all agents as well as their reflections. The *extended prediction variance* is then

$$\mathcal{E}(p_1, \dots, p_n) = -\det\left(\mathbf{K}_\tau(p_1, \dots, p_n, \gamma(p_1), \dots, \gamma(p_n))\right). \tag{3.1b}$$

Since $\mathcal{E}$ does not require inversion of the covariance matrix, it is always well-posed. Our goal is to find the network configurations $P = (p_1, \dots, p_n) \in \mathcal{D}^n$ that minimize the objective functions $\mathcal{M} : \mathcal{D}^n \to \mathbb{R}$ and $\mathcal{E} : \mathcal{D}^n \to \mathbb{R}$.

3.3 Optimal configurations for spatial prediction

In this section, we provide several results that characterize the optimal network configurations for the objective functions $\mathcal{M}$ and $\mathcal{E}$. In Section 3.3.1, we show that minima of $\mathcal{M}$ cannot contain coincident samples. Beyond the obvious benefit of collision avoidance, this fact is useful in Section 3.3.2 where we show that circumcenter and incenter Voronoi configurations are asymptotically optimal for $\mathcal{M}$ and $\mathcal{E}$, respectively.

3.3.1 Coincident configurations are not minima of the maximum error variance

In this section, we examine the effect of the location of a subset of agents on the error variance terms. In particular we are interested in comparing $\mathrm{Var}_{\mathrm{SK}}[z(s); P]$ against $\mathrm{Var}_{\mathrm{SK}}[z(s); i_{\mathbb{F}}(P)]$ for configurations P which contain one or more coincident locations. The following lemma provides a useful decomposition of $\mathrm{Var}_{\mathrm{SK}}$.

Lemma 3.3.1 *The estimation error variance function may be written in the form*

$$\mathrm{Var}_{SK}[z(s); P] = \mathrm{Var}_{SK}[z(s); \overline{P}] - \frac{\left(\mathcal{N}(s, p_1; \overline{P})\right)^2}{\mathrm{Var}_{SK}[z(p_1); \overline{P}] + \tau^2}, \tag{3.2}$$

with $\mathcal{N}(s, p_1; \overline{P}) = g_s(\|s - p_1\|) - \mathrm{Cor}[z(s), Y(\overline{P})]\mathbf{K}_\tau(\overline{P})^{-1}\,\mathrm{Cor}[Y(\overline{P}), y(p_1)]$ *and* $\overline{P} = (p_2, \dots, p_n) \in \mathcal{D}^{n-1}$.

This fact may be proved using [4, Proposition 8.2.4] for the inverse of a partitioned symmetric matrix. Equation (3.2) may be applied repeatedly to isolate the effects of any subset of locations in P. In the following proposition we consider the behavior of $\mathcal{M}$ as agents move around $\mathcal{D}$. The proof may be found in Appendix A.1.

Proposition 3.3.2 (Continuity of predictive variance) *Let $S_{coinc} \subset \mathcal{D}^n$ denote the set of all configurations with one or more coincident points, i.e.,*

$$S_{coinc} = \{P \in \mathcal{D}^n \mid p_i = p_j \; \textit{for some} \; i \neq j \in \{1, \dots, n\}\}.$$

Assume that the function g is differentiable, with $g'(0) \neq 0$, and $\tau^2 = 0$. Then, for each $s \in \mathcal{D}$, the predictive variance, $(p_1, \dots, p_n) \mapsto \mathrm{Var}_{SK}[z(s); p_1, \dots, p_n]$ is continuous. In addition, for $P \in S_{coinc}$ we have $\mathrm{Var}_{SK}[z(s); P] = \mathrm{Var}_{SK}[z(s); i_{\mathbb{F}}(P)]$.

Under the assumptions of Proposition 3.3.2, we can extend the mean-squared error function by continuity to include configurations in S_{coinc}. With a slight abuse of notation, in the case of no measurement error, we use $\mathrm{Var}_{\mathrm{SK}}[z(s); P]$ to denote $\mathrm{Var}_{\mathrm{SK}}[z(s); i_{\mathbb{F}}(P)]$ for $P \in S_{\mathrm{coinc}}$.

Proposition 3.3.3 (Minima of $\mathcal{M}$ are not in S_{coinc}) *Let $P^\dagger \in \mathcal{D}^n$ be a strict local minimum of the map $P \mapsto \mathcal{M}(P)$. Under the assumptions of Proposition 3.3.2, $P^\dagger \notin S_{coinc}$.*

Proof: We proceed by contradiction. Assume $P^\dagger \in S_{\mathrm{coinc}}$. Consider a configuration $P \in \mathcal{D}^n \setminus S_{\mathrm{coinc}}$ in a neighborhood of $P^\dagger$ such that $i_{\mathbb{F}}(P^\dagger) \subset i_{\mathbb{F}}(P)$. Let $s, s^\dagger \in \mathcal{D}$ such that $\mathcal{M}(P) = \mathrm{Var}_{\mathrm{SK}}[z(s); P]$ and $\mathcal{M}(P^\dagger) = \mathrm{Var}_{\mathrm{SK}}[z(s)^\dagger; P^\dagger]$. Using Lemma 3.3.1 and Proposition 3.3.2, one can deduce that $\mathrm{Var}_{\mathrm{SK}}[z(s); P^\dagger] \geq \mathrm{Var}_{\mathrm{SK}}[z(s); P]$. By the definition of $\mathcal{M}$, $\mathrm{Var}_{\mathrm{SK}}[z(s)^\dagger; P^\dagger] \geq \mathrm{Var}_{\mathrm{SK}}[z(s); P^\dagger]$. Therefore $\mathcal{M}(P^\dagger) = \mathrm{Var}_{\mathrm{SK}}[z(s)^\dagger; P^\dagger] \geq \mathrm{Var}_{\mathrm{SK}}[z(s); P^\dagger] \geq \mathrm{Var}_{\mathrm{SK}}[z(s); P] = \mathcal{M}(P)$, which is a contradiction. ■

3.3.2 Multicenter Voronoi configurations are asymptotically optimal

Let us consider the objective functions $\mathcal{M}$ and $\mathcal{E}$ introduced in Section 3.2 but with correlation function Cor^α, $\alpha \in \mathbb{Z}_{>0}$. As α grows, the correlation between distinct points in $\mathcal{D}$ vanishes. Note that Cor^α retains much of the shape of the original correlation function (e.g. smoothness, range, etc), so this analysis is helpful in determining the properties of the original problem as well. To ease the exposition, we denote by $\mathbf{k}^{\{\alpha\}}$, respectively $\mathbf{K}_\tau^{\{\alpha\}}$, the vector $\mathbf{k}$, respectively the matrix $\mathbf{K}_\tau$, with each element raised to the kth power. Similarly, let $\mathcal{M}^{\{\alpha\}}, \mathcal{E}^{\{\alpha\}} : \mathcal{D}^n \to \mathbb{R}$ be defined as

$$\mathcal{M}^{\{\alpha\}}(p_1, \dots, p_n) = (\sigma_0^2)^{\{\alpha\}} - (\sigma_\tau^2)^{\{\alpha\}} \min_{s \in \mathcal{D}} \{(\mathbf{k}^{\{\alpha\}})^T (\mathbf{K}_\tau^{\{\alpha\}})^{-1} \mathbf{k}^{\{\alpha\}}\},$$
$$\mathcal{E}^{\{\alpha\}}(p_1, \dots, p_n) = -\det\left(\mathbf{K}_\tau^{\{\alpha\}}\left(p_1, \dots, p_n, \gamma(p_1), \dots, \gamma(p_n)\right)\right).$$

First we establish a result on the cardinality of the minimum distance set. Let $C_{\mathrm{mds}} : \mathbb{R}^d \times \mathcal{D}^n \to \mathbb{R}$ such that $C_{\mathrm{mds}}(s, P) = g_s(\|s - p\|)$, for any $p \in \mathrm{mds}(s, P)$. Note that C_{mds} is well-defined.

Proposition 3.3.4 (Cardinality of minimum distance set) *Let the correlation function g_s be continuous. For $P \in \mathcal{D}^n \setminus S_{coinc}$, one has*

$$\min_{s \in \mathcal{D}} \{C_{\mathrm{mds}}(s, P)\, |\mathrm{mds}(s, P)|\} = \min_{s \in \mathcal{D}} \{C_{\mathrm{mds}}(s, P)\}.$$

The proof of Proposition 3.3.4 is in Appendix A.1. We are now ready to prove one of the main results of the paper. The proof follows a similar line of reasoning to [42].

Theorem 3.3.5 (Minima of $\mathcal{M}$ under near independence) *Let $P_{mcc} \in \mathcal{D}^n$ be a global minimizer of the multi-circumcenter problem. Then, as $k \to \infty$, P_{mcc} asymptotically globally optimizes $\mathcal{M}^{\{\alpha\}}$, that is, $\mathcal{M}^{\{\alpha\}}(P_{\mathrm{mcc}})$ approaches a global minimum.*

Proof: Note that minimizing $\mathcal{M}^{\{\alpha\}}$ is equivalent to finding the tuples P which maximize the function $L^{\{\alpha\}} : \mathcal{D}^n \to \mathbb{R}$ defined as

$$L^{\{\alpha\}}(P) = \min_{s \in \mathcal{D}} \left\{(\mathbf{k}^{\{\alpha\}}(s, P))^T (\mathbf{K}_\tau^{\{\alpha\}}(P))^{-1} (\mathbf{k}^{\{\alpha\}}(s, P))\right\}.$$

Let $\lambda_{\min}$ and $\lambda_{\max} : \mathcal{D}^n \times \mathbb{R} \to \mathbb{R}$ be such that $\lambda_{\min}(P, \alpha)$, $\lambda_{\max}(P, \alpha)$ denote, respectively, the minimum and the maximum eigenvalue of $\mathbf{K}_\tau^{\{\alpha\}}(P)$. We can see that $L^{\{\alpha\}}(P)$ is bounded above by $\lambda_{\max}(P, k) \sum_{p \in P} g_s(\|s - p\|)^{\{2\alpha\}}$ and below by $\lambda_{\min}(P, k) \sum_{p \in P} g_s(\|s-$

$p\|)^{\{2\alpha\}}$. For a given s, in terms of the minimum distance set we can write

$$\begin{aligned}\sum_{p\in P} g_s(\|s-p\|)^{\{2\alpha\}} &= \sum_{p\in \mathrm{mds}(s,P)} g_s(\|s-p\|)^{\{2\alpha\}} + \sum_{p\in P\setminus \mathrm{mds}(s,P)} g_s(\|s-p\|)^{\{2\alpha\}} \\ &= |\mathrm{mds}(s,P)|\, C_{\mathrm{mds}}(s,P)^{\{2\alpha\}} + \sum_{p\in P\setminus \mathrm{mds}(s,P)} g_s(\|s-p\|)^{\{2\alpha\}}.\end{aligned}$$

As $k \to \infty$ the elements in the minimum distance set dominate, so we are left with

$$\sum_{p\in P} g_s(\|s-p\|)^{\{2\alpha\}} = |\mathrm{mds}(s,P)|\, C_{\mathrm{mds}}(s,P)^{\{2\alpha\}} + o(C_{\mathrm{mds}}(s,P)^{\{2\alpha\}}).$$

From Proposition 3.3.4,

$$\min_{s\in\mathcal{D}} \{|\mathrm{mds}(s,P)|\, C_{\mathrm{mds}}(s,P)\} = \min_{s\in\mathcal{D}} \{C_{\mathrm{mds}}(s,P)\},$$

so we can write

$$\min_{s\in\mathcal{D}} \Big\{ \sum_{p\in P} g_s(\|s-p\|)^{\{2\alpha\}} \Big\} = \min_{s\in\mathcal{D}} \Big\{ C_{\mathrm{mds}}(s,P)^{\{2\alpha\}} \, (1+o(1)) \Big\}.$$

Consider, then, comparing an arbitrary configuration P^* against a global minimizer of $\mathcal{H}_{\mathrm{DC}}$, say P_{mcc}. In the zero measurement error case, by Proposition 3.3.3, we can assume without loss of generality that $P^* \notin S_{\mathrm{coinc}}$. Therefore, no matter what the value of τ is, we can safely use the eigenvalues of $(\mathbf{K}_\tau^{\{\alpha\}})^{-1}$ to provide bounds. Specifically,

$$\frac{L^{\{\alpha\}}(P^*)}{L^{\{\alpha\}}(P_{\mathrm{mcc}})} \le \frac{\lambda_{\max}(P^*,k) \min_{s\in\mathcal{D}} \Big\{ C_{\mathrm{mds}}(s,P^*)^{\{2\alpha\}} \, (1+o(1)) \Big\}}{\lambda_{\min}(P_{\mathrm{mcc}},k) \min_{s\in\mathcal{D}} \Big\{ C_{\mathrm{mds}}(s,P_{\mathrm{mcc}})^{\{2\alpha\}} \, (1+o(1)) \Big\}}. \tag{3.3}$$

Next we take a closer look at the eigenvalues. Note that $\lim_{k\to\infty} \mathbf{K}_\tau^{\{\alpha\}}(P) = I_n$, and it can be seen that $\lambda_{\min}(P,k)$ and $\lambda_{\max}(P,k)$ both tend to 1 for any configuration P. Finally, since P_{mcc} minimizes the maximum distance to any point $s \in \mathcal{D}$, it maximizes the minimum covariance, so for any $P \in \mathcal{D}^n$, $\min_{s\in\mathcal{D}} C_{\mathrm{mds}}(s,P) \le \min_{s\in\mathcal{D}} C_{\mathrm{mds}}(s,P_{\mathrm{mcc}})$. Thus the ratio (3.3) is bounded by $1+o(1)$. Therefore, in the limit as $k \to \infty$, minimizing $\mathcal{M}^{\{\alpha\}}$ is equivalent to solving the multi-circumcenter problem. ■

The proof of the theorem can be reproduced for local minimizers of the multi-circumcenter problem to arrive at the following result.

Corollary 3.3.6 *Let $P_{\mathrm{mcc}} \in \mathcal{D}^n$ be a local minimizer of the multi-circumcenter problem. Then, as $k \to \infty$, P_{mcc} asymptotically optimizes $\mathcal{M}^{\{\alpha\}}$, that is, $\mathcal{M}^{\{\alpha\}}(P_{\mathrm{mcc}})$ approaches a minimum.*

According to [18], under certain technical conditions, solutions to the multi-circumcenter problem are circumcenter Voronoi configurations. Next, let us present a similar asymptotic result for the extended prediction variance.

Theorem 3.3.7 (Minima of $\mathcal{E}$ under near independence) *Let $P_{\mathrm{mic}} \in \mathcal{D}^n$ be a global maximizer of the multi-incenter problem with lowest index. Then, as $k \to \infty$, P_{mic} asymptotically globally optimizes $\mathcal{E}^{\{\alpha\}}$, that is, $\mathcal{E}^{\{\alpha\}}(P_{\mathrm{mic}})$ approaches a global minimum.*

Proof: Expanding the objective function for asymptotically dominant terms, we may write $\mathcal{E}^{\{\alpha\}}(P) = -1 + J^{\{\alpha\}}(P) + o\left(J^{\{\alpha\}}(P)\right)$, where $J^{\{\alpha\}}(P) = \sum_{i \neq j} g_s(\|p_i - p_j\|)^{\{2\alpha\}} + \sum_{i,j=1}^{n} g_s(\|p_i - \gamma(p_j)\|)^{\{2\alpha\}} + \sum_{i \neq j} g_s(\|\gamma(p_i) - \gamma(p_j)\|)^{\{2\alpha\}}$. Asymptotically all but the largest terms in $J^{\{\alpha\}}(P)$ will drop out, and minimizing $\mathcal{E}^{\{\alpha\}}(P)$ becomes equivalent to minimizing those terms. The largest terms in $J^{\{\alpha\}}(P)$ correspond to the shortest distance between the locations of either the agents or their reflected images. For any two agent locations, $p_i, p_j \in \mathcal{D}$, and any of their reflections $\gamma(p_i), \gamma(p_j)$ the minimum distance between any two of the four points can be reduced to $\min\left\{\|p_i - p_j\|, \|p_i - \gamma(p_i)\|, \|p_j - \gamma(p_j)\|\right\}$ (note that this is not in general true for non-convex domains). Thus the shortest distance between agents in P and their reflections may be expressed as $2\mathcal{H}_{\mathrm{SP}}(P)$, though the index of P might be larger than 1. Therefore we have $J^{\{\alpha\}}(P) = N(P)\big(g_s(2\mathcal{H}_{\mathrm{SP}}(P))^{\{2\alpha\}}\big)(1 + o(1))$. Consider comparing an arbitrary configuration, $P^* \in \mathcal{D}^n$ against P_{mic}. We have

$$\frac{J^{\{\alpha\}}(P_{\mathrm{mic}})}{J^{\{\alpha\}}(P^*)} = \frac{N(P_{\mathrm{mic}})\left(g_s(2\mathcal{H}_{\mathrm{SP}}(P_{\mathrm{mic}}))^{\{2\alpha\}}\right)(1 + o(1))}{N(P^*)\left(g_s(2\mathcal{H}_{\mathrm{SP}}(P^*))^{\{2\alpha\}}\right)(1 + o(1))}.$$

If P^* is not a global solution of the multi-incenter problem, we have $\mathcal{H}_{\mathrm{SP}}(P_{\mathrm{mic}}) > \mathcal{H}_{\mathrm{SP}}(P^*)$, and since $g_s(\cdot)$ is decreasing this gives us

$$\lim_{k \to \infty} \frac{J^{\{\alpha\}}(P_{\mathrm{mic}})}{J^{\{\alpha\}}(P^*)} = 0.$$

If, on the other hand, P^* is a global solution of the multi-incenter problem, then, using the fact that P_{mic} has the lowest index among all of them, we deduce $\frac{J^{\{\alpha\}}(P_{\text{mic}})}{J^{\{\alpha\}}(P^*)} \leq 1+o(1)$.

■

The proof of the theorem can be reproduced for isolated local maximizers of the multi-incenter problem to arrive at the following result.

Corollary 3.3.8 *Let $P_{\text{mic}} \in \mathcal{D}^n$ be an isolated local maximizer of the multi-incenter problem. Then, as $k \to \infty$, P_{mic} asymptotically optimizes $\mathcal{E}^{\{\alpha\}}$, that is, $\mathcal{E}^{\{\alpha\}}(P_{\text{mic}})$ approaches a minimum.*

According to [18], under certain technical conditions, solutions to the multi-incenter problem are incenter Voronoi configurations.

3.3.3 Distributed coordination algorithms

Given the results in Theorems 3.3.5 and 3.3.7, it is of interest to design coordination algorithms that steer a network of mobile agents towards circumcenter and incenter Voronoi configurations. We do this following the exposition in [18]. In light of the results in Section 3.3.2, this enables the network to perform a spatial prediction which is asymptotically optimal as $k \to \infty$. Note that these algorithms are not intended to provide optimal trajectories for multiple sequential measurements. That problem is left for future work.

Let us assume each agent can move according to a first-order dynamical model $\dot{p}_i = u_i$, $i \in \{1, \ldots, n\}$. Consider the following coordination algorithms

$$\dot{p}_i = \text{CC}(V_i(P)) - p_i, \tag{3.4a}$$

$$\dot{p}_i \in \text{IC}(V_i(P)) - p_i, \tag{3.4b}$$

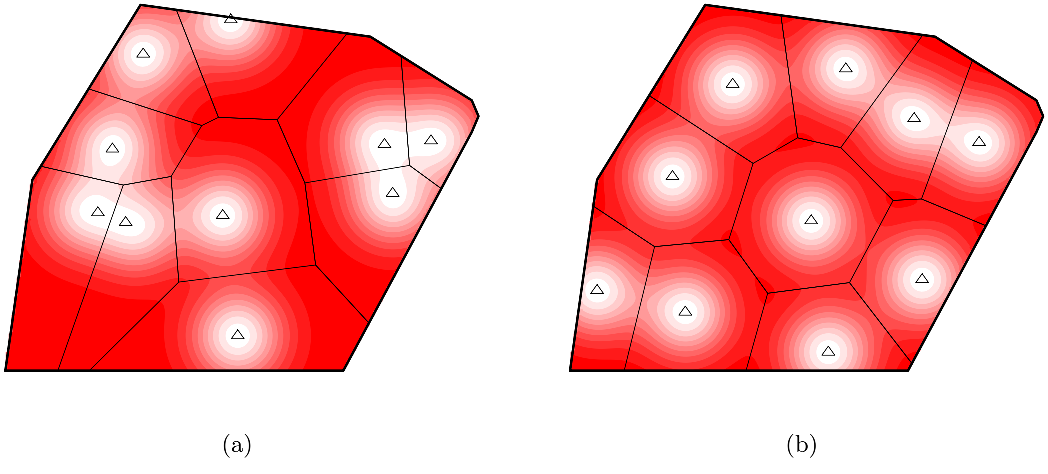
for each $i \in \{1, \ldots, n\}$. Note that (3.4b) is a differential inclusion. We understand its solutions in the Filippov sense [26]. Both coordination algorithms are Voronoi distributed, meaning that each agent only requires information from its Voronoi neighbors in order to execute its control law. The equilibrium points of the flow (3.4a) are the circumcenter Voronoi configurations, whereas the equilibrium points of the flow (3.4b) are incenter Voronoi configurations. Furthermore, the evolution of $\mathcal{H}_{\mathrm{DC}}$ along (3.4a) is monotonically decreasing, while the evolution of $\mathcal{H}_{\mathrm{SP}}$ along (3.4b) is monotonically increasing. The convergence properties of these coordination algorithms, as well as alternative flows with similar distributed properties that can also be used to steer the network to center Voronoi configurations, are studied in [18].

3.4 Simulations

We begin our simulation study with an example which demonstrates the optimality of circumcenter Voronoi partitions for the criterion, $\mathcal{M}$. Figure 3.1 shows the contours of the function $\mathrm{Var}_{\mathrm{SK}}[z(s); P]$ for an arbitrary configuration and a multi-circumcenter configuration. The number of agents was $n = 10$, and the domain was $\mathcal{D}$=$\{(0, .1), (2.5, .1), (3.45, 1.6), (3.5, 1.7), (3.45, 1.8), (2.7, 2.2), (1, 2.4), (0.2, 1.3)\}$. The covariance function used was the Gaussian,

$$g_s(\|s_1 - s_2\|) = e^{-\left(\frac{\|s_1-s_2\|}{.4}\right)^2}.$$

With the aim of illustrating the results presented in Section 3.3, we performed sim-


(a)

(b)

Figure 3.1: Contours of $\mathrm{Var}_{\mathbf{SK}}[z(s); P]$ for (a) an arbitrary configuration, and (b) a multi-circumcenter configuration. The correlation function used was Gaussian.

ulations for both objective functions $\mathcal{M}$ and $\mathcal{E}$ with $n = 5$ agents. In our simulations, we used as domain $\mathcal{D}$ the convex polygon with vertices $\{(0, 0.1), (2.5, 0.1), (3.45, 1.6), (3.5, 1.7), (3.45, 1.8), (2.7, 2.2), (1.0, 2.4), (0.2, 1.3)\}$ and as isotropic covariance the one defined via $g : \mathbb{R} \to \mathbb{R}$, $g_s(\|s_1 - s_2\|) = e^{-\frac{1}{5}\|s_1 - s_2\|}$. Note that the mean function, μ, does not play a role in determining the optimal network configurations. Figure 2.3 shows the multicenter configurations obtained with the flows (3.4).

3.4.1 Analysis of simulations for $\mathcal{M}^{\{\alpha\}}$

Using $\mathcal{M}^{\{1\}}$ we ran over 1000 random trials, each time running a gradient descent algorithm, and chose the local minimum configuration with the smallest value of $\mathcal{M}^{\{1\}}$ to be our approximation of a global minimum. From this configuration P_*, we generated a multi-circumcenter configuration using (3.4a), depicted in Figure 2.3(a). For increasing values of α, we ran a gradient descent of $\mathcal{M}^{\{\alpha\}}$ to find the best local configuration near P_*. We plotted $\mathcal{M}^{\{\alpha\}}$ as calculated with this new configuration against $\mathcal{M}^{\{\alpha\}}$ as calculated with the multi-circumcenter configuration. For comparison, we also plotted the performance of a random (static) configuration which was not a local minimum. Figure 3.2 illustrates the result in Theorem 3.3.5. We halt the experiment at around $\alpha = 15$ because the performance of the circumcenter Voronoi configuration becomes impossible to distinguish from the one of the minimizer of $\mathcal{M}^{\{\alpha\}}$ at this reso-

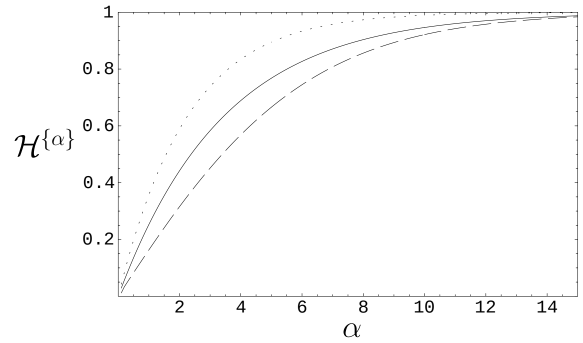


Figure 3.2: Value of $\mathcal{M}^{\{\alpha\}}$ for multi-circumcenter (solid), approximated global minimum (dashed) arrived at by gradient descent for each value of α, and random (dotted) configurations of 5 agents for increasing α. The covariance function is exponential.

lution.

3.4.2 Analysis of simulations for $\mathcal{E}^{\{\alpha\}}$

Using $\mathcal{E}^{\{1\}}$ we ran over 1000 random trials, each time running a gradient descent algorithm, and chose the local minimum configuration with the smallest value of $\mathcal{E}^{\{1\}}$ to be our approximation of a global minimum. From this configuration P_* we generated the multi-incenter configuration using (3.4b), depicted in Figure 2.3(b). For increasing values of α, we ran a gradient descent of $\mathcal{E}^{\{\alpha\}}$ to find the best local configuration near P_*. We plotted $\mathcal{E}^{\{\alpha\}}$ as calculated with this new configuration against $\mathcal{E}^{\{\alpha\}}$ as calculated with the multi-incenter configuration. For comparison, we also plotted the performance of a random (static) configuration which was not a local minimum. Figure 3.3 illustrates the result stated in Theorem 3.3.7.

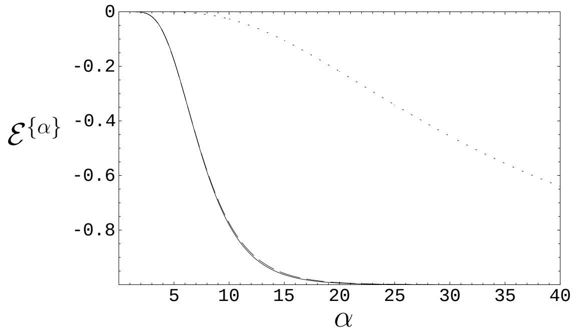


Figure 3.3: Value of $\mathcal{E}^{\{\alpha\}}$ for multi-incenter (solid), approximated global minimum (dashed) arrived at by gradient descent for each value of α, and random (dotted) configurations of 5 agents for increasing α. The covariance function is exponential. The performance of the global and multi-incenter configurations looks identical even though configurations are different at each α.

Remarkably, the performance of the incenter Voronoi configuration and the minimizer of $\mathcal{E}^{\{\alpha\}}$ are almost identical, even for low values of α. The numerical simulations suggest that multi-incenter Voronoi configurations are near-optimal for the extended prediction criterion.

The next chapter considers extension of these notions to trajectories of samples over time.

Chapter 4

Optimal trajectories for sampling dynamic fields under near independence

Here we summarize the work published in the conference paper [38], and the (submitted) follow-up paper [37]. In it, we extend the notion of geometric centering to optimize *trajectories* of samples taken by multiple agents over an interval of time. We consider the maximum predictive variance of the simple kriging predictor as optimality criterion. In the interest of clarity, we have placed most of the proofs in Appendix A.1.

4.1 Model assumptions

Assume that $n \in \mathbb{Z}_{>0}$ sensing agents take samples at each of a sequence of discrete timesteps $\{1, \ldots, k_{\max}\}$, with $k_{\max} \in \mathbb{Z}_{>0}$. Let $S_i = ({s_i}^{(1)}, \ldots, {s_i}^{(k_{\max})})^T \in \mathcal{D}^{k_{\max}}$ denote the spatial locations of samples taken over the course of the experiment by the ith agent, and let $S = (S_1^T, \ldots, S_n^T)^T \in (\mathcal{D}^{k_{\max}})^n$ denote the locations of all samples taken by the network. We use $I_{\text{samp}} = \{1, \ldots, n\} \times \{1, \ldots, k_{\max}\}$ to denote the set of index pairs into the sample vector. We refer often to vectors of elements indexed by both agent and timestep, such as the elements of S. To save space, we use the shorthand notation $({a_1}^{(1)}, \ldots, {a_n}^{(k_{\max})}) = ({a_1}^{(1)}, \ldots, {a_1}^{(k_{\max})}, \ldots, {a_n}^{(1)}, \ldots, {a_n}^{(k_{\max})})$. Let $Y =$

$(y_1{}^{(1)}, \ldots, y_n{}^{(k_{\max})})^T \in (\mathbb{R}^{k_{\max}})^n$ denote the values of all samples taken at locations S. As in Chapter 3, we assume that the correlation of z exhibits isotropy in the spatial dimensions, and that samples are corrupted with an i.i.d. error. Here, we make the additional assumption that the space-time correlation is separable, and we write,

$$\mathrm{Cor}[y_i{}^{(k)}, y_{i'}{}^{(k')}] = \begin{cases} 1 & \text{if } (i,k) = (i',k') \\ g_s(\|s_i{}^{(k)} - s_{i'}{}^{(k')}\|) g_t(k,k'), & \end{cases}$$

for correlation functions $g_s : \mathbb{R}_{\geq 0} \to (0, 1-\tau^2]$, and $g_t : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to [0,1]$. We assume that g_s is strictly decreasing and continuously differentiable with nonzero derivative except possibly at 0 (i.e., $g_s'(d) < 0$ for $d > 0$), and we assume that the sampling error term, τ^2 is strictly positive. Note the assumption that the image of the spatial correlation function is strictly nonzero. These assumptions include the popular exponential, Gaussian, and Matérn correlation functions [1]. Once again, we write the covariance matrix as $\sigma_\tau^2 \mathbf{K}_\tau$.

4.1.1 Objective function for spatial estimation

We consider the scenario where the robotic network is given a time frame $[1, k_{\max}]$, with $k_{\max} \in \mathbb{Z}_{>0}$, to sample the spatio-temporal process z. A natural objective is to design sampling trajectories in such a way as to minimize the uncertainty of an estimate of the field at time $k_{\max}$ generated from samples taken up to that time. Here, we consider an objective function inspired by the notion of G-optimality from optimal design [19, 67]. The *maximum predictive variance* $\mathcal{M} : (\mathcal{D}^{k_{\max}})^n \to \mathbb{R}$ of estimates made at time $k_{\max}$ over the region $\mathcal{D}$ is

$$\mathcal{M}(S) = \max_{s \in \mathcal{D}} \mathrm{Var}_{\mathrm{SK}}[z(s, k_{\max}); S] = \sigma_0^2 - \sigma_\tau^2 \min_{s \in \mathcal{D}} \left\{ \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} \right\}. \tag{4.1}$$

Note that $\mathcal{M}$ corresponds to a "worst-case scenario," where we consider locations in the domain at which the predictive variance of the LUMVE is maximal. Our goal is to find the sampling trajectories $S \in (\mathcal{D}^{k_{\max}})^n$ that minimize the objective function $\mathcal{M}$. Note that the simplified case of $k_{\max} = 1$ corresponds to the maximum predictive variance criterion of Chapter 3. The problem of trajectory optimization treated here is considerably more complex. We should also note that all of our results hold for predictions of the field made at times other than $k_{\max}$.

4.2 Optimal solutions under near-independence

The objective function $\mathcal{M}$ is nonconvex and nonsmooth. The problem of finding an explicit characterization for its optimizers is especially hard: even for $k_{\max} = 1$, the optimization of $\mathcal{M}$ is known to be NP-hard over discrete spaces [44]. In this section we consider instead the optimization of $\mathcal{M}$ under near-independence, much the same as we did in Chapter 3. Raising the correlation to the power $\alpha \in \mathbb{R}_{>0}$ is equivalent to considering the spatial and temporal correlation functions g_s^α and g_t^α. Note once again that the correlation function $(g_s g_t)^\alpha$ retains much of the shape of the original correlation function (e.g., smoothness, range, etc), so this analysis is helpful in determining the properties of the original problem as well. We define $\mathbf{K}_\tau^{\{\alpha\}}$, $\mathbf{k}_\tau^{\{\alpha\}}$, and $\mathcal{M}^{\{\alpha\}}$ as we did in Chapter 3, with the caveat that these are now functions of the vector of space-time samples.

Our objective is to characterize the asymptotic minimizers of $\mathcal{M}^{\{\alpha\}}$. To do so, we need to introduce a family of weighted distance measures based on correlation. Define $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ and $w : \{1, \dots, k_{\max}\} \to \mathbb{R}_{\geq 0}$ by,

$$\phi(d) = -\log(g_s(d)), \qquad w(k) = -\log(g_t(k_{\max}, k)).$$

The function w gives a weight which depends on the temporal correlation between sample time k and predictive time $k_{\max}$. The function ϕ is strictly increasing and continuously differentiable with strictly positive derivative except possibly at zero. It therefore admits an inverse, $\phi^{-1} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. The correlation between a sample at step k and prediction at step $k_{\max}$ induces the weighted distance function, $\delta_k : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_{\geq 0}$,

$$\delta_k(s_1, s_2) = \phi(\|s_1 - s_2\|) + w(k). \tag{4.2}$$

We refer to δ_k as the *correlation distance* associated with sample time k, and note that $\delta_k(s, {s_i}^{(k)}) = -\log\big(g_s(\|s - {s_i}^{(k)}\|)g_t(k_{\max}, k)\big)$. The following result classifies its level sets.

Lemma 4.2.1 (Correlation level sets) *For each $k \in \{1, \ldots, k_{max}\}$, $s \in \mathcal{D}$ and $c \in \mathbb{R}$, one has $\Omega_{lvl}(s' \mapsto \delta_k(s', s), c) = \mathrm{bnd}\left(\overline{B}(s, r_k(c))\right)$, where $r_k : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, defined by*

$$r_k(c) = \begin{cases} \phi^{-1}(c - w(k)) & \textit{if} c \geq w(k), \\ 0 & \textit{otherwise}, \end{cases}$$

is strictly increasing and continuously differentiable on the interval $(w(k), \infty)$, with derivative ${r_k}'(c) = \frac{1}{\phi'(r_k(c))}$.

We are interested in those samples with smallest correlation distance to a given predictive location. Note that this is equivalent to the samples with highest correlation to the predictive location. We must therefore consider the possibility of samples with identical correlation to all predictive locations. Let S_{unique} be the following set of possible trajectories, which ensures the spatio-temporal uniqueness of any samples that achieve the maximal correlation distance from any predictive location,

$$\begin{aligned} S_{\text{unique}} = \Big\{ S = ({s_1}^{(1)}, \ldots, {s_n}^{(k_{\max})})^T \in (\mathcal{D}^{k_{\max}})^n \mid \\ \nexists (i,k) \neq (j,l) \in I_{\text{samp}} \text{ and } s \in \mathcal{D}, \text{ s.t.} \\ \delta_k(s, {s_i}^{(k)}) = \min_{(i',k') \in I_{\text{samp}}} \delta_k(s, {s_{i'}}^{(k')}) \text{ and} \\ \delta_k(s', {s_i}^{(k)}) = \delta_l(s', {s_j}^{(l)}),\ \forall s' \in \mathcal{D} \Big\}. \end{aligned}$$

Note that for samples ${s_i}^{(k)}$ and ${s_j}^{(l)}$ to have identical correlation distance to all predictive locations requires that ${s_i}^{(k)} = {s_j}^{(l)}$ *and* $g_t(k_{\max}, k) = g_t(k_{\max}, l)$. We are now ready to characterize the minimizers of $\mathcal{M}^{\{\alpha\}}$ as α grows.

Theorem 4.2.2 (Global minimizers of $\mathcal{M}$ under near-independence) *Let $\mathcal{H} : (\mathcal{D}^{k_{max}})^n \to \mathbb{R}$ denote the* correlation distance disk-covering *function, defined by*

$$\mathcal{H}(S) = \max_{s \in \mathcal{D}} \Big\{ \min_{(i,k) \in I_{\text{samp}}} \{\delta_k(s, {s_i}^{(k)})\} \Big\}. \tag{4.3}$$

For $\Omega \subset (\mathcal{D}^{k_{max}})^n$ compact, let $S_{mcc} \in \Omega$ be a global minimizer of the correlation disk-covering function $\mathcal{H}$ over Ω. Further assume that $S_{mcc} \in S_{unique}$. Then, as $\alpha \to \infty$, S_{mcc} asymptotically globally optimizes $\mathcal{M}^{\{\alpha\}}$ over Ω, that is, $\mathcal{M}^{\{\alpha\}}(S_{mcc})$ approaches a global minimum over Ω.

