UNIVERSITY OF CALIFORNIA
SANTA CRUZ

TRAJECTORY PLANNING FOR AUTONOMOUS VEHICLES FOR OPTIMAL EXPLORATION OF SPATIAL PROCESSES

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

DOCTOR OF PHILOSOPHY

in

APPLIED MATHEMATICS AND STATISTICS

by

Sisi Song

December 2019

The Dissertation of Sisi Song
is approved:

Professor Abel Rodriguez, Chair

Professor Qi Gong

Professor Herbert Lee

Quentin Williams
Acting Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

Trajectory Planning for Autonomous Vehicles for Optimal Exploration of Spatial Processes

by

Sisi Song

Autonomous vehicles are becoming the platform of choice for large-scale exploration of environmental processes, owing to their low cost and dependability of sensors. Standard trajectory planning methods often preplan a trajectory in advance, or are based on information criteria, and do not use the observations taken at earlier locations on the trajectory to decide where the vehicle should go next. In this dissertation, we propose a framework for real-time, adaptive generation of trajectories that are optimal with respect to some exploration goal, with a focus on smooth continuous trajectories for nonholonomic vehicles. We develop an algorithm that enables the vehicle to gather data, reconstruct the environmental process, and generate piecewise optimal trajectory segments that use the data collected at previously visited locations. Our approach is based on Gaussian process priors and Bayesian optimal design. Gaussian processes provide a method to fuse any prior knowledge of the environmental process with the collected data to obtain the most updated estimation of the process. Through Bayesian optimal design, we develop reward functions that explicitly reflect the operational goal and naturally address the exploration-exploitation trade-off in a principled way. We include a number of empirical evaluations of the methodology and show the advantages of our algorithm on different archetypes of spatial processes. Field tests of the planner on an autonomous ground vehicle demonstrate the practical usefulness of our algorithm. We then extend the framework to incorporate

supplementary information from different types of off-vehicle sensors. Three incorporation methods, depending on the nature of the supplementary source, are developed and tested, resulting in performance improvement, especially in the early stages of exploration. Finally, we turn to the case of multiple vehicles and develop an extension of our optimal trajectory algorithm that uses collective information to estimate the process and individual trajectory planning for each vehicle. As a result, computation time per vehicle does not increase as more vehicles are added, while the time required to achieve the exploration goal is reduced nearly linearly.

To Jacko, Samus, Flux, and Wally

Acknowledgments

I cannot thank enough my advisor Abel Rodriguez, whose expertise in seemingly everything gave me an opportunity for expertise at something. Without his guidance and patience throughout this journey, this dissertation would not be possible.

Many thanks to the rest of my committee, Professors Qi Gong and Herbie Lee, for providing the gentle and necessary feedback along the way.

To everyone in the Applied Math and Statistics departments, faculty and classmates, who filled my days in the office with motivation and comradery, thank you! I will always remember the conversations and company of Harleigh, Yifei, Annalisa, Yuan, Sara, and Yingchen.

To my collaborators in engineering, Mircea, Sebastian, and especially Gordon for his awesome skills in putting my algorithm on the roomba.

To the friends, family, and groups that made these years so enjoyable. I am grateful for time spent with fellow grad students in various departments, for the GSC facilitating fun events, for the community and service found in Bonny Doon Fire and Higher Grounds, for the encouragement of Tiffany, Aunt Kate, Eva, and Connor in thesis writing, and for many other people and memories.

And to Andrew, thank you for your unconditional support.

Chapter 1

Introduction

The aim of this dissertation is to develop a flexible framework for optimal trajectory planning that enables autonomous vehicles to generate real-time trajectories onboard the vehicle that adapt to information collected by the vehicle. Automated trajectory planning (i.e., the determination of the path and velocity of the vehicle so that it can accomplish its mission) is a critical but difficult task, particularly in the case of uncertain environments (where trajectories cannot be pre-planned in advance but need to be determined online) and nonholonomic vehicles (in which vehicle constraints can seriously limit the number of trajectories available) [42, 56].

Figure 1.1 shows the four standard platforms for unmanned vehicles. While unmanned can (and often) mean remote operation by humans, this dissertation addresses methods for fully autonomous trajectory generation onboard the unmanned vehicles, i.e. no human involvement in deciding where the vehicle travels to and how it gets there.

(a) UAV **(b)** UGV **(c)** USV **(d)** AUV

Figure 1.1: Autonomous/unmanned vehicles and manufacturers: (a) unmanned aerial vehicle, Dragon Eye by AeroVironment, (b) unmanned ground vehicle, Mars rover prototype by NASA, (c) unmanned surface vehicle, SeaWorker by ASV Global, and (d) autonomous underwater vehicle, Remus by Woods Hole Oceanographic Institution.

1.1 Traditional goals of trajectory planning

Trajectory planning for autonomous vehicles has received considerable attention in the engineering literature. Traditional applications of trajectory planning are motion planning (i.e., navigating the vehicle from a beginning to end configuration), traveling salesman-like problems, and target tracking or following. Algorithms have been extensively developed for the case when the planner has access to a complete and accurate model of the environment in which the vehicle operates [43, 3, 41]. Trajectory planning in uncertain environments has been discussed in [75, 61] for target following and in [83, 45, 47] for target tracking, where the target location is unknown and moves around a possibly cluttered environment. In the context of motion planning while avoiding obstacles, a survey of methods is presented in [31], while navigation in unknown or dynamic environments are discussed in [38, 29]. Often navigation problems include an optimality goal such as using minimal time or distance [69, 74].

1.2 Informative, explorative goals

More recently, attention has turned to the problem of informative path planning [15, 11, 52], where the goal is to maximize the amount of sensor information obtained about an underlying field of interest. Informative path planning is especially critical for autonomous vehicles involved in environmental sensing. Autonomous vehicles provide an advantageous platform for studying environmental processes such as ocean currents, soil nutrient levels, and atmospheric pollution, especially in toxic environments (such as hazardous gas plumes) or for tasks that involve long time horizons (such as exploring a large part of an ocean).

In this context, the adaptive design of optimal trajectories that dynamically incorporate information as it is collected is a fundamental task. More specifically, we are interested in planning optimal trajectories online for autonomous vehicles that investigate unexplored environmental processes using onboard sensors taking periodic point-referenced measurements. The planning algorithm uses the data collected during the mission and direct the vehicle to sampling locations that contain the most useful information to achieve the specific task at hand. The optimality of trajectory planning algorithms is necessarily defined in relation to a specific operational goal. In this thesis we focus on two common tasks that appear in studying environmental fields: optimal field reconstruction, and extrema identification.

1.2.1 Optimal reconstruction goal

Optimal reconstruction, i.e., estimating the process with minimal error, is an essential task upon which other goals can be built. Previous work related to optimal reconstruction attempts to either minimize the variance associated with the field reconstruction (G- or I-optimality) [82, 48] or to optimize objective

functions based on information theoretic concepts such as mutual information or conditional entropy (A- or D-optimality) [32, 39]. The former focuses on selecting sampling locations that minimize the uncertainty of the field estimates, while the latter focuses on reducing uncertainty of the reconstruction by selecting sampling locations that are maximally informative about the parameters of the model of the environmental process. A method based on Kalman filtering that minimizes the norm of the state covariance matrix is presented in [44]. A feature of all these approaches is that their optimal solutions only depend on the sampling locations and not on the actual measurements gathered. This allows for the path to be pre-computed, thereby avoiding the challenge involved in continuously updating the trajectory, but fails to fully utilize the information collected by the vehicle.

1.2.2 Extremum identification goal

Another important operational goal we consider in this thesis is extrema identification, i.e., locating the maximum or minimum of the environmental process (and reporting the associated extreme value). Locating extrema can be of interest on its own, or used as a proxy for other features of interest, such as the unknown source of a chemical plume.

The bulk of previous work on finding extrema of a field uses gradient-based methods similar to those used for numerical optimization. For example, [13] proposes that an underwater vehicle emulate the steps taken by numerical optimization methods. In the situation where there is a single source or the goal is just to find local maxima or minima, [49] and [19] prove local exponential convergence to the source by a vehicle with constant forward velocity and angular velocity tuned by a control law that allows the vehicle to estimate the gradient. Gradient-based approaches work well when there is a single extremum and the surface is convex.

However, for more general surfaces and multiple sources, the resulting trajectories are susceptible to being trapped in local extrema and plateaus. An alternative approach that uses biased random walks inspired by bacteria chemotaxis is presented by [24]. This method can handle multiple, time-varying sources and multimodal functions, but it converges slowly.

1.3 Our framework

In this thesis we introduce a novel method for online, adaptive, informative trajectory planning for autonomous vehicles subject to movement constraints, such as a minimum speed or a maximum turning rate. Like in [32] and [39], our approach is inspired by ideas taken from the Bayesian sequential experimental design literature [16]. However, unlike previous work, we focus on the design of full trajectories rather than on waypoint or sensor placement selection. The focus on trajectory planning rather than waypoint selection is particularly key in situations in which the vehicle's movement is subject to constraints, where waypoint-based methods can lead to infeasible trajectories.

In our approach, the unknown field over a given region of interest being studied is modeled using Gaussian process priors. Gaussian process models have a long history in the statistics and machine learning literatures as nonlinear regression tools [64, 5, 51, 21, 68]. In the engineering literature, Gaussian process models have been widely used to model unknown environments [15, 78, 72, 39, 52]. From these Gaussian process models, we construct objective functions that trade off exploration of the unknown field with exploitation of its perceived features, allowing us to handle multimodal or information-heterogenous fields. Adaptivity is achieved by recomputing the optimal trajectory after each batch of new observations is collected.

In the context of unmanned aerial vehicles, the literature on payload directed flight addresses problems similar to the ones that motivate our work. Payload directed flight focuses on effectively using payload sensor data to accomplish mission objectives [40]. Most of this literature uses image data to perform vision-based tracking or mapping [8, 46], so the approach we present here for optimal exploration using scalar measurement data complements existing work.

1.3.1 Multiple sources of information

The framework detailed in the next two chapters allows for incorporating data from supplementary, off-vehicle sources (such as other stationary or remote sensors, or from a computer model of the environmental phenomenon). In this thesis we investigate the statistical challenges associated with incorporating supplementary data into the control system.

Incorporating supplementary information for the navigation of autonomous vehicles has been previously discussed in the literature by several authors. For example, [84] discusses the use of a mobile boat along with a stationary sensor network consisting of buoys to monitor the temperature of a lake surface. Similarly, [71] discusses the use of underwater vehicles to track ocean processes using predictions from a regional ocean model, and [33] considers the use of satellite imagery providing remote sensing data for monitoring chlorophyll concentration. On a slightly different vein, [57] looks at the sensor fusion problem of combining measurements arriving at different times and at different scale lengths.

1.3.2 Multiple vehicles

The framework also allows for adding additional vehicles, thereby creating a multi-vehicle system for exploring the environmental process. Cooperation of

multiple autonomous vehicles is a large research area on its own and has many applications including flight formation, mapping, surveillance, search and rescue, and mobile sensor networks for environmental sampling [55]. Having more vehicles has demonstrated advantages; [70] shows that two aerial robots are comparable to a stationary network of thirty sensors for monitoring a radiation release. In the literature for identifying the maximum of the process, [17] involves multiple vehicles and decentralizes the computations, and shows that even with an algorithm that uses gradient climbing, the vehicles can split up into two groups and find both peaks when the true field is bimodal. In related work, [60] extends the gradient following techniques to groups of vehicles and also addresses optimal formations for the group. For reconstruction goals, [22] extends mutual information methods to multiple underwater vehicles and shows consensus in field estimation by four vehicles.

The notion of formations is prevalent in the coordination of multiple vehicles, ranging from applications in pollution source localization to navigation with obstacles [7, 23, 26]. While formations have advantages of having a protocol for collision avoidance and improved efficiency in cooperative task allocation, we feel that the resulting trajectories are too restrictive. Another class of methods that produce restricted movement of vehicles are those that involve Voronoi diagrams, which are common for mobile sensor networks and distributed computation over a network [20]. In this thesis, the extension to optimal trajectory planning for multiple vehicles performing environmental exploration allows for vehicles to travel independently of each other in the region of interest.

1.4 Organization

This following chapters of the dissertation are organized as follows. Chapter 2 provides an overview of the mathematical concepts used and develops our models for the environmental and planning processes. Chapter 3 applies and evaluates the models to trajectory optimization for a single vehicle. Chapter 4 develops and evaluates methods to incorporate off-vehicle sensor information into the established framework. Chapter 5 extends the trajectory planning method from one to several vehicles. Lastly Chapter 6 summarizes the contributions and provides directions for future work.

Chapter 2

Theoretical Foundations

2.1 Gaussian process model

This section describes our statistical model for the unknown spatial field $f : \mathcal{S} \subset \mathbb{R}^2 \to \mathbb{R}$ that is to be reconstructed from noisy, point-referenced observations $y_1, y_2, \ldots$ collected at locations $\mathbf{s}_1, \mathbf{s}_2, \ldots \in \mathcal{S}$. Because our methods rely on a Bayesian approach to learning and prediction, building such model requires the specification of a likelihood describing the distribution of $y_1, y_2, \ldots$ conditional on the unknown function f, a prior distribution for f that captures our assumptions about the environmental process before any data is collected, and hyperprior distributions on key parameters of the prior distribution for f. The resulting posterior distribution for f given the data, which is obtained by applying Bayes theorem, is one of the essential ingredients necessary to derive the objective functions that are optimized by our trajectory planning algorithm.

To derive the likelihood function, we assume that, given f, the observations collected by the vehicle are independent with additive noise,

$$y_i = y(\mathbf{s}_i) = f(\mathbf{s}_i) + \varepsilon_i, \tag{2.1}$$

where $\varepsilon_i \sim \mathsf{N}(0, \sigma^2)$ represents the measurement noise and is modeled as following a normal distribution. The noisy observations $\{y_1, y_2, \ldots\}$ of the field f, along with the locations $\{\mathbf{s}_1, \mathbf{s}_2, \ldots\}$ where they are taken, are referred to as the data.

For our prior on f we use a Gaussian process prior with mean function $m(\mathbf{x}; \boldsymbol{\varphi})$ and covariance function $\mathrm{Cov}\{f(\mathbf{x}), f(\mathbf{x}')\} = C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi})$, where $\mathbf{x}, \mathbf{x}' \in \mathcal{S}$, and $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ are vectors of hyperparameters associated with the mean and covariance functions of the Gaussian process. (Notation: $f \mid \boldsymbol{\varphi}, \boldsymbol{\psi} \sim \mathcal{GP}(m(\cdot\,; \boldsymbol{\varphi}), C(\cdot\,, \cdot\,; \boldsymbol{\psi}))$.) The Gaussian process prior implies that, for any finite set of locations $\mathbf{x}_1, \ldots, \mathbf{x}_m$ in $\mathcal{S}$,

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_m) \end{bmatrix} \Bigg| \, \boldsymbol{\varphi}, \boldsymbol{\psi} \sim \mathsf{N}\left(\begin{bmatrix} m(\mathbf{x}_1; \boldsymbol{\varphi}) \\ \vdots \\ m(\mathbf{x}_m; \boldsymbol{\varphi}) \end{bmatrix}, \begin{bmatrix} C(\mathbf{x}_1, \mathbf{x}_1; \boldsymbol{\psi}) & \cdots & C(\mathbf{x}_1, \mathbf{x}_m; \boldsymbol{\psi}) \\ \vdots & \ddots & \vdots \\ C(\mathbf{x}_m, \mathbf{x}_1; \boldsymbol{\psi}) & \cdots & C(\mathbf{x}_m, \mathbf{x}_m; \boldsymbol{\psi}) \end{bmatrix} \right).$$

In this dissertation, we assume that no prior information is available about the shape of f and therefore use a constant prior mean function $m(\mathbf{x}; \boldsymbol{\varphi}) = \varphi$, where φ is a scalar parameter to be estimated from the data. This assumption is not central to our approach; if prior information is available, for example from a computer model of the underlying phenomena, it can be incorporated by altering the form of m. Section 4.4 considers this exact situation.

2.1.1 Covariance function choice

The choice of the covariance function controls properties of the Gaussian process such as the degree of differentiability of its realizations [4]. A common choice is the exponential covariance function

$$C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}) = \tau^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|}{\lambda} \right), \tag{2.2}$$

where $\boldsymbol{\psi} = (\tau^2, \lambda)$ and $\|\mathbf{x} - \mathbf{x}'\|$ is the distance between $\mathbf{x}$ and $\mathbf{x}'$. The hyperparameter τ^2 is the variance of the process, i.e., $\tau^2 = \text{Var}\{f(\mathbf{x})\}$ for any $\mathbf{x} \in \mathcal{S}$, and λ is the common length scale of the process in all directions.

Realizations from a stochastic process with an exponential covariance function are continuous but non-differentiable. Although lack of differentiability might be appropriate in some applications, it is restrictive unless substantial prior information is available. Furthermore, a single length scale λ that applies in every direction implies that realizations of the Gaussian process are isotropic, which means that the covariance depends only on the distance, $C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}) = C(\|\mathbf{x} - \mathbf{x}'\|; \boldsymbol{\psi})$. The isotropy assumption has been shown to lead to poor results when the real f does not satisfy this property [15]. Hence, we recommend the use of generalized Matérn covariance functions that allow for geometric anisotropy

$$C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}) = \frac{\tau^2 \|\mathbf{x} - \mathbf{x}'\|_{\mathbf{A}}^{\nu}}{2^{\nu-1}\Gamma(\nu)} K_\nu \left(\|\mathbf{x} - \mathbf{x}'\|_{\mathbf{A}} \right), \tag{2.3}$$

where $\boldsymbol{\psi} = (\tau^2, \nu, \mathbf{A})$, $\Gamma(\cdot)$ is the gamma function, and $K_\nu(\cdot)$ is the modified Bessel function of the second kind with order ν. Geometric anisotropy is a form of anisotropy that can be reduced to isotropy by a linear transformation of the coordinates. That is, if C is an isotropic covariance function, $C(\|\mathbf{x} - \mathbf{x}'\|_{\mathbf{A}}; \boldsymbol{\psi})$ is geometric anisotropic, where $\mathbf{A}$ is a 2×2 positive definite matrix and $\|\mathbf{x} - \mathbf{x}'\|_{\mathbf{A}} = \sqrt{(\mathbf{x} - \mathbf{x}')^\top \mathbf{A} (\mathbf{x} - \mathbf{x}')}$. It is convenient to parametrize $\mathbf{A} = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^\top$, where $\mathbf{P}$ is a 2×2 rotation matrix of angle ϑ in the plane, and $\boldsymbol{\Lambda}$ is a diagonal matrix with positive elements λ_1 and λ_2,

$$\mathbf{P} = \begin{bmatrix} \cos\vartheta & -\sin\vartheta \\ \sin\vartheta & \cos\vartheta \end{bmatrix}, \qquad \boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}.$$

In the anisotropic Matérn covariance function (2.3), the smoothness parameter

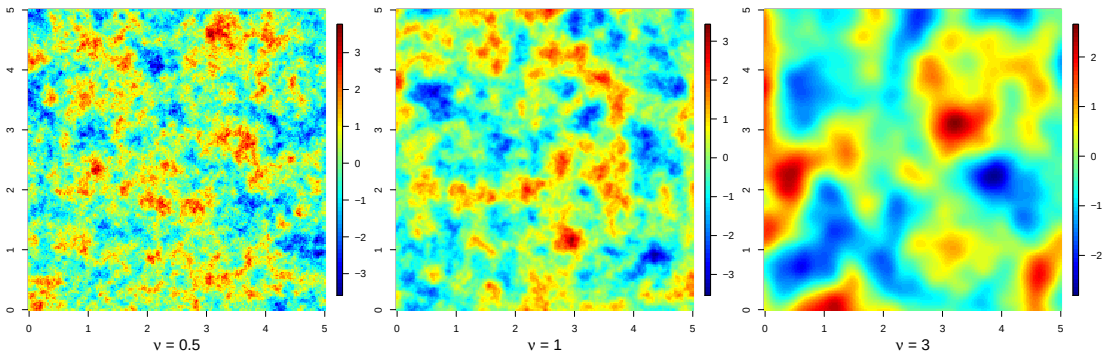


Figure 2.1: Simulated Gaussian processes using Matérn covariance functions with $\lambda_1 = \lambda_2$ and different smoothness parameters.

ν governs the differentiability of the realizations of the process, as illustrated in Figure 2.1, λ_1 and λ_2 are the length scales along the principal directions of the process, and the principal directions are given by the rotation of ϑ in $\mathbf{P}$. These parameters allow us to capture varying degrees of smoothness and geometric anisotropy in the environmental process, and as before, τ^2 is the variance of the process.

A third covariance function is the rational quadratic function extended to two dimensions

$$C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}) = \frac{\tau^2}{\left[1 + \left(\frac{x_1 - x_1'}{\lambda_1}\right)^2\right]\left[1 + \left(\frac{x_2 - x_2'}{\lambda_2}\right)^2\right]}, \tag{2.4}$$

where $\boldsymbol{\psi} = (\tau^2, \lambda_1, \lambda_2)$, $(x_1, x_2) = \mathbf{x}$, $(x_1', x_2') = \mathbf{x}'$, and $\tau^2, \lambda_1, \lambda_2$ are the variances and length scales. Despite having fewer hyperparameters, a major advantage of the rational quadratic is the computability of the integrals

$$\int\int C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}) \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{x}' \qquad \text{and} \qquad \int C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}) \mathrm{d}\mathbf{x} \tag{2.5}$$

in closed form (see Appendix). These integrals emerge as covariance values when f is integrated, which is required when aggregated observations are taken of f.

For example, in a satellite image of f, each pixel represents the average value of f over a small grid cell. This scenario is illustrated in detail in Section 4.3.

2.1.2 Estimation of the field

Gaussian process priors are particularly attractive because, conditional on the hyperparameters $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$, estimates of the field f at unobserved locations are straightforward to obtain. In our setting, denote $\mathbf{y}_{1:n} = (y_1, \ldots, y_n)$ as the observation history collected at locations $\mathbf{s}_1, \ldots, \mathbf{s}_n$. If f is modeled as a realization of a Gaussian process, the vector $\mathbf{f}_{1:n} = (f(\mathbf{s}_1), \ldots, f(\mathbf{s}_n))$, corresponding to the true values of the environmental process at the sampled locations, follows a normal distribution with mean $\mathbf{m}_{1:n} = (m(\mathbf{s}_1; \boldsymbol{\varphi}), \ldots, m(\mathbf{s}_n; \boldsymbol{\varphi}))$ and covariance matrix $\mathbf{C}_{1:n}$ with entries $[\mathbf{C}_{1:n}]_{ij} = C(\mathbf{s}_i, \mathbf{s}_j; \boldsymbol{\psi})$, for $i, j = 1, \ldots, n$. Given the assumption that observations are independent conditional on f, i.e., $\mathbf{y}_{1:n} \mid \mathbf{f}_{1:n} \sim \mathsf{N}(\mathbf{f}_{1:n}, \sigma^2 \mathbf{I}_n)$, then the marginal distribution of the observations is also a Gaussian distribution,

$$\mathbf{y}_{1:n} \mid \sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi} \sim \mathsf{N}\left(\mathbf{m}_{1:n}, \mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n\right), \tag{2.6}$$

and, for any arbitrary location $\mathbf{x} \in \mathcal{S}$, the posterior distribution of $f(\mathbf{x})$ is also Gaussian, $f(\mathbf{x}) \mid \mathbf{y}_{1:n}, \sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi} \sim \mathsf{N}\left(\mu_{\mathbf{y}_{1:n}}(\mathbf{x}), \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x})\right)$, where

$$\mu_{\mathbf{y}_{1:n}}(\mathbf{x}) = m(\mathbf{x}; \boldsymbol{\varphi}) + \mathbf{c}_{1:n}^{\mathsf{T}}(\mathbf{x}) \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n\right]^{-1} \left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right] \tag{2.7}$$

and

$$\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) = \tau^2 - \mathbf{c}_{1:n}^{\mathsf{T}}(\mathbf{x}) \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n\right]^{-1} \mathbf{c}_{1:n}(\mathbf{x}), \tag{2.8}$$

with $\mathbf{c}_{1:n}(\mathbf{x}) = (C(\mathbf{x}, \mathbf{s}_1; \boldsymbol{\psi}), \ldots, C(\mathbf{x}, \mathbf{s}_n; \boldsymbol{\psi}))$. Similarly, the marginal predictive distribution of a new observation $y(\mathbf{x})$ at an arbitrary location is

$$y(\mathbf{x}) \mid \mathbf{y}_{1:n}, \sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi} \sim \mathsf{N}\left(\mu_{\mathbf{y}_{1:n}}(\mathbf{x}), \sigma^2 + \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x})\right). \tag{2.9}$$

Equations (2.7) and (2.8) can be used to provide a reconstruction $\hat{f}_n$ of the true process f over $\mathcal{S}$ based on the n observations of $\mathbf{y}_{1:n}$. In particular, a point estimate at location $\mathbf{x} \in \mathcal{S}$ can be obtained from the mean of the posterior distribution $\hat{f}_n(\mathbf{x}) = \mathrm{E}\left\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}, \sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\right\} = \mu_{\mathbf{y}_{1:n}}(\mathbf{x})$, which is sometimes referred to as the kriging equation [73]. Similarly, $\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) = \mathrm{Var}\left\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}, \sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\right\}$ can be used to quantify the uncertainty of this estimate.

The quality of the reconstruction $\hat{f}_n$ depends on the fitness of the hyperparameters σ^2, $\boldsymbol{\varphi}$, and $\boldsymbol{\psi}$. Hence, rather than fixing their values, we treat them as unknown parameters and estimate them from the observed data using an empirical Bayes-like approach [54, 66],

$$\left(\widehat{\sigma}^2_n, \widehat{\boldsymbol{\varphi}}_n, \widehat{\boldsymbol{\psi}}_n\right) = \underset{(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi})}{\arg\max} \left[\log p\left(\mathbf{y}_{1:n} \mid \sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\right) + \log p\left(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\right)\right], \tag{2.10}$$

where $p\left(\mathbf{y}_{1:n} \mid \sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\right)$ is the marginal density of $\mathbf{y}_{1:n}$ implied by (2.6) and $p\left(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\right)$ is a prior distribution for the hyperparameters. Equation (2.10) can be maximized numerically using, for example, the Nelder-Mead algorithm [58]. With $\boldsymbol{\varphi} = \varphi$ constant, $\boldsymbol{\varphi}$ can be estimated analytically (see Appendix). We focus on this empirical Bayes approach rather than full Bayesian inference because a Markov Chain Monte Carlo algorithm would be slow, and we want to implement our algorithm onboard the vehicle in real-time.

2.2 Bayesian experimental design

In the trajectory planning setting, experimental design refers to trajectory design – determining the sampling locations for the autonomous vehicle in order to learn some relationship between the location and the environmental field. Bayesian approaches to experimental design involve the maximization of an expected utility function,

$$\tilde{U}(\mathbf{s}) = \int U(\mathbf{s}, f)\, p(f \mid \text{data})\, \mathrm{d}f, \tag{2.11}$$

where the data, consisting of the noisy measurements $\mathbf{y}_{1:n}$ and their locations, provide information about an unknown parameter f (which in our case corresponds to the environmental field under study), and $U(\mathbf{s}, f)$ is a utility function that depends on the unknown f and the design parameter $\mathbf{s}$ (which in our case corresponds to a location, or series of locations, where future measurements will be collected). $\tilde{U}(\mathbf{s})$ is the expectation of $U(\mathbf{s}, f)$ over the posterior distribution of f, $p(f \mid \text{data})$.

Bayesian design methods [16] have important advantages over classical statistical tools. For example, Bayesian approaches to experimental design naturally deal with the exploration-exploitation trade-off [80]. Furthermore, under mild conditions, Bayesian experimental design procedures are not influenced by the stopping rule used [10], so they can be applied in a sequential manner without the need for ad hoc adjustments. In the next subsections we discuss two expected utility functions that are the basis from which we construct the objective functions we use in our trajectory planning algorithm.

2.2.1 Optimal reconstruction of f

The current data provided by observations $\mathbf{y}_{1:n}$ at $\mathbf{s}_1, \ldots, \mathbf{s}_n$ offer a reconstruction of f through the point estimate $\hat{f}_n$. Suppose that our goal is to collect additional observations that allow us to make this estimate as close as possible to the true field f. An appropriate utility function evaluated at a new data point $\mathbf{s}$ and associated measurement $y(\mathbf{s})$ would aim to reduce the (squared) prediction error of the resulting point estimate $\hat{f}_{n+1}$. As the prediction error based on the current data is completely determined by (2.7), maximizing the reduction in prediction error is reduced to maximizing $-[f(\mathbf{x}) - \mu_{\mathbf{y}_{1:n},y(\mathbf{s})}(\mathbf{x})]^2$, where $\mu_{\mathbf{y}_{1:n},y(\mathbf{s})}(\mathbf{x}) = \mathrm{E}\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}, y(\mathbf{s}), \sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\}$ is the prediction at $\mathbf{x}$ based on the existing observations $\mathbf{y}_{1:n}$ and new observation $y(\mathbf{s})$. Averaging over all $\mathbf{x} \in \mathcal{S}$, the utility function is

$$U(\mathbf{s}, f) = -\int_{\mathcal{S}} \left[f(\mathbf{x}) - \mu_{\mathbf{y}_{1:n},y(\mathbf{s})}(\mathbf{x})\right]^2 \mathrm{d}\mathbf{x}, \tag{2.12}$$

whose maximization is equivalent to minimizing reconstruction error with additional data point $y(\mathbf{s})$. Both the function f and new observation $y(\mathbf{s})$, which is implicitly a function of both sampling location $\mathbf{s}$ and the function f but is not fully determined by them due to measurement noise, are unknown. Hence the expected utility is obtained by taking the expectation of (2.12) with respect to the joint posterior distribution of $y(\mathbf{s})$ and f given the observations collected so far,

$$\tilde{U}(\mathbf{s}) = -\int_{\mathcal{S}} \int \int \left[f(\mathbf{x}) - \mu_{\mathbf{y}_{1:n},y(\mathbf{s})}(\mathbf{x})\right]^2 p\left(y(\mathbf{s}),\, f \mid \mathbf{y}_{1:n}\right) \mathrm{d}y(\mathbf{s})\, \mathrm{d}f\, \mathrm{d}\mathbf{x}. \tag{2.13}$$

Under the Gaussian process model, this becomes

$$\tilde{U}(\mathbf{s}) = -\int_{\mathcal{S}} \left\{ \left[\mu_{\mathbf{y}_{1:n}}(\mathbf{x}) - \mu_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x}) \right]^2 + \kappa^2_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x}) \right\} \mathrm{d}\mathbf{x}, \tag{2.14}$$

where $\hat{y}(\mathbf{s}) = \mathrm{E}\{y(\mathbf{s}) \mid \mathbf{y}_{1:n}, \sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\} = \mu_{\mathbf{y}_{1:n}}(\mathbf{s})$ is the value of the observation we expect at $\mathbf{s}$ (which is equal to the posterior mean of the field at $\mathbf{s}$ given $\mathbf{y}_{1:n}$), and $\mu_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x})$ and $\kappa^2_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x})$ are the mean and variance of the posterior distribution of $f(\mathbf{x})$ given existing observations $\mathbf{y}_{1:n}$ and that location $\mathbf{s}$ is sampled with observed value $\hat{y}(\mathbf{s})$. See the Appendix for their explicit expressions, as well as the derivation of (2.14). In other words, the expected utility function is the spatial integral of

$$-\left[\mathrm{E}\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}\} - \mathrm{E}\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}, \hat{y}(\mathbf{s})\} \right]^2 - \mathrm{Var}\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}, \hat{y}(\mathbf{s})\} \tag{2.15}$$

over all $\mathbf{x} \in \mathcal{S}$. The integral (2.14) consists of two terms, an exploration term $-\kappa^2_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x})$ and an exploitation term $-\left[\mu_{\mathbf{y}_{1:n}}(\mathbf{x}) - \mu_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x}) \right]^2$. The exploration term is high (less negative) in regions where there is good coverage by the observations, and favors new locations $\mathbf{s}$ that would provide such coverage by being in an unexplored or comparatively less explored region. The exploitation term is high when the predictions with and without the new observation match, and favors new locations that are consistent with furthering the accuracy of the predictions. Maximizing (2.14) involves a trade-off between the exploration and exploitation terms that resembles the bias-variance trade-off that traditionally appears in many statistical procedures.

From a practical point of view, the integral (2.14) will need to be approximated numerically, for example, by using a Riemann sum that evaluates the integrand

over a uniform grid $\mathcal{G}$ that covers $\mathcal{S}$,

$$\tilde{U}(\mathbf{s}) \approx -\sum_{\mathbf{x}\in\mathcal{G}} \left\{ \left[\mu_{\mathbf{y}_{1:n}}(\mathbf{x}) - \mu_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x}) \right]^2 + \kappa^2_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x}) \right\}. \tag{2.16}$$

(Technically the expression (2.16) is not equivalent to the integral (2.14) because it is missing the area of the region that each grid point represents; however, the result of the optimization does not change if the grid is uniform and constant for all function evaluations of $\tilde{U}(\mathbf{s})$.)

2.2.2 Finding the maximum of f

For the operational goal of locating the maximum of f, we build our algorithm on the idea of expected improvement [81, 79, 34], which has been widely applied in finding optimal inputs of computer experiments [9, 76, 28]. Let $y^{\max}_{1:n} = \max\{y_1, \ldots, y_n\}$ be the maximum observed value so far. We define the improvement at location $\mathbf{s} \in \mathcal{S}$ as $U(\mathbf{s}, f) = \max\{f(\mathbf{s}) - y^{\max}_{1:n}, 0\}$, which depends on the unknown field f. If we take the expectation of $U(\mathbf{s}, f)$ with respect to the posterior distribution of f, the expected utility reduces to

$$\tilde{U}(\mathbf{s}) = \left[\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - y^{\max}_{1:n} \right] \Phi\left(\frac{\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - y^{\max}_{1:n}}{\kappa_{\mathbf{y}_{1:n}}(\mathbf{s})} \right) + \kappa_{\mathbf{y}_{1:n}}(\mathbf{s})\, \phi\left(\frac{\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - y^{\max}_{1:n}}{\kappa_{\mathbf{y}_{1:n}}(\mathbf{s})} \right), \tag{2.17}$$

where ϕ and Φ are the density and the cumulative distribution functions of the standard normal distribution. Intuitively, locations with the highest expected improvement provide the most substantial information about the location of the maximum of f.

Similar to (2.14), (2.17) consists of two terms that highlight how expected improvement trades off exploration and exploitation. For locations $\mathbf{s}$ where $\mu_{\mathbf{y}_{1:n}}(\mathbf{s})$ is large and $\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s})$ is small, the first term of (2.17) dominates, whereas for lo-

cations where $\mu_{\mathbf{y}_{1:n}}(\mathbf{s})$ is small and $\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s})$ is large the second term dominates. Hence the first (exploitation) term favors locations where we expect, with little uncertainty, f to be high, while the second (exploration) term favors locations where uncertainty of the field is large.

The expected improvement can be generalized by introducing a tuning parameter $c > 0$ such that $\tilde{U}(\mathbf{s}) = \mathrm{E}\left(\max\{f(\mathbf{s}) - cy^{\max}_{1:n}, 0\}\right)$. Tuning c can increase or decrease the first term of (2.17), which affects how long the vehicle stays close to the current maximum before the vehicle explores other regions. Assuming that $y^{\max}_{1:n} > 0$, which is usually true for real data from environmental processes, and if $0 < c < 1$, then $\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - cy^{\max}_{1:n} > \mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - y^{\max}_{1:n}$ and $\Phi\left(\frac{\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - cy^{\max}_{1:n}}{\kappa_{\mathbf{y}_{1:n}}(\mathbf{s})}\right) > \Phi\left(\frac{\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - y^{\max}_{1:n}}{\kappa_{\mathbf{y}_{1:n}}(\mathbf{s})}\right)$, so the contribution of the exploitation term increases. (The exploration term may increase or decrease, depending on how large $|\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - cy^{\max}_{1:n}|$ is compared to $|\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - y^{\max}_{1:n}|$.) Similarly, if $c > 1$, the contribution of the exploitation term decreases. This suggests that the tuning parameter c can be used to control how the expected utility trades off exploration and exploitation.

2.2.3 Alternative utility functions

Alternative utility functions for selecting the next sampling location are presented here.

Information based methods

Information theoretic methods can be used for the reconstruction problem. Consider a grid $\mathcal{G} \subset \mathcal{S}$ of possible locations to sample. The maximum entropy (D-optimal) solution is

$$\mathbf{s}^* = \arg\max_{\mathbf{s}\in\mathcal{G}} \log \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}), \tag{2.18}$$

and selects the grid location that has the largest reconstruction variance. Maximizing the related mutual information criterion gives

$$\mathbf{s}^* = \arg\max_{\mathbf{s}\in\mathcal{G}} \left[\log|\mathbf{\Sigma}_{\mathcal{G}\setminus\mathbf{s}}| + \log\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s})\right], \tag{2.19}$$

where $\mathbf{\Sigma}_{\mathcal{G}\setminus\mathbf{s}}$ is the posterior covariance matrix associated with $\mathcal{G}$ excluding $\mathbf{s}$,

$$\mathbf{\Sigma}_{\mathcal{G}\setminus\mathbf{s}} = \mathbf{C}_{\mathcal{G}\setminus\mathbf{s}} - \mathbf{C}^{\mathsf{T}}_{1:n,\mathcal{G}\setminus\mathbf{s}}\left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{C}_{1:n,\mathcal{G}\setminus\mathbf{s}} \tag{2.20}$$

with

$$[\mathbf{C}_{\mathcal{G}\setminus\mathbf{s}}]_{ij} = C(\mathbf{x}_i,\mathbf{x}_j), \quad \mathbf{x}_i,\mathbf{x}_j \in \mathcal{G}\setminus\mathbf{s}$$
$$[\mathbf{C}_{1:n,\mathcal{G}\setminus\mathbf{s}}]_{ij} = C(\mathbf{s}_i,\mathbf{x}_j), \quad \mathbf{s}_i \in \{\mathbf{s}_1,\ldots,\mathbf{s}_n\}, \quad \mathbf{x}_j \in \mathcal{G}\setminus\mathbf{s}.$$

These information-based criteria do not depend on the observed values $\mathbf{y}_{1:n}$, only on the existing sampling locations $\{\mathbf{s}_1,\ldots,\mathbf{s}_n\}$ so far.

P-algorithm for global optimization

The P-algorithm, extensively analyzed as a technique for stochastic global optimization in [85], selects the location that maximizes the probability of an increase of $\delta_n \geq 0$

$$\mathbf{s}^* = \arg\max_{\mathbf{s}\in\mathcal{S}} \Phi\left(\frac{\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - y^{\max}_{1:n} - \delta_n}{\kappa_{\mathbf{y}_{1:n}}(\mathbf{s})}\right). \tag{2.21}$$

For larger δ_n, search is more global, while $\delta_n = 0$ degenerates to repeated observations at the current maximum location. Criterion (2.21) is similar to the exploitation term in the expected improvement, and a choice of δ_n to be a fraction of the current maximum $y^{\max}_{1:n}$ results in a similar result as tuning c in the

generalized expected improvement. However, the exploitation term in the expected improvement has a multiplication factor of $\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - y_{1:n}^{\max}$ that naturally prevents the degeneration of repeated sampling at exactly the maximum location.

In Chapter 3, we compare the performance of these alternative utility functions to (2.14) and (2.17). More details of the derivations of the alternative utility functions are available in the Appendix. Note that entropy and mutual information criteria are also expected utilities of the form (2.11).

Chapter 3

Single Vehicle

In this chapter we build on the ideas introduced in Chapter 2 to construct our trajectory planning algorithm. In particular, our design parameter is the trajectory $\mathbf{Q} : \mathbb{R}_{\geq 0} \to \mathcal{S}$ followed by the vehicle. This trajectory determines the locations $\mathbf{s}_1, \mathbf{s}_2, \ldots$ at which observations are taken, with $\mathbf{s}_i = \mathbf{Q}(i\Delta)$ and Δ being the sampling period.

We construct the trajectory $\mathbf{Q}$ by stitching together optimal trajectory segments $\mathbf{q}_1^*, \mathbf{q}_2^*, \ldots$, each of which is followed by the vehicle for $k\Delta$ time units, i.e., $\mathbf{Q}(\{k[j-1]+i\}\Delta) = \mathbf{q}_j^*(i\Delta)$ for $i = 1, \ldots, k$. The segments are sequentially determined by repeatedly optimizing an appropriate objective function $R_j(\mathbf{q}_j)$ over a planning horizon of $pk\Delta$ time units. This objective function is either an extension of the expected gain in prediction accuracy (2.16) (if we are interested in optimal reconstruction) or the expected improvement (2.17) (if we are interested in locating the maximum of the function) over the planning horizon. pk is thought of as the number of lookahead steps, and the longer the time horizon $pk\Delta$, the more likely the computed trajectory is to be globally optimal. In practice, pk cannot be too large due to computation cost. The integer constants $p, k \geq 1$ are design parameters; p controls how myopic the algorithm is, and k controls how quickly

the trajectory adapts to new information. The choices of Δ, p, and k depend on the sampling frequency of sensors, the speed and maneuverability of the vehicle, and the size of $\mathcal{S}$. The algorithm is initialized by selecting an initial location and heading angle and setting $\mathbf{q}_1^*(t)$ to be a straight line for $t \in [0, k\Delta]$. It is halted when some stopping rule has been met, such as a fixed amount of time has passed, or the maximum reward over $\mathcal{S}$ falls below a predetermined threshold.

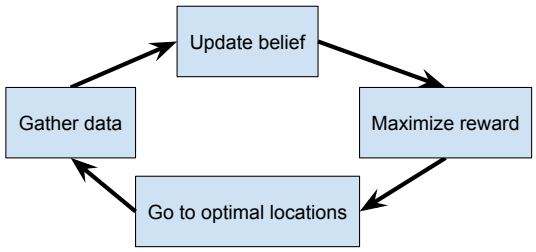


Figure 3.1: Components of the trajectory planning algorithm.

3.1 Objective functions for trajectory planning

We describe here the construction of objective functions R_{j+1} associated with the design of the $\{j+1\}$th optimal trajectory segment $\mathbf{q}_{j+1}^*$ that is appropriate for the two applications we've chosen. In both cases, the objective function uses the information contained in observations $\mathbf{y}_{1:n} = (y_1, \ldots, y_{jk})$ collected by the vehicle at previously visited locations $\mathbf{s}_1, \ldots, \mathbf{s}_{jk}$ to determine the best future sampling locations $\mathbf{s}_{jk+i} = \mathbf{q}_{j+1}(i\Delta)$ for $i = 1, \ldots, pk$ for the vehicle, of which k locations are visited, and the trajectory is recalculated with the newest k observations.

Both of the objective functions we present below share the following structure

$$R_{j+1}\left(\mathbf{q}_{j+1}\right) = \tilde{U}_{j+1}\left(\mathbf{q}_{j+1}\right) - \alpha P_{\text{leave}}\left(\mathbf{q}_{j+1}\right), \tag{3.1}$$

where $\tilde{U}_{j+1}$ is the extension of the expected utilities (2.16) and (2.17) to the

multiple locations of a planning segment, whose explicit forms are given in the following subsections, and P_{leave} is a penalty function

$$P_{\text{leave}}\left(\mathbf{q}_{j+1}\right) = \begin{cases} 0 & \text{if } \mathbf{q}_{j+1}(i\Delta) \in \mathcal{S}_B \text{ for all } i = 1, \ldots, pk \\ \max\limits_{i=1,\ldots,pk} \|\mathbf{q}_{j+1}(i\Delta) - \mathbf{x}_0\| & \text{if any } \mathbf{q}_{j+1}(i\Delta) \notin \mathcal{S}_B \end{cases}$$

where $\mathcal{S}_B \supset \mathcal{S}$ is a ball that contains $\mathcal{S}$ with center $\mathbf{x}_0$.

