Adaptive Sampling for Efficient Online Modelling

Troy Wilson, BComm (Hons 1), MQF

A thesis submitted in fulfillment

of the requirements of the degree of

Doctor of Philosophy

Australian Centre for Field Robotics

School of Aerospace, Mechanical and Mechatronic Engineering

The University of Sydney

June 2017

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Troy Wilson, BComm (Hons 1), MQF

15 June 2017

Abstract

Troy Wilson, BComm (Hons 1), MQF
The University of Sydney

Doctor of Philosophy
June 2017

Adaptive Sampling for Efficient Online Modelling

This thesis examines methods enabling autonomous systems to make active sampling and planning decisions in real time. Gaussian Process (GP) regression is chosen as a framework for its non-parametric approach allowing flexibility in unknown environments.

The first part of the thesis focuses on depth constrained full coverage bathymetric surveys in unknown environments. Algorithms are developed to find and follow a depth contour, modelled with a GP, and produce a depth constrained boundary. An extension to the Boustrophedon Cellular Decomposition, Discrete Monotone Polygonal Partitioning is developed allowing efficient planning for coverage within this boundary. Efficient computational methods such as incremental Cholesky updates are implemented to allow online Hyper Parameter optimisation and fitting of the GP's. This is demonstrated in simulation and the field on a platform built for the purpose.

The second part of this thesis focuses on modelling the surface salinity profiles of estuarine tidal fronts. The standard GP model assumes evenly distributed noise, which does not always hold. This can be handled with Heteroscedastic noise. An efficient new method, Parametric Heteroscedastic Gaussian Process regression, is proposed. This is applied to active sample selection on stationary fronts and adaptive planning on moving fronts where a number of information theoretic methods are compared. The use of a mean function is shown to increase the accuracy of predictions whilst reducing optimisation time. These algorithms are validated in simulation.

Algorithmic development is focused on efficient methods allowing deployment on platforms with constrained computational resources. Whilst the application of this thesis is Autonomous Surface Vessels, it is hoped the issues discussed and solutions provided have relevance to other applications in robotics and wider fields such as spatial statistics and machine learning in general.

Acknowledgements

This thesis would not have been possible without the help of my supervisor, my colleagues and the technical staff at the ACFR and most importantly the patience and understanding of my family.

Contents

Declaration **iv**

Abstract **v**

Acknowledgements **vi**

Contents **ix**

List of Figures **xi**

List of Tables **xii**

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Statement 2

1.3 Contributions 3

1.4 Outline 4

2 Background **6**

2.1 Estuaries 6

2.1.1 Estuarine Tidal Dynamics 8

2.1.2 Physical Dynamics Models 13

2.2 Adaptive Sampling 14

2.3 Gaussian Processes 20

3 Adaptive Path Planning for Depth Constrained Bathymetric Mapping **26**

3.1 Introduction 26

3.2 Related Work 27
3.3 Autonomy Suite of Algorithms 31
3.3.1 Online Gaussian Process Updates 32
3.3.2 Find and Follow the Intersection of a Depth Contour and a Boundary 33
3.3.3 Discrete Monotone Polygonal Partitioning and Path Generation 37
3.4 Simulation 43
3.5 Field Tests 45
3.6 Summary 51

4 Active Sample Selection with Parametric Heteroscedastic Gaussian Process Regression 53
4.1 Introduction 53
4.2 Related Work 55
4.3 Gaussian Processes 56
4.3.1 Heteroscedastic Noise 57
4.4 Modelling a Salinity Gradient 59
4.5 Evaluation of Model Fit 62
4.6 Adaptive Sampling Theory 68
4.6.1 Entropy 68
4.6.2 Mutual Information 68
4.6.3 Fisher Information 69
4.7 Adaptive Sampling Results 71
4.8 Summary 74

5 Adaptive Path Planning to Model a Dynamic Estuarine Tidal Front 76
5.1 Introduction 76
5.2 Related Work 77
5.3 Gaussian Processes 79
5.3.1 Time Dependent Heteroscedastic Noise 79
5.3.2 Mean Functions 80
5.4 Adaptive Path Planning 82
5.5 Experiments 83

5.5.1 Impact of Mean Function 83
5.5.2 Path Planning Results 85
5.6 Summary 90

6 Conclusion **92**
6.1 Summary of Contributions 92
6.1.1 Discrete Monotone Polygonal Partitioning 92
6.1.2 Autonomous Depth Bounded Bathymetric Modelling 93
6.1.3 Parametric Heteroscedastic Gaussian Process regression 93
6.1.4 Mean Functions in GPs 94
6.1.5 Active Sampling and Path Planning of Tidal Fronts 94
6.2 Future Work 95
6.2.1 Field Work 95
6.2.2 Scalability 95
6.2.3 Hyper Parameter Uncertainty 96
6.2.4 Incorporating Other Sensors 96
6.2.5 Parametric Heteroscedastic Noise Functions 97

A Development of a Small Autonomous Surface Vessel **98**
A.1 Hardware 98
A.2 Software 100
A.2.1 ROS 100
A.2.2 Heading Control and Sensor Fusion 102

B Hyper Parameter Optimisation **105**
B.1 Priors 105
B.2 Analytical Derivatives of Log Marginal Likelihood 108
B.2.1 Stationary Logistic Polynomial Heteroscedastic Noise Gaussian Process with Zero Mean 108
B.2.2 Drifting Logistic Polynomial Heteroscedastic Noise Gaussian Process with Parametric Mean 110

C Films and Animations **113**

Bibliography **115**

List of Figures

2.1 Schematic of Salt/Fresh water Interface 10

2.2 Cross-Sectional Schematic of Tidal Intrusion Front Plunge Line 11

2.3 Surface Manifestations of Tidal Front Classes 11

2.4 Photograph of a Bathymetrically Arrested Tidal Front 12

3.1 Example of RoseSolve Algorithm 34

3.2 Comparison of BCD to DMPP 38

3.3 DMPP and Path Generation 42

3.4 Simulated Depth Constrained Bathymetric Mapping 44

3.5 Autonomous Surface Vessel 46

3.6 Satellite Image of Field Test 47

3.7 Series of GP Bathymetric Estimates 48

3.8 Iterative Hyper Parameter Estimation 49

3.9 Estimated Kernel Density - Online vs. Prior 49

4.1 Simulated Surface Salinity 60

4.2 Comparison of Model Fit. Homoscedastic vs. Heteroscedastic 63

4.3 Sample Paths and Predictions from Standard GP and PHGP 66

4.4 Example Realisations Overlaid on Observations for Standard GP and PHGP 67

4.5 Model Fit and comparison Information Measures for PHGP 70

4.6 Convergence Error Rates in RMSE and RMSSDE for Different adaptive Sampling Methods 72

4.7 Comparison of Predicted Standard Deviation Under Different Adaptive Sampling Methods for an Example Simulation 73

5.1 Simple 1D example of the Impact of a Mean Function on GP Predicted Mean and Variance 82
5.2 Simulated tidal front distribution, Initial Scan and Potential Random Paths 84
5.3 Comparison of effect of Mean Function on Time, RMSE, RMSSDE vs No. Samples 86
5.4 RMSE Under Logistic Polynomial Mean Model 88
5.5 RMSSDE Under Logistic Polynomial Mean Model 89
5.6 Example Paths for the Logistic Polynomial Mean Model Across Information Methods 89
5.7 Hyper Parameter Evolution 91

A.1 Autonomous Surface Vessel 99
A.2 ASV Layout 100
A.3 ROS Nodes, Messages and Data Flow 100
A.4 Graphical User Interface 101
A.5 Controller Thrust Response 102

List of Tables

3.1 Parameter Settings for Simulation and Field Trials 43

4.1 Simulation Parameters for Bathymetrically Arrested Tidal Fronts 62

4.2 RMSE, RMSSDE and Equality of Variance Tests on Normalised Errors 64

4.3 Comparison of Computational Load 65

5.1 Simulation Parameters 83

5.2 Path Planning Depth and Model Variations 87

5.3 Prediction Errors for Varying Horizon and Evaluation Methods 87

B.1 Hyper Parameter Prior Distributions 106

Chapter 1

Introduction

1.1 Motivation

Autonomous robotic platforms provide the opportunity to gather data in ways that are not feasible for surveys conducted with direct human control or participation. Whether it be the cost of efforts involving large numbers of people, the danger or even complete inaccessibility for humans in some environments such the deep sea, disaster sites and exploring the universe, or relieving humans from repetitive tasks such as repeat surveying, autonomous robotic platforms are increasingly being deployed.

The ability to send an autonomous platform to explore an area with limited prior knowledge for the task of data collection requires the platform to interpret data as it is collected and adjust its sample selection locations or planned trajectories for both safety reasons, such as avoiding damage to the platform or the environment it is in, and to return with data that is most useful for the scientific objective of the mission. Regardless of the cost savings that an automated mission may provide, there will always be resource constraints limiting the amount of spatio-temporal coverage that can be achieved. For missions where the resulting data collected will be used to produce a model of some variable of interest, best use of the limited resources will be served by taking into account the quality of the predictions from the model whilst collecting the data and adjusting the sampling decisions accordingly. Estimating these models online also introduces the problem of limited computational resources and as such requires efficient models and algorithms to allow real time estimation and prediction

on embedded platforms.

Estuaries and tidal bays are important areas of our maritime coasts. They are widely used for recreational and commercial activities and provide habitats for estuarine marine life and breeding grounds for ocean marine life. Flows within these basins are highly dynamic due to the interaction of tides, freshwater outflows, winds and convection. The currents created can move sediment requiring frequent resurveying of navigational maps for boating and have complex effects on the mixing of salt and fresh water and the distribution of salinity, dissolved oxygen, pH, temperature and nutrients which impact marine life. An Autonomous Surface Vessel (ASV) could create navigational maps of these areas without the high cost of traditional surveying vessels. The dynamics of estuarine flows are generally modelled with fixed sensors, and occasionally with towed sensors or autonomous platforms performing fixed surveys. The ability to autonomously model a follow a dynamic tidal front would provide insights into how this turbulent process evolved spatio-temporally.

1.2 Problem Statement

This thesis focuses on the problems associated with efficiently using autonomous robotic platforms to collect data for the purpose of creating models of scalar fields. The focus is on the case where minimal prior information is available and thus the robots must build models of the data they are collecting and design sampling plans according to these models. The framework of Gaussian Process (GP) regression is selected due to the combination of the non-parametric approach it provides to model the data and the uncertainty information it produces when making predictions. Whilst the specific use cases presented are related to ASV's, the algorithms and methods produced are applicable to autonomous platforms working in any environment.

The first part of this thesis focuses on the problem of producing a full coverage bathymetric (underwater depth) survey within a given boundary subject to a minimum operating depth constraint for safety. This problem has two distinct parts. Firstly the ASV must model the bathymetry as it samples to find and follow the intersection of the depth contour and the boundary to safely create a new depth constrained boundary.

Secondly it must then efficiently plan and follow a path for coverage within this depth constrained boundary. The modelling and planning must be feasible to be conducted in real time on a small embedded CPU.

The second part of this thesis focuses on the problem of efficiently sampling to build a model of an estuarine tidal front. These fronts display location dependent noise due to the turbulent mixing process at the interface. Depending on the structure on the bathymetry and channel width these can either be stationary or moving. The standard GP framework must be adjusted to account for the location dependent noise, and in a manner that will be computationally tractable for embedded implementation. Planning for data collection within this model should be conducted to produce a model which quickly converges to as low an error as possible to maximise the return on limited resources.

1.3 Contributions

The main contributions of this thesis are as follows:

- Development of a suite of algorithms to simultaneously estimate a GP of bathymetry, with incremental Cholesky updates and online Hyper Parameter (HP) estimation, and find and follow the intersection of a depth contour and a bounding polygon. Efficient implementation of Gaussian Processes is demonstrated on a small Autonomous surface vessel.

- An extension to the Boustrophedon Cellular Decomposition, the Discrete Monotone Polygonal Partitioning (DMPP) is produced. This allows decomposition based on desired track width producing more efficient coverage when joining multiple polygonal areas. The order of joining the cells is decided in parallel with the cell decomposition producing shorter paths between areas and non-rectangular boundaries are explicitly dealt with.

- Demonstration of Autonomous depth constrained bathymetric mapping both in simulation and the field on a small ASV developed for the purpose.

- The case is made for explicitly modelling location dependent noise within a GP.

An efficient method for modelling location dependent noise with a parametric distribution is developed, Parametric Heteroscedastic Gaussian Process regression (PHGP). This is shown to be more computationally efficient than a more general method, Variational Heteroscedastic Gaussian Process regression. A new metric for measuring error in the second moment of the predicted distribution is presented, the Root Mean Square Standard Deviation Error(RMSSDE).

- Formulation of new kernels to model bathymetrically arrested and moving tidal fronts under PHGP. A parametric mean function is implemented which is simultaneously optimised with the HPs of the covariance function. This is shown to provide faster optimisation and lower prediction errors. The derivatives of the log marginal likelihood of all kernels w.r.t to the HPs are derived to allow fast estimation of the HPs through a gradient based solver.

- Comparison of a number of information theoretic measures for adaptive sample selection on simulated tidal fronts where Mutual Information is shown to provide superior predictive performance to random sampling or methods based on Entropy or Fisher Information.

1.4 Outline

This thesis is structured as follows:

Chapter 2 outlines the related literature and background in Coverage Path Planning, Gaussian Processes, Active Sensing, and Estuarine Tidal Fronts. The methods detailed here are extended in further chapters.

Chapter 3 presents an improved method for coverage path planning and a suite of algorithms for conducting autonomous depth constrained bathymetric mapping. These are tested in simulation and the field.

Chapter 4 establishes the case for explicitly modelling input dependent noise within Gaussian Processes and develops an efficient method to achieve this. A new metric for measuring errors in predicted standard deviation is defined. A new kernel and its analytic derivatives are presented allowing efficient implementation. A number of

information theoretic methods for adaptive sample selection are compared and Mutual Information is shown to provide superior predictive performance of the first and second moments of the distribution under PHGP by initially sampling widely then biasing sampling near areas of higher noise. This is applied to a simulated bathymetrically arrested tidal front.

Chapter 5 extends this model to account for moving processes. A Heteroscedastic extension to the standard squared exponential kernels and its derivatives is again presented. The use of a parametric mean function which is jointly estimated with the HPs of the kernel is shown to increase computational efficiency. Adaptive path planning is implemented and tested across a number of information theoretic models and planning horizons. An adjustment is made to the standard MI implementation in the temporal domain to temper the effect of the the curse of dimensionality from the addition of the time dimension. This is tested in simulation on a moving estuarine tidal front.

Chapter 6 concludes and discusses avenues for future research.

Chapter 2

Background

This thesis is focused on algorithms which enable efficient online adaptive sampling. Whilst applied to ASV's in estuaries, the algorithms developed are relevant to autonomous exploration in any domain. In this Chapter related work on water quality measures in estuaries, the relationships between them and the methods used to model them are reviewed. The dynamics of estuaries are presented, focusing on the salt/fresh water interface, highlighting the spatio-temporal dynamics involved. Turbulence is discussed, as the driving force behind heteroscedastic noise for which models are developed in Chapters 4 and 5. Physical models for simulating fluid interactions are examined highlighting the gap between high spatio-temporal resolution simulations and experiments of fluid mixing and the low spatial-temporal resolution of large scale physics models of oceans and estuaries. Related work in adaptive sampling is then presented followed by a review of Gaussian Processes (GPs) which is the modelling framework upon which this thesis is based.

2.1 Estuaries

Estuaries are important areas for a number of reasons. They provide habitats not only for estuarine marine life, but also breeding grounds for ocean marine life which is important both in its own right from a conservation perspective as well as for commercial and recreational uses. Many nutrients in coastal waters are provided by run-off from estuaries, driven by the flows from fresh water run-off and tidal mixing of salt and fresh

water. The edges of estuaries are highly populated and thus their health is important for the populations living and visiting these regions. They also provide shelter from the sea and are thus often used as ports. This heavy use for residential and commercial purposes puts the health of these areas under risk and as such there have been many studies on eutrophication (excess nutrient loading), and dissolved oxygen (DO) levels in estuaries around the world, i.e. [19, 27, 29, 46, 106].

The majority of environmental monitoring that has been undertaken to date in estuaries have had relatively sparse spatial coverage. There are numerous temporal studies which have been conducted via a small number of sparse fixed measurement stations, i.e. [8, 15, 137, 152]. These studies implicitly rely on the ability to extrapolate information to areas away from the sampling locations. It has been shown that there can be very significant changes in DO spatially on a transect from shallow water over a macrophyte bed to deeper water [50], and across a tidal interface [159]. DO has been studied in a bay in Norway [21]. The combined dynamics with other factors such as temperature and salinity, let alone tides is not discussed, and the distribution was assumed temporally static over the survey period (1 - 2 hrs), which would not work in areas subject to either tidal flux or organic processes measured in estuaries [8, 152, 159].

Salinity can be used to define the interface between salt and fresh water bodies being moved by tidal forces. These bodies of water may contain different level of nutrients, oxygen and suspended matter and have different temperatures. Some marine animals are also sensitive to the salinity levels in the water. The salinity and temperature levels will also impact the density. Density differences will define how the bodies mix or stratify. In the case of plumes of freshwater discharge into the ocean salinity has been shown to change significantly in value and variance on the scale of a few meters across saltwater/freshwater interchange [108, 109]. Whilst extensive spatial analysis has not been done on tidal fronts, data from high temporal resolution, but sparse spatial resolution fixed monitoring stations throughout an estuary show large changes in salintiy as the tide moves through [153].

There have been a number of studies looking at the density, current and salinity properties of the salt/fresh water interface of plumes where fresh water is ejected from the mouth of the estuary into the coastal waters [16, 83, 85, 95, 108, 109]. These have

not examined the dynamics inside the estuary and have ignored the spatio-temporal dynamics. Whilst some have measured through the water column, they have generally ignored the temporal component whilst traversing the scalar field and were more focused on determining the characteristics of the front in terms of distance from the front, rather than spatially across the whole regions around the front.

Remote sensing has been used to provide spatial coverage. The resolution of this is relatively coarse, on the order of 1km for satellites and several meters for aerial surveys [153]. These also need to be calibrated by field measurements and can have issues in shallow and clear waters due to optical readings returning information on the seabed rather than the water column. Aerial surveys are also restricted to providing spatial and not temporal coverage.

There is thus a need for fine resolution spatio-temporal sampling of environmental parameters in estuaries to calibrate and validate long term coarse resolution temporal studies. The highly dynamic nature of tidal fluxes within estuaries is important to understand due to the effect it has on the mixing of oxygen and nutrients between fresh and salt-water. High resolution spatial and temporal sampling of this dynamic front is an ideal candidate for a moving sensing platform. The shallow nature of some parts of estuaries lends itself to a small platform with minimal draft which can sample without disturbing the environment (i.e. though stirring up silt). The dynamic nature of the tidal fronts lends itself to a platform which can simultaneously model and plan in its environment.

2.1.1 Estuarine Tidal Dynamics

There are a wide range of factors influencing the dynamics of tides in estuaries. Various classification schemes have been created to describe the different types of estuaries. These can be based on Topography, Morphology or Salinity and Circulation Structure [37, 55]. The Topography classification divides into categories based on the geological processes that created them, such as Drowned River Valleys, Fjords and Bar Built estuaries. Morphological classification builds on this adding also the impacts of sediment, river flow, tides and waves. The Salinity and Circulation structure classifications instead look at the the distribution of salinity and structure of circulation within an

estuary. Classification on salinity may be divided into Highly Stratified with a lower salt water layer and an upper fresh water layer, Partially Mixed where the salinity gradient slowly decreases with distance from the mouth and Well Mixed with a relatively constant salinity level throughout the estuary. Additionally significant changes in bathymetry such as deep basins connected by shallow channels can create a number of abrupt fronts through the estuary as tides and currents push well mixed water through the channels into the basins, leading to classifications such as Front Dominated estuary [48].

Whilst these methods provide a general description of the average conditions in an estuary, they are not as useful in describing the dynamics temporally and spatially within a given estuary. For instance spatially across an estuary the salinity structure may transition from being well mixed at the mouth due to waves and tidal currents, partially mixed through the middle due to tidal action combined with the shape and roughness of the bathymetry to stratified further upstream where fresh water flows dominate [37]. Estuaries can also have large temporal differences in salinity structure if rainfall events significantly alter inflows of fresh water. A study on the Port Hacking estuary, south of Sydney, Australia [48], noted that the main body of the estuary is generally well mixed as fresh water inflows are of a similar magnitude to evaporation. After rainfall events, a large influx of fresh water flushes the upper reaches and leads to a stratified structure in the main basin. The incoming tide then brings well mixed water though a shallow channel and results in well defined front as it enters the basin.