The proof of the theorem can be reproduced for local minimizers of $\mathcal{H}$ over Ω to arrive at the following result.

Corollary 4.2.3 (Local minimizers of $\mathcal{M}$ under near-independence) *For $\Omega \subset (\mathcal{D}^{k_{max}})^n$ compact, let $S_{mcc} \in \Omega$ be a local minimizer of the correlation disk-covering function $\mathcal{H}$ over Ω. Then, as $\alpha \to \infty$, S_{mcc} asymptotically locally optimizes $\mathcal{M}^{\{\alpha\}}$ over Ω, that is, $\mathcal{M}^{\{\alpha\}}(S_{mcc})$ approaches a local minimum over Ω.*

The generality of the subspace Ω in Theorem 4.2.2 and Corollary 4.2.3 also allows us to apply the result to two situations of particular importance. First, we may restrict the samples to feasible trajectories based on vehicular movement limitations, and the initial positions of the vehicles, which we will call *anchor points.* This amounts to a restriction on each agent trajectory, and we define the range-based constraint set, $\Omega_{\mathrm{Rg}} \subset (\mathcal{D}^{k_{\max}})^n$ as, $\Omega_{\mathrm{Rg}} = \prod_{i=1}^{n} \Omega_{\mathrm{Rg}_i}$, where

$$\begin{aligned}\Omega_{\mathrm{Rg}_i} = \big\{({s_i}^{(1)}, \dots, {s_i}^{(k_{\max})})^T \in \mathcal{D}^{k_{\max}} \bigm| \|{s_i}^{(1)} - p_i(0)\| \leq u_{\max} \text{ and} \\ \|{s_i}^{(k)} - {s_i}^{(k-1)}\| \leq u_{\max},\ \forall k \in \{2, \dots, k_{\max}\}\big\}. \end{aligned} \tag{4.4}$$

Our results also hold for a more general problem, optimizing over all $P(0) \in \mathcal{D}^n$, however this setup is directed at online path planning where the benefits of distributed implementation shine. Second, a change in mission parameters at time $k-1$, $k \in \{2, \dots, k_{\max}\}$, might prompt optimization over just those locations not yet sampled, i.e., ${\Omega_{\mathrm{Rg}}}^{(\geq k)} = \prod_{i=1}^{n} {\Omega_{\mathrm{Rg}_i}}^{(\geq k)}$, where

$$\begin{aligned}{\Omega_{\mathrm{Rg}_i}}^{(\geq k)} = \big\{({s_i}^{(k)}, \dots, {s_i}^{(k_{\max})})^T \in \mathcal{D}^{k_{\max}-k+1} \bigm| \|{s_i}^{(k)} - p(k-1)\| \leq u_{\max} \text{ and} \\ \|{s_i}^{(k')} - {s_i}^{(k'-1)}\| \leq u_{\max},\ \forall k' \in \{k+1, \dots, k_{\max}\}\big\}. \end{aligned} \tag{4.5}$$

For ease of notation, we assume that these decisions and path adjustments are made at sample time instants, and thus the anchor points for optimization over ${\Omega_{\mathrm{Rg}_i}}^{(\geq k)}$ are the sample locations at step $k-1$, but the process is easily extensible to optimization between sample times.

Theorem 4.2.2 shows that the optimization of the maximum predictive variance is equivalent to a geometric optimization problem in the near-independence range. This remarkable result allows us to turn the search for the optimizers of $\mathcal{M}^{\{\alpha\}}$ into the search for the optimizers of the correlation disk-covering function $\mathcal{H}$ defined in (4.3). This is what we tackle in the following sections.

4.3 Maximal correlation partition

In this section, we introduce the maximal correlation partition associated to a network trajectory. This partition will be instrumental in determining the optimizers of $\mathcal{H}$. In the context of this work, a partition of $\mathcal{D}$ is a collection of compact subsets, $\mathcal{W} = \{{W_1}^{(1)}, \dots, {W_n}^{(k_{\max})}\}$ with disjoint interiors whose union is $\mathcal{D}$. For any $S \in S_{\text{unique}}$, let $\mathcal{MC}(S) = ({\mathrm{MC}_1}^{(1)}(S), \dots, {\mathrm{MC}_n}^{(k_{\max})}(S))$ denote the *maximal correlation partition* defined by

$${\mathrm{MC}_i}^{(k)}(S) = \big\{ s \in \mathcal{D} \,\big|\, \delta_k(s, {s_i}^{(k)}) \leq \delta_l(s, {s_j}^{(l)}),\ \forall (j,l) \neq (i,k) \big\}. \tag{4.6}$$

This partition corresponds to a generalized Voronoi partition [63] for distance measure ϕ and weights given by w. In general, the maximal correlation regions are neither convex nor star-shaped. Note that, depending on the weights and locations, ${\mathrm{MC}_i}^{(k)}(S)$ might be empty for some i. Let $\mathrm{I} : \mathfrak{P}(\mathcal{D}) \to \{1, \dots, n * k_{\max}\}$ map a partition to the number of nonempty cells it contains, which we term the *index* of the partition. The following lemma gives some special cases in which $\mathcal{MC}$ is equal to distance-based partitions known in the literature, see e.g., [63, 20].

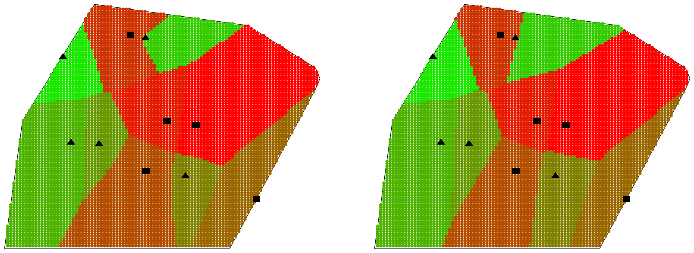
Lemma 4.3.1 (Special cases of $\mathcal{MC}$) *The maximal correlation partition $\mathcal{MC}(S)$ corresponds to*

- *the Voronoi partition of $\mathcal{D}$ with generators S, if all weights are equal,*

- *the power diagram, if the spatial correlation is the Gaussian, $g_s(d) = e^{-\alpha d^2}$, with $\alpha \in \mathbb{R}_{>0}$,*

- *the additively weighted Voronoi partition, if the spatial correlation is the exponential, $g_s(d) = e^{-\alpha d}$, with $\alpha \in \mathbb{R}_{>0}$.*

Figure 4.1 illustrates the latter two types of partitions. For $S \in S_{\text{unique}}$, the correlation distance disk-covering function can be restated in terms of the maximal correlation partition as,



(a) (b)

Figure 4.1: Examples of maximal correlation partition in which each cell is defined by the predictive locations with highest (a) exponential correlation and (b) Gaussian correlation to a given (generating) sample. In both cases, two timesteps are shown. Samples taken at step 1 are shown as filled triangles, those taken at step 2 are shown as filled boxes.

$$\mathcal{H}(S) = \max_{(i,k) \in I_{\text{samp}}} \Big\{ \max_{s \in \mathrm{MC}_i{}^{(k)}(S)} \{\delta_k(s, s_i{}^{(k)})\} \Big\}. \tag{4.7}$$

This expression is important because it clearly shows how $\mathcal{H}$ has a double dependence on the network trajectory S: through the value of the correlation distance and through the maximal correlation partition. This motivates us to define an extension of $\mathcal{H}$ as follows: for a given sample vector $S \in (\mathcal{D}^{k_{\max}})^n$ and a partition $W = \{W_1{}^{(1)}, \dots, W_n{}^{(k_{\max})}\} \subset \mathfrak{P}(\mathcal{D})$ of the predictive space, define $\mathcal{H}_{\mathcal{W}} : (\mathcal{D}^{k_{\max}})^n \to \mathbb{R}$ by

$$\mathcal{H}_{\mathcal{W}}(S) = \max_{\substack{(i,k) \in I_{\text{samp}} \\ W_i{}^{(k)} \neq \emptyset}} \Big\{ \max_{s \in W_i{}^{(k)}} \{\delta_k(s, s_i{}^{(k)})\} \Big\}. \tag{4.8}$$

Note that if $S \in S_{\text{unique}}$, then $\mathcal{H}(S) = \mathcal{H}_{\mathcal{MC}(S)}(S)$. This function is particularly useful in our search for the optimizers of $\mathcal{H}$ because it allows us to decouple the two dependencies of this function on the network trajectory. The following result characterizes the maximal correlation partition as the optimal partition for $\mathcal{H}_{\mathcal{W}}$ given a fixed network trajectory.

Proposition 4.3.2 ($\mathcal{H}$-optimality of the maximal correlation partition) *For any $S \in S_{unique}$ and any partition $\mathcal{W} \subset \mathfrak{P}(\mathcal{D})$ of $\mathcal{D}$ with $\mathrm{I}(\mathcal{W}) \leq \mathrm{I}(\mathcal{MC}(S))$,*

$$\mathcal{H}(S) \leq \mathcal{H}_{\mathcal{W}}(S), \tag{4.9}$$

that is, the maximal correlation partition $\mathcal{MC}(S)$ is optimal for $\mathcal{H}$ among all partitions of $\mathcal{D}$ of less than or equal index.

Proposition 4.3.2 implies that, in order to fully characterize the optimizers of $\mathcal{H}$, it is sufficient to characterize the optimizers of $\mathcal{H}_{\mathcal{W}}$ for a fixed arbitrary partition. The latter formulation is advantageous because of the single dependence of the value of $\mathcal{H}_{\mathcal{W}}$ on the network trajectory.

4.4 Unconstrained optimal trajectories for a given partition

In this section, our objective is to characterize the optimal network trajectories of $\mathcal{H}_{\mathcal{W}}$ for a fixed partition $\mathcal{W} = \{{W_1}^{(1)}, \dots, {W_n}^{(k_{\max})}\} \subset \mathfrak{P}(\mathcal{D})$ of $\mathcal{D}$. We will find it useful to start our analysis with the simplified problem of locating a single sample to minimize the maximum correlation distance to a single predictive region. We will then build on this analysis to tackle the more complex multiple sample problem.

4.4.1 Single sample unconstrained problem

For $(i, k) \in I_{\text{samp}}$ with ${W_i}^{(k)} \neq \emptyset$, consider the task of choosing where R_i should take the sample at time k. Let ${\mathrm{MCD}_i}^{(k)} : \mathcal{D} \to \mathbb{R}_{>0}$ be defined as,

$$ {\mathrm{MCD}_i}^{(k)}(s) = \max_{s' \in {W_i}^{(k)}} \delta_k(s', s). \tag{4.10} $$

Note that ${\mathrm{MCD}_i}^{(k)}$ corresponds to $\mathcal{H}_{\mathcal{W}}$ for a single agent and single sample at timestep k. For any $s \in \mathcal{D}$, it is important to note that the maximum correlation distance, ${\mathrm{MCD}_i}^{(k)}(s)$ is attained at the same locations in ${W_i}^{(k)}$ as the maximum Euclidean distance, i.e.,

$$ \operatorname*{argmax}_{s' \in {W_i}^{(k)}} \delta_k(s', s) = \operatorname*{argmax}_{s' \in {W_i}^{(k)}} \|s' - s\|. $$

In the next result, which follows from Lemma 4.2.1, we characterize the sublevel sets of ${\mathrm{MCD}_i}^{(k)}$.

Lemma 4.4.1 (Sublevel sets of MCD) *For any $c \in \mathbb{R}_{\geq 0}$, the set $\Omega_{sublvl}({\mathrm{MCD}_i}^{(k)}, c)$ is closed, bounded, and strictly convex.*

Figure 4.2 shows a two-dimensional example of the level sets of ${\mathrm{MCD}_i}^{(k)}$. The following result on the generalized gradient of the maximum correlation distance function makes use of [14, Theorem 2.1] and [15, Theorem 2.3.9].

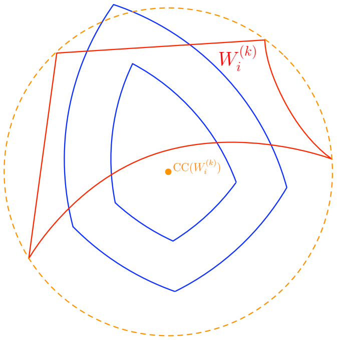


Figure 4.2: A two-dimensional example of the level sets of ${\mathrm{MCD}_i}^{(k)}$. The dashed circle is the circumcircle. The closed curves around the circumcenter represent two different level sets of ${\mathrm{MCD}_i}^{(k)}$.

Lemma 4.4.2 (Smoothness of $\mathbf{MCD}_i{}^{(k)}$) *The* $\mathrm{MCD}_i{}^{(k)}$ *is locally Lipschitz and regular, and its generalized gradient takes the form*

$$\partial \mathrm{MCD}_i{}^{(k)}(s) = \mathrm{co}\left\{ \phi'(\mathrm{d}_{\max}(s, W_i{}^{(k)}))\,\mathrm{vrs}(s - s') \mid s' \in \underset{s^* \in W_i{}^{(k)}}{\mathrm{argmax}}\, \delta_k(s^*, s) \right\}.$$

We next characterize the minimizers of $\mathrm{MCD}_i{}^{(k)}$.

Proposition 4.4.3 ($\mathrm{CC}(W_i{}^{(k)})$ minimizes $\mathbf{MCD}_i{}^{(k)}$) *The function* $\mathrm{MCD}_i{}^{(k)}$ *has a global minimum at* $\mathrm{CC}(W_i{}^{(k)})$ *and no other critical points.*

Remark 4.4.4 (Interpretation of Proposition 4.4.3) Note that Proposition 4.4.3 implies that the circumcenter minimizes the maximum Euclidean distance to an arbitrary set. •

4.4.2 Multiple sample unconstrained problem

Here, we use the results of Section 4.4.1 to tackle the multiple sample problem, i.e., the characterization of the optima of the network objective $\mathcal{H}_{\mathcal{W}}$. We can equivalently write (4.8) as

$$\mathcal{H}_{\mathcal{W}}(S) = \max_{\substack{(i,k) \in I_{\mathrm{samp}} \\ W_i{}^{(k)} \neq \emptyset}} \mathrm{MCD}_i{}^{(k)}(s_i{}^{(k)}).$$

The following result on the generalized gradient of $\mathcal{H}_{\mathcal{W}}$ follows from using Lemma 4.4.2 and [15, Proposition 2.3.12] on this expression.

Lemma 4.4.5 (Smoothness of $\mathcal{H}_{\mathcal{W}}$) *The function $\mathcal{H}_{\mathcal{W}}$ is locally Lipschitz and regular, and its generalized gradient takes the form*

$$\partial \mathcal{H}_{\mathcal{W}}(S) = \mathrm{co}\left\{ \partial \mathrm{MCD}_i{}^{(k)}(S) \mid (i,k) \in I_{\mathrm{samp}} \ s.t.\ \mathrm{MCD}_i{}^{(k)}(S) = \mathcal{H}_{\mathcal{W}}(S) \right\},$$

where, with a slight abuse of notation, we use $\mathrm{MCD}_i{}^{(k)}(S)$ *to denote the map* $S \mapsto \mathrm{MCD}_i{}^{(k)}(s_i{}^{(k)})$.

In order to extend Proposition 4.4.3 to the multiple sample case, we first need to introduce a piece of notation to account for the possibility of empty regions in the maximal correlation partition. Let $\overline{\mathrm{CC}} : \mathfrak{P}(\mathbb{R}^d) \times \mathbb{R}^d \to \mathbb{R}^d$ be defined by

$$\overline{\mathrm{CC}}(W, s) = \begin{cases} \mathrm{CC}(W) & \text{if } W \neq \emptyset, \\ s & \text{otherwise.} \end{cases}$$

Let $\overline{\mathrm{CC}}(\mathcal{W}, S) = \big(\overline{\mathrm{CC}}({W_1}^{(1)}, {s_1}^{(1)}), \dots, \overline{\mathrm{CC}}({W_n}^{(k_{\max})}, {s_n}^{(k_{\max})})\big)^T$ denote a vector of such circumcenter locations. We are now ready to state a generalization of Proposition 4.4.3.

Proposition 4.4.6 ($\mathcal{H}_{\mathcal{W}}$-optimal trajectories) *Let $\mathcal{W} = \{{W_1}^{(1)}, \dots, {W_n}^{(k_{max})}\} \subset \mathfrak{P}(\mathcal{D})$ denote a partition of $\mathcal{D}$. For any trajectories $S = ({s_1}^{(1)}, \dots, {s_n}^{(k_{max})})^T \in S_{unique}$, and $\tilde{S} = ({\tilde{s}_1}^{(1)}, \dots, {\tilde{s}_n}^{(k_{max})})^T \in (\mathcal{D}^{k_{max}})^n$,*

$$\mathcal{H}_{\mathcal{W}}\big(\overline{\mathrm{CC}}(\mathcal{W}, \tilde{S})\big) \leq \mathcal{H}_{\mathcal{W}}(S), \tag{4.11}$$

that is, the circumcenter locations $\overline{\mathrm{CC}}(\mathcal{W}, \tilde{S})$ are optimal for $\mathcal{H}_{\mathcal{W}}$ among all network trajectories.

Note the duality between the results in Proposition 4.3.2 (for a fixed network configuration, the maximal correlation partition is optimal) and Proposition 4.4.6 (for a fixed partition, the circumcenter locations are optimal). The combination of these two results allow us provide the following characterization of the optimizers of the correlation disk-covering function $\mathcal{H}$.

Proposition 4.4.7 (Generalized multicircumcenter trajectories optimize $\mathcal{H}$) *Consider $S = ({s_1}^{(1)}, \dots, {s_n}^{(k_{max})})^T \in (\mathcal{D}^{k_{max}})^n$ such that ${s_i}^{(k)} = \mathrm{CC}\big({MC_i}^{(k)}(S)\big)$ for each $(i, k) \in I_{\mathrm{samp}}$ with ${MC_i}^{(k)}(S) \neq \emptyset$. Then S is a local minimizer of $\mathcal{H}$ over $(\mathcal{D}^{k_{max}})^n$. We call such a network trajectory a* generalized multicircumcenter trajectory. *Furthermore, if $\mathrm{I}(\mathcal{MC}(S)) = n * k_{max}$, then S is a global minimizer of $\mathcal{H}$ over $(\mathcal{D}^{k_{max}})^n$.*

4.5 Range-constrained optimal trajectories for a given partition

In this section, our objective is to characterize the optimizers of $\mathcal{H}_{\mathcal{W}}$ over Ω_{Rg} for a fixed partition $\mathcal{W}$. We begin our discussion by providing a useful alternative expression for $\mathcal{H}_{\mathcal{W}}$. Let $\mathcal{W}_i = \{{W_i}^{(1)}, \dots, {W_i}^{(k_{\max})}\}$ denote the elements of the partition $\mathcal{W}$ assigned to the samples in the trajectory of R_i. With a slight abuse of notation, we may write

$$\mathcal{H}_{\mathcal{W}}(S) = \max_{\substack{i \in \{1,\dots,n\} \\ \mathcal{W}_i \neq \emptyset}} \mathcal{H}_{\mathcal{W}_i}(S_i), \quad \text{where} \quad \mathcal{H}_{\mathcal{W}_i}(S_i) = \max_{\substack{k \in \{1,\dots,k_{\max}\} \\ {W_i}^{(k)} \neq \emptyset}} \big\{ {\mathrm{MCD}_i}^{(k)}({s_i}^{(k)}) \big\}.$$

The condition $\mathcal{W}_i \neq \emptyset$ indicates that there is at least one nonempty ${W_i}^{(k)} \in \mathcal{W}_i$. The above expression clearly shows that, for a fixed partition, minimizing $\mathcal{H}_{\mathcal{W}}$ over the space of network trajectories is equivalent to (independently) minimizing each of the functions $\mathcal{H}_{\mathcal{W}_i}$ over the space of trajectories of the robot R_i. As a consequence, we structure our discussion in three parts. First, we deal with the single sample problem. Then, we build on this discussion to address the problem of finding an optimal sampling trajectory for a *single* agent. Finally, we combine individual agent trajectories into a network trajectory to find the constrained optimizers of $\mathcal{H}_{\mathcal{W}}$.

4.5.1 Single sample constrained problem

Proposition 4.4.3 allows a simple, geometric interpretation of the minimizer of ${\mathrm{MCD}_i}^{(k)}$. Our objective here is to obtain a similar characterization for the range-constrained problem. We first consider the single sample problem over a general closed convex constraint set.

Proposition 4.5.1 (Constrained minimizers of ${\mathrm{MCD}_i}^{(k)}$) *Assume that ${W_i}^{(k)} \neq \emptyset$. Let $\Gamma \subset \mathbb{R}^d$ be closed and convex. Then a point $s^* \in \Gamma$ is the unique minimizer of ${\mathrm{MCD}_i}^{(k)}$ over Γ if and only if $\mathbf{0} \in \partial {\mathrm{MCD}_i}^{(k)}(s^*) + N_\Gamma(s^*)$.*

Let us now specify the range based constraint set for ${s_i}^{(k)}$. The *set of constraining locations* of $(i,k) \in I_{\text{samp}}$ are the locations of robot R_i at sample times $k-1$ and $k+1$,

$$S_{\text{cs}}(k, S_i) = \left\{p(k') \mid k' \in K_{\text{cs}}(k)\right\}, \text{ where } K_{\text{cs}}(k) = \{k-1, k+1\} \cap \{0, \dots, k_{\max}\}.$$

Note that in all but the initial anchor point, this set corresponds to the sample locations immediately preceding and following the (i,k)th sample. Let $\Gamma^{(k)} : \mathcal{D}^{k_{\max}} \to \mathfrak{P}(\mathbb{R}^d)$ map a network trajectory to the intersection of $u_{\max}$-balls centered at locations in the set of constraining locations of (i,k), i.e.,

$$\Gamma^{(k)}(S_i) = \bigcap_{s \in S_{\text{cs}}(k,S_i)} \overline{B}(s, u_{\max}). \tag{4.12}$$

The set $\Gamma^{(k)}(S_i)$ corresponds to Ω_{Rg} with all other samples fixed in space. Restricting ${S_i}^{(k)}$ to $\Gamma^{(k)}(S_i)$ ensures that R_i does not violate the maximum distance requirement $u_{\max}$.

In order to state the main result of this section, we will find it useful to introduce an extension of the predictive set ${W_i}^{(k)}$ which incorporates the position of sample (i,k) relative to $\Gamma^{(k)}(S_i)$. To that end, define $\mathrm{EPt}^{(k:k')} : \mathcal{D}^{k_{\max}} \to \mathbb{R}^d$, $(i,k) \in I_{\text{samp}}$, $k' \in K_{\text{cs}}(k)$ by

$$\mathrm{EPt}^{(k:k')}(S_i) = {s_i}^{(k)} + r_k(\mathcal{H}_{\mathcal{W}_i}(S_i)) \frac{{s_i}^{(k')} - {s_i}^{(k)}}{u_{\max}}, \tag{4.13}$$

The reason for the use of $\mathcal{H}_{\mathcal{W}_i}(S_i)$ will be made apparent in Section 4.5.2. For now, it is only important that $\mathcal{H}_{\mathcal{W}}(S_i) \geq \mathrm{MCD}_i{}^{(k)}({s_i}^{(k)})$. The location $\mathrm{EPt}^{(k:k')}(S_i)$ can be seen as the projection of ${s_i}^{(k')}$ onto the surface of $\overline{B}({s_i}^{(k)}, r_k(\mathcal{H}_{\mathcal{W}_i}(S_i))\frac{\|{s_i}^{(k')} - {s_i}^{(k)}\|}{u_{\max}})$. Then, we extend the predictive set by the extended constraint points as follows. Let $\widetilde{W_i}^{(k)} : \mathcal{D}^{k_{\max}} \to \mathfrak{P}(\mathbb{R}^d)$, $(i,k) \in I_{\text{samp}}$ be the *constraint extended predictive set*,

$$\widetilde{W_i}^{(k)}(S_i) = \operatorname{co}\left({W_i}^{(k)}, \left\{\mathrm{EPt}^{(k:k')}(S_i) \mid k' \in K_{\text{cs}}(k)\right\}\right).$$

A point $s \in \widetilde{W_i}^{(k)}(S_i)$ is *active in centering* if there is no neighborhood of s which might be added to $\widetilde{W_i}^{(k)}(S_i)$ without changing the circumcenter. It can be seen from (4.13) that $\mathrm{EPt}^{(k:k')}(S_i)$ is active in centering if and only if the relation holds,

$$r_k(\mathcal{H}_{\mathcal{W}_i}(S_i))\frac{\|{s_i}^{(k)} - {s_i}^{(k')}\|}{u_{\max}} \geq r_k\big(\mathrm{MCD}_i{}^{(k)}({s_i}^{(k)})\big).$$

Figure 4.3 shows an example of the extended predictive set.

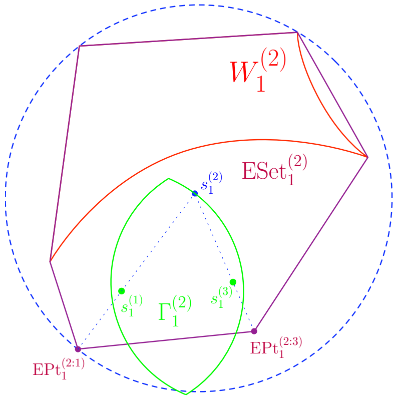


Figure 4.3: A two-dimensional example of the extended center representation of a critical point of the constrained problem. The dashed circle is the circumcircle of $\widetilde{W}_1{}^{(2)}$, with circumcenter ${s_1}^{(2)}$. Note that ${s_1}^{(2)}$ is on the boundary of $\Gamma^{(2)}$ formed by ${s_1}^{(1)}$, and thus EPt$^{(2:1)}$ is active in centering.

The next result gives a geometric interpretation of the constrained optimum in terms of $\widetilde{W}$.

Proposition 4.5.2 (Extended circumcenter minimizes $\mathrm{MCD}_i{}^{(k)}$ over $\Gamma^{(k)}(S_i)$) *Assume that $\Gamma^{(k)}(S_i)$ and ${W_i}^{(k)}$ are nonempty. Further assume that the scaling factor for the extended constraints satisfies $\mathcal{H}_{\mathcal{W}_i}(S_i) = \mathrm{MCD}_i{}^{(k)}({s_i}^{(k)})$. Then ${s_i}^{(k)}$ is the unique minimizer of* $\mathrm{MCD}_i{}^{(k)}$ *over* $\Gamma^{(k)}(S_i)$ *iff* ${s_i}^{(k)} = \mathrm{CC}\left(\widetilde{W}_i{}^{(k)}(S_i)\right)$.

4.5.2 Multiple sample single agent constrained problem

Here we extend the constrained solution above to a single agent optimizing its own trajectory. and characterize the optima of $\mathcal{H}_{\mathcal{W}_i}$ over the constraint set Ω_{Rg_i} defined in (4.4) in terms of *centered* sub-sequences. In order to facilitate discussion of generalized gradients, let $\mathrm{d}^{(k:k')} : \mathcal{D}^{k_{\max}} \to \mathbb{R}_{\geq 0}$, $k, k' \in \{1, \dots, k_{\max}\}$ be defined as $\mathrm{d}^{(k:k')}(S_i) = \|{s_i}^{(k)} - {s_i}^{(k')}\|$, and let $\mathrm{d}^{(1:0)}(S_i) = \mathrm{d}\,01(S_i) = \|{s_i}^{(1)} - p_i(0)\|$. With a slight abuse of notation, we use

$$\widetilde{W}_i{}^{(k)}(S_i; K_C) = \mathrm{co}\left(W_i{}^{(k)}, \left\{\mathrm{EPt}^{(k:k')}(S_i) \mid k' \in K_{\mathrm{cs}}(k) \cap K_C\right\}\right).$$

to denote constraint extended sets as calculated with a subset of the constraint points.

Lemma 4.5.3 (Centered sequences satisfy range constraint) *Let $S_i \in \mathcal{D}^{k_{max}}$, and let $K_C \subseteq \{1, \ldots, k_{max}\}$ define a sequence of consecutive samples from S_i such that each is at the circumcenter of the extended set formed by consecutive neighbors in the sequence, i.e.,*

$$s_i{}^{(k)} = \mathrm{CC}\left(\widetilde{W}_i{}^{(k)}(S_i; \{0\} \cup K_C)\right), \quad \textit{for all } k \in K_C,$$

Then $\mathrm{d}^{(k:k')}(S_i) \leq u_{max}$, for all $k \in K_C$ and $k' \in (\{0\} \cup K_C) \cap K_{\mathrm{cs}}(k)$. We call such a sequence centered.

Figure 4.4 shows an example of a centered sequence.

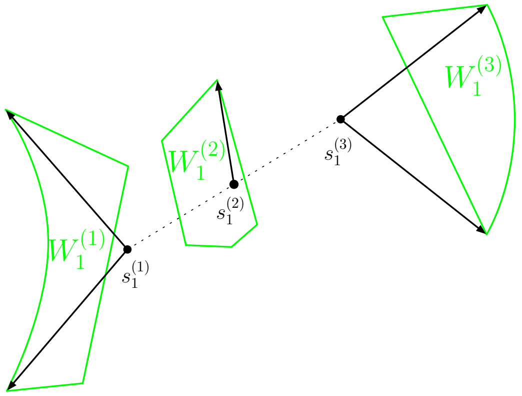


Figure 4.4: Two-dimensional three sample example of a centered sequence. The solid arrows show the directions from the sample to the farthest points in the associated predictive region. For illustrative purposes, we have used a correlation distance equivalent to Euclidean distance.

In the unconstrained case, optimizing $\mathcal{H}_{\mathcal{W}}$ takes the form of centering each sample within its predictive region, which may be characterized in terms of the generalized gradient of MCD. Given our discussion for the single sample constrained problem, in particular Proposition 4.5.2, we next characterize the gradient of the maximum correlation distance to the *extended* predictive region, $\widetilde{W}$, and thereby the optimal agent trajectories in terms of centered sequences. We begin with a result on the effect of the trajectory on the constraint extended predictive sets.

Lemma 4.5.4 (Correlation distance to extended constraints) *Let* $(i,k) \in I_{\text{samp}}$ *and* $k' \in K_{\text{cs}}(k)$*, and let* $S_i \in \mathcal{D}^{k_{max}}$ *such that* ${s_i}^{(k)} \neq {s_i}^{(k')}$*. Let* $\mathrm{CDE}_i{}^{(k:k')} : \mathcal{D}^{k_{max}} \to \mathbb{R}$ *be defined by*

$$\mathrm{CDE}_i{}^{(k:k')}(S_i) = \delta_k({s_i}^{(k)}, \mathrm{EPt}^{(k:k')}(S_i)).$$

The function $\mathrm{CDE}_i{}^{(k:k')}$ *is locally Lipschitz and regular near* S_i*, and its generalized gradient at* S_i *takes the form*

$$\partial \mathrm{CDE}_i{}^{(k:k')}(S_i) = \frac{\phi'(\|\mathrm{EPt}^{(k:k')}(S_i) - {s_i}^{(k)}\|)}{u_{max}} \times \\ \times \Big(r_k(\mathcal{H}_{\mathcal{W}_i}(S_i)) \partial\, \mathrm{d}^{(k:k')}(S_i) + \frac{\mathrm{d}^{(k:k')}(S_i)}{\phi'(r_k(\mathcal{H}_{\mathcal{W}_i}(S_i)))} \partial \mathcal{H}_{\mathcal{W}_i}(S_i) \Big).$$

In the expression of the gradient of $\mathrm{CDE}_i{}^{(k:k')}$ where $k' \neq 0$, note that since ${s_i}^{(k)} \neq {s_i}^{(k')}$, the set $\partial\, \mathrm{d}^{(k:k')}(S_i)$ consists of a single vector whose only nonzero components are the kth and k'th entries. Likewise $\partial\, \mathrm{d}^{(1:0)}(S_i)$ is nonzero only in the first entry.

We next characterize the function which maps the maximum correlation from a sample to any point in its constraint extended predictive set.

Lemma 4.5.5 (Extended set correlation distance) *Let* $(i,k) \in I_{\text{samp}}$ *and let the function* $\mathrm{MCD}_{\widetilde{W}}{}^{(k)} : \mathcal{D}^{k_{max}} \to \mathbb{R}$ *map the ith trajectory to the maximum correlation distance from* ${s_i}^{(k)}$ *to the corresponding constraint extended predictive set, i.e.,*

$$\mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i) = \max_{s \in \widetilde{W_i}^{(k)}(S_i)} \delta_k\big(s, {s_i}^{(k)}\big).$$

Further assume that either ${W_i}^{(k)} \neq \emptyset$, or *there is an* $s \in S_{\text{cs}}(k, S_i)$ *such that* ${s_i}^{(k)} \neq s$*. Then* $\mathrm{MCD}_{\widetilde{W}}{}^{(k)}$ *is locally Lipschitz and regular, and the generalized gradient takes the* *form*

$$\partial \mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i) = \begin{cases} \partial \mathrm{MCD}_i{}^{(k)}(S_i) & \textit{if}\, \mathrm{MCD}_i{}^{(k)}(S_i) > \mathrm{CDE}_{\max}{}^{(k)}(S_i), \\ \partial \mathrm{CDE}_{\max}{}^{(k)}(S_i) & \textit{if}\, \mathrm{MCD}_i{}^{(k)}(S_i) < \mathrm{CDE}_{\max}{}^{(k)}(S_i), \\ \mathrm{co}\left\{\partial \mathrm{MCD}_i{}^{(k)}(S_i),\right. & \\ \quad \left.\partial \mathrm{CDE}_{\max}{}^{(k)}(S_i)\right\} & \textit{if}\, \mathrm{MCD}_i{}^{(k)}(S_i) = \mathrm{CDE}_{\max}{}^{(k)}(S_i), \end{cases}$$

where $\mathrm{MCD}_i{}^{(k)}(S_i)$ *denotes the map* $S_i \mapsto \mathrm{MCD}_i{}^{(k)}(s_i{}^{(k)})$*, and the function* $\mathrm{CDE}_{\max}{}^{(k)} : \mathcal{D}^{k_{max}} \to \mathbb{R}_{>0}$ *and its gradient are given by,*

$$\mathrm{CDE}_{\max}{}^{(k)}(S_i) = \max_{l \in K_{\mathrm{cs}}(k)} \mathrm{CDE}_i{}^{(k:l)}(S_i)$$
$$\partial \mathrm{CDE}_{\max}{}^{(k)}(S_i) = \mathrm{co}\left\{ \partial \mathrm{CDE}_i{}^{(k:k')}(S_i) \mid k' \in \operatorname*{argmax}_{l \in K_{\mathrm{cs}}(k)} \mathrm{CDE}_i{}^{(k:l)}(S_i) \right\}.$$

The constrained objective function for a single agent may be defined as

$$\mathcal{H}_{\widetilde{W}_i}(S_i) = \max_{k \in \{1,\dots,k_{\max}\}} \mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i).$$

Note that this function may be calculated entirely by R_i. The following proposition describes the smoothness of the per-agent constrained objective function.

Proposition 4.5.6 (Extended maximum correlation distance) *Let* $i \in \{1, \dots, n\}$ *and assume that the set* $\mathcal{W}_i$ *contains at least one nonempty element. The function* $\mathcal{H}_{\widetilde{W}_i}$ *is locally Lipschitz and regular and its gradient takes the form*

$$\partial \mathcal{H}_{\widetilde{W}_i}(S_i) = \mathrm{co}\left\{\partial \mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i), k \in \{1, \dots, k_{max}\} \mid \mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i) = \mathcal{H}_{\widetilde{W}_i}(S_i)\right\}. \tag{4.14}$$

Lemma 4.5.7 (Equality of $\mathcal{H}_{\widetilde{W}_i}$ and $\mathcal{H}_{\mathcal{W}_i}$ over $\Omega_{\mathbf{Rg}_i}$) *Let* $i \in \{1, \dots, n\}$ *and* $S_i \in \Omega_{\mathrm{Rg}_i}$*. Then* $\mathcal{H}_{\widetilde{W}_i}(S_i) = \mathcal{H}_{\mathcal{W}_i}(S_i)$*.*

We next characterize the critical points of $\mathcal{H}_{\widetilde{W}_i}$ in terms of a special case of centered sequences.