The first term in (3.1) measures the performance of the trajectory segment with respect to our main inferential goal, while the second term is a penalty that discourages the vehicle from wandering away from $\mathcal{S}$ [65]. In our experience, the inclusion of the penalty term is important, particularly in the early stages of exploration. Indeed, when little information about the structure of the field f is available, or when there exist local maxima near the boundary of $\mathcal{S}$, directly optimizing $\tilde{U}_{j+1}$ can lead to trajectories that lie outside our region of interest. We have chosen a structure for P such that, if the trajectory segment $\mathbf{q}_{j+1}$ contains sampling locations that are outside $\mathcal{S}_B$, we penalize $\tilde{U}_{j+1}$ with a term that is proportional to the maximum distance between a sampling point and the center of $\mathcal{S}$. The constant $\alpha \geq 0$ is a tuning parameter that is generally a couple orders of magnitude smaller than the average value for $\tilde{U}_{j+1}$. When $\alpha = 0$ there is no penalty for exploring far away from $\mathcal{S}$, while taking $\alpha \to \infty$ implies that trajectories that move the vehicle out of $\mathcal{S}_B$ are disallowed.

3.1.1 Extended objective function for optimal reconstruction of f

The expected utility function (2.16) can be adapted to the context of our trajectory planning algorithm by evaluating the quality of the reconstruction provided by adding a group of pk additional measurements whose locations are given

by the trajectory segment $\mathbf{q}_{j+1}$. More specifically, we define

$$\tilde{U}_{j+1}(\mathbf{q}_{j+1}) = -\sum_{\mathbf{x}\in\mathcal{S}_g} \left\{ \left[\mu_{\mathbf{y}_{1:n}}(\mathbf{x}) - \mu_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})\right]^2 + \kappa^2_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x}) \right\}, \tag{3.2}$$

where $\hat{\mathbf{y}}_{\mathbf{q}_{j+1}} = (\mu_{\mathbf{y}_{1:n}}(\mathbf{q}_{j+1}(\Delta)), \mu_{\mathbf{y}_{1:n}}(\mathbf{q}_{j+1}(2\Delta)), \ldots, \mu_{\mathbf{y}_{1:n}}(\mathbf{q}_{j+1}(pk\Delta)))$ is a vector of the current estimates of f at the pk locations on $\mathbf{q}_{j+1}$, and $\mu_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ and $\kappa^2_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ are the posterior mean and variance of $f(\mathbf{x})$ computed from using both $\mathbf{y}_{1:n}$ and $\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}$ as the observations. The explicit expressions for $\mu_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ and $\kappa^2_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ are given in the Appendix.

3.1.2 Extended objective function for finding the maximum of f

When the goal is locating the maximum of the field f, we construct our utility function by extending the notion of expected improvement in (2.17) to handle multiple locations. There are many ways in which this can be accomplished, but here we focus on a relatively aggressive approach that selects the trajectory segment where the maximum of the expected improvement over the segment is the largest,

$$\tilde{U}_{j+1}\left(\mathbf{q}_{j+1}\right) = \max_{i=1,\ldots,pk} \mathrm{E}\left(\max\left\{f\left(\mathbf{q}_{j+1}(i\Delta)\right) - y^{\max}_{1:n},\, 0\right\}\right), \tag{3.3}$$

where $\mathrm{E}\left(\max\left\{f\left(\mathbf{q}_{j+1}(i\Delta)\right) - y^{\max}_{1:n},\, 0\right\}\right)$ is computed using $\tilde{U}(f\left(\mathbf{q}_{j+1}(i\Delta)\right))$ in (2.17) for each $i = 1, \ldots, pk$. Writing it all out, $\tilde{U}_{j+1}(\mathbf{q}_{j+1}) =$

$$\max_i \Big\{ \left[\mu_{\mathbf{y}_{1:n}}(\mathbf{q}_{j+1}(i\Delta)) - y^{\max}_{1:n}\right] \Phi\left(z(\mathbf{q}_{j+1}(i\Delta))\right) + \kappa_{\mathbf{y}_{1:n}}(\mathbf{q}_{j+1}(i\Delta)) \phi\left(z(\mathbf{q}_{j+1}(i\Delta))\right) \Big\}, \tag{3.4}$$

where $z(\mathbf{q}_{j+1}(i\Delta)) = \dfrac{\mu_{\mathbf{y}_{1:n}}(\mathbf{q}_{j+1}(i\Delta)) - y_{1:n}^{\max}}{\kappa_{\mathbf{y}_{1:n}}(\mathbf{q}_{j+1}(i\Delta))}$.

3.2 Constraints and parametrization of $\mathbf{q}_j$

Given $n = jk$ total observations collected during the first j locally optimal trajectory segments, the next local trajectory $\mathbf{q}_{j+1}^* = \arg\max R_{j+1}(\mathbf{q}_{j+1})$ is subject to the initial conditions and constraints

$$\begin{aligned} \mathbf{c}_1(\mathbf{q}_{j+1}) &= \mathbf{q}_{j+1}(0) - \mathbf{q}_j^*(k\Delta) = \mathbf{0}, \\ \mathbf{c}_2(\mathbf{q}_{j+1}) &= \dot{\mathbf{q}}_{j+1}(0) - \dot{\mathbf{q}}_j^*(k\Delta) = \mathbf{0}, \\ \mathbf{c}_3\left(\dot{\mathbf{q}}_{j+1}(t), \ddot{\mathbf{q}}_{j+1}(t), t\right) &\geq \mathbf{0}, \qquad t \in [0, pk\Delta] \end{aligned} \tag{3.5}$$

The conditions $\mathbf{c}_1$ and $\mathbf{c}_2$ enforce continuity of the position and velocity fields of the vehicle, while $\mathbf{c}_3$ represents problem-specific constraints on the vehicle's velocity and acceleration. These can include, for example, forward velocity only, bounded velocity, maximum acceleration, and maximum turning rate. With these constraints, the structure of a general trajectory planning algorithm is outlined in Algorithm 1. In practice, we simplify the optimization process by parametrizing $\mathbf{q}_1, \mathbf{q}_2, \ldots$ in terms of a small set of parameters $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots$, thereby transforming the trajectory planning problem into a low-dimensional nonlinear optimization problem (see [36] for a theoretical development of parametric optimal control).

Algorithm 1 Optimal trajectory algorithm (general, single vehicle)

Input: goal, $\mathcal{S}$, vehicle kinematics, vehicle's initial location and heading angle, sampling period Δ, trajectory segment size k, planning horizon parameter p, out-of-bounds penalty parameter α, prior distributions for hyperparameters $(\sigma^2, \boldsymbol{\varphi}, \psi)$, and stopping rule

Output: Optimal trajectory $\mathbf{Q}$

1: Set $j = 1$ and $n = k$.
2: Initialize $\mathbf{Q}$ with k locations by going straight along initial heading angle from vehicle's initial location
3: Initialize $\mathbf{y}_{1:n}$ with the measurements taken at $\mathbf{Q}$
4: **while** stopping rule is not met **do**
5: Estimate $(\sigma^2, \boldsymbol{\varphi}, \psi)$ by maximizing (2.10)
6: Obtain $\mathbf{q}^*_{j+1}$ by maximizing $R_{j+1}(\mathbf{q}_{j+1})$ in (3.1) subject to (3.5)
7: Append $\mathbf{q}^*_{j+1}(i\Delta)$, $i = 1, \ldots, k$ to $\mathbf{Q}$
8: Sample at $\mathbf{q}^*_{j+1}(i\Delta)$, $i = 1, \ldots, k$
9: Add new measurements to $\mathbf{y}_{1:n}$
10: Set $j = j + 1$ and $n = n + k$
11: **end while**

3.2.1 Constant speed circular arcs

As an example, consider the kinematic model for a simple car [43],

$$\begin{aligned} \dot{x} &= v \cos\theta \\ \dot{y} &= v \sin\theta \\ \dot{\theta} &= v \frac{u_\theta}{R_{\min}} \end{aligned} \tag{3.6}$$

with constant forward velocity v, minimum turning radius $R_{\min}$, and control input $u_\theta \in [-1, 1]$. In this context, x and y represent position coordinates, and θ represents the vehicle's heading angle relative to the x-direction. We can rewrite $\dot{\theta}$ as $\dot{\theta} = \frac{v}{R}$, where R is the turning radius with $|R| \geq R_{\min}$. If u_θ (and hence R) is constant[1] for $t \geq 0$, integrating the system (3.6) with initial conditions $x(0) = x_0$,

[1]Having the control sequence be piecewise constant over a specified number of time steps, and recomputing the trajectory at short intervals, allows for a longer planning horizon without too much cost. These solutions are considered 'quasi-optimal'.

$y(0) = y_0$, and $\theta(0) = \theta_0$ results in

$$x(t) = R\sin\left(\frac{v}{R}t + \theta_0\right) + x_0 - R\sin\theta_0$$
$$y(t) = -R\cos\left(\frac{v}{R}t + \theta_0\right) + y_0 + R\cos\theta_0.$$

Using the identities $\sin(\theta) = \cos(\theta - \pi/2)$ and $\cos(\theta) = -\sin(\theta - \pi/2)$, the above is equivalent to

$$x(t) = R\cos\left(\frac{v}{R}t + \theta_0 - \frac{\pi}{2}\right) + x_0 - R\sin\theta_0 \tag{3.7}$$
$$y(t) = R\sin\left(\frac{v}{R}t + \theta_0 - \frac{\pi}{2}\right) + y_0 + R\cos\theta_0, \tag{3.8}$$

which parametrizes a circular arc for the vehicle segment

$$\mathbf{q}_j(t) = \begin{bmatrix} q_{j,1}(t) \\ q_{j,2}(t) \end{bmatrix} = \begin{bmatrix} \xi_{j,1}\cos(\xi_{j,2}t + \xi_{j,3}) + \xi_{j,4} \\ \xi_{j,1}\sin(\xi_{j,2}t + \xi_{j,3}) + \xi_{j,5} \end{bmatrix}, \quad t \in [0, pk\Delta], \tag{3.9}$$

where $\xi_{j,1}\xi_{j,2} = v$ is the fixed forward velocity, and $\xi_{j,1} = R$ is the (signed) turning radius of the arc. $\xi_{j,1} > 0$ results in a counterclockwise arc and $\xi_{j,1} < 0$ clockwise, corresponding to turning left and right, respectively. A straight segment arises as a special case when $\xi_{j,1}$ goes to positive or negative infinity. An upper bound on the turning rate of the vehicle can be specified though a minimum turning radius, so we constrain $|\xi_{j,1}| \geq R_{\min}$.

For continuity of position and velocity, $\xi_{j,1}, \ldots, \xi_{j,5}$ in the parametrization (3.9)

are constrained by

$$
\begin{aligned}
|\xi_{j,1}| &\geq R_{\min} \\
\xi_{j,2} &= \frac{v}{\xi_{j,1}} \\
\xi_{j,3} &= \theta_{j-1} - \frac{\pi}{2} \\
\xi_{j,4} &= q^*_{j-1,1}(k\Delta) - \xi_{j,1} \sin\theta_{j-1} \\
\xi_{j,5} &= q^*_{j-1,2}(k\Delta) + \xi_{j,1} \cos\theta_{j-1}
\end{aligned}
$$

leaving only $\xi_{j,1}$, corresponding to how much and in which direction the vehicle will turn, for optimization. The derivation for these requirements are in the Appendix. Thus finding the optimal trajectory segment reduces to finding the optimal radius $\xi_{j,1}$ for the circular arc subject to $|\xi_{j,1}| \geq R_{\min}$. This is a univariate maximization problem that can be solved using a numerical method such as Brent's method [12]. Algorithm 2 summarizes the trajectory planning algorithm with the constant speed circular arc parametrization.

3.2.2 Variable speed circular arcs

For more generality, the vehicle is allowed to move at variable speeds in a range $v \in [v_{\min}, v_{\max}]$. Modifying (3.6) to allow for variable speeds gives

$$
\begin{aligned}
\dot{x} &= v\cos\theta \\
\dot{y} &= v\sin\theta \\
\dot{\theta} &= v\frac{u_\theta}{R_{\min}} = \frac{v}{R} \\
\dot{v} &= u_v
\end{aligned}
$$

Algorithm 2 Optimal trajectory algorithm (constant speed circular arcs)

Input: goal, $\mathcal{S}$, vehicle speed v, minimum turning radius $R_{\min}$, vehicle's initial location $(q_{1,1}(0), q_{1,2}(0))$ and heading angle θ_1, sampling period Δ, trajectory segment size k, planning horizon parameter p, out-of-bounds penalty parameter α, prior distributions for hyperparameters $(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi})$, and stopping rule

Output: Optimal trajectory $\mathbf{Q}$

1: Initialize $j = 1$ and $n = k$
2: Initialize $\mathbf{Q}$ by $\mathbf{q}_1^*(t) = (q_{1,1}(0) + vt\cos(\theta_1),\ q_{1,2}(0) + vt\sin(\theta_1))$ for $t \in [0, k\Delta]$
3: Initialize $\mathbf{y}_{1:n}$ with the measurements taken at $\mathbf{q}_1^*(\Delta), \ldots, \mathbf{q}_1^*(k\Delta)$
4: **while** stopping rule is not met **do**
5: Set $j = j + 1$
6: Estimate $(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi})$ by maximizing (2.10)
7: Compute $\mathbf{q}_j^* = \arg\max\ R_j(\mathbf{q}_j)$ subject to:
$|\xi_{j,1}| \geq R_{\min}$
$\xi_{j,2} = v/\xi_{j,1}$
$\xi_{j,3} = \theta_{j-1} - \pi/2$
$\xi_{j,4} = q^*_{j-1,1}(k\Delta) - \xi_{j,1}\sin\theta_{j-1}$
$\xi_{j,5} = q^*_{j-1,2}(k\Delta) + \xi_{j,1}\cos\theta_{j-1}$
8: Append $\mathbf{q}_j^*(i\Delta)$, $i = 1, \ldots, k$ to $\mathbf{Q}$
9: Sample at $\mathbf{q}_j^*(i\Delta)$, $i = 1, \ldots, k$ and append new measurements to $\mathbf{y}_{1:n}$
10: Set $\theta_j = \operatorname{atan2}\left(\dot{q}^*_{j,2}(k\Delta), \dot{q}^*_{j,1}(k\Delta)\right)$
11: Set $n = n + k$
12: **end while**

with control inputs u_θ and u_v. Again, supposing u_θ and u_v are constant for an integration time, the position of the vehicle is

$$x(t) = R\sin\left(\frac{u_v}{2R}t^2 + \frac{v_0}{R}t + \theta_0\right) + x_0 - R\sin\theta_0$$
$$y(t) = -R\cos\left(\frac{u_v}{2R}t^2 + \frac{v_0}{R}t + \theta_0\right) + y_0 + R\cos\theta_0,$$

which can be written in the form

$$\mathbf{q}_j(t) = \begin{bmatrix} q_{j,1}(t) \\ q_{j,2}(t) \end{bmatrix} = \begin{bmatrix} \xi_{j,1}\cos\left(\frac{1}{2}\frac{\xi_{j,2}}{\xi_{j,1}}t^2 + \xi_{j,3}t + \xi_{j,4}\right) + \xi_{j,5} \\ \xi_{j,1}\sin\left(\frac{1}{2}\frac{\xi_{j,2}}{\xi_{j,1}}t^2 + \xi_{j,3}t + \xi_{j,4}\right) + \xi_{j,6} \end{bmatrix}, \quad t \in [0, pk\Delta], \quad (3.10)$$

where $\xi_{j,1} = R$, $\xi_{j,2} = u_v$, and $\xi_{j,1}\xi_{j,3} = v_0$, the velocity at the start of the segment. Equation (3.10) parametrizes the vehicle segments as circular arcs with variable speed that are continuous for all t, and with angular velocities that are piecewise continuous during each iteration j of the algorithm. $\xi_{j,2}$ is a linear acceleration term and setting $\xi_{j,2} = 0$ reduces (3.10) to (3.9).

With this parametrization, $\xi_{j,3}$ and $\xi_{j,4}$ are set by enforcing continuity of velocity, and $\xi_{j,5}$ and $\xi_{j,6}$ are set by enforcing continuity of position between segments

$$\begin{aligned}
\xi_{j,3} &= \frac{\|\dot{\mathbf{q}}^*_{j-1}(k\Delta)\|}{\xi_{j,1}} \\
\xi_{j,4} &= \theta_{j-1} - \frac{\pi}{2} \\
\xi_{j,5} &= q^*_{j-1,1}(k\Delta) - \xi_{j,1}\sin\theta_{j-1} \\
\xi_{j,6} &= q^*_{j-1,2}(k\Delta) + \xi_{j,1}\cos\theta_{j-1}.
\end{aligned}$$

This leaves optimizing over $\xi_{j,1}$ and $\xi_{j,2}$ subject to the constraints

$$\begin{aligned}
&|\xi_{j,1}| \geq R_{\min} \\
&\xi_{j,2} \in \left[\frac{v_{\min} - \|\dot{\mathbf{q}}^*_{j-1}(k\Delta)\|}{t},\ \frac{v_{\max} - \|\dot{\mathbf{q}}^*_{j-1}(k\Delta)\|}{t}\right]
\end{aligned}$$

for all $t \in [0, pk\Delta]$, where $\|\dot{\mathbf{q}}^*_{j-1}(k\Delta)\|$ is the speed of the vehicle at the end of the previous segment. We solve this bivariate maximization problem using a quasi-Newton method with box constraints [14]. The summary of the optimal trajectory algorithm with variable speed circular arc parametrization is provided by Algorithm 3.

More complicated parametrizations are available; for example, [27] uses clothoid arcs to enforce continuous curvature when transitioning between curves of different curvatures. However, we focus on sequential circular arcs for fast computation

Algorithm 3 Optimal trajectory algorithm (variable speed circular arcs)

Input: goal, $\mathcal{S}$, $v_{\min}$, $v_{\max}$, $R_{\min}$, vehicle's initial location $\mathbf{q}_1(0)$, speed $\|\dot{\mathbf{q}}_1(0)\|$, and heading angle θ_1, sampling period Δ, trajectory segment size k, planning horizon parameter p, out-of-bound penalty α, prior distributions for hyperparameters $(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi})$, and stopping rule

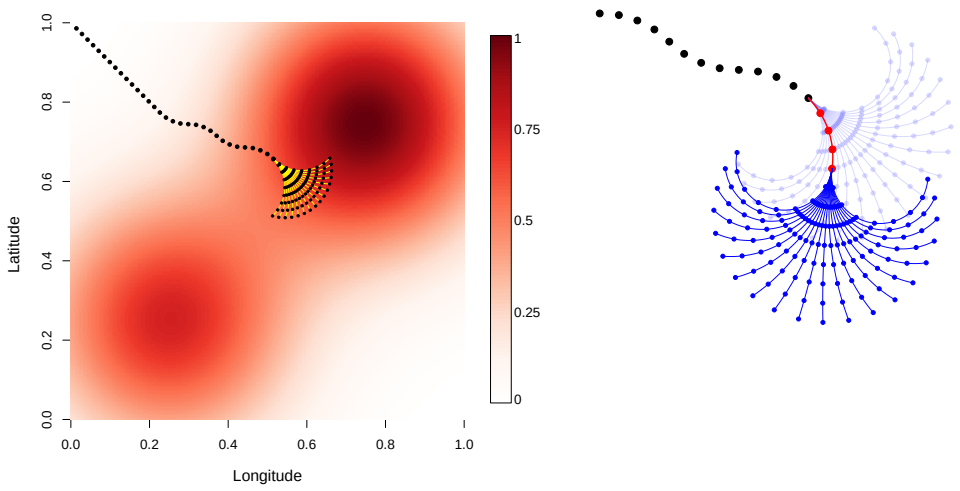
Output: Optimal trajectory $\mathbf{Q}$

Set $j = 1$ and $n = k$
Initialize $\mathbf{Q}$ by $\mathbf{q}_1^*(t) = (q_{1,1}(0) + \|\dot{\mathbf{q}}_1(0)\| t\cos(\theta_1),\ q_{1,2}(0) + \|\dot{\mathbf{q}}_1(0)\| t\sin(\theta_1))$ for $t \in [0, k\Delta]$
Initialize $\mathbf{y}_{1:n}$ with the measurements taken at $\mathbf{q}_1^*(\Delta), \ldots, \mathbf{q}_1^*(k\Delta)$
while stopping rule is not met **do**
    Set $j = j + 1$
    Estimate $(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi})$ by maximizing (2.10)
    Compute $\mathbf{q}_j^* = \arg\max R_j(\mathbf{q}_j)$ subject to:
      $|\xi_{j,1}| \geq R_{\min}$
      $v_{\min} \leq \|\dot{\mathbf{q}}_{j-1}(k\Delta)\| + \xi_{j,2} t \leq v_{\max}$ for all $t \in [0, pk\Delta]$
      $\xi_{j,3} = \|\dot{\mathbf{q}}_{j-1}(k\Delta)\| / \xi_{j,1}$
      $\xi_{j,4} = \theta_{j-1} - \pi/2$
      $\xi_{j,5} = q^*_{j-1,1}(k\Delta) - \xi_{j,1}\sin\theta_{j-1}$
      $\xi_{j,6} = q^*_{j-1,2}(k\Delta) + \xi_{j,1}\cos\theta_{j-1}$
    Append $\mathbf{q}_j^*(i\Delta)$, $i = 1, \ldots, k$ to $\mathbf{Q}$
    Sample at $\mathbf{q}_j^*(i\Delta)$, $i = 1, \ldots, k$ and append new measurements to $\mathbf{y}_{1:n}$
    Set $\theta_j = \text{atan2}(\dot{q}^*_{j,2}(k\Delta), \dot{q}^*_{j,1}(k\Delta))$
    Set $n = n + k$
end while

and leave the fine trajectory smoothing to the vehicle's autopilot.

3.3 Computational details

In summary, the trajectory planning algorithm consists of sequentially optimizing an objective function appropriate for the goal. The objective function is based on expected utility, where the expectation is taken with respect to the most recent distribution of f (the vehicle's current belief of f). The objective function and belief updates are based on the data collected so far, and are updated regularly at short intervals, allowing (1) the belief to adapt quickly to the new data,



(a) (b)

Figure 3.2: (a) A vehicle trajectory composed of straight lines and circular arcs. Large (small) black dots show where the vehicle has taken (will possibly take) measurements. Yellow curves sweep over an area of feasible trajectories for the next $pk\Delta$ time period. (b) Sequential replanning of trajectory segments.

and (2) efficient computation of optimal segments.

Figure 3.2a shows a short trajectory parametrized by constant speed circular arcs as described in Section 3.2.1. Figure 3.2b illustrates the planning and execution horizons. Note that the locations of the planning horizon form a fan-like shape. Figure 3.3 shows our algorithm in action, with the goal of finding the maximum. Pictured is a trajectory consisting of $n = 160$ sampling locations overlaid on the current belief of f. The Mean plot shows $\mu_{\mathbf{y}_{1:n}}(\mathbf{x})$ for $\mathbf{x}$ in a 101×101 regular grid of $\mathcal{S}$. The Variance plot is $\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x})$. The Expected Improvement plots $\mathrm{E}\left(\max\{f(\mathbf{x}) - y^{\max}_{1:n}, 0\}\right)$, and the red dots in the panel indicate the next optimal segment, having maximized (3.3) over the fan extending from $\mathbf{s}_{160}$.

Fast computation is important for real-time trajectory planning, and as $\mathbf{y}_{1:n}$ grows, computation time for our algorithm, which is dominated by matrix computations, increases polynomially. In addition to saving time, reducing the com-

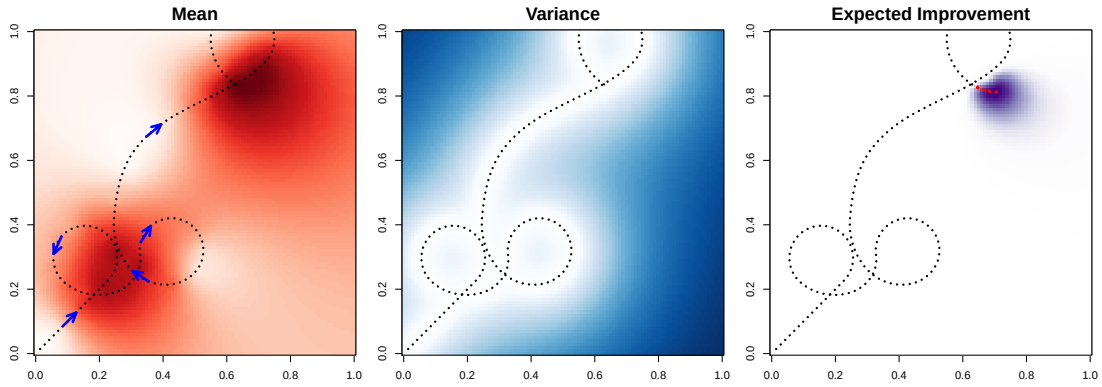


Figure 3.3: Plots of the belief after 160 observations taken at black dots. Red dots show the next optimal segment. Darker colors indicate higher values.

putational load is important for conserving battery life and maximizing vehicle deployment time. We can reduce the computational load by updating the hyperparameters $(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi})$ of the Gaussian process prior by the maximization in (2.10) less frequently, and by stopping the updates to these parameters after a predetermined number of observations has been collected.

Another way to reduce computation is by simplifying matrix inversions using formulas for block matrices. In the optimal reconstruction problem, calculating $\mu_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ in (3.2) requires the inversion of an $(n+pk) \times (n+pk)$ matrix. This calculation is required for every function evaluation of the reward function R_j, which is multiple times per iteration j, depending on how R_j is maximized[2]. Using blockwise inversion, N inversions of an $(n+pk) \times (n+pk)$ matrix involves a single $n \times n$ inversion (reflecting the current n observations) and N smaller $pk \times pk$ inversions (for the contending pk new observations on the next segment). The same application of inverting block matrices should be implemented if the hyperparameters are no longer being updated, as the upper left corner of the variance-covariance matrix $\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n$ does not change as the matrix grows.

[2]In our implementations, we first evaluate R_j over a grid of the parameters in $\mathbf{q}_j$, finding the interval (or box, in the variable speed parametrization) that contains the maximizer of R_j. This interval (or box) then provides the bounds of the numerical maximization method.

The trajectory planning algorithms can be slightly modified by adding a restart step in which the vehicle is moved to a location selected uniformly at random on $\mathcal{S}$ before computing the next segment of the trajectory. In that case, it can be shown that, under some mild conditions on the differentiability of the mean and covariance functions m and C, as well as the true field f, the point estimator $\hat{f}_n$ generated by the above algorithm is consistent as $n \to \infty$. The result follows from an application of Theorem 6 in [18] (see also [30] and [77]).

3.4 Evaluations

Here we implement the methods described in Sections 3.1 and 3.2 for optimally reconstructing or locating the maximum of a few different simulated fields. The evaluations in this section are carried out *in silico* using observations simulated from fields $f_i(\mathbf{x})$, for $i = 1, 2, 3$ and $\mathbf{x} \in \mathcal{S} = [0, 1]^2$. These true fields are designed with increasing complexity, exhibiting isotropy ($i = 1$), anisotropy ($i = 2$), and nonstationarity ($i = 3$). These true fields, which are shown in Figure 3.4, share a common functional form, a sum of two Gaussians

$$f_i(\mathbf{x}) = 0.75 \exp\Big(- a_{i1} \|\mathbf{x} - \mathbf{x}_{i1}\|_{\mathbf{A}_{i1}} \Big) + \exp\Big(- a_{i2} \|\mathbf{x} - \mathbf{x}_{i2}\|_{\mathbf{A}_{i2}} \Big), \tag{3.11}$$

with values $a_{i1}, a_{i2}, \mathbf{x}_{i1}, \mathbf{x}_{i2}$, $\mathbf{A}_{i2}$, and $\mathbf{A}_{i2}$ listed in Table 3.1. These fields all have a local maximum (with value around 0.75) in the lower left and a global maximum (with value around 1) in the upper right. In all cases, the true measurement noise is randomly generated from a normal distribution with mean 0 and variance 0.04.

We implement our algorithm using both the exponential and anisotropic Matérn

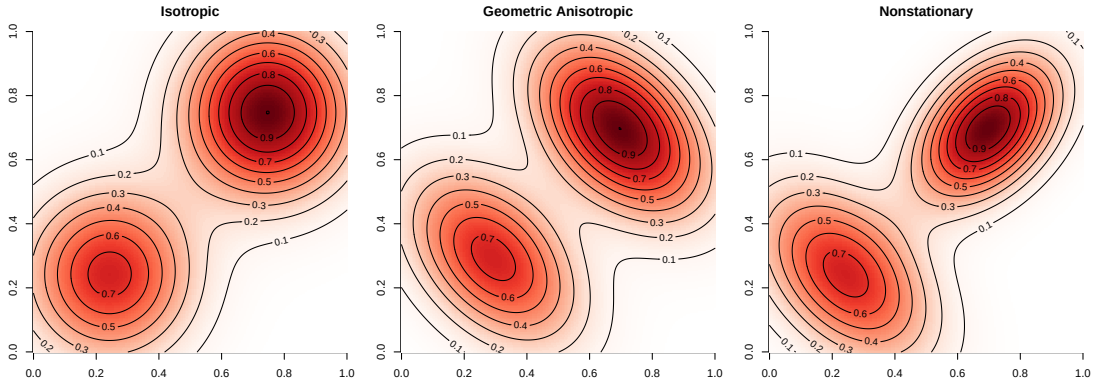


Figure 3.4: Fields used as the truth in simulations.

	Isotropic $i = 1$	Anisotropic $i = 2$	Nonstationary $i = 3$
a_{i1}	15	10	10
a_{i2}	15	10	15
$\mathbf{x}_{i1}$	(0.25, 0.25)	(0.3, 0.3)	(0.25, 0.25)
$\mathbf{x}_{i2}$	(0.75, 0.75)	(0.7, 0.7)	(0.7, 0.7)
$\mathbf{A}_{i1}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1.75 & 0.75 \\ 0.75 & 1.75 \end{bmatrix}$	$\begin{bmatrix} 1.75 & 0.75 \\ 0.75 & 1.75 \end{bmatrix}$
$\mathbf{A}_{i2}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1.75 & 0.75 \\ 0.75 & 1.75 \end{bmatrix}$	$\begin{bmatrix} 1.75 & -0.75 \\ -0.75 & 1.75 \end{bmatrix}$

Table 3.1: True field details.

covariance functions and compare their performance. The prior distributions[3] used for σ^2, $\boldsymbol{\varphi}$, and $\boldsymbol{\psi}$ are given in Table 3.2. As discussed in Section 2.1, the only hyperparameter in the mean function $m(\mathbf{x}; \boldsymbol{\varphi})$ is φ, which is given a flat prior distribution, $p(\varphi) \propto 1$. The prior distributions for $1/\sigma^2$, $1/\tau^2$, λ (in the exponential covariance), λ_1, λ_2 (in the anisotropic Matérn covariance), and ν are all given Gamma distributions with shape parameter 2 and scale parameter b set to a value that reflects the size of the parameter, leading to weakly informative priors. For $1/\sigma^2$ and $1/\tau^2$, b_σ and b_τ are set to equal to half of the variance of the observations

[3]The hyperpriors in Table 3.2 are consistently used for all experiments involved in this document, with the only modifications for length scales, as the area of $\mathcal{S}$ varies in different applications.

	$1/\sigma^2$	φ	$1/\tau^2$	λ	λ_1, λ_2	ν	ϑ
Exp.	$Ga\,(2, b_\sigma)$	$p(\varphi) \propto 1$	$Ga\,(2, b_\tau)$	$Ga\,(2, b_{\lambda_e})$	-	-	-
Matérn	$Ga\,(2, b_\sigma)$	$p(\varphi) \propto 1$	$Ga\,(2, b_\tau)$	-	$Ga\,(2, b_{\lambda_m})$	$Ga(2, 2)$	$U(0, 2\pi)$

Table 3.2: Hyperprior distributions. ϑ is the angle of rotation in rotation matrix $\mathbf{P}$ in equation (2.3). $Ga(a, b)$ represents the Gamma distribution with mean $\frac{a}{b}$ and variance $\frac{a}{b^2}$.

$\mathbf{y}_{1:n}$, up until $n = 100$, and equal to the variance of $\mathbf{y}_{1:100}$ once $n > 100$. This choice reflects the assumption that, a priori, we expect about half the variability of the data to be due to measurement noise and about half to come from the variability in the underlying field f. For the prior on the length parameter λ of the exponential, we set $b_{\lambda_e} = -\sqrt{2}\log(0.05)$, reflecting the assumption that, a priori, the correlation between measurements taken at the farthest points on $\mathcal{S}$ is around 0.05. The parameters λ_1 and λ_2 are given the same prior distribution, and similar reasoning is used for setting their hyperprior b_{λ_m}. The hyperprior choice for ν reflects the assumption that, a priori, f is on the edge of differentiability ($\nu \leq 1$ means the Gaussian process is nondifferentiable, and $\nu > 1$ generates increasingly smooth processes as ν increases). ϑ is uniform in the interval $(0, 2\pi)$, reflecting that the anisotropy in the Matérn covariance is equally likely in any direction.

In the following simulations, we use Algorithm 2 with $\Delta = 1$, $k = 4$, $p = 3$, $R_{\min} = 0.035$, and $v = 0.046$. These values correspond to a AeroVironment RQ-14 Dragon Eye UAV[4] flying at 20 km/h with maximum bank angle 45° over a 2.5 kilometer by 2.5 kilometer region taking measurements every ten seconds, scaled down to $\mathcal{S} = [0, 1]^2$.

To evaluate the performance of our methods, we run 200 Monte Carlo simulations using each of the exponential and Matérn covariance functions for each

[4]The AeroVironment RQ-14 Dragon Eye is used in the United States military as a small reconnaissance UAV. It is also a platform in NASA's Airborne Science Program for low altitude remote sensing and in situ sampling.

type of ground truth. In every simulation the vehicle starts at the origin with a heading angle of $\frac{\pi}{4}$, which means that the local maximum at $\mathbf{x}_{i1}$ obstructs access to the global maximum located at $\mathbf{x}_{i2}$. We assume that no data is available at the beginning of the simulation and instruct the vehicle to go straight for $k\Delta$ time units, collecting k measurements that are used to initialize our algorithm.

3.4.1 Results and comparisons: optimal reconstruction

First, we investigate the performance of our algorithm in reconstructing f as accurately as possible. With 200 Monte Carlo simulations of each case, the performance of our algorithm with exponential and Matérn covariances are compared to each other and also compared to that of a raster scan, "lawnmower" trajectory with the same speed. The flight time of each simulation is 332Δ, which is the time the lawnmower trajectory uses to cover $\mathcal{S}$ with turns with minimum turning radius $R_{\min}$. Figure 3.5 shows a full typical trajectory generated by our optimal reconstruction algorithm for each of the six cases. The trajectories are overlaid on the posterior mean $\hat{f}_{332} = \mu_{\mathbf{y}_{1:332}}$ (which provides the reconstruction at $t = 332$, first column), the variance of posterior distribution $\kappa^2_{\mathbf{y}_{1:n}}$ (which provides an estimate of the uncertainty associated with the reconstruction, second column) and the squared prediction error $\left[f - \hat{f}_{332}\right]^2$ (third column). Note that the trajectories tend to cover the region of interest evenly and that the uncertainty in the estimate $\hat{f}_{332}$ is roughly constant. These patterns appear reasonable given the operational goal of optimal reconstruction. Variance and squared error are plotted with the same scale for all six cases, suggesting that using the exponential covariance function results in higher variance and reconstruction error compared to the Matérn.

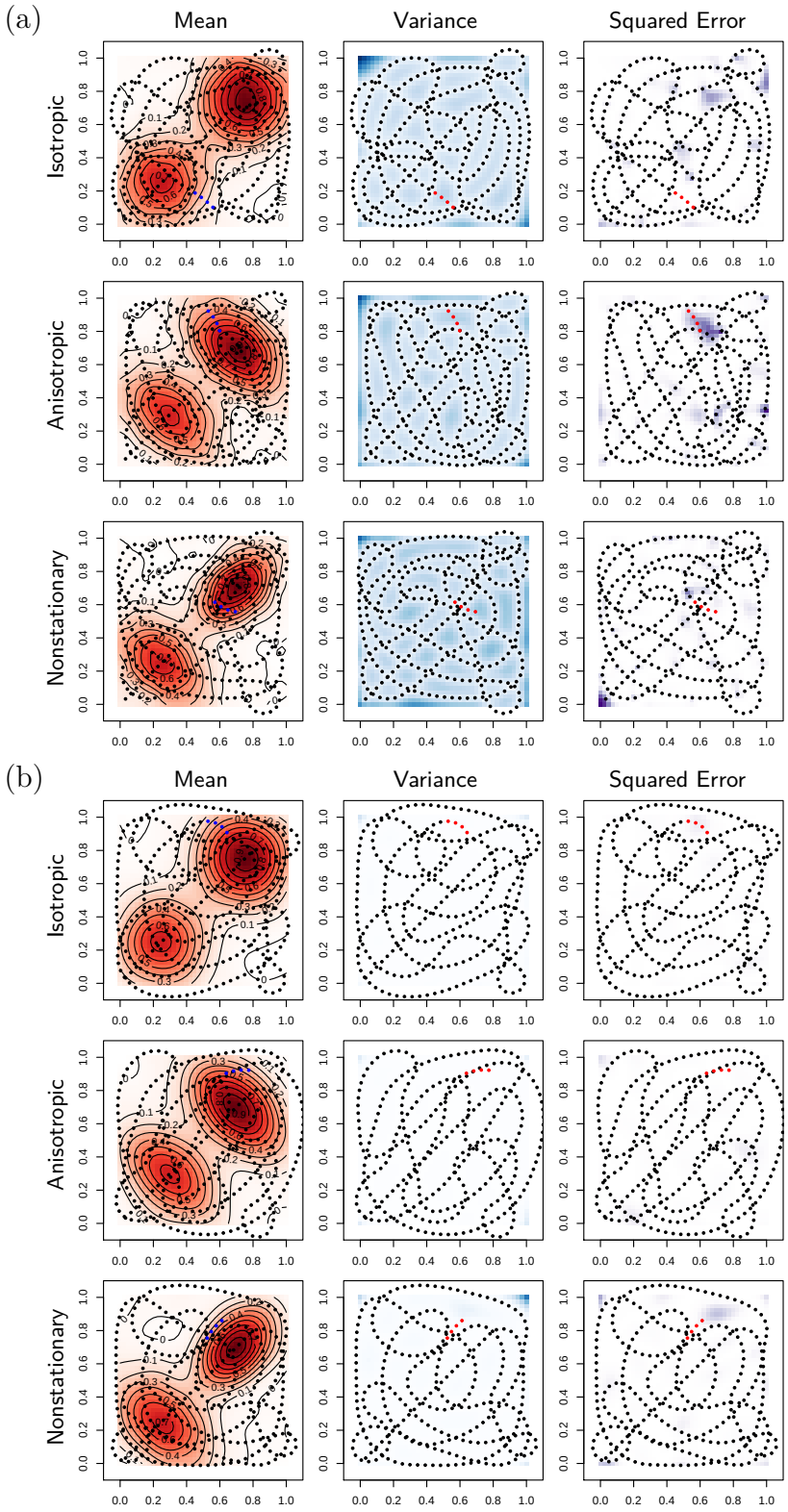


Figure 3.5: Trajectory samples of the six different cases using (a) exponential covariance and (b) Matérn covariance are computed for isotropic, anisotropic, and nonstationary fields.

To judge the accuracy of the reconstruction over time, we track the mean squared prediction error

$$\mathrm{MSE}(n\Delta) = \frac{1}{|\mathcal{S}_g|} \sum_{\mathbf{x} \in \mathcal{S}_g} \left[f(\mathbf{x}) - \hat{f}_n(\mathbf{x}) \right]^2, \quad n = 1, \ldots, 332, \tag{3.12}$$

for some grid $\mathcal{S}_g$ that covers $\mathcal{S}$, where $|\mathcal{S}_g|$ denotes the number of gridpoints. Note that we use the same $\mathcal{S}_g$ for the computation of the reward in (3.2) as in the computation of the mean squared prediction error in (3.12), but these sets can be different. The average log-mean squared error of the 200 simulations is plotted in Figure 3.6 for each of the six combinations of three true fields and two covariance functions. While the log-MSE values start at the same level in all cases, the log-MSE from the trajectories generated by our adaptive algorithm soon drops to about half of the beginning level and is much smaller for most of the run time. At the end of the simulations, when the lawnmower trajectories complete their sweep of $\mathcal{S}$, the log-MSE values are again close. Comparing across exponential and Matérn covariance models, we see that using the Matérn improves performance

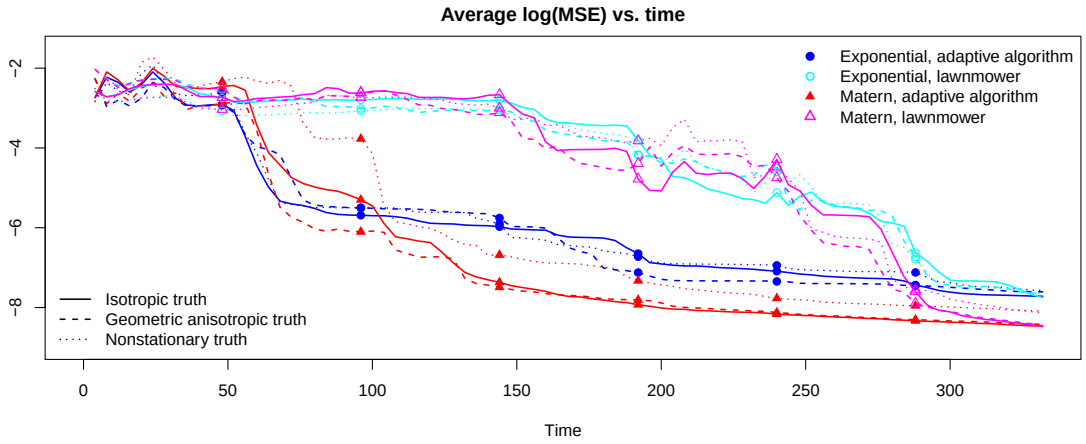


Figure 3.6: Average over all simulations of the log-mean squared prediction error for adaptive and lawnmower trajectories, using the exponential and Matérn covariances for prediction.

in optimal reconstruction as expected, and the nonstationary truth is usually the most difficult to accurately reconstruct.

Comparison with mutual information criterion

For a comparison to the mutual information (MI) criterion, we discretize $\mathcal{S}$ using a regular grid $\mathcal{G}$. Then we select the location within a radius of $pk\Delta$ distance from the vehicle's current location that maximizes (2.19). This location is the next waypoint target (but may not be reached before replanning occurs) and determines the next direction for the vehicle. A path connecting the current location to the MI maximizer is formed using a minimum radius turn and straight line, which is followed for $k\Delta$ time. The process is repeated for the same number of iterations, resulting in a trajectory of 332 locations. Like the previous set, 200 simulations are run for each truth type using the Matérn covariance, and the comparisons of resulting log-MSEs are shown in Figure 3.7. The performance using MI is comparable to that of our algorithm, doing slightly worse for the isotropic and anisotropic truths, but slightly better in the first half for the nonstationary truth.

A major drawback of using mutual information, at least for online planning of trajectories, is computation cost. In (2.19) the covariance matrix $\mathbf{\Sigma}_{\mathcal{G}\setminus\mathbf{s}}$ associated with the unsampled points is typically large. For example, if $\mathcal{G}$ is a 21×21 grid, $\mathbf{\Sigma}_{\mathcal{G}\setminus\mathbf{s}}$ is 440×440, which is larger than the $(n+pk) \times (n+pk)$ matrices involved in our algorithm. In our experience, a coarse 11×11 grid for $\mathcal{G}$ takes roughly the same computation time as our algorithm using (3.2) with $|\mathcal{S}_g| = 31 \times 31$. Another feature of MI is that the objective function (2.19) does not directly use the information $\mathbf{y}_{1:n}$. The values of the hyperparameters depend on $\mathbf{y}_{1:n}$, but the objective function only depends on the sampling locations and not on the observed values so far, nor the possible observed values in the future. This means

that with fixed hyperparameters, all simulations would have produced the same optimal trajectory. This indicates that MI is not very adaptive to data, and is better suited for offline trajectory planning.