In stratified estuaries, the sharp gradients in salinity are known as a Halocline. These bodies of water often have different temperatures and this temperature gradient is known as a Thermocline. Higher salinity and lower temperature increase the density of the water and thus an associated density gradient (Pycnocline) is also observed. The two bodies of water also have their own velocity structures. The movements of these bodies of water through an estuary can have biological impacts as in addition to the impacts of salinity, they are often also associated with changes in pH and dissolved oxygen. Various marine organisims have threshold levels which may have serious consequences, for instance juvenile weakfish with respect to dissolved oxygen [151] or the effect of acidity on the spawning of Perch, Smelt and Burbot [65]. Thus it is important

to model not just the average characteristics of an estuary, but the spatial and temporal distributions and to examine at the dynamics of individual fronts.

A schematic of a Halocline can be seen in Figure 2.1. This shows a cross section of a tidal front. Along the Halocline viscous friction is generated. Changing depth, width and roughness of the estuary can change the velocity ratios between the two water bodies temporally and spatially. Internal waves can form on the Halocline as the lower body of water follows the bottom structure. These perturbations in the Halocline due to internal waves combined with fluctuations in the velocity of the fluid on both sides lead to changes in the relative velocities of the water bodies.

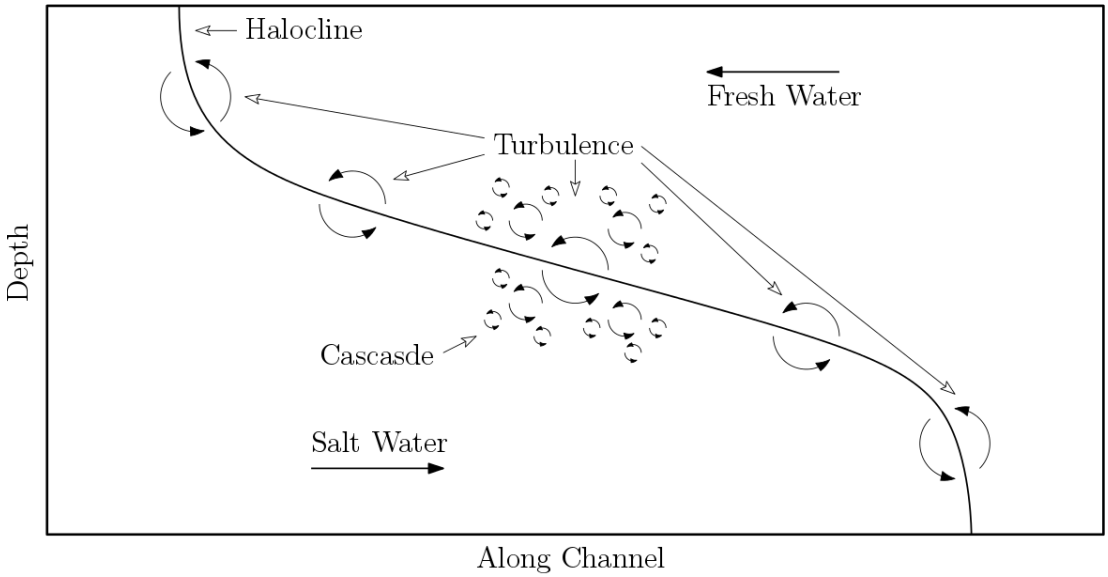


Figure 2.1: Schematic of the interface between salt and fresh water in a stratified estuary

Differences in density and velocity are important as they determine whether instabilities are absorbed or compounded. The Richardson Number Ri is driven by the ratio of density to velocity shear gradients:

$$Ri = -\frac{g}{\rho}\frac{\partial \rho}{\partial z} \Big/ \left(\frac{\partial u}{\partial z}\right)^2 \tag{2.1}$$

where g is the gravitational constant, ρ is density, z is depth and u is velocity.

This number defines whether there is a transition to turbulence. At levels of $Ri > 0.25$ the stratification is stable, though there is mixing in the form of entrainment where Holmboe waves form and small amounts of the slower moving fluid are drawn into

the faster fluid. For $Ri < 0.25$ a process called turbulent diffusion occurs in the form of Kelvin-Helmholtz waves [37]. These instabilities initially form large eddies which then cascade to smaller eddies until they are small enough (at the Kolmogorov length scale) that the energy can be absorbed by viscous friction [93]. This can be seen in Figure 2.1 where the size of the turbulent eddies decreases away from the initial large instability. The changing velocities in time and space lead to Ri oscillating through the critical 0.25 value. This leads to turbulence randomly forming and dissipating on the Halocline.

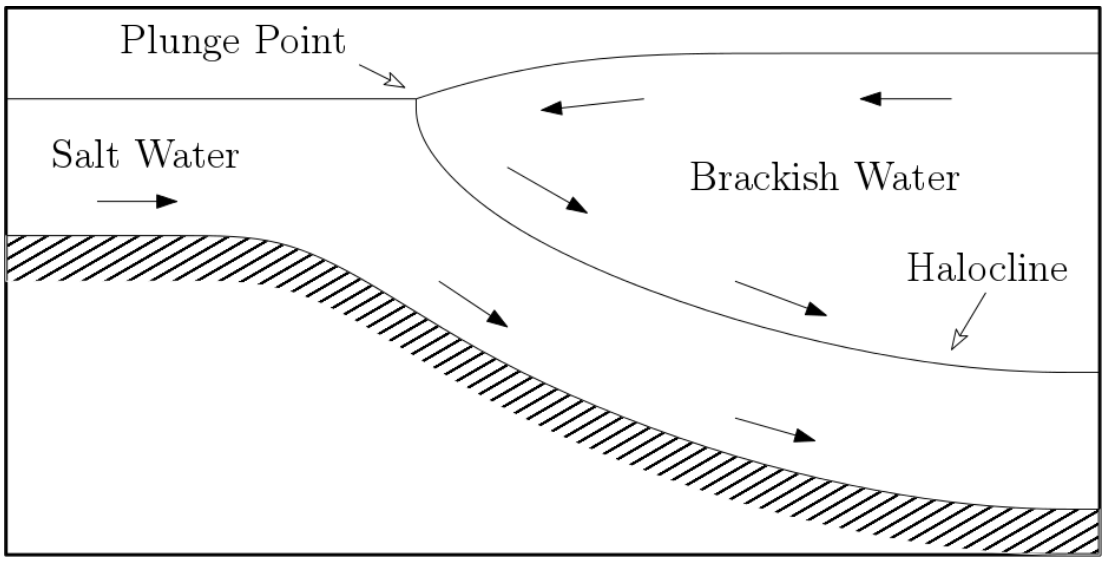
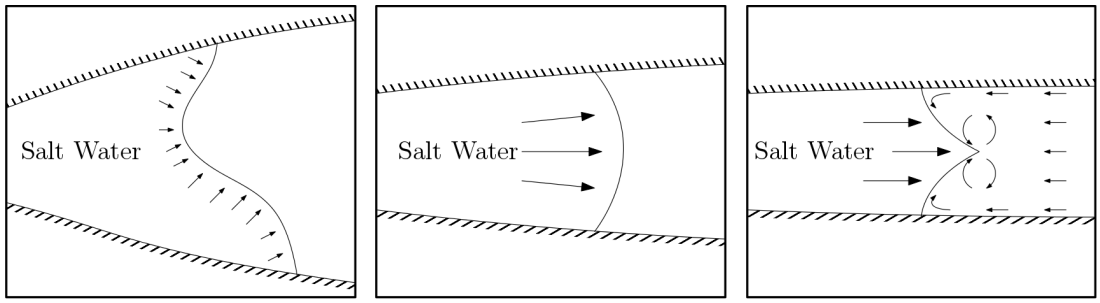


Figure 2.2: Streamwise cross-sectional schematic of tidal untrusion front plunge line. Based on Figures in [37, 66, 76]



(a) Class I (b) Class II (c) Class III

Figure 2.3: Surface manifestations of tidal front classes. Based on Figures in [76, 139]

A specific type of Halocline often seen in bar built estuaries is the Tidal Intrusion Front. These often occur on the upstream side of a shallow sandbar or channel where the incoming denser sea water on a flood tide reaches a deeper bay and plunges below the less dense estuarine waters. The phenomenon has been noted in estuauries around

(a)

Figure 2.4: Surface manifestations of a bathymetrically arrested (Class I) tidal front on an incoming tide at Lilli Pilli Point, Port Hacking, Sydney, Australia, 5th March 2017

the world [76]. Figure 2.2 shows a cross section of this phenomenon. The form that these fronts take in the cross stream dimension on the surface is a function of the width of the channel, width of the basin and the depth profile. Three general categories have been defined [76]. Class I fronts are created when the channel the tidal front progresses through is not constraining in width. When this front hits the drop in depth of the bay, the surface manifestation of the interface, the plunge line, follows the contours of the bathymetry. An example of this can be seen in Figure 2.4. Here the water is exiting a shallow channel on the left on a rising tide as the sand bar quickly drops away. A sharp line can be seen from the turbulant interface on the surface. This is the same location studied in [66]. Class II fronts are caused by a more constraining channel. In this case the plunge line forms a parabolic shape irrespective of the bathymetry. This is caused by the friction effects from the sides of the channel causing the front to be further ahead in the center of the channel. Class III fronts are caused by an even tighter constriction of the channel. In this case counter-rotating eddies form on either side of the center of the front causing a 'V' shaped front. In reality, the complex shapes of estuaries often mean along and cross stream velocities combine with sharp depth changes to produce plunge lines exhibiting a combination of the three types. The distance into the bay that these lines form depends on the combination of density and velocity differences. In the case of Class I fronts, as they are driven by stable bathymetric features, they are relatively stable in position throughout the flood tide [66].

2.1.2 Physical Dynamics Models

Laboratory experiments and CFD models of turbulence in the mixing of stratified fluids have been conducted for the lock exchanged problem [12]. This can be seen as a simplified version of mixing at the tidal front of a stratified estuary. These only focus on producing one realisation of the process. Whilst turbulence can be described by the deterministic Navier-Stokes equations, it is highly sensitive to non-linear terms. Even in laboratory environments, repeated experiments will not result in the same velocity field. Small perturbations in initial or boundary conditions result in different realisations of the process. If we can only define the initial conditions in probability, this is all we can hope to predict [7]. When this process is brought into the dynamic estuarine environment, with complex boundaries, these problems are only compounded.

The field of Uncertainty Quantification (UQ) examines ways of producing probability distributions from CFD models. Monte Carlo simulation is the simplest method to implement, where distributions are created for various parameters of the model, from which random values (potentially correlated) are drawn and the numerical solution solved. The slow convergence of Monte Carlo at $1/\sqrt{n}$ combined with the slow calculation of the numerical solutions themselves can often make this infeasible. Spectral expansions of the CFD model in the form of Karhunen-Loève or Polynomial Chaos (PC) representations propagate the parameter uncertainty through the model allowing distributional properties to estimated directly through a single numerical solution [82]. This has been applied to some problems of mixing in density interfaces such as particle laden flows into a less dense fluid [54] and the Richtmyer-Meshkov instability where two fluids of different densities are initially at rest through which a shockwave is passed through twice in [120].

There are a number of large physics based models for modelling the dynamics of oceans, coastal waters and estuaries, for example the Regional Ocean Modeling System (ROMS) [99] and the Semi-implicit Eulerian-Lagrangian Finite Element (SELFE) [166]. Due to the 3D numerical grids required to solve these models, resolution is limited. As such they will not capture the smaller scale dynamics of turbulent mixing around a front. These are deterministic models and produce one realisation of output predictions per run.

SELFE was originally designed for modelling the Columbia River estuary and plume. A recent application tested the accuracy of this model to the dynamics of the salt wedge in the Columbia River Estuary [71]. The fine grid model composed of 109,000 triangles. Due to the scale of the estuary covered at around 60km, with 17-34 vertical elements this still led to horizontal resolution of 180m. The finest time discretisation the model was run at was 36s. When predictions from the model were compared to physical measurements at 4 stations, they found Root Mean Square Error (RMSE) for bottom salinity ranged from 4.57 to 12.86 psu(practical salinity units, equal to g/Kg of salt), which for a salinity measure that ranges between 0 and 30, is quite large.

These models designed from solutions to the physical driving equations are also computationally intensive. For the CFD model in [12], whilst no run times are given, even this model of turbulence between two homogeneous fluids with smooth boundaries in an area 15cm deep and 30cm long over a time period of 50s, required tens of thousands of nodes, time steps of 0.01s and and array of computers to run. It is obvious this type of model will not scale to simulating an estuary. UQ models, as more complex versions of these, will be slower.

For the SELFE model, a version of the run in [130] with 54000 elements, and 30s time steps (approximately half the size of the finest grid in [71]), found the optimal configuration was 20 servers with 4 cores at 1GB RAM per core taking 8 hours of run time per day simulated.

2.2 Adaptive Sampling

From production line robots that work in relatively deterministic environments through robots doing simple tasks in dynamic environments like the iRobot Roomba robot vacuum cleaner to current state of the art robots performing complex tasks in dynamic environments such as the Mars Exploration Rover, there are various levels of autonomy in robotic systems. Adaptive sampling is the process where decisions on where, in space and/or time, to sample are made using information collected from an environment. Online adaptive sampling is when these decisions are made whilst the data is being collected from models built in real time. Related literature in which robotic systems

have been used to adaptively sample is now reviewed.

An early paper on autonomous exploration is [158]. They show how sensor view point can be planned based on model uncertainty thus driving the exploration behaviour of a robot to build a more accurate understanding of its environment. They prove convergence when applied to linear models and experimentally apply to super ellipsoid models to verify the applicability of local linear approximations. A method for driving autonomous exploration in robots based on frontiers, which are defined as the boundary between explored and unexplored space is introduced by [163]. The robot will always move to the nearest unexplored frontier thus greedily optimising a method to increase explored space. The authors in [87] look into the issue of robot exploration under the dual objective of reducing the global uncertainty about the environment and the uncertainty of the pose of the robot within this environment. They create a weighted linear combination of the two information metrics which is greedily optimised at each time step. This is extended in [86] by adding a third element to the mix. Instead of greedily optimising for the next time step, motion is allowed anywhere on the map but total utility is reduced by the cost of this travel. The relative weights of the distance and information costs are empirically tuned. It is not clear that there should be any stability in the relative weights on these different measures as the survey area changes in size thus limiting general application.

An adaptive sampling algorithm which uses nested stratified random sampling is introduced by [125]. The environment is sampled first at a coarse resolution. A threshold variance is defined and any grids with variance above that threshold are split and re-sampled. This process is iterated until there are no areas with variance above the threshold. Travel costs are not taken into account which limits this approach to a general robotics exploration task. Given a forecast of a moving ocean plume, it has been shown that an underwater glider can adjust its path online to either head towards the center or the edges of the plume based on its distance to these estimated locations and thus collect samples from this moving plume [141]. It has also been shown that when prospecting for hydrothermal vents, detours of small spirals from a coarse lawnmower pattern based on redox potential greater than a manually set threshold can increase sampling in interesting areas [41].

A simulated study with an AUV with a gimbaled sonar is shown to outperform on localisation tasks compared with a fixed sensor [128]. A particle filter is used for localisation. This allows for calculation of entropy and this entropy is used to test the sonar angle at each time step such that it scans at the angle which has the greatest reduction in expected entropy. A method for active localisation of an AUV using multi-beam sonar data is presented by [39]. Using a prior bathymetry map, for each 30m segment they compared straight line continuation, random direction heading and choosing the most informative straight line segment out of 8 cardinal compass directions. It was shown that the active approach outperformed the others, especially in the case where straight line would cover featureless terrain.

A method for autonomous sampling in an AUV was developed by [97]. They use a Hidden Markov Model (HMM) trained on prior data to determine online whether the AUV is in an area of interest, in this case an inter-nephoid layer(INL), based on optical data. If so, a sample is taken. The HMM is also used to examine the probability of having seen the INL in the previous transact and this information is used to decide the width of the next transect, thus providing adaptive sampling resolution. A utility based technique for adaptive sampling is developed by [44]. Their process uses knowledge of current and past utility combined with awareness of mission parameters such as remaining mission time to decide when to take a limited number of physical samples of the environment. The adaptivity is introduced through a sampling trigger which changes based on distance from previous samples, number of samples taken and time remaining.

Exploration for underwater hydrothermal vents in an AUV is examined by [41]. Initially a coarse survey plan is set out. Whilst the AUV is following this path, the detection of a hydrothermal plume triggers a tight spiral action to better localise the source, before continuing on the coarse survey plan. An initial trigger level for the anomaly and a suggested number of spirals is set. The trigger level is adjusted adaptively throughout the mission by the ratio of samples taken/suggested samples vs. percentage survey time remaining.

The on-board autonomous science investigation system implemented on the Mars Exploration Rover is discussed in [18]. Limited bandwidth means on-board analysis must

be conducted as to which images to send back to earth. When dust devils and clouds are detected those images are sent back, whereas empty images are not. Rocks are detected though stereo imaging or on mono images through boundary detection. The rocks are analysed against a previously determined list of target signatures as well as a novelty score against other rocks in the regions. Any rocks identified by either of these two methods are flagged for download and added to the science goals of the mission. The on-board planner then creates a new plan using the methodology described in [124]. This is compared to the previous plan and the one with the highest optimisation score is processed. Additionally an online classifier is run which groups rocks into classes and thus allows a representative sample of images from each class to be returned to earth.

Using the underwater robot presented in [133], [47] present an online classification scheme which produces a surprise score which is used the drive the speed of the AUV. They use Latent Dirichlet Analysis (LDA) to compute topics over a spatio-temporal neighbourhood. To keep computational time constant to allow online calculation they use a variation of Gibbs sampling where they draw a constant number of observations at each iteration from a beta distribution which gives higher probability to recent observations. They produce summaries of observations based on an extremum summary which minimises the distance of the worst outlier to the summary. New observations are then compared to this based on their Hausdorff distance to compute the surprise score.

The concept of novelty rather than fitness as a utility function for driving search behaviour is examined by [24]. Instead of optimising a user defined function, novelty based search selects features of interest based on how different they are from prior observations. They found in complex tasks that novelty alone could not be relied upon to beat fitness based operation, and parameters could always be set to make novelty perform badly. When used in combination with fitness, superior outcomes were achieved over fitness alone. This was tested in simulations on the standard Tartarus problem [5].

As the complexity of the environments robots operate in increases, it becomes infeasible for the robot to be programmed with exactly how to respond to every scenario it will

encounter. Methods for robots to actively learn models of policies of actions that achieve a given desired effect, called inverse models are examined in [6]. Efficient sampling to learn these models is difficult due to the high dimensional, continuous non-stationary nature of the space. They introduce an active learning approach in which the robot samples novel tasks in the task space which triggers goal directed learning of the action parameters to solve it. This inverse approach leads to dense sampling only in the area of action policies that are useful to achieve the desired tasks, as opposed to a forward model where actions leading to tasks result in sampling being directed in the action space. Regression is used to infer actions that will be useful to complete a given task based on previous action/task observations. Level of interest is calculated by the speed of change in progress towards reaching goals. This interest level is used to direct areas for new goal creation. New goals are split between areas of highest interest, the whole space and low interest areas (typical split 70/20/10), chosen randomly within each area. Additionally they reduce the initial set by defining a rest position that can be reached without planning. This method is analysed in the context low level actions (i.e. individual actuators) and the tasks are movements of the robot.

A paradigm for planning based on the stochastic effect of actions is the Markov decision process(MDP). This assumes that the world is fully observable. MDP's have been used for the case of path planning of underwater gliders where the ocean currents introduce a stochastic effect on control actions [115].This is used to plan paths to minimise the risk of collision with traffic in shipping lanes and the ocean floor. This was implemented on field studies where the planner was run on an offline computer and the plans communicated to the gliders when they resurfaced.

The extension to the more realistic environment that robots face with uncertainty in perception is the Partially Observable Markov Decision Process (POMDP). To solve the POMDP problem for robotics a value function must be maximised (possibly discounted) over a given time horizon (finite or infinite), a given a set of states, actions and observations, a distribution of moving between states given actions, a distribution of observing states after taking actions, and an initial belief function. This can be either continuous or discrete. An issue with applying the theoretical framework of POMDP's to the field of robotics planning is that an exact solution to the problem

quickly becomes intractable due to the high dimensionality of the problem [68].

Point Based Value Iteration(PBVI) was introduced as an approximation to th full POMDP problem [117]. This approximation reduces the dimensionality of the problem by discretising the belief space and restricting the value function to only those relevant to those belief points. This compares to the exact method which will solve value the function for belief spaces that are highly unlikely or even impossible for the robot to reach. The authors apply it to a real world sized problem (870 states). They prove that the error introduced by this approximation is bounded. As opposed to the exact solutions which scale in complexity exponentially with the number of states (curse of dimensionality) and exponentially within the planning horizon (curse of history), PVBI is shown to scale only quadratically with the planning horizon. The Monte-Carlo POMDP framework reduces the dimensionality of the problem compared to the exact method in continuous belief and state spaces by sampling those continuous distributions via Monte-Carlo methods [147]. A nearest neighbour approach is used for interpolation to apply the finitely sampled distribution across states. Mixed observability MDPs (MOMDP) combine the attributes of MDPs and POMDPs by having some staes fully observable, and asome states partially observable [111].