Lemma 4.5.8 (Maximal elements define sub-sequences within centered sequences) *Let $K_C \subseteq \{1, \dots, k_{max}\}$ define a centered sequence of samples in S_i with $\max\limits_{k \in K_C} \mathrm{MCD}_i{}^{(k)}(s_i{}^{(k)}) = \mathcal{H}_{\mathcal{W}_i}(S_i)$. Then there is a sub-sequence, $K_{MC} \subseteq K_C$ which is centered and such that every $k \in K_{MC}$ satisfies $\mathrm{MCD}_{\widetilde{W}}{}^{(k)}(s_i{}^{(k)}) = \mathcal{H}_{\mathcal{W}_i}(S_i)$. We refer to a sequence such as K_{MC} as* maximally centered.

Proposition 4.5.9 (Global minimizers of $\mathcal{H}_{\widetilde{W}_i}$ on $\Omega_{\mathbf{Rg}_i}$ contain maximally centered sequences) *A trajectory $S_i \in \Omega_{\mathrm{Rg}_i}$ is a critical point of $\mathcal{H}_{\widetilde{W}_i}$ if and only if it contains at least one maximally centered sequence of samples. Furthermore, any such critical point globally minimizes $\mathcal{H}_{\mathcal{W}_i}$ on Ω_{Rg_i}.*

4.5.3 Multiple agent constrained problem

Finally, we combine agent trajectories into a network trajectory to find the constrained optimizers of $\mathcal{H}_{\mathcal{W}}$. First, define $\mathcal{H}_{\widetilde{\mathcal{W}}} : (\mathcal{D}^{k_{\max}})^n \to \mathbb{R}$ by

$$\mathcal{H}_{\widetilde{\mathcal{W}}}(S) = \max_{i \in \{1,\dots,n\}} \mathcal{H}_{\widetilde{W}_i}(S_i). \tag{4.15}$$

The following result extends Lemma 4.5.7 to the network.

Lemma 4.5.10 (Equality of $\mathcal{H}_{\widetilde{\mathcal{W}}}$ and $\mathcal{H}_{\mathcal{W}}$ over $\Omega_{\mathbf{Rg}}$) *Let $S \in \Omega_{\mathrm{Rg}}$. Then $\mathcal{H}_{\widetilde{\mathcal{W}}}(S) = \mathcal{H}_{\mathcal{W}}(S)$.*

The critical points of the extended network objective function may now be characterized. The proof of this result follows from Proposition 4.5.9.

Proposition 4.5.11 (Global minima of $\mathcal{H}_{\widetilde{\mathcal{W}}}$ on $\Omega_{\mathbf{Rg}}$ contain maximally centered sequences) *A trajectory $S \in \Omega_{\mathrm{Rg}}$ is a critical point of $\mathcal{H}_{\widetilde{\mathcal{W}}}$ if and only if there is at least one $i \in \mathrm{argmax}_{i \in \{1,\dots,n\}} \mathcal{H}_{\mathcal{W}_i}(S_i)$ such that S_i contains at least one maximally centered sequence. Furthermore, any such critical point is a global minimum of $\mathcal{H}_{\mathcal{W}}$ over Ω_{Rg}*

Proposition 4.5.11 allows us to think of the optimization of $\mathcal{H}_{\mathcal{W}}$ independently for each agent. If each agent optimizes their own trajectory (cf. Proposition 4.5.9), then the resulting network trajectory is optimal. Along with Proposition 4.3.2, this allows the following result on the optimal trajectories of the correlation disk-covering function $\mathcal{H}$ over Ω_{Rg}.

Proposition 4.5.12 (Range-constrained generalized multicircumcenter trajectory) *Let $S = (S_1^T, \dots, S_n^T) \in (\mathcal{D}^{k_{max}})^n$ such that each S_i contains at least one maximally centered sequence with respect to the partition $\mathcal{W} = \mathcal{MC}(S)$. Then S is a local minimizer of $\mathcal{H}$ over Ω_{Rg}. We call such a network trajectory a* range-constrained generalized multicircumcenter trajectory*. Furthermore, if* $\mathrm{I}(\mathcal{MC}(S)) = n * k_{max}$*, then S is a global minimizer of $\mathcal{H}$ over Ω_{Rg}.*

Remark 4.5.13 (S centered implies it is multicircumcenter) Note that if each S_i is centered, then it must contain a maximally centered sequence, and thus S is a range-constrained generalized multicircumcenter trajectory. •

The following proposition allows for partial optimization of trajectories which are already under way, based on minimizing the maximum error *over the remainder of the experiment*. The proof is a direct result of Proposition 4.5.9, where the samples being optimized over are anchored by the last sample already taken.

Proposition 4.5.14 (Partially fixed range-constrained generalized multicircumcenter trajectory) *Let $k^* \in \{2, \dots, k_{max}\}$, and assume that samples $\{1, \dots, k^* - 1\}$ have been taken (thus the locations are now fixed). Let $S = (S_1^T, \dots, S_n^T) \in (\mathcal{D}^{k_{max}})^n$ such that, for each $i \in \{1, \dots, n\}$, $\exists\, K_i \subseteq \{k^*, \dots, k_{max}\}$ which defines a maximal sequence of samples in S_i, with anchor point $p_i(k^* - 1)$. Then S is a local minimizer of the map $({s_1}^{(k^*)}, \dots, {s_n}^{(k_{max})}) \mapsto \mathcal{H}(S)$ over ${\Omega_{\mathrm{Rg}}}^{(\geq k^*)}$. Furthermore, if* $\mathrm{I}(\mathcal{MC}(S)) = n * k_{max}$*, then S is a global minimum of the constrained problem.*

4.6 The Generalized Multicircumcenter Algorithm

Given our discussion in the previous sections, here we synthesize coordination algorithms to find the optimal trajectories of the correlation disk-covering $\mathcal{H}$ with and without range-constraints. The design of these strategies is based on the characterizations stated in Propositions 4.4.7 and 4.5.12 for the unconstrained and the constrained cases, respectively.

Table 4.1 presents the Generalized Multicircumcenter Algorithm, a modification of the well-known Lloyd algorithm for data clustering, by which the network may find a minimizer of $\mathcal{H}$ over $\Omega_{\mathrm{Rg}}{}^{(\geq k^*)}$ for some $k^* \in \{1, \dots, k_{\max}\}$. With slight adjustments, the same algorithm works for the unconstrained case.

Goal: Find a minimum of $\mathcal{H}$ over $\Omega_{\mathrm{Rg}}{}^{(\geq k^*)}$

Input: (i) Sample interval $[k^*, k_{\max}]$
(ii) Anchor points, $p_i(k^* - 1)$, $i \in \{1, \dots, n\}$
(ii) Initial trajectory, $S^{\{0\}} = (S_1^{\{0\}}, \dots, S_n^{\{0\}})^T \in \Omega_{\mathrm{Rg}}{}^{(\geq k^*)}$, with $S_i^{\{0\}}$ the ith *agent* trajectory

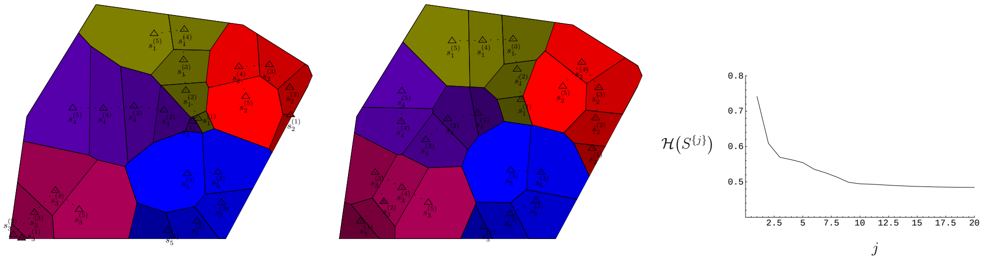
Assume: (i) R_i has a communication radius, $R \in \mathbb{R}_{>0}$ which is large enough to communicate its trajectory to any other agents whose samples are neighbors in $\mathcal{MC}$
(ii) If $k^* > 1$, R_i knows the locations of all past samples which neighbor any future samples of R_i in $\mathcal{MC}$

```
For $j \in \mathbb{Z}_{>0}$, each robot $\mathrm{R}_i$, $i \in \{1, \dots, n\}$ executes synchronously
  send all future elements of $S_i^{\{j-1\}}$ to robots within a distance of $R$
  calculate $\mathrm{MC}_i{}^{(k)}(S^{\{j-1\}})$ for $k \in \{k^*, \dots, k_{\max}\}$
  run gradient descent of $\mathcal{H}_{\widetilde{W}_i}$ on future samples only to find a centered agent trajectory,
    $S_i^{\{j\}} \in \Omega_{\mathrm{Rg}_i}{}^{(\geq k^*)}$
```

Table 4.1: Generalized Multicircumcenter Algorithm.

Figure 4.5 shows results of a simulation of the Generalized Multicircumcenter Algorithm, leaving out the initial anchor points to illustrate optimization over the set of all initial positions. The convergence properties of the algorithm are characterized in the following result.

Proposition 4.6.1 (Convergence of the Generalized Multicircumcenter Algorithm) *The* Generalized Multicircumcenter Algorithm *is distributed over the partition $\mathcal{MC}(S^{\{j\}})$, meaning that at step $j+1$,* R_i *need only communicate with* $\mathrm{R}_{i'}$ *for each $i' \in \{1,\dots,n\}$ such that ${MC_i}^{(k)}(S^{\{j\}})$ adjacent to ${MC_{i'}}^{(k')}(S^{\{j\}})$ for some k, k'. Furthermore, $S^{\{j\}} \in {\Omega_{\mathrm{Rg}}}^{(\geq k^*)}$, for all $j \in \mathbb{Z}_{>0}$. As $j \to \infty$, $S^{\{j\}}$ approaches a $S^* \in (\mathcal{D}^{k_{max}})^n$, and if $S^* \notin S_{unique}$, then S^* is a minimizer of $\mathcal{H}$ over ${\Omega_{\mathrm{Rg}}}^{(\geq k^*)}$.*



(a) (b) (c)

Figure 4.5: Simulation of 20 iterations of the Generalized Multicircumcenter Algorithm with no initial anchor points. (a) Shows the initial trajectory $S^{\{0\}}$. (b) Shows the final trajectory $S^{\{20\}}$. In each case, the associated maximal correlation partition is drawn, with the different colors representing different agents and different intensities of each color representing the timestep at which the given sample is to be taken (more intense colors represent later timesteps). The dashed lines show the path each agent will take. (c) Shows the value of $\mathcal{H}(S^{\{j\}})$ as a function of j.

Remark 4.6.2 (Limit points should be unique) We suspect that the limit points of the Generalized Multicircumcenter Algorithm are in S_{unique} except for initial conditions in a set of measure zero, but establishing this fact is challenging because of the delicate interplay between the objective function and the constraints. Extensive simulations have reinforced our idea that this intuition is correct. •

We next turn our attention to an adaptive approach to optimal path planning. Before moving to take the kth sample, an intelligent network of robotic sensors might receive updated information from an external source (a change in the environment or network composition, or even human input). One or more of the agents may switch from sensing mode to actuation mode, or back. The Generalized Multicircumcenter Algorithm directly applies to such a situation, because it optimizes over only those sample locations *not yet fixed.* The network will arrive at a trajectory which minimizes the maximum predictive variance over all trajectories feasible to the network moving forward. Table 4.2 describes the Sequential Generalized Multicircumcenter Algorithm for performing this sequential optimization. The convergence of the Sequential Generalized Multicircumcenter Algorithm follows from Proposition 4.6.1, and Figure 4.6 depicts an illustrative example.

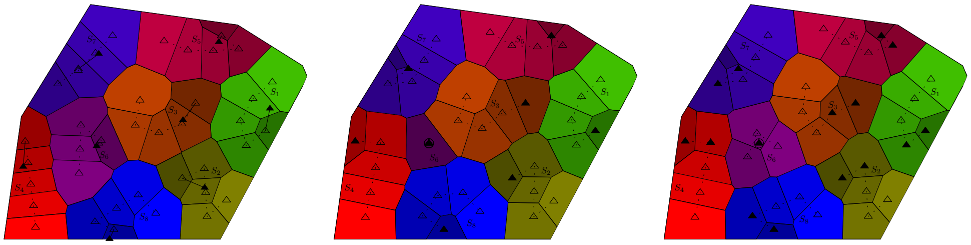
Goal: Sequentially updated optimization.

Input: (i) Initial trajectory, $S^{\{0\}} = (S_1^{\{0\}}, \dots, S_n^{\{0\}})^T \in \Omega_{\mathrm{Rg}}$, with $S_i^{\{0\}}$ the ith *agent* trajectory
(ii) Status information about correlation structure, domain boundaries, and network composition

```
Initialization
  network calculates the optimal trajectory, $S$, via the Generalized Multicircumcenter Algorithm
For $k \in \{1, \dots, k_{\max}\}$
  move to $k$th location in optimal trajectory and take $k$th sample
  if status input changed since previous optimization then
    run the Generalized Multicircumcenter Algorithm to calculate a new optimal network trajectory over ${\Omega_{\mathrm{Rg}}}^{(k+1)}$, holding the sample locations at steps $1, \dots, k$ fixed
  end if
```

Table 4.2: Sequential Generalized Multicircumcenter Algorithm

We next turn our attention to an entirely different strategy for optimal design: adaptive sampling by gradient methods.



(a) (b) (c)

Figure 4.6: Evolution of three steps of the Sequential Generalized Multicircumcenter Algorithm with $n = 8$ robots, $k_{\mathbf{max}} = 5$ steps, and Gaussian correlation. In (a), the initial trajectory is calculated from the initial anchor points $p_i(0)$. In (b), the first set of samples have been taken, and $\mathbf{R}_6$ has dropped out to perform another task (for this simulation, $\mathbf{R}_6$ remains stationary during this task). The figure shows the result of the Generalized Multicircumcenter Algorithm as run by the remaining 7 agents over timesteps $\{2, \dots, k_{\mathbf{max}}\}$. In (c), after the second set of samples have been taken, $\mathbf{R}_6$ joins the network again. The figure shows the result of optimizing over steps $\{3, \dots, k_{\mathbf{max}}\}$ with all agents. In all three plots, the anchor points and any past samples are shown as solid triangles, with solid lines connecting the initial anchors to the first samples, the optimized samples at steps $\{k^*, \dots, k_{\mathbf{max}}\}$ are empty triangles, with dashed lines connecting each agent trajectory. The last sample location of the dropped agent is circled. In each case, the associated maximal correlation partition is drawn, with the different colors representing different agents and different intensities of each color representing the timestep at which the given sample is to be taken (more intense colors represent later timesteps).

Part II

A hybrid network for gradient based adaptive sampling

Here, we relax some of the assumptions made in Part I, and consider instead a gradient based adaptive design approach. Adaptive design is the sequential process of choosing sampling locations which maximize the *gain* in information at each step. Using adaptive design and seeking relative optima via gradient descent allow us to consider a larger class of random field, in which uncertainty propagates through unknown parameters on the mean and covariance as well as the through the joint distribution of samples and predictions.

Of the existing work on distributed sensing tasks, those which consider random field models do so under an assumption of known covariance. To our knowledge this is the first work in the cooperative control arena which allows for uncertainty in the covariance of the spatiotemporal structure as well as the mean. We make use of a model derived in [43], which is the only spatial model we are aware of that makes a direct analytical connection between uncertainty in the covariance and the resulting predictive uncertainty. The key here being analytical. Aside from this model or derivatives, the common practice when confronted with unknown covariance is to either run a separate estimation procedure and then treat the covariance as known, or to use simulation methods such as Markov Chain Monte Carlo to estimate the posterior distribution. The work [80] addresses a method of choosing sample locations from a discrete space which are robust to misspecification of the covariance. Another method for handling unknown covariance has recently grown out of the exploration-exploitation approach of reinforcement learning (see, e.g. [75]). The work [45] applies this approach to the spatial estimation scenario by breaking up the objective into an exploration component which focuses on learning about the model in a discretized space and an exploitation component in which that knowledge is put to use in optimizing for prediction. Exploration is handled by discretizing the unknown parameters and developing a mixture model in which the parameters are known. Here, we provide a result in which no discretization is necessary and we take full advantage of the mobile capabilities of networks of autonomous sensors. We begin by describing some assumptions on the network and statistical model, and preliminary results which we make use of in the sequel.

4.6.1 Model assumptions

We next turn our attention to assumptions on the statistical model. Throughout the next two chapters, we use the Kitanidis model as our base, although extension to kriging is possible (see Remark 4.6.3). We assume a finite correlation range in space, $r \in \mathbb{R}_{>0}$, and in time, $r_{\mathrm{t}} \in \mathbb{R}_{>0}$, such that if $\|s_i - s_j\| \geq r$ or $|t_i - t_j| \geq r_{\mathrm{t}}$, then $\mathbf{K}_{ij} = \mathbf{K}_{ji} = 0$. For gradient calculations, we also assume that the correlation map $s_i \mapsto \mathbf{K}_{ij}$ is C^2 (which implies that $s_j \mapsto \mathbf{K}_{ij}$ is also C^2). Note that we make no restrictions on the correlation function itself, beyond the finite range and continuous spatial differentiability.

Remark 4.6.3 (Extension of subsequent results to Kriging) The simple and universal kriging results are simplified versions of our overall model, and results from the rest of this paper may be applied to those models with minimal modifications. An exception is that when approximating $\mathrm{Var}_{\mathrm{UK}}$ using subsets of measurements, care must be taken to ensure well-posedness. Specifically, an assumption that $n > p$ is required to ensure that the matrix $\mathbf{E}$ is nonsingular. •

4.6.2 Network assumptions

In order to provide a stable communication structure, to allow for the complex interaction of sample information over the entire spatial domain, and to reduce the computational burden on the mobile units, we introduce here the hybrid network comprised of mobile sensors and static nodes. Assume that each node has a limited communication radius, $R \in \mathbb{R}_{>0}$, and that they are positioned so that each one can communicate with its Voronoi neighbors. We will also require that the N_i can communicate with any robotic agents within a certain range, $R_{\mathrm{N:R}} \in \mathbb{R}_{>0}$, of the Voronoi cell, $V_i(Q)$. In order to ensure this, we assume that the communication range is,

$$R \geq \max_{i \in \{1,\ldots,m\}} \{\mathrm{CR}(V_i(Q))\} + R_{\mathrm{N:R}}. \tag{4.16}$$

The actual radius $R_{\mathrm{N:R}}$ will differ between Chapters 5 and 6. Figure 4.7 illustrates the communication requirements of the hybrid network.

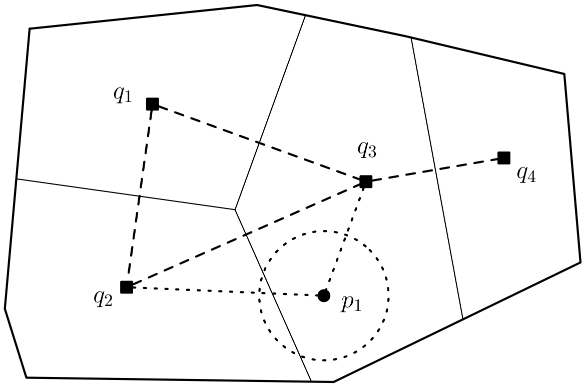


Figure 4.7: Illustration of the communication requirements of the hybrid network. The static nodes are depicted as filled boxes, with the Voronoi partition boundaries as solid lines. Each node can communicate with their Voronoi neighbors, and with any mobile robot within a radius of $R_{\mathrm{N:R}}$ (dotted circle) of the Voronoi cell. For example, q_2 needs to be able to communicate with p_1 in the above plot.

The robots can sense the positions of other robots within a distance of $2u_{\max}$. At discrete timesteps, each robot communicates the sample and spatial position to static nodes within communication range, along with the positions of any other sensed robots. The nodes then compute control vectors, and relay them back to robots within communication range. The implementation does not require direct communication between robots. We refer to this network model as $\mathcal{N}$, and the communication network of just the nodes as $\mathcal{Q}$.

4.6.2.1 Voronoi contraction for collision avoidance

We begin by specifying the region of allowed movement for the robotic agents. In addition to the maximum velocity and the restriction to $\mathcal{D}$, we impose a minimum distance requirement between robots. Beyond the benefit of collision avoidance, this restriction ensures that even under the assumption of zero sensor error, the posterior predictive variance is well-defined over the space of possible configurations.

Let $\omega \in \mathbb{R}_{>0}$ be a desired buffer width, assumed to be small compared to the size of $\mathcal{D}$. To ensure that the distance between two robots is never smaller than ω, we introduce a contraction of the Voronoi diagram. Consider the spatial locations $P = (p_1, \dots, p_n)$ of the n robotic agents at the kth timestep. Define ${\Omega_i}^{(k)} = (V_i(P))_{\omega/2} \cap \overline{B}(p_i, u_{\max})$, where $(V_i(P))_{\omega/2}$ denotes the $\frac{\omega}{2}$-contraction of $V_i(P)$. For each $j \neq i \in \{1, \dots, n\}$, we have $\mathrm{d}({\Omega_i}^{(k)}, {\Omega_j}^{(k)}) \geq \omega$. Between timesteps k and $k+1$, we restrict R_i to the region ${\Omega_i}^{(k)}$. Figure 4.8 shows an example in $\mathbb{R}^2$ of this set. The region of allowed movement of all robotic agents at timestep $k \in \mathbb{Z}_{\geq 0}$ is then the Cartesian product of the individual restrictions, i.e., $\Omega^{(k)} = \prod_{i=1}^{n} {\Omega_i}^{(k)} \subset (\mathbb{R}^d)^n$. Note that each ${\Omega_i}^{(k)}$ is the intersection of sets which are closed, bounded, and convex, and hence inherits this properties, which are in turn also inherited by $\Omega^{(k)}$.

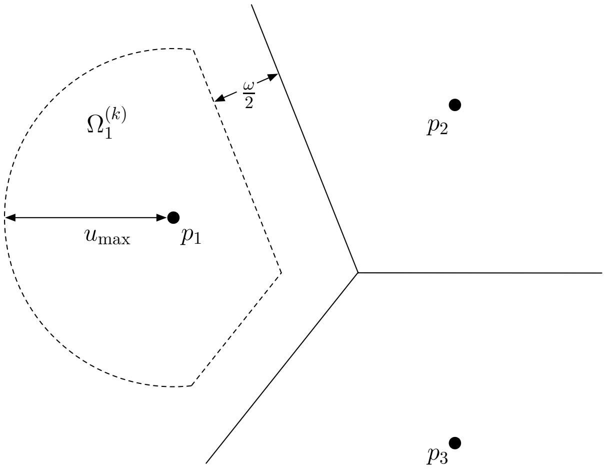


Figure 4.8: Example contraction region ${\Omega_1}^{(k)}$ (dashed) with Voronoi partition boundaries (solid) for comparison.

4.6.3 Projected gradient descent

Next, we describe the constrained optimization technique known as projected gradient descent [5] to iteratively find the minima of an objective function $F : \mathbb{R}^m \to$

$\mathbb{R}_{\geq 0}$. Let Ω denote a nonempty, closed, and convex subset of $\mathbb{R}^m$, $m \in \mathbb{Z}_{>0}$. Assume that ∇F is globally Lipschitz on Ω. Consider a sequence $\{s_k\} \in \Omega$, $k \in \mathbb{Z}_{>0}$, which satisfies

$$s_{k+1} = \mathrm{proj}_{\Omega}\left(s_k - a_k \nabla F(s_k)\right), \; s_1 \in \Omega, \tag{4.17}$$

where the step size, a_k, is chosen according to the Line Search Algorithm described in Table 4.3, evaluated at $s = s_k$.

Name:	Line Search Algorithm
Goal:	Determine step size for algorithm (4.17)
Input:	$s \in \Omega$
Assumes:	$\tau, \theta \in (0,1)$, max step $\alpha_{\max} \in \mathbb{R}_{>0}$
Output:	$\alpha \in \mathbb{R}_{\geq 0}$

```
1: $\alpha = \alpha_{\max}$
2: repeat
3:     $s_{\mathrm{new}} = \mathrm{proj}_{\Omega}\left(s - \alpha \nabla F(s)\right)$
4:     $\varpi = \frac{\theta}{\alpha}\|s - s_{\mathrm{new}}\|^2 + F(s_{\mathrm{new}}) - F(s)$
5:     if $\varpi > 0$ then
6:         $\alpha = \alpha\tau$
7:     end if
8: until $\varpi \leq 0$
```

Table 4.3: Line Search Algorithm.

Here the grid size τ determines the granularity of the line search. The tolerance θ may be adjusted for a more (larger θ) or less (smaller θ) strict gradient descent. With $\theta > 0$, the Line Search Algorithm must terminate in finite time. The Armijo condition (step 8) ensures that the decrease in F is commensurate with the magnitude of its gradient. A sequence $\{s_k\}_{k=1}^{\infty}$ obtained according to Equation (4.17) and Table 4.3 converges in the limit [5] as $k \to \infty$ to stationary points of F.

4.6.4 Distributed computational tools

Next we switch gears and discuss some useful tools for distributed algorithms. Consider a network, $\mathcal{Q}$, of m nodes with limited communication capabilities. We write $\mathcal{Q} = (Q, E)$, where $Q = (q_1, \dots, q_m)$ denotes the vector of nodes, and $E \in \mathbb{F}(\{1, \dots, m\} \times \{1, \dots, m\})$ the set of communication edges (i.e. $(i, j) \in E$ if q_i and q_j can communicate). We say that two nodes within the graph are connected if there is at least one sequence of edges, $L_{ij} = \{l_1, \dots, l_k\} \subset \{1, \dots, m\}$, $k \in \mathbb{Z}_{\geq 0}$, with $(i, l_1) \in E$, $(l_k, j) \in E$, and $(l_{k'}, l_{k'+1}) \in E$ for all $k' \in \{1, \dots, k-1\}$. We are only concerned with connected graphs (i.e. graphs in which every vertex is connected to every other vertex). We will make use of the *degree*, $\deg_{\mathcal{Q}}$, *diameter*, $\operatorname{diam}_{\mathcal{Q}}$, and *number of edges*, $\operatorname{Ed}_{\mathcal{Q}}$ of $\mathcal{Q}$ defined as,

$$\deg_{\mathcal{Q}} = \max_{i \in \{1,\dots,m\}} \deg_{\mathcal{Q}}(i) \qquad \operatorname{diam}_{\mathcal{Q}} = \max_{i,j \in \{1,\dots,m\}} |L_{\min}(i,j)| \qquad \operatorname{Ed}_{\mathcal{Q}} = |E|, \tag{4.18}$$

where we have used $L_{\min}(i, j)$ to denote a minimum length path between vertices i and j, and $\deg_{\mathcal{Q}}(i) = |\{j \in \{1, \dots, m\} \mid (i, j) \in E\}|$ the degree of node i.

Here we briefly describe some tools for distributed computations. Let $a_{ij} \in \{0, 1\}$, $i, j \in \{1, \dots, m\}$ be 1 if $(i, j) \in E$, and 0 otherwise. Let $b = (b_1, \dots, b_m)^T \in \mathbb{R}^m$, $C = [c_{ij}] \in \mathbb{R}^{m \times m}$, and assume node i knows b_i and the ith row of C. Additionally assume that $c_{ii} \neq 0$ and for $i \neq j$, $c_{ij} \neq 0$ iff i and j are communication neighbors. An example of such matrix vector pairs are the matrix $\mathbf{K}$ and vector $\mathbf{k}$ from Proposition 2.5.3, for the appropriate graph. Under these assumptions the following results hold.

JOR: The network can compute the vector $y = C^{-1} b$ via a *distributed Jacobi over-relaxation* algorithm [17, 6], formulated as the discrete-time dynamical system,

$$y_i(l+1) = (1-h) y_i(l) - \frac{h}{c_{ii}} \Big(\sum_{j \neq i} c_{ij} y_j(l) - b_i \Big), \tag{4.19}$$

for $l \in \mathbb{Z}_{\geq 0}$ and $i \in \{1, \dots, m\}$, where $y(0) \in \mathbb{R}^m$ and $h \in \Big(0, \frac{2}{\lambda_{\max}(C)}\Big)$. At the end of the algorithm, node i knows the ith element of $C^{-1} b$.

Discrete-time average consensus: The network can compute the arithmetic mean of elements of b via the discrete dynamical system [64],

$$x_i(l+1) = x_i(l) + \epsilon \sum_{j \neq i} a_{ij}(x_i(l) - x_j(l)),\ x(0) = b,$$

where $\epsilon \in (0, \frac{1}{\deg_{\mathcal{Q}}})$. At the end of the algorithm, all nodes know $\frac{\sum_{i=1}^{n} b_i}{m}$.

Maximum consensus: The network can calculate the maximum value of elements of b via a leader election algorithm [7]. Each node sends the current estimate of the maximum to all neighbors, then updates its estimate. If the process is repeated a number of times equal to the diameter of the network, then every node will know the maximum.

The first two results above are only exact asymptotically, but convergence is exponential with time.

Chapter 5

Average error minimization

In this chapter we summarize the work published in the conference papers [32] and [36], and the follow-up paper [34] (currently under revision). In it, we consider the average predictive variance as objective function, under the kriging and Kitanidis models. We develop a method of sequential optimization by projected gradient descent which may be executed in a distributed manner by the hybrid network of static nodes and robotic agents.

5.1 Problem statement

Here we outline specific assumptions on the model for the group of robotic agents and static nodes, and detail the overall objective. Since the focus of this work is the online planning of optimal sampling paths, any bounded delay incurred by network communication or calculations may be incorporated into this maximum radius of movement between sampling instants. Bounds on such delay may be inferred from the complexity analysis in Section 5.3.1. Each node will need to be able to communicate with any robot which may be within correlation range of the points in its Voronoi region at the following timestep. To that end, we assume that $R_{\mathrm{N:R}} = r + u_{\max}$ in Equation 4.16.

5.1.1 The average variance as objective function

For predictions over a region in space and time, the average variance is a natural measure of uncertainty. Using Proposition 2.5.1, we define the average over the spatiotemporal region of the posterior predictive variance,

$$\mathcal{A} = \varphi(Y, X) \int_{\mathcal{D}} \int_{T} \phi((s,t); X)\, dt\, ds. \tag{5.1}$$

Here, $Y \in (\mathbb{R}^n)^{k_{\max}}$ is a sequence of samples taken at discrete times $\{1, \ldots, k_{\max}\}$, $k_{\max} \in \mathbb{Z}_{>0}$, at space-time locations $X \in (\mathcal{D}_e^n)^{k_{\max}}$. We take $T = [1, k_{\max}]$ to be the time interval of interest, indicating that the goal of the experiment is to develop an estimate of the space-time process over the entire duration. Other time intervals may be of interest in different experiments. Their use follows with minimal changes to the methods described here.

One would like to choose the sample locations that minimize $\mathcal{A}$. Since samples are taken sequentially, with each new set restricted to a region nearby the previous, and since the sigma mean depends on the actual values of the samples, one cannot simply optimize over $(\mathcal{D}_e^n)^{k_{\max}}$ a priori.

Consider, instead, a greedy approach in which we use past samples to choose the positions for the next ones. At each timestep we choose the next locations to minimize the average posterior variance of the predictor given the data known so far. In Section 5.2, we develop a sequential formulation of the average posterior predictive variance and discuss its amenability to distributed implementation over the network $\mathcal{N}$.

5.2 Distributed criterion for adaptive design

In this section we develop an optimality criterion to maximally reduce the average predictive variance at each timestep. We begin by introducing some notation that will help us make the discussion precise.

Let $Y^{(k)} \in \mathbb{R}^n$, $k \in \{1, \ldots, k_{\max}\}$, denote the samples taken at timestep k, at space-time positions $X^{(k)} \in \mathcal{D}_e^n$. Let $Y^{(k_1:k_2)} = \left(Y^{(k_1)}, \ldots, Y^{(k_2)}\right) \in \mathbb{R}^{n(k_2-k_1+1)}$,

$k_1 < k_2$, denote the vector of samples taken over a range of timesteps, at positions $X^{(k_1:k_2)} = \big(X^{(k_1)}, \ldots, X^{(k_2)}\big) \in \mathcal{D}_e^{n(k_2-k_1+1)}$. At step k, the samples $Y^{(1:k)}$ have already been taken. We are interested in choosing spatial locations, $P \in \Omega^{(k)}$, at which to take the next samples. To that end, let $X^{(1:k+1)} : \Omega^{(k)} \to \mathcal{D}_e^{n(k+1)}$ map a new set of spatial locations to the vector of spatiotemporal locations which will result if the $(k+1)$st samples are taken there, i.e., $X^{(1:k+1)}(P) = \big(X^{(1:k)}, (P, k+1)\big)$. The adaptive design approach is then to use the samples that minimize the average prediction variance *so far*,

$$\mathcal{A}^{(k)}(P) = \varphi\big(Y^{(1:k+1)}, X^{(1:k+1)}(P)\big) \int_{\mathcal{D}} \int_{T} \phi\big((s,t); X^{(1:k+1)}(P)\big)\, dt\, ds. \tag{5.2}$$

This sequential formulation of the problem allows us to use past measurements without worrying about the ones at steps after $k+1$. However, efficient distributed implementation still suffers from three major obstacles. First, the spatially distributed nature of the problem implies that not all sample locations are accessible to any given agent at any given time. Second, inversion of the $n(k+1) \times n(k+1)$ correlation matrix, which grows with k^2, quickly becomes an unreasonable burden. Finally, the sigma mean also depends on the actual values of the samples at step $k+1$, which are not known until the measurements are taken. We handle these problems through a series of approximations, first to the sigma-conditional variance in Section 5.2.1 then to the sigma mean in Section 5.2.2, resulting in an approximation of $\mathcal{A}^{(k)}(P)$ which is both distributed in nature and computationally efficient.

5.2.1 Upper bound on sigma-conditional variance

We seek an efficient approximation of the sigma-conditional variance term $\phi\big((s,t); X^{(1:k+1)}(P)\big)$ in (5.2). As noted in Remark 2.5.2, ϕ represents the direct effect of the sample locations on the predictive uncertainty (i.e., conditional on σ^2). The network of static nodes provides a convenient method for calculating the spatial average. The average over the entire region may simply be written as the sum of averages over each cell in the Voronoi partition generated by the static nodes. As those samples which are spatially near a given cell have the most influence on reducing the variance of predictions there, we consider using *local information only* in these regional calculations. Likewise, the interaction between current samples and those far in the past is minimal, and we restrict attention to recent timesteps to avoid the problem of growing complexity. The following proposition gives an approximation of the integrated sigma-conditional variance which may be calculated by $\mathcal{Q}$ *based on local information only.*

Proposition 5.2.1 (Approximate integrated sigma-conditional variance) *Let* ${X_{\mathrm{Cor}:j}}^{(k+1)}(P)$ *denote an ordering of the set of past or current space-time locations correlated in space to* $V_j(Q)$ *and in time to* $k+1$ *such that*

$$i_{\mathbb{F}}\left({X_{\mathrm{Cor}:j}}^{(k+1)}(P)\right) = \left\{(s,t) \in i_{\mathbb{F}}\left(X^{(1:k+1)}(P)\right) \mid \mathrm{d}(s, V_j(Q)) < r \textit{ and } k+1-t < r_t\right\}.$$

Let ${\phi_j}^{(k)} : \mathcal{D}_e \times \Omega^{(k)} \to \mathbb{R}$ *map a prediction location* $x \in \mathcal{D}_e$ *and a vector of potential spatial locations to sample* $P \in \Omega^{(k)}$ *to the sigma-conditional variance of a prediction made at* x *using only the samples at space-time locations* ${X_{\mathrm{Cor}:j}}^{(k+1)}(P)$*. Then the following holds,*

$$\int_{\mathcal{D}} \int_{T} \phi((s,t); X^{(1:k+1)}(P))\, dt\, ds \leq \sum_{j=1}^{m} \int_{V_j(Q)} \int_{T} {\phi_j}^{(k)}((s,t); P)\, dt\, ds.$$

Proof: Note that although ${X_{\mathrm{Cor}:j}}^{(k+1)}(P)$ is not unique, the invariance of the sigma-conditional variance to permutations of the sample locations ensures uniqueness of $\phi_j\big(x; P\big)$. The result follows from Proposition B.0.2. ■

Next we examine the other part of $\mathcal{A}^{(k)}$, the sigma mean, and develop an efficient approximation which may be calculated by the network.

5.2.2 Approximate sigma mean

In this section, we describe our approach to deal with the term φ in (5.2). Note that the effect of the sigma mean on prediction is indirect, and its value has the same influence on predictions regardless of the predictive location. As such, we do not use spatially local approximations as we did for the sigma-conditional variance. However, to avoid the problem of complexity growth, we use samples from only a subset of the timesteps. We discuss this next. Subsequently, we address the issue of unrealized sample values by using a generalized least squares estimate.