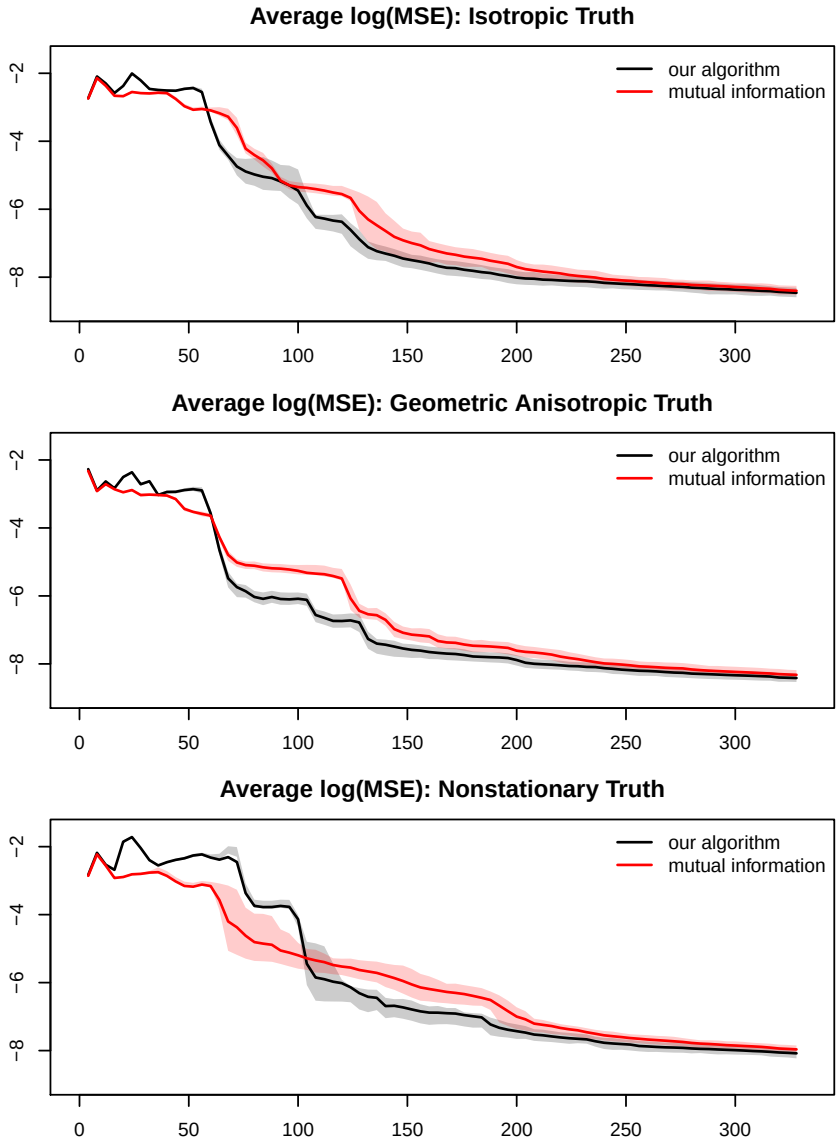


Figure 3.7: Comparing log(MSE) of our algorithm to one that uses mutual information. The shaded area indicates the 25th-75th percentile.

3.4.2 Results and comparisons: finding the maximum

We run another set of simulations in which we evaluate the ability of our algorithm to identify the location of the global maximum of the field. For each of the 200 simulations in each of the six cases, the algorithm is run for 200 iterations, resulting optimal trajectories consisting of 200 local trajectory segments and a total flight time of 800Δ.

Figure 3.8 shows snapshots taken at iteration $j = 83$ of typical trajectories obtained with the exponential and anisotropic Matérn covariance functions for estimating each of the three field types. The first two columns contain the estimate of the field and the associated variance, while the third column shows the expected improvement. Unlike the optimal reconstruction examples, the generated trajectories when using the reward function given in (3.3) do not attempt to cover the region of interest. Instead, they tend to find and circle the local maximum a few times and then head toward and find the global maximum. The exception is the case of the exponential covariance estimating the nonstationary field, where the trajectory does not leave the local maximum. Nonetheless, the expected improvement is also high in the unexplored region of the space, illustrating the trade-off between exploration and exploitation. We also note that the posterior mean surfaces computed using either the exponential or Matérn covariance function capture most of the features of the true field, although the contours of the Matérn examples are closer to the truth. Moreover, in the Matérn examples, the trajectories seem to concentrate less at the two peaks and explore more of the region.

A summary of the results for all 6×200 simulations locating the maximum is presented in Figure 3.9, which shows the probability that the trajectories generated by our algorithm have found the global maximum as a function of time.

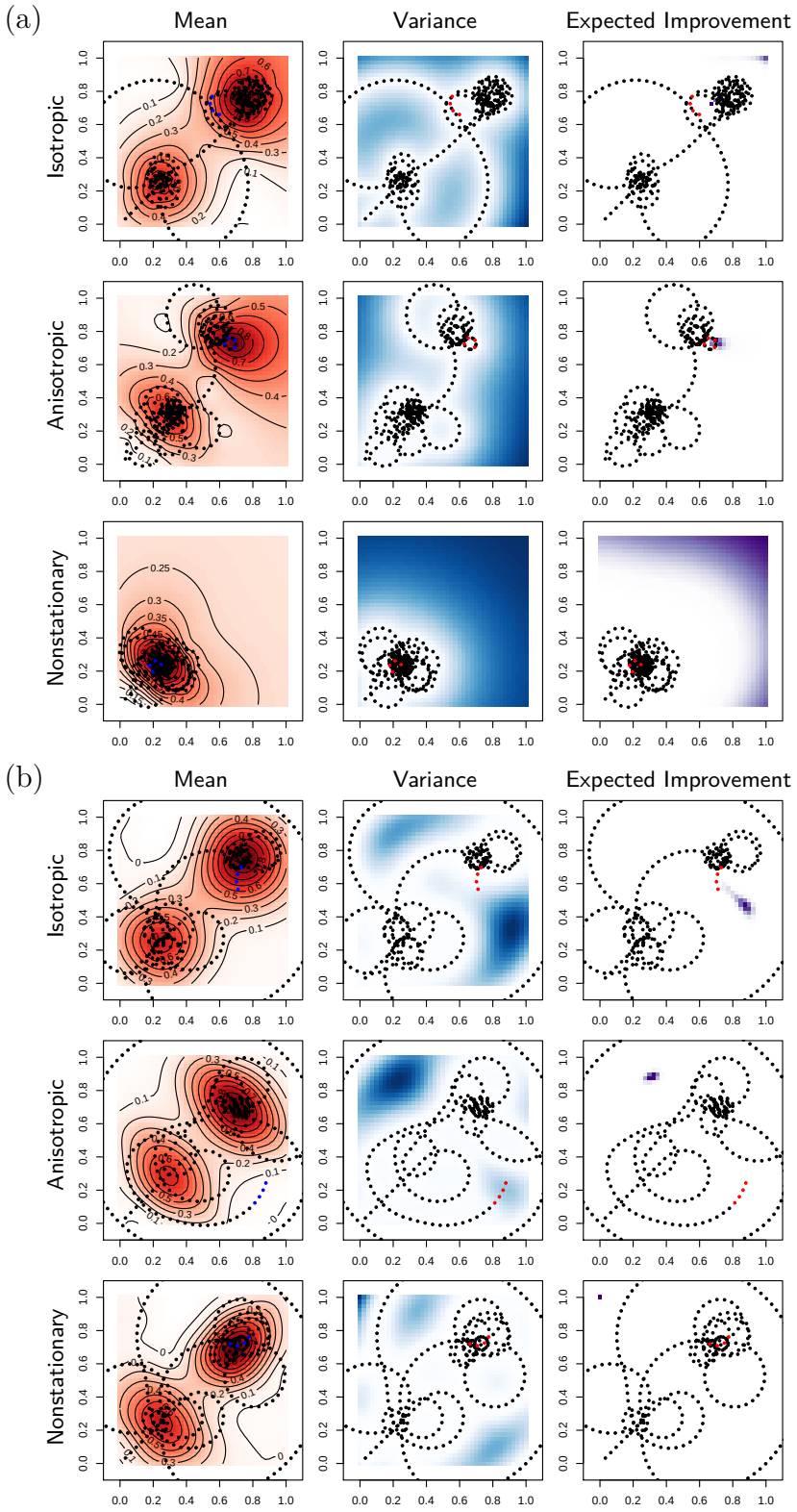


Figure 3.8: Trajectory snapshots at iteration $j = 83$ of the six different cases using (a) exponential covariance and (b) Matérn covariance.

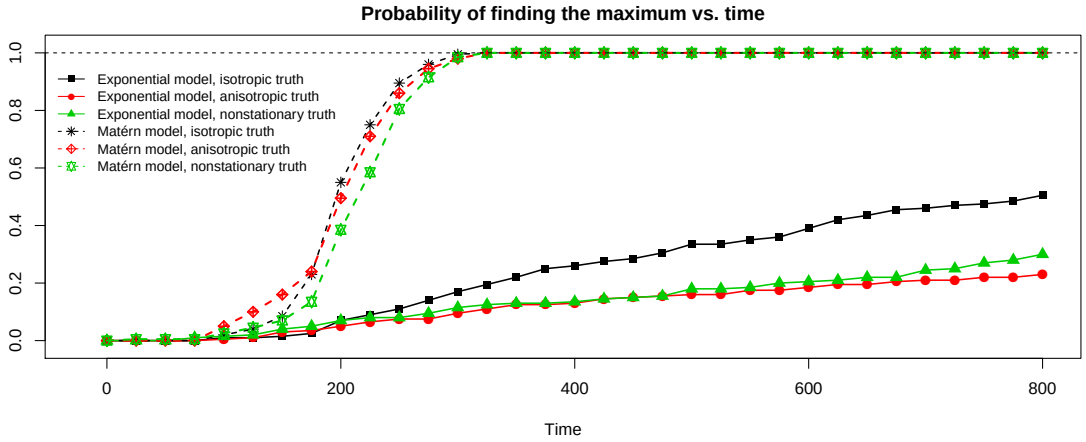


Figure 3.9: The proportion of our simulations that have located the global maximum as a function of time.

Based on the resolution for computing the estimate of the field, we determine that the global maximum has been found if the location of the maximum of the posterior mean is inside an ε-disk centered at the true global maximum's location, and preset $\varepsilon = 0.025\sqrt{2}$. The algorithm is more effective at locating the maximum of the isotropic truth, especially when using the exponential covariance, which is isotropic, compared to the other two truths. The performance for all three truths using the Matérn covariance is very close and show more exploratory trajectories and better estimation than using the exponential covariance. This can largely be explained by the Matérn covariance function's additional smoothness parameter ν. From equation (3.11) the true fields have smooth peaks that rapidly decay. This smoothness of the peaks can be captured by the estimate of ν whereas the exponential covariance is inflexible and rather unsmooth.

Controlling the exploration-exploitation trade-off

In Section 2.2.2 we discussed the ability to affect the exploration-exploitation trade-off by tuning c in $\mathrm{E}\left(\max\{f(\mathbf{s}) - cy_{1:n}^{\max}, 0\}\right)$. The previous results are with $c = 1$. To see the effect of tuning c, an additional set of Monte Carlo simulations

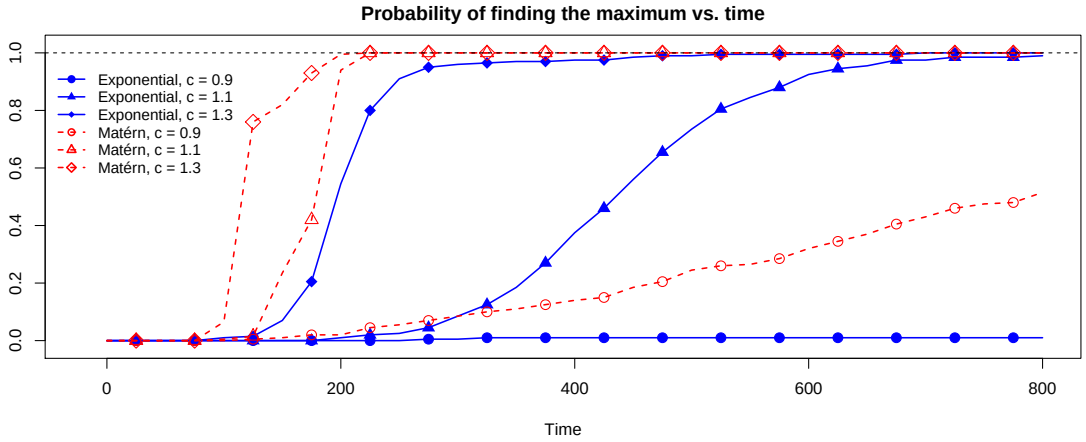


Figure 3.10: Controlling the exploration-exploitation trade-off.

with the anisotropic field as the truth are run with varying values of c. Figure 3.10 shows the effect of smaller and larger values of c under the exponential and Matérn covariance models. We see that increasing the value of c raises the probability curves, so the global maximum is found sooner as c increases. Hence if we are interested in exploring the field more aggressively, we could set c to be slightly greater than one. The results are in line with the analysis presented at the end of Section 2.2.2. Note that choosing $c > 1$ is especially helpful in the exponential case.

Comparison with variable speed and biased random walk

A third set of simulations are compared, where we run additional 200 simulations using Algorithm 3 (variable speed circular arcs) and 200 simulations using a biased random walk strategy[5]. Figure 3.11 shows snapshots of sample trajectories obtained from using Algorithm 2, Algorithm 3, and the biased random walk after 224 measurements have been taken. We see that the trajectories generated by

[5]The biased random walk chooses the next radius $\xi_{j,1}$ randomly from uniformly distributed curvature. If the resulting arc is climbing a gradient, the vehicle follows the same arc for $0.25k$ more sampling locations, chosen to comply with the bias of 25% used in [24].

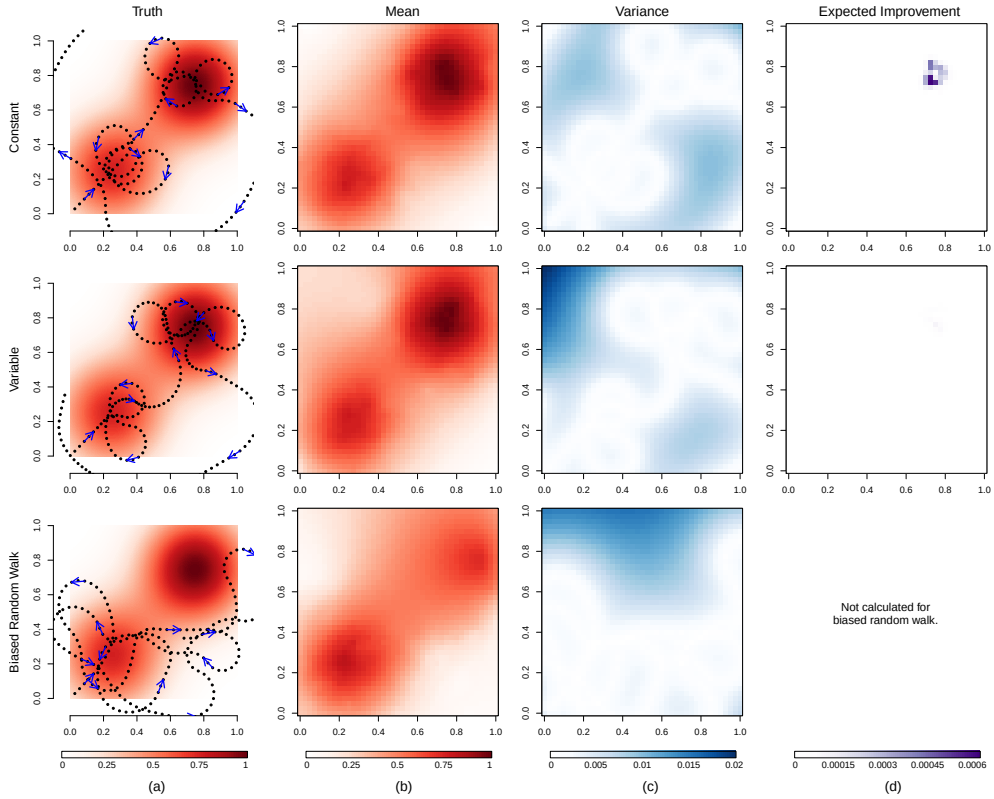


Figure 3.11: Examples from using constant speed, variable speed, and biased random walk algorithms.

our constant and variable speed algorithms find and circle a few times the local maximum and then head toward and find the global maximum. However, the biased random walk trajectory so far has stayed mostly in the lower half of $\mathcal{S}$ and has not gotten near enough to the global maximum to estimate the field or the location of the global maximum accurately.

Figure 3.12 shows the probability of finding the global maximum as a function of flight time for the three algorithms. In both constant and variable speed cases, our algorithm seems to be able to find the global maximum with high probability more quickly than the biased random walk strategy. Moreover, the curves for constant and variable speed are very close. We show in Figure 3.13 a representative path for the optimal speed resulting from Algorithm 3. The vehicle goes through

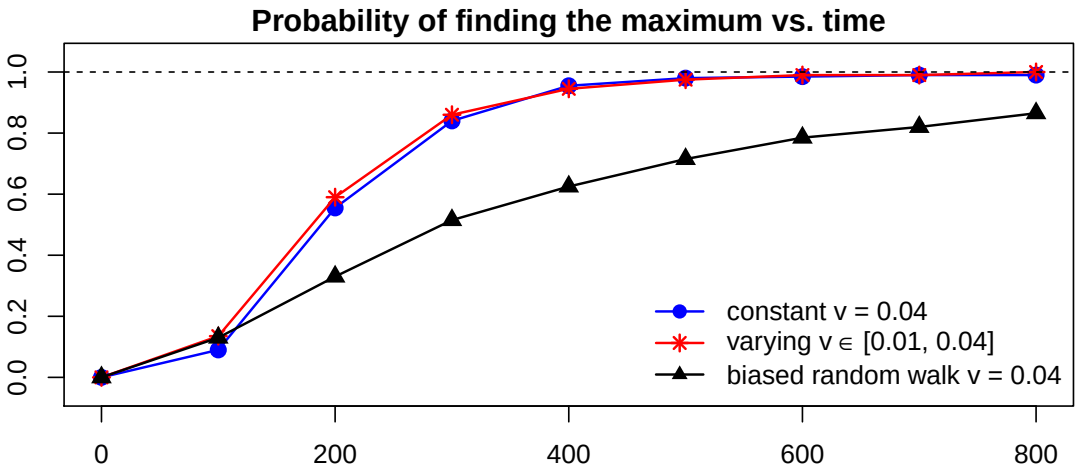


Figure 3.12: The proportion of our simulations that have located the global maximum as a function of time.

periods of flying at maximum speed, slowing down, and going back to maximum speed. Moreover, the vehicle tends to fly at maximum speed when it moves away from the currently estimated maximum and it tends to slow down when it approaches and flies over a local maximum. The result that the two algorithms seem to perform very similarly and the fact that the two-dimensional optimization associated with the variable speed algorithm is computationally expensive suggest that the constant speed algorithm is better suited for online, real-time trajectory generation for a physical system.

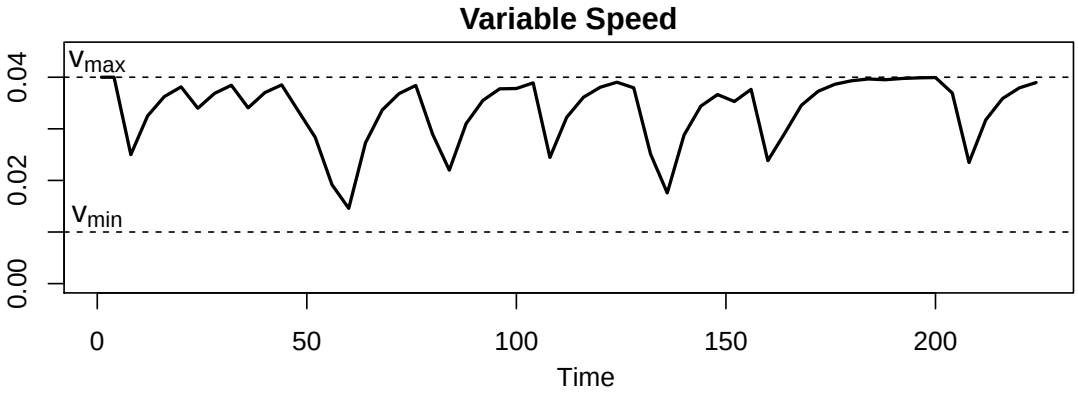


Figure 3.13: An example of the speed path under Algorithm 3.

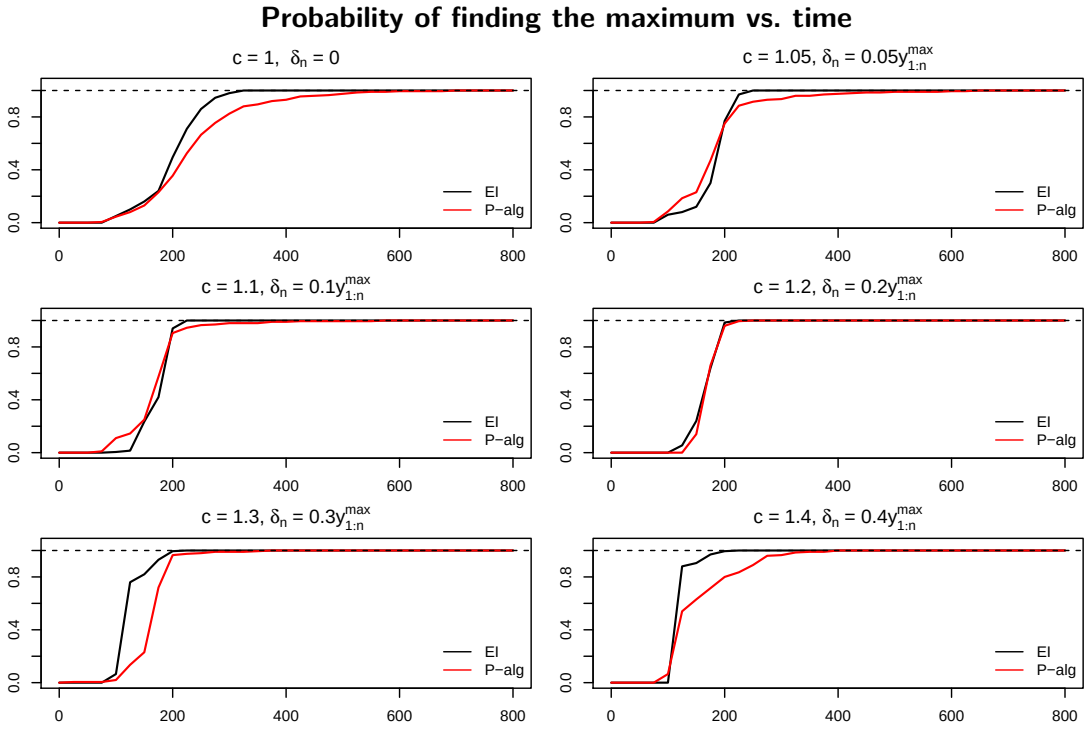


Figure 3.14: Comparing expected improvement to the P-algorithm.

Comparison with P-algorithm

For final comparison with the goal of locating the maximum of f, we implement the constant speed algorithm using the P-algorithm (2.21) replacing the expected improvement as the utility (and similarly extending it to the segment $\mathbf{q}_j$). The P-algorithm contains an exploration-exploitation tuning parameter δ_n. We set $\delta_n = [1-c]y_{1:n}^{\max}$ for a direct comparison to our use of the generalized expected improvement earlier this section. All simulation parameters, $v, R_{\min}, \alpha, k, p, \Delta$, are kept the same. Figure 3.14 shows the probability of locating the maximum of the anisotropic truth using the Matérn covariance, when using the P-algorithm with varying δ_n. As a baseline with $\delta_n = 0$, the P-algorithm is equivalent to maximizing the probability of improvement, and performs worse than maximizing the expected improvement. For the smaller $\delta_n > 0$ values, the P-algorithm

and expected improvement perform similarly, suggesting that the exploration-exploitation parameters in both are close to being optimally tuned. For higher δ_n, the P-algorithm seems to be too aggressive in exploration, and performance in locating the global maximum drops.

3.4.3 Time to run one simulation using Algorithm 2

Table 3.3 shows the number of minutes to run a simulation coded in C on an 1.3 GHz Intel Core i5 laptop, for both operational goals and covariance functions. Additional simulation studies using OpenMP (results not shown here) to speed up the two maximizations (lines 6 and 7 in Algorithm 2) show that run times for optimal reconstruction simulations can be further reduced by a factor of four.

goal	grid	covariance	length	run time (minutes)
optimal reconstruction	31×31	exponential	332Δ	6.5
optimal reconstruction	31×31	Matérn	332Δ	13
find maximum		exponential	800Δ	6
find maximum		Matérn	800Δ	35

Table 3.3: Simulation run times.

Figure 3.15 shows the total run time (black lines) per iteration j, as well as the breakdown of time spent estimating $(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi})$ (red dashes) and segment optimization (blue lines) corresponding to lines 6 and 7 in Algorithm 2. After iteration $j = 40$, we stop updating the hyperparameter estimates every iteration and only update every ten iterations, using the full data history. In every goal scenario, the computation time increases polynomially, as expected when the computations largely involve matrix multiplications and inversions. This is means that real-time implementations of Algorithm 2 are not feasible unless the time between measurement locations is sufficiently long. Methods to reduce computation time

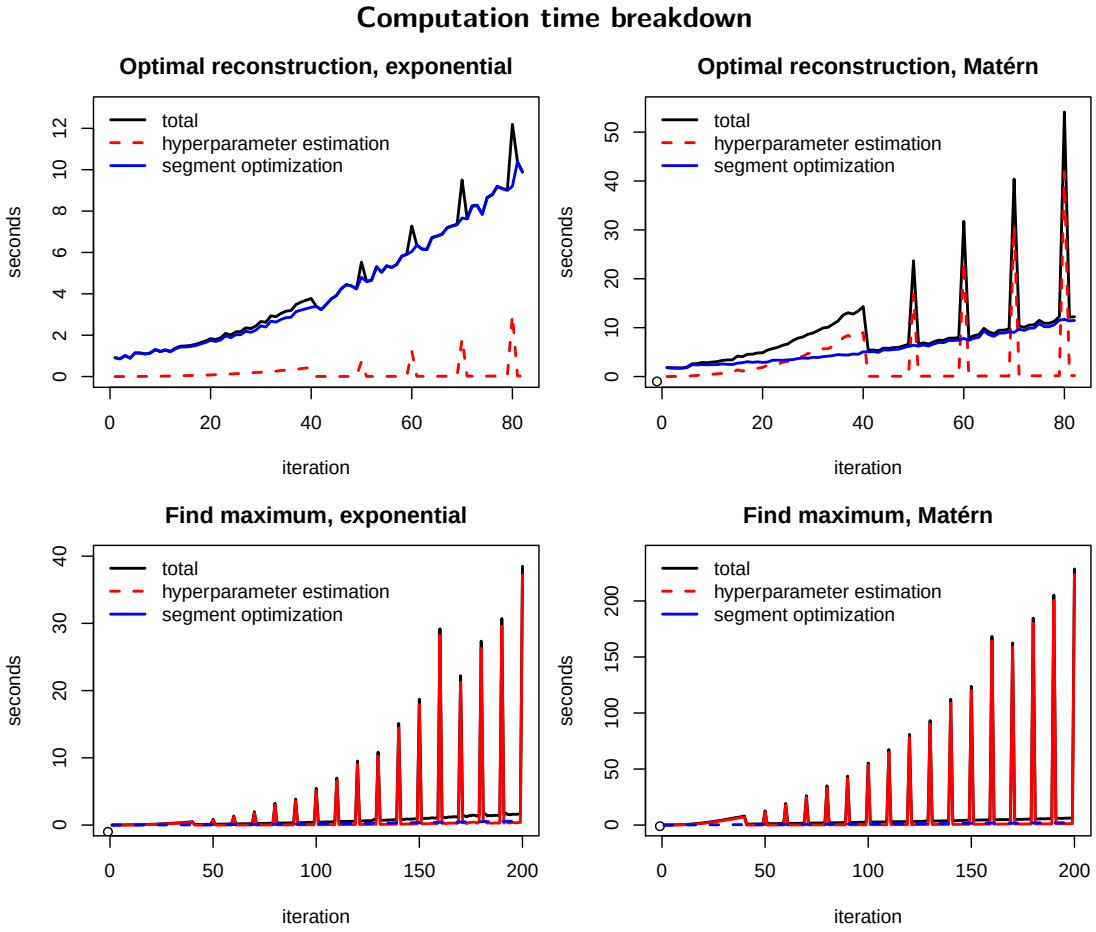
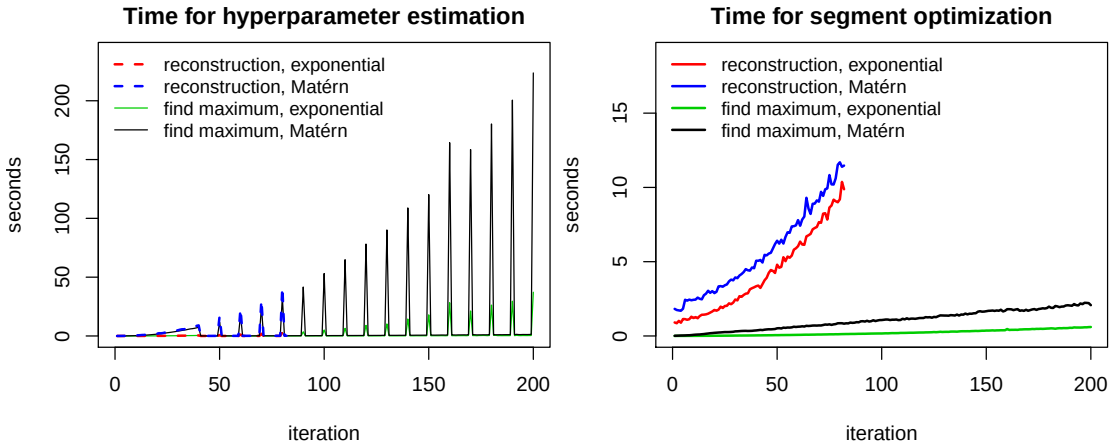


Figure 3.15: Computation time per iteration, compared across operational goals and covariance functions.

for real-time implementation are discussed in the future work section of Chapter 6.

Figure 3.16 shows the computation times for each goal scenario on the same plot, broken down by task. During the hyperparameter estimation task, the computation time is unaffected by the exploration goal, demonstrated by the coinciding values of the red dash and green line and those of the blue dash and black line. The main takeaway from Figure 3.16a is that the time required to compute hyperparameters of the Matérn covariance function $(\sigma^2, \tau^2, \nu, \lambda_1, \lambda_2, \vartheta)$ is approximately tenfold that of computing those of the exponential covariance function

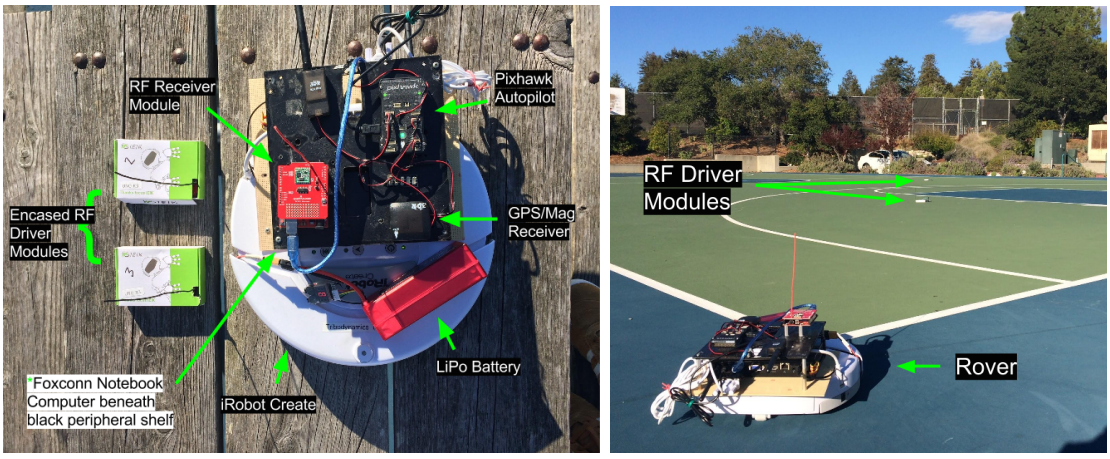


(a) **(b)**

Figure 3.16: Comparison of computation time by task.

$(\sigma^2, \tau^2, \lambda)$. This is primarily explained by the addition of three[6] more parameters, allowing for flexible smoothness and anisotropy in the Gaussian process model, and a more complicated posterior density (2.10) for which we find the maximizer. Figure 3.16b shows that for calculating the next optimal segment, the two covariance functions result in similar times, where the difference is due to the covariance matrix entries take slightly more time to evaluate when using the Matérn. Another observation from Figures 3.15 and 3.16 is that planning the next trajectory segment is cheapest (by a large margin) for finding the maximum, planning the next segment for optimal reconstruction takes more time, and hyperparameter estimation is the most expensive (and uses proportionately more time during finding the maximum) computation.

[6]This number can be reduced two, if the ratio τ^2/σ^2 is estimated instead of separately estimating σ^2 and τ^2, and is given an F-distribution prior.



(a) **(b)**

Figure 3.17: (a) The hardware components of the demonstration platform, consisting of two RF drivers, the rover, and onboard instruments. (b) The RF drivers generating a "radio plume" and the rover in starting position.

3.5 Experimental validation

We report the results from trials of our trajectory planning algorithm on an autonomous ground vehicle, demonstrating a proof-of-concept on the field beyond software simulation. The demonstration platform consists of a mobile vehicle and a network of static radio frequency (RF) driver modules broadcasting RF signals. The vehicle, an iRobot Create skid-steered rover with tank turn disabled to emulate a fixed speed Dubins vehicle, is equipped with a computer, autopilot, GPS/magnetometer, and an RF sensing unit. The magnitudes of received RF values from each static RF transmitter are summed to create a spatial "radio plume," which is the environmental process of interest, over a 15 meter by 15 meter region. The goal of the trials is to locate the maximum of the RF field. Figure 3.17 shows the hardware components of our demonstration platform. A thorough description of the demonstration platform as well as an earlier version of the results in this section can be found in [35].

Once the hardware components are in their starting configuration as in Fig-

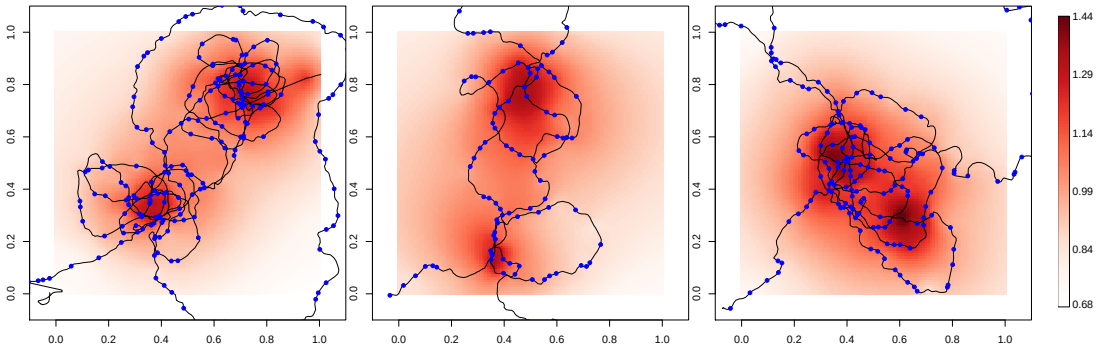


Figure 3.18: The resulting rover trajectories (black continuous curves) and sample locations (blue points) for locating the maximum of the "radio plume" for three different placements of the RF drivers.

ure 3.17(b), the trials are automated by a central Python script that facilitates all the communication among the software components: the trajectory planning algorithm, Mission Planner waypoint software, the RF sensor port, and the rover controls. The trajectory planning algorithm, which uses an exponential covariance function for the Gaussian process prior because we expect the RF field to be isotropic, calculates the next optimal trajectory segment $\mathbf{q}^*_{j+1}$ and passes the next k sampling locations $\mathbf{q}^*_{j+1}(\Delta), \ldots, \mathbf{q}^*_{j+1}(k\Delta)$ to Mission Planner as waypoints to be visited. Mission planner, along with the autopilot, computes and passes the rover control outputs to the motors, driving the rover to the instructed waypoints. Mission planner also determines if the waypoints have been reached and passes the coordinates of the locations that trigger as "waypoint hit" (which do not match exactly the instructed waypoint coordinates), to the trajectory planning algorithm to be processed as the actual locations where RF data is collected.

Figure 3.18 shows the trajectories of the rover overlaid on the estimate of the RF field for three separate trials with different initial placements of the static RF transceivers creating different plume shapes. There are a couple issues with the resulting trajectories in our experiment: the sequential sample locations on the trajectories are not equidistant, and the trajectories themselves are not as smooth

as what is expected from circular arcs stitched together. The former is partially due to Mission Planner's registration of a "waypoint hit" to be true if the rover is within one meter, the lowest possible setting, of the instructed waypoint's location. The extra windiness of the paths traveled by the rover is partially caused by latency between the communication, registration, and execution of the motor control values. Both issues are significantly exacerbated by GPS/magnetometer sensing errors, especially at the small spatial scale of the experiments. While these discrepancies from the theoretically-computed ideal trajectories are always present, they would be mitigated in applications where the region is larger and the vehicle is longer endurance. Figure 3.18 shows that, despite these known discrepancies, the trajectories produced by our algorithm in a field test look similar to those in the simulations of Section 3.4 and are able to locate the global maximum. This similarity suggests some robustness of the trajectory planning algorithm to disturbances inherent in real world vehicle behavior.

Chapter 4

Multiple Sources of Information

The procedure discussed in Chapter 3 assumes that all data available about the environmental process comes from measurements collected by the autonomous vehicle. However, there are many situations in which supplementary information that could improve the performance of the algorithm is readily available. For example, data about the process might be available from fixed sensors located within the region of interest $\mathcal{S}$, or from other remote sensing technologies such as satellite imagery. Alternatively, an educated guess about the structure of f might be available from a computer model. While incorporating any of these sources of information will increase the efficiency of the algorithm, particularly in the early stages of the exploration process, doing so properly requires subtle, situation-aware modifications of the model described in 2.1 in order to generate data fusion algorithms that properly account for the uncertainties associated with different types of supplementary sources. This chapter discusses and evaluates such modifications.

The remainder of the chapter is organized as follows. Section 4.1 introduces new notation and any notation modifications from the previous chapters. Sections 4.2, 4.3, and 4.4 detail three procedures to include supplementary informa-

tion sources into the control algorithm. Section 4.5 illustrates the performance of the algorithms in optimal reconstruction of different fields with different sources of supplementary information. The chapter is concluded with a discussion in Section 4.6.

4.1 Notation

From Chapters 2 and 3, we keep the notation of $\mathbf{y}_{1:n} = (y_1, \ldots, y_n)$ representing the vector of n observations taken at the locations $\{\mathbf{s}_1^y, \ldots, \mathbf{s}_n^y\}$ on the trajectory $\mathbf{Q}$ so far. We modify the notation of the vectors and matrices associated with $\mathbf{y}_{1:n}$ by replacing the subscript $1:n$ by a subscript y; so that $\mathbf{c}_{1:n}(\mathbf{x})$ becomes $\mathbf{c}_y(\mathbf{x})$, $\mathbf{m}_{1:n}$ becomes $\mathbf{m}_y$, and $\mathbf{C}_{1:n}$ becomes $\mathbf{C}_{yy}$. For the supplementary information, we introduce the vector $\mathbf{z} = (z_1, \ldots, z_r)$ and the associated locations $\{\mathbf{s}_1^z, \ldots, \mathbf{s}_r^z\}$. We assume that all of the supplementary information is contained in $\mathbf{z}$ and its locations, and $\mathbf{z}$ is available to the vehicle before any trajectory planning occurs. Thus r, the size of $\mathbf{z}$ is fixed while $\mathbf{y}_{1:n}$ grows.

In this chapter, the objective function to be maximized in order to design the next vehicle trajectory segment $\mathbf{q}_{j+1}^*$ for optimal reconstruction is based on

$$\tilde{U}_{j+1}(\mathbf{q}_{j+1}) = -\sum_{\mathbf{x}\in\mathcal{S}_g} \left\{ \left[\mu_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x}) - \mu_{\mathbf{z},\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x}) \right]^2 + \kappa^2_{\mathbf{z},\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x}) \right\}, \quad (4.1)$$

where

$$\begin{aligned}
\mu_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x}) &= \mathrm{E}\left\{ f(\mathbf{x}) \mid \mathbf{z}, \mathbf{y}_{1:n} \right\} \\
\mu_{\mathbf{z},\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x}) &= \mathrm{E}\left\{ f(\mathbf{x}) \mid \mathbf{z}, \mathbf{y}_{1:n}, \hat{\mathbf{y}}_{\mathbf{q}_{j+1}} \right\} \\
\kappa^2_{\mathbf{z},\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x}) &= \mathrm{Var}\left\{ f(\mathbf{x}) \mid \mathbf{z}, \mathbf{y}_{1:n}, \hat{\mathbf{y}}_{\mathbf{q}_{j+1}} \right\}
\end{aligned}$$

are now also conditioned on $\mathbf{z}$.

4.2 Incorporating information from point-referenced data sources

The simplest form of supplementary data for trajectory planning comes in the form of point-referenced measurements taken at a set of pre-specified locations within $\mathcal{S}$. This type of data might arise, for example, from a network of fixed monitoring stations. These networks are common for many weather related environmental processes (e.g., temperature, ozone, and particulate matter concentration), but are often sparse and provide only relatively low resolution estimates of the field they are designed to monitor.