There has been some progress on the efficiency of POMDPs. The second version of Hueuristc Search Value Iteration (HSVI2) advances PBVI methods by maintains bounds on the optimal value function allowing the use of heuristics for action and observation selection [143]. Another method, Successive Approximations of the Reachable Space under Optimal Policies (SARSOP) has shown further improvements in effciecy [75]. Whilst these and other approximations can speed up the POMDP framework, they still suffer from a method which is by design very high dimensional in its representation of the problem. For example, in the rock sample problem of planning in a small known grid, which rocks to sample, the planning time to solve under SARSOP and HSVI2 was 400 and 250s respectively [75]. Thus whilst interesting methods, these are not applicable to the objective of this thesis of real time online planning.

2.3 Gaussian Processes

The dynamic spatial and temporal characteristics of tidal fronts lends itself to modelling through a spatio-temporal Gaussian Process(GP). The use of a spatio-temporal GP for planning for sensor placement (ie where travel time is not considered) for the monitoring of environmental processes has been shown by [140]. It has also been demonstrated in simulation for the case of planning for energy gain and some exploration in the case of aerial soaring by [79]. An efficent non-stationary spatio-temporal model able to be learnt online is developed in [45]. In terms of modelling a tidal front, rather than implicitly characterising it through a non-stationary GP, it may be preferable for planning purposes to instead use a stationary GP for classification of the location of the front. This would be a proxy for the full non-stationary GP, which could still be fit in post-processing of the data to create the model of the scalar field, to allow faster implementation. Spatial sampling design with stationary Gaussian processes is analysed by [169]. They propose two algorithms for optimal sample design. GPs are applied to construct a probabilistic habitat map by [129]. They use Monte Carlo sampling to identify which mission from a set of potential mission plans would result in the greatest reduction entropy. Closed form bounds are provided pre-simulation to prune unlikely missions and thus reduce the number of simulations required. This is extended by [10] to the case where a fixed trajectory can be placed anywhere on a GP modelled area. GPs are used to model a wind field whilst simultaneously using that wind field to provide the energy required for exploration to refine the model by [80]. For stationary wind fields the three dimensions of the wind field are separately modelled though the use of a single set of shared HPs utilising a stationary covariance function. For the modelling of dynamic wind fields both separable and non-separable spatio-temporal covariance functions were analysed. The more complex non-separable model was found to be slower and less stable than the simpler separable function. The insight that wind fields drift over time was used to create a drifting component in the separable covariance function. At each planning cycle the point of maximum entropy in the map is located. If the vehicle has enough energy to reach this point, that is the current goal, otherwise the glider greedily exploits the wind field. GPs are applied to the case of path planning for an energy constrained solar harvesting robot by [119].

They estimate the solar field in an environment with shading structures using a GP and then plan energy efficient paths though this field.

Gaussian process(GP) regression is described in [127]. This is a very powerful method method for estimating a spatial process. It is a covariance based procedure where inference is strong close to observed data points, but as the distance from observed data increases the confidence of the prediction drops. Similar methods are called Kriging in the geostatistics literature and Least Squares Collocation(LSC) in geodesy. The model provides both a method for prediction at unknown points and variance information. A benefit of this method is that these variance or entropy maps can be used to guide robot path planning. Sparse representations of GPs using a greedy approximation algorithm to allow their efficient use on large data sets is developed by [23]. The problem of large data sets can be addressed though the parametrisation of a specific set of sparse covariance matrices that allows for exact inference [98]. Generally GPs are modelled with a mean function of zero. Trends or periodic cycles can be fit through the use of a parametric mean function. The standard covariance function used in GPs is extended to cover the case where there is Autoregressive Moving Average (ARMA) correlated noise by [101]. This provides more flexibility than the generally assumed Gaussian independently and identically distributed (i.i.d) white noise processes. The general method in the machine learning community is to handle these in a non-parametric fashion through the covariance function. An additive covariance function could be created which included a long length scale kernel to capture trends and a periodic kernel to capture periodicity [112].

The concept of using a combination of stationary GPs to model processes with spatially varying covariance is introduced by [150]. Kernel convolution is one method to fit non-stationary covariances. An example of this is provided by [26] for fitting gravity fields. They segment their data geographically into relatively stationary components and fit elliptical kernels to each segment. Estimation is conducted with an adjusted covariance function that combines the local covariances for each point in a pair. A more principled approach is to use Markov Chain Monte Carlo (MCMC) to optimise the HPs [113]. An approximation of this approach is to use point approximation rather than full MCMC integration. It has been shown empirically that this can lead to similar results at

speeds comparable to a stationary GP. The HPs can be recursively [72] or jointly [118] estimated. Alternatively, through a transformation of geometric space to a deformed space, a stationary covariance function can be given the properties of a non-stationary one [25, 134]. Both of these methods require estimation in high dimensional spaces and are thus likely to scale poorly. Non-stationarity of the covariance can also be achieved through a neural network kernel which has spatially varying HPs in each dimension [107, 155].

Generally GPs only predict independent outputs. Whilst joint prediction of dependent outputs can be done (called co-kriging in the geostatistics literature), there are issues with ensuring positive definiteness of the joint covariances. One solution to this problem is to model the outputs as a multidimensional stable linear filter with gaussian noise. The result of this is a GP with dependent outputs [13].

The kernels used in GPs need to be Mercer kernels. This requires they are Positive Semi-Definite(PSD). Apart from this restriction they can be quite flexible. Note that this restriction allows them to be inverted through the use of Cholesky decomposition, which is much more efficient than a regular matrix inverse operation. It can be quite difficult to prove that a given type of covariance matrix will be PSD. There are a number of ways it can be ensured that the constructed kernel is valid. One is to use Bochner's theorem which allows the construction of a kernel from the power spectrum of a function. Alternatively kernels can be created from combinations of other valid kernels. There at four methods which are guaranteed to create a valid kernel from combinations of valid kernels. Kernels can be input scaled (changing the signal variance), they can be output scaled (changing the length scale), they can be added (logical OR similarity) or they can be multiplied(logical AND similarity) [58].

Additive covariance kernels can have very strong predictive power [34]. This predictive power is also quite dangerous as it is proposing a very strong prior on the model, and thus must be carefully evaluated. All valid kernels are also reproducing kernel Hilbert spaces(RKHS). Hilbert spaces allow the used of euclidian geometry in infinite dimensional spaces. This is important for GPs as they are defined in infinitely dimensional function space. Functional ANOVA analysis can be used to analyse the effects of given components of a kernel [33].

The HPs of the GP can be quickly estimated through maximum likelihood estimation. A full Bayesian treatment here would require marginalising out the HPs. This is not generally analytically tractable, and thus sampling algorithms are required. Whilst this can be done with Metropolis Hastings MCMC, it can be quite slow to converge. An alternative is to use slice sampling [100].

These methods work well for batch processing of data. If conducting inference on sequentially arriving data, retraining the model from scratch as each new observation arrives is throwing away a lot of information and thus computationally expensive. If the previous covariance and inverse covariance matrix are kept, sequential updates to these objects can be applied at a lower cost of $O(n)^2$ rather than $O(n)^3$ [112].

Rather than using a GP, it is possible to model directly in the RKHS. It has been shown in [126] that a model of occupancy can be made by using a kernel on the data directly. They achieve speed by using stochastic gradient descent to optimise over a logistic function for which the log-likelihood reduces to a sum over the points which is convex in the parameters. This is done with a purely frequentist approach, and they mention a fully Bayesian extension whilst possible, would lose some of the properties which make their solution fast.

Under a Gaussian Process it is assumed that the joint distribution of training points X with realisations $\mathbf{y}$ and test points X_* with realisations $\mathbf{y}_*$ is jointly normal [127]:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{m}(X), \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{2.2}$$

where $\mathbf{m}(X)$ is a mean function (often assumed to be zero), K is the covariance matrix created by the chosen kernel function, σ_n^2 is the observation noise and I is the identity matrix. The formula's for the expectation and variance of the test points are as follows:

$$\hat{\mathbf{y}}_* = K_*(K + \sigma_n^2 I)^{-1}(\mathbf{y} - \mathbf{m}(X)) + \mathbf{m}(X) \tag{2.3}$$

$$\hat{\mathbf{\Sigma}}_* = K_{**} - K_*(K + \sigma_n^2 I)^{-1} K_*^T \tag{2.4}$$

where for compactness $K = K(X, X), K_* = K(X_*, X)$ and $K_{**} = K(X_*, X_*)$, and are using the convention of lower case letters for scalars or vectors and upper case for matricies.

The simplest kernels used in GPs are stationary. In these the variance of the output given at a test point is based on the distance between that point and the sample input points. For example the squared exponential kernel:

$$K(X, X_*) = \sigma_f^2 \exp\left(-\frac{1}{2}||X, X_*||^T \Sigma^{-1} ||X, X_*||\right) \tag{2.5}$$

where Σ is a diagonal matrix populated by the square of the characteristic length scales on each input dimension. If these length scales are set equal, the kernel is isotropic, with the same relation across the dimensions. Relationships between observations which change with movement across an input space can be modelled with a non-stationary kernel, examples of which can be found in [127].

A fully Bayesian treatment would marginalise out the HPs. For computational speed Maximum-Likelihood (ML) estimation is often used. To conduct ML estimation a function for the likelihood of the data given the HPs is required. For computational stability reasons the log transformation of this number, the Log Marginal Likelihood (LML) is often used and is defined as:

$$\log p(\mathbf{y_{adj}}|X, \theta) = -\frac{1}{2}\mathbf{y_{adj}}^T K_y^{-1} \mathbf{y_{adj}} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log 2\pi \tag{2.6}$$

where $\mathbf{y_{adj}} = \mathbf{y} - \mathbf{m}(X)$, I is the identity matrix, n is the number of observations and $K_y = K + \sigma^2 I$.

Cholesky factorisation is generally used instead of the direct matrix inversion required in Equations 2.3 and 2.4 for both numerical stability and speed. Even with this, the calculation of K when fitting the GP is $O(n^3)$. Once K is calculated, the partial derivatives of the LML w.r.t θ are relatively inexpensive to calculate which allows fast numerical gradient descent techniques to be used. For the SE kernel, with a length scale of l, this is:

$$\frac{\partial}{\partial \theta_i} \log p(\mathbf{y_{adj}}|X, \theta) = \frac{1}{2}\text{tr}\left((\alpha\alpha^T - K_y^{-1})\frac{\partial K_y}{\partial \theta_i}\right) \tag{2.7}$$

$$\frac{\partial K_y}{\partial \sigma_f^2} = \exp\left(-\frac{\|X, X\|^2}{2l^2}\right) \tag{2.8}$$

$$\frac{\partial K_y}{\partial l} = \frac{\|X, X\|^2}{l^3}\sigma_f^2 \exp\left(-\frac{\|X, X\|^2}{2l^2}\right) \tag{2.9}$$

$$\frac{\partial K_y}{\partial \sigma_n^2} = I \tag{2.10}$$

where $\alpha = K^{-1}\mathbf{y_{adj}}$.

When predictions from the GP are desired with n training points $\mathbf{x}$ and m test points $\mathbf{x}_*$ this would be calculated as a batch and thus be dominated by the calculation of K, K_* and K_{**} which would be $O((n+m)^3)$. For computational efficiency, if multiple predictions are to be made on the same set of test data, care should be taken to design algorithms such that all predictions are made in one batch. For instance predicting in a batch would be $O((n+m)^3)$ rather than $O(m * (n+1)^3)$ if calculated sequentially, which in the case of $n = 500$ and $m = 50$ would be 38x faster.

Chapter 3

Adaptive Path Planning for Depth Constrained Bathymetric Mapping

3.1 Introduction

Estuaries and tidal bays are valuable resources. They provide shelter from the sea and are often used as ports. They are also used for the disposal of residential and industrial waste water and products. The flow of water from tides, run-off and storms can significantly impact the bathymetry of these areas through the movement of large volumes of sediment. Anthropomorphic impacts in these areas through land reclamation on the waters edge, the construction of jetties, drainage impacts from land use change, dams, break walls and boating traffic impact the flow of water and sediment in these areas often resulting in large changes to the composition and dynamics of the bathymetry [92].

Navigational maps are important for the safe passage of recreational and commercial boating traffic. Traditionally these are created with sonar data collected from surveying vessels. The high cost of conducting these surveys impacts the frequency of re-surveying [136]. An Autonomous Surface Vessel (ASV) able to conduct these surveys autonomously could significantly reduce this cost and thus enable more frequent

surveying to occur for a fixed budget due to lower equipment and personnel costs. An additional benefit arises from the fact that the smaller draught and reduced thrust of the ASV will allow surveying in shallower waters and create a smaller wake resulting in less disturbance to the shoreline in sensitive estuarine environments such as mangroves. To behave in an Autonomous manner in unknown environments requires the ability to sense and interpret the environment, which requires building a model of the environment in real time as it is explored, and making decisions based upon projections from this model. Using a modelling framework which produces uncertainty estimates, such as Gaussian Process (GP) regression, allows the maps produced to be used knowing the amount of confidence the model has in its projections across space.

The contributions of this chapter are as follows. A suite of algorithms is developed to autonomously estimate the bathymetry of bounded area with a minimum depth constraint. Efficient implementation with incremental Cholesky updates and online HP estimation is demonstrated with real time operation in the field on an ASV built for the task. A new method for efficient decomposition of non-convex polygons is presented, the Discrete Monotone Polygonal Partitioning. Content from this chapter has been published [1].

The remainder of this Chapter is organised as follows. Section 3.2 presents related work, reviewing the current state of the art in autonomous route planning. Section 3.3 details the algorithmic suite developed to enable the autonomous bathymetric surveying. Section 3.4 and 3.5 then test these algorithms in simulation and the field. Section 3.6 summarises the work.

3.2 Related Work

Current robotic surveying work often involves pre-planned survey paths which require prior information on the area to be surveyed and cannot react to information as it is received [21, 53, 67]. Creating the optimal back and forth path for coverage of an area whilst staying within the workspace, which is referred to as a lawnmower path in the robotics literature [43] or an axis parallel solution to the milling problem

[1] T. Wilson and S. B. Williams, "Adaptive path planning for depth-constrained bathymetric mapping with an autonomous surface vessel," *Journal of Field Robotics*, vol 00, pp. 1-14, 2017.

in computational geometry, has been shown to be related to the Travelling Salesman Problem (TSP) and thus NP hard in general [4]. By partitioning the complex workspace into a number of simpler shapes, which can easily be solved, and then joining these spaces together, it is possible to produce feasible paths for coverage in polynomial time. The joining together of these cells themselves optimally is also a TSP problem, and thus approximations must be used here as well for polynomial time solutions.

A polygonal workspace can be spilt into its elemental trapezoids, known as the Trapezoidal decomposition [77]. These trapeziods are convex, and thus lawnmower paths in any direction can cover the space. This method whilst simple to implement, can result in an excessive number of elemental cells, which can lead to a large number of inefficient transit paths to join these together. Some of these cells could be merged back together to create larger elemental convex polygons and then joined. Heuristics for optimising the orientation of the tracks within each cell can also be implemented [110]. Alternatively, a sweep direction can be chosen through the polygon, which is then split into elemental polygons which are monotone to this sweep direction. A polygon is monotone to a sweep direction if lines orthogonal to this sweep direction cross the polygon at most twice. This guarantees by construction that these elemental polygons can be covered by tracks orthogonal to the sweep direction. This also means that all cells must be covered by tracks in the same orientation. This method was introduced as the Boustrophedon Cellular Decompostion (BCD), [20]. It results in a smaller number of elemental cells than even the merged version of the Trapezoidal Decomposition, thus reducing the number of transit paths between them. This is not guaranteed to reduce the total transit length, especially if using heuristics to solve the transit paths. Reducing the number of cells is especially important for online implementations given the optimal solution is NP hard in relation to the number of cells. A graph of the elemental cells and their neighbours is created during the decomposition, and a simple search through the graph to the next anti-clockwise cell is conducted to join the cells together. This approach has been extended to non-polygonal workspaces and non-linear sweep lines as the Morse Decomposition [2]. The BCD has also been expanded to work online in unstructured environments [1].

The advantage of adaptively reacting to sensed information has been demonstrated

[41]. The authors used a chemical sensing threshold based on median differences in a small moving window compared to a large moving window to trigger a detour from a coarse survey path into a fine spiral for increasing coverage in areas likely to contain hydro-thermal vents. Whilst the trigger level was adapted throughout the mission with both the moving windows and a factor based on percentage of spirals used vs percentage of mission completed, the absolute level of the trigger was adjusted by a pre-tuned parameter. It is not clear this parameter would easily be set without extensive prior surveying and would significantly impact its ability to start spirals in the most informative regions.

Using sensor data to build and plan within a model in real time offers the potential for surveying in unknown environments to produce models balancing levels of certainty and resource use. GPs offer a useful framework to deal with estimation under uncertainty and planning can be conducted on both mean and variance information. There is a growing literature on planning within GPs, for the problem of sensor placement, which ignores travel times [73]. Other methods such as Locally Weighted Projection Regression (LWRP) by [156] could be used. As discussed in [105], this method whilst potentially faster is more complex to implement and requires manually tuned parameters.

A GP is used to model bathymetry with an Autonomous Underwater Vehicle (AUV) in [64]. Whilst active planning is implemented based on the model predicted, this is all conducted offline either between dives or post mission on segments of dives. The HPs of the GP are optimised once off an initial dive. GPs are fit online, with HPs optimised off prior data by [144] and [89] to model terrain roughness and light distribution respectively with a ground vehicle. Greedy adaptive sampling is driven through a function based on the predicted mean plus uncertainty predictions, adjusted by a distance function which is useful in situations where maximum values are the primary interest such as in environmental pollutants. In a simailar vein, Level Set Estimation uses the mean and uncertainty predictions from a GP to classifiy areas into above, below or uncertain relative to a desired threshold [51]. This was conducted on prior data and the HPs were optimised from either a subet or all of the real data.

A GP is modelled to an environmental dataset of 2024 points in a vertical 1D transect of

Chlorphyll-a measurments in Lake Geneva. From this 10,000 simulated points are created, though without any noise re-introduced. These simulated points are then added either sequentially individually or in batches based on uncertainty and the resulting prediction performance of classification is compared by both number of samples, and normalised travel time. It has been shown in [79] that wind fields can be estimated with a GP and potential paths evaluated on the joint objective of reducing the uncertainty in the map and retaining enough potential energy to keep a glider aloft. Whilst the HPs are estimated online, this is done in simulation and the function used highly constrains the resulting parameters to an area close to a defined prior. Thus whilst GPs have been implemented online for planning purposes, their HPs are generally learnt off-line and I am not aware of any studies that demonstrate online learning of HPs in the field.

There have been a number of published studies on the design, development and testing of small autonomous surface vessels for robotics research in recent years. Twin hull vessels have the advantage of being relatively stable in roll and have been implemented by a number of authors [31, 62, 94, 149]. All but the last of which used differential thrust for increased manoeuvrability, allowing rotation on the spot. Some examples of larger autonomous surface vessels are the full scale catamaran used for methane sensing on an inland dam [32], the MIT AutoCat and Kayaks [88], Swordfish [40], Delfim [114] and the Springer USV [102]. There has also been some commercial development in ASV's such as the Wave Glider by Liquid Robotics [60] and the Saildrone [132].

These vehicles are generally underactuated and nonholonomic. This reduced number of degrees of freedom in their action space requires trade-offs to be made in the control algorithms. The environments they are deployed in also generally have external forces in the form of winds and currents acting on the vessel. Station keeping of a vessel in the presence of external forces has been shown to be possible with thrust control proportional to distance to target and yaw control using a proportional and integral controller [94], and a full PID controller has been demonstrated for speed and line following [61].

In this work it is shown how building a model of the bathymetry and simultaneously planning within this model allows exploration of the intersection of a depth contour and a bounding polygon in an unknown environment under uncertainty. The Discrete

Monotone Polygonal Partitioning algorithm is presented to produce elemental cells that can be exactly covered by the desired lawnmower path width thus resulting in more efficient paths than the BCD, and then fit a path through these cells for coverage of this area to build a model of the bathymetry. Methods for efficient updating of the GP are implemented to allow online prediction of the bathymetric profile and estimate and update the HPs as data is collected in real time. This is tested in both in simulation and in the field with a small low cost ASV built for the purpose.

3.3 Autonomy Suite of Algorithms

The aim of this study is to provide a suite of algorithms to allow an ASV to operate in an unstructured environment, with minimal prior information, and to autonomously explore the area and return a map of the bathymetry.

GPs have been chosen to model the bathymetric contours. Initially bivariate splines were tried as detailed in [30] and implemented in the Fortran routine SURFIT. This led to large instabilities resulting very quickly at short distances from sampled points. A zero mean function is set for the GP. This was to both to keep maximum flexibility by not assuming a parametric model for the mean and simplicity by not introducing extra parameters to estimate. Specification of a mean function has most impact far from sampled points when the covariance kernel has little impact. As the algorithm presented only searches relatively close to currently sampled points this extra complexity would be unlikely to add much value. The Squared Exponential kernel as defined in Equation 2.5 is used with the same length scale l for both input dimensions (the diagonals of Σ).