5.2.2.1 Incorporating new data.

Here we consider minimizing the value of φ as calculated with samples from a subset of timesteps. Let ${I_\varphi}^{(k)} \subset \{1, \dots, k_{\max}\}$ denote an index set of sample steps used in the approximation at step k. Since φ is invariant under permutations of the sample vector, the specific ordering is irrelevant. There are various reasons for using different sample subsets, ${I_\varphi}^{(k)}$, depending on the field under study, the objectives of the experiment, and the desired accuracy of optimization. We present here three specific subsets which trade off accuracy for computational burden, followed by a general formulation which allows for any one of the three. Proposition 5.2.2 serves as the basis for choosing the samples to include in an approximation of the sigma mean. The proof follows from Equation (B.1c) in Lemma B.0.3 in Appendix B.

Proposition 5.2.2 (Approximate sigma mean) *Let $Y_1 \in \mathbb{R}^{n_1}$ and $Y_2 \in \mathbb{R}^{n_2}$ denote two sample vectors of lengths $n_1, n_2 \in \mathbb{Z}_{>0}$, and let $Y = (Y_1, Y_2)$. Let $\varphi_2 = \mathrm{E}[\sigma^2 | Y_2]$ denote the value of the sigma mean conditional on only the samples in Y_2, and $\varphi = \mathrm{E}[\sigma^2 | Y]$ the value conditional on the whole sample vector, Y. Then,*

$$\varphi = \varphi_2 \Big(\frac{\nu + n_2 - 2}{\nu + n_1 + n_2 - 2} + \frac{(Y_1 - \mathrm{E}[Y_1|Y_2])^T \, \mathrm{Var}[Y_1|Y_2]^{-1} (Y_1 - \mathrm{E}[Y_1|Y_2])}{\nu + n_1 + n_2 - 2} \Big),$$

where $\mathrm{E}[Y_1|Y_2]$ and $\mathrm{Var}[Y_1|Y_2]$ denote the conditional expectation and variance, respectively, of Y_1 given Y_2 (see Lemma B.0.1 in Appendix B).

This result contains some important implications with respect to the optimization problem. First, if we use the full value of $\varphi(Y^{(1:k)}, X^{(1:k)}(P)) = \mathrm{E}[\sigma^2 | Y^{(1:k)}]$ in our optimality metric at each timestep, and all steps are optimized with respect to this measure, the new information gained at later steps diminishes significantly, while the amount of effort required to glean that information increases. Second, the additional information added by including Y_1 is directly related to how well Y_1 may be estimated from Y_2.

These observations lead us to suggest three possible strategies for choosing samples to use in the approximation of the sigma mean. The diminishing returns suggest using an exploration-exploitation approach [75]. Here a block of $t_{\mathrm{blk}} \in \{1, \ldots, k_{\max}\}$ sample steps at the beginning of the experiment designates an exploration phase, during which the sigma mean is taken into account in the optimization. Subsequent iterations constitute an exploitation phase in which the sigma mean is treated as a fixed constant and we optimize only the sigma-conditional variance. The sigma mean at step k is approximated using the sample steps,

$$I_{\mathrm{explore}}{}^{(k)} = \{1, \ldots, k\} \cap \{1, \ldots, t_{\mathrm{blk}}\}. \tag{5.3}$$

This is a very efficient method, but suggests the question of how big to make t_{blk} (especially tricky in online optimization), and may place undue weight on the initial phase of the experiment.

An alternative method is to always use the most recent samples block of samples. We call this the *recent block* method. Here we approximate the sigma mean at step k from sample steps,

$$I_{\mathrm{recent}}{}^{(k)} = \{\max\{1, k - t_{\mathrm{blk}} + 1\}, k\}, \tag{5.4}$$

i.e., those samples correlated in time to timestep k. This increases the computational burden over the exploration-exploitation approach, but ensures that each step takes the unknown covariance into account in optimization. Since the maximum size of correlation matrix required is $nt_{\mathrm{blk}} \times nt_{\mathrm{blk}}$, the complexity of the problem does not grow unbounded with k. Here the choice of block size is somewhat arbitrary.

A third choice is to use select blocks of t_{blk} timesteps for scheduled updates of the sigma mean. Let $t_{\mathrm{skip}} \in \mathbb{Z}_{>0}$ be a fixed number of timesteps to skip between updates. We call this the *block update* method. The advantage of this method in reducing computational complexity comes from an assumption that $t_{\mathrm{skip}} > \lceil r_{\mathrm{t}} \rceil - 1$ so that subsequent blocks are not correlated. To simplify notation, let $t_{\text{cycle}} = t_{\text{blk}} + t_{\text{skip}}$. The sigma mean is approximated with samples from steps,

$$I_{\text{update}}{}^{(k)} = \{1, \ldots, k\} \cap \{1, \ldots, t_{\text{blk}}, t_{\text{cycle}} + 1, \ldots, t_{\text{cycle}} + t_{\text{blk}}, \ldots\}. \tag{5.5}$$

Note that using the block update approximation means alternating between the exploration and exploitation phases described above. We will show that during the exploration phases the complexity of this approach is only a constant number of operations greater than that of the recent block method, while it requires no calculation during the exploitation phases. The computational burden of this approach is thus somewhere between the other two depending on the frequency of updates.

All three methods listed above involve one or more blocks of samples of t_{blk} timesteps. Since data sampled at different time lags provide different information about σ^2, we assume that $t_{\text{blk}} \geq \lfloor r_{\text{t}} \rfloor + 1$. Here we present some notation and results which are valid for each of the methods. Table 5.1 gives an example of how the following items might look for the different methods. For reasons which will become apparent later, we will break up the sample blocks into *previous* ones which have been completed and the *current* one in progress (if there is one in progress). Let $B_{\text{p}}{}^{(k)} = \left\lfloor \frac{|I_{\varphi}{}^{(k)}|}{t_{\text{blk}}} \right\rfloor$ denote the number of previously completed sample blocks at step k. If $B_{\text{p}}{}^{(k)} = 0$, let $Y_{\text{p}}{}^{(k)} = X_{\text{p}}{}^{(k)} = \emptyset$. Otherwise, let the previous blocks be defined by

$$Y_{\text{p}}{}^{(k)} = \big(Y_1, \ldots, Y_{B_{\text{p}}{}^{(k)}}\big), \qquad X_{\text{p}}{}^{(k)} = \big(X_1, \ldots, X_{B_{\text{p}}{}^{(k)}}\big),$$

where the block vectors Y_i and X_i each correspond to a block of t_{blk} samples. Note that in the exploration-exploitation approach, we have only Y_1 and X_1, while the recent sample method will always yield $B_{\text{p}}{}^{(k)} = 0$. Let $n_{\text{c}}{}^{(k)} = n * |I_{\varphi}{}^{(k)} \cap I_{\text{recent}}{}^{(k)}|$ denote the number of samples taken so far in the current block. If $n_{\text{c}}{}^{(k)} = 0$, let $Y_{\text{c}}{}^{(k)} = X_{\text{c}}{}^{(k)} = \emptyset$, otherwise let the current block be defined by,

$$Y_{\text{c}}{}^{(k)} = Y^{(k - \frac{n_{\text{c}}{}^{(k)}}{n} + 1:k)}, \qquad X_{\text{c}}{}^{(k)} = X^{(k - \frac{n_{\text{c}}{}^{(k)}}{n} + 1:k)}.$$

$k =$	1	2	3	4	5	6	7	8	9
exploration-exploitation method									
${Y_\mathrm{c}}^{(k)} =$	$Y^{(1)}$	$Y^{(1:2)}$	$Y^{(1:3)}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
${Y_\mathrm{p}}^{(k)} =$	$\emptyset$	$\emptyset$	$\emptyset$	Y_1	Y_1	Y_1	Y_1	Y_1	Y_1
recent sample method									
${Y_\mathrm{c}}^{(k)} =$	$Y^{(1)}$	$Y^{(1:2)}$	$Y^{(1:3)}$	$Y^{(2:4)}$	$Y^{(3:5)}$	$Y^{(4:5)}$	$Y^{(5:7)}$	$Y^{(6:8)}$	$Y^{(7:9)}$
${Y_\mathrm{p}}^{(k)} =$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
block update method									
${Y_\mathrm{c}}^{(k)} =$	$Y^{(1)}$	$Y^{(1:2)}$	$Y^{(1:3)}$	$\emptyset$	$\emptyset$	$Y^{(6)}$	$Y^{(6:7)}$	$Y^{(6:8)}$	$\emptyset$
${Y_\mathrm{p}}^{(k)} =$	$\emptyset$	$\emptyset$	$\emptyset$	Y_1	Y_1	Y_1	Y_1	Y_1	(Y_1, Y_2)
Block	1	1	1	skip	skip	2	2	2	skip

Table 5.1: Example of elements used by each method of approximating φ with $r_\mathbf{t} = 2.5$, and $t_\mathbf{blk} = 3$. For the block update method, $t_\mathbf{skip} = 2$ (thus $t_\mathbf{cycle} = 5$). Here $Y_\mathbf{c}$ denotes the vector of samples in the *current* block, and $Y_\mathbf{p}$ denotes the vector of samples in *previous* blocks, both give the values *after* the kth samples have been incorporated. The completed block vectors are given by $Y_1 = Y^{(1:3)}$ and $Y_2 = Y^{(5:7)}$, so that, e.g., using the block update method at timestep 7 yields, ${\varphi_\mathbf{blk}}^{(7)} = \mathrm{E}[\sigma^2 | (Y_1, Y^{(6:7)})]$.

Let $\mathbf{K}_i$, $\mathbf{F}_i$, and $\mathbf{E}_i$, respectively ${\mathbf{K}_\mathrm{c}}^{(k)}$, ${\mathbf{F}_\mathrm{c}}^{(k)}$, and ${\mathbf{E}_\mathrm{c}}^{(k)}$ denote the values of the matrices $\mathbf{K}$, $\mathbf{F}$, and $\mathbf{E}$ as calculated from the space-time locations X_i, respectively ${X_\mathrm{c}}^{(k)}$. For k such that ${n_\mathrm{c}}^{(k)} = 0$, let ${\mathbf{K}_\mathrm{c}}^{(k)} = {\mathbf{F}_\mathrm{c}}^{(k)} = \emptyset$, and let ${\mathbf{E}_\mathrm{c}}^{(k)} = \mathbf{0}_{p \times p}$. The following result shows how a running estimate of the sigma mean may be calculated from ${Y_\mathrm{c}}^{(k)}$ and ${X_\mathrm{c}}^{(k)}$.

Lemma 5.2.3 (Sequential formulation of the sigma mean) *Let $\tilde{\varphi}k$ denote the posterior predictive mean of σ^2 conditional on the sample vector ${Y_\varphi}^{(k)} = \big({Y_p}^{(k)}, {Y_c}^{(k)}\big)$. We may write,*

$$\tilde{\varphi}k = \frac{1}{\nu + nt_{blk}{B_p}^{(k)} + {n_c}^{(k)} - 2}\Big[q\nu + \beta_0^T \mathbf{K}_0^{-1}\beta_0 + \sum_{i=1}^{{B_p}^{(k)}} \Upsilon_i + {\Upsilon_c}^{(k)} -$$

$$\big(\mathbf{K}_0^{-1}\beta_0 + \sum_{i=1}^{{B_p}^{(k)}} \Gamma_i + {\Gamma_c}^{(k)}\big)^T \big(\mathbf{K}_0^{-1} + \sum_{i=1}^{{B_p}^{(k)}} \mathbf{E}_i + {\mathbf{E}_c}^{(k)}\big)^{-1} \big(\mathbf{K}_0^{-1}\beta_0 + \sum_{i=1}^{{B_p}^{(k)}} \Gamma_i + {\Gamma_c}^{(k)}\big)\Big],$$

where

$$\Upsilon_i = Y_i^T \mathbf{K}_i^{-1} Y_i \qquad {\Upsilon_c}^{(k)} = ({Y_c}^{(k)})^T ({\mathbf{K}_c}^{(k)})^{-1} {Y_c}^{(k)}$$

$$\Gamma_i = \mathbf{F}_i \mathbf{K}_i^{-1} Y_i \qquad {\Gamma_c}^{(k)} = {\mathbf{F}_c}^{(k)} \big({\mathbf{K}_c}^{(k)}\big)^{-1} {Y_c}^{(k)}.$$

Here, ${\Upsilon_c}^{(k)}$, and ${\Gamma_c}^{(k)}$ are taken to be 0 if ${n_c}^{(k)} = 0$.

Proof: Using Lemma B.0.3, the posterior mean of σ^2 from data ${Y_\varphi}^{(k)}$ may be restated as in Equation (B.1b). Since $t_{\mathrm{skip}} \geq \lfloor r_{\mathrm{t}} \rfloor$, the sample blocks Y_i are uncorrelated to each other, and uncorrelated to ${Y_{\mathrm{c}}}^{(k)}$, which implies that the correlation matrix of all samples is block diagonal. The result follows. ■

Lemma 5.2.3 demonstrates how an estimate of the sigma mean at step k can be built from previous calculations. If the values of Υ_i, Γ_i, and $\mathbf{E}_i$ have been calculated for all previous update blocks and their sums stored, the network need only calculate ${\Upsilon_{\mathrm{c}}}^{(k)}$, ${\Gamma_{\mathrm{c}}}^{(k)}$, and ${\mathbf{E}_{\mathrm{c}}}^{(k)}$ from recent information and assemble the parts of $\tilde{\varphi}k$.

The three approximation methods mentioned here trade off computational complexity for accuracy. Using any one of these three methods, we avoid computational complexities which grow out of proportion to the information gain, however we still have the problem that the sigma mean includes sample values which have not been taken yet. We address this in the next section.

5.2.2.2 Approximating unrealized sample values.

While seeking to optimize the $(k+1)$st set of measurements, we would like to incorporate the effect of their *locations* on the posterior variance, but the actual values have not yet been sampled. Our approach is to use a generalized least squares approximation of $Y^{(k+1)}$ given only the samples used in $\tilde{\varphi}k$. We describe this in detail in the following result. The proof is in Appendix B.

Proposition 5.2.4 (Generalized least squares estimate of sigma mean) *Let ${\hat{Y}_{LS}}^{(k)} : \Omega^{(k)} \to \mathbb{R}^n$ map a vector of spatial locations $P \in \Omega^{(k)}$ to the generalized least squares estimate, based on the sample vector ${Y_\varphi}^{(k)}$, of a vector of samples to be taken* *at space-time positions $(P, k+1)$. Now, let $\hat{\varphi}^{(k+1)} : \Omega^{(k)} \to \mathbb{R}$ be defined by*

$$\hat{\varphi}^{(k+1)}(P) = \tilde{\varphi}k \qquad \textit{if} \quad {n_c}^{(k+1)} = 0, \quad \textit{otherwise,}$$

$$\begin{aligned}\hat{\varphi}^{(k+1)}(P) = \frac{1}{\nu + nt_{blk}{B_p}^{(k+1)} + {n_c}^{(k+1)} - 2}\Big[&q\nu + \beta_0^T \mathbf{K}_0^{-1}\beta_0 + \sum_{i=1}^{{B_p}^{(k)}} \Upsilon_i + {\Upsilon_c}^{(k)} - \\ &\big(\mathbf{K}_0^{-1}\beta_0 + \sum_{i=1}^{{B_p}^{(k)}} \Gamma_i + {\Gamma_c}^{(k)}\big)^T \big(\mathbf{K}_0^{-1} + \sum_{i=1}^{{B_p}^{(k)}} \mathbf{E}_i + {\mathbf{E}_c}^{(k+1)}(P)\big)^{-1} \times \\ &\big(\mathbf{K}_0^{-1}\beta_0 + \sum_{i=1}^{{B_p}^{(k)}} \Gamma_i + {\Gamma_c}^{(k)}\big)\Big],\end{aligned}$$

where ${\mathbf{E}_c}^{(k+1)}(P)$ denotes the matrix $\mathbf{E}$ as calculated with space-time location vector ${X_c}^{(k+1)}(P) = \big({X_c}^{(k)}, (P, k+1)\big)$. After the new samples, $Y^{(k+1)}$ have been taken at locations $(P, k+1)$, let ${\overline{y}_{LS}}^{(k)} : \Omega^{(k)} \to \mathbb{R}^n$ denote the estimation error, i.e., ${\overline{y}_{LS}}^{(k)}(P) = Y^{(k+1)} - {\hat{Y}_{LS}}^{(k)}(P)$. If ${n_c}^{(k+1)} = 0$ we have $\tilde{\varphi}k + 1 = \hat{\varphi}^{(k+1)}(P)$. Otherwise we may write,

$$\begin{aligned}\tilde{\varphi}k + 1 = &\frac{\tilde{\varphi}k({\overline{y}_{LS}}^{(k)}(P) - 2\overline{\mu}_{2|1})^T \operatorname{Var}[Y^{(k+1)} | {Y_\varphi}^{(k)}]^{-1}({\overline{y}_{LS}}^{(k)}(P))}{\nu + n{B_p}^{(k+1)} + {n_c}^{(k+1)} - 2} + \\ &+ \hat{\varphi}^{(k+1)}(P), \ \textit{where} \\ \overline{\mu}_{2|1} = &(\mathbf{F}_2 - \mathbf{F}_1\mathbf{K}_1^{-1}\mathbf{K}_{12})^T(\mathbf{E}_1 + \mathbf{K}_0^{-1})^{-1}\left(\mathbf{F}_1\mathbf{K}_1^{-1}Y_1 + \mathbf{K}_0^{-1}\beta_0\right).\end{aligned}$$

In other words, $\tilde{\varphi}k + 1$ may be estimated by $\hat{\varphi}^{(k+1)}(P)$, and the estimation is exact if $Y^{(k+1)} = {\hat{Y}_{LS}}^{(k)}(P)$.

5.2.3 The aggregate average prediction variance and its smoothness properties

Building on the results from Sections 5.2.2 and 5.2.1, we define here the *aggregate average prediction variance* $\tilde{\mathcal{A}}^{(k)}$. Unlike $\mathcal{A}^{(k)}$, the function $\tilde{\mathcal{A}}^{(k)}$ may be computed efficiently in a distributed manner over the network $\mathcal{N}$. The following result is a direct consequence of Propositions 5.2.1 and 5.2.4.

Proposition 5.2.5 (Spatiotemporal approximation for distributed implementation) *Let ${\tilde{\mathcal{A}}^{(k)}}_j : \Omega^{(k)} \to \mathbb{R}$ be defined by*

$$\tilde{\mathcal{A}}^{(k)}{}_j(P) = \hat{\varphi}^{(k+1)}(P) \int_{V_j(Q)} \int_T \phi_j{}^{(k)}\left((s,t),P\right)\, dt\, ds.$$

Under the assumption that the error term from Proposition 5.2.4 satisfies,

$$\lim_{k\to\infty} \frac{\tilde{\varphi} k(\overline{y}_{LS}{}^{(k)}(P) - 2\overline{\mu}_{2|1})^T \,\mathrm{Var}[Y^{(k+1)}|Y_{\varphi}{}^{(k)}]^{-1}(\overline{y}_{LS}{}^{(k)}(P))}{\nu + nB_p{}^{(k+1)} + n_c{}^{(k+1)} - 2} = 0,$$

then $\tilde{\mathcal{A}}^{(k)}(P) = \sum_{j=1}^{m} \tilde{\mathcal{A}}^{(k)}{}_j(P)$ *satisfies* $\lim_{k\to\infty} \tilde{\mathcal{A}}^{(k)}(P) \geq \lim_{k\to\infty} \mathcal{A}^{(k)}(P)$.

Remark 5.2.6 (Diminishing error) Note that the assumption of diminishing error in the sigma mean approximation is not unjustified since each step optimizes for information gain. The denominator of the fraction continues to grow while the numerator is likely to plateau at some threshold. Under this assumption the comparison between $\tilde{\mathcal{A}}^{(k)}(P)$ and $\mathcal{A}^{(k)}(P)$ is somewhat stronger than the limiting result shown in Proposition 5.2.5. The quantity $\tilde{\mathcal{A}}^{(k)}(P)$ is comprised of the product of two terms. The approximate sigma mean is very close to the sigma mean in the limit, while the approximate sigma-conditional variance is an upper bound to the actual sigma-conditional variance for all k. •

Next, we characterize the smoothness properties of $\tilde{\mathcal{A}}^{(k)}$. Let ∇_{il} denote the partial derivative with respect to p_{il}, the lth spatial component of the spatial position of R_i. We denote by ∇_i the partial derivative with respect to p_i, i.e., $\nabla_i = (\nabla_{i1}, \dots, \nabla_{id})^T$. Thus the gradient of $\tilde{\mathcal{A}}^{(k)}$ at location P may be represented as the $n * d$-dimensional vector $\left(\nabla_1^T \tilde{\mathcal{A}}^{(k)}(P), \dots, \nabla_n^T \tilde{\mathcal{A}}^{(k)}(P)\right)^T$. Given a matrix A, we denote by $\nabla_{il}A$ the component-wise partial derivative of A. The proof of the following result amounts to a careful bookkeeping of the smoothness properties of the various ingredients involved in the expressions.

Lemma 5.2.7 (Gradient of sigma-conditional variance) *If* $f_1, \dots, f_p$ *are* C^1 *with respect to the spatial position of their arguments, then the map* $P \mapsto \phi_j{}^{(k)}(x,P)$ *is* C^1 *on* $\Omega^{(k)}$ *with partial derivative,*

$$\begin{aligned}\nabla_{il}{\phi_j}^{(k)}(x,P) = -2\mathbf{k}^T\mathbf{K}^{-1}\nabla_{il}\mathbf{k} + \mathbf{k}^T\mathbf{K}^{-1}\nabla_{il}\mathbf{K}\mathbf{K}^{-1}\mathbf{k} - (\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k})^T \times \\ \left(\mathbf{K}_0^{-1} + \mathbf{E}\right)^{-1} \nabla_{il}\mathbf{E} \left(\mathbf{K}_0^{-1} + \mathbf{E}\right)^{-1} (\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k}) + \\ 2(\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k})^T \left(\mathbf{K}_0^{-1} + \mathbf{E}\right)^{-1} \nabla_{il}(\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k}), \ \textit{with}\end{aligned}$$

$$\nabla_{il}(\mathbf{f}(x) - \mathbf{F}\mathbf{K}^{-1}\mathbf{k}) = -\nabla_{il}\mathbf{F}\mathbf{K}^{-1}\mathbf{k} - \mathbf{F}\mathbf{K}^{-1}\nabla_{il}\mathbf{k} + \mathbf{F}\mathbf{K}^{-1}\nabla_{il}\mathbf{K}\mathbf{K}^{-1}\mathbf{k},$$

$$\nabla_{il}\mathbf{E} = \nabla_{il}\mathbf{F}\mathbf{K}^{-1}\mathbf{F}^T + \mathbf{F}\mathbf{K}^{-1}\nabla_{il}\mathbf{F}^T - \mathbf{F}\mathbf{K}^{-1}\nabla_{il}\mathbf{K}\mathbf{K}^{-1}\mathbf{F},$$

where the matrices $\mathbf{K}$*,* $\mathbf{E}$*, and* $\mathbf{F}$ *and the vectors* $\mathbf{k}$*, are calculated from the space-time location subvector,* ${X_{\mathrm{Cor}:j}}^{(k+1)}(P)$*.*

If, in addition, the partial derivatives of $f_1, \ldots, f_p$ *are* C^1 *with respect to the spatial position of their arguments, then the map* $P \mapsto \nabla_i {\phi_j}^{(k)}(x,P)$ *is globally Lipschitz on* $\Omega^{(k)}$*.*

It is worth noting that the matrix $\nabla_{il}\mathbf{F}$ is nonzero only in column i. The matrix $\nabla_{il}\mathbf{K}$ is nonzero only in row and column i. Additionally, due to the finite correlation range in space and time, only those elements corresponding to correlation with other measurement locations $x = (s,t)$ which satisfy $\|p_i - s\| \leq r$ and $t > k+1-r_{\mathrm{t}}$ are nonzero.

Note that the value of $\hat{\varphi}^{(k+1)}(P)$ depends on P only through the matrix ${\mathbf{E}_{\mathrm{c}}}^{(k+1)}$, whose partial derivative is analogous to that of $\mathbf{E}$ in Lemma 5.2.7. This leads us to the following result.

Lemma 5.2.8 (Gradient of sigma mean) *If* $f_1, \ldots, f_p$ *are* C^1 *with respect to the spatial position of their arguments, then* $\hat{\varphi}^{(k+1)}$ *is* C^1 *on* $\Omega^{(k)}$ *with partial derivative,*

$$\nabla_{il}\hat{\varphi}^{(k+1)}(P) = \begin{cases} \mathbf{0} & \textit{if} \quad {n_c}^{(k+1)} = 0 \\ -\frac{\Psi(P)^T \, \nabla_{il}{\mathbf{E}_c}^{(k+1)}(P) \, \Psi(P)}{\nu + nt_{blk}{B_p}^{(k+1)} + {n_c}^{(k+1)} - 2} & \textit{otherwise,} \end{cases}$$

$$\textit{where,} \quad \Psi(P) = \big(\mathbf{K}_0^{-1} + \sum_{i=1}^{{B_p}^{(k)}} \mathbf{E}_i + {\mathbf{E}_c}^{(k+1)}(P)\big)^{-1}\big(\mathbf{K}_0^{-1}\beta_0 + \sum_{i=1}^{{B_p}^{(k)}} \Gamma_i + {\Gamma_c}^{(k)}\big).$$

Additionally, if the partial derivatives of $f_1, \ldots, f_p$ *are* C^1 *with respect to the spatial position of their arguments, the gradient* $\nabla\hat{\varphi}^{(k+1)}$ *is globally Lipschitz on* $\Omega^{(k)}$*.*

We are now ready to state the smoothness properties of $\tilde{\mathcal{A}}^{(k)}$ and provide an explicit expression for its gradient. This is a direct consequence of the lemmas above.

Proposition 5.2.9 (Gradient of approximate average variance) *If $f_1, \ldots, f_p$ are C^1 with respect to the spatial position of their arguments, then $\tilde{\mathcal{A}}^{(k)}$ is C^1 on $\Omega^{(k)}$ with partial derivative,*

$$\nabla_i \tilde{\mathcal{A}}^{(k)}(P) = \hat{\varphi}^{(k+1)}(P) \int_{V_j(Q)} \int_T \nabla_i {\phi_j}^{(k)}\left((s,t), P\right) \, dt \, ds \\ + \nabla_i \hat{\varphi}^{(k+1)}(P) \int_{V_j(Q)} \int_T {\phi_j}^{(k)}\left((s,t), P\right) \, dt \, ds.$$

Additionally, if the partial derivatives of $f_1, \ldots, f_p$ are C^1 with respect to the spatial position of their arguments, the gradient $\nabla \tilde{\mathcal{A}}^{(k)}$ is globally Lipschitz on $\Omega^{(k)}$.

5.2.4 Distributed computation of aggregate average prediction variance and its gradient

Here, we substantiate our assertion that the aggregate average prediction variance and its gradient introduced in Section 5.2.3 are distributed over the network $\mathcal{N}$. Since $\mathcal{V}(Q)$ is a partition of the physical space, we may partition all sample locations spatially by region. Thus for each $(s,t) \in i_{\mathbb{F}}(X)$, there is exactly one $j \in \{1, \ldots, m\}$ such that $s \in V_j(Q)$. In order for the network to calculate $\tilde{\mathcal{A}}^{(k)}$ and its gradient at P, it is sufficient for N_j to compute ${\tilde{\mathcal{A}}^{(k)}}_j$ and $\nabla_i {\tilde{\mathcal{A}}^{(k)}}_j$ for each robot in $V_j(Q)$. Then $\tilde{\mathcal{A}}^{(k)}$ may be calculated via discrete-time average consensus (cf. Section 4.6.4), while $\nabla_i \tilde{\mathcal{A}}^{(k)}$ may be calculated from information local to R_i. From Propositions 5.2.5 and 5.2.9, it can be seen that the calculation of ${\tilde{\mathcal{A}}^{(k)}}_j$ and $\nabla_i {\tilde{\mathcal{A}}^{(k)}}_j$ requires only local information in addition to the (global) values of $\hat{\varphi}^{(k+1)}(P)$ and $\nabla_i \hat{\varphi}^{(k+1)}(P)$. Let us explain how these two quantities can be calculated.

In this section we are concerned with elements of the vectors and matrices associated with the *current* update block of $\hat{\varphi}^{(k+1)}(P)$. For $i \in \{1, \ldots, {n_{\mathrm{c}}}^{(k)}\}$, let ${x_{\mathrm{c}:i}}^{(k)}$ and ${y_{\mathrm{c}:i}}^{(k)}$ denote the ith element of the vector ${X_{\mathrm{c}}}^{(k)}$ and ${Y_{\mathrm{c}}}^{(k)}$, respectively. Let ${\mathrm{I}_{\mathrm{Local}}}^{(k)} : \mathbb{Z}_{>0} \to \mathbb{F}(\mathbb{Z}_{>0})$ map the index of the node to the set of indices of samples in the current update block whose spatial position lies inside its Voronoi cell, and whose time element is correlated to time $k+1$,

$$\mathrm{I_{Local}}^{(k)}(j) = \begin{cases} \emptyset & \text{if}\, n_{\mathrm{c}}^{(k)} = 0, \\ \left\{ i \in \{1, \dots, n_{\mathrm{c}}^{(k)}\} \mid x_{\mathrm{c}:i}^{(k)} = (s,t) \text{ and } s \in V_j(Q) \right\} & \text{otherwise.} \end{cases}$$

With a slight abuse of notation, define $\mathrm{I_{Local}}^{(k+1)}(j,P)$ to be the equivalent set of indices into the full vector of space-time measurement locations, $X_{\mathrm{c}}^{(k+1)}(P)$, with the caveat that $\mathrm{I_{Local}}^{(k+1)}(j,P) = \emptyset$ if $n_{\mathrm{c}}^{(k+1)} = 0$.

In the following results we assume that some (fixed) level of accuracy is known a priori to all nodes so that an execution of the distributed JOR or average consensus algorithms have some finite termination criterion. Unless stated otherwise, the executions of these iterative algorithms may take place in serial or parallel. Our first result illustrates the parts of $\hat{\varphi}^{(k+1)}(P)$ which do not include the locations P.

Proposition 5.2.10 (Distributed calculations without P) *Assume that* N_j, *for* $j \in \{1, \dots, m\}$, *knows* $x_{c:i}^{(k)}, y_{c:i}^{(k)}$ *for each* $i \in I_{Local}^{(k)}(j)$. *After* $p+1$ *executions of the JOR algorithm and two subsequent consensus algorithms,* N_j *has access to,*

#1: element i *of* $(\mathbf{K}_c^{(k)})^{-1} Y_c^{(k)} \in \mathbb{R}$, $i \in I_{Local}^{(k)}(j)$ *via JOR;*

#2: $\mathrm{col}_i \left(\mathbf{F}_c^{(k)} (\mathbf{K}_c^{(k)})^{-1} \right) \in \mathbb{R}^p$, $i \in I_{Local}^{(k)}(j)$ *via JOR;*

#3: $\Gamma_c^{(k)} \in \mathbb{R}^p$ *via consensus;*

#4: $\Upsilon \in \mathbb{R}^p$ *via consensus.*

Proof: Under the assumptions on $\mathcal{N}$, the matrix $\mathbf{K}_{\mathrm{c}}^{(k)}$, satisfies the requirements of the distributed JOR algorithm. The results here build on this fact and the connectedness of $\mathcal{Q}$ (which allows for consensus). ■

Next, we describe calculations that the network can execute when robotic agents are at locations P.

Proposition 5.2.11 (Distributed calculations with P) *Given $P \in \Omega^{(k)}$, assume that N_j, for $j \in \{1, \dots, m\}$, knows ${x_{c:i}}^{(k)}$ for each $i \in {I_{Local}}^{(k+1)}(j, P)$ and the results of Proposition 5.2.10. Let ${\mathbf{F}_c}^{(k+1)}(P)$ denote the matrix of basis functions evaluated at locations ${X_c}^{(k+1)}$. After p executions of JOR and $\frac{p(p+1)}{2}$ executions of consensus algorithms, N_j has access to,*

#5: $\mathrm{col}_i \left({\mathbf{F}_c}^{(k+1)}(P) ({\mathbf{K}_c}^{(k+1)}(P))^{-1} \right) \in \mathbb{R}^p$, *$i \in {I_{Local}}^{(k+1)}(j, P)$ via JOR;*

#6: ${\mathbf{E}_c}^{(k+1)}(P) \in \mathbb{R}^{p \times p}$ *via consensus.*

After these computations, N_j can calculate $\nabla_{il} {\mathbf{E}_c}^{(k+1)}$ for $l \in \{1, \dots, d\}$. Under the assumption that N_j knows the quantities $\sum_{i=1}^{{B_c}^{(k)}} \mathbf{E}_i$, $\sum_{i=1}^{{B_c}^{(k)}} \Upsilon_i$, and $\sum_{i=1}^{{B_c}^{(k)}} \Gamma_i$, then N_j can calculate $\hat{\varphi}^{(k+1)}(P)$ and $\nabla_i \hat{\varphi}^{(k+1)}(P)$ for each robot in $\{i \in \{1, \dots, n\} \mid p_i \in V_j(Q)\}$.

Proof: The matrix ${\mathbf{K}_{\mathrm{c}}}^{(k+1)}(P)$ satisfies the requirements of the distributed JOR algorithm by the assumptions on $\mathcal{N}$. The itemized results follow from this, and the symmetry of the matrix ${\mathbf{E}_{\mathrm{c}}}^{(k+1)}(P)$. The calculation of $\hat{\varphi}^{(k+1)}(P)$ and its partial derivatives follow from Lemmas 5.2.3 and 5.2.8. ■

5.3 Distributed optimization of the aggregate average predictive variance

Here we present a distributed projected gradient descent algorithm which is guaranteed to converge to a stationary point of $\tilde{\mathcal{A}}^{(k)}$ on $\Omega^{(k)}$. The DISTRIBUTED AVERAGE VARIANCE PROJECTED GRADIENT DESCENT ALGORITHM in Table 5.3 allows the network of static nodes and mobile agents to find local minima of $\tilde{\mathcal{A}}^{(k)}$ on $\Omega^{(k)}$. At timestep k, the nodes follow a gradient descent algorithm, defining a sequence of configurations, $\{P_l^\dagger\}$, $l \in \mathbb{Z}_{>0}$, such that $P_1^\dagger$ is $P^{(k)} \in \Omega^{(k)}$, the vector of current spatial locations of the robotic agents and

$$P_{l+1}^\dagger = \mathrm{proj}_\Omega \left(P_l^\dagger - \alpha \nabla \tilde{\mathcal{A}}|_{P_l^\dagger} \right), \ \alpha \in \mathbb{R}_{\geq 0},$$

Name: Distributed Average Variance Line Search Algorithm

Goal: Compute step size for projected gradient descent of $\tilde{\mathcal{A}}^{(k)}$

Input: Configuration, $P = (p_1, \dots, p_n) \in \Omega^{(k)}$

Assumes: (i) Connected network of static nodes
(ii) N_j knows p_i, $\tilde{\mathcal{A}}^{(k)}{}_j(P)$, $\nabla_i \tilde{\mathcal{A}}^{(k)}(P)$ and Ω_i for each robot within communication range
(iii) $\|\nabla_i \tilde{\mathcal{A}}^{(k)}(P)\| \neq 0$ for at least one $i \in \{1, \dots, n\}$
(iv) N_j knows items #3 and #4 from Proposition 5.2.10
(v) Shrinkage factor τ and tolerance $\theta \in (0,1)$ known a priori by all static nodes

Uses: (i) Projection of next set of locations on Ω_i,
$P'_j(\alpha, P) = \Big\{ \mathrm{proj}_{\Omega_i}(p_i + \alpha \nabla_i \tilde{\mathcal{A}}(P)), \forall\, i \,\text{with}\, \mathrm{d}\,(p_i, V_j(Q)) \leq r + u_{\max} + \omega \Big\}.$

(ii) Total distance traveled by robots entering $V_j(Q)$,
$$\mathrm{d}_j\,(\alpha, P)^2 = \sum_{\substack{i \in \{1,\dots,n\} \text{ such that} \\ \mathrm{proj}_{\Omega_i}\left(p_i + \alpha \nabla_i \tilde{\mathcal{A}}(P)\right) \in V_j(Q)}} \| \mathrm{proj}_{\Omega_i} \Big(p_i + \alpha \nabla_i \tilde{\mathcal{A}}(P) \Big) - p_i \|^2.$$

Output: Step size $\tau \in \mathbb{R}$

```
Initialization
N_1, ..., N_m calculate $\alpha_{\max} = \frac{u_{\max}}{\min\{\|\nabla_i \tilde{\mathcal{A}}(P)\| \mid \|\nabla_i \tilde{\mathcal{A}}(P)\| \neq 0\}}$ via *maximum consensus*
For $j \in \{1, \dots, m\}$, node $\mathrm{N}_j$ sets $\alpha = \alpha_{\max}$, and executes concurrently

repeat
    calculates $\hat{\varphi}^{(k+1)}\big(P'_j(\alpha, P)\big)$ according to Proposition 5.2.11
    calculates $\mathrm{d}_j\,(\alpha, P)^2$ and $\tilde{\mathcal{A}}^{(k)}{}_j\,\big(P'_j(\alpha, P)\big)$
    execute consensus algorithm to calculate the following:
        $\tilde{\mathcal{A}}^{(k)}\,(P'(\alpha, P)) = \sum_{j=1}^{m} \tilde{\mathcal{A}}^{(k)}{}_j\,\big(P'_j(\alpha, P)\big)$, and $\|P - P'(\alpha, P)\|^2 = \sum_{j=1}^{m} \mathrm{d}_j\,(\alpha, P)^2$
    $\varpi = \frac{\theta}{\alpha}\,\|P - P'(\alpha, P)\|^2 + \tilde{\mathcal{A}}^{(k)}(P'(\alpha, P)) - \tilde{\mathcal{A}}^{(k)}(P)$
    if $\varpi > 0$ then
        $\alpha = \alpha\tau$
    end if
until $\varpi \leq 0$
```

Table 5.2: Distributed Average Variance Line Search Algorithm.

where α is chosen via the DISTRIBUTED AVERAGE VARIANCE LINE SEARCH ALGORITHM outlined in Table 5.2. The DISTRIBUTED AVERAGE VARIANCE LINE SEARCH ALGORITHM is a distributed version of the LINE SEARCH ALGORITHM from Table 4.3. The maximum stepsize, $\alpha_{\max} \in \mathbb{R}_{>0}$, is designed to ensure that all robots with nonzero partial derivatives can move the maximum distance.