Extending our model to accommodate supplementary sources of point-reference measurements is straightforward. In particular, let $\mathbf{z} = (z(\mathbf{s}_1^z), \ldots, z(\mathbf{s}_r^z))$ denote the observations arising from r fixed monitoring stations located at positions $\mathbf{s}_1^z, \ldots, \mathbf{s}_r^z$. Similar to before in Section 2.1, we take

$$z_l = z(\mathbf{s}_l^z) = f(\mathbf{s}_l^z) + \varepsilon_l^z, \quad y_i = y(\mathbf{s}_i^y) = f(\mathbf{s}_i^y) + \varepsilon_i^y, \quad l = 1, \ldots, r, \quad i = 1, \ldots, n,$$

where $\varepsilon_l^z \sim \mathsf{N}(0, \sigma_z^2)$ and $\varepsilon_i^y \sim \mathsf{N}(0, \sigma_y^2)$ independently for every l and i. This model implicitly assumes that the sensors on both the monitoring network and the autonomous vehicle measure the same underlying field (although using potentially different instruments, leading to potentially different levels of measurement noise). When combined with our Gaussian process prior for f, $f \mid \boldsymbol{\varphi}, \boldsymbol{\psi} \sim$

$\mathcal{GP}(m(\cdot\,;\boldsymbol{\varphi}), C(\cdot\,,\cdot\,;\boldsymbol{\psi}))$, this observational model implies that

$$\begin{bmatrix}\mathbf{z}\\ \mathbf{y}\end{bmatrix} \;\Big|\; \sigma_z^2, \sigma_y^2, \boldsymbol{\varphi}, \boldsymbol{\psi} \sim \mathsf{N}\left(\begin{bmatrix}\mathbf{m}_z\\ \mathbf{m}_y\end{bmatrix}, \begin{bmatrix}\mathbf{C}_{zz} & \mathbf{C}_{zy}\\ \mathbf{C}_{zy}^\mathsf{T} & \mathbf{C}_{yy}\end{bmatrix} + \begin{bmatrix}\sigma_z^2\mathbf{I}_r & \mathbf{0}\\ \mathbf{0} & \sigma_y^2\mathbf{I}_n\end{bmatrix}\right), \tag{4.2}$$

where the mean vectors $\mathbf{m}_z$ and $\mathbf{m}_y$ are given by

$$\mathbf{m}_z = (m(\mathbf{s}_1^z;\boldsymbol{\varphi}), \ldots, m(\mathbf{s}_r^z;\boldsymbol{\varphi})), \qquad \mathbf{m}_y = (m(\mathbf{s}_1^y;\boldsymbol{\varphi}), \ldots, m(\mathbf{s}_n^y;\boldsymbol{\varphi})),$$

and the blocks $\mathbf{C}_{zz}$, $\mathbf{C}_{zy}$, and $\mathbf{C}_{yy}$ satisfy

$$\begin{aligned}[\mathbf{C}_{zz}]_{l,l'} &= C(\mathbf{s}_l^z, \mathbf{s}_{l'}^z; \boldsymbol{\psi}),\\ [\mathbf{C}_{zy}]_{l,i} &= C(\mathbf{s}_l^z, \mathbf{s}_i^y; \boldsymbol{\psi}),\\ [\mathbf{C}_{yy}]_{i,i'} &= C(\mathbf{s}_i^y, \mathbf{s}_{i'}^y; \boldsymbol{\psi}),\end{aligned}$$

for $l, l' = 1, \ldots, r$ and $i, i' = 1, \ldots, n$. The rest of the results from Section 2.1.2 can be similarly extended. For example, the posterior distribution of the value of f at any $\mathbf{x} \in \mathcal{S}$ is $f(\mathbf{x}) \mid \mathbf{z}, \mathbf{y}_{1:n}, \sigma_z^2, \sigma_y^2, \boldsymbol{\varphi}, \boldsymbol{\psi} \sim \mathsf{N}\left(\mu_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x}), \kappa^2_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x})\right)$, where

$$\mu_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x}) = m(\mathbf{x};\boldsymbol{\varphi}) + \begin{bmatrix}\mathbf{c}_z^\mathsf{T}(\mathbf{x}) & \mathbf{c}_y^\mathsf{T}(\mathbf{x})\end{bmatrix} \begin{bmatrix}\mathbf{C}_{zz} + \sigma_z^2\mathbf{I}_r & \mathbf{C}_{zy}\\ \mathbf{C}_{zy}^\mathsf{T} & \mathbf{C}_{yy} + \sigma_y^2\mathbf{I}_n\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{z} - \mathbf{m}_z\\ \mathbf{y} - \mathbf{m}_y\end{bmatrix}, \tag{4.3}$$

and

$$\kappa^2_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x}) \;=\; \tau^2 \;-\; \begin{bmatrix}\mathbf{c}_z^\mathsf{T}(\mathbf{x}) & \mathbf{c}_y^\mathsf{T}(\mathbf{x})\end{bmatrix} \begin{bmatrix}\mathbf{C}_{zz} + \sigma_z^2\mathbf{I}_r & \mathbf{C}_{zy}\\ \mathbf{C}_{zy}^\mathsf{T} & \mathbf{C}_{yy} + \sigma_y^2\mathbf{I}_n\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{c}_z(\mathbf{x})\\ \mathbf{c}_y(\mathbf{x})\end{bmatrix}. \tag{4.4}$$

In the previous expressions the vectors $\mathbf{c}_z(\mathbf{x})$ and $\mathbf{c}_y(\mathbf{x})$ are defined as

$$\mathbf{c}_z(\mathbf{x}) = (C(\mathbf{s}_1^z, \mathbf{x}; \boldsymbol{\psi}), \ldots, C(\mathbf{s}_r^z, \mathbf{x}; \boldsymbol{\psi})),$$
$$\mathbf{c}_y(\mathbf{x}) = (C(\mathbf{s}_1^y, \mathbf{x}; \boldsymbol{\psi}), \ldots, C(\mathbf{s}_n^y, \mathbf{x}; \boldsymbol{\psi})).$$

For hyperparameter estimation, we use the same empirical Bayes method described in Section 2.1.2, modified to include σ_z^2 and the data $\mathbf{z}$:

$$\left(\widehat{\sigma}_{z,n}^2, \widehat{\sigma}_{y,n}^2, \widehat{\boldsymbol{\varphi}}_n, \widehat{\boldsymbol{\psi}}_n\right) = \underset{\left(\sigma_z^2, \sigma_y^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\right)}{\arg\max} \left[\log p\left(\mathbf{z}, \mathbf{y}_{1:n} \mid \sigma_z^2, \sigma_y^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\right) + \log p\left(\sigma_z^2, \sigma_y^2, \boldsymbol{\varphi}, \boldsymbol{\psi}\right) \right], \quad (4.5)$$

and $\boldsymbol{\varphi}$ can similarly be estimated separately if we take a constant mean function $m(.; \boldsymbol{\varphi}) = \varphi$ (see the Appendix for details).

Implementation of the objective function for optimal reconstruction is the same as in Section 3.1. To go from $\mu_{\mathbf{y}_{1:n}, \hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ to $\mu_{\mathbf{z}, \mathbf{y}_{1:n}, \hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$, and $\kappa^2_{\mathbf{y}_{1:n}, \hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ to $\kappa^2_{\mathbf{z}, \mathbf{y}_{1:n}, \hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ replace $\mathbf{c}_{1:n}(\mathbf{x})$ and $\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n$ by

$$\begin{bmatrix} \mathbf{c}_z(\mathbf{x}) \\ \mathbf{c}_y(\mathbf{x}) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{C}_{zz} + \sigma_z^2 \mathbf{I}_r & \mathbf{C}_{zy} \\ \mathbf{C}_{zy}^\mathsf{T} & \mathbf{C}_{yy} + \sigma_y^2 \mathbf{I}_n \end{bmatrix},$$

respectively, wherever they occur in the expressions (A.30) and (A.31).

4.3 Incorporating information from aggregated data sources

Not all remote sensors provide point-referenced measurements. For example, in satellite imagery, the data collected for each pixel usually corresponds to an

average of the true underlying field over the area covered by the pixel. More generally, when the resolution of the supplementary sensor is much lower than the resolution of the sensor on the autonomous vehicle, it is often more accurate to think about the supplementary sensor as measuring a (potentially weighted) average of the underlying field over each pixel. While it would be tempting to ignore this fact and use the model from Section 4.2 as if observations were point-referenced to the center of each pixel, such an approach ignores the (potentially important) effect of the averaging process. In this section we discuss an alternative modeling approach that acknowledges the aggregate nature of the supplementary data, and illustrate the theoretical differences with that in Section 4.2.

Assume the region $\mathcal{S}$ has been exhaustively partitioned into r non-overlapping pixel regions $S_1, \ldots S_r$, and let z_l be the measurement associated with pixel l coming from a supplementary, low-resolution image sensor. As before, we assume that observations collected by the autonomous vehicle satisfy $y(\mathbf{s}_i^y) = f(\mathbf{s}_i^y) + \varepsilon_i^y$ with $\varepsilon_i^y \sim \mathsf{N}(0, \sigma_y^2)$. However, for the observations generated by the image sensor, we assume

$$z_l = g_l + \varepsilon_l^z, \quad \varepsilon_l^z \sim \mathsf{N}(0, \sigma_z^2)$$

where

$$g_l = \frac{1}{|S_l|} \int_{S_l} f(\mathbf{x}) \, \mathrm{d}\mathbf{x},$$

and $|S_l| = \int_{S_l} \mathrm{d}\mathbf{x}$ is the area of S_l. If f is assigned a Gaussian process prior, then the quantities $g_1, \ldots, g_r$ are inherently random variables in the same way that $f(\mathbf{s}_1^y), \ldots, f(\mathbf{s}_n^y)$ are random, and they are normally distributed. To construct our predictive model and implement the control algorithm, we must obtain the joint distribution of the vectors $\mathbf{g} = (g_1, \ldots, g_r)$ and $\mathbf{f} = (f(\mathbf{s}_1^y), \ldots, f(\mathbf{s}_n^y))$, which in

this case is known to follow a multivariate normal distribution with

$$\begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} \Bigg| \, \boldsymbol{\varphi}, \boldsymbol{\psi} \sim \mathsf{N}\left(\begin{bmatrix} \mathbf{m}_z \\ \mathbf{m}_y \end{bmatrix}, \begin{bmatrix} \mathbf{C}_{zz} & \mathbf{C}_{zy} \\ \mathbf{C}_{zy}^{\mathsf{T}} & \mathbf{C}_{yy} \end{bmatrix} \right), \tag{4.6}$$

where

$$\begin{aligned} \mathbf{m}_z &= \left(\frac{1}{|S_1|} \int_{S_1} m(\mathbf{x}; \boldsymbol{\varphi}) \, \mathrm{d}\mathbf{x}, \ldots, \frac{1}{|S_r|} \int_{S_r} m(\mathbf{x}; \boldsymbol{\varphi}) \, \mathrm{d}\mathbf{x} \right), \\ \mathbf{m}_y &= \left(m(\mathbf{s}_1^y; \boldsymbol{\varphi}), \ldots, m(\mathbf{s}_n^y; \boldsymbol{\varphi}) \right), \end{aligned}$$

and the blocks $\mathbf{C}_{zz}$, $\mathbf{C}_{zy}$, and $\mathbf{C}_{yy}$ satisfy

$$[\mathbf{C}_{zz}]_{l,l'} = \frac{1}{|S_l||S_{l'}|} \int_{S_l} \int_{S_{l'}} C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{x}', \tag{4.7}$$

$$[\mathbf{C}_{zy}]_{l,i} = \frac{1}{|S_l|} \int_{S_l} C(\mathbf{x}, \mathbf{s}_i^y; \boldsymbol{\psi}) \, \mathrm{d}\mathbf{x}, \tag{4.8}$$

$$[\mathbf{C}_{yy}]_{i,i'} = C(\mathbf{s}_i^y, \mathbf{s}_{i'}^y; \boldsymbol{\psi})$$

for $l, l' = 1, \ldots, r$ and $i, i' = 1, \ldots, n$. While computing (4.7) and (4.8) in closed form for general covariance functions can be difficult, the calculation is possible for some commonly used covariance functions. For example, for a rational quadratic covariance function, the integrals are given explicitly in the Appendix. With the modifications to the definitions of $\mathbf{m}_z$, $\mathbf{C}_{zz}$, and $\mathbf{C}_{zy}$ above, the expressions developed for the point-referenced case in Section 4.2 for the distribution in (4.2), posterior mean in (4.3) and variance in (4.4), and hyperparameter estimation in (4.5) all apply directly to the case of aggregated observations.

To illustrate the differences between the approach we just discussed and the point-referenced model in Section 4.2, consider a concrete scenario in which $\mathcal{S} =$

$[0,1]^2$ and define a regular $u \times u$ grid of points located at

$$\mathbf{s}^z_{(l_1-1)u+l_2} = \left(\frac{1}{2u} + \frac{l_1 - 1}{u}, \frac{1}{2u} + \frac{l_2 - 1}{u}\right)$$

for $l_1, l_2 = 1, \ldots, u$, as well as a collection of subsets $A^\delta_1, \ldots, A^\delta_{u^2}$ of $\mathcal{S}$ that correspond to squares with edge length $\delta \leq 1/u$ centered at each of $\mathbf{s}^z_1, \ldots \mathbf{s}^z_{u^2}$. Figure 4.1a presents a representation of this scenario. Then the covariance matrix $\mathbf{C}^\delta_{zz}$ of the joint distribution of

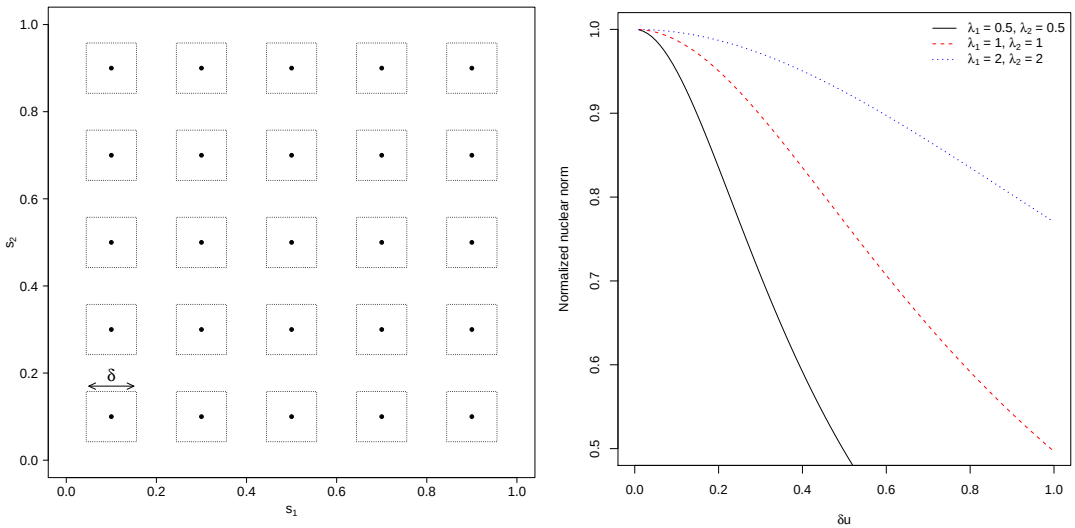
$$\left(\frac{1}{|A^\delta_1|}\int_{A^\delta_1} f(\mathbf{x})\,\mathrm{d}\mathbf{x}, \ldots, \frac{1}{|A^\delta_{u^2}|}\int_{A^\delta_{u^2}} f(\mathbf{x})\,\mathrm{d}\mathbf{x}\right)$$

is equal to that of $\mathbf{C}_{zz}$ described in Section 4.2 when $\delta \to 0$, and to the one discussed in this section when $\delta = 1/u$.

Figure 4.1b presents the value of the normalized nuclear norm of $\mathbf{C}^\delta_{zz}$, defined as $\frac{1}{u^2}\mathrm{tr}\left\{\mathbf{C}^\delta_{zz}\right\}$, as a function of δ for various members of the rational quadratic family of covariance functions in (2.4). Note that the normalized norm decreases monotonically with δ. Hence, using the point-referenced model (which, as we said before, corresponds to $\delta = 0$) for observations that are actually collected under the aggregated model (which corresponds to $\delta = 1/u$) leads to a distribution that is overdispersed with respect to the truth. The larger variance of the point-referenced model means that using it when the true nature of the data is aggregated wastes information. While the impact of this waste depends on the true structure of the field (and how influential the prior is), and the impact will be lessened as the vehicle collects more data, its importance could potentially be large in the early stages of the exploration process.

While the previous discussion suggests that using the prior discussed in this section for aggregated data will lead to a more efficient path planning algorithm,

there is a tradeoff. As we mentioned before, computation of the covariance function for spatially aggregated data in (4.7) can be computationally demanding outside some standard families because the integrals required might not be available in closed form. Hence, if an intractable covariance function is chosen, the application of the approach discussed in this section might be unfeasible in the context of a real-time algorithm.



(a) **(b)**

Figure 4.1: A concrete example of the differences between the point-reference and aggregate data approaches. Panel (a) shows the setup associated with this illustration, which involves a regular $u \times u$ grid on $\mathcal{S} = [0,1]^2$ that is used as the center of a collection of square subregions $\{A^\delta\}$ of $\mathcal{S}$ with edge length δ. Panel (b) shows the normalized nuclear norm of $\mathbf{C}^\delta_{zz}$, the covariance matrix associated with the joint distribution of the average value of the field over each subregion, for various members of the rational quadratic family of covariance functions.

4.4 Incorporating supplementary information through the mean function

Finally, we consider the situation in which we do not have direct observations of the field f beyond those collected by the autonomous vehicle, but instead have access to some other information about the shape of the field. For example, we might have access to a "computer model" of the physical phenomenon being monitored, i.e., a solver for a system of differential equations that describes (a simplified version of) the underlying physics at each location $\mathbf{x} \in \mathcal{S}$. In this kind of situation we do not necessarily expect the output of the computer model at a location $\mathbf{x}$ to be exactly equal to (or noisy measurements of) $f(\mathbf{x})$. However, we do expect that the output of the computer model will be at least somewhat close to the true field. Hence, the most appropriate way to incorporate this information is by using the output of the computer model, denoted as $h(\mathbf{x})$, as the mean function of our Gaussian process prior for f, i.e., setting $m(\mathbf{x}; \boldsymbol{\varphi}) = h(\mathbf{x})$.

The main challenge with this approach is the need to repeatedly evaluate $h(\mathbf{x})$ at each segment of the trajectory (and at other segments during the evaluations of the objective function during segment optimization). In practice, this repeated evaluation at any arbitrary location is infeasible because the evaluation of h often requires the use of time-consuming numerical solvers that are simply too slow for online applications such as ours. As an alternative we propose to approximate $h(\mathbf{x})$ with an emulator $\hat{h}(\mathbf{x})$. More specifically, we propose to pre-evaluate the computer model at a small number of locations $\mathbf{s}_1^h, \ldots, \mathbf{s}_r^h$ to obtain $\mathbf{z} = (h(\mathbf{s}_1^h), \ldots, h(\mathbf{s}_r^h))$, construct an emulator $\hat{h}(\mathbf{x})$ of h by interpolating the values $\mathbf{z} = (h(\mathbf{s}_1^h), \ldots, h(\mathbf{s}_r^h))$, and then set $m(\mathbf{x}; \boldsymbol{\varphi}) = \hat{h}(\mathbf{x})$. Although emulators can be constructed using a variety of nonlinear regression tools, our preference is to use an interpolating

Gaussian process [37, 68] with mean function $m^o(\mathbf{x};\boldsymbol{\varphi}^o)$ and covariance function $C^o(\mathbf{x},\mathbf{x}';\boldsymbol{\psi}^o)$. This leads to an interpolator of the form

$$\hat{h}(\mathbf{x}) = m^o(\mathbf{x};\boldsymbol{\varphi}^o) + \mathbf{c}_z^o(\mathbf{x})\left[\mathbf{C}_{zz}^o\right]^{-1}\left[\mathbf{z}-\mathbf{m}_z^o\right],$$

where $c_z^o(\mathbf{x}) = \left(C^o(\mathbf{x},\mathbf{s}_1^h;\boldsymbol{\psi}^o),\ldots,C^o(\mathbf{x},\mathbf{s}_r^h;\boldsymbol{\psi}^o)\right)$, $\left[\mathbf{C}_{zz}^o\right]_{ij} = C^o(\mathbf{s}_i^h,\mathbf{s}_j^h;\boldsymbol{\psi}^o)$, and $\mathbf{m}_z^o = \left(m^o(\mathbf{s}_1^h;\boldsymbol{\varphi}^o),\ldots,m^o(\mathbf{s}_r^h;\boldsymbol{\varphi}^o)\right)$. The use of this emulator in the context of our planning algorithm is feasible because the most expensive computation, which is of the vector $\left[\mathbf{C}_{zz}^o\right]^{-1}\left[\mathbf{z}-\mathbf{m}_z^o\right]$, required to obtain $\hat{h}(\mathbf{x})$ at an arbitrary location $\mathbf{x}$ can be done offline. To construct the interpolator, the covariance function C^o is chosen to ensure that realizations of the process have at least one continuous derivative (in order to allow the resulting control algorithm to be consistent, recall our discussion at the end of Section 3.3), and the mean function m^o is typically chosen to be a constant. Furthermore, the values of $\boldsymbol{\varphi}^o$ and $\boldsymbol{\psi}^o$ are set by using the same empirical Bayes approach used to estimate $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ in (2.10) using the data $\mathbf{z}$.

To better understand the difference between this approach and the one in Section 4.2 for point-referenced supplementary data, we can compare the conditional distributions of the the field f at the locations where the vehicle collects samples, $\mathbf{f}_y = (f(\mathbf{s}_1^y),\ldots,f(\mathbf{s}_n^y))$, given the vector $\mathbf{z}$, as implied by both models. These conditional distributions summarize the information that the supplementary data $\mathbf{z}$ provides about the field f, and roughly corresponds to the prior to be used by the vehicle when estimating f. For the joint, point-referenced model from Section 4.2, this is

$$\mathbf{f}_y \mid \mathbf{z} \sim \mathsf{N}\left(\mathbf{m}_y + \mathbf{C}_{yz}\left[\mathbf{C}_{zz}+\sigma_z^2\mathbf{I}_r\right]^{-1}\left[\mathbf{z}-\mathbf{m}_z\right], \mathbf{C}_{yy} - \mathbf{C}_{yz}\left[\mathbf{C}_{zz}+\sigma_z^2\mathbf{I}_r\right]^{-1}\mathbf{C}_{zy}\right), \tag{4.9}$$

while for the two-step model in this section it is

$$\mathbf{f}_y \mid \mathbf{z} \sim \mathsf{N}\left(\mathbf{m}_y^o + \mathbf{C}_{yz}^o \left[\mathbf{C}_{zz}^o\right]^{-1} \left[\mathbf{z} - \mathbf{m}_z^o\right], \mathbf{C}_{yy}\right), \tag{4.10}$$

where $\mathbf{m}_y^o = \left(m^o(\mathbf{s}_1^y; \boldsymbol{\varphi}^o), \ldots, m^o(\mathbf{s}_n^y; \boldsymbol{\varphi}^o)\right)$ and $\left[\mathbf{C}_{yz}^o\right]_{ij} = C^o(\mathbf{s}_i^y, \mathbf{s}_j^h; \boldsymbol{\psi}^o)$. Note that the variance in (4.10) is $\mathbf{C}_{yy}$ and not $\mathbf{C}_{yy}^o$.

The expressions in (4.9) and (4.10) are similar, but not identical. One subtle but important difference is that $m^o(\mathbf{x}; \boldsymbol{\varphi}^o)$ and $C^o(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}^o)$ (which are associated with the emulator of the computer model h) are potentially different from $m(\mathbf{x}; \boldsymbol{\varphi}^o)$ and $C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}^o)$ (which are associated with the true underlying field we are trying to reconstruct, f). Another difference is that we have chosen to use an *interpolating* Gaussian process for our emulator, which implies that $\sigma_z^2 = 0$. If we adjust for these two differences, then the mean vectors in (4.9) and (4.10) are identical and the two distributions are centered in the same place. However, even if we use a *smoothing* Gaussian process for our emulator and assume that the priors for both f and h share the same hyperparameters, it is clear that the variance-covariance matrix associated with the two priors are very different. In particular, it is straightforward to see that total variance of the prior in (4.9) (measured, for example, by the nuclear norm, as we did in Section 4.3) is strictly smaller than the variance in (4.10). This means that, while both priors are roughly centered around the same values, the prior associated with the two-step approach we introduced in this section is, by construction, less concentrated (and therefore, less informative) about the underlying field f than the prior from Section 4.2. This behavior is consistent with our original observation that the information provided by a computer model of the physical process is less reliable than that provided by point-referenced sensors measuring the real process.

With the two-step model, the marginal distribution of the observations taken

by the vehicle has the distribution

$$\mathbf{y}_{1:n} \mid \sigma_y^2, \boldsymbol{\varphi}^o, \boldsymbol{\psi}^o, \boldsymbol{\psi} \sim \mathsf{N}\left(\mathbf{m}_y^o + \mathbf{C}_{yz}^o \left[\mathbf{C}_{zz}^o\right]^{-1} \left[\mathbf{z} - \mathbf{m}_z^o\right], \mathbf{C}_{yy} + \sigma_y^2 \mathbf{I}_n\right), \tag{4.11}$$

which has the same variance as that of (2.6) in Section 2.1.2, but the mean vector is the interpolator $\hat{h}$ at the locations $\mathbf{s}_1^y, \ldots, \mathbf{s}_n^y$ rather than a constant $\varphi \mathbf{1}_n$. For $\mathbf{x} \in \mathcal{S}$, the posterior distribution of $f(\mathbf{x})$ given all the data is $f(\mathbf{x}) \mid \mathbf{z}, \mathbf{y}_{1:n}, \sigma_y^2, \boldsymbol{\varphi}^o, \boldsymbol{\psi}^o, \boldsymbol{\psi} \sim \mathsf{N}\left(\mu_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x}),\, \kappa^2_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x})\right)$, where

$$\begin{aligned} \mu_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x}) = m^o(\mathbf{x}; \boldsymbol{\varphi}^o) &+ \mathbf{c}_z^o(\mathbf{x}) \left[\mathbf{C}_{zz}^o\right]^{-1} \left[\mathbf{z} - \mathbf{m}_z^o\right] \\ &+ \mathbf{c}_y(\mathbf{x})^{\mathsf{T}} \left[\mathbf{C}_{yy} + \sigma_y^2 \mathbf{I}_n\right]^{-1} \left[\mathbf{y}_{1:n} - \mathbf{m}_y^o - \mathbf{C}_{yz}^o \left[\mathbf{C}_{zz}^o\right]^{-1} \left[\mathbf{z} - \mathbf{m}_z^o\right]\right], \end{aligned} \tag{4.12}$$

$$\kappa^2_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x}) = \tau^2 - \mathbf{c}_y(\mathbf{x})^{\mathsf{T}} \left[\mathbf{C}_{yy} + \sigma_y^2 \mathbf{I}_n\right] \mathbf{c}_y(\mathbf{x}). \tag{4.13}$$

Both $\mathbf{z}$ and $\mathbf{y}_{1:n}$ contribute to $\mu_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x})$, but $\kappa^2_{\mathbf{z},\mathbf{y}_{1:n}}(\mathbf{x})$ does not depend on $\mathbf{z}$ and is equal to the posterior variance when there is no supplementary information, (2.8) in Section 2.1.2.

The expression for the expected utility for designing the next trajectory segment $\mathbf{q}_{j+1}^*$ is based off of (3.2) in Section 3.1 with occurrences of $m(\mathbf{x}; \boldsymbol{\varphi})$ replaced by $\hat{h}(\mathbf{x})$.

4.5 Illustrations

In this section we evaluate the data fusion algorithms described in Sections 4.2, 4.3, and 4.4 in the context of our trajectory planning algorithm. We consider four scenarios for our evaluation. For each scenario we first define the true field f_T that will be used to randomly generate vehicle data during flight according to (2.1), and then a set of observations $\mathbf{z}$ that serve as the supplementary data for

that scenario. In each case we run a total of 200 simulations, all of them sharing the same supplementary data and true field f_T, but based on different realizations $y(\mathbf{s}_1^y), y(\mathbf{s}_2^y), \ldots$ collected at locations determined by our path planning algorithm.

Our first three scenarios are conceived to provide concrete examples of situations in which each fusion algorithm is most appropriate, and to illustrate the advantages of incorporating the different types of supplementary information in the path planning algorithm (see Table 4.1 for a summary of the parameters used for each scenario). In each of these three scenarios we compare four path planning algorithms: the adaptive path planning algorithm that ignores supplementary information that was discussed in Chapter 3 (AA for short), the appropriate version of the adaptive path-planning algorithm that incorporates supplementary information from Sections 4.2-4.4 (AA+S/P, AA+S/A and AA+S/T for short, respectively), and non-adaptive raster-scan "lawnmower" trajectories that exhaustively explore $\mathcal{S}$ that either ignore or incorporate the supplementary information (denoted by LAW and LAW+S, respectively). The fourth scenario is meant to illustrate the differences between the different data fusion algorithms by simultaneously applying them to the same problem. We focus on optimal reconstruction; thus comparisons across planning algorithms are based on the mean squared prediction error over flight time

$$\mathrm{MSE}(n\Delta) = \frac{1}{|\mathcal{S}_g|} \sum_{\mathbf{x} \in \mathcal{S}_g} \left[f_T(\mathbf{x}) - \hat{f}_n(\mathbf{x}) \right]^2, \tag{4.14}$$

where $\mathcal{S}_g$ is a dense regular grid that covers $\mathcal{S}$, and $|\mathcal{S}_g|$ denotes the number of gridpoints. Because the supplementary information is finite and fixed, the consistency result $\hat{f}_n \to f_T$ as $n \to \infty$ still holds.

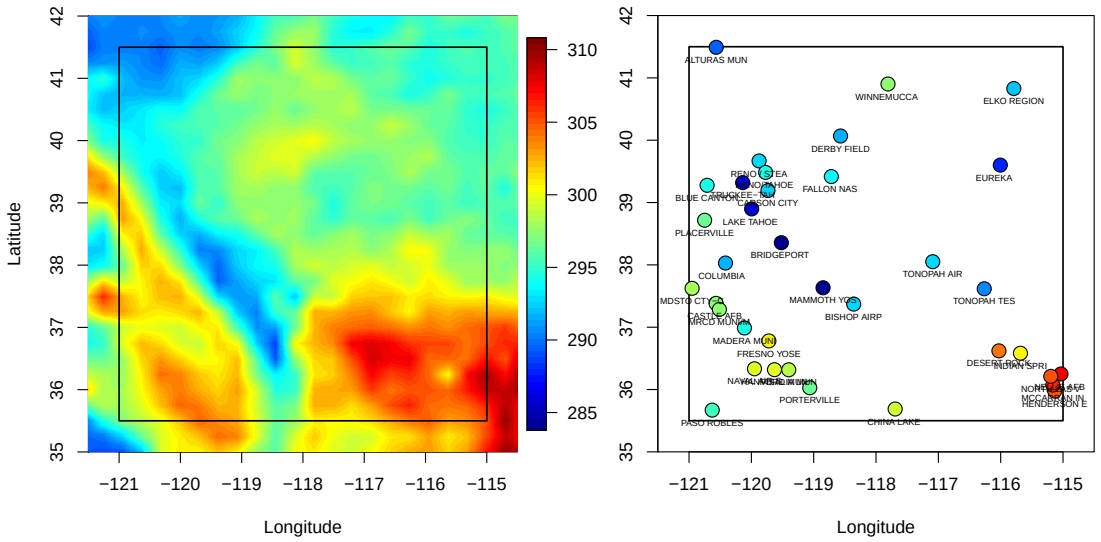
	Point-referenced	Aggregated	Mean function
source	weather stations	satellite image	computer model
f	temperature	aerosol optical depth	carbon concentration
f range	$290 - 308$ K	$0 - 1$	$10 - 65\%$
covariance	Matérn	rational quadratic	Matérn
true σ_y^2	1 K	0.025	2.5%
$\mathcal{S}$	$[-121, -115] \times [35.5, 41.5]$	$[26, 34] \times [-22, -14]$	$[-108, -98] \times [17, 27]$
r	36	16	63
v	0.1884	0.2512	0.314
$R_{\min}$	0.18	0.24	0.3
Δ	1	1	1
k	4	4	4
p	3	3	3
flight time	560Δ	560Δ	560Δ

Table 4.1: Simulation information and parameters.

4.5.1 An example using point-referenced data sources: monitoring temperature in the western USA

To illustrate the performance of our data fusion algorithm for point-referenced data, we investigate the problem of estimating the temperature field over the region $[-121, -115]$ longitude by $[35.5, 41.5]$ latitude, which covers most of Nevada and eastern California. The true temperature field used for generating vehicle measurements during the simulation corresponds to the GEOS-FP meteorological data product from NASA's Global Modeling and Assimilation Office (GMAO) on June 29, 2013 at 0900-UTC. This product provides temperatures over a regular grid with 0.3125° longitudinal and 0.25° latitudinal resolution, and we use bilinear interpolation to obtain temperature values at arbitrarily higher resolutions (see Figure 4.2a). On the other hand, the supplementary information comes in the form of temperature recordings at various fixed stations (located mostly at airports), as provided by the DS3505 Surface Data Hourly Global dataset from the National Climatic Data Center (NCDC) (see Figure 4.2b). For each station,

we use the temperature recorded between 0900-UTC and 1000-UTC (if multiple recordings exist in the hour, we use the one closest to 0900). Generally speaking, the overall pattern of the measurement at the stations agrees with the true field. However, there are some important differences. For example, the measurement at EUREKA station, as well as those along a ridge running from MODESTO CITY to PORTERVILLE, seem to be systematically lower that what would be expected from the shape of the true field.



(a) True temperature field **(b)** Observations at weather stations

Figure 4.2: True field and point-referenced supplementary information for our first evaluation scenario.

Figure 4.3 shows the behavior over time of the log-MSE over 200 experiments for the four trajectory planning methods discussed earlier in the introduction to Section 4.5. The figure presents both the mean log-MSE as well as the interquartile range computed over 200 trajectories. The variability across the 200 trajectories is relatively low in all cases, but is larger for the adaptive algorithms. This is to be expected. In the case of the LAW and LAW+S, the vehicle trajectory is fixed in advance and the only source of variability across experiments in the measurement noise associated with the vehicle sensor. On the other hand, in the

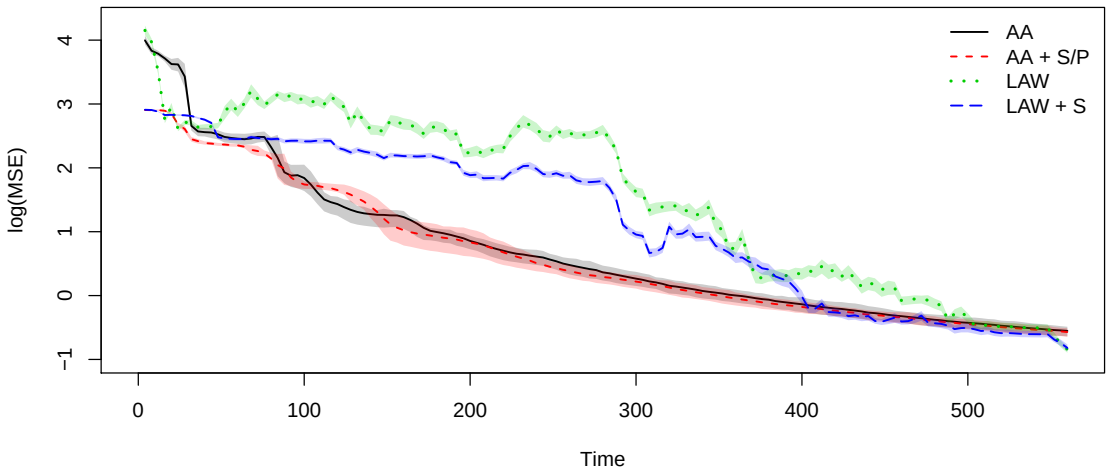


Figure 4.3: Reconstruction error as a function of time over 200 simulations for four trajectory planning algorithms for our first evaluation scenario. Lines correspond to the mean reconstruction errors, while the shaded area indicates the 25th-75th percentile.

case of AA and AA+S/P, the trajectories themselves will change in response to the measurements taken, introducing an additional source of variability across experiments. In terms of the average behavior of the log-MSE, we note that AA and AA+S/P tend to outperform LAW and LAW+S, in spite of the fact that LAW+S incorporates the information from the fixed stations into the estimate of the latent field f. Furthermore, we can see that, with the exception of brief periods of time, AA+S/P and LAW+S tend to outperform their respective counterparts AA and LAW. Finally, we also can see that, as the flight time increases, the MSE values of all four methods tend to converge to a common value.

To better understand the behavior of the different algorithms, we present log-MSE traces for a single experiment under each planning method (see Figure 4.4), as well as snapshots of the trajectories $\mathbf{Q}$, field estimates $\hat{f}$, and squared prediction errors $(f_T - \hat{f})^2$ at four time points in those trajectories (see Figures 4.5 and 4.6). First, we note that the behavior of the log-MSE for the particular experiment depicted in Figure 4.4 agrees with the average behavior we observed in

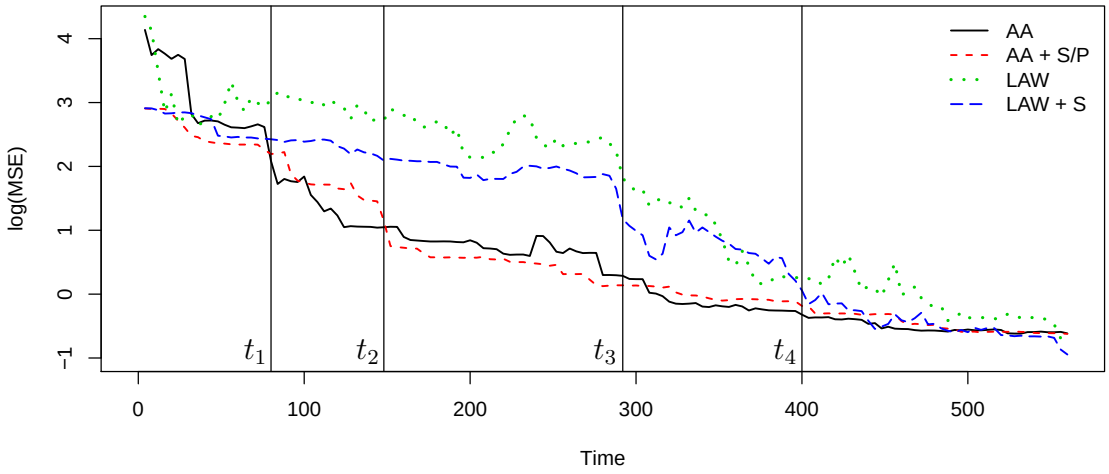


Figure 4.4: Reconstruction error as a function of time for a single simulation for each of the four methods in our first evaluation scenario. The times t_1, t_2, t_3, t_4 correspond to the snapshots of the trajectories, reconstructions, and errors shown in Figures 4.5 and 4.6.

Figure 4.3. This is not surprising given the relatively narrow uncertainty bands we had originally observed. Furthermore, we note that AA seems to provide a better reconstruction of the temperature field than AA+S/P between t_2 and t_3, and observe some sharp drops in MSE at specific time points, particularly for LAW and LAW+S. Focusing now on the snapshots, we see that AA and AA+S/P are able to provide more accurate reconstructions of the field f because they explore the lower-right corner of the sampling space early. Similarly, the more accurate reconstruction provided by AA over AA+S/P between t_2 and t_3 seems to be due to the fact that AA happens to visit the area around the high-temperature ridge running from MODESTO CITY to PORTERVILLE during this period, while AA+S/P does not visit it until later (preferring to explore an unsampled region first). As was noted before, this is an area were the supplementary information seems to systematically underestimate the temperature field. This bias leads AA+S/P to similarly underestimate the temperature values in this regions, an issue that is eventually corrected for once the vehicle visits the area (which happens around

t_3). At this point, AA+S/P starts to again outperform AA. Finally, Figures 4.5 and 4.6 also help explain the sudden drops in MSE values. For example, we note that t_3 is when both lawnmower trajectories first hit the high temperatures of the southeastern region, which allows the algorithm to identify the local mode located west of DESERT ROCK.

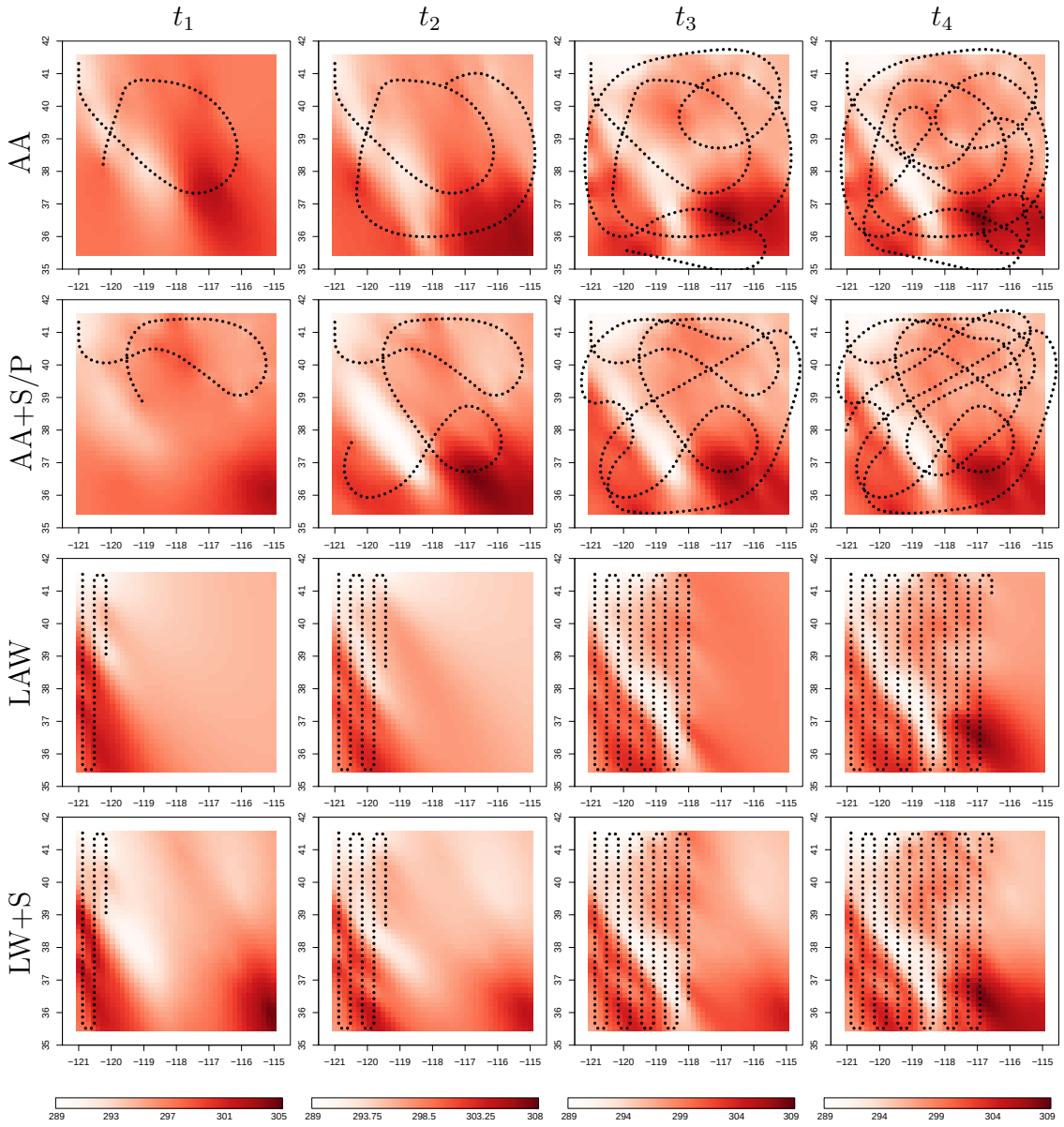


Figure 4.5: Reconstruction of the truth using measurements from locations determined by the four trajectory planning methods in our first evaluation scenario.