There are three main components to the algorithm. Firstly, the GP which is updated with data as it is collected by the ASV. For this to run online on a small embedded CPU care must be taken in how the GP is updated and analytical gradients are used for efficient estimation of the HPs. Secondly, an algorithm is developed to follow the intersection of a bounding polygon and the depth contour as predicted by the GP. Thirdly, once this has concluded, Discrete Monotone Polygonal Partitioning (DMPP) is proposed as an efficient method to decompose the resultant intersection polygon allowing

a lawnmower pattern to be planned for coverage which the ASV then follows.

3.3.1 Online Gaussian Process Updates

The main computational load in fitting and predicting from GPs lies in the Cholesky factorisation of the covariance matrix which is $O(n^3)$. When adding data, or predicting m test points, instead of recalculating the entire Cholesky factorisation, the extra columns and rows related to the new data/test points can be simply calculated, and then added to the matrix already calculated. This can be done exactly for the case of additions and approximately for deletions as detailed in [135]. This is applied in [112] for fitting GPs to sensor data and [105] in using GPs to model control of robotic joints. Outside of robotics GPs are generally used on a batch of data once it has been collected. Optimisations in software implementations of GPs focus on sparsifying or reducing the size of the covariance matrix to reduce computation time on one batch of data. The author is not aware of any GP packages which implement incremental data updates and as such this procedure for updating the Cholesky matrix is breifly described [112].

The positive semi-definite (p.s.d) covariance matrix is K_{11} and its upper triangular Cholesky matrix L_{11}. When adding new data points to the end of the covariance matrix it becomes $\begin{bmatrix} K_{11} & K_{12} \\ K_{12}^T & K_{22} \end{bmatrix}$. The new elements K_{12}, its transpose K_{12}^T and K_{22} are calculated from the new data or test points using Equation 2.5. The resulting Cholesky Matrix is $\begin{bmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{bmatrix}$. For a triangular $\mathbf{A}$ backwards substitution can be used to solve $\mathbf{Ax} = \mathbf{b}$, defined as $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$. This leads to the following solutions for the elements of S:

$$S_{11} = L_{11} \tag{3.1}$$

$$S_{12} = L_{11}^T \backslash K_{12} \tag{3.2}$$

$$S_{22} = \text{Chol}(K_{22} - S_{12}^T S_{12}) \quad \text{or for } d = 1 \quad S_{22} = \sqrt{K_{22}} \tag{3.3}$$

where Chol() signifies the Cholesky decomposition and d is the number of dimensions

in the inpur space. For m new training/test points this reduces the update step from $O((n+m)^3)$ to the larger of $O(n^2)$ or $O(m^3)$, due to Equations 3.2 and 3.3 respectively. The GP is run on its own thread and its covariance and Cholesky matrices are updated as sonar data arrives using the incremental method described above. When predictions are required from the GP, the current covariance matrix of training points is taken, the new covariance for the test points with themselves, K_{22}, and with the training points, K_{12}, are calculated and then the incremental method is used to update the Cholesky matrix to solve the GP for $\mathbf{y}^*$. Depending on the amount of data that is collected, there may still be issues with the size of this matrix. Methods for controlling the size of the covariance through sparsification are discussed in [123] and the approximate Cholesky downdate can be used to efficiently achieve this.

The GP also requires determination of its HPs. As stated in Section 2.3, the analytical gradients calculated in Equations 2.7 - 2.10 are used to maximise the LML of the data given the parameters. This is run after an initialisation period to collect some data and then at regular intervals. It is run on a separate thread, and when it returns new HPs these are then used by the GP. It is important to note that these HPs are used to define the covariance matrix, and thus when they change, both the covariance matrix and the Cholesky matrix need to be fully recalculated, before performing any new incremental data updates or predictions.

3.3.2 Find and Follow the Intersection of a Depth Contour and a Boundary

The adaptive autonomy of the ASV is provided by the algorithm for finding the desired depth contour, defined by a target depth z_t, and following the intersection of this with the bounding polygon, which is detailed in Algorithm 1. The depth contour is set at the maximum of a safe operating depth and a minimum depth of interest for the study. For a point sensor only sensing directly downwards floating objects such as bouys, or very steep gradients such as vertical seafloor rises cannot be sensed. The vessel should be able to both operate safely due to the combination of the bounding polygon and minimum safe operating depth and obtain the sonar data required to create a bathymetric map of the area. The bounding polygon will keep the ASV both in an

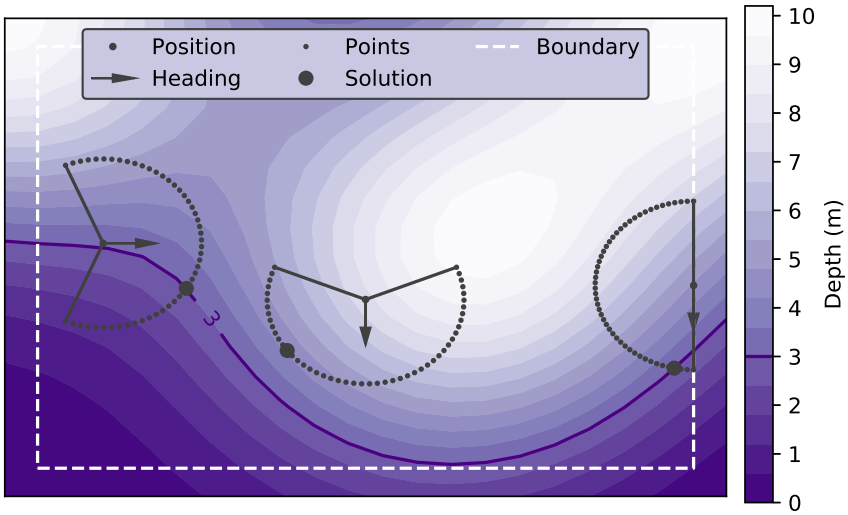


Figure 3.1: RoseSolve algorithm. Examples here show the heading resolved based on differing start points to either follow the minimum depth contour (left), head towards the minimum depth contour (middle) or leave boundary following to return to contour following (right) based on target depth of 3.

area of interest and away from obstacles unable to be predicted through modelling of the bathymetry. Inspired by Bug type algorithms such as DistBug [69], the algorithm follows the desired depth contour until it hits a boundary. Upon hitting the boundary it then follows this boundary until it finds the boundary taking it shallower than the target depth, at which point it leaves the boundary and again begins following the depth contour. This is continued until a circuit has been completed. A key difference in this algorithm to the bug algorithms is that it is not aiming for a single goal but always searching for a desired depth at a distance r, the search radius from the current position. This search is detailed in Algorithm 2. In addition, the surface upon which the search is conducted, the GP model of the bathymetry, is changing as data is obtained and both the HP's of the model re-estimated and the model re-fit. This model changes faster in the initial stages when there are small amounts of localised data. This can lead to the initial path turning back on itself. To counteract this noise, and prevent early determination of boundary closure, a parameter on the boundary completion test for loop closure is set to ignore the most recent *loopBuffer* points. The value of *loopBuffer* should be set relative to the expected length of the boundary.

The ASV can only sample depth directly downwards. To find the depth constrained boundary, from a given staring point not on this boundary, the ASV should follow

Algorithm 1 Find and follow contour within bounding polygon

```
procedure FFCB(poly, r, z_t, loopBuffer, ψ_adj)
    mode = Contour, polyDir = ∅
    repeat at 10Hz
        Obtain current GP and pose estimate x, y, ψ from separate threads
        if mode = Contour then
            ψ_s, ψ_e = ψ − ψ_adj, ψ + ψ_adj
            x_w, y_w = roseSolve(GP, z_t, ψ, x, y, r, ψ_s, ψ_e)
            if x_w, y_w is not in poly then                    ▷ Crossing boundary
                mode = Boundary
                if polyDir = ∅ then
                    Set polyDir to index direction along edge to deeper water
                Get vertex x_w, y_w in direction polyDir from edge of poly crossed
        else
            Get x_w, y_w from current vertex in direction polyDir
            if Distance to x_w, y_w < r then
                Get x_w, y_w from next edge in direction polyDir
            if Depth predicted at x_w, y_w < z_t then          ▷ Leave boundary
                mode = Contour
                Set ψ_s and ψ_e to the intersection of an arc of radius r centered at x, y
with poly
                x_w, y_w = roseSolve(GP, z_t, ψ, x, y, r, ψ_s, ψ_e)
        Calculate heading ψ_d to x_w, y_w from x, y
        if boundary or contour has been found then
            Append x, y to boundaryList
        Send ψ_d to heading controller
    until Boundary overlap greater than loopBuffer
return boundaryList
```

Algorithm 2 Solve for heading to target depth t_z

```
procedure roseSolve(GP, z_t, ψ, x, y, r, ψ_s, ψ_e)
    Set ψ_* as a linearly spaced vector from ψ_s to ψ_e with 50 increments
    Predict z_* at distance r from x, y for headings ψ_* from GP in a batch
    for each sequential pair of {z_*, ψ_*} points do
        Set {ψ̄, z̄} as linearly interpolated based the target depth z_t, bounded by the
two ψ_* headings
    Set ψ_d as the ψ̄ with the minimum abs(z̄ − z_t), with ties broken by minimum
abs(ψ̄ − ψ)
    Set x_w, y_w as the Cartesian coordinates of the ray projected from x, y at ψ_d for
r
return x_w, y_w
```

the contours of the bathymetric predictions from the GP to find the minimum depth contour and this follow the intersection of this contour with the imposed boundary. This is achieved by the RoseSolve() algorithm. It searches for the heading leading to the target depth on the circumference of an arc on the compass rose of radius r around the current position, as shown in Figure 3.1. Note as discus. Whilst in contour following mode, this arc is centred around the current heading and bounded by the start angle, ψ_s and end angle, ψ_e. This range, $\pm\psi_{adj}$ can be chosen to reduce the search space so computation is not wasted searching where the ASV has just come from. When exiting boundary following mode, ψ_s and ψ_e for the search are set based on the boundary of the polygon, such that the arc is inside the polygon. Iteratively querying from a GP is expensive, as discussed in Section 2.3. Thus rather than using an iterative root finding algorithm such as Muller's method, the arc is split evenly into a number of points, queried as a batch and linearly interpolated between the points with the closest to the target depth. The error from this linear interpolation can be managed by the number of points chosen and will quickly be smaller than errors in control, and is reset at each control loop. The best of these solutions is returned with ties broken based on the distance to the current heading. This behaviour can be seen in Figure 3.1, where the algorithm is solving for a desired depth of 3. For the position on the left and near the middle, an arc centered around the current heading is searched, whilst in the position on the right boundary it can be seen an arc within the boundary is searched. The depth for all the points on these arcs are queried from the GP, and then the best segment linearly interpolated to get the solution, indicated by the solid grey dot. For the positions on the left and right, the algorithm finds the heading which will lead to the desired depth, whereas the position in the middle returns the heading which is closest to the desired depth.

Originally this was implemented as a recursive bi-section search on the arc. The computational cost of repeatedly querying the GP for 1 prediction point method led to implementation of a batch procedure. The number of splits is a design parameter. With 50 splits, there is a prediction point no more than $1/40\pi$ radians apart, which with a search radius $r = 5$m equates to test point spacing of approximately 0.4m. The velocity and sampling frequency on the ASV resulted in spacing of the sampling of points of around 1m along the path of the vehicle. Thus the combination of this test

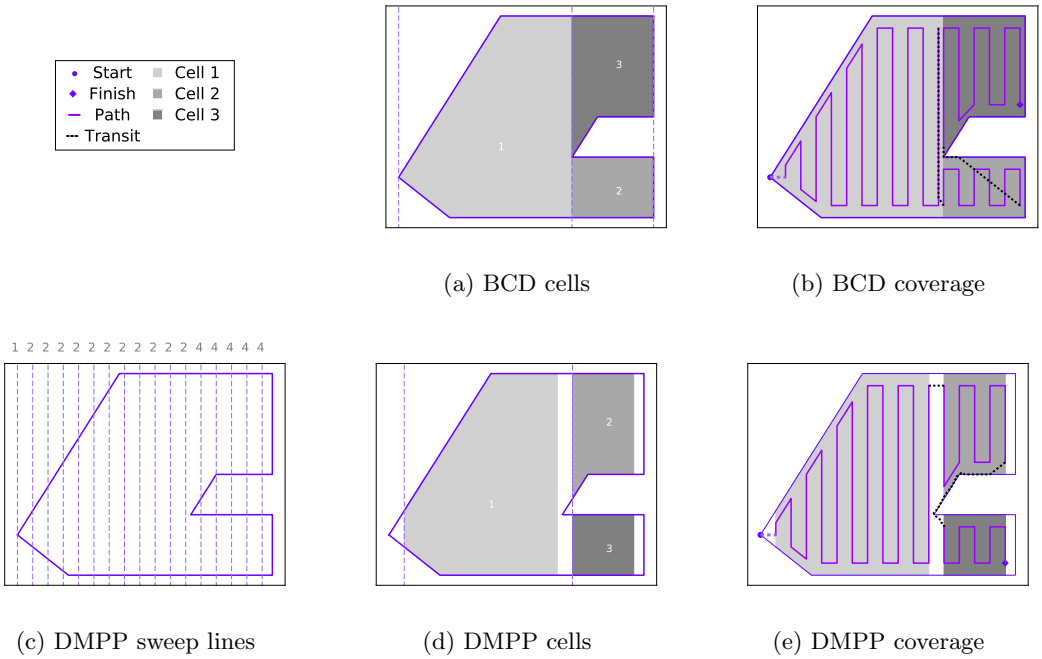
point spacing with linear interpolation, the smooth surface provided by the GP on this scale given the sampling scale, and the frequency on the control loop speed was found to be a reasonable compromise for computational load and precision.

A key tunable parameter in this algorithm is the search radius. A number of factors come into play in the choice of the search radius. The smaller this radius is, the smaller the area in the GP around the current position of the vessel is searched for the desired heading. A larger radius will result in a smoother path. This will also result in a larger tracking error between the desired depth found on this radius, and the depth sensed directly below the vessel. The expected rate of change of the bathymetry should be taken into account such that the search radius is set to allow the vessel to follow these changes. The speed of the vessel and the sampling rate of the sonar should also be taken into account. As will be shown in the simulated experiments, the algorithms are robust to sensible choices here.

3.3.3 Discrete Monotone Polygonal Partitioning and Path Generation

Upon completion of the intersection of the depth contour and the bounding polygon, a new intersection polygon is created whose boundary has already been sensed and which now needs to be planned within for coverage. For the purpose of this study the track width is a design parameter. Similar to the BCD, a method to create elemental polygons which are monotone to a given sweep direction is implemented. There are some differences which are have implemented to produce a more optimal path given the desired path spacing.

Under BCD, a sweep direction is chosen. A line orthogonal to this sweep direction is traced through the polygon. At any point where the number of crossings of this line with the polygon changes there is a connectivity event. These events are used to create the elemental polygons which are monotone to this sweep direction. This can be seen in Figure 3.2(a), where there are 3 changes in the connectivity count which is used to create the three elemental polygons (note this is using a horizontal sweep direction left to right). The union of these three polygons returns the original polygon. These polygons are then joined in an anti-clockwise order as shown by Figure 3.2(a), with



(a) BCD cells

(b) BCD coverage

(c) DMPP sweep lines

(d) DMPP cells

(e) DMPP coverage

Figure 3.2: Comparison of BCD (a,b) with DMPP (c,d,e) showing more efficient coverage with DMPP both in terms of number of vertical transects (BCD requiring 1 more in cells 2 and 3) and distance to join cells.

lawnmower paths then drawn within them as shown by Figure 3.2(b).

The first issue with this partitioning is that the width of these polygons is not necessarily a multiple of the track width. The result of this can be seen in Figure 3.2(b), where the last up track in cell 1 is close to the cell boundary. The first vertical tracks in cells 2 and 3 are then closer to this track than the desired spacing which results in longer total path length and irregular coverage. For the application of a point sensor this results in some sensed points being closer than required. In the application from [20] of perfect sensing of a fixed width scanner, this would result in a significant amount of scan overlap on the cell edges, again a waste of resources. The Discrete Monotone Polygonal Partitioning(DMPP) method has been developed to explicitly deal with this, which will be discussed after detailing the other shortcomings of the BCD method.

The second issue with the BCD method is that the transit paths joining the cells are not optimised. The order of joining is simply conducted through an anti-clockwise search of the neighbours of the current cell for the first cell that has not yet been covered, repeated until all cells are covered. It is not until all the transit paths are computed that the lawnmower path within the cell is computed. This is clearly sub-optimal. As

can be seen in Figure 3.2(b), lawnmower coverage of cell 1 ends in the top right hand corner. Clearly the best cell to go to would be the upper right cell. The BCD method is not aware of this and instructs a transit to the bottom right cell. Which corner of the cell to join to should also be dependent on which corner of the prior cell was exited from. The BCD method does not detail how the lawnmower paths within the cell should be constructed, which in any case are only fit after the order of cell joining is determined. This second issue is dealt with by calculating the lawnmower path within the closest cell, then solving an A* search [28] from the exit point of this cell to the nearest corner of an unexplored cell, and repeating until all cells are covered.

Finally the BCD is designed to handle polygonal objects within its boundary. The boundary is set up as a simple rectangle. Whilst there is nothing to stop their general approach being applied to more complex boundaries, they do not mention this case and how it would be approached. The DMPP algorithm explicitly deals with this case.

Algorithm 3 details the DMPP. The input parameters for this algorithm are the boundary polygon *poly*, the desired track width δ and the sweep direction ψ_{sd}. By constraining the sweep direction in the range $-\pi/2$ to $\pi/2$, without loss off generality, the polygon can be swept from left to right from the bottom left corner of *poly*. Lines 4-6 of Algorithm 3 detail this sweeping process of producing a list of crossings of a line orthogonal to the sweep direction with the polygon which is sorted from the bottom. This discrete stepping is the key difference to a standard polygon sweeping algorithm which would sweep in continuous space. This discrete list naturally handles cusp points. These only have an impact when they result in a change in the number of connectivity events on one of the discrete sweep lines. Multiple reversing cusp points occurring between two sweep lines are ignored, with only their resultant impact on the discrete sweep lines considered. An example of this can also be seen in Figure 3.2(c) where the sweep lines are shown, and their corresponding crossing count shown on the top of the Figure.

With this list of crossings, changes from one sweep line to the next are identified, indicating a connectivity change event. When this happens, open cells are closed in order from bottom to top, before new cells are opened, again ordered from bottom to top. The co-ordinates of the individual cell corners created are ordered clockwise

from the bottom left corner. These 4 points define the first and last track lines in the cell. The boundary is then followed between point 2 to 3 and point 4 to 1 (as the other two sides are the track lines), to fully define the elemental cell, completing the DMPP.

Algorithm 3 Discrete Monotone Polygonal Partitioning (DMPP)

```
procedure DMPP(poly, δ, ψ_d)
    Set angle of the line ψ_l to ψ_d + π/2
    Set x, y to the minimum values of poly in the x and y directions
    while x or y inside poly do
        Project ray from x, y at angle ψ_l, append sorted list of crossings to sweepCrossings
        Adjust x, y in direction ψ_d by δ
    for each list of crossing points in sweepCrossings do
        if there is a change in the number of crossing compared to the prior list then
            if there are open cells then
                for each pair of crossing points in the prior list of sweepCrossings do
                    Set crossing points as closing corners of associated open cell, append cell to cells
            for each pair of crossing points do
                Open a new cell, set crossing points as opening corners
    for each cell in cells do
        Add additional required points on the edges without track lines to trace the boundary poly
return cells
```

A path for coverage through the space is now generated. The cell corners returned from DMPP are shrunk towards the inside of the polygon by δ at a direction orthogonal to ψ_d. This gives the corners which are spaced at the desired distance away from the boundary already traced in Algorithm 1.

The transit path from the current position is then calculated, which is assumed to be on or inside the polygon, to the closest corner of this list of adjusted corners. This is done in 2 steps. Firstly the euclidean distance between all corners and the current position is calculated. The shortest path is then found, and checked to confirm it is inside the polygon. If so, the best solution has been found and a series of way points is generated spaced δ apart between the start and end points and set this as the transit path. This allows transit paths between adjoining cells to be quickly found before resorting to more computationally intensive searches. If it is not, a series of A* searches are conducted,

using the euclidean distance as a heuristic, from the current position to all the corners in the list of adjusted corners. The shortest length path is found and set as the transit path and appended to the lawnmower path. The cell transited to is popped from this list and the position of the entry corner is recorded.

Within this current cell a lawnmower path is created by ray tracing lines orthogonal to the sweep direction, at a spacing of δ, starting at the entry corner position. These resultant line segments are shrunk by δ from the cell boundary, waypoints created on them δ apart, and joined together. Once lawnmower path for the current cell is created, the transit path loop is again run to find the path to the next nearest cell and continued until all cells are covered. At which point the final path for coverage of the entire polygon is returned.