When $|\tilde{\mathcal{A}}^{(k)}(P^\dagger_{l+1}) - \tilde{\mathcal{A}}^{(k)}(P^\dagger_l)| = 0$, the algorithm terminates, and the nodes set $P^{(k+1)} = P^\dagger_{l+1}$. By the end of this calculation, each node knows the identity of robotic agents in its Voronoi cell at timestep $k+1$. Node N_j transmits $p_i(k+1)$ to robot R_i, which then moves to the location between timesteps. In the interest of brevity, we have used the informal shorthand, $\mathrm{R}_i \in R_{\mathrm{cov}}(j)$ to signify "for each robot in $R_{\mathrm{cov}}(j)$". Similarly, we use $\mathrm{N}_j \in S_{\mathrm{cov}}(i)$ to signify "for each node in $S_{\mathrm{cov}}(i)$". Note that although each robot may be sending position and sample information to multiple nodes, the approximate average prediction variance is calculated *within the Voronoi cell.* As the Voronoi cells do not overlap, there is no problem with information repetition.

The following result describes some nice properties of the DISTRIBUTED AVERAGE VARIANCE PROJECTED GRADIENT DESCENT ALGORITHM. Its proof is a direct result of the construction of the algorithm and the fact that it is equivalent to a centralized projected gradient descent.

Proposition 5.3.1 (Properties of the Distributed Average Variance Projected Gradient Descent Algorithm) *The* DISTRIBUTED AVERAGE VARIANCE PROJECTED GRADIENT DESCENT ALGORITHM *is distributed over the network $\mathcal{N}$. Moreover, if the partial derivatives of $f_1, \dots, f_p$ are C^1 with respect to the spatial position of their arguments, any execution is such that the robots do not collide and, at each timestep after the first, measurements are taken at stationary configurations of $P \mapsto \tilde{\mathcal{A}}^{(k)}(P)$ over $\Omega^{(k)}$.*

The proposed algorithm is robust to failures in the *mobile* agents. If an agent stops sending position updates, it ceases to receive new control vectors. The rest of the network continues operating with the available resources and will eventually sample the

Name:	Distributed Average Variance Projected Gradient Descent Algorithm
Goal:	Find a local minimum of $\tilde{\mathcal{A}}^{(k)}$ within $\Omega^{(k)}$.
Assumes:	(i) Connected network of static computing nodes and mobile robotic sensing agents (ii) Static nodes deployed over $\mathcal{D}$ such that $R \geq \max_{i\in\{1,\dots,m\}}\{\mathrm{CR}(V_i(Q))\} + r + u_{\max}$, robotic agents in initial configuration $P^{(1)} \in \Omega^{(k)}$ (iii) Line search shrinkage factor τ and tolerance value $\theta \in (0,1)$ known a priori by all nodes (iv) A termination marker known to all nodes and robots which may be sent to mark the end of a gradient descent loop.

```
At step $k \in \mathbb{Z}_{\geq 0}$, each $\mathrm{N}_j$ executes:
$R_{\mathrm{cov}}(j) := \{\mathrm{R}_i \mid \mathrm{d}(p_i(k), V_j(Q)) \leq r\}$
collect initial sample and position from $\mathrm{R}_i \in R_{\mathrm{cov}}(j)$.
compute first $\tilde{\mathcal{A}}^{(k)}{}_j\big(P^{(k)}\big)$ and then $\tilde{\mathcal{A}}^{(k)}\big(P^{(k)}\big)$ via consensus
$P_{\mathrm{next}} := P^{(k)}$
repeat
    $P_{\mathrm{cur}} := P_{\mathrm{next}}(j)$, compute $-\nabla\tilde{\mathcal{A}}^{(k)}{}_j|_{P_{\mathrm{cur}}}$
    send vector $\nabla_i\tilde{\mathcal{A}}^{(k)}{}_j(P_{\mathrm{cur}})$ to $\mathrm{R}_i \in R_{\mathrm{cov}}(j)$
    collect sum $\nabla_i\tilde{\mathcal{A}}^{(k)}(P_{\mathrm{cur}})$ from $\mathrm{R}_i \in R_{\mathrm{cov}}(j)$
    get $\alpha$ via DISTRIBUTED AVERAGE VARIANCE LINE SEARCH ALGORITHM at $P_{\mathrm{cur}}$
    $P_{\mathrm{next}} := P_{\mathrm{cur}} + \alpha\nabla\tilde{\mathcal{A}}^{(k)}|_{P_{\mathrm{cur}}}$
    calculate $|\tilde{\mathcal{A}}^{(k)}(P_{\mathrm{next}}) - \tilde{\mathcal{A}}^{(k)}(P_{\mathrm{cur}})|$ from known quantities
until $|\tilde{\mathcal{A}}^{(k)}(P_{\mathrm{next}}) - \tilde{\mathcal{A}}^{(k)}(P_{\mathrm{cur}})| = 0$
$P^{(k+1)} := P_{\mathrm{next}}$, send next position to robots in $V_j(Q)$
```

```
Meanwhile, each $\mathrm{R}_i$ executes:
take measurement at $p_i(k)$
$S_{\mathrm{cov}}(i) := \{\mathrm{N}_j \mid \mathrm{d}(p_i(k), V_j(Q)) \leq r\}$
send measurement and position to all nodes in $S_{\mathrm{cov}}(i)$
repeat
    receive $\nabla_i\tilde{\mathcal{A}}^{(k)}{}_j(P^{(k)})$ from $\mathrm{N}_j \in S_{\mathrm{cov}}(i)$
    calculate sum $\nabla_i\tilde{\mathcal{A}}^{(k)}(P^{(k)})$
    send $\nabla_i\tilde{\mathcal{A}}^{(k)}(P^{(k)})$ to $\mathrm{N}_j \in S_{\mathrm{cov}}(i)$
until receive termination marker from any node
receive next location $p_i(k+1)$
move to $p_i(k+1)$.
```

Table 5.3: Distributed Average Variance Projected Gradient Descent Algorithm.

areas previously covered by the failing agents. With minor modifications, the algorithm could be made robust to a certain number of node failures as well. However, this would require larger communication radius and extra storage (essentially having each node keep track of the sample locations stored by its Voronoi neighbors).

Remark 5.3.2 (Extension to relative positioning) It is interesting to observe that, due to the fact that the actual positions of samples are only required in a local context, our algorithm can also be implemented in a robotic network with relative positioning. The only requirements are the following: that each node can calculate the mean basis function for all local samples; that each node can calculate the correlations between pairs of local samples and that neighboring nodes can agree on the ordering of those samples within the global matrix. These modifications would not impact the convergence properties of the algorithm. •

5.3.1 Complexity analysis

Here we examine in detail the complexity of the Distributed Average Variance Projected Gradient Descent Algorithm in terms of the number of robotic agents and the number of static nodes. For reference, we compare our proposed algorithm against a centralized algorithm that uses all-to-all broadcast and global information, and does not take advantage of the distributed nature of the problem. Proofs of results for this section may be found in Appendix A.3.

Given that the Distributed Average Variance Projected Gradient Descent Algorithm is sequential, and designed to run for a fixed number of timesteps, we are concerned here with complexities involved in performing a single step. Below, where we refer to complexity notions over multiple iterations of an algorithm, we are considering the nested algorithms such as JOR, or consensus, which run during a single step of the Distributed Average Variance Projected Gradient Descent Algorithm.

We examine the algorithm performance against the following notions of complexity, see [7, 53, 65],

Communication complexity: the maximum number of bits transmitted over all (directed) communication channels between nodes in the network over the course of the algorithm;

Time complexity: the maximum number of iterations to completion of the algorithm times the maximum number of bits sent over any channel during one iteration;

Space complexity: the total number of bits for which space may be required *by a single node at any given time.*

We consider the complexity of the algorithms in terms of the number of agents, n, and the number of nodes, m, independently. We use the well-known *Bachmann-Landau* notation for upper and lower bounds. Given functions $f, g : \mathbb{Z}_{>0} \times \mathbb{Z}_{>0} \to \mathbb{R}_{\geq 0}$, we say that $f \in O(g)$ (respectively $f \in \Omega(g)$) if there exist $A \in \mathbb{R}_{>0}$ (respectively $a \in \mathbb{R}_{>0}$) and $\gamma_n, \gamma_m \in \mathbb{Z}_{>0}$ such that $f(n, m) \leq Ag(n, m)$ (respectively $f(n, m) \geq ag(n, m)$) for all $n \geq \gamma_n$ and $m \geq \gamma_m$. If f and g satisfy both $f \in O(g)$ and $f \in \Omega(g)$, then we say $f \in \Theta(g)$.

Throughout this section we make the following assumption on the diameter, degree, and number of edges of the communication graph $\mathcal{Q}$ of the network of static nodes.

Regularity Assumption- We assume that the group of static nodes is regular in the sense that the following three bounds are satisfied as m increases:

$$\text{diam}_{\mathcal{Q}} \in \Theta(\sqrt[d]{m}) \qquad \text{Ed}_{\mathcal{Q}} \in \Theta(m) \qquad \text{deg}_{\mathcal{Q}} \leq \text{deg}_{\text{max}},$$

where $\text{deg}_{\text{max}} \in \mathbb{R}$ constant with respect to m.

Remark 5.3.3 (Network assumptions are reasonable) In two and three dimensions, the maximum diameter requirement has been shown to be consistent with a hexagonal grid network [8, 11], which is also consistent (in terms of number of neighbors) with the average case for large Voronoi networks [63]. The requirement of bounded degree would also be satisfied by a hexagonal grid. The total number of edges is half the sum of the number of neighbors over all nodes, so bounded degree yields $\mathrm{Ed}_{\mathcal{Q}} \propto m$. •

We are now ready to characterize the complexities of our algorithms, beginning with the inner iterations.

Proposition 5.3.4 (Average consensus complexity) *Let $b = (b_1, \dots, b_m)^T \in \mathbb{R}^m$ denote a vector distributed across $\mathcal{Q}$ in the sense that N_j knows b_j for each $j \in \{1, \dots, m\}$. The discrete time consensus algorithm to calculate $\frac{b^T b}{m}$ to an accuracy of ϵ has communication complexity in $O_\epsilon\left(m^2 \sqrt[d]{m}\right)$, time complexity in $O_\epsilon\left(m \sqrt[d]{m}\right)$, and space complexity in $O_\epsilon(1)$.*

Since $\tilde{\mathcal{A}}^{(k)}$ uses only measurements correlated in time, the size of the matrices and vectors is limited to a constant multiple of n. Recall from Section 5.2.2 the definitions of ${n_\mathrm{c}}^{(k)}$ and ${\mathbf{K}_\mathrm{c}}^{(k)}$ as the number and correlation matrix of samples in the *current* update block.

Proposition 5.3.5 (Leader election complexity) *The leader election algorithm may be run on $\mathcal{Q}$ to calculate the quantity $\displaystyle\max_{i \in \{1, \dots, {n_c}^{(k)}\}} \sum_{j=1}^{{n_c}^{(k)}} [{\mathbf{K}_c}^{(k)}]_{ij}$, with communication complexity in $O\left(m \sqrt[d]{m}\right)$, time complexity in $O\left(\sqrt[d]{m}\right)$, and space complexity in $O(1)$.*

For the algorithms considered next, the distribution of samples in the region defines two different regimes for complexity. We will consider both the worst case and the average based on a uniform distribution.

Proposition 5.3.6 (JOR complexity) *Assume that there is some constant $\varpi_\lambda \in (0, 1)$, known a priori, such that $\lambda_{min}({\mathbf{K}_c}^{(k)}) > \varpi_\lambda$. Regarding the sparsity of ${\mathbf{K}_c}^{(k)}$, assume that any one sample is correlated to at most $N_{cor} \in \mathbb{Z}_{>0}$ others, and that, for any $j \in \{1, \dots, m\}$, the number of samples in $\mathcal{D} \setminus V_j(Q)$ which are correlated to samples in $V_j(Q)$ is upper bounded by a constant, $N_{msg} \in \mathbb{Z}_{>0}$. Let $b = (b_1, \dots, b_{n_{c^{(k)}}})^T \in \mathbb{R}^{{n_c}^{(k)}}$ be distributed on the network of nodes in the sense that if N_j knows $\mathrm{col}_i({\mathbf{K}_c}^{(k)})$, then N_j*

knows b_i. Using the distributed JOR algorithm, the network may calculate $({\mathbf{K}_c}^{(k)})^{-1}b$ to accuracy ϵ with communication complexity in $O_\epsilon(m\sqrt[d]{m})$, time complexity in $O_\epsilon(\sqrt[d]{m})$, and space complexity in $O_\epsilon(n)$ worst case, $O_\epsilon(\frac{n}{m})$ average case.

Remark 5.3.7 (Interpretation of sparsity assumptions) The assumptions on the sparsity of ${\mathbf{K}_\mathrm{c}}^{(k)}$ in Proposition 5.3.6 have the following interpretation: samples do not cluster in space as measured with respect to the distribution of the Voronoi cells and their size relative to the correlation range. •

The above results allow us to characterize the complexities of the DISTRIBUTED AVERAGE VARIANCE PROJECTED GRADIENT DESCENT ALGORITHM.

Proposition 5.3.8 (Complexity of the Distributed Average Variance Projected Gradient Descent Algorithm) *Under the assumptions of Table 5.3, the* DISTRIBUTED AVERAGE VARIANCE PROJECTED GRADIENT DESCENT ALGORITHM *may be completed within tolerance ϵ with communication complexity in $O_\epsilon(m^2\sqrt[d]{m})$, time complexity in $O_\epsilon(m\sqrt[d]{m})$, and space complexity in $O_\epsilon(n^2)$ worst case, $O_\epsilon\big(\frac{n^2}{m^2}\big)$ average case.*

Proof: The space complexity is dominated by the need to store the inverse correlation matrix of known samples required for ${\tilde{\mathcal{A}}^{(k)}}_j$. Even though the correlation matrix is sparse, the inverse is in general not, requiring the whole $\frac{{n_\mathrm{c}}^{(k)}\times({n_\mathrm{c}}^{(k)}+1)}{2}$ storage space for the upper or lower triangle of the symmetric matrix. The worst case corresponds to all ${n_\mathrm{c}}^{(k)}$ samples correlated to one Voronoi region, and the average to samples distributed uniformly. The time and communication complexities are dominated by the requirement of the consensus algorithm. ■

5.3.1.1 Broadcast method for comparison

One way to judge the efficiency of our method would be to compare it against a simple algorithm which floods the network with new information at each sample time. This algorithm would work as follows. At each timestep, all samples and locations are disseminated throughout the network, such that each node obtains the entire vectors X and Y. A (centralized) projected gradient descent algorithm could then be run by N_j, $j \in \{1, \ldots, m\}$ to find the next sample locations for those agents within $V_j(Q)$. Since all nodes have the same information, any such algorithms should converge to the same final locations, so there would be no difficulty with overlapping computations. Since this method is only given for comparison, we will assume that this is the case. Once a node has calculated the next location for all of the agents which will be in that Voronoi cell, the control vectors may be transmitted to those agents. The information dissemination in this algorithm corresponds to an all-to-all broadcast in which each node begins with a distinct message of length $|\mathrm{I}_{\mathrm{Local}}{}^{(k+1)}(j, P)|$ units. There are a number of different ways this may be carried out. Here we assume the simple flooding method proposed in [76], which is optimal for time complexity. In this method, every node continues to transmit any new information to all neighbors until new information is exhausted.

Proposition 5.3.9 (Complexity of the broadcast method) *Under the regularity assumption on $\mathcal{Q}$, local minima of $\tilde{\mathcal{A}}^{(k)}$ may be found by all-to-all broadcast of agent positions and subsequent local projected gradient descent with,*

- *communication complexity in* $\Theta(nm)$
- *time complexity in* $\Theta(n + m)$
- *space complexity in* $\Theta(n^2)$

Remark 5.3.10 (Broadcast method requires global positioning) It should be noted here that while the Distributed Average Variance Projected Gradient Descent Algorithm might be extended to systems with relative positioning (see Remark 5.3.2), the broadcast method requires global coordinates. •

Table 5.4 lists the complexity bounds side by side for comparison. Note that the bounds on the broadcast method are both upper and lower bounds. It can be seen that the distributed method scales better overall with the number of mobile agents.

Complexity Type	Broadcast	Distributed PGD	
		Worst	Average
Communication	$\Theta(mn)$	$O_\epsilon(m^2\sqrt[d]{m})$	$O_\epsilon(m^2\sqrt[d]{m})$
Time	$\Theta(n+m)$	$O_\epsilon(m\sqrt[d]{m})$	$O_\epsilon(m\sqrt[d]{m})$
Space	$\Theta(n^2)$	$O_\epsilon(n^2)$	$O_\epsilon\big((\frac{n}{m})^2\big)$

Table 5.4: Algorithm complexities. The worst and average cases are over distributions of samples, with the average corresponding to a uniform distribution in $\mathcal{D}$. The bounds for the broadcast method are derived from results in [76].

The results with respect to increasing the number of static nodes are less favorable, but include a tradeoff between the average storage requirement and the communication and time complexities. There is an additional benefit to increasing the number of nodes not mentioned here which is that the average computational burden on each node decreases.

Chapter 6

Adaptive maximum entropy sampling

This chapter summarizes work published in the conference paper [35]. In it, we make use of the hybrid network and the projected gradient descent methods developed in Chapter 5 to enable adaptive sampling with the maximum entropy as optimality criterion. While the methods used are overall similar, the challenges in adapting a different criterion to the distributed setting are considerable.

6.1 Problem statement

Here we have a single assumption on the robotic network model in addition to those already stated. We will require that N_i be able to communicate with any robot which may be inside $V_j(Q)$ at the following timestep. We therefore let $R_{\mathrm{N:R}} = u_{\max}$ in Equation 4.16. In Section 6.1.1 we detail the overall network objective.

6.1.1 Network objective

Between measurement instants, we would like to move the robots to those locations which ensure a maximum gain in information appropriate to the goal of the experiment. When the goal is to make inference about model parameters, we would like an objective function which maximizes gain in information about the model. A generally accepted practice [48, 54, 10] is to choose a set of measurement locations which maximize the entropy of the joint posterior predictive distribution. Intuitively, to maximize the gain in information we choose to measure those locations about which we currently know the least.

6.2 A distributed criterion for one-step-ahead data collection

In this section, we derive an expression for the entropy of the joint posterior predictive distribution given the model (2.2). As this function is not amenable to distributed computations, we propose instead an alternative which is. We finish with important smoothness properties of our proposed objective function, including an expression for its gradient.

6.2.1 Entropy of the random field estimation

We begin with a novel reformulation of the conditional entropy which will enable an approximate objective function.

Proposition 6.2.1 (Reformulation of conditional entropy) *The conditional entropy presented in Section 2.5.3 may be reformulated as,*

$$\begin{aligned}\log\det\left(\phi(X_u;X_s)\right) = \log\det\left(\Upsilon\right) + \log\det\left(\mathbf{K}\right) - \\ - \log\det\left(\Upsilon_s\right) - \log\det\left(\mathbf{K}_s\right). \qquad (6.1)\end{aligned}$$

Here Υ, respectively Υ_s, is the inverse of the posterior covariance matrix for the parameter vector β, given the all the data, respectively the sampled data. Explicitly,

$$\Upsilon = \mathbf{K}_0^{-1} + \mathbf{F}\mathbf{K}^{-1}\mathbf{F}^T \text{ and } \Upsilon_s = \mathbf{K}_0^{-1} + \mathbf{F}_s{\mathbf{K}_s}^{-1}{\mathbf{F}_s}^T.$$

Note from (6.1) that $\log\det\left(\phi(X_u;X_s)\right)$ does not depend on the values of the measurements, *only* on their locations, and that the last two terms do not depend on the new locations at all. Thus we are interested in maximizing

$$\widetilde{\mathcal{E}} = \log\det(\Upsilon) + \log\det(\mathbf{K}), \tag{6.2}$$

over potential measurement sites. However, the full distributed computation of these terms or their gradients over the robotic sensor network is not straightforward. We show later that the term $\log\det(\Upsilon)$ can be handled using known distributed computation tools. To deal with the term $\log\det(\mathbf{K})$, we follow the route presented next.

6.2.2 Alternative criterion for adaptive design

In this section, we propose an alternative aggregate objective function to maximize the posterior predictive entropy at each timestep. Let $\mathcal{E}^{(k)} : \mathcal{D}^n \to \mathbb{R}$ be defined by

$$\mathcal{E}^{(k)}(P) = \log\det(\Upsilon) - \frac{1}{2}\mathrm{tr}\big((\mathbf{K} - \boldsymbol{I})^2 \big), \tag{6.3}$$

where the matrices $\Upsilon = \Upsilon^{(k)}(P)$ and $\mathbf{K} = \mathbf{K}^{(k)}(P)$ are calculated using the spatial positions $P \in \mathcal{D}^n$ at time $k+1$ for unsampled locations. We avoid the explicit functional notation for ease of exposition.

Proposition 6.2.2 ($\mathcal{E}^{(k)}(P)$ is a second order approximation of $\widetilde{\mathcal{E}}$) *Let $\mathcal{T}^{(k)} \subset \mathbb{R}^{n(k-1)}$ denote the following set of configurations based on the correlation matrix,*

$$\mathcal{T}^{(k)} = \left\{ P \in \mathcal{D}^n \mid \max_{i=1}^{n(k-1)} \{ \sum_{j=1}^{n(k-1)} [\mathbf{K}]_{ij} \} < 2 \right\}.$$

The function $\mathcal{E}^{(k)}$ is a second order approximation of $\widetilde{\mathcal{E}}$ over the region $\mathcal{T}^{(k)}$ in the sense that $-\frac{1}{2}tr\big((\mathbf{K} - \boldsymbol{I})^2 \big)$ is the second order Taylor approximation of $\log\det(\mathbf{K})$.

Throughout the sequel, we assume $P \in \mathcal{T}^{(k)}$. Note that the trace may be written as a sum of the form,

$$\mathrm{tr}\left((\mathbf{K} - \boldsymbol{I})^2\right) = \sum_{i=1}^{n(k+1)} \mathrm{row}_i(\mathbf{K} - \boldsymbol{I})\mathrm{col}_i(\mathbf{K} - \boldsymbol{I}). \tag{6.4}$$

Under the spatial model described in Section 2.5, the ith term in the sum is a function only of the locations of measurements within a spatial distance of r from $x_{u:i}$. We call those measurements *correlation neighbors* of R_i.

Remark 6.2.3 (Approximating $\log\det(\mathbf{K})$ to higher order at the cost of higher communication complexity) If a higher order approximation is desired, it can be done by passing information for each extra term. For example, the third order approximation requires the additional term,

$$\frac{1}{3}\mathrm{tr}\left((\mathbf{K}-\boldsymbol{I})^3\right) = \frac{1}{3}\sum_{i=1}^{n(k+1)} \mathrm{row}_i\left((\mathbf{K}-\boldsymbol{I})^2\right)\mathrm{col}_i(\mathbf{K}-\boldsymbol{I}).$$

Here, the vector $\mathrm{row}_i\left((\mathbf{K}-\boldsymbol{I})^2\right)$ can be calculated from $\mathrm{row}_j(\mathbf{K}-\boldsymbol{I})$ where j ranges over i and its correlation neighbors. Thus the third order term in the Taylor series may be calculated with information from the two-hop correlation neighbors. Similarly, the fourth order term may be calculated with three-hop information and so on. •

Note that it is possible to use the requirement of a minimum distance between agents to strictly enforce the assumption $P \in \mathcal{T}^{(k)}$. We do not provide a formal method here, but a simple rule of thumb may be applied. Under the restriction of a minimum distance between agents, the most dense configuration is known to be a hexagonal grid. In two dimensions, for example, this implies that there are a maximum of six robots at a distance of ω of any one robot, then a maximum of six others at a distance of $\frac{3}{2}\omega$, etc. By counting the possible number of samples at each interval, including those from previous and subsequent correlated timesteps, it is possible to obtain an upper bound on the maximum row sum as a function of ω.

6.2.3 Smoothness properties of $\mathcal{E}^{(k)}$

Here we study the smoothness properties of $\mathcal{E}^{(k)}$ and provide an expression for its gradient. Let $p_{i:l}$, $l \in \{1,\dots,d\}$ denote the lth coordinate of p_i. For notational simplicity, let $\nabla_{i:l}$ denote the partial derivative operator with respect to $p_{i:l}$, i.e., $\nabla_{i:l} = \frac{\partial}{\partial p_{i:l}}$, and let ∇_i denote the gradient operator with respect to p_i, i.e., $\nabla_i = (\nabla_{i:1}, \dots, \nabla_{i:d})^T$. The following result establishes the smoothness of $\mathcal{E}^{(k)}$.

Proposition 6.2.4 *Assume that $f_1, \dots, f_p$ and the covariance of z are C^1 with respect to the spatial positions of their arguments. Then $\mathcal{E}^{(k)}$ is C^1 on $\Omega^{(k)}$. Furthermore, the gradient, $\nabla\mathcal{E}^{(k)}$ at P may be written as the nd-dimensional vector, $\nabla\mathcal{E}^{(k)}|_P = \big((\nabla_1\mathcal{E}^{(k)}(P))^T, \dots, (\nabla_n\mathcal{E}^{(k)}(P))^T\big)^T$, where the partial derivatives take the form,*

$$\begin{aligned}\nabla_{i:l}\mathcal{E}^{(k)}(P) &= tr\left(\Upsilon^{-1}\nabla_{i:l}\Upsilon\right) - \\ &- \frac{1}{2}\mathrm{row}_i\left(\mathbf{K} - \boldsymbol{I}\right)\mathrm{col}_i\left(\nabla_{i:l}\mathbf{K}\right), \ \textit{where} \\ \nabla_{i:l}\Upsilon &= \mathbf{F}\mathbf{K}^{-1}\nabla_{i:l}\mathbf{F}^T + \left(\mathbf{F}\mathbf{K}^{-1}\nabla_{i:l}\mathbf{F}^T\right)^T - \\ &- \mathbf{F}\mathbf{K}^{-1}\left(\nabla_{i:l}\mathbf{K}\right)\mathbf{K}^{-1}\mathbf{F}^T.\end{aligned}$$

Here the matrix partials are taken component-wise.

Remark 6.2.5 The second term of $\nabla_{i:l}\mathcal{E}^{(k)}(P)$, corresponding to the partial derivative of the second order Taylor series expansion, may be seen as a weighted sum of vectors in directions *away* from correlated measurement locations, with weights determined by the correlations. Thus optimizing this term forces the new measurements to be spread out from each other and from the previous measurements. This agrees with previous results [42, 71], and with the findings in Chapter 3, which suggest that the determinant of the covariance function may be optimized by maximizing the distances between the locations. •

Lemma 6.2.6 *Under the assumptions of Proposition 6.2.4, assume, in addition, that the partial derivatives of $f_1, \dots, f_p$ and the covariance of z are C^1 with respect to the spatial positions of their arguments. Then the map $P \to \nabla\mathcal{E}^{(k)}|_P$ is globally Lipschitz on $\Omega^{(k)}$.*

6.3 Adaptive sampling via distributed entropy optimization

The function $\mathcal{E}^{(k)}$ depends on all of its arguments as well as all of the past measurement locations (X_s) in a nontrivial and nonlinear way. In this section, we show how both $\mathcal{E}^{(k)}$ and $\nabla\mathcal{E}^{(k)}$ may be calculated in a distributed way over $\mathcal{N}$. This allows us to propose a distributed projected gradient descent algorithm which ensures that measurements are taken at local minima of $\mathcal{E}^{(k)}$ over $\Omega^{(k)}$.

6.3.1 Distributed calculations

Here, we describe a distributed method for calculating $\mathcal{E}^{(k)}$ and its gradient. In general, the matrices involved in the calculation depend on samples and locations known to multiple nodes. Furthermore, multiple samples and locations are known to each node. Distributed consensus algorithms may be performed in a similar manner whether each node knows one element or multiple elements, as long as the network is connected and each element is known by exactly one node. Since $\mathcal{V}(Q)$ describes a partition of the physical space, we may partition all measurement locations by region. Thus for each $(s,t) \in i_{\mathbb{F}}(X_s)$, there is exactly one $j \in \{1,\dots,m\}$ such that $s \in V_j(Q)$. Let ${R_{\text{in}}}^{(1:k)} : \mathbb{Z}_{>0} \to \mathbb{F}(\mathbb{Z}_{>0})$ and ${R_{\text{in}}}^{(k+1)} : \mathbb{Z}_{>0} \times \mathcal{D}^n \to \mathbb{F}(\mathbb{Z}_{>0})$ be defined as follows,

$$\begin{aligned} {R_{\text{in}}}^{(1:k)}(j) &= \{i \in \{1,\dots,nk\} \mid x_{s:i} = (s,t),\ s \in V_j(Q)\} \\ {R_{\text{in}}}^{(k+1)}(j,P) &= \{i + nk \mid i \in \{1,\dots,n\} \text{ and } p_i \in V_j(Q)\}\,. \end{aligned}$$

These index sets list columns of the matrices $\mathbf{K}$ and $\mathbf{F}$ which correspond to past $({R_{\text{in}}}^{(1:k)})$ and hypothetical future $({R_{\text{in}}}^{(k+1)})$ sample locations in the jth Voronoi cell. With a slight abuse of notation, define ${R_{\text{in}}}^{(1:k+1)} : \mathbb{Z}_{>0} \times \mathcal{D}^n \to \mathbb{F}(\mathbb{Z}_{>0})$ as the union of the two sets, ${R_{\text{in}}}^{(1:k+1)}(j,P) = {R_{\text{in}}}^{(1:k)}(j) \cup {R_{\text{in}}}^{(k+1)}(j,P)$. The following result shows how pieces of $\mathcal{E}^{(k)}$ can be calculated.

Lemma 6.3.1 *Let $P \in \mathcal{D}^n$ be a potential set of sites for the next measurement. Assume that N_j for each $j \in \{1,\dots,m\}$ knows the following quantities,*

- $\{x_{s:i} = (s,t) \in i_{\mathbb{F}}(X_s) \mid \mathrm{d}(s, V_j(Q)) < r\}$
- $\{p_i \in i_{\mathbb{F}}(P) \mid \mathrm{d}(p_i, V_j(Q)) < r\}$*;*
- $\mathbf{K}_0 \in \mathbb{R}^{p \times p}$*.*

Using consensus and distributed JOR [6] algorithms, the network can calculate the matrices $\mathbf{FK}^{-1}$ *and* Υ*. After running the algorithms,* N_j *has access to the quantities,* Υ*, and* $\mathrm{col}_i\left(\mathbf{FK}^{-1}\right) \in \mathbb{R}^p$*,* $i \in {R_{in}}^{(1:k+1)}(j, P)$*.*

Next we present our main distributed computation result.

Proposition 6.3.2 *For any* $j_1 \neq j_2 \in \{1, \dots, m\}$*, and any* $i_1 \in {R_{in}}^{(1:k+1)}(j_1, P)$ *and* $i_2 \in {R_{in}}^{(1:k+1)}(j_2, P)$*, assume that if* $[\mathbf{K}]_{i_1 i_2} \neq 0$ *then* N_{j_1} *can communicate with* N_{j_2}*. Then, under the assumptions of Lemma 6.3.1,* $\mathcal{E}^{(k)}$ *and its gradient at* $P \in \mathcal{D}^n$ *can be calculated in a distributed manner by* $\mathcal{N}$*.*

6.3.2 Distributed gradient descent algorithm

Here we outline a distributed version of the projected gradient descent algorithm (see, e.g. [5]), which is guaranteed to converge to a stationary point of $\mathcal{E}^{(k)}$ on $\Omega^{(k)}$. Let ${\kappa_j}^{(k)} : \mathcal{D}^n \to \mathbb{R}$ denote the partial sum,

$${\kappa_j}^{(k)}(P) = \sum_{i \in {R_{\mathrm{in}}}^{(1:k+1)}(j,P)} \mathrm{row}_i(\mathbf{K}^{(k)}(P) - \boldsymbol{I})\mathrm{col}_i(\mathbf{K}^{(k)}(P) - \boldsymbol{I}).$$

Then ${\kappa_j}^{(k)}(P)$ may be calculated by N_j, and $\mathrm{tr}\left(\left(\mathbf{K}^{(k)}(P) - \boldsymbol{I}\right)^2\right) = \sum_{j=1}^{m} {\kappa_j}^{(k)}(P)$. Table 6.1 describes a distributed line search with a starting position of $P \in \Omega$. The maximum stepsize, $\alpha_{\max}$, ensures that all robots with nonzero partial derivatives can move the maximum distance.

We are ready to present our technique for a greedy optimization algorithm. At timestep k, the nodes follow a gradient descent algorithm to define a sequence of configurations, $\{P_l^\dagger\}$, $l \in \mathbb{Z}_{>0}$, such that $P_1^\dagger$ is $P^{(k)} \in \mathcal{D}^n$, the vector of current spatial locations of the robotic agents and

$$P^{\dagger}_{l+1} = \mathrm{proj}_{\Omega}\left(P^{\dagger}_{l} - \alpha \nabla \mathcal{E}^{(k)}|_{P^{\dagger}_{l}}\right),\ \alpha \in \mathbb{R}_{\geq 0},$$

where α is chosen via Distributed Entropy Line Search Algorithm. When $|\mathcal{E}^{(k)}(P^{\dagger}_{l+1}) - \mathcal{E}^{(k)}(P^{\dagger}_{l})| = 0$, the algorithm terminates, and the nodes set $P^{(k+1)} = P^{\dagger}_{l+1}$. By the end of this calculation, each node knows the identity of robotic agents in its Voronoi cell at timestep $k+1$. Node N_j transmits $p_i(k+1)$ to robot R_i, which then moves to that location between timesteps. The overall algorithm is in Table 6.2.

Proposition 6.3.3 *The* Distributed Entropy Projected Gradient Descent Algorithm *is distributed over the network $\mathcal{N}$. Moreover, under the assumptions of Lemma 6.2.6, any execution is such that the robots do not collide and, at each timestep after the first, measurements are taken at stationary configurations of $P \mapsto \mathcal{E}^{(k)}(P)$ over $\Omega^{(k)}$.*

6.3.3 Simulations

We implemented the Distributed Entropy Projected Gradient Descent Algorithm in several simulations. The one presented here was run with d=2 spatial dimensions, m=10 static nodes, n=30 robotic agents, and the domain $\mathcal{D}$=$\{(0,.1),\ (2.5,.1),\ (3.45,1.6),\ (3.5,1.7),\ (3.45,1.8),\ (2.7,2.2),\ (1,2.4),\ (0.2,1.3)\}$. We used the separable covariance function defined by $\mathrm{Cov}[z(s_1,t_1), z(s_2,t_2)] = C_{\mathrm{tap}}(\|s_1 - s_2\|, 0.5)C_{\mathrm{tap}}(|t_1 - t_2|, 6.5)$, where

$$C_{\mathrm{tap}}(\delta, r) = \begin{cases} e^{-\frac{\delta}{10r}}\left(1 - \frac{3\delta}{2r} + \frac{\delta^3}{2r^3}\right) \text{ if } \delta \leq r, \\ 0 \text{ otherwise.} \end{cases}$$

This is a tapered exponential function belonging to the class of covariance functions suggested in [27]. We used $\omega = 0.02$, and $u_{\max} = 0.2$. For the mean regression functions f_i, we used $f((x,y),t) = (1, \sin(2\pi x), \sin(2\pi y))^T$.