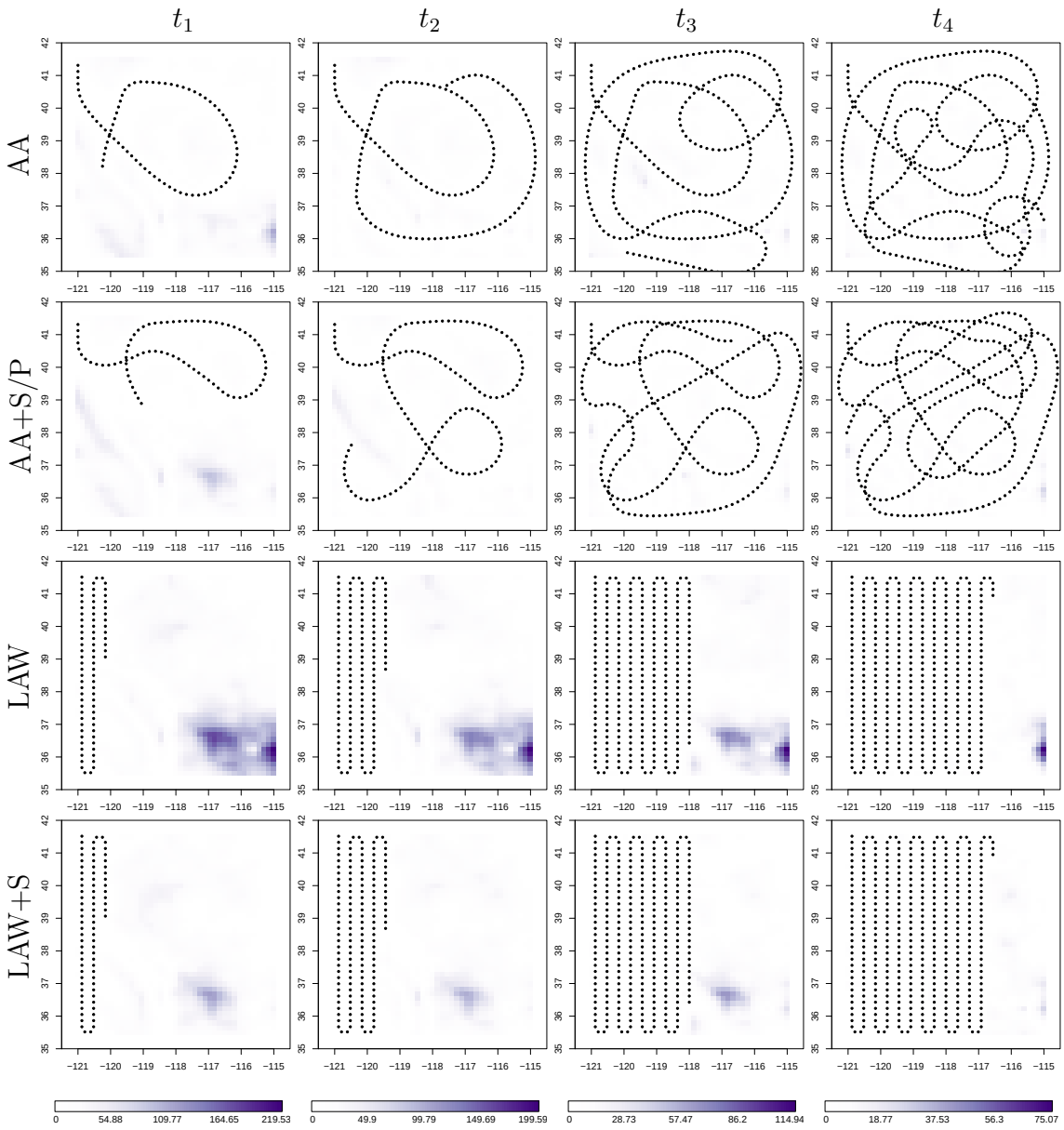


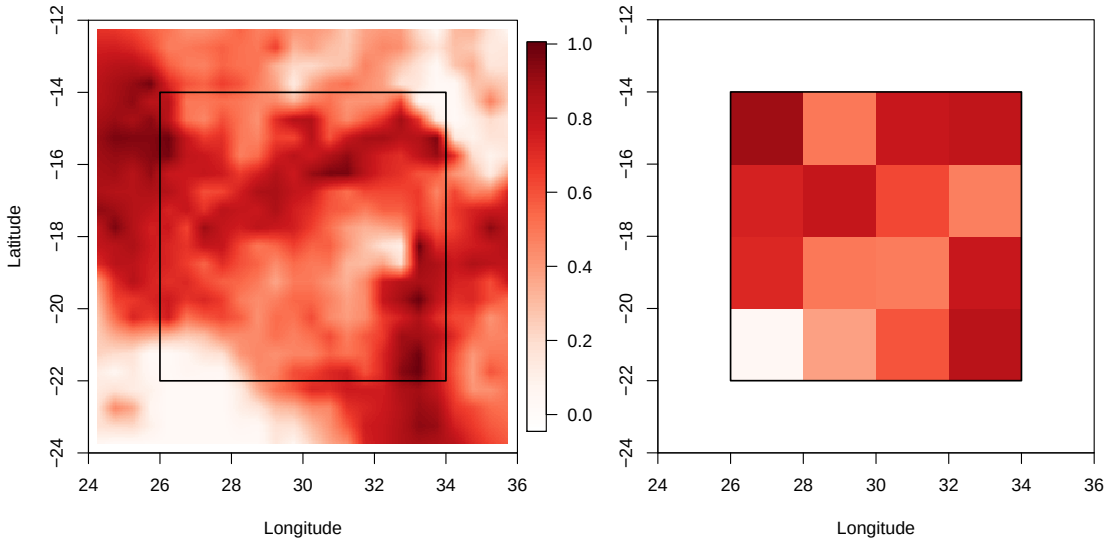
Figure 4.6: Squared reconstruction error at the selected times in our first evaluation scenario.

4.5.2 An example using aggregated data sources: monitoring air quality in Zimbabwe

We proceed now to illustrate the performance of the version of our control algorithm that incorporates aggregated supplementary information introduced in

Section 4.3 (AA+S/A). The motivation in this case is the monitoring of the aerosol optical depth (AOD) field over the region $[26, 34]$ longitude by $[-22, -14]$ latitude, which covers most of Zimbabwe. AOD is an indicator of how much sunlight is blocked by particles in the atmosphere, with higher values corresponding to hazier conditions.

Both the true field used to generate vehicle observations and the supplementary information correspond to images taken by the SeaWiFS sensor on the OrbView-2 satellite. We focus on the 550 nm light wavelength and September 22, 2010. For the truth, we use an image of $0.5°$ longitudinal and $0.5°$ latitudinal resolution, upsampled using a bilinear interpolator (see Figure 4.7a). For the supplementary information, we use a satellite image is $2° \times 2°$, downsampled from the standard $1° \times 1°$ product (see Figure 4.7b). All data products are available through NASA's Giovanni web interface.



(a) True AOD field **(b)** Low resolution satellite image

Figure 4.7: True field and aggregated supplementary information for our second evaluation scenario.

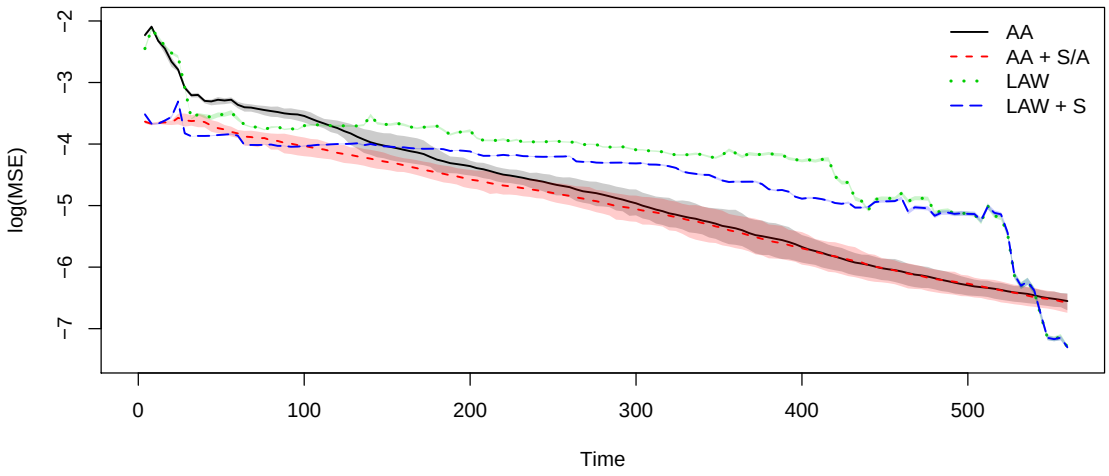


Figure 4.8: Reconstruction error as a function of time over 200 simulations for four trajectory planning algorithms for our second evaluation scenario. Lines correspond to the mean reconstruction errors, while the shaded area indicates the 25th-75th percentile.

Figure 4.8 shows the behavior over time of the log-MSE over 200 experiments for the four trajectory planning methods. Like the weather station scenario, the two approaches that use satellite data outperform the methods that do not. Although LAW+S seems to outperform AA+S during the very early stages of exploration, the average log-MSE for both AA and AA+S/A decreases smoothly with a steeper slope than the average log-MSE of LAW and LAW+S. This indicates a faster learning rate for the adaptive algorithms. One key difference with our first evaluation scenario is that AA+S/A consistently outperforms AA, on average. It is also worth noticing in Figure 4.8 that, towards the end of the simulations, the log-MSE for LAW and LAW+S falls sharply, and by the time the vehicle has systematically explored the space, these methods clearly outperform the adaptive ones.

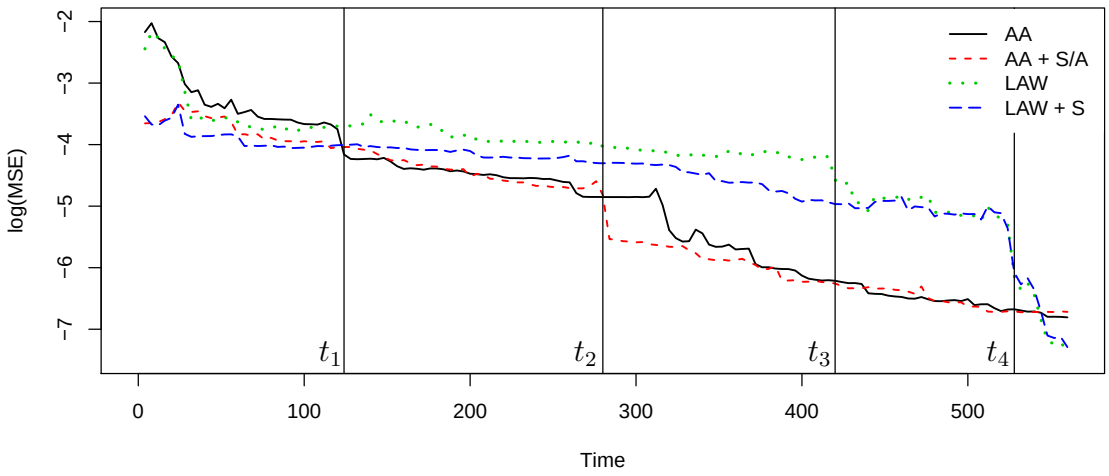


Figure 4.9: Reconstruction error as a function of time for a single simulation for each of the four methods in our second evaluation scenario. The times t_1, t_2, t_3, t_4 correspond to the snapshots of the trajectories, reconstructions, and errors shown in Figures 4.10 and 4.11.

Figure 4.9 presents log-MSE traces for a single experiment under each planning method. These plots highlight some patterns that are not present in the averages presented in Figure 4.8. For example, we can also identify a few short periods of time, mostly between t_1 and t_2 and between t_3 and t_4, in which AA briefly outperforms AA+S/A. The outperformance, however, is never as marked as what we saw in Figure 4.4. In addition, we can see a sharp improvement of the MSE for AA+S/A around t_2 (and for AA slightly later) that does not appear in the summary plot. Looking at Figures 4.10 and 4.11, which present snapshots of the trajectories $\mathbf{Q}$, field estimates $\hat{f}$, and squared prediction errors $\left(f_T - \hat{f}\right)^2$ at t_1, t_2, t_3 and t_4, it is apparent that the sharp drop in MSE is due to a more accurate reconstruction of the upper right corner of the field, where most of the error is concentrated. Sampling the upper right corner at t_4 also explains the sharp drop in MSE for the LAW and LAW+S simulations.

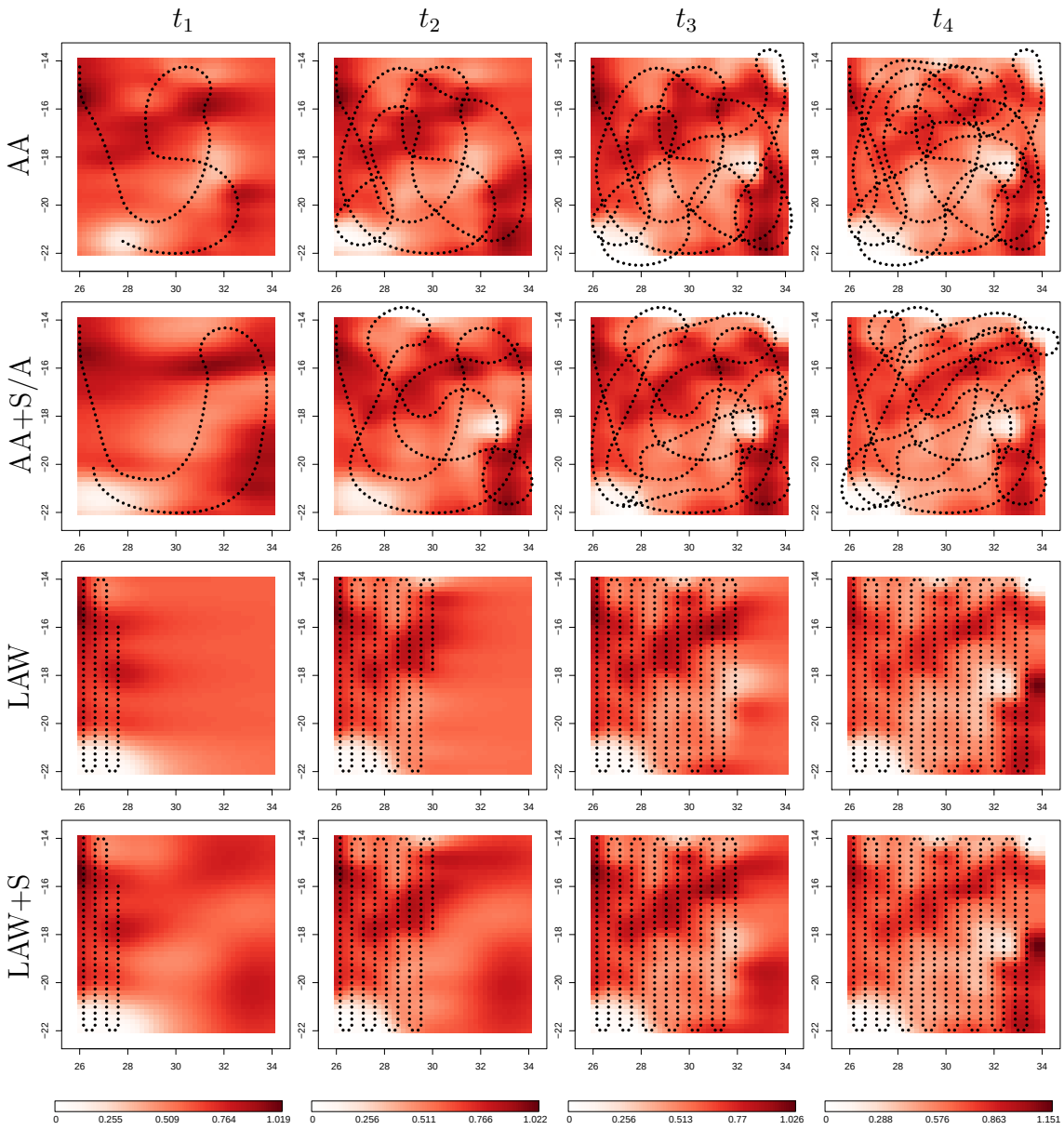


Figure 4.10: Reconstruction of the truth using measurements from locations determined by the four trajectory planning methods.

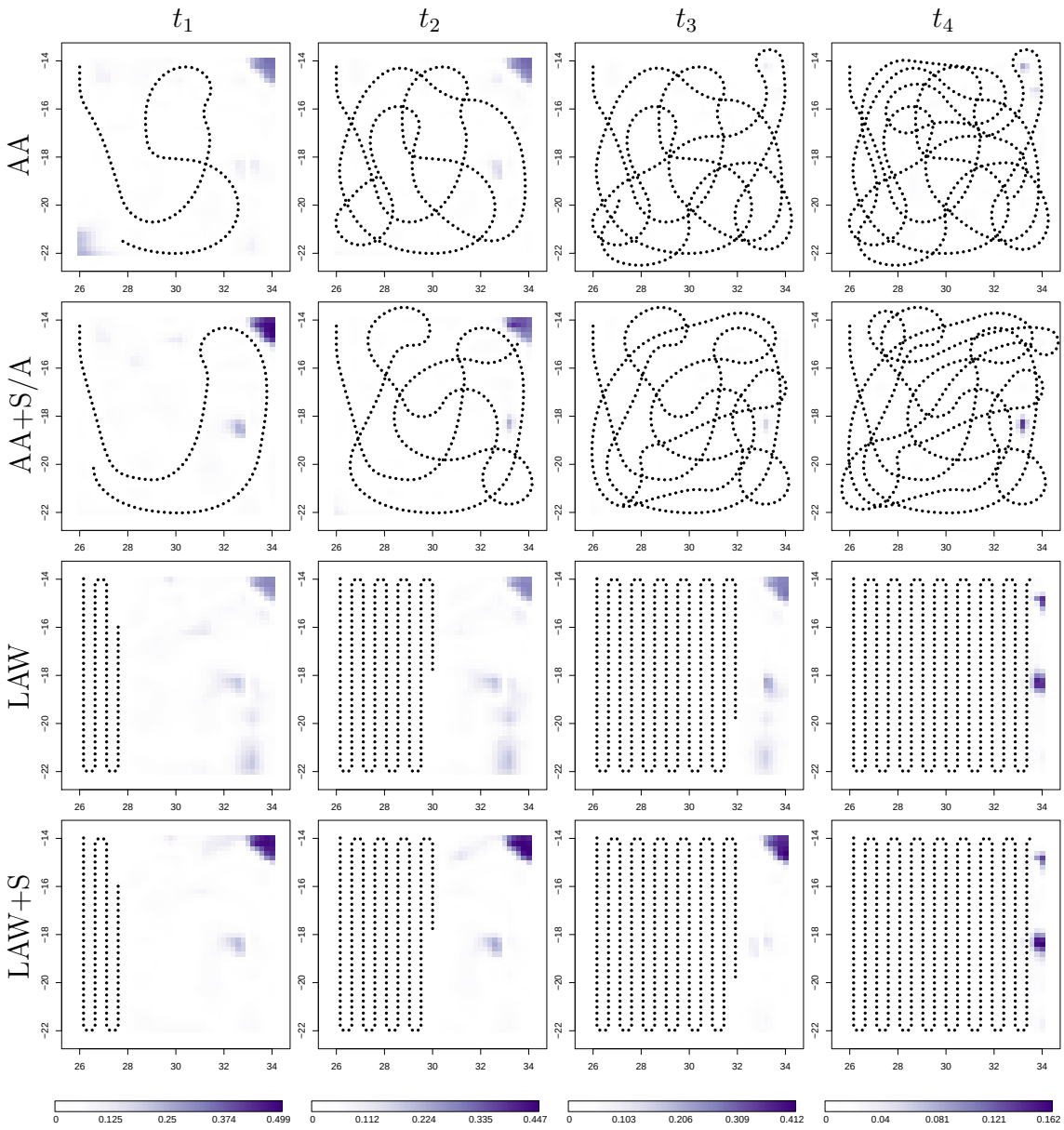
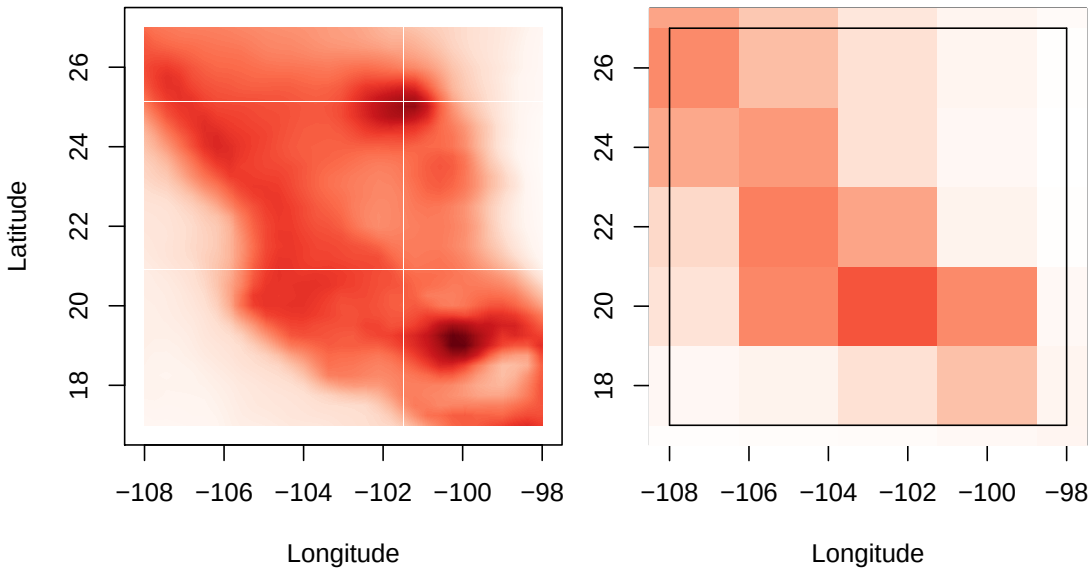


Figure 4.11: Squared reconstruction error at the selected times.

4.5.3 An example using external assessments: monitoring organic carbon over central Mexico

Our third scenario illustrates the performance of the version of our algorithm that incorporates supplementary information through the mean function of the



(a) True organic carbon field **(b)** Computer model output

Figure 4.12: True field and computer model supplementary information for our third evaluation scenario.

Gaussian process model for f. For this scenario, we investigate the percentage of total aerosol mass that is made up by organic carbon (OC) over the region $[-108, -98]$ longitude by $[17, 27]$ latitude, which covers central Mexico. Both the true field used to generate the observations taken by the vehicle and the supplementary information are provided by computer models. For the true OC field, we use a regional GEOS-Chem model prediction with 0.3125° longitudinal and 0.25° latitudinal resolution over North America, which is upsampled using a bilinear interpolator (see Figure 4.12a). For the supplementary information, we use a global GEOS-Chem model prediction at $2.5° \times 2°$ resolution (see Figure 4.12b. Both predictions are for August 2013. Generally speaking, the supplementary information agrees with the true field. However, the field predicted by the low resolution computer model is slightly shifted to the west, and the resulting prior mean function is substantially smoother than the field we assume as the truth.

Figure 4.13 shows the behavior over time of the log-MSE over 200 experiments

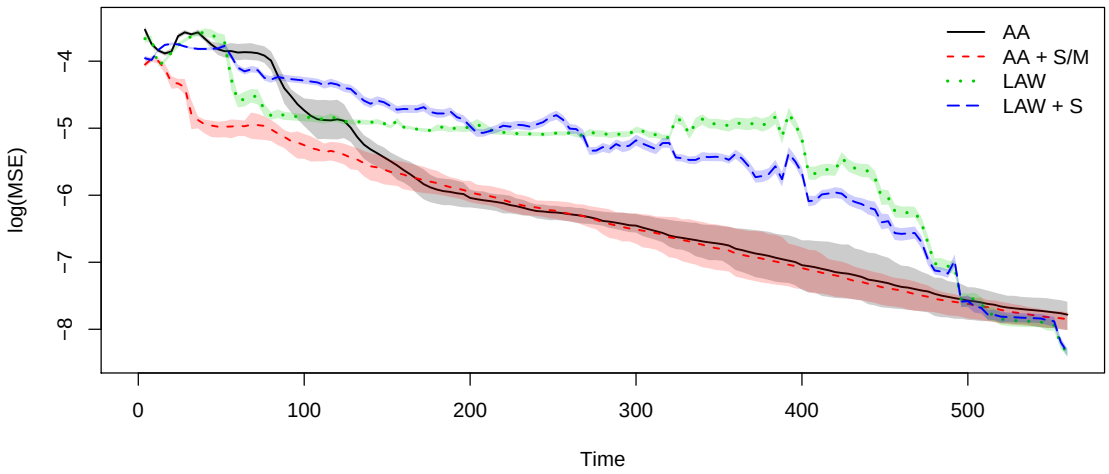


Figure 4.13: Reconstruction error as a function of time over 200 simulations for four trajectory planning algorithms for our third evaluation scenario. Lines correspond to the mean reconstruction errors, while the shaded area indicates the 25th-75th percentile.

for the four trajectory planning methods. For our adaptive algorithm, using the computer model information results in lower log-MSE in the beginning of the simulations, which is also observed in the weather station and satellite scenarios. Like the satellite scenario, AA+S/M outperforms AA on average, with the exception of one brief period of time. However this improvement is not observed in LAW and LAW+S. The average log-MSE is lower for LAW than for LAW+S for the first half of the flight time. Incorporating the computer model into the lawnmower method does not initially lower the log-MSE as quickly as the basic LAW does. Like the two previous scenarios, the average log-MSE for both AA and AA+S/A decreases smoothly with a steeper slope than the average log-MSE of LAW and LAW+S for most of the time. And again, the log-MSE for LAW and LAW+S falls sharply towards the end of the simulations when the vehicle has systematically explored the space.

Figure 4.14 shows log-MSE traces for a single experiment under each planning method. The behavior of the log-MSE for the single experiment agrees with the

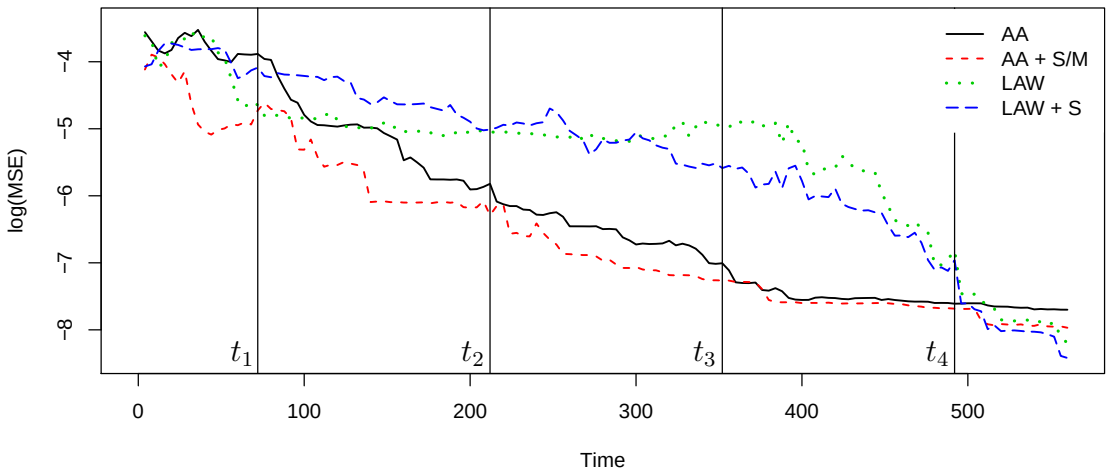


Figure 4.14: Reconstruction error as a function of time for a single simulation for each of the four methods in our third evaluation scenario. The times t_1, t_2, t_3, t_4 correspond to the snapshots of the trajectories, reconstructions, and errors shown in Figures 4.15 and 4.16.

average behavior observed in Figure 4.13. In particular, AA+S/M outperforms AA until a point in time when the two log-MSE traces converge, and LAW+S has higher MSE compared to LAW for the first half of the simulation. The snapshots the trajectories, field estimates, and squared errors of Figures 4.15 and 4.16 are taken at times t_1, t_2, t_3, t_4 evenly spaced between $t = 0$ and $t = 560$. Looking at these snapshots, the slower learning by LAW+S compared to LAW until after time t_2 is explained by the large error contribution by inaccurately estimating the region of high OC in the upper right corner. As this region has not yet been visited by the vehicle, the error is due to the inaccuracy in the computer model.

At the first time t_1, the reconstruction from the AA method is too flat due to the trajectory having only sampled at locations with moderate/high concentration. The other three methods do not experience this issue. AA+S/M has the computer model information, and LAW samples across the region, gathering high and low concentration measurements. Over time the AA trajectory covers the region, and reconstruction is greatly improved. The trajectories of the 200

AA simulations typically start this way: from the initial locations, the vehicle cuts more or less diagonally across the region and makes a right-angle turn three-quarters of the way across, mostly sampling moderate to high concentrations, as seen in Figure 4.12a.

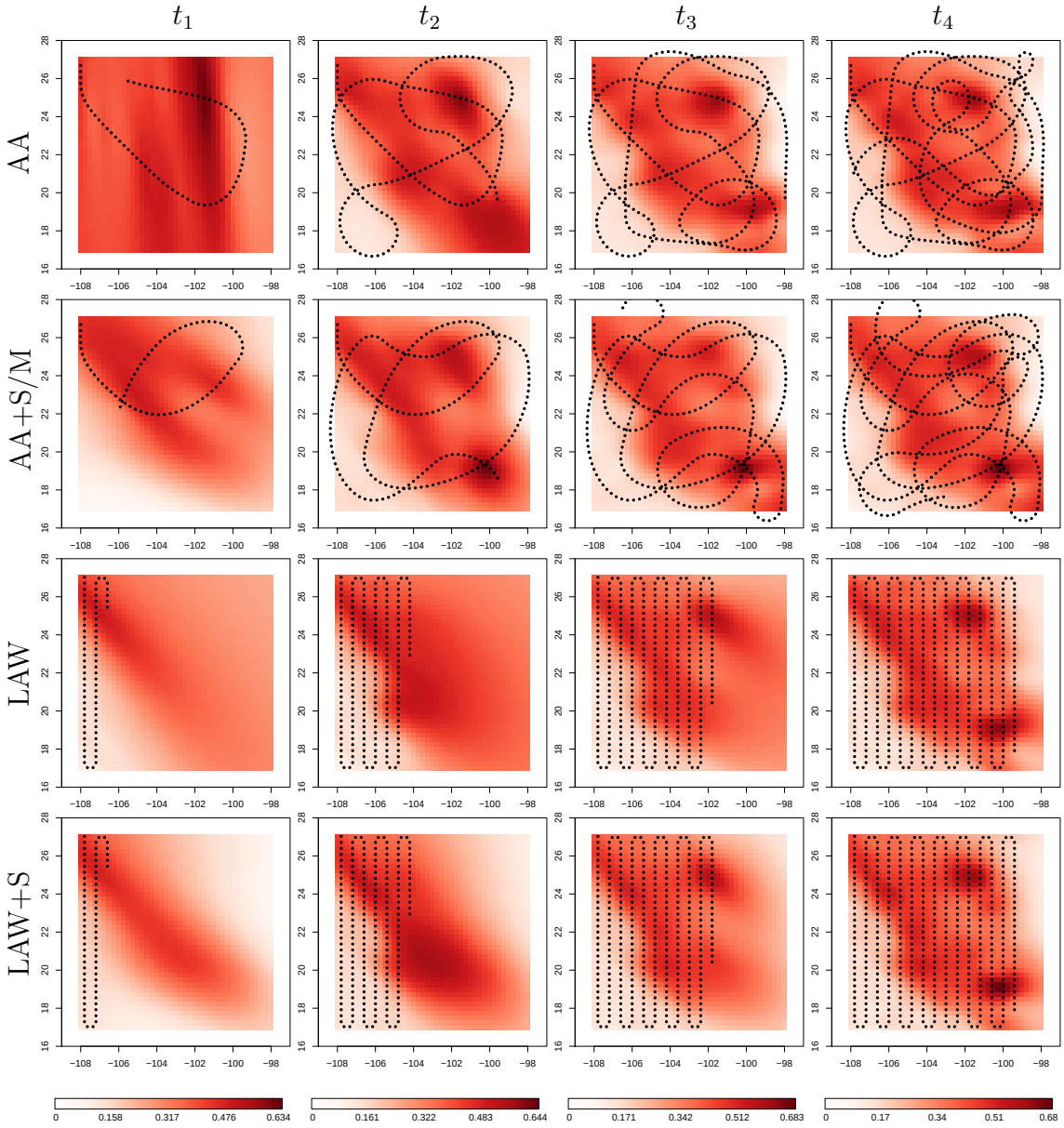


Figure 4.15: Reconstruction of the truth using measurements from locations determined by the four trajectory planning methods.

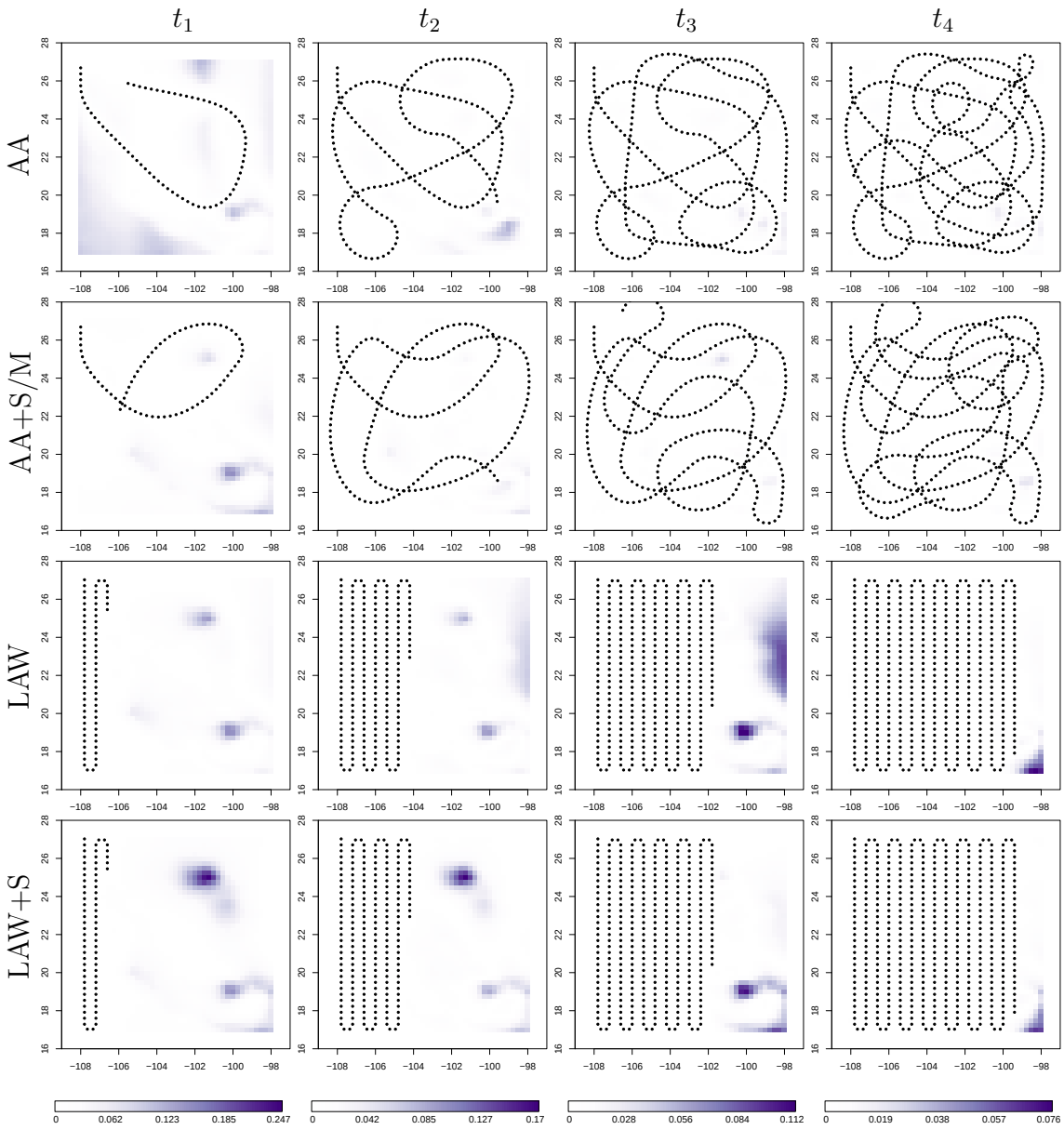


Figure 4.16: Squared reconstruction error at the selected times.

4.5.4 A comparison of data fusion approaches

In our final evaluation scenario, we compare the performance of the different data fusion algorithms in the AOD setting introduced in Section 4.5.2. This looks into the issue of using the wrong data fusion method, as the satellite image will be treated as point-referenced in AA+S/P and as the prior mean function in

AA+S/M. Similarly to previous evaluations, we present in Figure 4.17 a summary of the evolution of the log-MSE values corresponding to 200 experiments for each of AA+S/P, AA+S/A and AA+S/M, as well as for the basic adaptive path planning algorithm (AA). Since the supplementary information has the most impact on the early performance of the algorithms, we focus on their behavior during the first 100 time points. It is clear from this graph that including supplementary information is preferable to not including it, no matter which of the three approaches is used to incorporate it. We also see that, in this case, AA+S/M has the worst performance among the three data fusion algorithms, and that the performance of AA+S/P and AA+S/A is quite comparable.

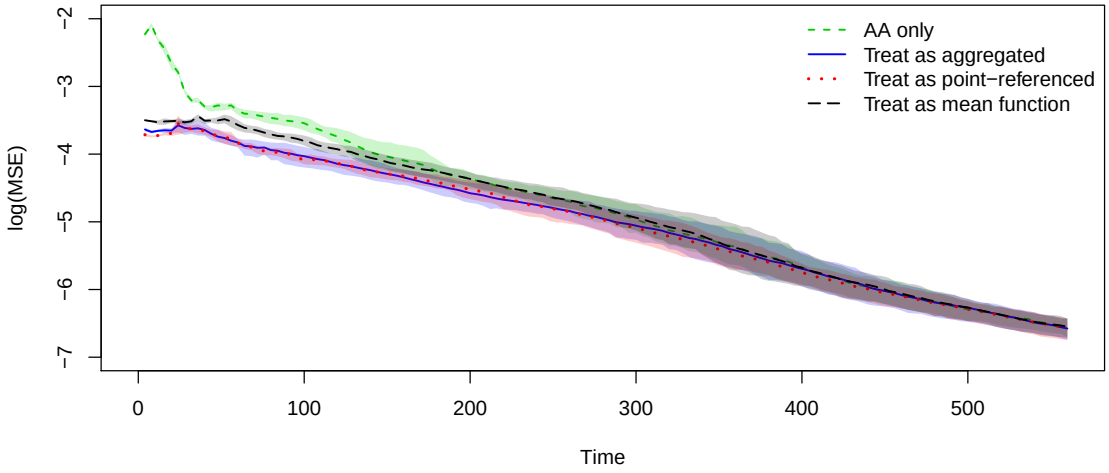


Figure 4.17: Reconstruction error as a function of time over 200 simulations. Lines correspond to the mean reconstruction errors, while the shaded area indicates the 25th-75th percentile.

4.6 Conclusions

We have developed methods to incorporate supplementary information in three ways, depending on the nature of the supplementary source. Through simulations evaluating each incorporation method, we have shown that including the supple-

mentary information into the planning algorithms improves the performance of the algorithms, especially in the early stages of exploration.

The amount of improvement, and how long the improvement persists, vary across the different data sources and incorporation methods. One factor is the accuracy and spatial coverage of the supplementary data. In the point-referenced scenario with the weather stations, the supplementary data is concentrated at specific locations, and we mentioned earlier there are some systematically lower temperatures in a subset of locations. In the other scenarios, the supplementary information is evenly distributed over the entire region, and the supplementary satellite data in the aggregated scenario is more accurate due to it likely being a low resolution version of the assumed truth. Another factor is the incorporation method itself; in particular, incorporating information through the mean function of the Gaussian process prior is inherently different than treating the supplementary data as direct observations of the true field. A future project can be a more formal analysis on the factors that influence the performance of our incorporation methods.

The approximate times (in minutes, non-parallelized) to run a single simulation are shown in Table 4.2.

	Scenario 1		Scenario 2		Scenario 3	
computation grid size $\lvert\mathcal{S}_g\rvert$	AA	AA+S/P	AA	AA+S/A	AA	AA+S/M
21×21	30	40	13	14	31	25
31×31	47	58	27	29	47	41
41×41	72	87	48	52	74	68

Table 4.2: Simulation run times.

Chapter 5

Multiple Vehicles

The procedure discussed in Chapter 3 gives a comprehensive method for one vehicle to plan its optimal trajectory for optimal reconstruction or finding the maximum. However, there are situations where there is limited time to accomplish the goal, such as when data collection by one vehicle alone cannot resolve the timescale of the environmental process. To this end, we extend the trajectory planning algorithm to multiple vehicles and investigate the rate of performance improvement from including additional vehicles.

This chapter is organized as follows. Sections 5.1 and 5.2 discuss concepts that arise when dealing with multiple vehicles, such as communication, detection, and avoiding collisions between vehicles. Section 5.3 extends the trajectory planning algorithm from Chapter 3 to the multiple vehicle case, modifying the objective functions in Section 3.1, with attention to collision avoidance. Section 5.4 illustrates the performance of the extended method, with focus on the "speedup" of time in achieving the mission goal.

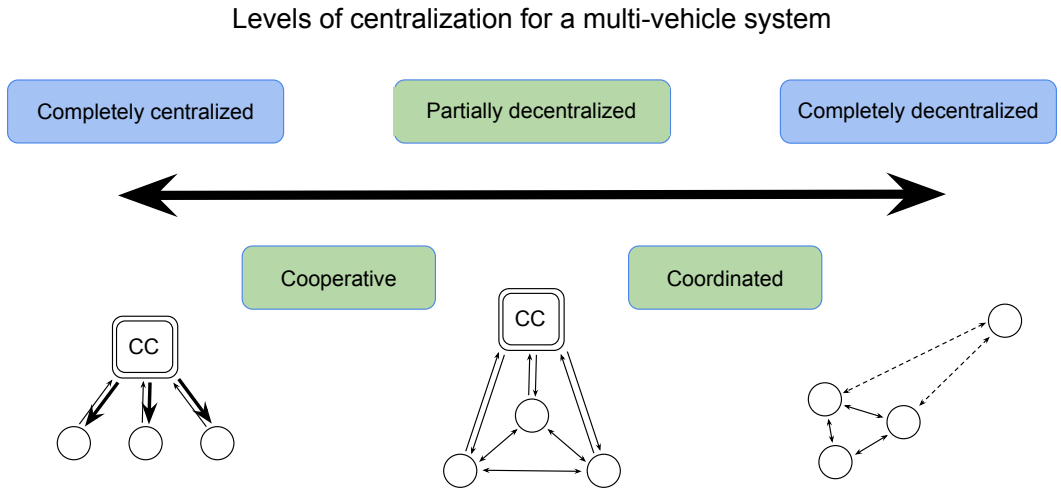


Figure 5.1: The spectrum of centralization, cooperation, and coordination.

5.1 Centralization and communication

A multi-vehicle system can consist of only vehicles, or can involve a fixed computer in addition to the vehicles. The vehicles can carry out the the heavy computation, perhaps in some distributed way, or most of the computation can be done by the fixed computer by communicating the data off the vehicles. The following is a discussion of possible ways for task allocation among the components of the multi-vehicle system, graphically represented in Figure 5.1, and applied to our context of sampling, estimation, and trajectory planning.

In a **fully centralized** scheme, a central computer (CC) does all of the data processing and decision making. In our context, in each iteration of the trajectory planning algorithm, the CC receives measurement and location data from all vehicles, then computes the hyperparameters to reconstruct the field, and then jointly optimizes the combined next locations for the vehicles. There is no communication regarding trajectory planning among vehicles.

In a **partially centralized** scheme, the CC receives measurement and loca-

tion data from all vehicles and computes and broadcasts the hyperparameters. However, the vehicles use the hyperparameters to optimize their next trajectory segment individually. The vehicles can also broadcast their current location for collision avoidance. (The CC can broadcast this too).

In a fully decentralized scheme, there is no CC. Location and measurement history are kept within each vehicle. Each vehicle computes the hyperparameters and optimizes its next trajectory segment individually. For collision avoidance, the vehicles broadcast their current locations to other vehicles that are within some distance based on communication capacity and collision avoidance needs. Optionally, the vehicles broadcast their location history, measurements, or hyperparameters for better performance.

Another way to characterize the interactions between components of a multi-vehicle system is to use the terms cooperation and coordination [62]. In a **coordinated** system, the vehicles plan individually on their current knowledge and only exchange knowledge (such as locations, measurements, or hyperparameters) via some established network. In a **cooperative** system, the vehicles share future intent in addition to the aforementioned information.