It can be proved with modular arithmetic that when planning coverage through DMPP vs. BCD, the combined path returned has less than or equal to the number of track lines orthogonal to the sweep direction. The following four terms are defined to aid in the proof.

$$N_{DMPP} = W_p \ div \ \delta$$

$$R_{DMPP} = \Sigma_{s_D} \left(W_{s_D} \ mod \ \delta\right)$$

$$N_{BCD} = \Sigma_{s_B} \left(W_{s_B} \ div \ \delta\right)$$

$$R_{BCD} = \Sigma_{s_B} \left(W_{s_B} \ mod \ \delta\right)$$

where N_{DMPP}, N_{BCD} are the number of tracklines orthogonal to the sweep direction due to the DMPP and BCD respectively,R_{DMPP}, R_{BCD} are the sums of the remainders for each method, W_p is the width of the polygon at its widest point in the given sweep direction, W_{s_D} and W_{s_B} are the width of the polygon of the current segment in direction ψ_d for the DMPP and BCD methods respectively, where the segement is defined by connectivity events in the relevant scheme and following [52] $A \ div \ B = \lfloor A/B \rfloor$, $A \ mod \ B = A - B\lfloor A/B \rfloor$ where $\lfloor . \rfloor$ is the floor function.

Since DMPP by definition creates cells which are a multiple of the track width δ,

$R_{DMPP} = 0$. BCD on the other hand does not consider δ when discretising the cells. Thus $R_{BCD} \geq 0$, and in any realistic scenario in the field $R_{BCD} > 0$. If $R_{BCD} \geq \delta$ then BCD will create additional track lines over and above DMPP. Thus:

$$N_{DMPP} \leq N_{BCD} \text{ with equality iff } R_{BCD} < \delta$$

There are then the additional inefficiencies in the BCD method which does not search for the most efficient way to join the cells and decides how to join the cells before knowing where the paths exit and enter them. For even the simple example shown in Figure 3.2 the BCD method results in total within cell path 6% longer and transit paths 98% longer for a total path which is 11% longer than the DMPP and path generation algorithm. An example for a more complex polygon can be seen in Figure 3.3. Unlike the BCD in which the union of the cells is the polygon, in the DMPP there are spaces between the cells. These are designed such that the track lines are exactly upon these edges and thus even coverage is achieved as desired. This has the additional advantage that joining adjacent elemental cells is not required as they will be covered efficiently. This can be seen in Figure 3.3 where the two cells on the bottom left could be combined into one cell, but it would make no difference in the planned path.

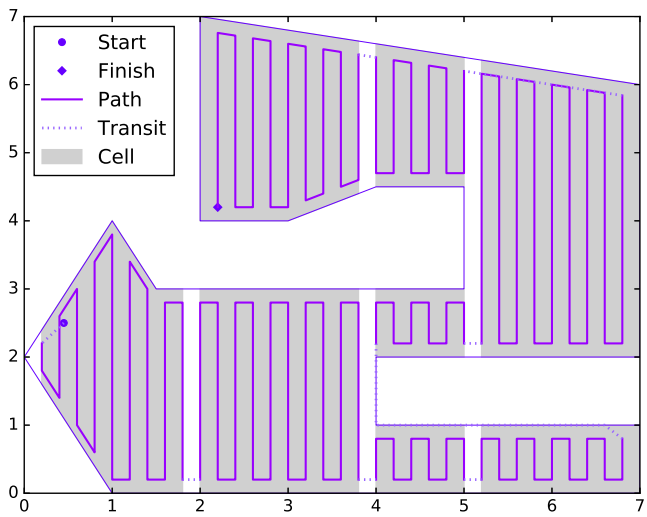


Figure 3.3: DMPP and path generation. A more complex example showing how separation between adjacent cell boundaries (white space) allows both perfect alignment of desired track spacing and how it handles additional splits without problems in track length spacing.

Whilst this is a greedy method in that it is only ever looking ahead one cell, it is superior

to the BCD, which naively takes the nearest anti-clockwise cell regardless of transit distance. The resultant path length could be improved at the expense of computation time by a deeper search, though an exhaustive search would quickly become infeasible as the number of cells grows due to the NP nature of the problem. The sweep direction could also be optimised, through running a number of potential rotations similar to the trapezoidal sweep optimisation in [110], though unlike their method, DMPP must choose the same track line orientation for all cells as they have been created monotone to the same sweep line orientation. Whilst the DMPP and path fitting algorithms have only been shown for polygons without holes, as this was the use case, the algorithms themselves can easily be adapted to this by recursively running the the boundary following and DMPP algorithms whenever a internal boundary is encountered whilst following the lawnmower path.

3.4 Simulation

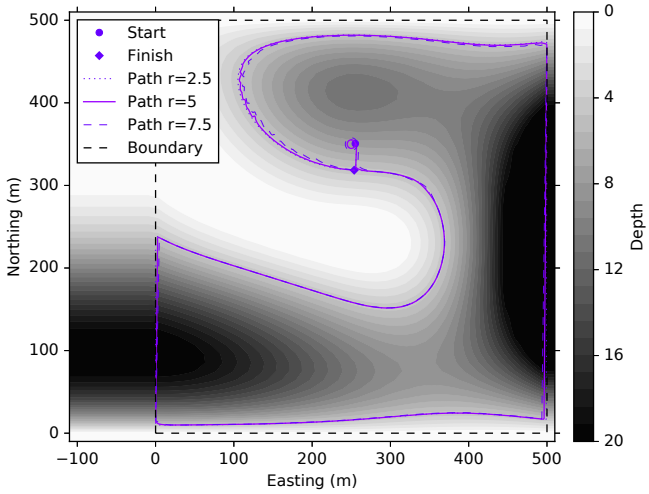
A simulated bathymetry has been created to demonstrate the algorithms presented in Sections 3.3.2 and 3.3.3. A vessel with perfect localisation, sensing and control is tested to focus on validating the performance of the coverage algorithm itself. The parameter settings can be seen in Table 3.1. From the start point the vehicle is driven in a circle of radius 5m for 50s to gain some initialisation points for the GP. After this initialisation, the HPs are estimated, and then again every 30s.

	Simulation	Field Trial
Velocity (m/s)	1.0	≈ 1.0
z_t (m)	4.5	4.0
r (m)	2.5, 5.0, 7.5	5.0
δ (m)	10.0	5.0
ψ_d (rad)	0.0	0.0
HP re-estimation interval (s)	30	30
IMU (Hz)	1	50
GPS (Hz)	1	1
Control Loop (Hz)	1Hz	5Hz

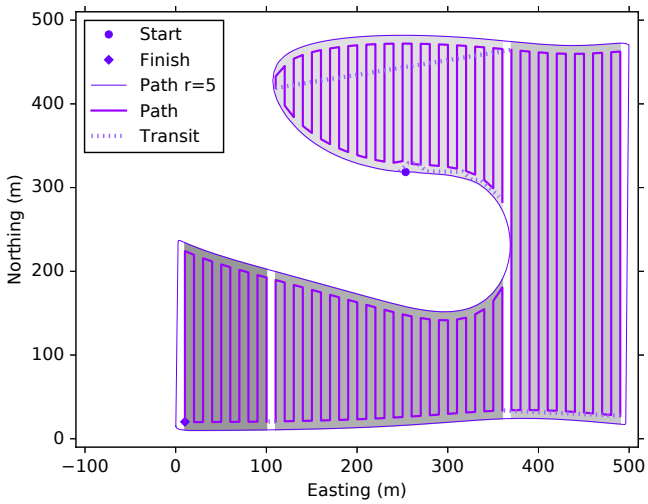
Table 3.1: Parameter settings for simulation and field trials

As can be seen in Figure 3.4(a) the vessel follows the contour gradient it has discovered by searching on its GP model of the bathymetry and follows south until it arrives at the target depth. It then turns east and follows this contour until it gets to the western

boundary at (0,237). At this point it follows the boundary south into deeper water until this boundary following would take it shallower than the target depth at which point it turns east again and follows the contour. After another boundary and contour following section it completes tracing the intersection of the boundary and the depth contour.



(a) Contour/boundary path and bathymetry



(b) Contour/boundary path and coverage

Figure 3.4: Simulated depth constrained bathymetric mapping. Contour found for 3 different values of r, indicating robustness to value chosen, overlaid on truth bathymetry in (a), coverage path for $r = 5$ boundary shown in (b).

The algorithm now moves on to partitioning this intersection and creating a path for coverage. Figure 3.4 (b) shows the result. The intersection is split into 4 cells. From the start point the closest cell corner, on the south west, cannot be transited to in a straight line as this would take it out of the polygon. An A* path is generated to transit to this point. A lawnmower path is then generated to cover this cell to the west. A

transit path from the end of this cell is then generated transit to the north west corner of the next cell, which can be done directly as this stays within the polygon. The lawn mower path is then generated for this cell and the process repeated for the final 2 cells until a plan for coverage of the entire space is generated.

The main design parameters for these algorithms are the target depth z_t, the path spacing δ and the search radius r. The target depth should be chosen based on a combination of the safe operating depth of the vehicle and the depths of interest for the study. The path spacing should be chosen based on the coverage density desired. The search radius impacts the operation of the algorithm in a number of ways. A larger search radius expands the search horizon, though if this is too large it may move the ASV away from where the model has certainty. As this search radius increases the ability of the vessel to smoothly follow tight turns in the contour is reduced, and a tracking error between what is directly under the vehicle compared to the depth at the planning horizon whilst following a curve is introduced. As such this parameter should be bounded from above based on an expectation of the minimum radius of curves in the contours it is following. On the lower range of this variable the planning horizon should be longer than the distance covered by the vessel between planning points (in the simulation case this is 1m due to a velocity of 1m/s and a control loop of 1Hz). Empirical testing has shown the solution to be robust to the choice of r as can be seen in Figure 3.4 (a) where setting r at 2.5m or 7.5m results in a very similar path to $r =$ 5m.

3.5 Field Tests

To demonstrate the robustness of these algorithms to the noise introduced from uncertainty in sensing and localisation from operating in the field and the computational limits imposed by an embedded CPU and real time operation, a small autonomous surface vessel was built. This can be seen in Figure 3.5. Details of the hardware and software are given in Appendix A.

The area chosen for this task is a secluded part of the Port Hacking river, to the south of Sydney, Australia, called Cabbage Tree Basin. This in an interesting area

Figure 3.5: The Autonomous Surface Vessel built and used for the field trials

for a number of reasons. The Port Hacking river was the first estuary in Australia closed to commercial fishing in the late 19th century. It is bordered on one side by a residential area, and on the other by the Royal National Park, which is the second oldest national park in the world (after Yellowstone), established in 1879. Cabbage Tree Basin itself is a significant area of heritage value with a long history of Aboriginal occupation. It was the site of the first marine hatchery in Australia in 1900 and is one of the earliest described estuarine wetland areas in Australia [157]. The long shallow entrance to the basin also significantly limits access to recreational boating traffic, enabling unobstructed operation of the ASV.

A bounding polygon 100-150m wide by 40m high was set, as can be seen by the white trapezoid in figure 3.6. This area encompassed depths from less than 50cm to 8m. The parameters used can be seen in Table 3.1. The mission starts at the North-East corner of the white trapezoid which can be seen in Figure 3.6. The vessel was manually driven south-west for approximately 10m until it reached a depth of 1m. It was then driven in an arc for 5s (achieving a quarter of a circle). The GP HPs are estimated from these initial points, and then again every 30s. The FFCB control loop, Algorithm 1, then started operation. Figure 3.7 shows some snapshots of the GP model, planning and path travelled during the mission. In each of the 6 pairs of figures the upper figure shows the path of the vessel overlaid, at a given point in time, on the bathymetry estimated by the model, whilst the lower figure shows the confidence of the model at this time point through the standard deviation (in log scale), with 1 data point per second. A video showing the evolution of the model as each new data point arrives

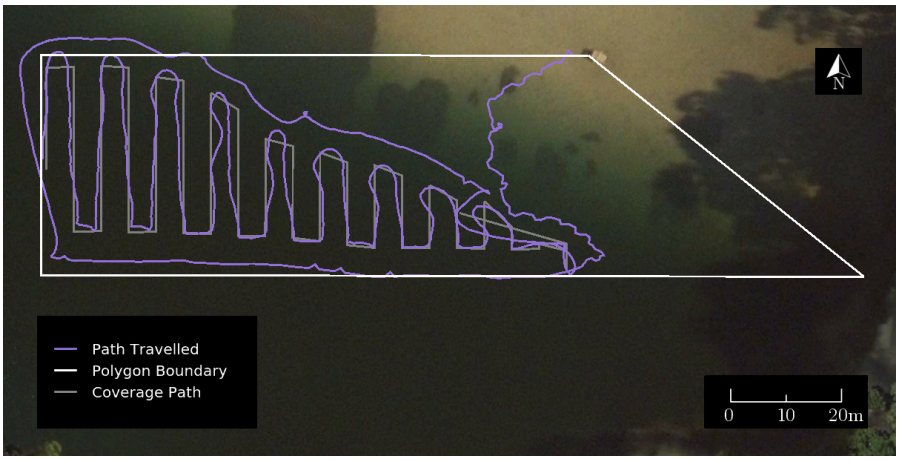


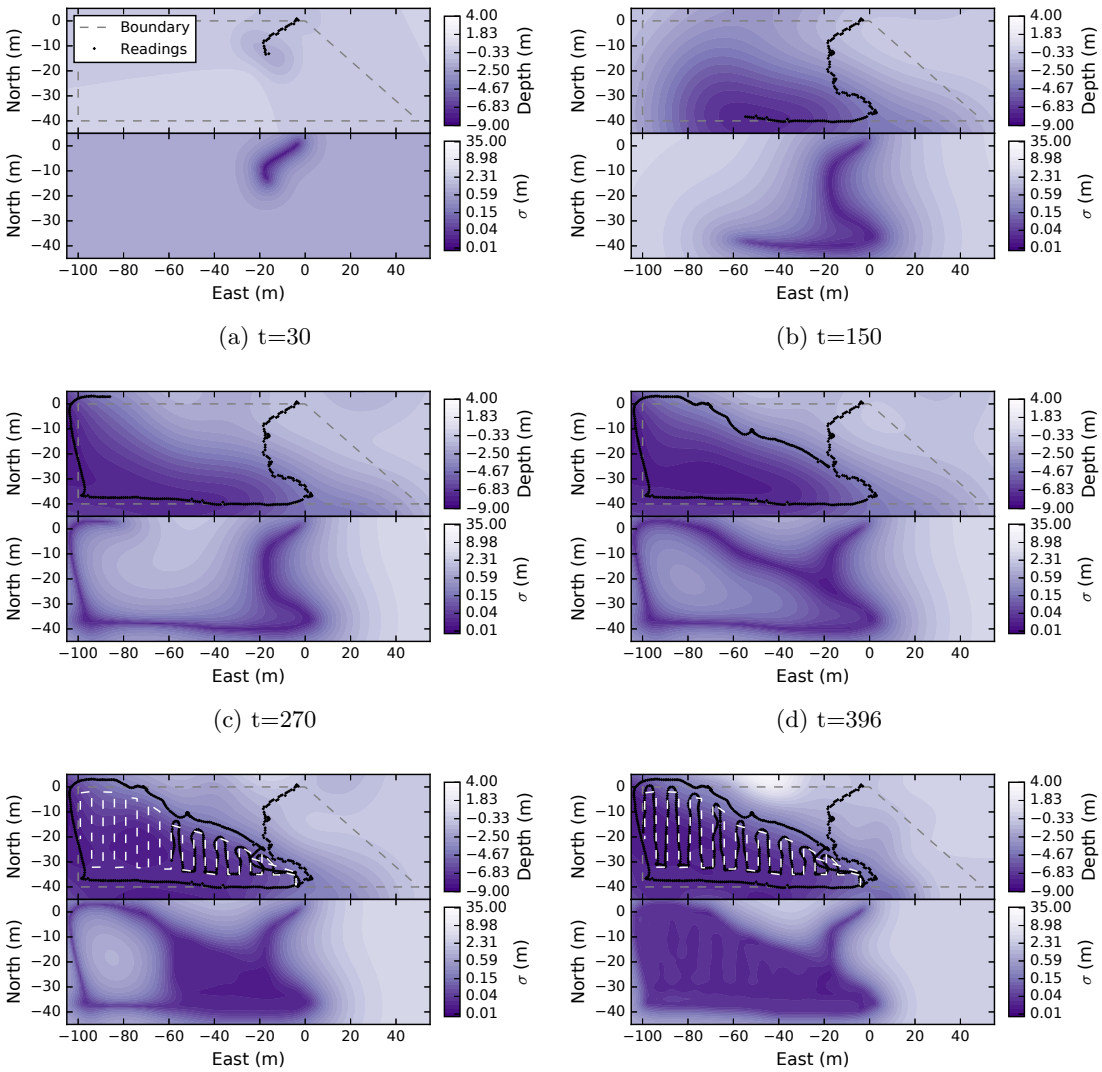
Figure 3.6: Boundary, path planned and contour followed overlaid on satellite image of survey area for field trial in Cabbage Tree Basin, Port Hacking Estuary, Sydney, Australia

can also be seen online [2] and a video of the ASV executing part of the mission is also available [3].

To increase the robustness of the algorithm, in the field trials an exponentially smoothed average of the current estimated heading was used, with a half life of 5s (equivalent to the time taken to cross the planning horizon). This was used in contour following mode to center the search space for the roseSolve() algorithm to smooth out the short term effects of any environmental forcing on the instantaneous heading.

Figure 3.7(a) shows the path and model immediately after the initialisation period and first HP estimation. At this point the algorithm does not have a very good model to work with when trying to follow the contour. Due to this the vessel traces a tight circle between t=30s and t=37s before continuing to follow the contour south until it hits the desired depth. It then follows this depth contour south east until it gets to the southern boundary. This boundary is followed along the bottom edge, up the western side and partially across the top until this would take it too shallow, as can be seen in Figure 3.7(c), at which point it switches back to contour following mode with the vessel then following the contour back toward where it first found the contour. This is achieved by t=396s. The intersection of the depth contour and boundary polygon is now used by the DMPP and path generation algorithm to plan a path. In this case the intersection is already a monotone polygon with respect to the sweep direction and thus only 1 cell

[2] `https://youtu.be/G88L7FATtKQ`
[3] `https://youtu.be/YH2nymgKXws`



(a) t=30

(b) t=150

(c) t=270

(d) t=396

(e) t=700

(f) t=1033

Figure 3.7: Scanned points recorded in field trial overlaid on online GP estimated depth contours (top) and standard deviations (bottom) at t=30,150,270,396,700 and 1033

was created and a lawnmower path was fit to it. The vessel then followed this path for coverage. As can be seen from the Figures 3.7(d) to 3.7(f), the standard deviation of the map within the intersection falls to around 3cm after the coverage task has been completed at point 1033.

The history of the HPs as they were optimised throughout the mission can be seen in Figure 3.8. The range of the noise standard deviation σ_n, whilst it looks large, is on a different scale and 2 orders of magnitude smaller than the model standard deviation σ_f, so is irrelevant in terms of its implications for the control algorithm. The model standard deviation, relatively quickly stabilizes around 3m to 4m by the 3rd point. The characteristic length scale estimate (l), slowly decreases as more data is

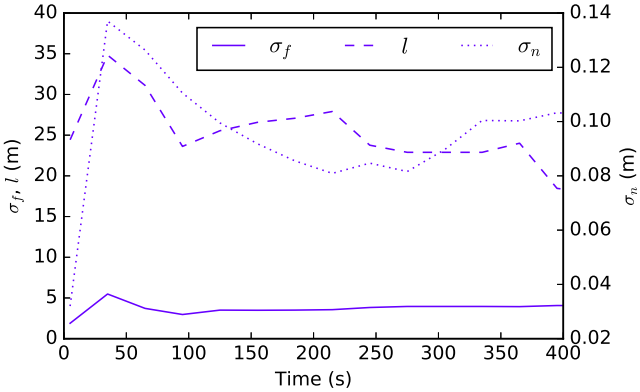


Figure 3.8: Iterative Hyper Parameter estimation from field trial

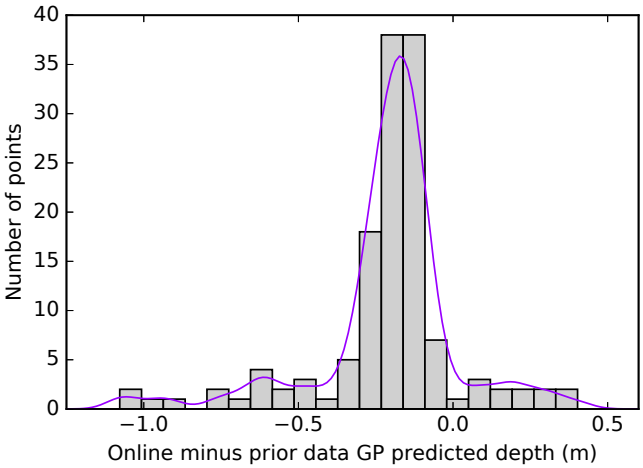


Figure 3.9: Histogram and estimated kernel density of differences between online predictions of depth and predictions from GP fit to prior data whilst contour following of waypoints generated by RoseSolve.

collected, though from the 4th point the estimate is in a range of 16.8 to 27.9m. The final estimated HPs in the contour following phase are $\sigma_f = 4.07$m, $l = 18.45$m and $\sigma_n = 0.103$m.