Fig. 6.1 shows the trajectories taken by the robots. In Fig. 6.2 we compare the performance of our algorithm against two algorithms that pre-plan agent trajectories. The first is a static approach in which the agents spread out around the region and

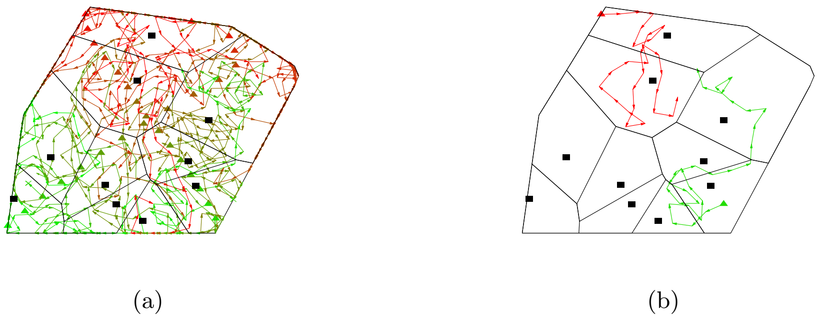


Figure 6.1: (a) Trajectories of all robots, and (b) two representative trajectories, both from the same run of the distributed projected gradient descent algorithm. The filled squares represent the (static) positions of the nodes, and the filled triangles show the starting positions of the robots.

remain in place. The second is a lawnmower-type algorithm in which the agents march back and forth across the region in evenly space (horizontal) lines. In all cases, two agents lost contact part way through. Note that both dynamic algorithms perform much better than the static one, but the gradient descent algorithm performs better than the lawnmower. This is due to the facts that the gradient algorithm reacts to the basis functions of the model and that the lawnmower does not compensate for the dropped agents.

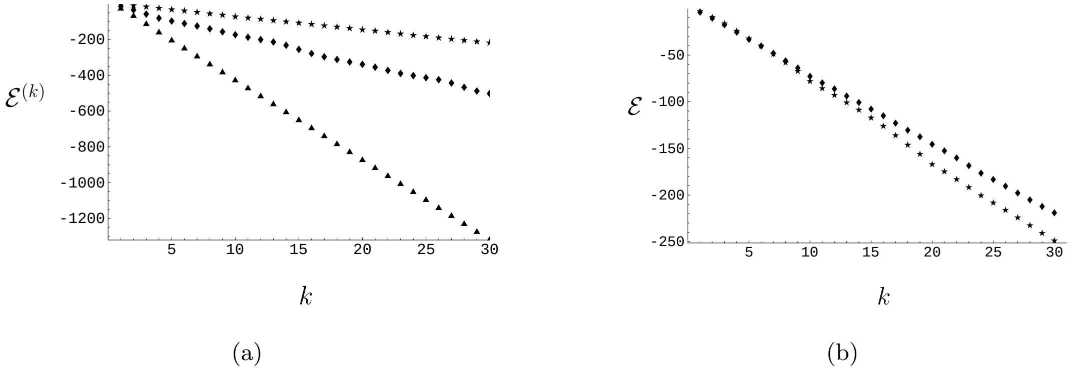


Figure 6.2: Plot (a) shows the progression of $\mathcal{E}^{(k)}$ as k increases, resulting from the static (triangle), lawnmower (diamond), and gradient descent (star) approaches. For the gradient descent algorithm only, plot (b) compares the value of $\widetilde{\mathcal{E}}$ (stars) against the approximation, $\mathcal{E}^{(k)}$ (diamonds).

Name: Distributed Entropy Line Search Algorithm

Goal: Compute step size for gradient descent of $\mathcal{E}^{(k)}$

Input: Configuration, $P = (p_1, \dots, p_n) \in \mathcal{D}^n$

Assumes: (i) Connected network of static nodes
(ii) N_j knows $\mathcal{E}^{(k)}(P)$, as well as p_i, $\nabla_i \mathcal{E}^{(k)}(P)$, $\mathrm{row}_i(\mathbf{K} - \boldsymbol{I})$ and ${\Omega_i}^{(k)}$ for each robot within communication range
(iii) Shrinkage factor τ, tolerance $\theta \in (0,1)$, and prior β-correlation matrix, $\mathbf{K}_0$ known a priori

Uses: (i) $p_i'(\alpha, P) = \mathrm{proj}_{{\Omega_i}^{(k)}}(p_i + \alpha \nabla_i \mathcal{E}^{(k)}(P))$
(ii) Square distance of robots entering $V_j(Q)$,

$$\mathrm{d}_j\left(\alpha, P\right)^2 = \sum_{\substack{i \in \{1,\dots,n\} \text{ such that} \\ p_i'(\alpha,P) \in V_j(Q)}} \|p_i'(\alpha, P) - p_i\|^2$$

Output: Step size $\alpha \in \mathbb{R}$, next configuration $P'(\alpha, P) = (p_1'(\alpha, P), \dots, p_n'(\alpha, P))^T$, and $\mathcal{E}^{(k)}(P'(\alpha, P))$.

Initialization

```
$\mathrm{N}_1, \dots, \mathrm{N}_m$ calculate $\alpha_{\max} = \max\left\{\|\nabla_i \mathcal{E}^{(k)}(P)\|^{-1} \middle| \nabla_i \mathcal{E}^{(k)}(P) \neq 0\right\} u_{\max}$ via *maximum consensus*
```

For $j \in \{1, \dots, m\}$, node N_j sets $\alpha = \alpha_{\max}$ and executes concurrently

```
**repeat**
    calculates $\Upsilon$ according to Lemma 6.3.1
    calculates $\mathrm{d}_j\left(\alpha, P\right)^2$ and ${\kappa_j}^{(k)}(P'(\alpha, P))$
    executes consensus algorithm to calculate the following:
        $\mathrm{tr}\left((\mathbf{K} - \boldsymbol{I})^2\right) = \sum_{j=1}^{m} {\kappa_j}^{(k)}(P'(\alpha, P))$ and $\|P - P'(\alpha, P)\|^2 = \sum_{j=1}^{m} \mathrm{d}_j\left(\alpha, P\right)^2$
    $\mathcal{E}^{(k)}\left(P'(\alpha, P)\right) = \log\det\left(\Upsilon\right) + \mathrm{tr}\left((\mathbf{K} - \boldsymbol{I})^2\right)$
    $\varpi = \frac{\theta}{\alpha}\|P - P'(\alpha, P)\|^2 - \mathcal{E}^{(k)}(P'(\alpha, P)) + \mathcal{E}^{(k)}(P)$
    **if** $\varpi > 0$ **then**
        $\alpha = \alpha\tau$
    **end if**
**until** $\varpi \leq 0$
```

Table 6.1: Distributed Entropy Line Search Algorithm.

Name: Distributed Entropy Projected Gradient Descent Algorithm

Goal: Find a local minimum of $\mathcal{E}^{(k)}$ within $\Omega^{(k)}$.

Assumes: (i) Connected network of nodes and robots

(ii) Static nodes deployed over $\mathcal{D}$ such that $R \geq \max_{i \in \{1,\ldots,m\}} \{\mathrm{CR}(V_i(Q))\} + u_{\max}$, initial configuration $P^{(1)} \in \mathcal{D}^n$

(iii) Line search shrinkage factor τ, tolerance $\theta \in (0,1)$, and prior β-correlation matrix, $\mathbf{K}_0$ known a priori by all nodes.

```
At time $k \in \mathbb{Z}_{\geq 0}$, robot $\mathrm{R}_i$ executes:
  takes measurement at $p_i(k)$
  sends position to $\mathrm{N}_j$, where $p_i(k) \in V_j(Q)$
  receives next location $p_i(k+1)$
  moves to $p_i(k+1)$.
At time $k \in \mathbb{Z}_{\geq 0}$, node $\mathrm{N}_j$ executes:
  collects location from each $\mathrm{R}_i$ with $\mathrm{d}(p_i(k), V_j(Q)) < u_{\max}$ as well as locations of nearby agents
  updates ${R_{\mathrm{in}}}^{(k+1)}(j, P)$ and ${R_{\mathrm{in}}}^{(1:k+1)}(j)$
  calculates $\Upsilon$ (cf. Lemma 6.3.1)
  computes ${\kappa_j}^{(k)}\left(P^{(k)}\right)$, and then $\mathcal{E}^{(k)}\left(P^{(k)}\right)$ via consensus
  sets $P_{\mathrm{next}} = P^{(k)}$
  repeat
     stores $P_{\mathrm{cur}} = P_{\mathrm{next}}$ and $\mathcal{E}^{(k)}(P_{\mathrm{cur}}) = \mathcal{E}^{(k)}(P_{\mathrm{next}})$
     calculates $-\nabla_i \mathcal{E}^{(k)}(P_{\mathrm{cur}})$ for each $i \in {R_{\mathrm{in}}}^{(k+1)}(j, P_{\mathrm{cur}})$ (cf. Prop. 6.3.2)
     runs Distributed Entropy Line Search Algorithm at $P_{\mathrm{cur}}$ to get $\alpha$, $P_{\mathrm{next}}$, and $\mathcal{E}^{(k)}(P_{\mathrm{next}})$
  until $|\mathcal{E}^{(k)}(P_{\mathrm{next}}) - \mathcal{E}^{(k)}(P_{\mathrm{cur}})| = 0$
  sets $P^{(k+1)} = P_{\mathrm{next}}$
  conveys $p_i(k+1)$ to $\mathrm{R}_i$ for each $i \in {R_{\mathrm{in}}}^{(k+1)}(j, P_{\mathrm{cur}})$
```

Table 6.2: Distributed Entropy Projected Gradient Descent Algorithm.

Chapter 7

Conclusions and future work

In Chapter 3, we have used the maximum error variance and the extended variance of the LUMVE as metrics for optimal placement of mobile sensor networks estimating random fields. We have shown that under the assumption of near independence, circumcenter configurations minimize the maximum error variance and incenter configurations minimize the extended variance of the estimator. Under limited time or energy resources, or as a starting point for further exploration, a group of robotic sensors can begin by moving toward these configurations to start the estimation procedure. Future work in this area will explore: (i) regarding the asymptotic analysis, the determination of lower and upper bounds on the parameter α that guarantee that multicenter Voronoi configurations achieve a given a desired level of performance. In particular, we would like to determine the near-optimality in general of incenter Voronoi configurations for the extended variance criterion; and (ii) the extension of the results to similar error metrics for the universal kriging predictor, where the mean function is unknown.

In Chapter 4, we have considered a robotic sensor network taking samples of a spatio-temporal process. As criterion for optimization we have taken the maximum predictive variance of the prediction made at the end of the experiment. Under the asymptotic regime of near-independence, we have shown that minimizing this error is equivalent to minimizing the correlation distance disk-covering function, thus allowing geometric solutions. We have introduced the maximal correlation partition and showed that it is the optimal partition of the predictive space for the disk-covering function given a fixed network trajectory. We have introduced the novel notion of multicircumcenter trajectories and established their optimality with regards to the disk-covering function given a fixed partition. We have also defined a notion of extended sets which encodes a maximum movement restriction into a form of geometric centering, yielding the constrained multicircumcenter trajectory which is optimal over the set of all range-constrained trajectories. On the design front, we have synthesized distributed strategies that allow the network to calculate an optimal trajectory. In an ongoing experiment, the optimization can be executed online to recalculate the remaining sample locations in the face of changes in the environment, network structure, or human input. Future work will include the study of more complex predictive regions and of alternative optimality criteria.

In Chapter 5, we have considered a network of static computing nodes and mobile robotic sensing platforms taking measurements of a time-varying random process with covariance known up to a scaling parameter. We have used a Bayesian approach, treating the field as a spatiotemporal Gaussian random process, and developed a novel approximation of the variance of the posterior predictive distribution which may be calculated in a sequential and distributed fashion. Using this formulation, we have developed a projected gradient descent algorithm which is distributed over the network of nodes and robots. We have examined the complexity of this approach, and compared it against the lower bound complexity of a centralized "broadcast" method, showing that the distributed approach scales better with the number of mobile agents. Future work will focus on theoretical guarantees on the accuracy of the approximation $\tilde{\mathcal{A}}^{(k)}$ (see Remark 5.2.6) and on the robustness to failure of the proposed coordination algorithm. An interesting topic of future work would be the extension of these methods to a hybrid network in which the static agents are replaced by slow moving ones. As mentioned in Remark 4.6.3, special care must be taken to avoid singularities when generating local approximations for the universal kriging model. A topic of future work will be to provide rigorous methods for handling this situation.

In Chapter 6, we have designed a distributed algorithm for adaptive sampling of spatiotemporal processes with unknown mean and covariance known up to a scaling parameter. At each time step, an heterogeneous network composed of static nodes and mobile agents optimizes an aggregate objective function to maximize the information provided by future data. We have shown that the objective function is a second-order approximation of the conditional entropy, defined as the posterior predictive entropy conditional on the covariance scaling parameter. We have characterized the correctness of the proposed coordination algorithm and provided several simulations of its performance. Immediate future work will investigate the invariance of the region $\mathcal{T}^{(k)}$ under the gradient of $\mathcal{E}^{(k)}$, comparison against a smarter, self-adjusting lawnmower algorithm, and quantification of the communication and computational complexity of the algorithms. In the longer term, we plan to continue exploring methods to cooperatively estimate stochastic processes considering statistical models with increasing generality.

Appendix A

Proofs and supporting results

A.1 Proofs and supporting results from Chapter 3

Proof: [Proof of Proposition 3.3.4] We proceed by contradiction. If the statement is false, then there exists $s^\dagger \in \mathcal{D}$ such that $s^\dagger \in \operatorname{argmin}_{s\in\mathcal{D}} \{C_{\text{mds}}(s, P)\,|\text{mds}(s, P)|\}$, and $\left|\text{mds}(s^\dagger, P)\right| > 1$. Let $p^* \in \text{mds}(s^\dagger, P)$, and define $r^\dagger = \|s^\dagger - p^*\|$. Note that

$$\text{mds}(s^\dagger, P) \subset \partial\overline{B}(s^\dagger, r^\dagger). \tag{A.1}$$

Let $s^* \in]s^\dagger, p^*[$ such that $\|s^* - s^\dagger\| < \epsilon$ for some $\epsilon \in \mathbb{R}_{>0}$ and let $r^* = \|s^* - p^*\|$. By construction, $r^* < r^\dagger$. From (A.1), we deduce that $\left\{p \in i_{\mathbb{F}}(P) \mid p \in \overline{B}(s^*, r^*)\right\} = \{p^*\}$, which leads to $|\text{mds}(s^*, P)| = 1$. Since g_s changes continuously with the distance between its arguments, it is clear that we may choose ϵ small enough to result in

$$C_{\text{mds}}(s^*, P)\,|\text{mds}(s^*, P)| < C_{\text{mds}}(s^\dagger, P)\left|\text{mds}(s^\dagger, P)\right|,$$

which is a contradiction. ■

A.1.1 Continuity

Here we prove our main continuity result for the estimation error variance function. We will first need some supporting results. Let $h \in \mathbb{R}$ denote the distance between agent locations p_1 and p_2, and we are interested in the behavior of Var_{SK} as

$h \to 0$. For any agent location p_i, $i \in (3, \ldots, n)$ we would like a measure of the distance between p_i and p_1 in terms of h, call it $h_i \in \mathbb{R}$. To find h_i, consider the triangle formed between p_i, p_1, and p_2. Let $\mathbf{u}_{jk} \in \mathbb{R}^d$ denote the unit vector from agent j to agent k, i.e.,

$$\mathbf{u}_{jk} = \frac{p_k - p_j}{\|p_k - p_j\|}.$$

Let $\mathbf{u}_{ib}$ denote the unit vector in the direction which bisects the angle between $\mathbf{u}_{i1}$ and $\mathbf{u}_{i2}$ (see Figure A.1). As in the figure, consider a point p_1^* which is a distance of $\|p_1 - p_i\|$ from p_i, but in the direction $\mathbf{u}_{i2}$, and note that $h_i = \|p_1^* - p_2\|$. Next, note that the projection of the vector $p_1^* - p_2$ onto $\mathbf{u}_{ib}$ is equal to the projection of the vector $p_1 - p_2$ onto the vector $\mathbf{u}_{ib}$, so that we may write

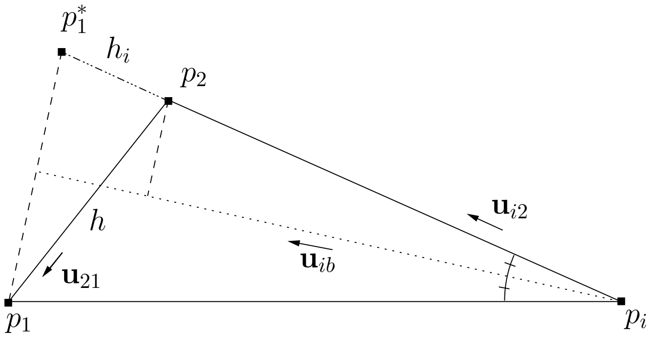


Figure A.1: Triangle between p_i, p_1, and p_2.

$$h_i \mathbf{u}_{ib}{}^T \mathbf{u}_{i2} = \mathbf{u}_{ib}{}^T \mathbf{u}_{21} h$$
$$h_i = \frac{\mathbf{u}_{ib}{}^T \mathbf{u}_{21}}{\mathbf{u}_{ib}{}^T \mathbf{u}_{i2}} h.$$

Note that in the limit as $p_1 \to p_2$, $h_i \to 0$ and $\mathbf{u}_{ib} \to \mathbf{u}_{i2}$.

In the following treatment, we define a function $f : \mathbb{R}^d \to \mathbb{R}$ to be directionally differentiable at a point $a \in \mathbb{R}^d$ if and only if the following limit exists for every $\mathbf{u} \in \mathbb{R}^d$

$$D_{\mathbf{u}} f(a) = \lim_{h \downarrow 0} \frac{f(a + h\mathbf{u}) - f(a)}{h}.$$

This is a common notion in the optimization literature (see for example [23, 70]), where it is sometimes referred to as weak directional differentiability. We will say that a function f is directionally differentiable on D if and only if $D_{\mathbf{u}}f(a)$ exists for all $a \in D$ and all $\mathbf{u} \in \mathbb{R}^d$. Recall from Lemma 3.3.1 the definition of $\overline{P} = (p_2, \dots, p_n)$ being the ordered set of agents in P with the first agent removed. Now we are ready to present our results.

Lemma A.1.1 *Let $f_1, f_2 : \mathbb{R}^d \to \mathbb{R}$ be directionally differentiable on $\mathcal{D}$. Let $F : \mathbb{R}^d \times \mathbb{R}^d \times \mathcal{D}^{n-1} \to \mathbb{R}$ be defined as*

$$F(p_1, p_2, \overline{P}) = \frac{(f_1(p_1) - f_1(p_2))(f_2(p_1) - f_2(p_2))}{\mathrm{Var}_{SK}[p_1; \overline{P}]}.$$

Under the assumption that $g_s'(0) \neq 0$ then

$$\lim_{p_1 \to p_2} F(p_1, p_2, \overline{P}) = 0.$$

Proof: First note that in the limit as $p_1 \to p_2$, both the numerator and denominator of the fraction tend to zero. With a little manipulation, we can rewrite F in terms of h. Remembering that $\mathrm{Cor}[\overline{P}, p_2]$ is the first column of the matrix $\mathbf{K}(\overline{P})$, we can write

$$F(p_1, p_2, \overline{P}) = \frac{(f_1(p_1) - f_1(p_2))(f_2(p_1) - f_2(p_2))}{\left(\mathrm{Cor}[\overline{P}, p_1]^T + \mathrm{Cor}[\overline{P}, p_2]^T\right) \mathbf{K}_\tau(\overline{P})^{-1} \left(\mathrm{Cor}[\overline{P}, p_2] - \mathrm{Cor}[\overline{P}, p_1]\right)}.$$

In terms of g_s, we can write

$$\begin{aligned}
\mathrm{Cor}[\overline{P}, p_2] - \mathrm{Cor}[\overline{P}, p_1] &= \begin{bmatrix} g_s(\|p_2 - p_2\|) - g_s(\|p_1 - p_2\|) \\ g_s(\|p_2 - p_3\|) - g_s(\|p_1 - p_3\|) \\ g_s(\|p_2 - p_4\|) - g_s(\|p_1 - p_4\|) \\ \vdots \end{bmatrix} \\
&= \begin{bmatrix} \frac{g_s(0) - g_s(h)}{h} h \\ \frac{g_s(\|p_2 - p_3\|) - g_s(\|p_1 - p_3\|)}{h_3} \frac{\mathbf{u}_{3b}{}^T \mathbf{u}_{21}}{\mathbf{u}_{3b}{}^T \mathbf{u}_{32}} h \\ \frac{g_s(\|p_2 - p_4\|) - g_s(\|p_1 - p_4\|)}{h_4} \frac{\mathbf{u}_{4b}{}^T \mathbf{u}_{21}}{\mathbf{u}_{4b}{}^T \mathbf{u}_{42}} h \\ \vdots \end{bmatrix} \\
&= -\Delta h,
\end{aligned}$$

where $\Delta \in \mathbb{R}^{n-1}$ is the vector

$$\Delta = \begin{bmatrix} \frac{g_s(h)-g_s(0)}{h} \\ \frac{g_s(\|p_2-p_3\|+h_3)-g_s(\|p_2-p_3\|)}{h_3} \frac{\mathbf{u}_{3b}{}^T\mathbf{u}_{21}}{\mathbf{u}_{3b}{}^T\mathbf{u}_{32}} \\ \frac{g_s(\|p_2-p_4\|+h_4)-g_s(\|p_2-p_4\|)}{h_4} \frac{\mathbf{u}_{4b}{}^T\mathbf{u}_{21}}{\mathbf{u}_{4b}{}^T\mathbf{u}_{42}} \\ \vdots \end{bmatrix}.$$

Note that the limit as $h \to 0$ corresponds to the limit as $p_1 \to p_2$ along the straight line direction $\mathbf{u}_{12}$, and that

$$\lim_{h \to 0} \Delta = \begin{bmatrix} g_s'(0) \\ \mathbf{u}_{32}{}^T\mathbf{u}_{21} g_s'(\|p_2-p_3\|) \\ \mathbf{u}_{42}{}^T\mathbf{u}_{21} g_s'(\|p_2-p_4\|) \\ \vdots \end{bmatrix}.$$

Plugging this back into F, we can write

$$\begin{aligned} &F(p_1, p_2, \overline{P}) \\ &= \frac{h^2 \left(\frac{(f_1(p_2+h\mathbf{u}_{21})-f_1(p_2))}{h}\right)\left(\frac{(f_2(p_2+h\mathbf{u}_{21})-f_2(p_2))}{h}\right)}{-h\left(\operatorname{Cor}[\overline{P}, p_1]^T + \operatorname{Cor}[\overline{P}, p_2]^T\right)\mathbf{K}_\tau(\overline{P})^{-1}\Delta} \\ &= \frac{h \left(\frac{(f_1(p_2+h\mathbf{u}_{21})-f_1(p_2))}{h}\right)\left(\frac{(f_2(p_2+h\mathbf{u}_{21})-f_2(p_2))}{h}\right)}{-\left(\operatorname{Cor}[\overline{P}, p_1]^T + \operatorname{Cor}[\overline{P}, p_2]^T\right)\mathbf{K}_\tau(\overline{P})^{-1}\Delta}. \end{aligned}$$

Note that for any directional unit vector $\mathbf{u} \in \mathbb{R}^d$,

$$\begin{aligned} \lim_{\substack{p_1 \to p_2 \\ along \mathbf{u}}} F(p_1, p_2, \overline{P}) = \left(D_{\mathbf{u}} f_1(p_2)\right)\left(D_{\mathbf{u}} f_2(p_2)\right) \times \\ \left(\lim_{h \to 0} \frac{h}{-\left(\operatorname{Cor}[\overline{P}, p_1]^T + \operatorname{Cor}[\overline{P}, p_2]^T\right)\mathbf{K}_\tau(\overline{P})^{-1}\Delta}\right). \end{aligned}$$

Regardless of the direction of $\mathbf{u}$, the numerator approaches zero. In the limit, the denominator evaluates to

$$-2\operatorname{Cor}[\overline{P}, p_2]^T \mathbf{K}_\tau(\overline{P})^{-1} \begin{bmatrix} g_s'(0) \\ \mathbf{u}_{32}{}^T\mathbf{u}_{21} g_s'(\|p_2-p_3\|) \\ \mathbf{u}_{42}{}^T\mathbf{u}_{21} g_s'(\|p_2-p_4\|) \\ \vdots \end{bmatrix} = -2g_s'(0).$$

Since this is constant with respect to the direction of approach, as long as $g_s'(0) \neq 0$, we have

$$\lim_{p_1 \to p_2} F(p_1, p_2, \overline{P}) = 0.$$

$\blacksquare$

Corollary A.1.2 *Under the assumption of zero measurement error, if $g_s'(0) \neq 0$ then*

$$\lim_{p_1 \to p_2} \mathrm{Var}_{SK}[z(s); P] = \mathrm{Var}_{SK}[z(s); \overline{P}]$$

Proof: Using Lemma 3.3.1 we can write

$$\begin{aligned}
\mathrm{Var}_{\mathrm{SK}}[z(s); P] &= \mathrm{Var}_{\mathrm{SK}}[z(s); \overline{P}] + \frac{\big(g_s(\|s - p_1\|) - \mathrm{Cor}[\overline{P}, s]^T \mathbf{K}_\tau(\overline{P})^{-1} \mathrm{Cor}[\overline{P}, p_1]\big)^2}{\mathrm{Var}_{\mathrm{SK}}[y_1; \overline{P}]} \\
&= \mathrm{Var}_{\mathrm{SK}}[z(s); \overline{P}] + \\
&\quad \frac{\big(g_s(\|s - p_1\|) - \mathrm{Cor}[\overline{P}, s]^T \mathbf{K}_\tau(\overline{P})^{-1} \delta - g_s(\|s - p_2\|)\big)^2}{\mathrm{Var}_{\mathrm{SK}}[y_1; \overline{P}]},
\end{aligned}$$

where $\delta = \mathrm{Cor}[\overline{P}, p_1] - \mathrm{Cor}[\overline{P}, p_2]$. Note that the second term here can be multiplied out as

$$\begin{aligned}
&\frac{(g_s(\|s - p_1\|) - g_s(\|s - p_2\|))^2}{\mathrm{Var}_{\mathrm{SK}}[y_1; \overline{P}]} + \\
&\qquad 2 \frac{(g_s(\|s - p_1\|) - g_s(\|s - p_2\|)) \,\mathrm{Cor}[\overline{P}, s]^T \mathbf{K}_\tau(\overline{P})^{-1} \delta}{\mathrm{Var}_{\mathrm{SK}}[y_1; \overline{P}]} + \\
&\qquad\qquad \frac{\big(\mathrm{Cor}[\overline{P}, s]^T \mathbf{K}_\tau(\overline{P})^{-1} \delta\big)^2}{\mathrm{Var}_{\mathrm{SK}}[y_1; \overline{P}]}.
\end{aligned}$$

Since C is directionally differentiable everywhere, the first term fits the criteria of Lemma A.1.1, and goes to zero in the limit. For the other two, note that

$$\mathrm{Cor}[\overline{P}, s]^T \mathbf{K}_\tau(\overline{P})^{-1} \delta = \sum_{i=1}^{n} \alpha_i \left(g_s(\|p_1 - p_i\|) - g_s(\|p_2 - p_i\|) \right),$$

where the α_i's do not depend on p_1. By Lemma A.1.1, for all i, j in $(1, \dots, n)$ we can say

$$\lim_{p_1 \to p_2} \frac{\left(g_s(\|p_1 - p_i\|) - g_s(\|p_2 - p_i\|)\right)\left(g_s(\|s - p_1\|) - g_s(\|s - p_2\|)\right)}{\mathrm{Var}_{\mathrm{SK}}[y_1; \overline{P}]} = 0,$$
$$\lim_{p_1 \to p_2} \frac{\left(g_s(\|p_1 - p_i\|) - g_s(\|p_2 - p_i\|)\right)\left(g_s(\|p_1 - p_j\|) - g_s(\|p_2 - p_j\|)\right)}{\mathrm{Var}_{\mathrm{SK}}[y_1; \overline{P}]} = 0.$$

Thus all of the parts of our equation which depend on p_1 go to zero in the limit and we are left with

$$\lim_{p_1 \to p_2} \mathrm{Var}_{\mathrm{SK}}[z(s); P] = \mathrm{Var}_{\mathrm{SK}}[z(s); \overline{P}].$$

■

We are now ready to present our main continuity result.

Proof: [Proof of Proposition 3.3.2] Let $s \in \mathcal{D}$. Note that the map $P \mapsto \mathrm{Var}_{\mathrm{SK}}[z(s); P]$ is continuous when $P \in \mathcal{D}^n \setminus S_{\mathrm{coinc}}$. Since the correlation function g_s is differentiable and hence continuous, it follows from Proposition 2.5.3 that $\mathrm{Var}_{\mathrm{SK}}$ is continuous with respect to $P = (p_1, \dots, p_n) \in \mathcal{D}^n$ except possibly where the matrix $\mathbf{K}_\tau(P)$ is not full rank. With $\tau^2 \neq 0$, we have that $\mathbf{K}_\tau$ is always full rank. Therefore, let us consider the ideal sensor case in which $\tau^2 = 0$, $Y(p_i) = Z(p_i)$, and $\mathbf{K}_\tau(P) = \mathbf{K}(P)$. Note that $\mathbf{K}$ being rank deficient corresponds precisely to the case when $P \in S_{\mathrm{coinc}}$. Let us then take $P^\dagger \in S_{\mathrm{coinc}}$. It suffices to show that

$$\lim_{P \to P^\dagger} \mathrm{Var}_{\mathrm{SK}}[z(s); P] = \mathrm{Var}_{\mathrm{SK}}[z(s); i_{\mathbb{F}}(P^\dagger)], \tag{A.2}$$

where $P \in \mathcal{D}^n \setminus S_{\mathrm{coinc}}$. We begin by considering the case in which only two agents sit at the same location in the configuration $P^\dagger$. Since $\mathrm{Var}_{\mathrm{SK}}$ is invariant under permutations of the agents, without loss of generality we can assume that $p_1^\dagger = p_2^\dagger$. Let then $P^\dagger = (p_1^\dagger, \overline{P})$. Since all points in $\overline{P}$ are distinct, we have

$$\lim_{P \to P^\dagger} \mathrm{Var}_{\mathrm{SK}}[z(s); P] = \lim_{p_1 \to p_1^\dagger = p_2^\dagger} \mathrm{Var}_{\mathrm{SK}}[z(s); P].$$

Using Corollary A.1.2, we can write

$$\lim_{p_1 \to p_1^\dagger = p_2^\dagger} \mathrm{Var}_{\mathrm{SK}}[z(s); P] = \mathrm{Var}_{\mathrm{SK}}[z(s); \overline{P}]. \tag{A.3}$$

Since $\overline{P}$ is a specific ordering of $i_{\mathbb{F}}(P^\dagger)$, equation result (A.2) follows.

The case when more than two points in $P^\dagger$ are coincident can be dealt with similarly. If $|i_{\mathbb{F}}(P^\dagger)| = m \leq n-2$, we assume without loss of generality that $i_{\mathbb{F}}(P^\dagger) = \{p^\dagger_{m+1}, \dots, p^\dagger_n\}$ using the fact that $\mathrm{Var}_{\mathrm{SK}}$ is invariant under permutations. Then, we have

$$\lim_{P \to P^\dagger} \mathrm{Var}_{\mathrm{SK}}[z(s); P] = \lim_{p_1 \to p_1^\dagger} \lim_{p_2 \to p_2^\dagger} \dots \lim_{p_m \to p_m^\dagger} \mathrm{Var}_{\mathrm{SK}}[z(s); P].$$

Repeatedly using (A.3), the limit above is well defined and, moreover, we conclude (A.2).

■

A.2 Proofs and supporting results from Chapter 4

The supporting results for Chapter 4 have been organized here according to section.

Proofs and supporting results from Section 4.2

We begin with some notation and preliminary results. We define the *minimal correlation distance set* (MCDS), denoted by mcds : $\mathcal{D} \times (\mathcal{D}^{k_{\max}})^n \to \mathbb{F}(I_{\mathrm{samp}})$, as,

$$\mathrm{mcds}(s, S) = \underset{(i,k) \in I_{\mathrm{samp}}}{\operatorname{argmin}} \left\{ \delta_k(s, {s_i}^{(k)}) \right\}.$$

Note that mcds defines the set of samples in S with the highest correlation to s. Let $g_{\max} : \mathcal{D} \times (\mathcal{D}^{k_{\max}})^n \to \mathbb{R}$ map location and trajectory to this maximal correlation value, i.e.,

$$g_{\max}(s, S) = g_s(\|s - {s_i}^{(k)}\|) g_t(k_{\max}, k), \quad \forall\, (i,k) \in \mathrm{mcds}(s, S).$$

The following result describes a useful result on the dimensionality of the intersection of any two correlation distance surfaces.

Lemma A.2.1 (Equidistant sets are at most $d{-}1$ dimensional surfaces) *Assume that $S \in S_{unique}$, and let $(i,k), (j,l) \in I_{\mathrm{samp}}$. Let $\gamma = \left\{ s \in \mathbb{R}^d \mid \delta_k(s, {s_i}^{(k)}) = \delta_l(s, {s_j}^{(l)}) \right\}$.*

Then $\gamma = \mathbb{R}^d$ if and only if $(i,k) = (j,l)$. Otherwise, if $\gamma \neq \emptyset$, then it describes a surface in $\mathbb{R}^d$ which is at most $d-1$ dimensional.

Proof: First, consider the shape of the correlation distance surfaces $s \mapsto \delta_k(s, {s_i}^{(k)})$ and $s \mapsto \delta_l(s, {s_j}^{(l)})$ in $\mathbb{R}^{d+1}$. From (4.2), it can be seen that the two surfaces differ only by a translation which is a result of both the spatial and temporal locations of the sample. The assumption that $S \in S_{\text{unique}}$ implies that $\gamma = \mathbb{R}^d$ if and only if $(i,k) = (j,l)$. Next, assume $\gamma \neq \mathbb{R}^d$ and $\gamma \neq \emptyset$. It can be shown that either the two correlation distance surfaces are tangent and that the tangent surface is contained within a one-dimensional line, or the gradient of the function $s \mapsto \delta_k(s, {s_i}^{(k)}) - \delta_l(s, {s_j}^{(l)})$ over $\gamma \setminus \{{s_i}^{(k)}, {s_j}^{(l)}\}$ is nonzero, implying that the dimension of γ is at most $d-1$. ■

The above lemma allows the following result on the cardinality of the MCDS.

Proposition A.2.2 (Cardinality of MCDS) *Assume that $S \in S_{unique}$. Then,*

$$\min_{s \in \mathcal{D}} \left\{ g_{\max}(s,S) \,|\, \text{mcds}(s,S)| \right\} = \min_{s \in \mathcal{D}} \left\{ g_{\max}(s,S) \right\}.$$

Proof: We proceed by contradiction. If the statement is false, then there exists $s^\dagger \in \mathcal{D}$ such that $s^\dagger \in \text{argmin}_{s \in \mathcal{D}} \left\{ g_{\max}(s,S) \,|\, \text{mcds}(s,S)| \right\}$, and $|\text{mcds}(s^\dagger, S)| > 1$. Define $\Gamma \subset \mathcal{D}$ by $\Gamma = \{ s \in \mathcal{D} \mid |\text{mcds}(s,S)| > 1 \}$. Note that $s^\dagger \in \Gamma$, and $\Gamma \subseteq \bigcup_{i \neq j} \gamma_{ij}$. Lemma A.2.1 shows that Γ is the union of a finite number of surfaces of dimension at most $d-1$ embedded in $\mathbb{R}^d$. For any $\epsilon \in \mathbb{R}_{>0}$, there is a location $s^* \in \mathcal{D} \setminus \Gamma$ which satisfies $\|s^\dagger - s^*\| < \epsilon$. Thus $|\text{mcds}(s^*, S)| = 1$. Since $g_{\max}(s,S)$ changes continuously with s, for ϵ small enough we have, $g_{\max}(s^*, S)|\text{mcds}(s^*, S)| < g_{\max}(s^\dagger, S)|\text{mcds}(s^\dagger, S)|$, which is a contradiction. ■

We are now ready to prove the main result.