In developing our extension to multiple vehicles, we will use a partially centralized scheme that is considered more cooperative than coordinated.

5.2 Collision avoidance

Collision avoidance is important area of research. In particular, in the context of multiple vehicles sharing the same spatial exploration goal, the vehicles are naturally drawn to similar locations. However, getting too close spatially (to other vehicles or other obstacles) results in collisions, of which consequences range from the vehicles being thrown off course, to causing material damage, to catastrophic

crash and mission failure. The latter is especially serious in aerial vehicles.

Generally, collision avoidance methods are of two main types: global (motion planning methods) and local (reactive methods). Global methods search through the vehicle's possible trajectories for the best collision-free trajectory with respect to some cost function, often with a graph-based planner, such as A^* or rapidly-exploring random tree (RRT) [1]. Replanning may be required in the case when not all obstacles are known ahead of time [59]. On the other hand, local methods do not find the optimal trajectory, but rather compute changes in the control input (possibly obtained from a global method) using local, sensor information about the obstacles and/or other vehicles. For collision avoidance between vehicles, these reactive methods often develop control laws that guarantee separation between vehicles in the presence of position uncertainties [67] and result in robot formations [23].

The extension of our trajectory planning method to multiple vehicles is a global one; we maintain optimal trajectory generation for each vehicle. However, these trajectories need to be as collision-free as possible, and in the case of an impending collision, the existence of an local, reactive controller is assumed.

Local methods is a rich topic of research in itself and has seen applications in both ground and aerial vehicles, with the former having more developed systems. For example ground collision avoidance systems in self-driving cars include the use of radar, LIDAR, video cameras, and an automatic emergency brake system. Commercially available autonomous robotic vacuums use infrared sensors for navigating around walls and preventing from falling down stairs. These sensors are also being used in developing obstacle/collision detection and avoidance methods for aerial vehicles. In [6], LIDAR onboard a remotely-operated UAV provides obstacle detection, and a control algorithm modifies the operator's input to avoid

collisions while attempting to maintain the operator's intent. A testing platform for the development of fully autonomous flight including collision avoidance in [25] uses radar, video, and infrared video to generate and follow real-time escape trajectories. In [53], quadrotor UAVs demonstrate coordinated flight while avoiding dynamic obstacles by using a motion capture system with onboard navigation sensors.

5.3 Global method for multiple vehicles

In Chapters 3 and 4, implementation of the trajectory planning method for a single vehicle uses a reward function of the form

$$R(\mathbf{q}_j) = \tilde{U}(\mathbf{q}_j) - \alpha P_{\text{leave}}(\mathbf{q}_j) \tag{5.1}$$

This reward can be extended to incorporate multiple vehicles in our framework. Let N_v denote the number of vehicles. One approach maximizes at each iteration j a joint reward over all vehicles

$$R_{\text{joint}}\left(\mathbf{q}_j^1, \ldots, \mathbf{q}_j^{N_v}\right) = \tilde{U}_{\text{joint}}\left(\mathbf{q}_j^1, \ldots, \mathbf{q}_j^{N_v}\right) - \alpha_1 \sum_{\ell=1}^{N_v} P_{\text{leave}}\left(\mathbf{q}_j^\ell\right) - \sum_{\ell=1}^{N_v} P_{\text{collision}}\left(\mathbf{q}_j^\ell, \mathbf{q}_j^{-\ell}\right) \tag{5.2}$$

where $\tilde{U}_{\text{joint}}\left(\mathbf{q}_j^1, \ldots, \mathbf{q}_j^{N_v}\right)$ is the expected utility from considering all segments $\left\{\mathbf{q}_j^\ell\right\}_{\ell=1}^{N_v}$ jointly, P_{leave} is the same penalty function as in (5.1) to discourage the vehicles from leaving $\mathcal{S}$, $P_{\text{collision}}$ is a penalty function that discourages each vehicle from colliding with the other vehicles, and $\mathbf{q}_j^{-\ell}$ denotes the trajectory segments associated with all other vehicles that are not the ℓ^{th} vehicle. The structure of (5.2) makes the common assumption that the penalties can be assessed separately from the joint expected utility.

This approach requires a central control unit to solve an N_v-dimensional optimization problem that jointly selects the next trajectory segments of all N_v vehicles simultaneously, which is not ideal in real-time systems due to the curse of dimensionality characterized by N_v. To reduce computation time and reliance on the central computer, and to allow vehicles to compute their own trajectories, we compute

$$R_{\text{total}}\left(\mathbf{q}_j^1,\ldots,\mathbf{q}_j^{N_v}\right)=\tilde{U}_{\text{total}}\left(\mathbf{q}_j^1,\ldots,\mathbf{q}_j^{N_v}\right)-\alpha_1\sum_{\ell=1}^{N_v}P_{\text{leave}}\left(\mathbf{q}_j^{\ell}\right)-\sum_{\ell=1}^{N_v}P_{\text{collision}}\left(\mathbf{q}_j^{\ell},\mathbf{q}_j^{-\ell}\right) \tag{5.3}$$

where

$$\tilde{U}_{\text{total}}\left(\mathbf{q}_j^1,\ldots,\mathbf{q}_j^{N_v}\right)=\sum_{\ell=1}^{N_v}\tilde{U}\left(\mathbf{q}_j^{\ell}\right) \tag{5.4}$$

is the sum of the expected utilities derived in 3.1 for a single vehicle. With this definition, the total reward in (5.3) can be rewritten as

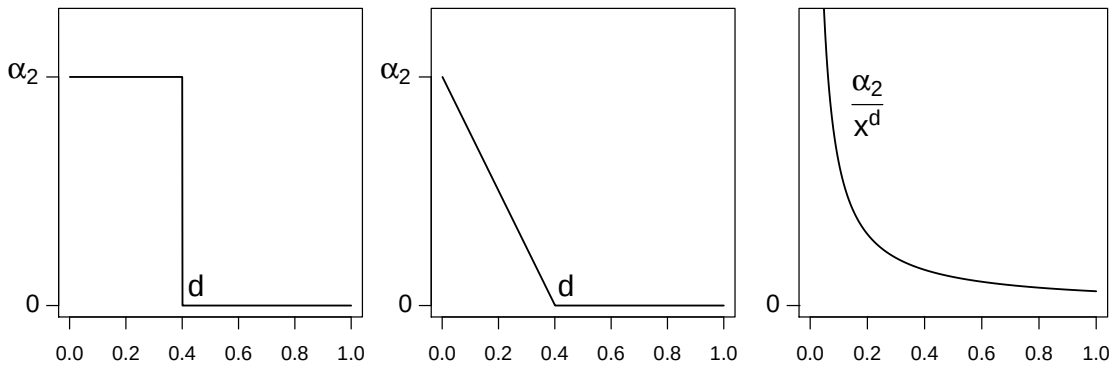
$$R_{\text{total}}(\mathbf{q}_j^1,\ldots,\mathbf{q}_j^{N_v})=\sum_{\ell=1}^{N_v}\left[R\left(\mathbf{q}_j^{\ell}\right)-P_{\text{collision}}\left(\mathbf{q}_j^{\ell},\mathbf{q}_j^{-\ell}\right)\right], \tag{5.5}$$

which is a sum of individual rewards (5.1), modified to include a collision avoidance term, discussed below.

5.3.1 Collision penalty function

The goal of collision avoidance means that, for each iteration j, we want to ensure that the minimum distances between all vehicles do not go below a certain distance. This gives rise to a collision penalty function of the form

$$P_{\text{collision}}\left(\mathbf{q}_j^{\ell},\mathbf{q}_j^{-\ell}\right)=\sum_{r\in-\ell}\alpha_2 K\left(\min_{t\in[0,pk]}\left\|\mathbf{q}_j^{\ell}(t)-\mathbf{q}_j^{r}(t)\right\|\right), \tag{5.6}$$



(a) step **(b)** ramp **(c)** inverse

Figure 5.2: Three choices for K. Along with K, the penalty function is controlled by α_2 and d.

where K is a decreasing function, whose argument is the minimum distance between two trajectory segments over the planning horizon, and $\alpha_2 > 0$ is a tuning parameter.

Common choices of K are step, ramp, and inverse functions, as shown in Figure 5.2. Each of these involves a threshold parameter d that turns off or attenuates the collision penalty. In the case of the step function, a high α_2 guarantees that vehicle ℓ will always be at least a distance d from the other vehicles. Use of a step function is equivalent to setting an additional constraint on $\mathbf{q}_j^\ell$ to be at least d distance from the other segments. The ramp function behaves similarly to the step, but the penalty is scaled by the distance. The inverse function does not require specifying the distance that turns off the penalty, and as the distance between two vehicles goes to zero, a high collision penalty is automatic.

For faster computation, and to allow for each vehicle to plan its next segment in parallel, the trajectory segments are discretized, and we modify (5.6) to obtain

$$P_{\text{collision}}\left(\mathbf{q}_j^\ell, \mathbf{q}_j^{-\ell}\right) = \sum_{r \in -\ell} \alpha_2 K\left(\min_{i=1,\ldots,pk} \left\|\mathbf{q}_j^\ell(i\Delta) - \mathbf{q}_j^r(0)\right\|\right). \tag{5.7}$$

Because $\mathbf{q}_j^r(0) = \mathbf{q}_{j-1}^{*r}(k\Delta)$, (5.7) only requires vehicle ℓ to know the last sam-

pling location of the other vehicles (and not their plans for the current iteration). Although the minimum distance guarantees are weakened, this difference between (5.6) and (5.7) is key for realtime implementation.

With $P_{\text{collision}}$ specified by (5.7), the total reward to be maximized is the sum

$$\sum_{\ell=1}^{N_v} \left[\tilde{U}\left(\mathbf{q}_j^\ell\right) - \alpha_1 P_{\text{leave}}\left(\mathbf{q}_j^\ell\right) - \alpha_2 \sum_{r \in -\ell} K\left(\min_{i=1,\ldots,pk} \left\| \mathbf{q}_j^\ell(i\Delta) - \mathbf{q}_j^r(0) \right\| \right) \right]. \tag{5.8}$$

The parameters α_1 and α_2 tune the penalties for leaving and colliding, respectively. Appropriate values for α_1 and α_2 may be difficult to know beforehand and depend on the scales of f and $\mathcal{S}$. To make α_1 and α_2 agnostic to these scales, and preclude the need to tune them to different values for every specific application, we set it them to be fractions of $\tilde{U}\left(\mathbf{q}_j^\ell\right)$. Then the reward for an individual vehicle is

$$\tilde{U}\left(\mathbf{q}_j^\ell\right) \left[1 - a_1 P_{\text{leave}}\left(\mathbf{q}_j^\ell\right) - a_2 \sum_{r \in -\ell} K\left(\min_{i=1,\ldots,pk} \left\| \mathbf{q}_j^\ell(i\Delta) - \mathbf{q}_j^r(0) \right\| \right) \right], \tag{5.9}$$

where a_1 and a_2 satisfy $\alpha_1 = a_1 \tilde{U}(\mathbf{q}_j^\ell)$ and $\alpha_2 = a_2 \tilde{U}(\mathbf{q}_j^\ell)$. With K as the step function, we can interpret a_1 and a_2 as the reductions in the utility of the segment due to the possibility of leaving $\mathcal{S}$ or of colliding with another vehicle. For example, with $a_2 = 0.05$, the reward associated with the vehicle segment $\mathbf{q}_j^\ell$ is reduced by 5 percent if $\mathbf{q}_j^\ell$ comes too close to the last known location of another vehicle.

To illustrate the efficacy of setting α_2 as a proportion of $\tilde{U}$, consider the four possible combinations of $\tilde{U}(\mathbf{q}_j^\ell)$ and $K\left(\min_i \left\| \mathbf{q}_j^\ell(i\Delta) - \mathbf{q}_j^r(0) \right\|\right)$:

1. low utility and high collision penalty $\longrightarrow$ segment has low reward
2. high utility and low collision penalty $\longrightarrow$ segment has high reward
3. high utility and high collision penalty $\longrightarrow$ segment is less likely to be chosen because it is unsafe

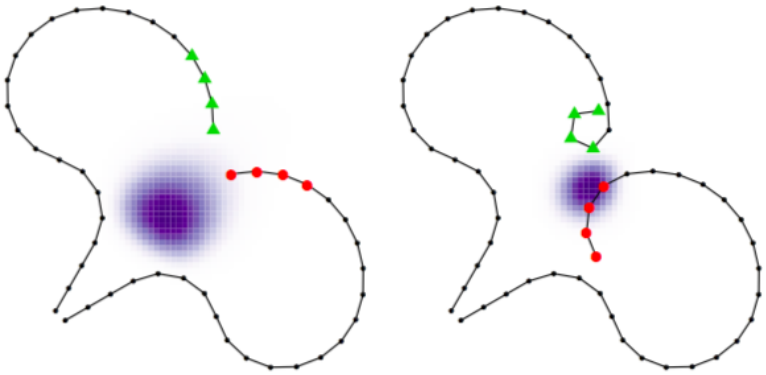


Figure 5.3: Collision avoidance illustration.

4. low utility and low collision penalty $\longrightarrow$ segment is relatively more likely to be chosen because it is safe

The first two combinations are straightforward – the segment is not (is) likely to be chosen because both utility and collision are unfavorable (favorable). The last two combinations balance the reward by decreasing (increasing) it relative to other segments.

Figure 5.3 shows an effective case of the collision avoidance policy. Both the red and green vehicles are heading into a region of high utility (shown in purple). Because the current location of the red vehicle is nearby, and heading into the high utility region will decrease the already small distance between the two vehicles, the green vehicle turns around rather than explore the high utility region to avoid a collision. The resulting segment for the red vehicle is also affected; it reaches the high utility region but at an angle that turns away from the green vehicle instead of heading towards the center of the high utility region. Sometimes, both vehicles will avoid the high utility region in order to avoid each other, as illustrated in the beginning of the trajectories for the two vehicles.

Algorithm 4 summarizes the multiple vehicle planning algorithm, with parallelized vehicle tasks highlighted in blue and communication highlighted in red.

This algorithm assumes a coordinated, partially decentralized structure, with communication between a central computer and the vehicles.

Algorithm 4 Optimal trajectory algorithm (general, multiple vehicles)

Input: goal, $\mathcal{S}$, number of vehicles N_v, vehicle kinematics, vehicles' initial locations and heading angles, sampling period Δ, trajectory segment size k, planning horizon parameter p, out-of-bounds penalty parameter α_1, collision penalty parameter α_2, prior distributions for hyperparameters $(\sigma^2, \boldsymbol{\varphi}, \psi)$, and stopping rule

Output: Optimal trajectories $\mathbf{Q}^1, \ldots, \mathbf{Q}^{N_v}$

Set $j = 1$ and $n = k$.
For each vehicle ℓ, initialize $\mathbf{Q}^\ell$ with k locations by going straight along initial heading angle from vehicle's initial location
For each vehicle ℓ, initialize $\mathbf{y}^\ell_{1:n}$ with the measurements taken at $\mathbf{Q}^\ell$
while stopping rule is not met **do**
    Vehicles broadcast most recent k locations and observations
    Estimate $(\sigma^2, \boldsymbol{\varphi}, \boldsymbol{\psi})$ by maximizing (2.10) (done by CC)
    CC broadcasts $\widehat{\sigma}^2, \widehat{\boldsymbol{\varphi}}, \widehat{\boldsymbol{\psi}}$
    For each vehicle ℓ, obtain $\mathbf{q}^{*\ell}_{j+1}$ by maximizing individual reward in (5.9) subject to (3.5)
    For each vehicle ℓ, append $\mathbf{q}^{*\ell}_{j+1}(i\Delta)$, $i = 1, \ldots, k$ to $\mathbf{Q}^\ell$
    For each vehicle ℓ, sample at $\mathbf{q}^{*\ell}_{j+1}(i\Delta)$, $i = 1, \ldots, k$
    For each vehicle ℓ, add new measurements to $\mathbf{y}^\ell_{1:n}$
    Set $j = j + 1$ and $n = n + k$
end while

5.4 Evaluations

In this section, we evaluate our multiple vehicle method described in Section 5.3 for the goal of optimal reconstruction (and preliminarily, finding the maximum of f) for several true fields. The simulations here aim to show that using more vehicles achieves the goal faster, and we investigate the rate the improvement in performance. Furthermore we demonstrate the effectiveness of the collision avoidance policy by comparing simulations that include collision avoidance to simulations without collision avoidance.

f	Isotropic field	Temperature	Carbon concentration
f range	$0-1$	$290-308$	$10-65\%$
covariance	Matérn	Matérn	Matérn
true σ^2	0.04	1 K	2.5%
$\mathcal{S}$	$[0,1]^2$	$[-121,-115]\times[35.5,41.5]$	$[-108,-98]\times[17,27]$
v	0.036	0.1884	0.314
$R_{\min}$	0.035	0.18	0.3
Δ	1	1	1
k	4	4	4
p	3	3	3
total observations	400	400	400
α_1	$0.02\,\tilde{U}$	$0.02\,\tilde{U}$	$0.02\,\tilde{U}$
α_2	$0.02\,\tilde{U}$	$0.02\,\tilde{U}$	$0.02\,\tilde{U}$
$K(x)$	$1/(x/0.5)^3$	$1/(x/3)^3$	$1/(x/5)^3$

Table 5.1: Simulation parameters for evaluating the multiple vehicle method.

The environmental fields we use are the isotropic truth from Section 3.4, the temperature data from Section 4.5.1, and the organic carbon concentration data from Section 4.5.3. The simulation parameters are listed in Table 5.1. Two hundred simulations are run for each $N_v = 1, 2, 3, 4, 5$ for each of the three true fields. In all simulations, the number of sampling locations is constant. This means that a simulation with two vehicles will generate trajectories of size 200 for each vehicle, while a simulation with four vehicle will generate four trajectories of size 100, halving the flight time. Depending on N_v, the simulations are run for 100, 50, 33, 25, and 20 iterations, equivalent to 400 total sampling locations. All simulations start in the southwest corner of $\mathcal{S}$ with heading angles equally spaced between 0 and $\pi/2$.

Empirically, we find the inverse kernel to work well and do not need to set the minimum allowable distance between vehicles, and use the following K function:

$$K(x) = \frac{1}{\left(\frac{x}{0.5L}\right)^3}, \tag{5.10}$$

where L is the side length of $\mathcal{S}$, so that the length scale of $\mathcal{S}$ does not affect the value of K. As previously mentioned, we set α_1 and α_2 to be proportional to the expected utility obtained for the contending segment, and find empirically that a value of 0.02 for the proportion factors a_1 and a_2 balances the utility and collision penalty.

As before in Sections 3.4 and 4.5, we track the mean squared error over time for the goal of optimal reconstruction, and the probability of identifying the maximum over time for the goal of finding the maximum. A video for each simulation scenario is included in the supplementary files to this thesis.

5.4.1 Optimal reconstruction results

Simulated isotropic truth

Our first scenario uses the simulated isotropic truth. Figure 5.4 shows the behavior over time of the log-MSE over 200 experiments for each N_v. With the exception of $N_v = 5$ the simulations all stop with similar log-MSE, on average. This says that, if we cut the time by a factor but increase the number of vehicles by the same factor, the final outcome with respect to the optimal reconstruction goal is the same.

Figure 5.5 shows the median time required to reduce the starting mean squared error by various percentages. In the case of the simulated isotropic field, a 99% reduction in the MSE is achieved for all vehicle numbers. When increasing the number of vehicles to two or three vehicles, the reduction time is linear. However, the relatively flat segments between three and five vehicles indicate that further increasing the number of vehicles highly diminishes reductions in MSE.

The next figures compare the previous results to the setting of no collision avoidance, which is equivalent to $\alpha_2 = 0$ in the collision penalty. Not having a

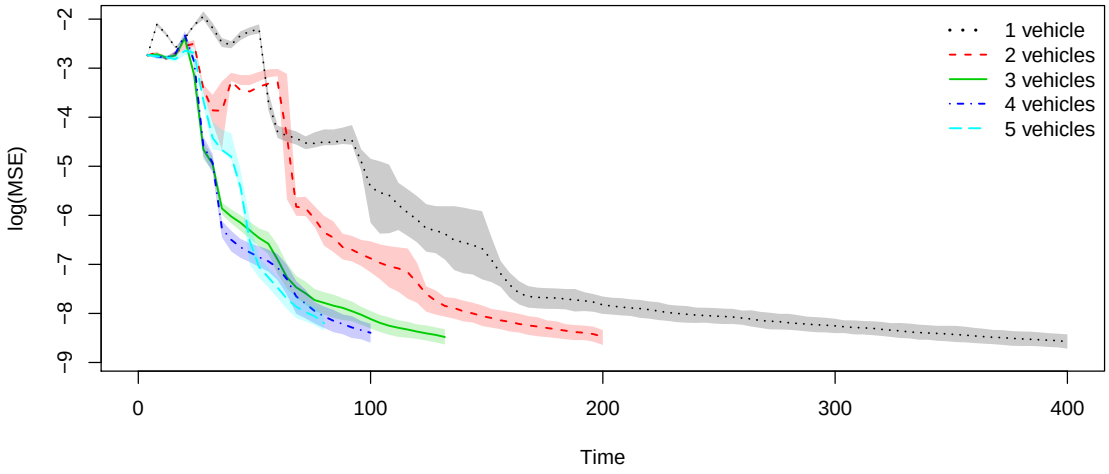


Figure 5.4: Reconstruction error as a function of time over 200 simulations with the simulated isotropic truth. Lines correspond to the mean reconstruction errors, while the shaded area indicates the 25th-75th percentile.

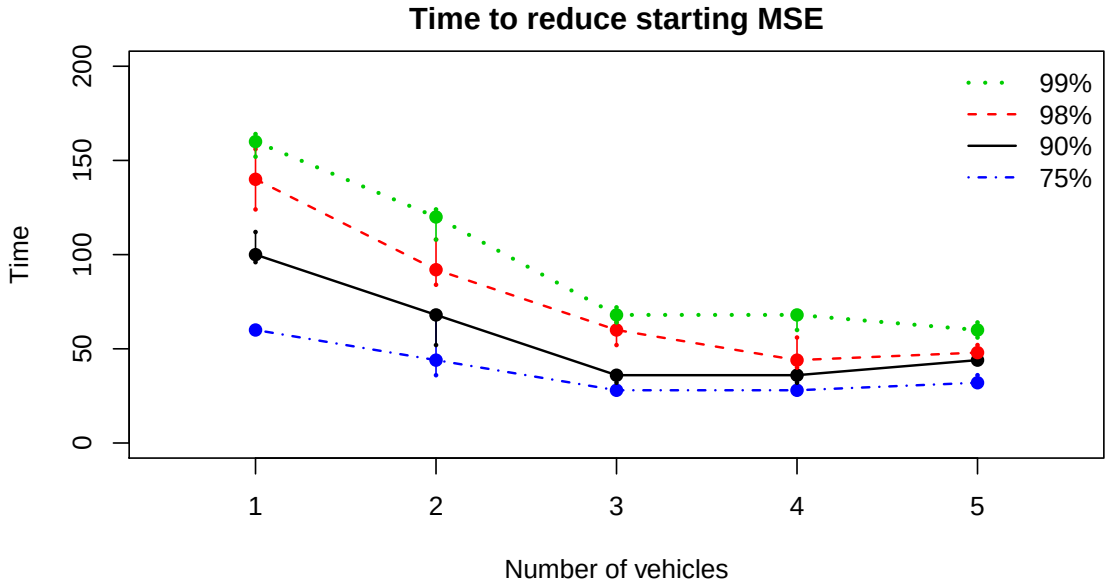


Figure 5.5: Median time required to reduce the starting mean squared error by various percentages for different N_v.

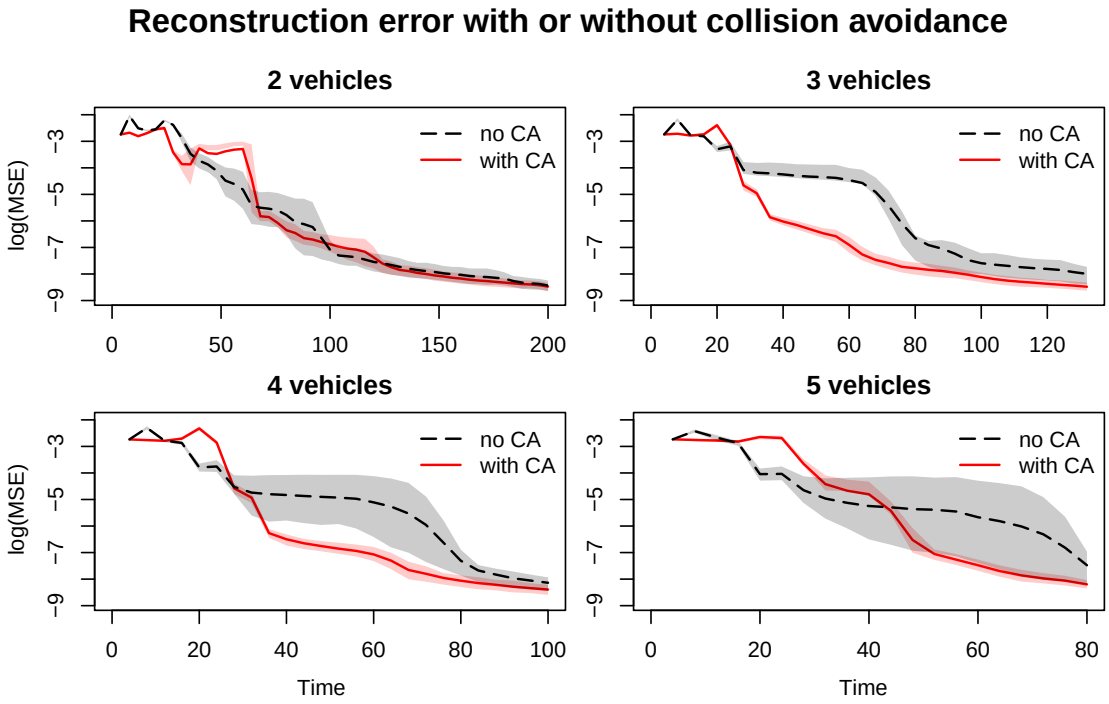
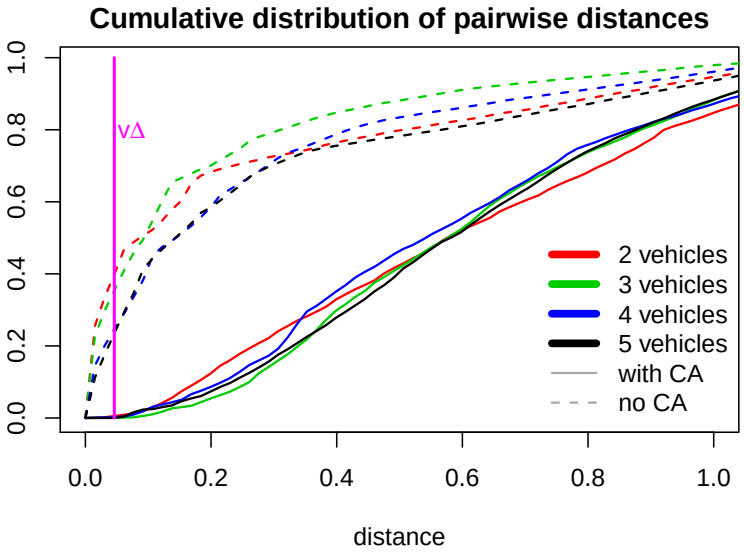


Figure 5.6: Reconstruction error comparing the effect of the collision avoidance policy in the scenario with the simulated isotropic truth.

collision avoidance policy results in an even less cooperative multi-vehicle system. The vehicles all have the same goal but do not affect the actions of the other vehicles. Some consequences of this independence are seen in Figure 5.6, which shows the log-MSE behavior of the 200 experiments with and without the collision avoidance policy (labelled as "with CA" and "no CA", respectively) in the same plot, for the same N_v. One would expect that including the collision avoidance penalty would result in reduced performance due to selecting suboptimal trajectory segments (with respect to $\tilde{U}$) to avoid collisions. However, in the cases of $N_v = 3, 4$, and 5, including collision avoidance results in better performance. With the exception of the beginning time period, the average log-MSE of "with CA" is remarkably lower, as is the variability across experiments. These results can be explained by the setting of the simulations. Both the true field (shown in Fig-



(a)

Pr(distance $< v\Delta$)		
N_v	no CA	with CA
2	0.39907	0.00522
3	0.35362	0.00048
4	0.24625	0.00275
5	0.23514	0.00011

(b)

Figure 5.7: Distribution of the distance between vehicles, with or without the collision avoidance policy, in the simulated scenario.

ure 3.4) and vehicles' initial conditions are symmetric across the 45° diagonal of $\mathcal{S}$. Because each vehicle has the same goal of optimal reconstruction and the same planning instructions (which ignore the other vehicles), the vehicles can end up planning and following collocated trajectories (i.e., merged trajectories that visit the same location at around the same time) for a period of time. This negates the benefit of having additional vehicles.

Figure 5.7 compares the distributions of the pairwise distances between vehicles at sampling locations

$$\left\|\mathbf{Q}^{\ell}(i\Delta) - \mathbf{Q}^{r}(i\Delta)\right\|, \quad 1 \le \ell < r \le N_v, \quad i = 1, 2, \ldots \tag{5.11}$$

The cumulative distributions of distances resulting from "with CA" and "no CA", for all 200 simulations for each $N_v = 2, 3, 4, 5$, are shown in Figure 5.7a. We can set a threshold distance of $v\Delta$, which is the distance between two consecutive

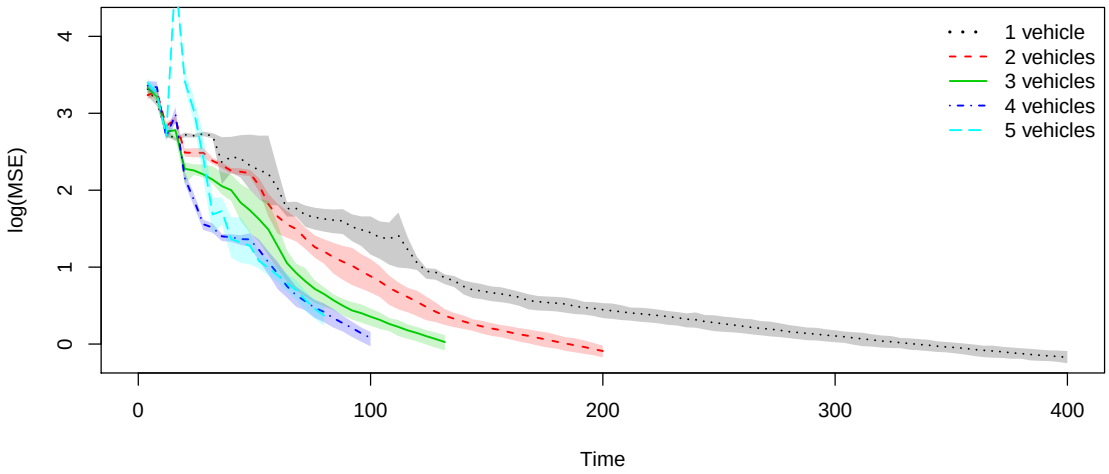


Figure 5.8: Reconstruction error as a function of time over 200 simulations with the temperature data. Lines correspond to the mean reconstruction errors, while the shaded area indicates the 25th-75th percentile.

sampling locations, shown as the vertical line in magenta, to be at risk for collision. Figure 5.7b lists the empirical probabilities of pairwise distances being less than $v\Delta$. Without collision avoidance, the probability of distance being less than the threshold $v\Delta$ is in the range 0.35 to 0.4 for two and three vehicles, and around 0.24 for four and five vehicles. With collision avoidance, the probabilities range from 0.00011 to 0.00522. Including a collision avoidance policy remarkably reduces the probability of vehicles being too close and drives up the inter-vehicle distances in general, which is desirable for exploring the region for optimal reconstruction.

Temperature truth

The true field is depicted in Figure 4.2a, which is different from the simulated isotropic truth in spatial scale and lacks symmetry. Figure 5.8 shows the average log-MSE for simulations using Algorithm 4. The average log-MSE for $N_v = 1, 2, 3, 4$ follow a similar pattern as in the scenario with the simulated truth. For these vehicle numbers, the end log-MSE levels are the same when we include additional vehicles and cut the flight time by the same factor. For $N_v = 5$, the

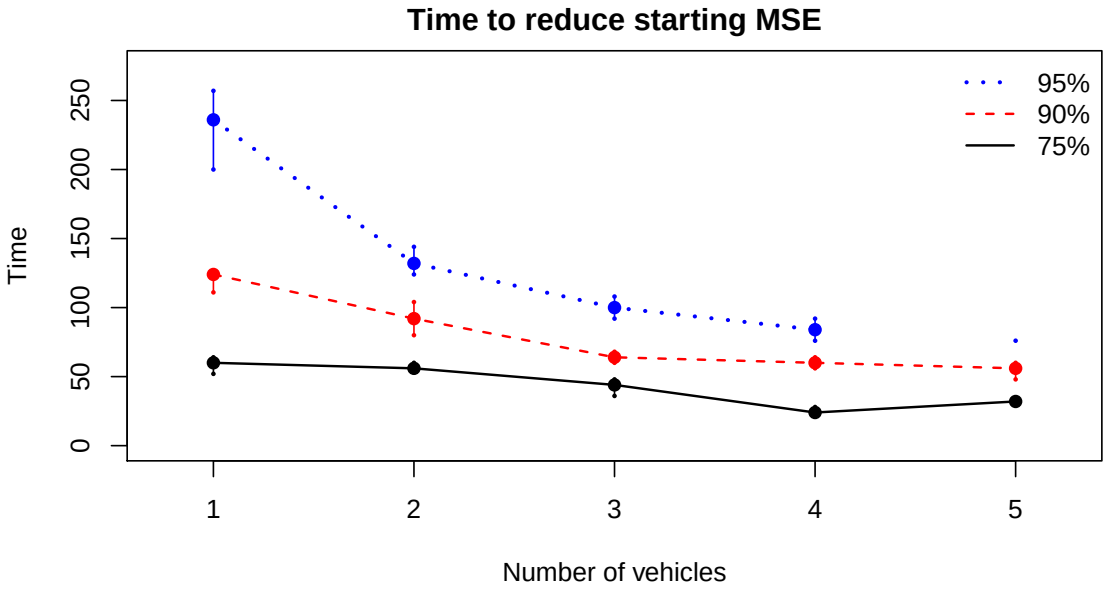


Figure 5.9: Median time required to reduce the starting mean squared error by various percentages for different N_v.

log-MSE actually increases in the beginning of the simulations. This is because the five vehicles' starting positions are clustered in the same southwest corner of the region, where the vehicles' trajectories are dominated by collision avoidance, and some time is required before the expected utility term draws the vehicles out of "collision avoidance paralysis". See the supplemental files for an example of this situation. Figure 5.9 shows the time to reduce the starting MSE. With the exception of $N_v = 5$, the majority of simulations were able to reduce the starting MSE by 95%.

For comparison to experiments without collision avoidance, Figure 5.10 shows the behavior of log-MSE from "with CA" side-by-side with "no CA". As in the previous scenario, the variability across experiments is slightly higher for "no CA". For two and three vehicles, the log-MSE behavior is comparable between "with CA" and "no CA". At four vehicles, there is a small spike at $t = 12$ for "with CA" that temporarily results in higher log-MSE compared to "no CA" for four iterations.

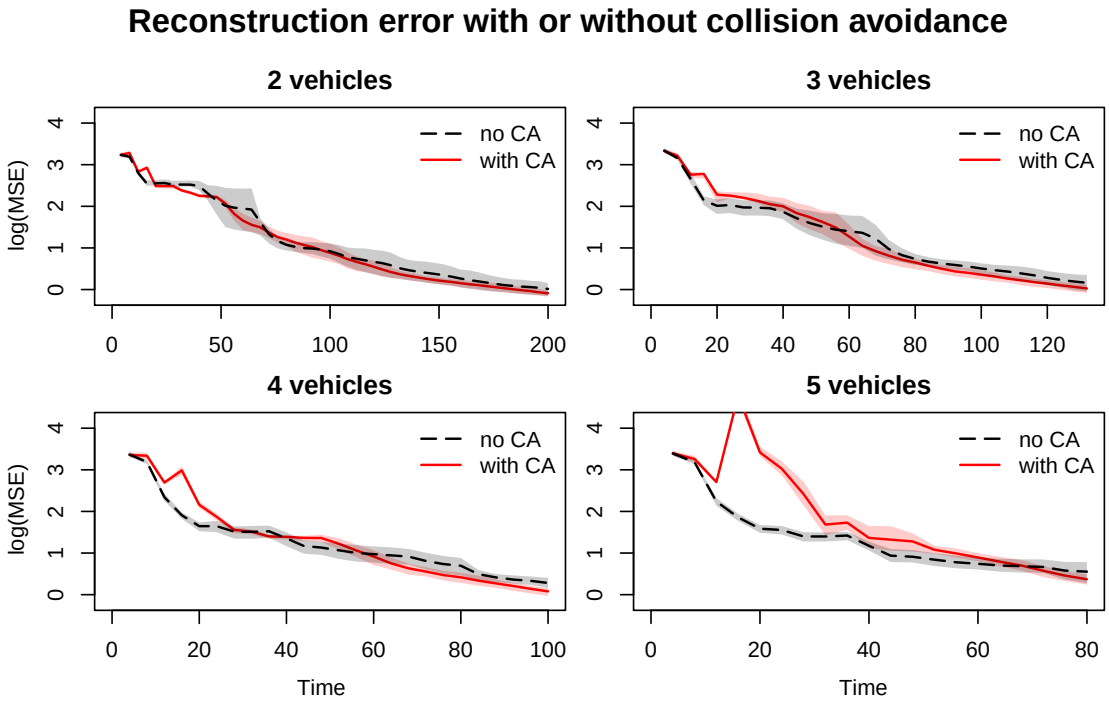
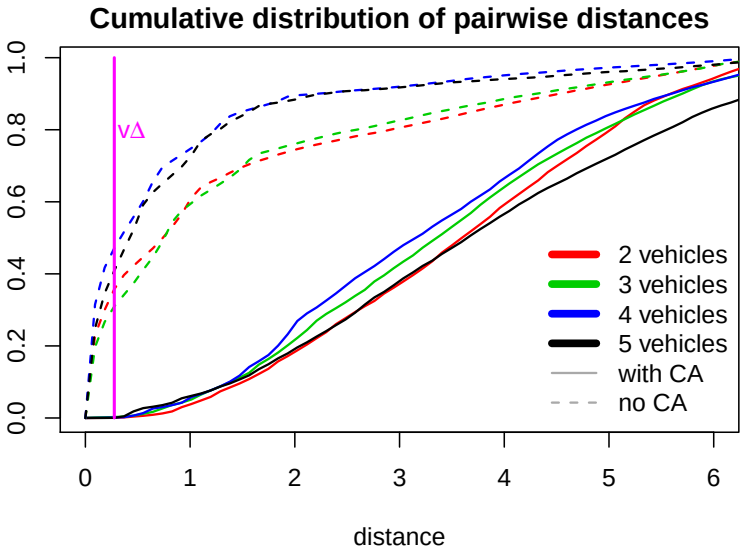


Figure 5.10: Reconstruction error comparing the effect of the collision avoidance policy in the scenario with the temperature data.

At five vehicles, this spike is very pronounced and the higher log-MSE persists until about halfway through the simulation time. These spikes near the beginning of the simulations, caused by the collision avoidance policy, are explained by the vehicles not moving out of the starting corner (and generating increasingly inaccurate estimates of the field due to the reinforcement from repeated sampling near the same locations) until the exploration term in the utility takes over. This behavior occurs more in the temperature scenario possibly because the southwest corner is relatively flat in the temperature truth compared to the isotropic truth.

The CDFs for pairwise inter-vehicle distances for the vehicles in "with CA" and "no CA", shown in Figure 5.11a, are similarly grouped as in the isotropic scenario. With collision avoidance, the probabilities of the distances being under $v\Delta$ are similarly small, all less than 0.003. Without, the probabilities are similarly much



(a)

Pr(distance $< v\Delta$)		
N_v	no CA	with CA
2	0.35757	0.00075
3	0.30948	0.00259
4	0.47131	0.00128
5	0.40895	0.00036

(b)

Figure 5.11: Distribution of the distance between vehicles, with or without the collision avoidance policy, in the temperature data scenario.

larger, with probabilities of between 0.30 and 0.36 for two and three vehicles, and greater than 0.4 (which is an increase of more than 50% compared to the isotropic case) for four and five vehicles.

Organic carbon truth

Our last evaluation scenario is with the organic carbon data, shown in Figure 4.12a. Figure 5.12 shows the log-MSE for 200 experiments for $N_v = 1, 2, 3, 4, 5$. For the first twenty time points, the log-MSE from one vehicle is the least, on average with very little variability. After $t = 20$, the log-MSE corresponding to more vehicles are lower and continue to decrease at a faster rate. This behavior is different from both the isotropic and temperature data in Figures 5.4 and 5.8, where the log-MSE from one vehicle starts off either higher or is the same as the log-MSE from higher vehicle numbers. Similar to the other scenarios, Figure 5.12 shows that increasing the number of vehicles improves performance, by almost

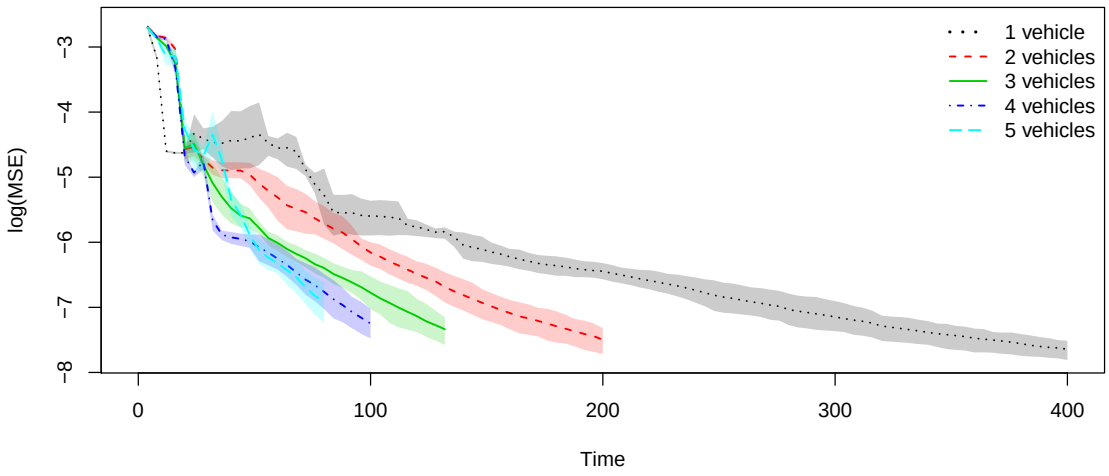


Figure 5.12: Reconstruction error as a function of time over 200 simulations with the carbon concentration data. Lines correspond to the mean reconstruction errors, while the shaded area indicates the 25th-75th percentile.

the same factor as the increase in the number of vehicles, up until $N_v = 5$.

Figure 5.13 shows the time required to reduce the starting MSE. Almost all simulations for all vehicle numbers achieve a reduction of 98%, and reductions of 99% are achieved by most simulations for $N_v = 1, 2$, and 3. The rate of the reductions as N_v increases is the highest going from one vehicle to two, and is slightly slower as more vehicles are introduced, particular for higher reduction percentages.

Figure 5.14 compares the effect of the collision avoidance penalty. For all $N_v = 2, 3, 4, 5$, "no CA" starts off with a large drop in log-MSE that is comparable to that of the initial lower log-MSE for $N_v = 1$ in Figure 5.12. This highly suggests that the initial higher log-MSE for $N_v = 2, 3, 4, 5$ is completely explained by including collision avoidance. After a later large drop in the log-MSE of "with CA" that surpasses "no CA" at around $t = 20$, the performance of "with CA" continues to be better for the remaining of the simulation time. Coupled with a higher variability across the 200 simulations for "no CA", this suggests that there may be some collocation in the trajectories without collision avoidance.