The level of the prior uncertainty and thus the asymptotic level that is approached as the ASV moves away from observed data points is controlled by σ_f, whilst σ_n is the estimated noise. Neither of these parameters impact the algorithm. Higher noise from sensors or errors in localisation would show up in an increased σ_n. The length scale parameter is the key parameter here as it estimates how far away from a test point to consider other points. If this were to be solved at a value significantly larger than the space sampling is conducted in, 1000 for instance, then a flat plane would essentially be fit through the data points. If it was much smaller then only be using a very local range of points would be used in prediction. If sampling in an area with rougher bathymetry,

then it could be expected that a smaller length scale would be required to model this. No serious issues with instability of the HPs in the field trial or simulation leading to problems in contour following were experienced. This would be most likely to occur at the start of the mission with a small number of measurements. If this were to become an issue, potential solutions would be a longer initialisation period, or to take a more Bayesian approach and impose prior distributions on the HPs. It can be seen that with the changes in the HPs throughout the mission, the algorithm showed its ability to follow an estimated contour, which appears reasonable when compared to the final estimate of the bathymetry as can be seen in Figure 3.7(f).

To estimate the accuracy of the online contour following algorithms a GP was fit to 3731 data points collected in the area on a number of missions in the 2 hours prior to the mission shown. Post mission at each point where a sonar reading was taken, heading that would have been generated by the RoseSolve algorithm given the data and HPs that were in place at that time is calculated. For each of these headings, the depth at r is predicted from both the online model and the model generated from prior data. A histogram of these points overlaid with the estimated kernel density (using a Gaussian kernel with a bandwidth of 0.05) is shown in Figure 3.9. The mean of these errors is -0.21m and the standard deviation is 0.23m. This mean bias is likely due to the fact that the tide was receding and the tidal range that day was 0.74m. The standard deviation of 0.23m is reasonable given the estimated noise on the prior model was 0.18m. The largest errors occurred in the north west corner which had the least coverage in prior data as whilst all missions were conducted in the same boundary, the final mission pushed slightly outside of that boundary due to increasing winds. Thus these differences may be due to errors in the prior model rather than the online model. From these results it can be seen that estimating the HPs and building the GP online produced a model which was comparable to one produced offline with a larger set of data. Thus the ability to learn these models and parameters online in unknown environments with little prior knowledge has been demonstrated in this domain.

On the day of the test there was a moderate breeze from the south east. This resulted in the vessel straying slightly from the boundary path whilst in boundary following mode, as can be seen in Figure 3.6, where the path travelled is slightly inside the

polygon boundary on the south west corner and slightly outside on the north west corner. Superior tracking performance in the presence of external forces such as winds and currents could likely be achieved with a more sophisticated controller than the proportional controller used on the ASV. Another problem occasionally encountered in control of the vessel was the presence of floating seaweed becoming tangled in the propellers, in the worst cases requiring a restart of the mission. Whilst the propellers do have a cage around them for safety, a finer mesh could perhaps help with deflecting the seaweed before it becomes entangled around the prop shaft.

The choice of a relatively short range greedy search to decide on the path for the vessel to take is seen to be justified by the relatively local nature of the information available. Given the ASV is only sensing directly downwards, there is a limit to how far the model can confidently project this information. This can be seen by comparing the sequential plots in Figure 3.7. In Figure 3.7(b) the model believes that the depth starts to rise about 7m away from the current position, whereas once this area has been sampled, which can be seen in Figure 3.7(c), it can be seen that it actually stayed constant. If the ASV had tried to make a plan to follow the whole contour from the data and model it had at the time of Figure 3.7(b), it would not have achieved the result seen in Figure 3.7(c) by following the contour in a local region.

3.6 Summary

In this Chapter a suite of algorithms for autonomously finding and following the intersection of a bathymetric contour and a bounding polygon, and then fitting a path for coverage within this boundary for the purpose of producing a map of the bathymetry in an unknown area was presented. A new algorithm for the partitioning of complex polygonal workspaces and the planning of coverage paths within them was developed, which is more efficient than the BCD method, and explicitly handles the complex boundary shapes. Computationally efficient methods for the updating of Cholesky matrices used by the GP were implemented to allow online fitting and prediction of a bathymetric map, including online optimisation of the HPs of the GP. Code for all algorithms and the implementation on the ASV are provided on GitHub [4].

[4] `https://zenodo.org/record/47963`

A small autonomous surface vehicle was developed which can test this and other robotics algorithms and conduct surveys in estuarine water ways. These algorithms were tesed in simulation and in the field and through these experiments the robustness of the platform and algorithms to uncertainty introduced by sensor noise and environmental forcing in a dynamic environment combined with the ability to run in real time on a small embedded system has been demonstrated.

Chapter 4

Active Sample Selection with Parametric Heteroscedastic Gaussian Process Regression

4.1 Introduction

Modelling the structure of random processes is important across a range of disciplines. In robotics randomness is introduced both through the environments operated within, as well as through the sensors used to perceive them and the actuators used to interact with them [3, 142, 148]. If sampling strategies can be adaptively planned from models built online as data is collected, these sampling strategies can be designed to create the best possible model of the phenomenon of interest with limited resources. As such there are two related problems. A model must be selected which can capture the attributes of the data of interest. It must then be decided how to sample to improve this model.

The framework of Gaussian Process (GP) regression allows a non-parametric approach to this modelling. A covariance function, or kernel, is defined which allows the projection of outputs of both the expectation and uncertainty across the space of potential inputs based on sampled inputs. Whilst this kernel has parameters itself, called Hyper Parameters (HPs), these can be learned in a principled manner from the data. The

standard model for GPs assumes uniform level of noise across the input sapce (Homoscedasticity). There are often fundamental reasons why the process generating the data of interest has noise which is dependent upon the input dimension [36, 38, 84, 131]. Ignoring this dependence can result in models which are incorrect in their variance predictions across the input space.

The contributions of this Chapter are as follows. Firstly the fact that prediction noise should be taken into account in the standard GP model even in the case of Homoscedastic noise is presented. A parametric model for Heteroscedastic noise which allows analytical gradients of the Log Marginal Likelihood (LML) with respect to the parameters to be derived facilitating simultaneous optimisation of the kernel and noise HPs in a single GP model is then described. The fact that using Homoscedastic models with both stationary and non-stationary kernels in the presence of Heteroscedastic noise results in predicted variance which has large errors is shown. Two methods to quantitatively analyse this are employed. A novel application of Equality of Variance tests on the normalised errors is implemented and a new measure called the Root Mean Square Standard Deviation Error (RMSSDE) is proposed. Finally it is shown that under PHGP adaptive sample selection based on Mutual Information outperforms other methods in the predictive power of the models it produces. Content from this chapter has been presented as a conference paper [1].

The remainder of this Chapter is organised as follows. Section 4.3 discusses model noise and predictive variance in GPs, details Heteroscedastic noise and introduces the PHGP. Section 4.2 presents related work, reviewing literature on heteroscedastic noise and adaptive sampling. Section 4.4 introduces the simulated environment for arrested bathymetric fronts and Section 4.5 tests a number of GP models for their ability to fit this data. Section 4.6 discusses measures which can be used to drive active sampling decisions. Section 4.7 provides convergence results for PHGP across these methods and Section 4.8 summarises.

[1] T. Wilson and S. B. Williams, "Active sample selection in scalar fields exhibiting non-stationary noise with parametric heteroscedastic gaussian process regression," in *International Conference on Robotics and Automation*, IEEE, pp. 6455-6462, 2017.

4.2 Related Work

Heteroscedastic noise processes are widely used in the field of Econometrics with the Autoregrssive Conditional Heteroscedasticity (ARCH) introduced in [38] and its many subsequent variants. It has been used to model the relation between pixel intensity and noise for the purpose of differentiating the cameras used to take the images [145] and to model the relationship between noise and accumulative irradiation in solar cells [138]. The importance of testing and correcting for Heteroscedasticity in ecological spatial data has been demonstrated with the abundance of Oribata, or moss mites, exhibiting spatial Heteroscedasticity [36]. Heteroscedasicity has been found in modelling benthic macroalgal biomass [84], and is also commonly found in water quality and salinity data [131], which is the motivating application for this Chapter.

Heteroscedastic noise processes were first used in GPs in [49]. The authors modelled the noise process with a second GP, and used Markov Chain Monte Carlo (MCMC) to jointly integrate out the HPs of the model. This fully Bayesian process is quite computationally demanding. A second GP is similarly used to model the noise process in [72]. In an Expectations Maximisiation type process, it iterates between estimating the HPs of the main and noise models until convergence. This can lead to oscillations and [81] instead propose a variational approach, Variational Heteroscedastic Gaussian Process (VHGP) regression where they maximise an analytical lower bound on the maximum likelihood of the parameters given the data at each step. This method is still quite computationally demanding.

A parametric model is proposed for the noise process and include it in a single GP model, Parametric Heteroscedastic Gaussian Process (PHGP). The gradients of the likelihood function w.r.t to the noise parameters in addition to the standard HPs are derived and thus all the parameters of this single GP model are estimated. This is compared to two Homoscedastic GP models as well as VHGP. The performance on a simulated dataset with input dependent noise is examined. All four models show comparable performance in fitting the mean, whilst only the two Heteroscedastic models correctly model the variance. Whilst the PHGP and VHGP model are comparable in predictive performance, the PHGP model does so with significantly less computational resources.

Within the PHGP model an appropriate adaptive sampling strategy must be decided. Information measures such as Entropy and Mutual Information(MI) are often used to drive sampling decision as discussed in [73]. Bayesian Optimisation (BO) based on Upper Confidence Bounds are used by [89] to drive sampling to areas which are more likely to posses extreme values, which can be of interest in their use case of pollutant monitoring. BO is compared to sampling based on mean and confidence bound forecasts in [17]. Fisher Information (FI) can be used to drive sampling decisions by producing estimates of HP uncertainty. This is applied in [162] on a Homoscedastic Gaussian process. FI driven sampling is compared to random sampling in terms of convergence to the known HPs generating the simulation. They do not compare to other Information theoretic measures, or on the basis of the predictive power of the model.

4.3 Gaussian Processes

The standard model for GPs is discussed in Section 2.3, along with a common stationary kernel, the Squared Exponential Kernel. Relationships between observations which change as the input space is traversed can be modelled with a non-stationary kernel, examples of which can be found in [127].

Augmented Variables have been proposed to introduce non-stationarity into a kernel in a simple and efficient way [116]. An extra dimension is added to the input space that is a function of the other input dimensions which induces non-stationarity in the original dimensions in the model. This is applied with a continuous latent variable in [64, 146].

There are two important points to note about this standard model. Firstly, the variance predicted by Equation (2.4) is the noise free prediction. To estimate the expected uncertainty of sampling at any given point in the sample space, the following should be used [154]:

$$\hat{\mathbf{\Sigma}}_* = K_{**} - K_*(K + \sigma_n^2 I)^{-1} K_*^T + \sigma_n^2 I \tag{4.1}$$

The second point to note is the form of the noise term which is added to the diagonal

of the covariance matrix, $\sigma_n^2 I$. The assumption of a constant noise variance makes this a Homoscedastic regression. In the same way that the relationship between test locations can vary throughout the input space, thus requiring non-stationary covariance functions, the noise in measurements can also vary throughout the input space. This case is called Heteroscedastic regression. The general issue with Heteroscedastic regression is that the analytical solution to the marginal likelihood is no longer simple and thus estimation of the HPs of this noise variance becomes difficult.

Taking these two points into account the model becomes:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K+R & K_* \\ K_*^T & K_{**}+R_* \end{bmatrix}\right) \tag{4.2}$$

$$\hat{\mathbf{y}}_* = K_*(K+R)^{-1}\mathbf{y} \tag{4.3}$$

$$\hat{\mathbf{\Sigma}}_* = K_{**} - K_*(K+R)^{-1}K_*^T + R_* \tag{4.4}$$

where $R = R(X)$ and $R_* = R(X_*)$ are the input location dependent noise functions for the observations and predictions.

4.3.1 Heteroscedastic Noise

If the environment is characterised by noise which is independent of input location then driving planning through Equations (4.1) or (4.4) will give the same results. If the environment is characterised by Heteroscedastic noise, then the results will be different.

The first attempt at solving this problem in GPs used a second GP to model the noise variance (or more specifically its logarithm, to ensure positiveness) and Markov Chain Monte Carlo(MCMC) in a fully Bayesian approach to integrate out the this noise variance along with the other parameters of the kernel [49]. A less optimal but potentially faster technique is to initially model the GP as homoscedastic, then fit a GP to the errors to model the noise. This is then fed back as the noise model into a heteroscedastic GP. The errors are again modelled and this process is iterated until it converges to a process where errors are stationary with respect to the input

space [72]. This method can be susceptible to oscillations. A variational approach has been proposed to solve this, whereby an analytically tractable lower bound on the exact marginal likelihood of the parameters given the data is maximised [81]. All of these methods are significantly slower than a Homoscedastic GP due to the need to run multiple GP models and expensive techniques to estimate the parameters of these models jointly or iteratively.

A GP can be used to model the non-stationary noise R. In this case:

$$R(X) \sim \mathcal{N}(0, e^{g(x)}) \tag{4.5}$$

$$g(X) \sim GP(\mu_0, K_g(X, X)) \tag{4.6}$$

where an exponential function is used to ensure positivity of the variance.

The use of a GP to model non-stationary noise leads to a marginal log likelihood of the data given the parameters that is not analytically tractable. The VHGP model [81] provides an approximation to this model that is analytically tractable. In their model R is a diagonal matrix with $[R]_{ii} = e^{[\mu]_i - [V]_{ii}/2}$, $\mu = K_g(\Lambda - \frac{1}{2}I)1 + \mu_0 1$, $V = (K_g^{-1} + \Lambda)^{-1}$ and Λ is a diagonal weighting matrix optimised in a conjugate gradient descent procedure. This leads to:

$$\hat{\mathbf{\Sigma}}_* = K_{**} - K_* K_y^{-1} K_*^T + e^{\mu_* + V_*^2/2} \tag{4.7}$$

where $\mu_* = K_{g*}(\Lambda - \frac{1}{2}I)1 + \mu_0 1$ and $V_* = K_g** - K_{g*}(K_g + \Lambda^{-1})^{-1} K_{g*}^T$

This model has HPs for the 2 kernels K and K_g. It also has the diagonal matrix of the variational weights Λ which has 1 parameter for each data point. Thus in the simplest case where the squared exponential kernel with a single isotropic length scale l_x is used for both kernels leading to $6 + n$ HPs. As these variational parameters are tied to the observation points, there is no easy way to incrementally add data. Whenever new data is added to the model, the variational parameters must be optimised again.

The use of some prior knowledge of the process generating the distribution can be used to define a simpler and more computationally efficient parametric model for the noise

process. The following is proposed:

$$R = \sigma_{n1}^2 I + \sigma_{n2}^2 \text{diag}(D(X)) \tag{4.8}$$

where σ_{n1}^2 is a constant noise applied across the input space, σ_{n2}^2 is a scaling factor on the Heteroscedastic noise and diag$(D(X))$ is a square matrix with the elements of the vector $D(X)$ on the diagonal. $D(X)$ is a parametric function to represent the heteroscedastic noise.

In a general setting, the belief that there is a source of noise somewhere in the input space could be modelled by a Gaussian distribution which diffuses the noise around this point. This would be represented as the standard multi-variate Gaussian cumulative density function:

$$D(X) = \sqrt{det(2\pi\Sigma)} exp\left(-\frac{1}{2}(X - P(X))^T \Sigma^{-1} (X - P(X))\right) \tag{4.9}$$

where Σ is the covariance matrix of the Gaussian distribution, and $P(X)$ is the mean function specifying the location of the center of the noise across the input dimensions. Multiple noise point sources could be modelled by a mixture of Gaussians.

In a one dimensional input space where X is the vector $\mathbf{x}$, this collapses to:

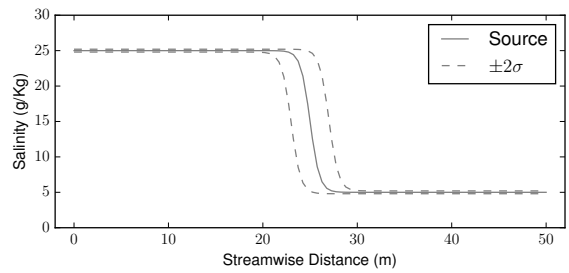
$$D(\mathbf{x}) = \frac{1}{l_d\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{\mathbf{x} - \mu_d}{l_d}\right)^2\right) \tag{4.10}$$

where μ_d and l_d are the mean and standard deviation of the Gaussian probability density function.

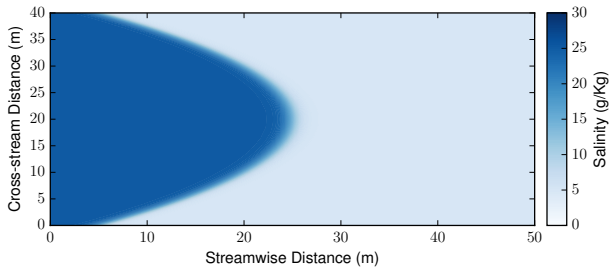
This allows for efficient estimation of the HPs through gradient descent as the gradients of the small number of HPs with respect to the LML can be analytically derived.

4.4 Modelling a Salinity Gradient

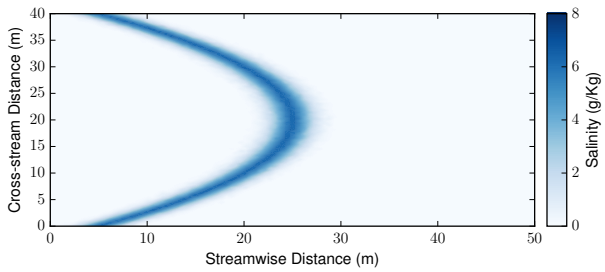
The motivating problem for this chapter was the modelling of the surface of a salinity front as it progresses up an estuary with the tide. Where this body of dense salt



(a) Streamwise Distribution



(b) Mean



(c) Standard Deviation

Figure 4.1: Simulated surface salinity. 1D distribution shown in (a), 2D distribution shown in (b) and (c)

water meets a less dense body of fresh water, a turbulent mixing process ensues as the salt water flows under the fresh water. There is a mixing layer at the interface and a turbulent core rotates within the fresh overflow near the front [96]. This physical process will result in higher noise around the mixing front than on either side where the salinity and density of the water bodies is more homogenous as can be seen by the velocity and density isocontours in simulations of gravity-current fronts [56]. Two simulated environments are created.

Firstly, one input dimension is modelled, representing surface salinity streamwise with the direction of flow in an estuary. This represents a stationary arrested bathymetric estuarine front, such as studied in [66]. The mean of this mixing process is modelled with a deterministic logistic function to which Gaussian noise is added with a standard deviation of 0.1 evenly across the domain. The use of the logistic function is motivated

by studies of the salinity change across a river plume in [109]. To approximate the dependency of noise with distance from the front, Gaussian noise with a standard deviation of 1 is added to the location of the front which when transformed through the logistic function results in the source distribution shown in Figure 4.1(a).

Secondly, the surface salinity is modelled in two spatial dimensions. The same distribution of the salinity across the tidal front as in the first model is used. The cross-stream location of the front is modelled by a 2nd order polynomial curve. This is motivated by the Class I bathymetrically arrested fronts as discussed in Section 2.1.1 where incoming tidal dense saline flows plunge under the lighter fresh or brackish estuarine water resulting in a surface manifestation of the mixing front which follows the shape of the bathymetry. A time lapse video of this phenomenon at Lilli Pilli Point, on the Port Hacking River in Sydney, Australia can be seen online [2]. This results in the following weighting function for the Heteroscedastic noise:

$$D(\mathbf{x_1}, \mathbf{x_2}) = \frac{1}{l_d\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{\mathbf{x_1} - \mathbf{p}}{l_d}\right)^2\right) \tag{4.11}$$

$$\mathbf{p} = p_0 + p_1\mathbf{x_2} + p_2{\mathbf{x_2}}^2 \tag{4.12}$$

where $\mathbf{p}$ is the location of the midpoint of the logistic function in the $\mathbf{x_1}$ dimension associated with the values contained in $\mathbf{x_2}$ and p_0, p_1, p_2 are the polynomial factors up to degree 2. The streamwise cross section of this curve can be seen in Figure 4.1(a) and across the 2D surface in Figure 4.1(b) and (c). The Squared Exponential covariance function from Equation 2.5, with an isotropic length scale (l) is used. The derivatives of the LML w.r.t the HPs can be analytically derived and are presented in Appendix B.2.1.

Any other parametric expression could also be used to create the weightings in D. The key insight here is that by imposing a parametric prior on the noise function, the time required to estimate the HPs can be significantly reduced, and thus also the computational and memory requirements in fitting and predicting the GP.