Proof: [Proof of Theorem 4.2.2] Note that minimizing $\mathcal{M}^{\{\alpha\}}$ on Ω_{Rg} is equivalent to maximizing the function $L^{\{\alpha\}} : \Omega_{\text{Rg}} \to \mathbb{R}$ defined by $L^{\{\alpha\}}(S) = \min_{s \in \mathcal{D}} \left\{ (\mathbf{k}^{\{\alpha\}})^T \times (\mathbf{K}_\tau^{\{\alpha\}})^{-1} (\mathbf{k}^{\{\alpha\}}) \right\}$. Let $\lambda_{\min}$ and $\lambda_{\max} : \Omega_{\text{Rg}} \times \mathbb{R} \to \mathbb{R}$ be such that $\lambda_{\min}(S, \alpha)$, $\lambda_{\max}(S, \alpha)$ denote, respectively, the minimum and the maximum eigenvalue of $\mathbf{K}_\tau^{\{\alpha\}}$. Note that with $\tau^2 \neq 0$, we have $0 < \lambda_{\min}(S, \alpha) \leq \lambda_{\max}(S, \alpha)$. Gershgorin circles and Proposition A.2.2

yield the asymptotic bounds,

$$\frac{g_0^2}{\lambda_{\max}(S,\alpha)} \min_{s\in\mathcal{D}}\{g_{\max}(s,S)^{2\alpha}(1+o(1))\} \leq L^{\{\alpha\}}(S) \leq \frac{g_0^2}{\lambda_{\min}(S,\alpha)} \min_{s\in\mathcal{D}}\{g_{\max}(s,S)^{2\alpha}(1+o(1))\}.$$

Consider, then, comparing an arbitrary sampling trajectory $S^* \in \Omega_{\mathrm{Rg}}$ against a global minimizer of $\mathcal{H}$ on Ω_{Rg}, say S_{mcc}. We can write,

$$\frac{L^{\{\alpha\}}(S^*)}{L^{\{\alpha\}}(S_{mcc})} \leq \frac{\frac{1}{\lambda_{\max}(S^*,\alpha)} \min_{s\in\mathcal{D}} \left\{g_{\max}(s,S^*)^{2\alpha}(1+o(1))\right\}}{\frac{1}{\lambda_{\min}(S_{mcc},\alpha)} \min_{s\in\mathcal{D}} \left\{g_{\max}(s,S_{mcc})^{2\alpha}(1+o(1))\right\}}. \tag{A.4}$$

Next we take a closer look at the eigenvalues. Note that the correlation matrix, $\mathbf{K}_{\mathcal{T}}^{\{\alpha\}}$ satisfies $\lim_{\alpha\to\infty} \mathbf{K}_{\mathcal{T}}^{\{\alpha\}} = \boldsymbol{I}_{nk_{\max}}$, and thus $\lambda_{\max}(S,\alpha)$ and $\lambda_{\min}(S,\alpha)$ tend to 1 for any sample trajectory $S \in \Omega_{\mathrm{Rg}}$. Finally, since S_{mcc} minimizes the maximum over s of the minimum over (i,k) of $\delta_k(s, {s_i}^{(k)}) = \phi(\|s - {s_i}^{(k)}\|) - w(k)$, it equivalently maximizes the minimum value of $g_{\max}(s,S)$. For any $S \in \Omega_{\mathrm{Rg}}$, $\min_{s\in\mathcal{D}}\{g_{\max}(s,S)^{2\alpha}\} \leq \min_{s\in\mathcal{D}}\{g_{\max}(s,S_{mcc})^{2\alpha}\}$. Thus the ratio (A.4) is bounded by $1+o(1)$. Therefore, in the limit as $\alpha \to \infty$, minimizing $\mathcal{M}^{\{\alpha\}}$ over Ω_{Rg} is equivalent to minimizing the maximum covariance disk-covering function, $\mathcal{H}$ on Ω_{Rg}. ■

Proofs and supporting results from Section 4.3

Proof: [Proof of Proposition 4.3.2] Let $(i,k) \in I_{\mathrm{samp}}$ and $s_* \in \mathcal{D}$ be such that $\mathcal{H}(S) = \delta_k(s_*, {s_i}^{(k)})$. By definition, given a partition $\mathcal{W} = \{{W_1}^{(1)}, \ldots, {W_n}^{(k_{\max})}\}$ of $\mathcal{D}$, there exists a pair, $(j,l) \in I_{\mathrm{samp}}$, such that $s_* \in {W_j}^{(l)}$. The definition of $\mathcal{MC}$ and the assumption that $\mathrm{I}(\mathcal{W}) \leq \mathrm{I}(\mathcal{MC}(S))$ leads to the implication chain, $\mathcal{H}(S) = \delta_k(s_*, {s_i}^{(k)}) \leq \delta_l(s_*, {s_j}^{(l)}) \leq \max_{s\in {W_j}^{(l)}} \delta_l(s, {s_j}^{(l)}) \leq \mathcal{H}_{\mathcal{W}}(S)$. ■

Proofs and supporting results from Section 4.4

Proof: [Proof of Lemma 4.4.1] For $c \leq w(k)$, we have $\Omega_{\mathrm{sublvl}}(\mathrm{MCD}_i{}^{(k)}, c) = \emptyset$. Otherwise, it is the intersection of an infinite set of closed d-spheres, which is a strictly convex set. ■

Proof: [Proof of Proposition 4.4.3] First, note that $\mathrm{MCD}_i{}^{(k)}$ and the map $s \mapsto \mathrm{d}_{\max}(s, W_i{}^{(k)})$ have the same extrema. In [18] it is shown that the latter function has a unique global minimum at $\mathrm{CC}(W_i{}^{(k)})$, when $W_i{}^{(k)}$ is taken to be a convex polygon. Identical reasoning yields the same result for any closed, bounded and nonempty $W_i{}^{(k)}$. Thus $\mathrm{CC}(W_i{}^{(k)})$ is a global minimum of $\mathrm{MCD}_i{}^{(k)}$. The requirement that $\phi'(d) > 0$ for all $d > 0$ suffices to ensure that $\mathrm{MCD}_i{}^{(k)}$ does not have any critical points which are not critical points of the Euclidean maximum distance function. Since that function has no critical points other than $\mathrm{CC}(W_i{}^{(k)})$, the result follows. ■

Proof: [Proof of Proposition 4.4.6] For each $(i, k) \in I_{\mathrm{samp}}$ with $W_i{}^{(k)} \neq \emptyset$, we can write,

$$\max_{s \in W_i{}^{(k)}} \big\{\delta_k\big(s, \overline{\mathrm{CC}}(W_i{}^{(k)}, \tilde{s}_i{}^{(k)})\big)\big\} = \phi\big(\max_{s \in W_i{}^{(k)}} \|s - \overline{\mathrm{CC}}(W_i{}^{(k)}, \tilde{s}_i{}^{(k)}))\|\big) + w(k) \leq$$

$$\leq \phi(\max_{s \in W_i{}^{(k)}} \|s - s_i{}^{(k)}\|) + w(k) = \max_{s \in W_i{}^{(k)}} \big\{\delta_k(s, s_i{}^{(k)})\big\}.$$

Taking the maximum over all nodes implies (4.11). ■

Proofs and supporting results from Section 4.5.1

We begin with this supporting result on strictly convex sets.

Lemma A.2.3 (Strict convexity) *Let $G \subset \mathbb{R}^d$ be closed, bounded, and strictly convex. For any $s_1, s_2 \in G$ and $v \in N_G(s_2) \setminus \{\mathbf{0}\}$, $v^T \mathrm{vrs}(s_1 - s_2) < 0$. Equivalently, $\mathrm{vrs}(s_1 - s_2) \in \mathrm{int}(T_G(s_2))$.*

Proof: Since $s_1, s_2 \in G$, $\mathrm{vrs}(s_1 - s_2) \in T_G(s_2)$, and because of the strict convexity of G, we can choose an $\epsilon \in \mathbb{R}_{>0}$ small enough that $w \in T_G(s_2)$ for all $w \in \overline{B}(\mathrm{vrs}(s_1 - s_2), \epsilon)$. Since $v \in N_G(s_2)$ and $\mathrm{vrs}(s_1 - s_2) \in T_G(s_2)$, we have $v^T \mathrm{vrs}(s_1 - s_2) \leq 0$. But we know that $v^T \mathrm{vrs}(s_1 - s_2) \neq 0$ because if it did then there would be some $w \in \overline{B}(\mathrm{vrs}(s_1 - s_2), \epsilon)$ such that $v^T w > 0$, which violates the assertion that $w \in T_G(s_2)$. The result follows. ■

Proof: [Proof of Proposition 4.5.1] Necessity is a result of [15, Corollary to Proposition 2.4.3]. To show sufficiency, assume that $\mathbf{0} \in \partial \mathrm{MCD}_i{}^{(k)}(s^*) + N_\Gamma(s^*)$, and we consider two cases. If $\mathrm{CC}({W_i}^{(k)}) \in \Gamma$, the result follows by Proposition 4.4.3. We proceed by contradiction. Assume that $s^* \neq \mathrm{CC}({W_i}^{(k)})$, and $\mathbf{0} \in \partial {\mathrm{MCD}_i}^{(k)}(s^*) + N_\Gamma(s^*)$, but s^* is not a unique minimizer. Then $\exists s^\dagger \in \Gamma$ such that ${\mathrm{MCD}_i}^{(k)}(s^\dagger) \leq {\mathrm{MCD}_i}^{(k)}(s^*)$. By Proposition 4.4.3, s^* is not a critical point of ${\mathrm{MCD}_i}^{(k)}$. It follows that there is at least one nonzero vector, $v_G \in \partial {\mathrm{MCD}_i}^{(k)}(s^*)$ with $-v_G \in N_\Gamma(s^*)$, which implies $v_G^T \operatorname{vrs}(s^\dagger - s^*) \geq 0$. We know that $s^\dagger \in \Omega_{\mathrm{sublvl}}({\mathrm{MCD}_i}^{(k)}, {\mathrm{MCD}_i}^{(k)}(s^*))$, and by Lemma 4.4.1, $\Omega_{\mathrm{sublvl}}({\mathrm{MCD}_i}^{(k)}, {\mathrm{MCD}_i}^{(k)}(s^*))$ is strictly convex. By [15, Theorem 2.4.7 Corollary 1], $v_G \in N_{\Omega_{\mathrm{sublvl}}({\mathrm{MCD}_i}^{(k)}, {\mathrm{MCD}_i}^{(k)}(s^*))}(s^*)$. Lemma A.2.3 yields, $v_G^T \operatorname{vrs}(s^\dagger - s^*) < 0$, a contradiction. Therefore s^* is the unique global minimizer of ${\mathrm{MCD}_i}^{(k)}$ over Γ. ■

We will need this supporting result on the circumcenter of the extended set.

Lemma A.2.4 **($s = \mathrm{CC}(\widetilde{W_i}^{(k)}(S_i))$ implies $s \in \Gamma^{(k)}(S_i)$)** *Assume that ${W_i}^{(k)} \neq \emptyset$. Let $S_i \in \mathcal{D}^{k_{max}}$ such that $\Gamma^{(k)}(S_i) \neq \emptyset$. If ${s_i}^{(k)} = \mathrm{CC}\left(\widetilde{W_i}^{(k)}(S_i)\right)$ then ${s_i}^{(k)} \in \Gamma^{(k)}(S_i) \bigcap \mathcal{D}$.*

Proof: Assume that ${s_i}^{(k)} = \mathrm{CC}\left(\widetilde{W_i}^{(k)}(S_i)\right)$. Equation (4.13) and the fact that ${s_i}^{(k)} \in \mathrm{co}\left(\widetilde{W_i}^{(k)}(S_i)\right)$ imply that ${s_i}^{(k)} \in \mathcal{D}$. That ${s_i}^{(k)} \in \Gamma^{(k)}(S_i)$ follows by contradiction from the fact that ${s_i}^{(k)} \notin \Gamma^{(k)}(S_i)$ implies that ${s_i}^{(k)} = \mathrm{CC}(\mathrm{co}\{S_{\mathrm{cs}}(k, S_i)\})$, and the fact that $\Gamma^{(k)}(S_i)$ is the nonempty intersection of d-spheres of equal radii centered at points in $S_{\mathrm{cs}}(k, S_i)$. ■

Proof: **(Proof of Proposition 4.5.2)** As a result of Lemma A.2.4, ${s_i}^{(k)} = \mathrm{CC}\left(\widetilde{W_i}^{(k)}(S_i)\right)$ implies that ${s_i}^{(k)} \in \Gamma^{(k)}(S_i)$. We may therefore assume ${s_i}^{(k)} \in \Gamma^{(k)}(S_i)$. Note that since ${s_i}^{(k)} \in \Gamma^{(k)}(S_i)$, we may write, $\mathrm{d}_{\max}({s_i}^{(k)}, \widetilde{W_i}^{(k)}(S_i)) = r_k(\mathcal{H}_{\mathcal{W}_i}(S_i)) = r_k\big({\mathrm{MCD}_i}^{(k)}({s_i}^{(k)})\big)$. If, in addition, $\mathrm{d}^{(k:k')}(S_i) = u_{\max}$ for some $k' \in K_{\mathrm{cs}}(k)$, then we also have, $r_k(\mathcal{H}_{\mathcal{W}_i}(S_i)) = \|{s_i}^{(k)} - \mathrm{EPt}^{(k:k')}(S_i)\|$. Let $\xi_{\mathrm{EPt}} \subset \mathbb{R}^d$, respectively $\xi_W \subset \mathbb{R}^d$ denote the sets of unit vectors pointing from ${s_i}^{(k)}$ to the extended constraint points at a distance of $r_k(\mathcal{H}_{\mathcal{W}_i}(S_i))$, respectively to the points in ${W_i}^{(k)}$ at a distance of $r_k(\mathcal{H}_{\mathcal{W}_i}(S_i))$, i.e.,

$$\begin{aligned}\xi_{\mathrm{EPt}} &= \big\{ \operatorname{vrs}({s_i}^{(k')} - {s_i}^{(k)}) \big| k' \in K_{\mathrm{cs}}(k) \text{ s.t. } \|{s_i}^{(k)} - \mathrm{EPt}^{(k:k')}(S_i)\| = r_k(\mathcal{H}_{\mathcal{W}_i}(S_i)) \big\} \\ \xi_W &= \left\{ \operatorname{vrs}(s - {s_i}^{(k)}) \mid s \in {W_i}^{(k)} \text{ s.t. } \|{s_i}^{(k)} - s\| = r_k(\mathcal{H}_{\mathcal{W}_i}(S_i)) \right\}.\end{aligned}$$

It can be deduced from Equation (4.13) that the set $\{\mathbf{0} \bigcup \xi_{\text{EPt}}\}$ spans $N_{\Gamma^{(k)}(S_i)}({s_i}^{(k)})$. By extension of Proposition 4.4.3, we may conclude that ${s_i}^{(k)} = \text{CC}\left(\widetilde{W_i}^{(k)}(S_i)\right)$ if and only if $\mathbf{0} \in \text{co}\{\xi_W \bigcup \xi_{\text{EPt}}\}$. It can be seen that $\mathbf{0} \in \text{co}\{\xi_W \bigcup \xi_{\text{EPt}}\}$ if and only if $\mathbf{0} \in \partial\text{MCD}_i{}^{(k)}({s_i}^{(k)}) + N_{\Gamma^{(k)}(S_i)}({s_i}^{(k)})$. By Proposition 4.5.1, we have our result. ■

Proofs and supporting results from Section 4.5.2

Proof: [Proof of Lemma 4.5.3] The result follows by simple contradiction from two observations for any $k' \in K_{\text{cs}}(k) \cap K_C$. First, if $\text{d}^{(k:k')} > u_{\max}$, then ${s_i}^{(k)} = \text{CC}(\widetilde{W_i}^{(k)}(S_i; K_C))$ implies that $K_{\text{cs}}(k) \cap K_C = \{k-1, k+1\}$ and ${s_i}^{(k)} = \frac{{s_i}^{(k-1)} + {s_i}^{(k+1)}}{2}$. Second, the first and last samples in the sequence must satisfy $\delta_k\left(\text{EPt}^{(k:k')}(S_i), {s_i}^{(k)}\right) \leq \text{MCD}_i{}^{(k)}(S_i)$. ■

Proof: [Proof of Lemma 4.5.4] From Equation (4.13), we can write,

$$\|\text{EPt}^{(k:k')}(S_i) - {s_i}^{(k)}\| = \frac{r_k(\mathcal{H}_{\mathcal{W}_i}(S_i))}{u_{\max}} \, \text{d}^{(k:k')}(S_i).$$

It has been established that $\mathcal{H}_{\mathcal{W}}$ is locally Lipschitz and regular, as is $\text{d}^{(k:k')}$. The gradient is derived from [15, Proposition 2.3.13] and a special case of [15, Theorem 2.3.9]. ■

The following result characterizes critical points of $\partial\text{CDE}_i{}^{(k:k')} \subset \partial\mathcal{H}_{\widetilde{W_i}}(S_i)$.

Corollary A.2.5 (Critical points of $\partial\text{CDE}_i{}^{(k:k')}(S_i)$) *Let $S_i \in \Omega_{\text{Rg}_i}$, and let $k, k' \in \{1, \dots, k_{max}\}$. If $\mathbf{0} \in \partial\text{CDE}_i{}^{(k:k')}(S_i) \subset \partial\mathcal{H}_{\widetilde{W_i}}(S_i)$ then* all *of the following hold,*

$$\mathcal{H}_{\mathcal{W}_i}(S_i) = \text{MCD}_i{}^{(k)}(S_i) = \text{MCD}_i{}^{(k')}(S_i) \tag{A.5a}$$

$$\mathbf{0} \in \text{co}\{\partial\text{MCD}_i{}^{(k)}({s_i}^{(k)}), \partial\text{MCD}_i{}^{(k')}({s_i}^{(k')})\} \tag{A.5b}$$

$${s_i}^{(k)} = \text{CC}\left(\widetilde{W_i}^{(k)}(S_i; \{k'\})\right). \tag{A.5c}$$

Proof: First, note that since $S_i \in \Omega_{\text{Rg}_i}$, we have $\partial\text{CDE}_i{}^{(k:k')}(S_i) \subset \partial\mathcal{H}_{\widetilde{W_i}}(S_i)$ if and only if $\text{d}^{(k:k')}(S_i) = u_{\max}$. From Lemma 4.5.4 it can be seen that $\partial\text{CDE}_i{}^{(k:k')}(S_i)$ is proportional to the sum of two vector sets, one of which consists of a single vector which is nonzero only in the kth and k'th components, and the other is $\partial\mathcal{H}_{\mathcal{W}_i}(S_i)$. Any vector in $\partial\mathcal{H}_{\mathcal{W}_i}(S_i)$ is zero everywhere except (possibly) the element corresponding to a single timestep. Thus $\mathbf{0} \in \partial\mathrm{CDE}_i{}^{(k:k')}(S_i)$ only if Equation (A.5a) holds. Solving the two simultaneous equations $\mathbf{0} \in \pi_k(\partial\mathrm{CDE}_i{}^{(k:k')})$ and $\mathbf{0} \in \pi_{k'}(\partial\mathrm{CDE}_i{}^{(k:k')})$ yields the other results. ■

Proof: **(Proof of Lemma 4.5.5)** Note that we may write,

$$\mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i) = \max\left\{\mathrm{MCD}_i{}^{(k)}(S_i), \mathrm{CDE}_{\max}{}^{(k)}(S_i)\right\}.$$

By Lemmas 4.5.4 and 4.4.2, $\mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i)$ can be seen to be the maximum of locally Lipschitz and regular functions, and therefore locally Lipschitz and regular itself. The form of the gradient follows from application of [15, Proposition 2.3.12]. ■

Proof: [Proof of Proposition 4.5.6] The $\mathrm{MCD}_{\widetilde{W}}{}^{(k)}$ is locally Lipschitz and regular for all $k \in \mathrm{argmax}_{k\in\{1,\dots,k_{\max}\}} \mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i)$. Since $\mathcal{H}_{\widetilde{W}_i}$ is the maximum of locally Lipschitz and regular functions, it is locally Lipschitz and regular itself. The form of the gradient follows from application of [15, Proposition 2.3.12]. ■

Proof: [Proof of Lemma 4.5.7] For any $k \in \{1,\dots,k_{\max}\}$ and $k' \in K_{\mathrm{cs}}(k)$, $S_i \in \Omega_{\mathrm{Rg}}$ implies that $\mathrm{CDE}_i{}^{(k:k')}(S_i) \leq \mathcal{H}_{\mathcal{W}_i}(S_i)$. By definition, we also have $\mathrm{MCD}_i{}^{(k)}(S_i) \leq \mathcal{H}_{\mathcal{W}_i}(S_i)$, with equality for at least one k. We may then write,

$$\mathcal{H}_{\widetilde{W}_i}(S_i) = \max_{k\in\{1,\dots,k_{\max}\}} \mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i) = \mathcal{H}_{\mathcal{W}_i}(S_i).$$

■

Proof: [Proof of Lemma 4.5.8] First, note that since $S_i \in \Omega_{\mathrm{Rg}_i}$, for any $k \in K_C$ and $k' \in K_{\mathrm{cs}}(k)$, we have $\mathrm{CDE}_i{}^{(k:k')}(S_i) \leq \mathcal{H}_{\mathcal{W}_i}(S_i)$, with equality if and only if $\mathrm{d}^{\,(k:k')}(S_i) = u_{\max}$. If this condition is not met, then sample $s_i{}^{(k')}$ is not active in the centering of $s_i{}^{(k)}$. Furthermore, if $\mathrm{MCD}_{\widetilde{W}}{}^{(k')}(S_i) < \mathcal{H}_{\mathcal{W}_i}(S_i)$, then $\mathrm{CDE}_i{}^{(k':k)}(S_i) < \mathcal{H}_{\mathcal{W}_i}(S_i)$. Thus any sample which does not have maximal distance to its *extended* set can not be active in the centering of a sample which does. If k is maximal, and k' is not, then the sub-sequence which includes k but not k' is also centered. Thus a maximally centered sequence may be constructed around any maximal sample in K_C. ■

Proposition A.2.6 (Maximally centered trajectories are optimal) *Let $\mathcal{W}_i \subset$*

$\mathfrak{P}(\mathcal{D})$ and $S_i \in \Omega_{\mathrm{Rg}_i}$ such that the entire sequence, S_i is maximally centered. Then S_i is the unique strict global minimizer of $\mathcal{H}_{\widetilde{W}_i}$ over Ω_{Rg_i}.

Proof: Let $\tilde{S}_i = (\tilde{s}_i{}^{(1)}, \dots, \tilde{s}_i{}^{(k_{\max})})^T \in \Omega_{\mathrm{Rg}_i}$ such that $\mathcal{H}_{\widetilde{W}_i}(\tilde{S}_i) \le \mathcal{H}_{\widetilde{W}_i}(S_i)$. By Lemma 4.4.1, the set $G_{\mathrm{Sub}}{}^{(k)} = \Omega_{\mathrm{sublvl}}(\mathrm{MCD}_i{}^{(k)}, \mathcal{H}_{\widetilde{W}_i}(S_i))$ is convex for any $k \in \{1, \dots, k_{\max}\}$. Let $G_{\mathrm{CSub}}{}^{(0)} = \{p_i(0)\}$, and let

$$G_{\mathrm{CSub}}{}^{(k)} = \left\{ s \in \mathbb{R}^d \mid \exists k' \in K_{\mathrm{cs}}(k),\ s' \in G_{\mathrm{Sub}}{}^{(k')} \text{ with } \|s - s'\| \le u_{\max} \right\},$$

also a convex set. Since $\tilde{S}_i \in \Omega_{\mathrm{Rg}_i}$, $\tilde{s}_i{}^{(k)} \in G_{\mathrm{CSub}}{}^{(k)}$ for each $k \in \{1, \dots, k_{\max}\}$, and since $\mathcal{H}_{\widetilde{W}_i}(\tilde{S}_i) \le \mathcal{H}_{\widetilde{W}_i}(S_i)$, $\tilde{s}_i{}^{(k)} \in G_{\mathrm{Sub}}{}^{(k)}$. Making use of the similarity between the extended set formulation and the Lagrangian of the constrained one-center problem, it can be shown that $G_{\mathrm{ESub}}{}^{(k)} \cap G_{\mathrm{Sub}}{}^{(k)} = \{s_i{}^{(k)}\}$. Thus $\tilde{S}_i = S_i$ is the unique global minimum of $\mathcal{H}_{\widetilde{W}_i}$ over Ω_{Rg_i}. ■

Proof: [Proof of Proposition 4.5.9] We begin with the critical point result. We consider three separate cases inspired by Lemma 4.5.5 and Proposition 4.5.6. First, if there is a $k \in \{1, \dots, k_{\max}\}$ with $\mathbf{0} \in \partial \mathrm{MCD}_i{}^{(k)}(S_i) \subset \partial \mathcal{H}_{\widetilde{W}_i}(S_i)$, then $\{k\}$ defines a maximally centered sequence in S_i.

Second, assume that $\mathbf{0} \notin \mathcal{H}_{\mathcal{W}_i}(S_i)$, but that $\exists k \in \underset{k' \in \{1, \dots, k_{\max}\}}{\operatorname{argmax}} \mathrm{MCD}_{\widetilde{W}}{}^{(k')}(S_i)$ with $\mathbf{0} \in \partial \mathrm{CDE}_{\max}{}^{(k)}(S_i)$. From Corollary A.2.5, it can be deduced that $\exists k' \in \operatorname{argmax}_{l \in K_{\mathrm{cs}}(k)} \mathrm{CDE}_i{}^{(k:l)}(S_i)$ such that $\{k, k'\}$ is a maximally centered sequence.

Finally, assume that $\mathbf{0} \notin \mathcal{H}_{\mathcal{W}_i}(S_i)$ and there is no k with $\mathbf{0} \in \partial \mathrm{CDE}_{\max}{}^{(k)}(S_i) \subset \partial \mathcal{H}_{\widetilde{W}_i}(S_i)$. With a slight abuse of notation, let $\mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i; K) = \underset{s \in \widetilde{W}_i{}^{(k)}(S_i;K)}{\max} \delta_k\big(s, s_i{}^{(k)}\big)$. In this case it can be shown that $\mathbf{0} \in \partial \mathcal{H}_{\widetilde{W}_i}(S_i)$ if and only if there is a sequence $K^* \subseteq \{1, \dots, k_{\max}\}$ of two or more consecutive samples which satisfies,

$$\mathbf{0} \in \mathrm{co} \left\{ \mathrm{MCD}_i{}^{(k)}(S_i) \mid k \in K^* \right\},$$

and for all $k \in K^*$, $\mathbf{0} \in \pi_k(\partial \mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i; K^*))$ and $\partial \mathrm{MCD}_{\widetilde{W}}{}^{(k)}(S_i; K^*) \subset \mathcal{H}_{\widetilde{W}_i}(S_i)$. It can be shown that the first two conditions are satisfied if and only if K^* defines a centered sequence, while the last requires that it be maximal.

This proves that S_i is a critical point if and only if it contains at least one maximally centered sequence. That any critical point is a global minimum follows by applying Proposition A.2.6 to any maximally centered sequence in S_i. ■

Proofs and supporting results from Section 4.6

Proof: [Proof of Proposition 4.6.1] We use the discrete-time LaSalle invariance principle [47] to show convergence. Let $\mathrm{T} : (\mathcal{D}^{k_{\max}})^n \to (\mathcal{D}^{k_{\max}})^n$ denote the evolution map of the Generalized Multicircumcenter Algorithm, i.e., $S^{\{j\}} = \mathrm{T}(S^{\{j-1\}})$. Note that Ω is positively invariant with respect to T, and that $\mathcal{H}$ is non-increasing along T on Ω. Since Ω is bounded, any evolution is bounded. The maps T and $\mathcal{H}$ are both continuous on Ω. By the discrete time LaSalle invariance principle, any evolution with initial condition $S^{\{0\}} \in \Omega$ must converge to M, the largest invariant set with respect to T contained in $z = \{S \in \Omega \mid \mathcal{H}(\mathrm{T}(S)) = \mathcal{H}(S)\} \subset \Omega$.

Now, let $M_{\min}$ denote the set of all global minimizers of $\mathcal{H}$ on Ω, and note that $M_{\min} \subseteq M$. We reason by contradiction to show that $M_{\min} = M$. Assume that there is a trajectory, $S^{\{0\}} \in M \setminus M_{\min}$. Since $M \subset z$, we have $\mathcal{H}(S^{\{1\}}) = \mathcal{H}(S^{\{0\}})$. Consider the fixed-partition optimization at step 0. Let $\mathcal{W} = \mathcal{MC}(S^{\{0\}})$, and let $i \in \operatorname{argmax}_{i' \in \{1,\dots,n\}} \mathcal{H}_{\mathcal{W}_i}(S_i^{\{0\}})$. Since $S^{\{0\}}$ is not a global minimizer of $\mathcal{H}$ over Ω, it is not a global minimizer of $\mathcal{H}_{\widetilde{\mathcal{W}}}$ over Ω, thus $S_i^{\{0\}}$ is not a global minimizer of $\mathcal{H}_{\widetilde{W_i}}$ over Ω_i. On the other hand, $S_i^{\{1\}}$ is a global minimizer of $\mathcal{H}_{\widetilde{W_i}}$, and we have $\mathcal{H}_{\widetilde{W_i}}(S_i^{\{1\}}) < \mathcal{H}_{\widetilde{W_i}}(S_i^{\{0\}})$. This is true for all such i, thus $\mathcal{H}_{\widetilde{\mathcal{W}}}(S^{\{1\}}) < \mathcal{H}_{\widetilde{\mathcal{W}}}(S^{\{0\}})$. By Lemma 4.5.10 and Proposition 4.3.2, we can write, $\mathcal{H}_{\mathcal{W}}(S^{\{0\}}) > \mathcal{H}_{\mathcal{W}}(S^{\{1\}}) \geq \mathcal{H}(S^{\{1\}})$. Thus $\mathcal{H}(S^{\{0\}}) > \mathcal{H}(S^{\{1\}})$, which contradicts the assumption that $S^{\{0\}} \in z$. Therefore $M_{\min} = M$, and the result follows. ■

A.3 Proofs and supporting results from Chapter 5

Proof: [Proof of Proposition 5.3.4] Each node sends a single message to each neighbor at each step, so the time complexity is bounded by the number of iterations to completion. The error at iteration t may be written $e_{\text{ave}}(t) = \|w_{\text{ave}}(t) - w\|$, where w is the m-vector whose elements are all $b^T b$, and $w_{\text{ave}}(t)$ is the vector of current approximate values. This may be bounded in terms of the initial error as $e_{\text{ave}}(t) \leq \left(1 - \frac{4}{m \operatorname{diam}_{\mathcal{Q}}}\right)^t e_{\text{ave}}(0)$, where we have used [59, Equation (6.10)] to lower bound the algebraic connectivity of $\mathcal{Q}$. Thus the number of steps required to guarantee error less than ϵ is bounded by, $t^*_{\text{ave}} \in O_\epsilon\left(-\log^{-1}\left(1 - \frac{4}{m \operatorname{diam}_{\mathcal{Q}}}\right)\right)$. Applying the bound on the growth of the network diameter yields $t^*_{\text{ave}} \in O_\epsilon\left(-\log^{-1}\left(1 - \frac{4}{m \sqrt[d]{m}}\right)\right)$. For large m, we can replace the logarithm with the series representation. The higher order terms drop off and we are left with $t^*_{\text{ave}} \in O_\epsilon\left(m \sqrt[d]{m}\right)$. At each step of the algorithm, each node stores a single value for each neighbor, and a constant number of other values. Thus the space requirement is bounded simply by $\deg_{\mathcal{Q}}$, which is in $O_\epsilon(1)$ by assumption. Communication complexity is bounded by a single message over each channel of the network at each iteration. The total number of such messages from each node is bounded by a constant, which gives us the final result. ■

Proof: [Proof of Proposition 5.3.5] First, note that for each $j \in \{1, \ldots, m\}$, N_j can calculate the row sums which correspond to samples within $V_j(Q)$ and subsequently their maximum, so calculating the maximum row sum is simply a matter of finding the largest value in the network. At each step of the algorithm, each node sends a single message to each neighbor. The algorithm is complete after a number of iterations equal to the diameter of the network. This proves the time and communication complexities under the regularity assumption on $\mathcal{Q}$. At each iteration, all nodes store a single value, the current maximum, which takes care of the space complexity result. ■

Proof: [Proof of Proposition 5.3.6] The first step of the JOR algorithm is to calculate the relaxation parameter. For correlation matrices, Appendix C describes a near optimal relaxation parameter in the sense of minimizing the completion time. Using two leader election algorithms, the network can calculate the maximum off-diagonal element, $\beta = \max_{i \neq j \in \{1,\ldots,n\}} [{\mathbf{K}_{\text{c}}}^{(k)}]_{ij}$ and the maximum off-diagonal row sum $\alpha = \max_{i \in \{1,\ldots,n\}} \sum_{j \neq 1}^{n} [{\mathbf{K}_{\text{c}}}^{(k)}]_{ij}$. The relaxation parameter is then given by, $h^* = \frac{2}{2+\alpha-\beta}$. The time complexity of the JOR algorithm can be broken down into the maximum number of messages any node sends over any channel times the number of iterations. The number of messages N_j will send is equal to the number of nonzero off-diagonal entries $[{\mathbf{K}_{\mathrm{c}}}^{(k)}]_{ii'}$, $i \neq i'$, where $s_i \in V_j(Q)$ and $s_{i'} \in V_{j'}(Q)$, with $j \neq j'$. By assumption, this number is bounded by N_{cor}. The error at iteration t of the JOR algorithm may be written $e_{\mathrm{JOR}}(t) = \|w_{\mathrm{JOR}}(t) - ({\mathbf{K}_{\mathrm{c}}}^{(k)})^{-1}b\|$, where $w_{\mathrm{JOR}}(t)$ is the vector of current approximate values. An upper bound on the error at step t may be obtained by [6] $e_{\mathrm{JOR}}(t) \leq \big(\mathrm{sprad}\big(\boldsymbol{I} - h{\mathbf{K}_{\mathrm{c}}}^{(k)}\big)\big)^t e_{\mathrm{JOR}}(0)$, where $\boldsymbol{I}$ is the ${n_{\mathrm{c}}}^{(k)} \times {n_{\mathrm{c}}}^{(k)}$ identity matrix. By Proposition C.0.8, we have the bounds, $0 \leq \mathrm{sprad}\,(\boldsymbol{I} - h\mathbf{K}) < 1 - h^*\varpi_\lambda$. The assumption of sparsity, and the fact that ${\mathbf{K}_{\mathrm{c}}}^{(k)}$ is a correlation matrix give us $0 \leq \alpha < N_{\mathrm{msg}}$, and $0 \leq \beta < 1$, which results in $1 - h^*\varpi_\lambda < 1 - \frac{2\varpi_\lambda}{2+N_{\mathrm{msg}}}$. Since both ϖ_λ and N_{msg} are positive, we have $1 - \frac{2\varpi_\lambda}{2+N_{\mathrm{msg}}} < 1$. Thus the number of iterations required to reach error ϵ is upper bounded by $t^* = (e(0) - e^*)\left(- \log^{-1}\left(1 - \frac{2\varpi_\lambda}{2+N_{\mathrm{msg}}}\right)\right)$ in $O_\epsilon(1)$. The time complexity is dominated by the time complexity of the leader election algorithm outlined in Proposition 5.3.5. For space complexity, we note that the maximum number of samples in a given Voronoi cell is bounded by ${n_{\mathrm{c}}}^{(k)}$, while the average number is $\frac{{n_{\mathrm{c}}}^{(k)}}{m}$. The space complexity is dominated by the requirement to store vectors of length given by the number of samples in the cell, and the same number of rows of ${\mathbf{K}_{\mathrm{c}}}^{(k)}$. This yields the given result. For communication complexity, the overall algorithm requires a maximum of one message to be sent per nonzero off-diagonal entry in ${\mathbf{K}_{\mathrm{c}}}^{(k)}$, each iteration, plus the number of messages required for the leader election. ■

Proof: [Proof of Proposition 5.3.9 The time and communication complexities of this method are dominated by the requirement of an all-to-all broadcast. For the flooding method, complexities are given in [76] in terms of the quantities m, $\mathrm{Ed}_{\mathcal{Q}}$, and the total number of initial messages to be disseminated, M. This last quantity in our case is n times the number of bits required to convey a spatial location plus the number required to convey a sample value. Since the last two are constant, we have $M \in \Theta(n)$. The time complexity is given as $M + m - 2$ and the communication complexity is between $M\,\mathrm{Ed}_{\mathcal{Q}}$ and $2M\,\mathrm{Ed}_{\mathcal{Q}}$. The results for time and communication complexity follow from the assumptions on $\mathcal{Q}$. The space complexity of the algorithm is dominated by the requirement to store the entire inverse correlation matrix at each node. ■

A.4 Proofs and supporting results from Chapter 6

Proof: [Proof of Proposition 6.2.2] First, note that since $\mathbf{K}$ is positive definite, it admits a matrix logarithm. The Gershgorin Circle Theorem yields the implication chain

$$P \in \mathcal{T}^{(k)} \implies \lambda_{\max}(\mathbf{K}) < 2 \implies \lambda_{\max}(\mathbf{K} - \boldsymbol{I}) < 1.$$

Since the matrix $\mathbf{K} - \boldsymbol{I}$ is nonnegative and Hermitian, the inequality (2.8) holds for all $P \in \mathcal{T}^{(k)}$, using the maximum eigenvalue which is a normalized, submultiplicative norm. Thus the series $\log(\mathbf{K})$ in (2.9) converges, and we write $\log\det(\mathbf{K}) = \operatorname{tr}(\log(\mathbf{K}))$. Writing the corresponding Taylor series,

$$\log\det(\mathbf{K}) = \operatorname{tr}\Big(\sum_{i=1}^{\infty} \frac{(-1)^{i+1}}{i}(\mathbf{K} - \boldsymbol{I})^i\Big),$$

we conclude the result. ■

Proof: [Proof of Proposition 6.2.4] The first term is a result of (2.10) and the product rule for matrices. The second term stems from (6.4). ■

Proof: [Proof of Lemma 6.3.1] Under the given assumptions, N_j knows $\mathrm{row}_i(\mathbf{K})$ and $\mathrm{col}_i(\mathbf{F})$ for each $i \in {R_{\mathrm{in}}}^{(1:k+1)}(j, P)$. Recall that $\Upsilon = \mathbf{K}_0^{-1} + \mathbf{F}\mathbf{K}^{-1}\mathbf{F}^T$. The result follows from using p iterations of the JOR algorithm to calculate $\mathbf{F}\mathbf{K}^{-1}$, and an additional p^2 consensus algorithms to calculate $\mathbf{F}\mathbf{K}^{-1}\mathbf{F}$. ■

Proof: [Proof of Proposition 6.3.2] By Lemma 6.3.1, the quantity Υ and thereby its determinant may be calculated by all nodes. From (6.4), we can see that the quantity $\operatorname{tr}\big((\mathbf{K} - \boldsymbol{I})^2\big)$ is a sum over quantities known to each node and may therefore be calculated by consensus. This gives us $\mathcal{E}^{(k)}(P)$.