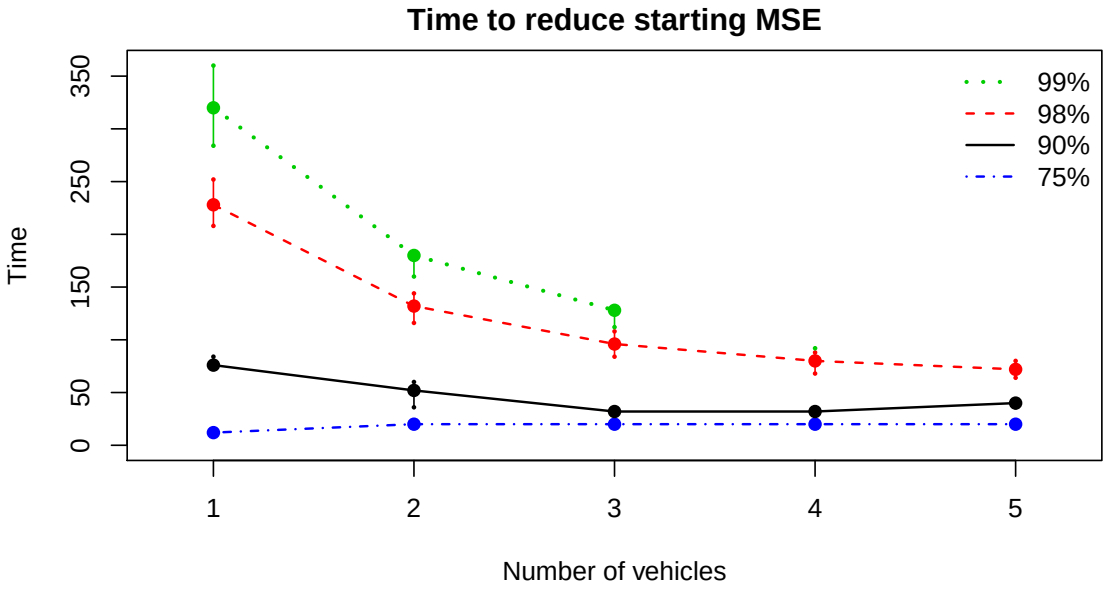


Figure 5.13: Median time required to reduce the starting mean squared error by various percentages for different N_v.

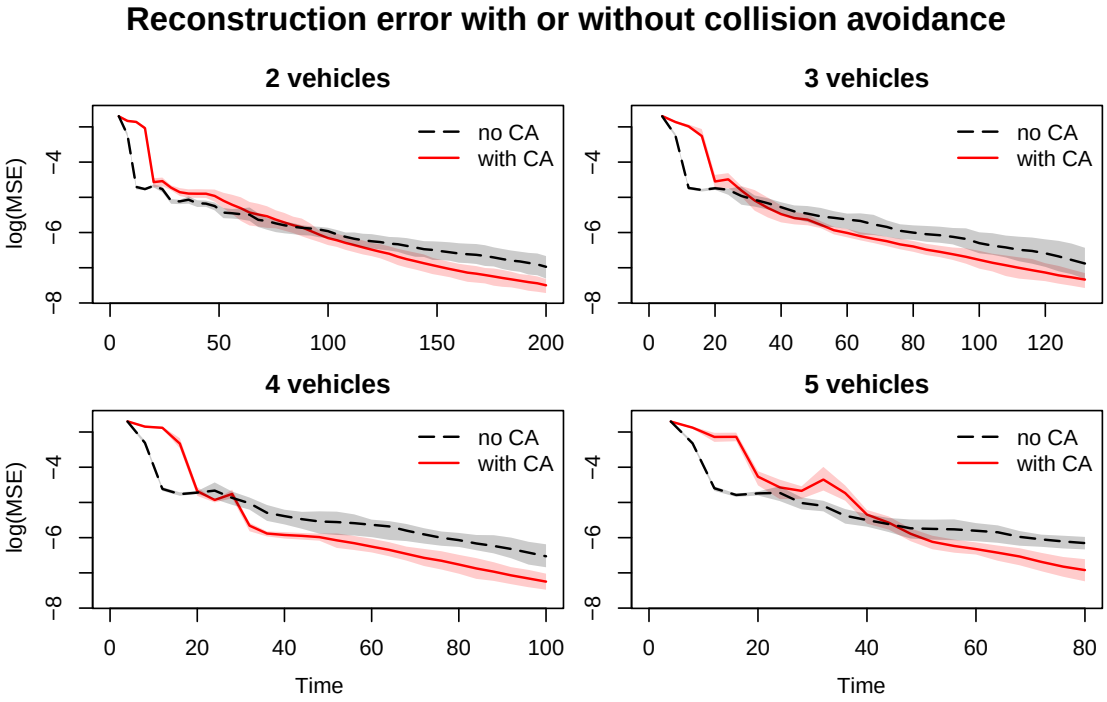
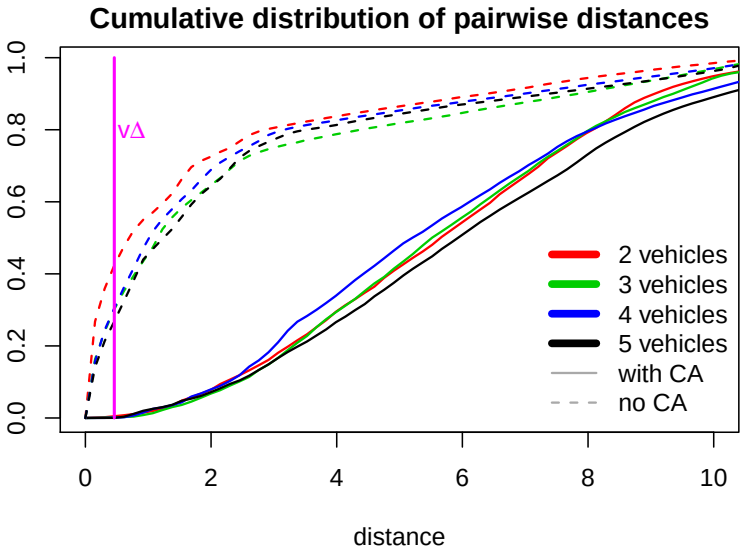


Figure 5.14: Reconstruction error comparing the effect of the collision avoidance policy in the scenario with the carbon concentration data.



(a)

Pr(distance $< v\Delta$)		
N_v	no CA	with CA
2	0.42357	0.00482
3	0.30165	0.00182
4	0.30472	0.00097
5	0.26809	0.00028

(b)

Figure 5.15: Distribution of the distance between vehicles, with or without the collision avoidance policy, in the carbon data scenario.

Figure 5.15 contains information on the distributions of the pairwise inter-vehicle distances for $N_v = 2, 3, 4, 5$ with and without collision avoidance. The distributions are similar to the previous two scenarios, and the effectiveness of the collision avoidance penalty to reduce the incidence of small inter-vehicle distances is shown.

5.4.2 Finding the maximum results

We also have preliminary results for the goal of finding the maximum. Results using the isotropic truth for 200 simulations for $N_v = 1, 2, 3, 4, 5$ and $c = 1, 1.05, 1.1$ are shown in Figure 5.16. As in the optimal reconstruction experiments, the simulations are run for lengths of time so that the total number of observations collected by all the vehicles is the same. Within each panel of Figure 5.16, the probability curves shift up and to the left as we increase the number of vehicles from $N_v = 1$ to $N_v = 4$. Similar to the optimal reconstruction simu-

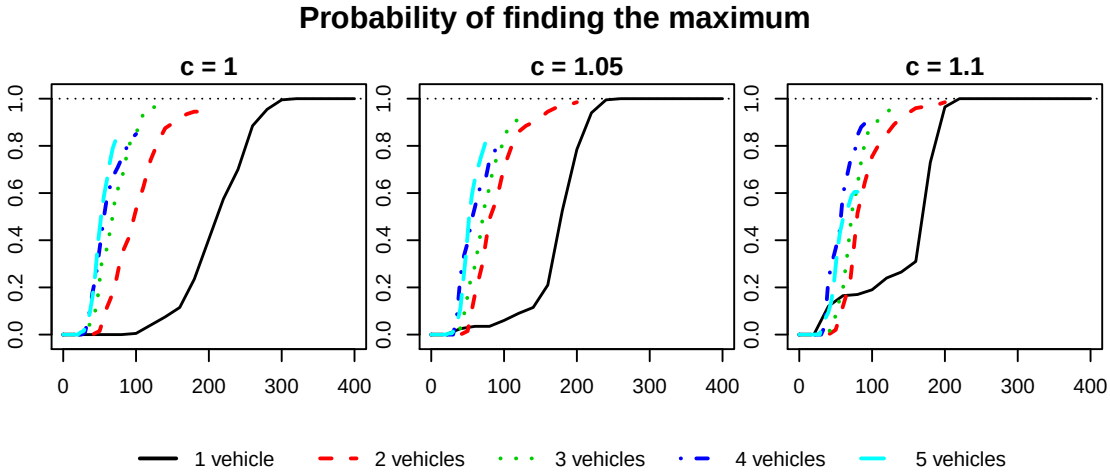
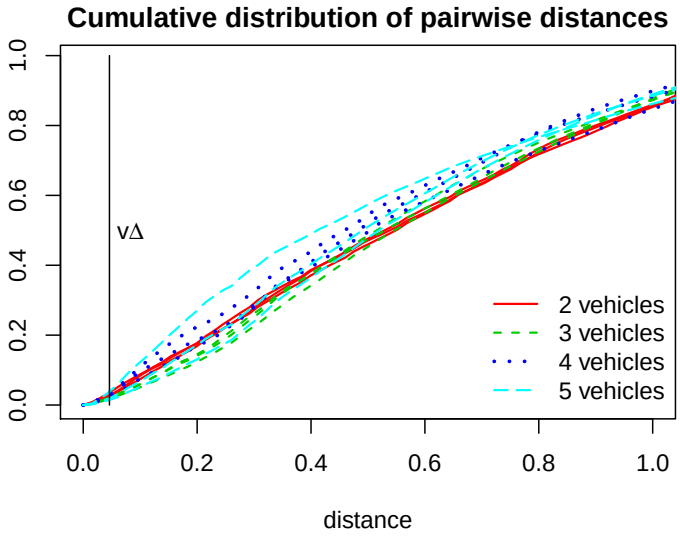


Figure 5.16: Comparing the probability of locating the maximum using $N_v = 1, 2, 3, 4, 5$ vehicles for three values of the exploration-exploitation parameter c.

lations, increasing the number of vehicles to five no longer provides performance improvement. Varying the exploration-exploitation parameter c affects $N_v = 1$ the most, so that almost all of the 200 simulations find the maximum by times $t = 300, 250, 200$ for $c = 1, 1.05, 1.1$ respectively. However, tuning c does not dramatically speed up finding the maximum for more vehicles; and in particular, for $c = 1.1$ and $N_v = 5$, performance declines. As discussed in Section 2.2.2, increasing c results in more exploration by reducing the time spent in local maxima. Being stuck in local maxima is more concerning for fewer vehicles because there are fewer vehicles remaining to find other maxima. With more vehicles that already follow dispersive trajectories from collision avoidance, increasing c is less beneficial.

The distributions of pairwise distances are shown in Figure 5.17a, and the probabilities of the distances being less than the threshold $v\Delta$ are provided. Across N_v and c, the probabilities do not seem to follow any pattern and are all greater by approximately one order of magnitude than those resulting from optimal reconstruction, shown in Figure 5.7b. This increase in the probabilities can be



(a)

Pr(distance $< v\Delta$)			
N_v	$c = 1$	$c = 1.05$	$c = 1.1$
2	0.027	0.026	0.035
3	0.018	0.020	0.021
4	0.032	0.037	0.025
5	0.014	0.019	0.038

(b)

Figure 5.17: Distribution of the distance between vehicles, combined for three values of the exploration-exploitation parameter c, with the simulated isotropic truth.

explained by the different utility functions for finding the maximum and optimal reconstruction. For finding the maximum, high utility is usually concentrated at a few small regions that can attract multiple vehicles simultaneously. For optimal reconstruction, utility is spread out and does not lead to converging trajectories to the same degree.

Simulations without collision avoidance were not run. We expect that the distribution of inter-vehicle distances when there is no collision avoidance to be similar (or higher, due to the convergent nature of trajectories that seek the maximum) to those shown in Figure 5.7a for "no CA".

5.5 Conclusions

We have developed an extension of our optimal trajectory algorithm for multiple vehicles that uses collective information to estimate the field and individual

trajectory planning for each vehicle. As a result, computation time per vehicle does not increase as more vehicles are added, and in theory, our framework can include a large number of vehicles. However in practice, as shown in the simulations, performance improvement scales (almost) linearly up to four vehicles, and stops scaling at five vehicles. We think that another approach would be more appropriate for a higher number of vehicles, such as one including swarm dynamics. Nonetheless, for the goals being considered, and focusing on the speedup of achieving the coals, we think that allowing the vehicles to go wherever and independently (except for collision avoidance) makes sense.

Chapter 6

Conclusion

6.1 Summary

This dissertation makes a contribution to the problem of automatic and adaptive trajectory generation for autonomous vehicles for environmental exploration. It develops and demonstrates the advantages of using reward functions that naturally trade off exploration and exploitation, in particular a new utility for the optimal reconstruction goal that minimizes reconstruction error. With the development of incorporation methods capable of handling different types of off-vehicle sensor information types and additional vehicles in a partially centralized manner, the efficacy of the new utility function persists throughout the multimodal and multi-vehicle extensions in this work.

6.2 Directions for future work

The following describes several suggestions for future work directions that address the limitations and areas of improvement of the current work.

6.2.1 Reimplementation on a real system

Section 3.5 presents an implementation of the trajectory generation framework to find the location of maximum radio signal by an autonomous ground vehicle. As discussed earlier, location inaccuracy from GPS and Mission Planner caused the largest discrepancies between the rover's trajectories and those from simulation. Another area of improvement is providing guarantees that computation costs do not decrease performance[1].

A reimplementation should have the exploration goal of optimal reconstruction, and we must be aware that optimal reconstruction takes more time per iteration for selecting the next segment compared to finding the maximum. The limitations of GPS and Mission Planner can be addressed by either (1) use a specialized localization system[2] that does not rely on GPS, or (2) deploy in a larger spatial region. Assuming the latter is chosen, a larger region allows for a longer sampling period.

To reduce total computation time, we take advantage of the multiple core architecture of the onboard computer. At least one core is dedicated to updating hyperparameters, and instead of updating hyperparameters every ten or so iterations, the core(s) can continuously update. Another two cores (one for each turning direction) then find the optimal trajectory segments using the most up-to-date hyperparameter estimate available, leaving the cost of segment optimization as the limiting factor. The following procedure suggests a reasonable sampling period:

[1] At the time of the rover test, we did not know if there would be an issue with computation time. Using the times in Section 3.4.3, the maximum time required for hyperparameter estimation is approximately 0.74 seconds for 208 observations, while the rover's sampling period was approximately three minutes (2Hz subsampled by an approximate factor of six.)

[2] An example is a motion capture system, which is reasonable for validation but not for a real field deployment.

1. Set the maximum allowable distance between two consecutive sampling locations and calculate the travel time $T_{\max}$ between them.
2. Compute the segment optimization cost curve shown in Figure 3.16, and choose the number of observations that correspond to $T_{\max}$. Recall that each iteration contains k observations, and the growth of the cost curve depends on k. Denote this number of observations as $n_{\max}$. When computing the cost curve, be consistent with the number of cores (and threads, if also using OpenMP) used.
3. Choose $N < n_{\max}$ either as a stopping rule or subsample size. With the latter, the vehicle continues to explore the region, but only N (randomly chosen, for simplicity) observations are used to compute the next optimal segment. Randomization recurs before every optimization.

With this procedure, after the vehicle has visited the most recent k locations and is computing the next optimal segment, the vehicle should follow a straight line so as to be most ready to head to the first location in the resulting optimal segment. An alternative procedure starts planning the next segment before all locations on the current segment are visited, allowing for longer computation time and more trajectory continuity. Because computation time increases without bound per iteration, a similar procedure that selects a max (sub)sample size N is also required.

6.2.2 Sampling at very high frequencies

Sampling at a very high frequency quickly amasses a large data history, and computations using the entire data history take too long for real-time implementation. The procedure in the previous section is one method to resolve large data histories, and it can be improved by efficient subsetting techniques [50]. Reduced

rank and other approximate representations [63] for Gaussian processes also reduce size. A faster $O(n \log^2 n)$ method for matrix inversions [2] can directly reduce computation time.

Appendix A

A.1 Expected reconstruction error if including observation at s

The expected utility function is (the negative of) the mean squared prediction error, when including $\mathbf{s}$, evaluated over all $\mathbf{x} \in \mathcal{S}$,

$$\tilde{U}(\mathbf{s}) = -\int_{\mathcal{S}} \int \int \left[f(\mathbf{x}) - \mu_{\mathbf{y}_{1:n}, y(\mathbf{s})}(\mathbf{x}) \right]^2 p\left(y(\mathbf{s}),\, f \mid \mathbf{y}_{1:n}\right) \mathrm{d}y(\mathbf{s})\, \mathrm{d}f\, \mathrm{d}\mathbf{x}. \tag{A.1}$$

The inner two integrals are the sum of three terms:

$$\int \int f(\mathbf{x})^2\, p(y(\mathbf{s}) \mid f, \mathbf{y}_{1:n})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}y(\mathbf{s})\, \mathrm{d}f, \tag{A.2}$$

$$-2 \int \int f(\mathbf{x})\, \mu_{\mathbf{y}_{1:n}, y(\mathbf{s})}(\mathbf{x})\, p(y(\mathbf{s}) \mid f, \mathbf{y}_{1:n})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}y(\mathbf{s})\, \mathrm{d}f, \tag{A.3}$$

$$\int \int \mu_{\mathbf{y}_{1:n}, y(\mathbf{s})}(\mathbf{x})^2\, p(y(\mathbf{s}) \mid f, \mathbf{y}_{1:n})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}y(\mathbf{s})\, \mathrm{d}f. \tag{A.4}$$

Integral (A.2) evaluates to

$$\begin{aligned}
&\int\int f(\mathbf{x})^2\, p(y(\mathbf{s}) \mid f, \mathbf{y}_{1:n})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}y(\mathbf{s})\, \mathrm{d}f \\
&\quad = \int f(\mathbf{x})^2\, p(f \mid \mathbf{y}_{1:n}) \int p(y(\mathbf{s}) \mid f, \mathbf{y}_{1:n})\, \mathrm{d}y(\mathbf{s})\, \mathrm{d}f \\
&\quad = \int f(\mathbf{x})^2\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}f \\
&\quad = \mathrm{Var}\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}\} + \mathrm{E}\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}\}^2 = \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) + \mu^2_{\mathbf{y}_{1:n}}(\mathbf{x}). \qquad \text{(A.5)}
\end{aligned}$$

For integral (A.3), consider the joint distribution of $f(\mathbf{x})$, $\mathbf{y}_{1:n}$, and $y(\mathbf{s})$

$$\begin{bmatrix} f(\mathbf{x}) \\ \mathbf{y}_{1:n} \\ y(\mathbf{s}) \end{bmatrix} \sim \mathsf{N}\left(\begin{bmatrix} m(\mathbf{x}) \\ \mathbf{m}_{1:n} \\ m(\mathbf{s}) \end{bmatrix}, \begin{bmatrix} \tau^2 & \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T} & C(\mathbf{x},\mathbf{s}) \\ \mathbf{c}_{1:n}(\mathbf{x}) & \mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n & \mathbf{c}_{1:n}(\mathbf{s}) \\ C(\mathbf{s},\mathbf{x}) & \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T} & \tau^2 + \sigma^2 \end{bmatrix} \right), \qquad \text{(A.6)}$$

and denote

$$\begin{bmatrix} \mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n & \mathbf{c}_{1:n}(\mathbf{s}) \\ \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T} & \tau^2 + \sigma^2 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{\Gamma}_{11} & \boldsymbol{\Gamma}_{12} \\ \boldsymbol{\Gamma}_{21} & \boldsymbol{\Gamma}_{22} \end{bmatrix}, \qquad \text{(A.7)}$$

with

$$\begin{aligned}
\boldsymbol{\Gamma}_{11} &= \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n - \mathbf{c}_{1:n}(\mathbf{s}) \left(\tau^2 + \sigma^2\right)^{-1} \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T}\right]^{-1} \quad \text{or} \\
&= \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} + \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \mathbf{c}_{1:n}(\mathbf{s}) \boldsymbol{\Gamma}_{22} \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T} \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \\
\boldsymbol{\Gamma}_{21} &= -\left[\tau^2 + \sigma^2\right]^{-1} \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T} \boldsymbol{\Gamma}_{11} \quad \text{or} \quad -\boldsymbol{\Gamma}_{22} \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T} \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \\
\boldsymbol{\Gamma}_{12} &= -\left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \mathbf{c}_{1:n}(\mathbf{s}) \boldsymbol{\Gamma}_{22} \quad \text{or} \quad -\boldsymbol{\Gamma}_{11} \mathbf{c}_{1:n}(\mathbf{s}) \left[\tau^2 + \sigma^2\right]^{-1} \\
\boldsymbol{\Gamma}_{22} &= \left\{\tau^2 + \sigma^2 - \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T} \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \mathbf{c}_{1:n}(\mathbf{s})\right\}^{-1} = \left[\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \sigma^2\right]^{-1}
\end{aligned}$$

from block matrix inversion. Then

$$
\begin{aligned}
\boldsymbol{\mu}_{\mathbf{y}_{1:n},y(\mathbf{s})}(\mathbf{x}) &= \mathrm{E}\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}, y(\mathbf{s})\} \\
&= m(\mathbf{x}) + \begin{bmatrix} \mathbf{c}_{1:n}(\mathbf{x})^{\mathsf{T}} & C(\mathbf{x},\mathbf{s}) \end{bmatrix} \begin{bmatrix} \mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n & \mathbf{c}_{1:n}(\mathbf{s}) \\ \mathbf{c}_{1:n}(\mathbf{s})^{\mathsf{T}} & \tau^2 + \sigma^2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}_{1:n} - \mathbf{m}_{1:n} \\ y(\mathbf{s}) - m(\mathbf{s}) \end{bmatrix} \\
&= m(\mathbf{x}) + \underbrace{\left[\mathbf{c}_{1:n}(\mathbf{x})^{\mathsf{T}} \boldsymbol{\Gamma}_{11} + C(\mathbf{x},\mathbf{s}) \boldsymbol{\Gamma}_{21}\right]}_{\mathbf{a}_1(\mathbf{x},\mathbf{s})^{\mathsf{T}}} [\mathbf{y}_{1:n} - \mathbf{m}_{1:n}] \\
&\qquad + \underbrace{\left[\mathbf{c}_{1:n}(\mathbf{x})^{\mathsf{T}} \boldsymbol{\Gamma}_{12} + C(\mathbf{x},\mathbf{s}) \boldsymbol{\Gamma}_{22}\right]}_{\mathbf{a}_2(\mathbf{x},\mathbf{s})} [y(\mathbf{s}) - m(\mathbf{s})]. \qquad \text{(A.8)}
\end{aligned}
$$

Substituting (A.8) into integral (A.3) gives

$$
\begin{aligned}
-2 \int \int f(\mathbf{x}) \left\{ m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^{\mathsf{T}} [\mathbf{y}_{1:n} - \mathbf{m}_{1:n}] + \mathbf{a}_2(\mathbf{x},\mathbf{s}) [y(\mathbf{s}) - m(\mathbf{s})] \right\} \times \\
p(y(\mathbf{s}) \mid f, \mathbf{y}_{1:n})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}y(\mathbf{s})\, \mathrm{d}f \\
= -2 \int f(\mathbf{x}) \left\{ m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^{\mathsf{T}} [\mathbf{y}_{1:n} - \mathbf{m}_{1:n}] + \mathbf{a}_2(\mathbf{x},\mathbf{s}) [f(\mathbf{s}) - m(\mathbf{s})] \right\} \times \\
p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}f.
\end{aligned}
$$

To compute the term involving $\int f(\mathbf{x})\, f(\mathbf{s})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}f$, consider

$$
\begin{bmatrix} f(\mathbf{x}) \\ f(\mathbf{s}) \end{bmatrix} \;\Bigg|\; \mathbf{y}_{1:n} \sim \mathsf{N}(mean, cov)
$$

with mean

$$
\begin{bmatrix} m(\mathbf{x}) \\ m(\mathbf{s}) \end{bmatrix} + \begin{bmatrix} \mathbf{c}_{1:n}(\mathbf{x})^{\mathsf{T}} \\ \mathbf{c}_{1:n}(\mathbf{s})^{\mathsf{T}} \end{bmatrix} \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n\right]^{-1} [\mathbf{y}_{1:n} - \mathbf{m}_{1:n}] = \begin{bmatrix} \mu_{\mathbf{y}_{1:n}}(\mathbf{x}) \\ \mu_{\mathbf{y}_{1:n}}(\mathbf{s}) \end{bmatrix}
$$

and covariance

$$\begin{bmatrix} \tau^2 & C(\mathbf{x},\mathbf{s}) \\ C(\mathbf{s},\mathbf{x}) & \tau^2 \end{bmatrix} - \begin{bmatrix} \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T} \\ \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T} \end{bmatrix} \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \begin{bmatrix} \mathbf{c}_{1:n}(\mathbf{x}) & \mathbf{c}_{1:n}(\mathbf{s}) \end{bmatrix} =$$
$$\begin{bmatrix} \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) & C(\mathbf{x},\mathbf{s}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T} \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \mathbf{c}_{1:n}(\mathbf{s}) \\ C(\mathbf{s},\mathbf{x}) - \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T} \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \mathbf{c}_{1:n}(\mathbf{x}) & \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) \end{bmatrix}$$

So

$$\begin{aligned} \int f(\mathbf{x})\, f(\mathbf{s})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}f &= \mathrm{Cov}\{f(\mathbf{x}), f(\mathbf{s}) \mid \mathbf{y}_{1:n}\} + \mathrm{E}\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}\}\, \mathrm{E}\{f(\mathbf{s}) \mid \mathbf{y}_{1:n}\} \\ &= C(\mathbf{x},\mathbf{s}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T} \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \mathbf{c}_{1:n}(\mathbf{s}) + \mu_{\mathbf{y}_{1:n}}(\mathbf{x})\, \mu_{\mathbf{y}_{1:n}}(\mathbf{s}), \end{aligned}$$

and integral (A.3) becomes

$$\begin{aligned} -2\Big[\mu_{\mathbf{y}_{1:n}}(\mathbf{x}) \left\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T} \left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right]\right\} \\ + \mathbf{a}_2(\mathbf{x},\mathbf{s}) \left\{C(\mathbf{x},\mathbf{s}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T} \left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1} \mathbf{c}_{1:n}(\mathbf{s}) + \mu_{\mathbf{y}_{1:n}}(\mathbf{x}) \left[\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right]\right\}\Big]. \end{aligned} \tag{A.9}$$

Integral (A.4):

$$
\begin{aligned}
&\int\int \mu_{\mathbf{y}_{1:n},y(\mathbf{s})}(\mathbf{x})^2\, p(y(\mathbf{s}) \mid f, \mathbf{y}_{1:n})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}y(\mathbf{s})\, \mathrm{d}f \\
&= \int\int \Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right] + \mathbf{a}_2(\mathbf{x},\mathbf{s})\left[y(\mathbf{s}) - m(\mathbf{s})\right]\Big\}^2 \\
&\qquad\qquad p(y(\mathbf{s}) \mid f, \mathbf{y}_{1:n})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}y(\mathbf{s})\, \mathrm{d}f \\
&= \int\int \Big[\Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right]\Big\}^2 \\
&\qquad\qquad + 2\Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right]\Big\}\Big\{\mathbf{a}_2(\mathbf{x},\mathbf{s})\left[y(\mathbf{s}) - m(\mathbf{s})\right]\Big\} \\
&\qquad\qquad + \Big\{\mathbf{a}_2(\mathbf{x})\left[y(\mathbf{s}) - m(\mathbf{s})\right]\Big\}^2\Big]\, p(y(\mathbf{s}) \mid f, \mathbf{y}_{1:n})\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}y(\mathbf{s})\, \mathrm{d}f \\
&= \int \Big[\Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right]\Big\}^2 \\
&\qquad\qquad + 2\Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right]\Big\}\Big\{\mathbf{a}_2(\mathbf{x},\mathbf{s})\left[f(\mathbf{s}) - m(\mathbf{s})\right]\Big\} \\
&\qquad\qquad + \mathbf{a}_2(\mathbf{x},\mathbf{s})^2\Big\{\sigma^2 + \left[f(\mathbf{s}) - m(\mathbf{s})\right]^2\Big\}\Big]\, p(f \mid \mathbf{y}_{1:n})\, \mathrm{d}f \\
&= \Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right]\Big\}^2 \\
&\qquad\qquad + 2\Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right]\Big\}\Big\{\mathbf{a}_2(\mathbf{x},\mathbf{s})\left[\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right]\Big\} \\
&\qquad\qquad + \mathbf{a}_2(\mathbf{x},\mathbf{s})^2\Big\{\sigma^2 + \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \left[\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right]^2\Big\}. \qquad \text{(A.10)}
\end{aligned}
$$

Combining (A.5), (A.9), and (A.10) gives the inner two integrals of $\tilde{U}(\mathbf{s})$:

$$
\begin{aligned}
&\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) + \mu^2_{\mathbf{y}_{1:n}}(\mathbf{x}) - 2\mu_{\mathbf{y}_{1:n}}(\mathbf{x})\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}]\} \\
&\quad - 2\mathbf{a}_2(\mathbf{x},\mathbf{s})\{C(\mathbf{x},\mathbf{s}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{c}_{1:n}(\mathbf{s}) + \mu_{\mathbf{y}_{1:n}}(\mathbf{x})\left[\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right]\} \\
&\quad + \{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}]\}^2 \\
&\quad + 2\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}]\}\,\{\mathbf{a}_2(\mathbf{x},\mathbf{s})[\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})]\} \\
&\quad + \mathbf{a}_2(\mathbf{x},\mathbf{s})^2\{\sigma^2 + \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + [\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})]^2\}.
\end{aligned}
$$

With a little rearrangement, the previous quantity is

$$\begin{aligned}
&\mu^2_{\mathbf{y}_{1:n}}(\mathbf{x}) + \Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}]\Big\}^2 + \mathbf{a}_2(\mathbf{x},\mathbf{s})^2\left[\boldsymbol{\mu}_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right]^2 \\
&\quad - 2\mu_{\mathbf{y}_{1:n}}(\mathbf{x})\Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}] + \mathbf{a}_2(\mathbf{x},\mathbf{s})\left[\boldsymbol{\mu}_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right]\Big\} \\
&\quad + 2\Big\{m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}]\Big\}\,\mathbf{a}_2(\mathbf{x},\mathbf{s})\left[\boldsymbol{\mu}_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right] \\
&\quad + \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) - 2\mathbf{a}_2(\mathbf{x},\mathbf{s})\Big\{C(\mathbf{x},\mathbf{s}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\left[\mathbf{C} + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{c}_{1:n}(\mathbf{s})\Big\} \\
&\quad + \mathbf{a}_2(\mathbf{x},\mathbf{s})^2\left[\sigma^2 + \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s})\right],
\end{aligned}$$

which further rearranges and combines to

$$\begin{aligned}
&\Big\{\mu_{\mathbf{y}_{1:n}}(\mathbf{x}) - m(\mathbf{x}) - \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right] - \mathbf{a}_2(\mathbf{x},\mathbf{s})\left[\boldsymbol{\mu}_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right]\Big\}^2 \\
&\quad + \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) - 2\mathbf{a}_2(\mathbf{x},\mathbf{s})\Big\{C(\mathbf{x},\mathbf{s}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\left[\mathbf{C} + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{c}_{1:n}(\mathbf{s})\Big\} \\
&\quad + \mathbf{a}_2(\mathbf{x},\mathbf{s})^2\left[\sigma^2 + \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s})\right].
\end{aligned} \tag{A.11}$$

With

$$\begin{aligned}
\mathbf{a}_2(\mathbf{x},\mathbf{s}) &= \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\boldsymbol{\Gamma}_{12} + C(\mathbf{x},\mathbf{s})\boldsymbol{\Gamma}_{22} \\
&= \Big\{C(\mathbf{x},\mathbf{s}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{c}_{1:n}(\mathbf{s})\Big\}\left[\sigma^2 + \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s})\right]^{-1},
\end{aligned}$$

the expression (A.11) for the inner integrals of $\tilde{U}(\mathbf{s})$ finally becomes

$$\begin{aligned}
&\Big\{\mu_{\mathbf{y}_{1:n}}(\mathbf{x}) - m(\mathbf{x}) - \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right] - \mathbf{a}_2(\mathbf{x},\mathbf{s})\left[\boldsymbol{\mu}_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right]\Big\}^2 \\
&\quad + \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) - \Big\{C(\mathbf{s},\mathbf{x}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{c}_{1:n}(\mathbf{s})\Big\}^2\left[\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \sigma^2\right]^{-1},
\end{aligned} \tag{A.12}$$

which leaves the following to be shown:

$$\mu_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x}) = m(\mathbf{x}) + \mathbf{a}_1(\mathbf{x},\mathbf{s})^\mathsf{T}\left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n}\right] + \mathbf{a}_2(\mathbf{s},\mathbf{x})\left[\mu_{\mathbf{y}_{1:n}}(\mathbf{s}) - m(\mathbf{s})\right], \quad \text{(A.13)}$$

$$\kappa^2_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x}) = \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) - \left\{C(\mathbf{x},\mathbf{s}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{c}_{1:n}(\mathbf{s})\right\}^2\left[\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \sigma^2\right]^{-1}. \quad \text{(A.14)}$$

$\mu_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x})$ is $\mu_{\mathbf{y}_{1:n},y(\mathbf{s})}(\mathbf{x})$ evaluated with $y(\mathbf{s}) = \hat{y}(\mathbf{s}) = \mathrm{E}\{y(\mathbf{s}) \mid \mathbf{y}_{1:n}\} = \mu_{\mathbf{y}_{1:n}}(\mathbf{x})$, so (A.13) holds. *From*(A.6), *compute*

$$\begin{aligned}
&\kappa^2_{\mathbf{y}_{1:n},\hat{y}(\mathbf{s})}(\mathbf{x}) = \mathrm{Var}\{f(x) \mid \mathbf{y}_{1:n}, \hat{y}(\mathbf{s})\} = \mathrm{Var}\{f(x) \mid \mathbf{y}_{1:n}, y(\mathbf{s})\} \\
&= \tau^2 - \begin{bmatrix}\mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T} & C(\mathbf{x},\mathbf{s})\end{bmatrix}\begin{bmatrix}\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n & \mathbf{c}_{1:n}(\mathbf{s}) \\ \mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T} & \tau^2 + \sigma^2\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{c}_{1:n}(\mathbf{x}) \\ C(\mathbf{s},\mathbf{x})\end{bmatrix} \\
&= \tau^2 - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\boldsymbol{\Gamma}_{11}\mathbf{c}_{1:n}(\mathbf{x}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\boldsymbol{\Gamma}_{12}C(\mathbf{s},\mathbf{x}) \\
&\qquad - C(\mathbf{x},\mathbf{s})\boldsymbol{\Gamma}_{21}\mathbf{c}_{1:n}(\mathbf{x}) - C(\mathbf{x},\mathbf{s})\boldsymbol{\Gamma}_{22}C(\mathbf{s},\mathbf{x}) \\
&= \tau^2 - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{x}) \\
&\quad - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{s})\left[\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \sigma^2\right]^{-1}\mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{x}) \\
&\quad + \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{s})\left[\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \sigma^2\right]^{-1}C(\mathbf{x},\mathbf{s}) \\
&\quad + C(\mathbf{x},\mathbf{s})\left[\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \sigma^2\right]^{-1}\mathbf{c}_{1:n}(\mathbf{s})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{x}) \\
&\quad - C(\mathbf{x},\mathbf{s})\left[\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \sigma^2\right]^{-1}C(\mathbf{s},\mathbf{x}) \\
&= \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) - \left\{\mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{s}) - C(\mathbf{x},\mathbf{s})\right\} \\
&\qquad \times \left[\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \sigma^2\right]^{-1}\left\{\mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{s}) - C(\mathbf{x},\mathbf{s})\right\},
\end{aligned}$$

which shows (A.14).

A.2 Expected improvement

For a random variable X that follows a normal distribution with mean μ and variance σ^2, and for a fixed value c, the expected improvement of X over c is

$$\begin{aligned}
\mathrm{E}\left(\max\{X-c,0\}\right) &= \int \max\{x-c,0\}\, p(x)\,\mathrm{d}x \\
&= \int_c^\infty [x-c]\, p(x)\,\mathrm{d}x \\
&= \int_c^\infty \frac{x-\mu}{\sigma}\sigma\, p(x)\,\mathrm{d}x + \int_c^\infty [\mu-c]\, p(x)\,\mathrm{d}x \\
&= \int_c^\infty \frac{x-\mu}{\sigma}\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,\mathrm{d}x + [\mu-c]\int_c^\infty p(x)\,\mathrm{d}x \\
&= \sigma\int_{\frac{c-\mu}{\sigma}}^\infty z\frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}}\,\mathrm{d}z + [\mu-c]\mathrm{Pr}(X>c) \\
&= \sigma\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{c-\mu}{\sigma}\right)^2} + [\mu-c]\mathrm{Pr}\left(Z>\frac{c-\mu}{\sigma}\right) \\
&= \sigma\phi\left(\frac{c-\mu}{\sigma}\right) + [\mu-c]\left[1-\Phi\left(\frac{c-\mu}{\sigma}\right)\right] \\
&= \sigma\phi\left(\frac{\mu-c}{\sigma}\right) + [\mu-c]\Phi\left(\frac{\mu-c}{\sigma}\right).
\end{aligned}$$

In the context of selecting the next vehicle location $\mathbf{s}$ that maximizes expected improvement, let

$$\begin{aligned}
X &= f(\mathbf{s}) \\
\mu &= \mu_{\mathbf{y}_{1:n}}(\mathbf{s}) \\
\sigma &= \kappa_{\mathbf{y}_{1:n}}(\mathbf{s}) \\
c &= y_{1:n}^{\max}.
\end{aligned}$$

A.3 Entropy and mutual information

In information theory, *entropy* represents the amount of information hidden in a random variable X and is defined as

$$\mathcal{H}(X) = -\mathrm{E}\left\{\log p(x)\right\} = -\int p(x)\, \log p(x)\, \mathrm{d}x. \tag{A.15}$$

If X is multivariate normal, $X \sim \mathsf{N}_n(\cdot, \Sigma)$,

$$\mathcal{H}(X) = \frac{1}{2}\log(|\Sigma|) + \frac{n}{2}\log(2\pi e).$$

Extending to two random variables X and Y, the *joint entropy* is

$$\mathcal{H}(X,Y) = -\mathrm{E}\left\{\log p(x,y)\right\} = -\int\int p(x,y)\, \log p(x,y)\, \mathrm{d}x\, \mathrm{d}y. \tag{A.16}$$

The *conditional entropy* is defined as

$$\mathcal{H}(X \mid Y) = -\mathrm{E}\left\{\log p(x \mid y)\right\} = -\int\int p(x,y)\, \log p(x \mid y)\, \mathrm{d}x\, \mathrm{d}y, \tag{A.17}$$

and the three definitions are related by

$$\begin{aligned}\mathcal{H}(X,Y) &\equiv \mathcal{H}(Y \mid X) + \mathcal{H}(X) \\ &\equiv \mathcal{H}(X \mid Y) + \mathcal{H}(Y).\end{aligned}$$

The *mutual information* of two random variables represents the amount of information contained in one random variable about the other random variable,

$$\mathcal{I}(X;Y) = \int\int p(x,y)\, \log \frac{p(x,y)}{p_X(x)\, p_Y(y)}\, \mathrm{d}x\, \mathrm{d}y, \tag{A.18}$$

and is related to entropy by the following equivalences

$$\begin{aligned}\mathcal{I}(X;Y) &\equiv \mathcal{H}(X) - \mathcal{H}(X \mid Y) \\ &\equiv \mathcal{H}(Y) - \mathcal{H}(Y \mid X) \\ &\equiv \mathcal{H}(X) + \mathcal{H}(Y) - \mathcal{H}(X,Y).\end{aligned}$$

Both entropy and mutual information can be used as criteria for selecting sampling locations.

The joint entropy is the information revealed by evaluating X and Y simultaneously or consecutively (by first evaluating the value of X, and then revealing the value of Y given that the value of X is known). This means that the conditional entropy $\mathcal{H}(Y \mid X)$ is a measure of what X does *not* say about Y, measuring the amount of uncertainty remaining after X is known. Let $\mathcal{G}$ be a grid of locations that covers $\mathcal{S}$, where each grid point is a possible sampling location, and denote $\mathbf{f}_{\mathcal{G}} = \{f(\mathbf{s}) : \mathbf{s} \in \mathcal{G}\}$. A possible strategy to select the next sampling location is

$$\begin{aligned}\mathbf{s}^* &= \arg\min_{\mathbf{s}\in\mathcal{G}} \mathcal{H}(\mathbf{f}_{\mathcal{G}\setminus\mathbf{s}} \mid f(\mathbf{s})) \\ &= \arg\min_{\mathbf{s}\in\mathcal{G}} \left[\mathcal{H}(\mathbf{f}_{\mathcal{G}}) - \mathcal{H}(f(\mathbf{s}))\right] \\ &= \arg\max_{\mathbf{s}\in\mathcal{G}} \mathcal{H}(f(\mathbf{s})) \\ &= \arg\max_{\mathbf{s}\in\mathcal{G}} \frac{1}{2}\left[\log \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \log 2\pi e\right],\end{aligned}$$

which reduces to finding the location with the largest posterior variance. This is known as the maximum entropy (D-optimal) solution and does not depend on the observed values $\mathbf{y}_{1:n}$.