[2] `https://youtu.be/id1YWLujGX8`

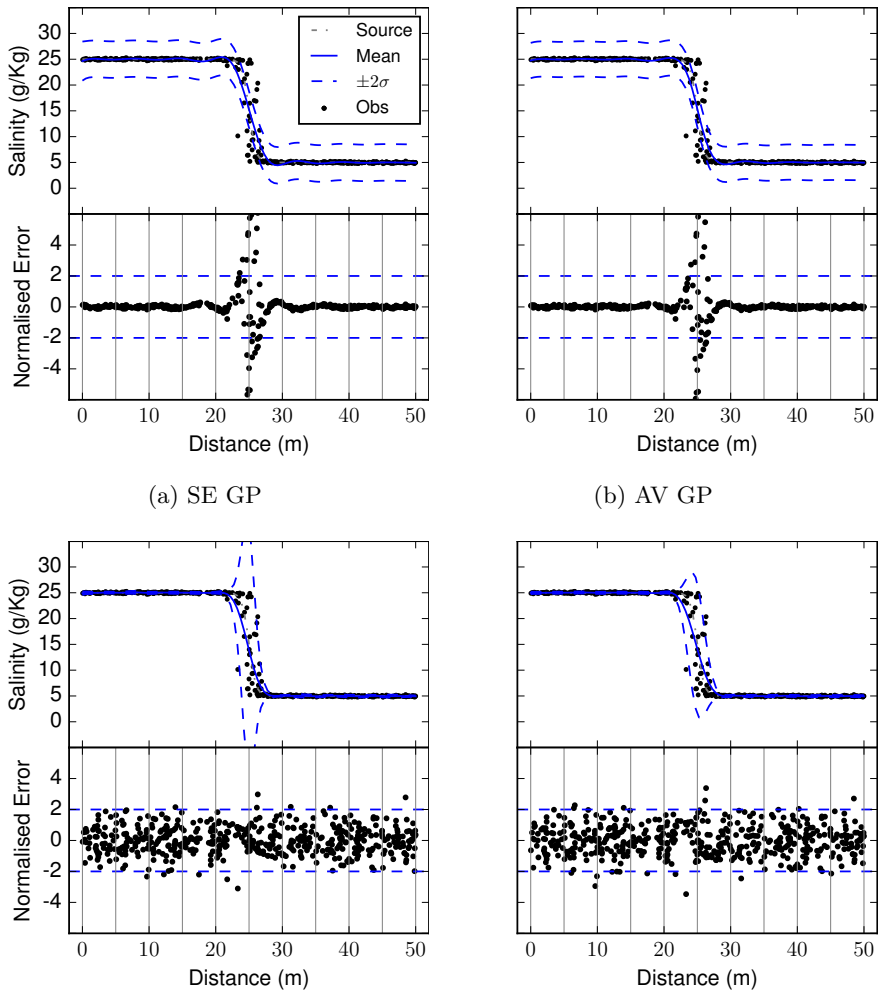
	1D	2D
Initial Obs	500	10
Planning Obs		100
Test Obs	100	1000
MI $\bar{X}$ obs		2500
Simulations	500	100
σ_{n1}	0.1	0.1
σ_{loc}	1	1
Input Space	[0,50]	[0,0,50,40]
Front Location	[25]	[[0,5],[25,20],[50,5]]

Table 4.1: Parameters used in one and two dimensional simulations of bathymetrically arrested tidal fronts.

4.5 Evaluation of Model Fit

The one dimensional simulation is used to compare the models in terms of quality of fit and computational time required for a batch of data. The simulation parameters used here and in the later two dimensional simulations are listed in Table 4.1. A test dataset is created by randomly drawing 500 points in the domain of [0,50] and passing them through the first simulated environment above. Four models are tested. The HPs of all 4 models are estimated by maximising the likelihood of the data given the parameters. The first model is a simple GP with a squared exponential kernel (SE GP). Second is the Augmented Variable model (AV GP) of [116] with an isotropic squared exponential kernel. The augmented variable is the predicted level from a simple squared exponential GP. The third model is the Variational Heteroscedastic Gaussian Process (VH GP) of [81] with a squared exponential kernel used for both the main and noise GPs. The final model is the Parametric Heteroscedastic GP (PH GP) which uses a squared exponential kernel and Gaussian weighted noise as described in Equations (4.8) and (4.10).

The mean predictions from all four models are similar, although the Homoscedastic models SE GP and AV GP perform slightly better as can be seen from Figure 4.2 and the Root Mean Square Error (RMSE) results in Table 4.2. The models differ largely in their ability to model the input dependent noise. The Homoscedastic models can only fit one noise profile across the whole space, as can be seen in Figure 4.2(a) and 4.2(b), and thus overestimate the noise on the side and underestimate it around the center. In comparison the Heteroscedastic models fit a varying noise profile across the input space as can be seen in Figure 4.2(c) and 4.2(d). It is important to note that



(a) SE GP

(b) AV GP

(c) VH GP

(d) PH GP

Figure 4.2: Model fit (top), normalised errors (bottom) on 500 randomly generated points from the source distribution in Figure 4.1(a) illustrating Homoscedastic models (a) SE GP and (b) AV GP underestimate the variance in the center and overestimate it on either side compared to the Heteroscedastic models (c) VH GP and (d) PH GP

even though all models in this case perform similarly on the mean prediction, the errors for the Homoscedastic model is significant. It will give a false sense of security in its predictions around the front, and a lack of confidence in its predictions away from the front. Decisions often need to be made from models based up on safety margins away from mean predictions. As discussed in Section 2.1.1, marine organisms have threshold levels to various water quality measures and thus knowledge of the distribution rather than just the mean levels is important. Other examples include civil infrastructure where knowledge of the distribution of potential values allows structures to be build to withstand extremes. There is no use in a dam which only holds up to an average rainfall event.

	RMSE	RMSSDE	Levene Test
SE GP	1.712	1.803	24.86
AV GP	1.700	1.758	21.70
VH GP	1.740	0.963	0.36*
PH GP	1.735	0.175	0.85*

Table 4.2: RMSE, RMSSDE and equality of Variance tests on normalised errors predicted from homoscedastic and heteroscedastic models on simulated sampled from one dimensional source distribution in Figure 4.1. Statistically significant results at $\alpha = 0.01$ level indicated by *

As the source distribution is known in this case, Monte Carlo simulation is used to estimate the distribution of the standard deviation across the input space. This distribution is sampled 1000 times at each of $n = 100$ evenly spaced locations across the input domain and estimate the empirical standard deviation $\hat{\sigma_s}(i)$ at each point. A new measure called the Root Mean Squared Standard Deviation Error (RMSSDE) is defined to obtain an estimate of the error in the predicted standard deviation across the input space as follows:

$$RMSSDE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\sigma_s}(i) - \hat{\sigma_p}(i)\right)^2} \tag{4.13}$$

where $\hat{\sigma_p}(i)$ is the standard deviation estimate predicted by the GP calculated at the ith input location. It can be seen from Table 4.2 that the RMSSDE for the first 2 models is relatively similar at around 1.75-1.8, whereas for the Heteroscedastic noise models it is much lower at 0.963 and 0.175 indicating these models provide a much closer fit of the second moment of the source distribution across the input space.

If the source distribution was not available, there are still tests which can be conducted to examine the fit of the second moment of the data. The sample dimension is very densely covered in this example relative to the estimated characteristic length scales of the kernel. As such the variance predicted by the model is dominated by the noise variance. A standard GP model would expect the errors between the observations and the mean predictions to be white noise. In a Heteroscedastic model it is not expected these errors will be white noise. A quantitative test that could be used even if the source distribution of the data is unkown would be useful.

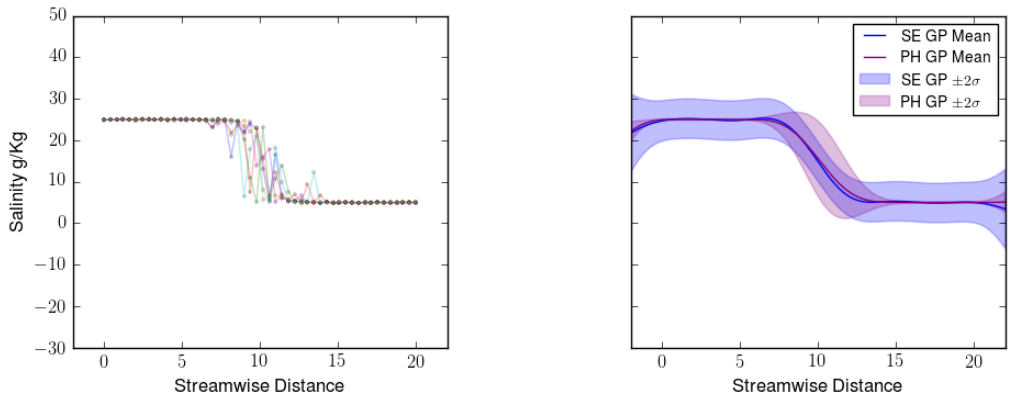
Standard gaussian process regression assumes independently and identically distributed

	50	100	250	500
SE GP	0.02 (0.006)	0.07 (0.014)	0.24 (0.055)	1.39 (0.185)
AV GP	0.05 (0.019)	0.17 (0.046)	0.89 (0.186)	5.16 (0.760)
PH GP	0.24 (0.092)	0.53 (0.167)	1.51 (0.443)	4.75 (1.336)
VH GP	1.39 (0.577)	3.26 (1.820)	23.7 (13.67)	138 (89.03)

Table 4.3: Average time (standard deviation) in seconds for SE GP, AV GP, PH GP and VH GP models to optimise HPs and calculate their kernels for data sets of size 50, 100, 250 and 500 points from 100 simulations

errors which are distributed $\mathcal{N}(0, \Sigma)$. If the errors are divided by the predicted standard deviation of the model at each point, these normalised errors should be distributed $\mathcal{N}(0, 1)$. This data can now be split into bins spatially, and a test for equality of the variance of these normalised errors between the bins applied. It can be seen that these normalised errors appear to be much closer to white noise for the Heteroscedatic models in Figure 4.2(c) and 4.2(d) than the Homoscedastic models in Figure 4.2(a) and 4.2(b). The Levene test [14] compares the null hypothesis of equality of variance across all the bins. In this case the resulting statistic is compared to a standard F table with k-1 and N-k degrees of freedom. The simulated data is divided into 10 bins. The results of the Levene can be seen in Table 4.2 where only the Heteroscedastic noise models match the variance of the data which failing to reject the Null hypothesis of equality of variance at the 1% level with critical value of 2.15.

The computational resources required for the 4 models were tested on 50, 100, 250 and 500 randomly sampled points. Each model was fit 100 times and the average and standard deviation of the time taken to estimate the HPs and fit the model are shown in Table 4.3. All processes were run in python on a single thread of a quad core 3.60Ghz Intel i7-3820 CPU with 16GB Ram. It can be seen from this that fitting the AV GP model which requires two separate GPs averages about 3.7x slower than the SE GP across all data sizes. The PH GP model starts out at 15x the SE GP for 50 data points but this drops to 3.4x by 500 data points, better than the AV GP. The VH GP on the other hand starts out at 57x the SE GP for 50 data points and grows up to 138x for 500 data points as the load on optimising the 500 variational parameters grows. The large run time and standard deviation for the VH GP, which increases with the data set, make it an unlikely candidate for real time adaptive sampling in an embedded CPU, which is the future direction this work is motivated by.



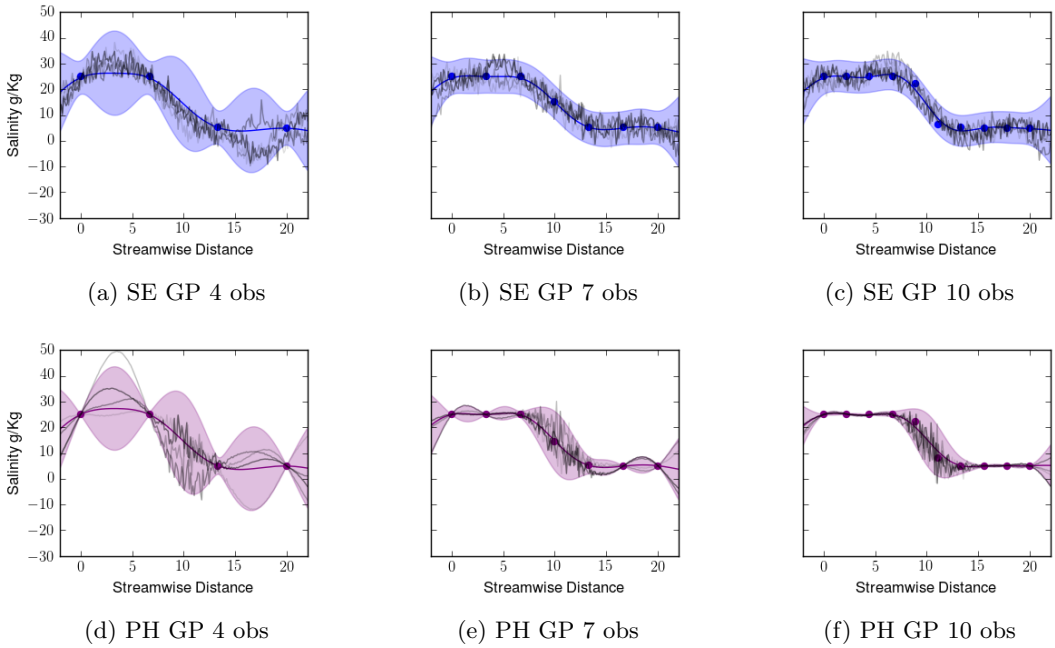
(a) (b)

Figure 4.3: Samples paths drawn from the 1D source distribution shown in Figure 4.1 (a), predicted mean and 2 standard deviations bounds for SE GP and PH GP models (b)

The SE GP and the PH GP model are now compared to examine the model produced and the potential paths generated for a given set of observations. Using the same noise characteristics as in the 1D example above, though this time with the domain set at [0,20] and the front located at [10], 6 sets of 50 observations are drawn evenly spaced through the domain. This can be seen in Figure 4.3 (a). This is to simulate sampling across the front 6 times and receiving different observations each time from the same generating function. Both the standard SE GP and the PH GP have HPs optimised to this data. The resulting mean and $\pm 2\sigma$ bounds of the resulting models can be seen in Figure 4.3 (b).

With these HPs, 4, 7 and 10 observations are now drawn from the same generating distribution. Figure 4.4 shows the resulting GP mean and 2 standard deviation bounds that result from these observations as well as overlaying 4 possible realisations of the process which could have generated these points under the two models. As seen in Figure 4.2, the SE GP model fits a flat noise across the domain, whereas the PH GP model fits a higher noise in around the front. The possible paths show another aspect of this. It can be seen in the SE GP paths in Figure 4.4 (a)-(c) that the paths show a similar level of noise throughout the sample space due as would be expected from a standard Weiner process. The PH GP model in comparison shows much smoother paths away from the front, whilst increasing in noise around the front.

In summary, for data that exhibits Heteroscedastic noise, Homoscedastic models may



(a) SE GP 4 obs (b) SE GP 7 obs (c) SE GP 10 obs

(d) PH GP 4 obs (e) PH GP 7 obs (f) PH GP 10 obs

Figure 4.4: Possible paths overlaid on observations, mean and 2 standard deviation bounds for SE GP and PH GP models with HPs optimised from training set in Figure 4.3 for 4, 7 and 10 observations

be adequate for modelling the mean of the distribution but they fail in modelling the distribution of the variance across the input space. As producing a model of the uncertainty is a key benefit of Gaussian Processes, if it is suspected that the data may exhibit this property, then a model should be chosen which can cater to this. This can be handled flexibly with VHGP. If prior knowledge can allow the noise to be modelled with a parametric function, both the complexity of the model and the computational resources required can be reduced. Higher instability was seen in the models resulting from fitting the VHGP, likely due to the excessive flexibility in the model which has more HPs than data points. It was also noted that the resulting kernel HPs in the VHGP model were highly dependent on the initial values given, even with a very similar resulting model. This was possibly due to the flexibility in the model provided by the variational weight parameters. This leads to the issues that compared to the other 3 models examined, the values of the kernel parameters in the VHGP cannot be used to describe the behaviour of the data, as it is only when combined with the vary large number of variational weight parameters that they have any meaning. Finally the VHGP does not allow for easy addition of new observations or transfer of the model to a new data set as the variational weights are tied to the observations in the model

and re-optimisation of HPs to calculate the new variational weights must be conducted in both cases. This is in contrast to the other models where the data relationships are controlled solely by a small number of HPs which can be used directly on additional data, or a new data set, without running a new gradient descent optimisation of the LML w.r.t the HPs.

4.6 Adaptive Sampling Theory

One key benefit of GPs is that they provide both mean and variance information. The expected variance $\hat{\mathbf{\Sigma}}_*$ at a test point X_*, given the samples X, and the HPs θ, is based on the location of that point. Two measures which use this information to drive sampling decisions and one which instead considers the uncertainty in the HPs themselves will now be discussed.

4.6.1 Entropy

Entropy provides a measure of information content. In the context of GPs the entropy of locations in the input space can be calculated from the predicted variance. The differential entropy in d dimensions is:

$$H(X_*|X,\theta) = \frac{1}{2}\log|\hat{\mathbf{\Sigma}}_*| + \frac{d}{2}(\log 2\pi e)) \tag{4.14}$$

adaptive sampling decisions are then made according to:

$$X_*^E = \underset{X_* \subset \bar{X}}{\arg\max}\, H(X_*|X,\theta) \tag{4.15}$$

where $\bar{X}$ is the set of potential sampling points.

4.6.2 Mutual Information

An alternative measure is based on MI [73]. The authors use the following greedy approximation:

$$X_*^{MI} = \underset{X_* \subset \bar{X}}{\arg\max}\, H(X_*|X) - H(X_*|\bar{X} \backslash X_+) \tag{4.16}$$

where $X_+ = X \cup X_*$ and for simplicity of exposition it is assumed the potential sampling points $\bar{X}$ are also the locations at wish it is desired to predict. By taking account of the areas at which there is interest to predict, predictive power is concentrated in these areas. Whilst this method does not explicitly deal with uncertainty in the HPs, as discussed in [73], MI is robust to HP uncertainty.

4.6.3 Fisher Information

A key flaw with entropy and MI as normally used is they do not take into account uncertainty in the HPs. Generally θ is not known, only its estimate, $\hat{\theta}$. The Fisher Information (FI) matrix measures the information the observations give about the parameter estimates. Its elements are defined as [91]:

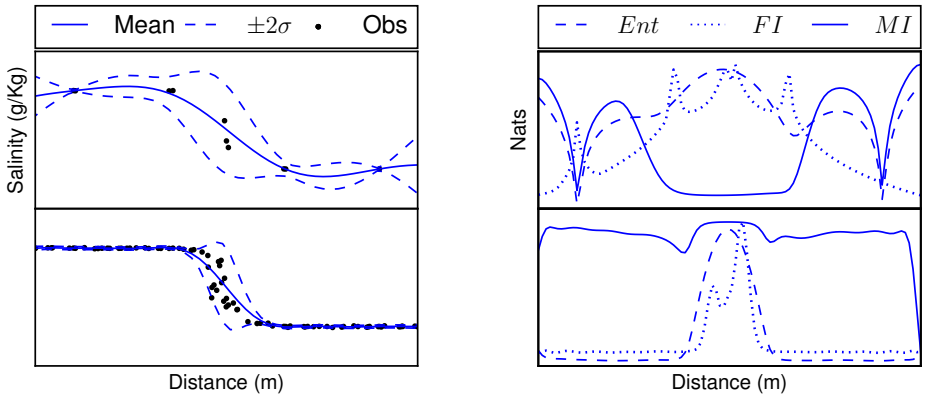
$$[F(X_+, \hat{\theta})]_{ij} = \frac{1}{2}\text{tr}\left(K_{R+}^{-1} \frac{\partial K_{R+}}{\partial \hat{\theta}_i} K_{R+}^{-1} \frac{\partial K_{R+}}{\partial \hat{\theta}_j} \right) \tag{4.17}$$

where tr(.) is the trace operator, $K_{R+} = K(X_+, X_+) + R$ and the partial derivatives are those derived in Appendix B.2.1, taking into account the potential sample point. Its inverse gives the Cramér Rao inequality:

$$\Sigma_{\hat{\theta}}(X_+, \hat{\theta}) \geq F(X_+, \hat{\theta})^{-1} \tag{4.18}$$

which is an asymptotic lower bound on the variance of the parameters. It has been shown empirically in [170] that this asymptotic bound is a reasonable predictor of the ranking of actual parameter variance in small samples. The Fisher Information matrix has been used in a number of ways to guide sample selection. The trace of its inverse is used in [162] which considers the sum of the variances, whilst the determinant used in [168] also considers the covariances and leads to the following metric.

$$X_*^{FI} = \underset{X_* \subset \bar{X}}{\arg\max}\, ln|F(X_+, \hat{\theta})| \tag{4.19}$$



(a) (b)

Figure 4.5: Model fit (a) and comparison information measures from which the maximum points should be sampled (b) for 10 points (top), 80 points (bottom) drawn from distribution in Figure 4.1(a).