For the gradient, once Υ is known to all nodes, the network must calculate the following for each $l \in \{1, \ldots, d\}$,

$$\mathrm{row}_i \left(\mathbf{K} - \boldsymbol{I}\right) \mathrm{col}_i \left(\nabla_{i:l}\mathbf{K}\right); \tag{A.6}$$

$$\mathbf{F}\mathbf{K}^{-1}\nabla_{i:l}\mathbf{F}^T; \tag{A.7}$$

$$\mathbf{F}\mathbf{K}^{-1}\nabla_{i:l}\mathbf{K}\mathbf{K}^{-1}\mathbf{F}^T. \tag{A.8}$$

Let $j_1 \in \{1, \ldots, m\}$ such that $i \in R_{\mathrm{in}}{}^{(1:k+1)}(j_1, P)$. Note that the partial derivatives $\nabla_{i:l}\mathbf{K}$ and $\nabla_{i:l}\mathbf{F}^T$ for each $l \in \{1, \ldots, d\}$ may be calculated by N_{j_1}. Calculation of (A.6) follows directly. Since the only nonzero part of $\nabla_{i:l}\mathbf{F}^T$ is the $i + nk$th *row*, we can write,

$$\mathbf{F}\mathbf{K}^{-1}\nabla_{i:l}\mathbf{F}^T = \mathrm{col}_i \left(\mathbf{F}\mathbf{K}^{-1}\right) \mathrm{row}_i \left(\nabla_{i:l}\mathbf{F}^T\right) \in \mathbb{R}^{p \times p},$$

which is accessible to N_{j_1} via Lemma 6.3.1. This gives us (A.7). Lastly, the matrix $\nabla_{i:l}\mathbf{K}$ may be written as $\zeta + \zeta^T$, where ζ is nonzero only in the $i + nk$th *row*. This yields,

$$\begin{aligned}\mathbf{F}\mathbf{K}^{-1}\nabla_{i:l}\mathbf{K}\mathbf{K}^{-1}\mathbf{F}^T = \mathbf{F}\mathbf{K}^{-1}\zeta\mathbf{K}^{-1}\mathbf{F}^T + \\ + \left(\mathbf{F}\mathbf{K}^{-1}\zeta\mathbf{K}^{-1}\mathbf{F}^T\right)^T.\end{aligned}$$

Note that the quantity $\mathbf{F}\mathbf{K}^{-1}\zeta \in \mathbb{R}^{p \times n(k+1)}$ is nonzero only in the *columns*,

$$h \in \{1, \ldots, n(k+1)\},$$

corresponding to measurements with nonzero correlation to the space-time location $(p_i, k+1)$. This implies, by the first assumption of the proposition, that the spatial position corresponding to the hth measurement either lies within $V_{j_1}(Q)$, or within $V_{j_2}(Q)$ for some $j_2 \in \{1, \ldots, m\}$ such that N_{j_1} can communicate with N_{j_2}. In the former case, $\mathrm{row}_h(\mathbf{K}^{-1}\mathbf{F}^T)$ is known to N_{j_1}, while in the latter case it is known by N_{j_2}. Thus with communication from its neighbors, N_j can compute (A.8). ■

Appendix B

Predictions with a subset of measurements

We present here a series of results on the Kitanidis model concerning the relationship between subsets of sample locations and hypothetical predictions made from partial information. We present here a series of results concerning the relationship between subsets of spatiotemporal sample locations and hypothetical predictions made from partial information. Let $Y \in \mathbb{R}^n$ denote a full set of measurements at locations $X \in \mathcal{D}_e^n$. Let $n_1, n_2 \in \mathbb{Z}_{>0}$ such that $n_1 + n_2 = n$. Consider a partition of the measurements $Y = (Y_1, Y_2)$ such that $Y_1 \in \mathbb{R}^{n_1}$ and $Y_2 \in \mathbb{R}^{n_2}$, and a similar partition of X. Note that due to the invariance of φ and ϕ under permutations of the samples, the subsequent results do not require that the elements of the partition be sequential. We will use $\mathbf{K}_1$, respectively $\mathbf{K}_2$, to denote the correlation matrix of locations X_1, respectively X_2, and analogous notation for the matrices $\mathbf{F}_1, \mathbf{F}_2, \mathbf{E}_1, \mathbf{E}_2$. Let $\mathbf{K}_{12} = \mathbf{K}_{21}^T \in \mathbb{R}^{n_1 \times n_2}$ denote the matrix of cross-correlation between the two location vectors.

We begin with a multivariate version of the posterior predictive variance from Proposition 2.5.1, which can be considered the hypothetical distribution of the measurements at space-time locations X_2 given the samples Y_1. As in the univariate case, this result can be obtained by applying Bayes Theorem to the prior model.

Lemma B.0.1 (Multivariate posterior predictive distribution) *Under the prior assumptions in Equations (2.2) and (2.3), the multivariate posterior predictive distribution of hypothetical samples Y_2 conditional on data Y_1 is the n_2-variate shifted Students t distribution with $\nu + n_1$ degrees of freedom, which takes the form,*

$$p(Y_2|Y_1) \propto \det\left(\mathrm{Var}[Y_2|Y_1]\right)^{-\frac{1}{2}} \times \\ \left(1 + \frac{(Y_2 - \mathrm{E}[Y_2|Y_1])^T\, \mathrm{Var}[Y_2|Y_1]^{-1}\, (Y_2 - \mathrm{E}[Y_2|Y_1])}{\nu + n_1 - 2}\right)^{-\frac{\nu+n_1+n_2}{2}}.$$

Here, the expectation is given by

$$\mathrm{E}[Y_2|Y_1] = \xi_{2|1}^T(\mathbf{E}_1 + \mathbf{K}_0^{-1})^{-1}\left(\mathbf{F}_1\mathbf{K}_1^{-1}Y_1 + \mathbf{K}_0^{-1}\beta_0\right) + \mathbf{K}_{21}\mathbf{K}_1^{-1}Y_1,$$

where $\xi_{2|1} = \mathbf{F}_2 - \mathbf{F}_1\mathbf{K}_1^{-1}\mathbf{K}_{12}$. The covariance matrix is given by

$$\mathrm{Var}[Y_2|Y_1] = \varphi(Y_1, X_1)\phi(X_2; X_1),$$

where, with a slight abuse of notation, we have used $\phi(X_2; X_1)$ to denote the following multivariate extensions of ϕ and φ,

$$\phi(X_2; X_1) = \mathbf{K}_2 - \mathbf{K}_{21}\mathbf{K}_1^{-1}\mathbf{K}_{12} + \xi_{2|1}^T\left(\mathbf{K}_0^{-1} + \mathbf{E}_1\right)^{-1}\xi_{2|1},$$
$$\varphi(Y_1, X_1) = \frac{1}{\nu + n_1 - 2}\left(q\nu + \left(Y_1 - \mathbf{F}_1^T\beta_0\right)^T\left(\mathbf{K}_1 + \mathbf{F}^T\mathbf{K}_0\mathbf{F}\right)^{-1}\left(Y_1 - \mathbf{F}_1^T\beta_0\right)\right).$$

In the sequel, we will find useful the matrices $\mathcal{M} \in \mathbb{R}^{(n+p)\times(n+p)}$ and $M_2 \in \mathbb{R}^{(n_2+p)\times(n_2+p)}$ and vectors $\mathcal{U} \in \mathbb{R}^{n+p}$ and $U_2 \in \mathbb{R}^{n_2+p}$ defined as,

$$\mathcal{M} = \begin{bmatrix} \mathbf{K} & \mathbf{F}^T \\ \mathbf{F} & -\mathbf{K}_0^{-1} \end{bmatrix}, \quad \mathcal{U} = \begin{bmatrix} Y \\ -\mathbf{K}_0^{-1}\beta_0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} \mathbf{K}_2 & \mathbf{F}_2^T \\ \mathbf{F}_2 & -\mathbf{K}_0^{-1} \end{bmatrix}, \quad U_2 = \begin{bmatrix} Y_2 \\ -\mathbf{K}_0^{-1}\beta_0 \end{bmatrix}.$$

Note that M_2 is the lower right submatrix of $\mathcal{M}$ under a different partition, and U_2 is the corresponding subvector of $\mathcal{U}$. It can be shown that the matrices $\mathcal{M}$ and M_2 are invertible.

Proposition B.0.2 (Approximate conditional variance) *The term $\phi(x;X)$ may be written in terms of spatiotemporal locations X_2 as,*

$$\phi(x;X) = \phi(x;X_2) - (\mathbf{k}_1 - \mu_1)^T \phi(X_1;X_2)^{-1} (\mathbf{k}_1 - \mu_1), \textit{ where}$$

$$\mu_1 = \begin{bmatrix} \mathbf{K}_{21} \\ \mathbf{F}_1 \end{bmatrix}^T M_2^{-1} \begin{bmatrix} \mathbf{k}_2 \\ \mathbf{f}(x) \end{bmatrix}, \quad \mathbf{k}_1 = \mathrm{Cor}[Z(x), Y_1], \quad \mathbf{k}_2 = \mathrm{Cor}[Z(x), Y_2].$$

Therefore $\phi(x;X) \le \phi(x;X_2)$ with equality if and only if $\mathbf{k}_1 = \mu_1$.

Proof: First, we note that the conditional variance can be written using $\mathcal{M}$ as,

$$\phi(x;X) = \mathrm{Cor}[Z,Z] - \begin{bmatrix} \mathbf{k} \\ \mathbf{f}(x) \end{bmatrix}^T \mathcal{M}^{-1} \begin{bmatrix} \mathbf{k} \\ \mathbf{f}(x) \end{bmatrix}.$$

Next, we point out that with the proper partitioning of $\mathcal{M}$, the matrix $\phi(X_1;X_2)$ is the Schur Complement, $(M_2 \,|\mathcal{M})$. Using this, and a similar partition of the vector $\mathbf{k}$, one arrives at the result. ■

The following result illustrates a number of ways in which the sigma mean may be restated.

Lemma B.0.3 (Restated sigma mean) *The quadratic form $\mathcal{U}^T\mathcal{M}^{-1}\mathcal{U}$ admits the following representations,*

$$\mathcal{U}^T\mathcal{M}^{-1}\mathcal{U} = (Y - \mathbf{F}^T\beta_0)^T(\mathbf{K} + \mathbf{F}^T\mathbf{K}_0\mathbf{F})^{-1}(Y - \mathbf{F}^T\beta_0) - \beta_0^T\mathbf{K}_0^{-1}\beta_0 \tag{B.1a}$$

$$= Y^T\mathbf{K}^{-1}Y - \left(\mathbf{K}_0^{-1}\beta_0 + \mathbf{F}\mathbf{K}^{-1}Y\right)^T\left(\mathbf{K}_0^{-1} + \mathbf{E}\right)^{-1}\left(\mathbf{K}_0^{-1}\beta_0 + \mathbf{F}\mathbf{K}^{-1}Y\right) \tag{B.1b}$$

$$= U_2^T M_2^{-1} U_2 + (Y_1 - \mathrm{E}[Y_1|Y_2])^T\phi(X_1;X_2)^{-1}(Y_1 - \mathrm{E}[Y_1|Y_2]). \tag{B.1c}$$

Furthermore, the term $\varphi(Y,X)$ may be written as,

$$\varphi(Y,X) = \frac{q\nu + \beta_0^T\mathbf{K}_0^{-1}\beta_0 + \mathcal{U}^T\mathcal{M}^{-1}\mathcal{U}}{\nu + n - 2}. \tag{B.2}$$

Proof: Each of the three representations of the quadratic form may be derived directly by using [4, Proposition 2.8.7] to expand the inverse matrix onto Schur complements. Plugging representation (B.1a) into equation (B.2) yields the form given in Proposition 2.5.1. ■

The generalized least squares (GLS) approximation arises naturally from partitioning the elements of the term $\mathbf{K}^{-1}Y$. The following lemma gives explicit form in terms of the GLS error.

Lemma B.0.4 (Generalized least squares approximations) *Let $\hat{Y}_{LS} = \mathbf{K}_{21}\mathbf{K}_1^{-1}Y_1$ be the generalized least squares estimate of Y_2 based on samples Y_1 (conditional on all parameters), and let $\overline{y}_{LS} = Y_2 - \hat{Y}_{LS}$. Then we can write,*

$$\mathbf{K}^{-1}Y = \begin{bmatrix} \mathbf{K}_1^{-1}Y_1 \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{K}_1^{-1}\mathbf{K}_{12}(\mathbf{K}_1\,|\mathbf{K}\,)^{-1}\overline{y}_{LS} \\ (\mathbf{K}_1\,|\mathbf{K}\,)^{-1}\overline{y}_{LS} \end{bmatrix}. \tag{B.3}$$

Proof: [Proof of Proposition 5.2.4] If ${n_{\mathrm{c}}}^{(k+1)} = 0$, then $\hat{\varphi}^{(k+1)}(P) = \tilde{\varphi}k$. By Lemma 5.2.3, we also have $\tilde{\varphi}k + 1 = \tilde{\varphi}k$. For ${n_{\mathrm{c}}}^{(k+1)} > 0$, we use Equation (B.3) in Lemma B.0.4 to write,

$$\begin{aligned} Y^T\mathbf{K}^{-1}Y &= Y_1^T\mathbf{K}_1^{-1}Y_1 + (Y_2 - \hat{Y}_{\mathrm{LS}})^T(\mathbf{K}_1\,|\mathbf{K}\,)(Y_2 - \hat{Y}_{\mathrm{LS}}) \\ \mathbf{F}\mathbf{K}^{-1}Y &= \mathbf{F}_1^T\mathbf{K}_1^{-1}Y_1 + \xi_{2|1}(\mathbf{K}_1\,|\mathbf{K}\,)(Y_2 - \hat{Y}_{\mathrm{LS}}). \end{aligned}$$

Here we have used the simpler indexed notation, $Y_1 = {Y_\varphi}^{(k)}$ and $Y_2 = Y^{(k+1)}$ to simplify the algebra. Applying these results to Equation (B.1b) in Lemma B.0.3 yields,

$$\begin{aligned} \mathcal{U}^T\mathcal{M}^{-1}\mathcal{U} = Y_1^T\mathbf{K}_1^{-1}Y_1 - (\mathbf{K}_0^{-1}\beta_0 + \mathbf{F}_1\mathbf{K}_1^{-1}Y_1)^T(\mathbf{K}_0^{-1} + \mathbf{E})^{-1}(\mathbf{K}_0^{-1}\beta_0 + \mathbf{F}_1\mathbf{K}_1^{-1}Y_1) + \\ + (Y_2 - \hat{Y}_{\mathrm{LS}})^T(\mathbf{K}_1\,|\mathbf{K}\,)^{-1}(Y_2 - \hat{Y}_{\mathrm{LS}}) - \\ - 2(\mathbf{K}_0^{-1}\beta_0 + \mathbf{F}_1\mathbf{K}_1^{-1}Y_1)^T(\mathbf{K}_0^{-1} + \mathbf{E})^{-1}(\xi_{2|1}(\mathbf{K}_1\,|\mathbf{K}\,)^{-1}(Y_2 - \hat{Y}_{\mathrm{LS}})) - \\ - (\xi_{2|1}(\mathbf{K}_1\,|\mathbf{K}\,)^{-1}(Y_2 - \hat{Y}_{\mathrm{LS}}))^T(\mathbf{K}_0^{-1} + \mathbf{E})^{-1}(\xi_{2|1}(\mathbf{K}_1\,|\mathbf{K}\,)^{-1}(Y_2 - \hat{Y}_{\mathrm{LS}})). \end{aligned}$$

Using, e.g. [40, Equation (12,17)], we can write,

$$\phi(X_2; X_1) = (\mathbf{K}_1\,|\mathbf{K}\,)^{-1} + (\mathbf{K}_1\,|\mathbf{K}\,)^{-1}\xi_{2|1}^T(\mathbf{K}_0^{-1} + \mathbf{E})^{-1}\xi_{2|1}(\mathbf{K}_1\,|\mathbf{K}\,)^{-1}.$$

With some algebraic manipulation we arrive at the result,

$$\begin{aligned} \mathcal{U}^T\mathcal{M}^{-1}\mathcal{U} = Y_1^T\mathbf{K}_1^{-1}Y_1 - (\mathbf{K}_0^{-1}\beta_0 + \mathbf{F}_1\mathbf{K}_1^{-1}Y_1)^T(\mathbf{K}_0^{-1} + \mathbf{E})^{-1}(\mathbf{K}_0^{-1}\beta_0 + \mathbf{F}_1\mathbf{K}_1^{-1}Y_1) + \\ + (\overline{y}_{\mathrm{LS}} - 2\overline{\mu}_{2|1})\phi(X_2; X_1)^{-1}\overline{y}_{\mathrm{LS}}. \end{aligned}$$

The result follows from Lemma B.0.3. ■

Appendix C

Near optimal relaxation parameter for JOR

Here we present some results regarding a relaxation parameter for the JOR algorithm which is nearly optimal with respect to the rate of convergence of the algorithm for a certain class of matrices. Specifically we are interested in the class of symmetric, positive definite matrices C with ones on the diagonal. Let $y(t) = (y_1(t), \ldots, y_n(t))^T \in \mathbb{R}^n$ be the vector updated during the JOR iteration in (4.19). Let $e(t) = \|C^{-1}y - y(t)\|$ denote the error at iteration t. We may write,

$$e(t) \leq (\operatorname{sprad}(\boldsymbol{I} - hC))^t \, e(0), \tag{C.1}$$

giving a bound on the error at step t based on the initial error. The value of $\operatorname{sprad}(\boldsymbol{I} - hC)$ therefore controls the rate of convergence, and choosing the relaxation parameter, h, is of vital importance. Throughout this section we will use the shorthand $\lambda_{\max} = \lambda_{\max}(C)$ and $\lambda_{\min} = \lambda_{\min}(C)$. The work [78] provides results concerning the convergence of the JOR algorithm, including an optimal relaxation parameter, which in our case is equivalent to $h_{\text{opt}} = \frac{2}{\lambda_{\max}+\lambda_{\min}}$. In this section we will introduce an approximation to this optimal value which may be calculated in a distributed manner.

Proposition C.0.5 *Assume that $C \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix with all diagonal entries equal to 1. Let β and α denote the maximum off-diagonal* *entry of C and the maximum off-diagonal row sum of C, respectively,*

$$\beta = \max_{i \neq j \in \{1,\ldots,n\}} \{c_{ij}\} \geq 0, \qquad \alpha = \max_{i \in \{1,\ldots,n\}} \Big\{ \sum_{\substack{j=1 \\ j \neq i}}^{n} c_{ij} \Big\} \geq 0.$$

Let $h^* = \frac{2}{2+\alpha-\beta}$. Then using h^* as the relaxation parameter in the JOR algorithm to solve $y = C^{-1}b$ results in guaranteed convergence.

Proof: Recall from Section 4.6.4 that convergence of the JOR algorithm is guaranteed as long as $h^* \in \left(0, \frac{2}{\lambda_{\max}}\right)$. This can also be seen from Equation C.1, since h outside of this range would yield $\text{sprad}(\boldsymbol{I} - hC) > 1$. Since C is symmetric positive definite with 1's on the diagonal, all off-diagonal entries must have magnitude strictly less than 1. Thus $1 - \beta > 0$. The Gershgorin circle theorem (e.g. [4, Fact 4.10.13]) tells us that $\lambda_{\max} \leq 1 + \alpha$. Together these two results yield $2 + \alpha - \beta > \lambda_{\max}$, which implies that $\frac{2}{2+\alpha-\beta} < \frac{2}{\lambda_{\max}}$, and the result follows. ■

Lemma C.0.6 *Under the assumptions of Proposition C.0.5, $h^* \leq \frac{1}{\lambda_{min}}$, with equality if and only if C is the $n \times n$ identity matrix.*

Proof: First, note the following implication chain,

$$\lambda_{\min} \leq 1 \implies 2\lambda_{\min} \leq 2 + \alpha - \beta \implies h^* \leq \frac{1}{\lambda_{\min}}.$$

Now, assume that $h^* = \frac{1}{\lambda_{\min}}$. This implies that $\lambda_{\min} = 1 + \alpha - \beta$, but $\lambda_{\min} \leq 1$, and $\alpha \geq \beta$. So we must have $\lambda_{\min} = 1$. Since the diagonal entries of C are all 1, the smallest eigenvalue can only be 1 if all off-diagonal entries are zero, i.e., if $C = \boldsymbol{I}_n$. ■

Lemma C.0.7 *Under the assumptions of Proposition C.0.5, $|1-h^*\lambda_{min}| \geq |1-h^*\lambda_{max}|$.*

Proof: Using Lemma C.0.6, we have $|1 - h^*\lambda_{\min}| = 1 - h^*\lambda_{\min}$. The result may then be shown by two separate cases. First, note that if $h^* \leq \frac{1}{\lambda_{\max}}$ then we have,

$$|1 - h^*\lambda_{\max}| = 1 - h^*\lambda_{\max} \leq |1 - h^*\lambda_{\min}|,$$

so the result holds in this case. For the second case, assume that $h^* > \frac{1}{\lambda_{\max}}$. Then $|1 - h^*\lambda_{\max}| = h^*\lambda_{\max} - 1$. The inclusion principle and the fact that C is positive definite give us the bounds $0 < \lambda_{\min} \leq 1 - \alpha$. Combined with the previously mentioned Gershgorin bound, $\lambda_{\max} \leq 1 + \alpha$, this allows us to write,

$$\begin{aligned}
\frac{\lambda_{\max} + \lambda_{\min}}{2 + \alpha - \beta} &\leq 1 \\
2\frac{\lambda_{\max} + \lambda_{\min}}{2 + \alpha - \beta} &\leq 2 \\
h^* \left(\lambda_{\max} + \lambda_{\min}\right) &\leq 2 \\
h^* \lambda_{\max} - 1 &\leq 1 - h^* \lambda_{\min}.
\end{aligned}$$

Thus in all cases, $|1 - h^* \lambda_{\max}| \leq |1 - h^* \lambda_{\min}|$. ■

Proposition C.0.8 *Under the assumptions of Proposition C.0.5, further assume that* $\lambda_{min} \geq \varpi_\lambda$ *for some* $\varpi_\lambda \in (0, 1)$*. Then* $0 \leq \operatorname{sprad}(\boldsymbol{I} - h^* C) < 1 - \frac{2\varpi_\lambda}{2+\alpha-\beta}$

Proof: First note that the spectral radius is given by

$$\operatorname{sprad}(\boldsymbol{I} - h^* C) = \max\left\{|1 - h^* \lambda_{\min}|, |1 - h^* \lambda_{\max}|\right\},$$

and is clearly nonnegative. From Lemma C.0.7, we have $\operatorname{sprad}(\boldsymbol{I} - h^* C) = |1 - h^* \lambda_{\min}|$. From Lemma C.0.6, we can infer $\operatorname{sprad}(\boldsymbol{I} - h^* C) = 1 - h^* \lambda_{\min}$. The upper bound follows by comparing $1 - h^* \lambda_{\min}$ against $1 - h^* \varpi_\lambda$. ■

Bibliography

[1] P. Abrahamsen. A review of Gaussian random fields and correlation functions. Technical Report 917, Norwegian Computing Center, Oslo, Norway, 1997. Electronically available at http://publications.nr.no/917_Rapport.pdf.

[2] P. K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Computing Surveys*, 30(4):412–458, 1998.

[3] O. Berke. On spatiotemporal prediction for on-line monitoring data. *Communications in Statistics - Theory and Methods*, 27(9):2343–2369, 1998.

[4] D. S. Bernstein. *Matrix Mathematics*. Princeton University Press, Princeton, NJ, 2005.

[5] D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21(2):174–184, 1976.

[6] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[7] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks*. Applied Mathematics Series. Princeton University Press, 2009. Electronically available at http://coordinationbook.info.

[8] J. Carle and J. F. Myoupo. Topological properties and optimal routing algorithms for three dimensional hexagonal grid networks. In *High performance computing in the Asia-Pacific region*, pages 116–121, 2000.

[9] W. F. Caselton and J. V. Zidek. Optimal monitoring network designs. *Statistics & Probability Letters*, 2(4):223 – 227, 1984.

[10] K. Chaloner and I. Verdinelli. Bayesian experimental design, a review. *Statistical Science*, 10(3):273–304, 1995.

[11] M. S. Chen, K. G. Shin, and D. D. Kandlur. Addressing, routing, and broadcasting in hexagonal mesh multiprocessors. *IEEE Transactions on Computers*, 39(1):10–18, jan 1990.

[12] J. Choi, J. Lee, and S. Oh. Biologically-inspired navigation strategies for swarm intelligence using spatial Gaussian processes. In *IFAC World Congress*, Seoul, Korea, July 2008.

[13] J. S. Clark and A. E. Gelfand. *Hierarchical modelling for the environmental sciences.* Oxford University Press, 2006.

[14] F. H. Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

[15] F. H. Clarke. *Optimization and Nonsmooth Analysis.* Canadian Mathematical Society Series of Monographs and Advanced Texts. Wiley, 1983.

[16] David A. Cohn. Neural network exploration using optimal experiment design. In *Neural Networks*, pages 679–686. Morgan Kaufmann, 1994.

[17] J. Cortés. Distributed Kriged Kalman filter for spatial estimation. *IEEE Transactions on Automatic Control*, 54(12):2816–2827, 2009.

[18] J. Cortés and F. Bullo. Coordination and geometric optimization via distributed dynamical systems. *SIAM Journal on Control and Optimization*, 44(5):1543–1574, 2005.

[19] N. A. C. Cressie. *Statistics for Spatial Data.* Wiley, New York, 1993. revised edition.

[20] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 2 edition, 2000.

[21] J. B. Delos. Semiclassical calculation of quantum mechanical wavefunctions. *Advances in Chemical Physics*, 65:161–241, 1986.

[22] M. A. Demetriou and I. I. Hussein. Estimation of spatially distributed processes using mobile spatially distributed sensor network. *SIAM Journal on Control and Optimization*, 48(1):266–291, 2009.

[23] V.F. Dem'yanov and V.N. Malozemov. *Introduction to Minimax*. Wiley, New York, 1974.

[24] P. Diaconis, S. Holmes, and R. Montgomery. Dynamical bias in the coin toss. *SIAM Review*, 49(2):211–235, 2007.

[25] Z. Drezner and H. W. Hamacher, editors. *Facility Location: Applications and Theory*. Springer, 2001.

[26] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*, volume 18 of *Mathematics and Its Applications*. Kluwer Academic Publishers, 1988.

[27] R. Furrer, M. G. Genton, and D. Nychka. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523, 2006.

[28] H. Gao, J. Wang, and P. Zhao. The updated kriging variance and optimal sample design. *Mathematical Geology*, 28(3):295–313, 1996.

[29] M. Gaudard, M. Karvson, E. Linder, and D. Sinha. Bayesian spatial prediction. *Environmental and Ecological Statistics*, 6:147–171, 1999.

[30] L. Gottschalk. Interpolation of runoff applying objective methods. *Stochastic Hydrology and Hydraulics*, 7:269–281, 1993.

[31] R. Graham and J. Cortés. Asymptotic optimality of multicenter Voronoi configurations for random field estimation. In *IEEE Conf. on Decision and Control*, pages 3127–3132, New Orleans, LA, 2007.

[32] R. Graham and J. Cortés. A cooperative deployment strategy for optimal sampling in spatiotemporal estimation. In *IEEE Conf. on Decision and Control*, pages 2432–2437, Cancun, Mexico, December 2008.

[33] R. Graham and J. Cortés. Asymptotic optimality of multicenter Voronoi configurations for random field estimation. *IEEE Transactions on Automatic Control*, 54(1):153–158, 2009.

[34] R. Graham and J. Cortés. Cooperative adaptive sampling of random fields with partially known covariance. *International Journal on Robust and Nonlinear Control*, 2009. Submitted.

[35] R. Graham and J. Cortés. Cooperative adaptive sampling via approximate entropy maximization. In *IEEE Conf. on Decision and Control*, pages 7055–7060, Shanghai, China, December 2009.

[36] R. Graham and J. Cortés. Distributed sampling of random fields with unknown covariance. In *American Control Conference*, pages 4543–4548, St. Louis, MI, 2009.

[37] R. Graham and J. Cortés. Adaptive information collection by robotic sensor networks for spatial estimation. *IEEE Transactions on Automatic Control*, 2010. Submitted.

[38] R. Graham and J. Cortés. Generalized multicircumcenter trajectories for optimal design under near-independance. In *IEEE Conf. on Decision and Control*, Atlanta, Georgia, December 2010. Submitted.

[39] R. B. Gramacy. *Bayesian Treed Gaussian Process Models.* PhD thesis, University of California, Santa Cruz, CA 95064, December 2005. Department of Applied Math & Statistics.

[40] H. V. Henderson and S. R. Searle. On deriving the inverse of a sum of matrices. *SIAM Review*, 23(1):53–60, 1981.

[41] G. M. Hoffmann and C. J. Tomlin. Mobile sensor network control using mutual information methods and particle filters. *IEEE Transactions on Automatic Control*, 55(1):32–47, 2010.

[42] M. E. Johnson, L. M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26:131–148, 1990.

[43] P. K. Kitanidis. Parameter uncertainty in estimation of spatial functions: Bayesian analysis. *Water Resources Research*, 22:449–507, 1986.

[44] C-W. Ko, J. Lee, and M. Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.

[45] A. Krause and C. Guestrin. Nonmyopic active learning of Gaussian processes: an exploration-exploitation approach. In *International Conference on Machine Learning*, Corvallis, OR, 2007.

[46] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[47] J. P. LaSalle. *The Stability and Control of Discrete Processes*, volume 62 of *Applied Mathematical Sciences.* Springer, 1986.

[48] N. D. Lee and J. V. Zidek. *Statistical Analysis of Environmental Space-Time Processes.* Springer Series in Statistics. Springer, New York, 2006.

[49] 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*, 95(1):48–74, 2007.

[50] T. Leonard and J. S. J. Hsu. *Bayesian Methods*, volume 1 of *Cambridge series in statistical and probabilistic mathematics*. Cambridge University Press, Cambridge, UK, 1999.

[51] E. P. Liski, N. K. Mandal, K. R. Shah, and B. K. Sinha. *Topics in Optimal Design*, volume 163 of *Lecture Notes in Statistics*. Springer, New York, 2002.

[52] K. M. Lynch, I. B. Schwartz, P. Yang, and R. A. Freeman. Decentralized environmental modeling by mobile sensor networks. *IEEE Transactions on Robotics*, 24(3):710–724, 2008.

[53] N. A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1997.

[54] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[55] K. V. Mardia, C. Goodall, E. J. Redfern, and F. J. Alonso. The Kriged Kalman filter. *Test*, 7(2):217–285, 1998. With discussion.

[56] K. V. Mardia and R. J. Marshall. Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1):135–146, 1984.

[57] S. Martínez. Distributed interpolation schemes for field estimation by mobile sensor networks. *IEEE Transactions on Control Systems Technology*, 18(2):491–500, 2010.

[58] S. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, 2007.

[59] B. Mohar. The Laplacian spectrum of graphs. In Y. Alavi, G. Chartrand, O. R. Oellermann, and A. J. Schwenk, editors, *Graph Theory, Combinatorics, and Applications*, volume 2, pages 871–898. Wiley, 1991.

[60] P. Müller, B. Sansó, and M. De Iorio. Optimal Bayesian design by inhomogeneous Markov chain simulation. *Journal of the American Statistical Association*, 99(467):788–798, 2004.

[61] M. F. Mysorewala. *Simultaneous robot localization and mapping of parameterized spatio-temporal fields using multi-scale adaptive sampling.* PhD thesis, University of Texas at Arlington, 2008.

[62] P. Ögren, E. Fiorelli, and N. E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, 2004.

[63] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* Wiley Series in Probability and Statistics. Wiley, 2 edition, 2000.

[64] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, 2004.

[65] D. Peleg. *Distributed Computing. A Locality-Sensitive Approach.* Monographs on Discrete Mathematics and Applications. SIAM, 2000.

[66] D. O. Popa, K. Sreenath, and F. L. Lewis. Robotic deployment for environmental sampling applications. In *International Conference on Control and Automation*, pages 197–202, Budapest, Hungary, June 2005.

[67] F. Pukelsheim. *Optimal Design of Experiments*, volume 50 of *Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 2006.

[68] C. P. Robert and G. Casella. *Monte Carlo statistical methods.* Springer texts in statistics. Springer, New York, 2004.

[69] Sambu Seo, Marko Wallat, Thore Graepel, and Klaus Obermayer. Gaussian process regression: Active data selection and test point rejection. In *In Proceedings of the International Joint Conference on Neural Networks (IJCNN*, pages 241–246. IEEE, 2000.

[70] A. Shapiro. On concepts of directional differentiability. *Journal of Optimization Theory & Applications*, 66(3):477–487, 1990.

[71] M. C. Shewry and H. P. Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, 1987.

[72] A. Singh, A. Krause, C Guestrin, and W. J. Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, 34:707–755, 2009.

[73] M. L. Stein. *Interpolation of Spatial Data. Some Theory for Kriging.* Springer Series in Statistics. Springer, New York, 1999.

[74] M. L. Stein. The screening effect in kriging. *The Annals of Statistics*, 30(1):298–323, February 2002.

[75] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

[76] D. M. Topkis. Concurrent broadcast for information dissemination. *IEEE Transactions on Software Engineering*, SE-11(10):1107–1112, oct 1985.

[77] D. Ucinski. *Optimal Measurement Methods for Distributed Parameter System Identification.* CRC Press, 2005.

[78] F. E. Udwadia. Some convergence results related to the JOR iterative method for symmetric, positive-definite matrices. *Applied Mathematics and Computation*, 47(1):37–45, 1992.

[79] H. Wackernagel. *Multivariate Geostatistics.* Springer, New York, 3rd edition, 2006.

[80] D. P. Wiens. Robustness in spatial studies I: minimax prediction and II: minimax design. *Environmetrics*, 16:191–203 and 205–217, 2005.

[81] S. S. Wilks. Certain generalizations in the analysis of variance. *Biometrika*, 24(3/4):471–494, 1932.

[82] J. S. Willcox, J. G. Bellingham, Y. Zhang, and A. B. Baggeroer. Performance metrics for oceanographic surveys with autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 26(4):711–725, 2001.

[83] F. Zhang, E. Fiorelli, and N. E. Leonard. Exploring scalar fields using multiple sensor platforms: tracking level curves. In *IEEE Conf. on Decision and Control*, pages 3579–3584, New Orleans, LA, December 2007.

[84] F. Zhang and N. E. Leonard. Cooperative filters and control for cooperative exploration. *IEEE Transactions on Automatic Control*, 55(3):650–663, March 2010.