A solution based on mutual information $\mathcal{I}(X;Y) = \mathcal{H}(Y) - \mathcal{H}(Y|X)$, inter-

preted as the amount of uncertainty in Y which is removed by knowing X, is

$$\begin{aligned}
\mathbf{s}^* &= \arg\max_{\mathbf{s}\in\mathcal{G}} \mathcal{I}(\mathbf{f}_{\mathcal{G}\setminus\mathbf{s}}; f(\mathbf{s})) \\
&= \arg\max_{\mathbf{s}\in\mathcal{G}} \left[\mathcal{H}(\mathbf{f}_{\mathcal{G}\setminus\mathbf{s}}) + \mathcal{H}(f(\mathbf{s})) - \mathcal{H}(\mathbf{f}_{\mathcal{G}})\right] \\
&= \arg\max_{\mathbf{s}\in\mathcal{G}} \left[\mathcal{H}(\mathbf{f}_{\mathcal{G}\setminus\mathbf{s}}) + \mathcal{H}(f(\mathbf{s}))\right] \\
&= \arg\max_{\mathbf{s}\in\mathcal{G}} \frac{1}{2}\left[\log|\boldsymbol{\Sigma}_{\mathcal{G}\setminus\mathbf{s}}| + [n_{\mathcal{G}}-1]\log(2\pi e) + \log\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s}) + \log 2\pi e\right] \\
&= \arg\max_{\mathbf{s}\in\mathcal{G}} \left[\log|\boldsymbol{\Sigma}_{\mathcal{G}\setminus\mathbf{s}}| + \log\kappa^2_{\mathbf{y}_{1:n}}(\mathbf{s})\right],
\end{aligned}$$

where $\boldsymbol{\Sigma}_{\mathcal{G}\setminus\mathbf{s}}$ is the posterior covariance matrix

$$\boldsymbol{\Sigma}_{\mathcal{G}\setminus\mathbf{s}} = \mathbf{C}_{\mathcal{G}\setminus\mathbf{s}} - \mathbf{C}^{\mathsf{T}}_{1:n,\mathcal{G}\setminus\mathbf{s}}\left[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n\right]^{-1}\mathbf{C}_{1:n,\mathcal{G}\setminus\mathbf{s}} \tag{A.19}$$

with

$$\begin{aligned}
[\mathbf{C}_{\mathcal{G}\setminus\mathbf{s}}]_{ij} &= C(\mathbf{x}_i,\mathbf{x}_j), \quad \mathbf{x}_i,\mathbf{x}_j \in \mathcal{G}\setminus\mathbf{s}, \\
[\mathbf{C}_{1:n,\mathcal{G}\setminus\mathbf{s}}]_{ij} &= C(\mathbf{s}_i,\mathbf{x}_j), \quad \mathbf{s}_i \in \{\mathbf{s}_1,\ldots,\mathbf{s}_n\}, \mathbf{x}_j \in \mathcal{G}\setminus\mathbf{s}.
\end{aligned}$$

For selecting a fixed number of locations $A \subset \mathcal{G}$ at once, the entropy criterion gives

$$\begin{aligned}
A^* &= \arg\max_{A\subset\mathcal{G}} \mathcal{H}(\mathbf{f}_A) \\
&= \arg\max_{A\subset\mathcal{G}} \log|\boldsymbol{\Sigma}_A|,
\end{aligned}$$

and the mutual information criterion gives

$$\begin{aligned} A^* &= \arg\max_{A \subset \mathcal{G}} \mathcal{I}(\mathbf{f}_{\mathcal{G} \setminus A}; \mathbf{f}_A) \\ &= \arg\max_{A \subset \mathcal{G}} \left[\log |\mathbf{\Sigma}_{\mathcal{G} \setminus A}| + \log |\mathbf{\Sigma}_A| \right] \end{aligned}$$

with $\mathbf{\Sigma}_{\mathcal{G} \setminus A}$ and $\mathbf{\Sigma}_A$ defined analogously to (A.19). Note that these are NP-hard problems, and approximate greedy or dynamic programming solutions are presented in [39] and [50].

A.4 P-algorithm

The P-algorithm [85] selects the location that maximizes the probability of an increase of δ_n:

$$\begin{aligned} s^* &= \arg\max_{\mathbf{s} \in \mathcal{S}} \Pr\left(f(\mathbf{s}) \geq y_{1:n}^{\max} + \delta_n \mid \text{data}\right) \\ &= \arg\max_{\mathbf{s} \in \mathcal{S}} \Pr\left(\frac{f(\mathbf{s}) - \mu_{\mathbf{y}_{1:n}}(\mathbf{s})}{\kappa_{\mathbf{y}_{1:n}}(\mathbf{s})} \geq \frac{y_{1:n}^{\max} + \delta_n - \mu_{\mathbf{y}_{1:n}}(\mathbf{s})}{\kappa_{\mathbf{y}_{1:n}}(\mathbf{s})}\right) \\ &= \arg\max_{\mathbf{s} \in \mathcal{S}} \Phi\left(\frac{y_{1:n}^{\max} + \delta_n - \mu_{\mathbf{y}_{1:n}}(\mathbf{s})}{\kappa_{\mathbf{y}_{1:n}}(\mathbf{s})}\right). \end{aligned}$$

A.5 Circular arc parametrization details

For continuity of position and velocity, $\mathbf{q}_j$ must satisfy

$$\mathbf{q}_j(0) = \mathbf{q}_{j-1}^*(k\Delta), \tag{A.20}$$

$$\dot{\mathbf{q}}_j(0) = \dot{\mathbf{q}}_{j-1}^*(k\Delta). \tag{A.21}$$

In order for the jth segment to start where the $\{j-1\}$th ends with the heading angle $\theta_{j-1} = \texttt{atan2}\left(\frac{\dot{q}^*_{j-1,2}(k\Delta)}{\dot{q}^*_{j-1,1}(k\Delta)}\right)$,

$$\mathbf{q}_j(t) = \mathbf{q}_{j-1}(k\Delta) + \begin{bmatrix} \cos(\theta_{j-1} - \frac{\pi}{2}) & -\sin(\theta_{j-1} - \frac{\pi}{2}) \\ \sin(\theta_{j-1} - \frac{\pi}{2}) & \cos(\theta_{j-1} - \frac{\pi}{2}) \end{bmatrix} \begin{bmatrix} \xi_{j,1}\cos(\xi_{j,2}t) - \xi_{j,1} \\ \xi_{j,1}\sin(\xi_{j,2}t) \end{bmatrix}.$$

The perhaps unexpected terms above are explained in the following:
$\begin{bmatrix} \xi_{j,1}\cos(\xi_{j,2}t) \\ \xi_{j,1}\sin(\xi_{j,2}t) \end{bmatrix}$ parametrizes a circle that begins at the point $(\xi_{j,1}, 0)$ with heading angle $\frac{\pi}{2}$, so we need to shift the first component by $-\xi_{j,1}$ and rotate by an angle of $\theta_{j-1} - \frac{\pi}{2}$ to have the desired heading of θ_{j-1}. This gives, with the help of angle sum identities for sine and cosine,

$$\mathbf{q}_j(t) = \begin{bmatrix} q^*_{j-1,1}(k\Delta) + \xi_{j,1}\cos(\xi_{j,2}t + \theta_{j-1} - \frac{\pi}{2}) - \xi_{j,1}\cos(\theta_{j-1} - \frac{\pi}{2}) \\ q^*_{j-1,2}(k\Delta) + \xi_{j,1}\sin(\xi_{j,2}t + \theta_{j-1} - \frac{\pi}{2}) - \xi_{j,1}\sin(\theta_{j-1} - \frac{\pi}{2}) \end{bmatrix}.$$

So $\xi_{j,1}, \ldots, \xi_{j,5}$ in the parametrization (3.9) are constrained by:

$$|\xi_{j,1}| \geq R_{\min}, \tag{A.22}$$

$$\xi_{j,2} = \frac{v}{\xi_{j,1}}, \tag{A.23}$$

$$\xi_{j,3} = \theta_{j-1} - \frac{\pi}{2}, \tag{A.24}$$

$$\xi_{j,4} = q^*_{j-1,1}(k\Delta) - \xi_{j,1}\sin\theta_{j-1}, \tag{A.25}$$

$$\xi_{j,5} = q^*_{j-1,2}(k\Delta) + \xi_{j,1}\cos\theta_{j-1}, \tag{A.26}$$

leaving one free parameter $\xi_{j,1}$, which corresponds to how much and in which direction the vehicle will turn, for optimization.

A.6 Expressions of $\mu_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ and $\kappa^2_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$

The evaluation of the objective function on a segment $\mathbf{q}_{j+1}$ for optimal reconstruction involves the terms

$$\mu_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x}) = \mathrm{E}\left\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}, \hat{\mathbf{y}}_{\mathbf{q}_{j+1}}\right\}, \tag{A.27}$$

$$\kappa^2_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x}) = \mathrm{Var}\left\{f(\mathbf{x}) \mid \mathbf{y}_{1:n}, \hat{\mathbf{y}}_{\mathbf{q}_{j+1}}\right\}. \tag{A.28}$$

For their explicit expressions, consider the distribution of $f(\mathbf{x})$, $\mathbf{y}_{1:n}$, and $\mathbf{y}_{\mathbf{q}_{j+1}}$:

$$\begin{bmatrix} f(\mathbf{x}) \\ \mathbf{y}_{1:n} \\ \mathbf{y}_{\mathbf{q}_{j+1}} \end{bmatrix} \sim \mathsf{N}\left(\begin{bmatrix} m(\mathbf{x}) \\ \mathbf{m}_{1:n} \\ \mathbf{m}_{\mathbf{q}_{j+1}} \end{bmatrix}, \begin{bmatrix} \tau^2 & \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T} & \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^\mathsf{T} \\ \mathbf{c}_{1:n}(\mathbf{x}) & \mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n & \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} \\ \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x}) & \mathbf{C}^\mathsf{T}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} & \mathbf{C}_{\mathbf{q}_{j+1},\mathbf{q}_{j+1}} + \sigma^2\mathbf{I}_{pk} \end{bmatrix}\right), \tag{A.29}$$

where

$$\begin{aligned}
\mathbf{m}_{\mathbf{q}_{j+1}} &= \Big(m(\mathbf{q}_{j+1}(\Delta)), \ldots, m(\mathbf{q}_{j+1}(pk\Delta))\Big), \\
\mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x}) &= \Big(C(\mathbf{x}, \mathbf{q}_{j+1}(\Delta)), \ldots, C(\mathbf{x}, \mathbf{q}_{j+1}(pk\Delta)\Big), \\
\Big[\mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}\Big]_{il} &= C\Big(\mathbf{s}_i, \mathbf{q}_{j+1}(l\Delta)\Big),\ i \in 1, \ldots, n,\ l \in 1, \ldots, pk\Delta, \\
\Big[\mathbf{C}_{\mathbf{q}_{j+1},\mathbf{q}_{j+1}}\Big]_{il} &= C\Big(\mathbf{q}_{j+1}(i\Delta),\ \mathbf{q}_{j+1}(l\Delta)\Big),\ i, l \in 1, \ldots, pk\Delta.
\end{aligned}$$

Then the conditional mean of $f(\mathbf{x})$ on the other two is

$$\begin{aligned}
\boldsymbol{\mu}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}(\mathbf{x}) &= \mathrm{E}\left\{ f(\mathbf{x}) \mid \mathbf{y}_{1:n}, \mathbf{y}_{\mathbf{q}_{j+1}} \right\} \\
&= m(\mathbf{x}) + \\
&\qquad \begin{bmatrix} \mathbf{c}_{1:n}(\mathbf{x})^{\mathsf{T}} & \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n & \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} \\ \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}^{\mathsf{T}} & \mathbf{C}_{\mathbf{q}_{j+1},\mathbf{q}_{j+1}} + \sigma^2 \mathbf{I}_{pk} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}_{1:n} - \mathbf{m}_{1:n} \\ \mathbf{y}_{\mathbf{q}_{j+1}} - \mathbf{m}_{\mathbf{q}_{j+1}} \end{bmatrix} \\
&= m(\mathbf{x}) + \left[\mathbf{c}_{1:n}(\mathbf{x})^{\mathsf{T}} \boldsymbol{\Gamma}_{11} + \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^{\mathsf{T}} \boldsymbol{\Gamma}_{21} \right] \left[\mathbf{y}_{1:n} - \mathbf{m}_{1:n} \right] \\
&\qquad + \left[\mathbf{c}_{1:n}(\mathbf{x})^{\mathsf{T}} \boldsymbol{\Gamma}_{12} + \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^{\mathsf{T}} \boldsymbol{\Gamma}_{22} \right] \left[\mathbf{y}_{\mathbf{q}_{j+1}} - \mathbf{m}_{\mathbf{q}_{j+1}} \right], \qquad \text{(A.30)}
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{\Gamma}_{11} &= \{ \mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n - \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} \left[\mathbf{C}_{\mathbf{q}_{j+1},\mathbf{q}_{j+1}} + \sigma^2 \mathbf{I}_{pk} \right]^{-1} \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}^{\mathsf{T}} \}^{-1} \quad \text{or} \\
&= \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n \right]^{-1} + \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n \right]^{-1} \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} \boldsymbol{\Gamma}_{22} \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}^{\mathsf{T}} \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n \right]^{-1} \\
\boldsymbol{\Gamma}_{21} &= - \left[\mathbf{C}_{\mathbf{q}_{j+1},\mathbf{q}_{j+1}} + \sigma^2 \mathbf{I}_{pk} \right]^{-1} \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}^{\mathsf{T}} \boldsymbol{\Gamma}_{11} \quad \text{or} \quad - \boldsymbol{\Gamma}_{22} \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}^{\mathsf{T}} \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n \right]^{-1} \\
\boldsymbol{\Gamma}_{12} &= - \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n \right]^{-1} \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} \boldsymbol{\Gamma}_{22} \quad \text{or} \quad - \boldsymbol{\Gamma}_{11} \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} \left[\mathbf{C}_{\mathbf{q}_{j+1},\mathbf{q}_{j+1}} + \sigma^2 \mathbf{I}_{pk} \right]^{-1} \\
\boldsymbol{\Gamma}_{22} &= \left\{ \mathbf{C}_{\mathbf{q}_{j+1},\mathbf{q}_{j+1}} + \sigma^2 \mathbf{I}_{pk} - \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}^{\mathsf{T}} \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n \right]^{-1} \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} \right\}^{-1} \\
&\equiv \left[\mathbf{K}_{\mathbf{y}_{1:n},\mathbf{q}_{j+1}} + \sigma^2 \mathbf{I}_{pk} \right]^{-1},
\end{aligned}$$

with $\mathbf{K}_{\mathbf{y}_{1:n},\mathbf{q}_{j+1}} = \mathbf{C}_{\mathbf{q}_{j+1},\mathbf{q}_{j+1}} - \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}^{\mathsf{T}} \left[\mathbf{C}_{1:n} + \sigma^2 \mathbf{I}_n \right]^{-1} \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}$.

If $\mathbf{y}_{\mathbf{q}_{j+1}}$ is fixed at $\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}$, then $\mu_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x})$ is equal to (A.30) with $\mathbf{y}_{\mathbf{q}_{j+1}}$ replaced by $\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}$, where

$$\hat{\mathbf{y}}_{\mathbf{q}_{j+1}} = \mathrm{E}\left\{ \mathbf{y}_{\mathbf{q}_{j+1}} \mid \mathbf{y}_{1:n} \right\} = \left(\mu_{\mathbf{y}_{1:n}}(\mathbf{q}_{j+1}(\Delta)), \ldots, \mu_{\mathbf{y}_{1:n}}(\mathbf{q}_{j+1}(pk\Delta)) \right).$$

And the conditional variance term is

$$
\begin{aligned}
\kappa^2_{\mathbf{y}_{1:n},\hat{\mathbf{y}}_{\mathbf{q}_{j+1}}}(\mathbf{x}) &= \mathrm{Var}\left\{f(x) \mid \mathbf{y}_{1:n}, \hat{\mathbf{y}}_{\mathbf{q}_{j+1}}\right\} = \mathrm{Var}\left\{f(x) \mid \mathbf{y}_{1:n}, \mathbf{y}_{\mathbf{q}_{j+1}}\right\} \\
&= \tau^2 - \begin{bmatrix}\mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T} & \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^\mathsf{T}\end{bmatrix} \begin{bmatrix}\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n & \mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} \\ \mathbf{C}^\mathsf{T}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} & \mathbf{C}_{\mathbf{q}_{j+1},\mathbf{q}_{j+1}} + \sigma^2\mathbf{I}_{pk}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{c}_{1:n}(\mathbf{x}) \\ \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})\end{bmatrix} \\
&= \tau^2 - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\boldsymbol{\Gamma}_{11}\mathbf{c}_{1:n}(\mathbf{x}) - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}\boldsymbol{\Gamma}_{12}\mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x}) - \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^\mathsf{T}\boldsymbol{\Gamma}_{21}\mathbf{c}_{1:n}(\mathbf{x}) \\
&\qquad - \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^\mathsf{T}\boldsymbol{\Gamma}_{22}\mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x}) \\
&= \tau^2 - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{x}) \\
&\qquad - \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}\left[\mathbf{K}_{\mathbf{y}_{1:n},\mathbf{q}_{j+1}} + \sigma^2\mathbf{I}_{pk}\right]^{-1} \times \\
&\qquad\qquad \mathbf{C}^\mathsf{T}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{x}) \\
&\qquad + \mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}\left[\mathbf{K}_{\mathbf{y}_{1:n},\mathbf{q}_{j+1}} + \sigma^2\mathbf{I}_{pk}\right]^{-1}\mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x}) \\
&\qquad + \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^\mathsf{T}\left[\mathbf{K}_{\mathbf{y}_{1:n},\mathbf{q}_{j+1}} + \sigma^2\mathbf{I}_{pk}\right]^{-1}\mathbf{C}^\mathsf{T}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{x}) \\
&\qquad - \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^\mathsf{T}\left[\mathbf{K}_{\mathbf{y}_{1:n},\mathbf{q}_{j+1}} + \sigma^2\mathbf{I}_{pk}\right]^{-1}\mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x}) \\
&= \kappa^2_{\mathbf{y}_{1:n}}(\mathbf{x}) - \left\{\mathbf{c}_{1:n}(\mathbf{x})^\mathsf{T}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{C}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}} - \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})^\mathsf{T}\right\} \times \\
&\qquad \left[\mathbf{K}_{\mathbf{y}_{1:n},\mathbf{q}_{j+1}} + \sigma^2\mathbf{I}_{pk}\right]^{-1}\left\{\mathbf{C}^\mathsf{T}_{\mathbf{y}_{1:n},\mathbf{y}_{\mathbf{q}_{j+1}}}[\mathbf{C}_{1:n} + \sigma^2\mathbf{I}_n]^{-1}\mathbf{c}_{1:n}(\mathbf{x}) - \mathbf{c}_{\mathbf{q}_{j+1}}(\mathbf{x})\right\}.
\end{aligned}
\tag{A.31}
$$

A.7 Estimating φ in $m(\cdot\,;\varphi) = \varphi$

The measurement noise variance σ^2, along with the parameters $\varphi, \boldsymbol{\psi}$ of the mean and covariance functions of the Gaussian process prior, are estimated using the data. To do so, we use Bayes rule, which says the posterior distribution of the parameters is proportional to the product of the data likelihood and the prior distribution of the parameters:

$$
p(\text{parameters} \mid \text{data}) \propto p(\text{data} \mid \text{parameters})\, p(\text{parameters}) \tag{A.32}
$$

Using a flat prior for φ, i.e., $p(\varphi) \propto 1$, and assuming independence among σ^2, φ, and $\boldsymbol{\psi}$, then

$$\begin{aligned} p(\sigma^2, \varphi, \boldsymbol{\psi} \mid \mathbf{y}) &\propto p(\mathbf{y} \mid \sigma^2, \varphi, \boldsymbol{\psi})\, p(\sigma^2, \varphi, \boldsymbol{\psi}) \\ &\propto \mathsf{N}\left(\mathbf{y} \mid \varphi \mathbf{1}_n,\, \mathbf{C} + \sigma^2 \mathbf{I}_n\right)\, p\left(\sigma^2\right)\, p(\boldsymbol{\psi}) \\ &\propto \left|\mathbf{C} + \sigma^2 \mathbf{I}_n\right|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left[\mathbf{y} - \varphi \mathbf{1}_n\right]^{\mathsf{T}} \left[\mathbf{C} + \sigma^2 \mathbf{I}_n\right]^{-1} \left[\mathbf{y} - \varphi \mathbf{1}_n\right]\right\}\, p\left(\sigma^2\right)\, p(\boldsymbol{\psi}) \end{aligned}$$

Taking the logarithm gives

$$\begin{aligned} &\log p(\sigma^2, \varphi, \boldsymbol{\psi} \mid \mathbf{y}) = \\ &-\frac{1}{2}\log\left|\mathbf{C} + \sigma^2 \mathbf{I}_n\right| - \frac{1}{2}\left[\mathbf{y} - \varphi \mathbf{1}_n\right]^{\mathsf{T}} \left[\mathbf{C} + \sigma^2 \mathbf{I}_n\right]^{-1} \left[\mathbf{y} - \varphi \mathbf{1}_n\right] + \log\left(\sigma^2\right) + \log p(\boldsymbol{\psi}). \end{aligned} \tag{A.33}$$

φ only appears in the second term, so we can use the profile maximization method and obtain the estimate of φ

$$\hat{\varphi} = \frac{\mathbf{1}_n^{\mathsf{T}} \left[\mathbf{C} + \sigma^2 \mathbf{I}_n\right]^{-1} \mathbf{y}}{\mathbf{1}_n^{\mathsf{T}} \left[\mathbf{C} + \sigma^2 \mathbf{I}_n\right]^{-1} \mathbf{1}_n}. \tag{A.34}$$

Then this estimate of φ is substituted back in eq (A.33) and we continue to find numerically the values of σ^2 and $\boldsymbol{\psi}$ that maximize the log-joint posterior distribution.

A.7.1 Estimating φ when there is additional information $\mathbf{z}$

With additional data $\mathbf{z}$, there is the additional parameter σ_z^2. Using a flat prior for φ, i.e., $p(\varphi) \propto 1$, and assuming independence among σ_z^2, σ_y^2, φ, $\boldsymbol{\psi}$, then

$$
\begin{aligned}
p(\sigma_z^2, \sigma_y^2, \varphi, \boldsymbol{\psi} \mid \text{data}) &\propto p(\text{data} \mid \sigma_z^2, \sigma_y^2, \varphi, \boldsymbol{\psi})\, p(\sigma_z^2, \sigma_y^2, \varphi, \boldsymbol{\psi}) \\
&\propto \mathsf{N}\left(\begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} \,\middle|\, \begin{bmatrix} \varphi \mathbf{1}_r \\ \varphi \mathbf{1}_n \end{bmatrix}, \underbrace{\begin{bmatrix} \mathbf{C}_{zz} + \sigma_z^2 \mathbf{I}_r & \mathbf{C}_{zy} \\ \mathbf{C}_{zy}^{\mathsf{T}} & \mathbf{C}_{yy} + \sigma_y^2 \mathbf{I}_n \end{bmatrix}}_{\boldsymbol{\Gamma}} \right) p\left(\sigma_z^2\right)\, p\left(\sigma_y^2\right)\, p(\boldsymbol{\psi}) \\
&\propto |\boldsymbol{\Gamma}|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} \begin{bmatrix} \mathbf{z} - \varphi \mathbf{1}_r \\ \mathbf{y} - \varphi \mathbf{1}_n \end{bmatrix}^{\mathsf{T}} \boldsymbol{\Gamma}^{-1} \begin{bmatrix} \mathbf{z} - \varphi \mathbf{1}_r \\ \mathbf{y} - \varphi \mathbf{1}_n \end{bmatrix} \right\} p\left(\sigma_z^2\right)\, p\left(\sigma_y^2\right)\, p(\boldsymbol{\psi}).
\end{aligned}
$$

Taking the logarithm:

$$
\begin{aligned}
&\log p(\sigma_z^2, \sigma_y^2, \varphi, \boldsymbol{\psi} \mid \text{data}) = \\
&\quad -\frac{1}{2} \log |\boldsymbol{\Gamma}| - \frac{1}{2} \begin{bmatrix} \mathbf{z} - \varphi \mathbf{1}_r \\ \mathbf{y} - \varphi \mathbf{1}_n \end{bmatrix}^{\mathsf{T}} \boldsymbol{\Gamma}^{-1} \begin{bmatrix} \mathbf{z} - \varphi \mathbf{1}_r \\ \mathbf{y} - \varphi \mathbf{1}_n \end{bmatrix} + \log p\left(\sigma_z^2\right) + \log\left(\sigma_y^2\right) + \log p(\boldsymbol{\psi}).
\end{aligned}
$$

φ only appears in the second term, so we can use the profile maximization method and obtain the estimate of φ

$$
\hat{\varphi} = \frac{\mathbf{1}_{r+n}^{\mathsf{T}} \boldsymbol{\Gamma}^{-1} \begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix}}{\mathbf{1}_{r+n}^{\mathsf{T}} \boldsymbol{\Gamma}^{-1} \mathbf{1}_{r+n}} = \frac{\mathbf{1}_{r+n}^{\mathsf{T}} \begin{bmatrix} \mathbf{C}_{zz} + \sigma_z^2 \mathbf{I}_r & \mathbf{C}_{zy} \\ \mathbf{C}_{zy}^{\mathsf{T}} & \mathbf{C}_{yy} + \sigma_y^2 \mathbf{I}_n \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix}}{\mathbf{1}_{r+n}^{\mathsf{T}} \begin{bmatrix} \mathbf{C}_{zz} + \sigma_z^2 \mathbf{I}_r & \mathbf{C}_{zy} \\ \mathbf{C}_{zy}^{\mathsf{T}} & \mathbf{C}_{yy} + \sigma_y^2 \mathbf{I}_n \end{bmatrix}^{-1} \mathbf{1}_{r+n}}.
$$

A.8 Details of the rational quadratic covariance

In one dimension, the rational quadratic covariance for two points on the line is

$$C(x, x'; \boldsymbol{\psi}) = \frac{\tau^2}{1 + \left(\frac{x-x'}{\lambda}\right)^2}, \qquad s \in \mathbb{R}, \qquad \boldsymbol{\psi} = (\tau^2, \lambda). \tag{A.35}$$

For two dimensions (as in the case of f over $\mathcal{S}$), we assume separability. Let $\mathbf{x} = (x_1, x_2)$, $\mathbf{x}' = (x_1', x_2') \in \mathbb{R}^2$. Then

$$C(\mathbf{x}, \mathbf{x}'; \boldsymbol{\psi}) = \frac{\tau^2}{\left[1 + \left(\frac{x_1-x_1'}{\lambda_1}\right)^2\right]\left[1 + \left(\frac{x_2-sx_2'}{\lambda_2}\right)^2\right]}, \qquad \boldsymbol{\psi} = (\tau^2, \lambda_1, \lambda_2). \tag{A.36}$$

Using this two dimenstional rational quadratic covariance function, and assuming that $\mathcal{S}$ is partitioned into rectangular cells $S_1, \ldots, S_r$ with $S_i = [a_i, b_i] \times [c_i, d_i]$, we can derive the expressions for $\mathbf{C}_{zz}, \mathbf{C}_{zy}$, and $\mathbf{c}_z(\mathbf{x})$. For more compact notation, first define

$$\ell_1 = \frac{a_i - a_j}{\lambda_1} \qquad \ell_5 = \frac{c_i - c_j}{\lambda_2}$$
$$\ell_2 = \frac{a_i - b_j}{\lambda_1} \qquad \ell_6 = \frac{c_i - d_j}{\lambda_2}$$
$$\ell_3 = \frac{b_i - a_j}{\lambda_1} \qquad \ell_7 = \frac{d_i - c_j}{\lambda_2}$$
$$\ell_4 = \frac{b_i - b_j}{\lambda_1} \qquad \ell_8 = \frac{d_i - d_j}{\lambda_2}$$

Then

$$
\begin{aligned}
[\mathbf{C}_{zz}]_{ij} &= \frac{1}{|S_i||S_j|}\int_{S_i}\int_{S_j} C(\mathbf{x},\mathbf{x}';\boldsymbol{\psi})\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{x}' \\
&= \frac{\tau^2}{(b_i-a_i)(d_i-c_i)(b_j-a_j)(d_j-c_j)} \\
&\quad\times\int_{c_j}^{d_j}\int_{a_j}^{b_j}\int_{c_i}^{d_i}\int_{a_i}^{b_i}\frac{1}{\left[1+\left(\frac{x_1-x_1'}{\lambda_1}\right)^2\right]\left[1+\left(\frac{x_2-x_2'}{\lambda_2}\right)^2\right]}\,\mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}x_1'\,\mathrm{d}x_2' \\
&= \frac{\tau^2\lambda_1^2\lambda_2^2}{(b_i-a_i)(d_i-c_i)(b_j-a_j)(d_j-c_j)} \\
&\quad\times\left[\ell_2\,\mathrm{atan}(\ell_2)+\ell_3\,\mathrm{atan}(\ell_3)-\ell_1\,\mathrm{atan}(\ell_1)-\ell_4\,\mathrm{atan}(\ell_4)+\frac{1}{2}\log\left(\frac{[1+\ell_1^2]\,[1+\ell_4^2]}{[1+\ell_2^2]\,[1+\ell_3^2]}\right)\right] \\
&\quad\times\left[\ell_6\,\mathrm{atan}(\ell_6)+\ell_7\,\mathrm{atan}(\ell_7)-\ell_5\,\mathrm{atan}(\ell_5)-\ell_8\,\mathrm{atan}(\ell_8)+\frac{1}{2}\log\left(\frac{[1+\ell_5^2]\,[1+\ell_8^2]}{[1+\ell_6^2]\,[1+\ell_7^2]}\right)\right],
\end{aligned}
$$

$$
\begin{aligned}
[\mathbf{C}_{zy}]_{ij} &= \frac{1}{|S_i|}\int_{S_i} C(\mathbf{x},\mathbf{s}_j^y;\boldsymbol{\psi})\,\mathrm{d}\mathbf{x} \\
&= \frac{\tau^2}{(b_i-a_i)(d_i-c_i)}\int_{c_i}^{d_i}\int_{a_i}^{b_i}\frac{1}{\left[1+\left(\frac{x_1-s_{j,1}^y}{\lambda_1}\right)^2\right]\left[1+\left(\frac{x_2-s_{j,2}^y}{\lambda_2}\right)^2\right]}\,\mathrm{d}x_1\,\mathrm{d}x_2 \\
&= \frac{\tau^2\lambda_1\lambda_2}{(b_i-a_i)(d_i-c_i)}\left[\mathrm{atan}\left(\frac{b_i-s_{j,1}^y}{\lambda_1}\right)-\mathrm{atan}\left(\frac{a_i-s_{j,1}^y}{\lambda_1}\right)\right] \\
&\qquad\times\left[\mathrm{atan}\left(\frac{d_i-s_{j,2}^y}{\lambda_2}\right)-\mathrm{atan}\left(\frac{c_i-s_{j,2}^y}{\lambda_2}\right)\right],
\end{aligned}
$$

and

$$
\begin{aligned}
[\mathbf{c}_z(\mathbf{x})]_i = \frac{\tau^2\lambda_1\lambda_2}{(b_i-a_i)(d_i-c_i)}&\left[\mathrm{atan}\left(\frac{b_i-s_1}{\lambda_1}\right)-\mathrm{atan}\left(\frac{a_i-x_1}{\lambda_1}\right)\right] \\
&\times\left[\mathrm{atan}\left(\frac{d_i-x_2}{\lambda_2}\right)-\mathrm{atan}\left(\frac{c_i-x_2}{\lambda_2}\right)\right].
\end{aligned}
$$

Bibliography

[1] Markus W Achtelik, Simon Lynen, Stephan Weiss, Margarita Chli, and Roland Siegwart. Motion-and uncertainty-aware path planning for micro aerial vehicles. *Journal of Field Robotics*, 31(4):676–698, 2014.

[2] Sivaram Ambikasaran, Daniel Foreman-Mackey, Leslie Greengard, David W Hogg, and Michael O'Neil. Fast direct methods for gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):252–265, 2015.

[3] Erik P Anderson, Randal W Beard, and Timothy W McLain. Real-time dynamic trajectory smoothing for unmanned air vehicles. *IEEE Transactions on Control Systems Technology*, 13(3):471–477, 2005.

[4] S Banerjee and AE Gelfand. On smoothness properties of spatial processes. *Journal of Multivariate Analysis*, 84(1):85–100, 2003.

[5] Sudipto Banerjee, Alan E Gelfand, and Bradley P Carlin. *Hierarchical Modeling and Analysis for Spatial Data*. CRC Press, 2004.

[6] Daman Bareiss, Jur van den Berg, and Kam K Leang. Stochastic automatic collision avoidance for tele-operated unmanned aerial vehicles. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4818–4825. IEEE, 2015.

[7] Behzad Bayat, Naveena Crasta, Alessandro Crespi, António M Pascoal, and Auke Ijspeert. Environmental monitoring using autonomous vehicles: a survey of recent searching techniques. *Current opinion in biotechnology*, 45:76–84, 2017.

[8] Randal W Beard, Clark Taylor, Jeff Saunders, Ryan Holt, and Timothy W McLain. Payload directed flight of miniature air vehicles. In *AIAA Infotech @ Aerospace Conf.*, April 2009.

[9] Romain Benassi, Julien Bect, and Emmanuel Vazquez. Robust Gaussian process-based global optimization using a fully Bayesian expected improve-

ment criterion. In *Learning and Intelligent Optimization*, pages 176–190. Springer, 2011.

[10] James O Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, second edition, 1993.

[11] Jonathan Binney and Gaurav S Sukhatme. Branch and bound for informative path planning. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2147–2154. IEEE, 2012.

[12] Richard P Brent. *Algorithms for Minimization Without Derivatives*. Courier Corporation, 1973.

[13] Erik Burian, Dana Yoerger, Albert Bradley, and Hanumant Singh. Gradient search with autonomous underwater vehicles using scalar measurements. In *Proc. 1996 Symposium on Autonomous Underwater Vehicle Technology*, pages 86–98. IEEE, 1996.

[14] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[15] Nannan Cao, Kian Hsiang Low, and John M Dolan. Multi-robot informative path planning for active sensing of environmental phenomena: A tale of two algorithms. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 7–14. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[16] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[17] Jongeun Choi, Songhwai Oh, and Roberto Horowitz. Distributed learning and cooperative control for multi-agent systems. *Automatica*, 45(12):2802–2814, 2009.

[18] Taeryon Choi and Mark J Schervish. On posterior consistency in nonparametric regression problems. *Journal of Multivariate Analysis*, 98(10):1969–1987, 2007.

[19] J. Cochran and M. Krstic. Nonholonomic source seeking with tuning of angular velocity. *IEEE Trans. Autom. Control*, 54(4):717–731, April 2009.

[20] Jorge Cortés, Sonia Martínez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 20(2):243, 2004.

[21] N Cressie. *Statistics for Spatial Data*. Wiley, 1993.

[22] Rongxin Cui, Yang Li, and Weisheng Yan. Mutual information-based multi-auv path planning for scalar field sampling using multidimensional rrt. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(7):993–1004, 2015.

[23] Jaydev P Desai, James P Ostrowski, and Vijay Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.

[24] A. Dhariwal, G. Sukhatme, and A. Requicha. Bacterium-inspired robots for environmental monitoring. In *IEEE Int. Conf. Robotics and Automation*, volume 2, pages 1436–1443, April 2004.

[25] Giancarmine Fasano, Domenico Accardo, Antonio Moccia, Ciro Carbone, Umberto Ciniglio, Federico Corraro, and Salvatore Luongo. Multi-sensor-based fully autonomous non-cooperative collision avoidance system for unmanned air vehicles. *Journal of aerospace computing, information, and communication*, 5(10):338–360, 2008.

[26] Rafael Fierro, Peng Song, Aveek Das, and Vijay Kumar. Cooperative control of robot formations. In *Cooperative control and optimization*, pages 73–93. Springer, 2002.

[27] Thierry Fraichard and Alexis Scheuer. From reeds and shepp's to continuous-curvature paths. *IEEE Trans. Robotics*, 20(6):1025–1035, 2004.

[28] Marcus Frean and Phillip Boyle. Using Gaussian processes to optimize expensive functions. In *AI 2008: Advances in Artificial Intelligence*, pages 258–267. Springer, 2008.

[29] Kikuo Fujimura and Toshiyasu L Kunii. *Motion Planning in Dynamic Environments*. Springer-Verlag New York, Inc., 1992.

[30] Subhashis Ghosal and Anindya Roy. Posterior consistency of gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, pages 2413–2429, 2006.

[31] C Goerzen, Zhaodan Kong, and Bernard Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, 57(1-4):65–100, 2010.

[32] Rishi Graham and Jorge Cortés. Cooperative adaptive sampling of random fields with partially known covariance. *Int. Journal Robust and Nonlinear Control*, 22(5):504–534, 2012.

[33] Fadi Halal, Pablo Pedrocca, Thomas Hirose, Ana-Maria Cretu, and Marek B Zaremba. Remote-sensing based adaptive path planning for an aquatic platform to monitor water quality. In *Robotic and Sensors Environments (ROSE), 2014 IEEE International Symposium on*, pages 43–48. IEEE, 2014.

[34] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[35] Gordon J Keller, Sebastian Hening, Sisi Song, Mircea Teodorescu, Brat Guillaume, Nhan T Nguyen, and Corey A Ippolito. Demonstration platform for trajectory planning of an autonomous nonholonomic skid-steer system for investigating spatial phenomena. In *Infotech @ Aerospace*, pages 3673–3680. AIAA, 2016.

[36] Alonzo Kelly and Bryan Nagy. Reactive nonholonomic trajectory generation via parametric optimal control. *The International Journal of Robotics Research*, 22(7-8):583–601, 2003.

[37] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[38] Sven Koenig and Maxim Likhachev. Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics*, 21(3):354–363, 2005.

[39] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.

[40] Parth Kumar and Corey Ippolito. Modeling and trajectory control of an autonomous sailboat for path planning. In *AIAA Infotech @ Aerospace Conf.*, April 2009.

[41] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[42] J.P. Laumond, S. Sekhavat, and F. Lamiraux. Guidelines in nonholonomic motion planning for mobile robots. In *Robot Motion Planning and Control*, volume 229 of *Lecture Notes in Control and Information Sciences*, pages 1–53. Springer, 1998.

[43] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[44] Jerome Le Ny and George J Pappas. On trajectory optimization for active sensing in gaussian process models. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 6286–6292. IEEE, 2009.

[45] Jusuk Lee, Rosemary Huang, Andrew Vaughn, Xiao Xiao, J Karl Hedrick, Marco Zennaro, and Raja Sengupta. Strategies of path-planning for a UAV to track a ground vehicle. In *AINS Conference*, 2003.

[46] Ritchie Lee and Corey Ippolito. A perception and mapping approach for plume detection in payload directed flight. In *AIAA Infotech@Aerospace Conf.*, April 2009.

[47] Sung-On Lee, Young-Jo Cho, Myung Hwang-Bo, Bum-Jae You, and Sang-Rok Oh. A stable target-tracking control for unicycle mobile robots. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, volume 3, pages 1822–1827, 2000.

[48] N.E. Leonard, D.A Paley, F. Lekien, R. Sepulchre, D.M. Fratantoni, and R.E. Davis. Collective motion, sensor networks, and ocean sampling. *Proc. IEEE*, 95(1):48–74, Jan 2007.

[49] Shu-Jun Liu and Miroslav Krstic. Stochastic source seeking for nonholonomic unicycle. *Automatica*, 46(9):1443–1453, 2010.

[50] Kai-Chieh Ma, Lantao Liu, and Gaurav S Sukhatme. Informative planning and online learning with sparse gaussian processes. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4292–4298. IEEE, 2017.

[51] Jay D Martin and Timothy W Simpson. On the use of kriging models to approximate deterministic computer models. In *ASME Int. Design Engineering Tech. Conf. and Computers & Information in Engineering Conf.*, pages 481–492. American Society of Mechanical Engineers, 2004.

[52] Alexandra Meliou, Andreas Krause, Carlos Guestrin, and Joseph M Hellerstein. Nonmyopic informative path planning in spatio-temporal models. In *AAAI*, volume 10, pages 16–7, 2007.

[53] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE International Conference on Robotics and Automation*, pages 2520–2525. IEEE, 2011.

[54] Carl N Morris. Parametric empirical Bayes inference: theory and applications. *Journal of the American Statistical Association*, 78(381):47–55, 1983.

[55] Richard M Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):571–583, 2007.

[56] R.M. Murray and S.S. Sastry. Nonholonomic motion planning: steering using sinusoids. *Automatic Control, IEEE Transactions on*, 38(5):700–716, May 1993.

[57] Muhammad F Mysorewala, Dan O Popa, and Frank L Lewis. Multi-scale adaptive sampling with mobile agents for mapping of forest fires. *Journal of Intelligent and Robotic Systems*, 54(4):535, 2009.

[58] John A Nelder and Roger Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[59] Matthias Nieuwenhuisen and Sven Behnke. 3d planning and trajectory optimization for real-time generation of smooth mav trajectories. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–7. IEEE, 2015.

[60] P. Ogren, E. Fiorelli, and N.E. Leonard. Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Trans. Autom. Control*, 49(8):1292–1302, Aug 2004.

[61] Tiago Oliveira, Pedro Encarnaçao, and A Pedro Aguiar. Moving path following for autonomous robotic vehicles. In *2013 European Control Conference (ECC)*, pages 3320–3325. IEEE, 2013.

[62] Lynne E Parker. Distributed intelligence: Overview of the field and its application in multi-robot systems. In *AAAI Fall Symposium: Regarding the Intelligence in Distributed Intelligent Systems*, pages 1–6, 2007.

[63] Joaquin Quinonero-Candela and Carl Edward Rasmussen. Analysis of some methods for reduced rank gaussian process regression. In *Switching and learning in feedback systems*, pages 98–127. Springer, 2005.

[64] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.

[65] E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Trans. Robot. Autom.*, 8(5):501–518, Oct 1992.

[66] Herbert Robbins. The empirical Bayes approach to statistical decision problems. *The Annals of Mathematical Statistics*, pages 1–20, 1964.

[67] Erick J Rodríguez-Seda, Dušan M Stipanović, and Mark W Spong. Collision avoidance control with sensing uncertainties. In *Proceedings of the 2011 American Control Conference*, pages 3363–3368. IEEE, 2011.

[68] Jerome Sacks, William J Welch, Toby J Mitchell, and Henry P Wynn. Design and analysis of computer experiments. *Statistical Science*, pages 409–423, 1989.

[69] K. Savla, E. Frazzoli, and F. Bullo. Traveling salesperson problems for the dubins vehicle. *IEEE Trans. Autom. Control*, 53(6):1378–1391, July 2008.

[70] Václav Šmídl and Radek Hofman. Tracking of atmospheric release of pollution using unmanned aerial vehicles. *Atmospheric Environment*, 67:425–436, 2013.

[71] Ryan N Smith, Yi Chao, Peggy P Li, David A Caron, Burton H Jones, and Gaurav S Sukhatme. Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model. *The International Journal of Robotics Research*, 29(12):1475–1497, 2010.

[72] Cyrill Stachniss, Christian Plagemann, Achim J Lilienthal, and Wolfram Burgard. Gas distribution modeling using sparse gaussian process mixture models. In *Robotics: Science and Systems*, volume 3, 2008.

[73] Michael L Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media, 1999.

[74] A Stentz. Optimal and efficient path planning for partially-known environments. In *IEEE Int. Conf. Robotics and Automation*, volume 4, pages 3310–3317, May 1994.

[75] Petr Švec, Atul Thakur, Eric Raboin, Brual C Shah, and Satyandra K Gupta. Target following with motion prediction for unmanned surface vehicle operating in cluttered environments. *Autonomous Robots*, 36(4):383–405, 2014.

[76] Matthew Tesch, Jeff Schneider, and Howie Choset. Using response surfaces and expected improvement to optimize snake robot gait parameters. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 1069–1074. IEEE, 2011.

[77] Surya T Tokdar and Jayanta K Ghosh. Posterior consistency of logistic gaussian process priors in density estimation. *Journal of statistical planning and inference*, 137(1):34–42, 2007.

[78] Shrihari Vasudevan, Fabio Ramos, Eric Nettleton, and Hugh Durrant-Whyte. Gaussian process modeling of large-scale terrain. *Journal of Field Robotics*, 26(10):812–840, 2009.

[79] Emmanuel Vazquez and Julien Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference*, 140(11):3088–3095, 2010.

[80] Nikos Vlassis, Mohammad Ghavamzadeh, Shie Mannor, and Pascal Poupart. Bayesian reinforcement learning. In *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 359–386. Springer, 2012.

[81] Tobias Wagner, Michael Emmerich, André Deutz, and Wolfgang Ponweiser. On expected-improvement criteria for model-based multi-objective optimization. In *Parallel Problem Solving from Nature*, volume 6238 of *Lecture Notes in Computer Science*, pages 718–727. Springer, 2010.

[82] N.K. Yilmaz, C. Evangelinos, P. F J Lermusiaux, and N.M. Patrikalakis. Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *IEEE J. Ocean. Eng.*, 33(4):522–537, Oct 2008.

[83] Huili Yu, Kevin Meier, Matthew Argyle, and Randal W Beard. Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles. *IEEE/ASME Transactions on Mechatronics*, 20(2):541–552, 2014.

[84] Bin Zhang and Gaurav S Sukhatme. Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3673–3680. IEEE, 2007.

[85] Anatoly Zhigljavsky and Antanasz Zilinskas. *Stochastic global optimization*, volume 9. Springer Science & Business Media, 2007.