The effect of these information measures on samples drawn from the one dimensional distribution shown in Figure 4.1(a) can be seen in Figure 4.5. Two sets of samples with 10 and 80 points are chosen to highlight some of the typical differences observed. It can be seen that in the top sub-figure, Entropy has a maximum near the center of the distribution, which is where its estimate of the front lies. It also has some lower peaks located in the spaces between observations. FI also shows a maximum in the center on the top sub-figure, though with some peaks located near other observations, as this can help in determining the HPs related to noise. Both of these measures quickly converge to a maximum near their estimate of the center of the heteroscedastic noise, as shown in the bottom sub-figure, and then do not sample elsewhere in the space. In the case of Entropy this is due to the heteroscedastic noise quickly dominating the total uncertainty. For FI, the HPs relating to the covariance are equally measured across the input space, whilst those relating to the location of the front require sampling near the estimate of the front. The combination of these is thus dominated by input locations near the front estimate. This leads to issues of both poor predictive performance away from the front and the potential of becoming stuck in local maximums if the estimate of the front location is wrong. An animation of this across a larger number of timesteps can also be seen online [3].

In comparison, MI initially has local maxima away from the estimated front location, as can be seen in the top sub-figure. After this area is sampled, the maximum MI tends

[3] urlhttps://youtu.be/zR4abnICCjM

to move around the whole input space, though with a bias towards sampling in the area around the front. An example of this is seen in the bottom sub-figure. This behaviour is further examined in the next section on a two dimensional example.

4.7 Adaptive Sampling Results

Numerical simulations are now run to show the impact of adaptive sampling with these different measures on the predictive power of the model. Adaptive sampling is tested based on the following methods: Random, Entropy, MI, and FI. The two dimensional simulation of an arrested bathymetric front shown in Figure 4.1(b) and (c) is used.

For all the methods except random planning, the relevant measure at a number of points at each time step is evaluated to decide which points to sample. To cover the input space with a uniform discretisation, especially as the dimensionality of the problem increases, becomes computationally expensive. It may also be beneficial to sample with non-uniform spacing as discussed w.r.t. FI in [170]. In relation to the PHGP model, some parts of the input space have more relevance for HP uncertainty than others, i.e. around the heteroscedastic noise center. Instead of a uniform discretisation, standard techniques in Monte Carlo numerical integration are followed and 100 independently and identically distributed points are randomly drawn at each step to be evaluated, from which one is chosen for each model. These are drawn from a uniform distribution across the input space and thus all locations are equally likely to be sampled. This allows each model over the course of each simulation to choose the spacing required between the sampling locations that best optimises its measure as detailed in Equations (4.15), (4.16) and (4.19).

Note for the MI models the $\bar{X}$ points are evenly distributed across the dimensions of the input space. To handle sampling points X_+ which are offset from the grid locations of these uniformly discretised $\bar{X}$ points, in evaluating $\bar{X} \backslash X_+$, the X_+ points are rounded to the nearest $\bar{X}$ location.

For computational expediency, whilst the HPs are optimised and the models evaluated at every step for the choice of the next sampling location, RMSE and RMSSDE are tested only every 5 steps, at which 1000 random points are drawn from a uniform

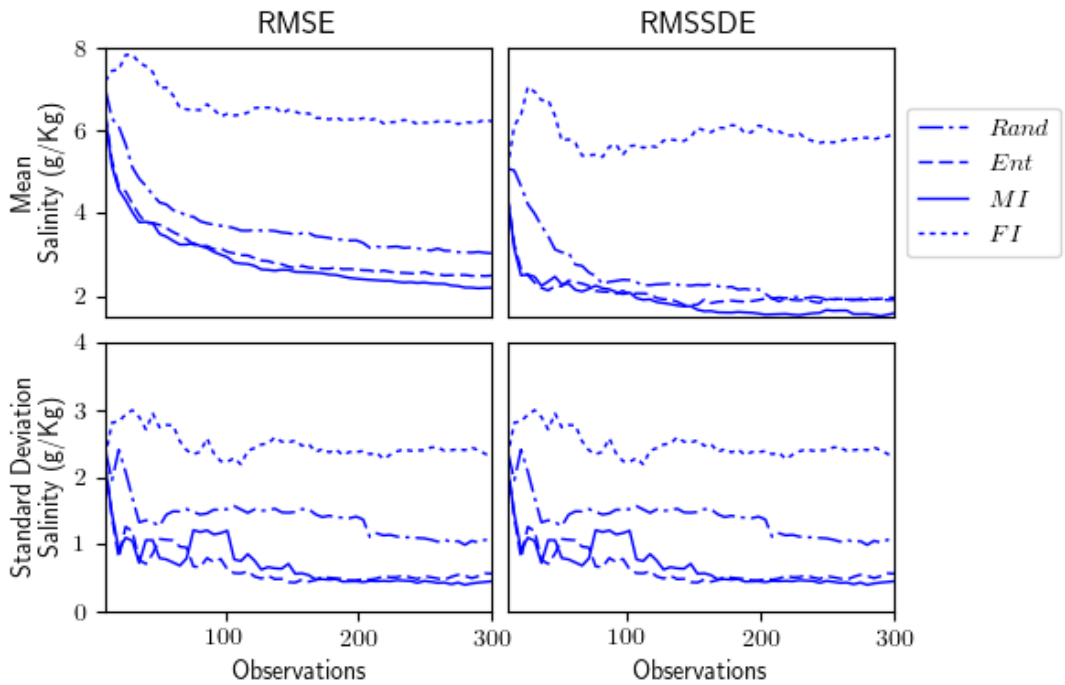
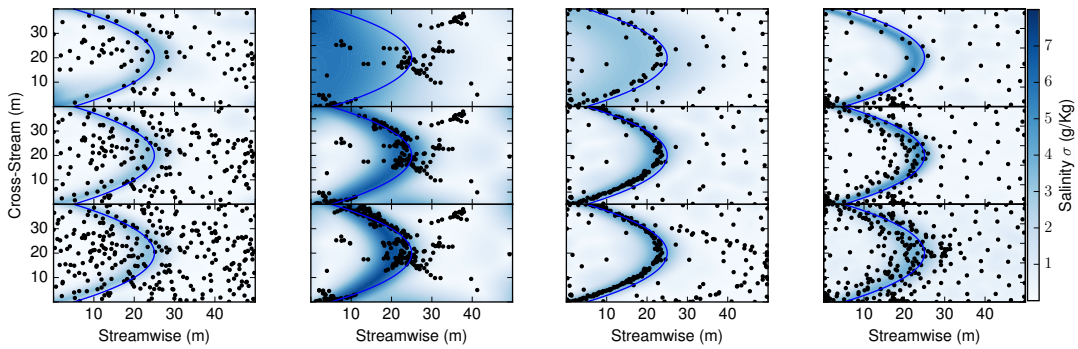


Figure 4.6: Convergence results from 100 simulations as the number of observations increase with Random sampling compared to adaptive sampling based on sequentially choosing the points which maximise Entropy, Mutual Information or Fisher Information for the mean (top) and standard deviation (bottom) of RMSE (left) and RMSSDE (right)

distribution across the input space against which each model's predictive power is evaluated. The results averaged across 100 simulations can be seen in Figure 4.6 with the mean in the top half and the standard deviation on the bottom.

It can be seen that planning based purely on Fisher Information does poorly in terms of the predictive power of the first two moments of the distribution across the input space. Random, Entropy and MI all do much better. MI clearly outperforms on both measures being around equal with Entropy for the first half and then outperforming in the second half. The reasons for this can be seen in Figure 4.7 which shows the sampling locations for one simulation for each of the 4 methods at 3 points in time, after 100, 200 and all adaptively sampled points and associated predicted standard deviation. As expected random sampling relatively evenly covers the entire space. MI based sampling also results in the lowest standard deviation across the simulations. This is important as it gives confidence that the mean results shown in the top half of Figure 4.6 will actually be achieved when conducted in the field. FI in comparison, is only driven by sampling at points to reduce the uncertainty in the HPs, as can be



(a) Random (b) Fisher Information (c) Entropy (d) Mutual Information

Figure 4.7: Sample locations (black dots) for one run for the first 100 (top), 200 (middle) and all (bottom) samples for different adaptive sampling methods. Source location of front indicated by the blue line, predicted standard deviation as surface plot.

seen in Figure 4.7(b). Under PHGP the locations which are most relevant to reducing HP uncertainty are those relating to the location of the heteroscedastic noise and thus sampling is focused around the area that will help the current estimate. This leads to two problems. Firstly points away from the current estimate of the front are not sampled resulting in poor predictive performance there, as can be seen by the predicted standard deviation. Secondly, if the estimate of the location of the front is wrong, the model can get stuck in a local minimum and not find the correct location.

Entropy based sampling initially explores the input space relatively evenly until it produces a compact estimate of the heteroscedastic noise. Once achieved, this noise overpowers the uncertainty from the kernel due to sparsity of observations and sampling is very tightly focused around the estimate of the front location. This can be seen by the sample points close to the blue line in Figure 4.7(c). Note also in the bottom panel of Figure 4.7(c) that there is an arc of points further across from the true mean where the model converged on some incorrect parameters for the location of the front for a while. Whilst sampling directly on the middle of the predicted front will help in optimising the HPs relating to the location of the front $\mathbf{m}$, it doesn't particularly help in evaluating the width of the noise around the front as determined by l_d^2 and σ_{n2}^2. This can be seen in the top panel of 4.7(c) where even with the tight sampling around the front, the noise estimate is not tight around the front (compared for instance with the top panel in 4.7(d)) as the model has been unable to correctly estimate the width of the heteroscedastic noise. By sampling so locally to the predicted front location it also

can get caught in a local maximum around an incorrect estimate, though the wider initial sampling than FI helps to counter this.

From the sampling locations shown in Figure 4.7(d), it can be seen why MI outperforms the other methods in its predictive performance. As with entropy it initially samples widely across the input space. The uncertainty due to the heteroscedastic noise which overpowers sample selection in Entropy is tempered in MI as it occurs in both the first and second component of Equation (4.16). Sampling is biased around the estimate of the location of the front, but not as closely as in pure Entropy driven sampling. This aids in estimating l_d^2 and σ_{n2}^2, as can be seen in the standard deviation prediction in the top panel of Figure 4.7(d). In addition, as it is not sampling as close to its current estimate, it can be seen can see in the bottom panel of Figure 4.7(d), MI based sampling continues to place some points away from the front, thus improving predictive power here. An animation of this process can also be seen online [4].

4.8 Summary

It has been shown in this chapter that when modelling processes exhibiting non-stationary noise, Heteroscedastic GPs provide superior predictions. A standard Homoscedastic GP with a stationary covariance function and the addition of Augmented Variables were shown to be able to match the mean but not the variance of these distributions.

Heteroscedastic GPs were shown in simulation to be able to model the mean and the standard deviation of a simulated salinity gradient across the input space. Equality of variance tests were applied in a novel manner on the normalised errors showing they converged to white noise only in the case of the Heteroscedastic models. Additionally a new measure, the Root Mean Square Standard Deviation Error (RMSSDE) was defined and used to show that only the Heteroscedastic models converged towards the spatial distribution of the variance predicted from the source distribution.

By placing a parametric prior on the distribution of the noise function it is possible to jointly estimate the HPs for a simple kernel and the noise function simultaneously.

[4] `https://youtu.be/lwyaeyZOR-M`

This compares to the Variational Heteroscedastic GP which whilst able to be jointly estimated, required a second GP for the noise and a diagonal matrix of variational parameters with one entry for each observation. The proposed parametric model was shown to match the predictive performance of this more complicated model with lower computational resources. The small number of parameters of the model also allows their influence to be directly understood whereas the VHGP model kernel parameters can be offset by the variational parameters and thus have no direct interpretation. The kernel parameters of PHGP allow new data to be added or the parameters to be applied to a new data set directly, whilst VHGP requires re-optimisation of the HPs to calculate the variational parameters.

Numerical simulations are presented which show MI driven sample selection under PHGP is superior to Entropy, FI or Random sampling. It is shown to provide convergence to lower levels of error in the estimates of the first and second moments of the the distribution of interest. This is due to MI driven sampling covering the input space with a bias towards the areas of higher noise. In the next Chapter this work is extended to path planning with holonomic constraints and model dynamic salinity fronts.

Chapter 5

Adaptive Path Planning to Model a Dynamic Estuarine Tidal Front

5.1 Introduction

In Chapter 4 the problem of efficiently sampling within a scalar field exhibiting input location dependent noise was examined. The scalar field in question was assumed to be stationary, representing the surface salinity observed in a bathymetrically arrested Tidal Front as discussed in [66] and shown in Figure 2.3 (a). This is now extended to deal with a scalar field where the noise process is dependent on a location that slowly moves with time. This type of process would be observed in the surface salinity manifestations of a Class II tidal front as discussed in [76, 139] and shown in Figure 2.3 (b). The analysis from sample selection is also extended to online path planning for a holonomically constrained simulated Autonomous Surface Vessel, such as that used in Chapter 3.

The contributions of this chapter are as follows. The work in the previous chapter on Parametric Heteroscedastic Gaussian Processes is extended to handle a moving uncertainty front. Path planning with holonomic constraints is applied under this model and Entropy, Mutual Information (MI) and Random path planning are compared on

the accuracy of the first and second moments of the models produced. An adjustment to MI is suggested to reduce the dimensionality introduced by the time domain. It is shown how defining a parametric mean function, the parameters of which are fit online jointly with the kernel HPs, increases both the speed of computation and the accuracy of models produced. Analytical derivatives of the Log Marginal Likelihoods are derived to enable the use of fast gradient descent solvers. A number of path depths are compared to show the trade-offs between exploiting the current best model and re-optimising the model.

This chapter proceeds as follows. Section 5.2 presents related work in the areas of spatio-temporal GPs and path planning. Section 5.3 discusses GPs, the spatio-temporal kernel and mean functions to be used and provides analytical expressions for the LML gradients of this model. Section 5.4 briefly outlines the information theoretic measures which will be used to evaluate paths for path planning. Experiments on simulated moving salinity fronts are presented in Section 5.5. Section 5.6 provides a summary of the Chapter.

5.2 Related Work

Moving processes can be handled under GPs by modelling time in addition to spatial dimensions. The creation of valid spatio-temporal stationary covariance kernels is discussed in [22]. A separable covariance function with a drift term is used in [79] to model moving wind fields and in [90] to model a simulated moving hotspot. Both separable and non-separable non-stationary covariance kernels are used in [140] and non-separable non-stationary covariance kernels in [45] for sample selection in environmental monitoring. Whilst non-separable kernels theoretically allow for more complex relationships between the spatial and temporal dimensions [22], to do so requires additional complexity and does not neccesarily result in more accurate predictions. Analysis in [140] comparing combinations of stationary/non-stationary and separable/non-separable kernels on two experiments found varying results in terms of RMSE, though in both tests the simpler stationary non-separable kernel converged to the best results either equalling or outperforming the more complex and computationally demanding kernels.

Unlike the sample selection problem in which the submodularity of Entropy and MI can be relied upon to provide guarantees on the optimality of a greedy strategy for planning [73], moving to path planning where successive sample points are restricted by the sampling frequency of the sensors and the holonomic constraints on the platform can lead to a greedy strategy becoming trapped in local minima. Integrating the value of longer paths allows the planner to escape from these minima. It has been shown in [104] that deeper planning can produce more optimal outcomes. The HP's of the model are learned as sampling progresses, which dictates the information of a given path. Thus planning horizon depth is a trade off between escaping local minima and planning under an incorrect model. Note that finding the optimal path is NP-hard. For online implementation we can only expect to find an approximate solution.

Rapidly-exploring Random Trees (RRT) are a sampling based method where a tree is randomly grown from the start position until a path is found to the goal [78]. The path can also be grown from both ends. Probabilistically it is guaranteed to eventually find a path if one exists as the sample space is filled [74], with an exponential rate of decay in the probability of failure as the sample size increases [42]. RRT* is introduced as an asymptotically optimal variant, further enhancing the theoretical support [70]. This is shown to easily incorporate ordinary differential constraints of the vehicle in planning.

Potential methods for path creation such as Rapidly-exploring Information Gathering (RIG) [63] and forms of MCTS such as implemented by [79, 104] which generate paths from discrete distributions could also be used. The MCTS results in computational complexity which is exponential relative to the depth examined. This exponential cost is countered in [79] by evaluating the paths at each depth and only propagating the highest ranked paths. This provides no guarantee that optimal paths are not pruned early. The early pruning of paths they use to reduce the search space can also inhibit the ability of the planner to produce paths which escape local minima by pruning initially suboptimal plans. It does not appear these methods scale to real time planning in the field. The example implementation of RIG in the field of an ASV relied on offline planning then uploaded to the vehicle with an aggressive pruning strategy consuming 1 minute of planning time. These methods also require iterative querying of the GP,

and thus for the same computational budget are unlikely to be able to sample as densely.

Pure random walks can be used for each planning cycle allowing a relatively large number of paths generated in continuous space to be evaluated quickly [103]. A pure random walk is probabilistically complete, in that it is guaranteed to cover the search space within the horizon as the number of paths increases. At each step a heading change is drawn from a uniform distribution within the turn rate limits, and a new sampling position is chosen by progressing at a constant velocity for the time step determined by the sampling frequency, constrained such that the heading chosen does not take the path outside the mission boundary. As the potential paths can be generated independently of their evaluation, the GP can be queried as a batch. As the control actions are chosen from a continuous space, the paths are not restricted to a discretised set of control actions or sampling locations as in Monte Carlo Tree Search (MCTS) algorithms. In this study, potential paths to a given horizon are generated by a pure random walk considering holonomic constraints and a boundary area.

5.3 Gaussian Processes

5.3.1 Time Dependent Heteroscedastic Noise

The general framework for Gaussian Processes was outlined in Section 2.3 and extended to the case of Heteroscedastic noise in Section 4.3.1. A 2D front was modelled with the noise across the front (streamwise) in the $\mathbf{x_1}$ dimension approximated by a Gaussian distribution and the center of this Gaussian in the $\mathbf{x_2}$ dimension (cross-stream) approximated by a 2nd degree polynomial, as shown in Figure 4.1. Time was ignored as it was assumed this process was stationary. Extending the results to a moving front with constant velocity, the following parametric noise function $D(\mathbf{x})$ is proposed:

$$D(\mathbf{x_1}, \mathbf{x_2}, \mathbf{t}) = \frac{1}{l_d\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{\mathbf{x_1} - \mathbf{p}}{l_d}\right)^2\right) \tag{5.1}$$

$$\mathbf{p} = p_{dt}\mathbf{t} + p_0 + p_1\mathbf{x_2} + p_2{\mathbf{x_2}}^2 \tag{5.2}$$

where $\mathbf{p}$ is the location of the midpoint of the noise in the $\mathbf{x_1}$ dimension associated with the values contained in $\mathbf{x_2}$ and $\mathbf{t}$, $\mathbf{p_{dt}}$ is the drift w.r.t $\mathbf{t}$ and p_0, p_1, p_2 are the polynomial factors up to degree 2.

5.3.2 Mean Functions

The framework for GPs easily handles mean functions. These are often implemented as zero [79, 89, 140], and sometimes without even mentioning this is being done [9, 11, 64, 144, 164]. Prediction close to observed points is dominated by the covariance function, so in cases of high density of data and prediction close to these observations relative to the estimated length scales, use of a mean function will have little impact. Prediction further away from observations, and/or in case of low observation density reverts to the mean function, which if set at zero may result in large prediction errors. In cases where the GPs are only being used to produce uncertainty information, for instance to drive planning in [11], it may seem that since the predicted covariance function as shown in Equation 2.4 does not depend on the mean function, this assumption would have no impact. This will only hold in the case of known HPs. If the HPs are to be estimated from the data, then the use of a different mean function may impact the HPs which are estimated. It is shown in Appendix B.2.2, that the observations adjusted by the mean function, $\mathbf{y_{adj}}$, explicitly appear in the Log Marginal Likelihood function, which is maximized to determine the HPs which are most likely for the chosen model given the data.

This leaves the question of how to choose a functional form for the mean function. If some prior knowledge of a parametric mean function which may suit the data can be applied, this can be included in the LML functions and its parameters can be fit along with the HPs in the model.

It is known that downstream of the tidal front salinity should be high and upstream it should be lower. Using this knowledge it is proposed to model the mean of this transition with a logistic function to the $\mathbf{x_1}$ dimension. The midpoint of this function is the associated with the midpoint of the noise function, as driven by the underlying physical processes. This will reduce the number of parameters to estimate in the model and allows the optimisation of these parameters to use both the level and noise of the

Y observations. The location of this midpoint in the $\mathbf{x_2}$ dimension is thus determined by the same polynomial function as in the noise process, i.e Equation 5.2. The adjusted salinity value defined by the Logistic Polynomial mean function is:

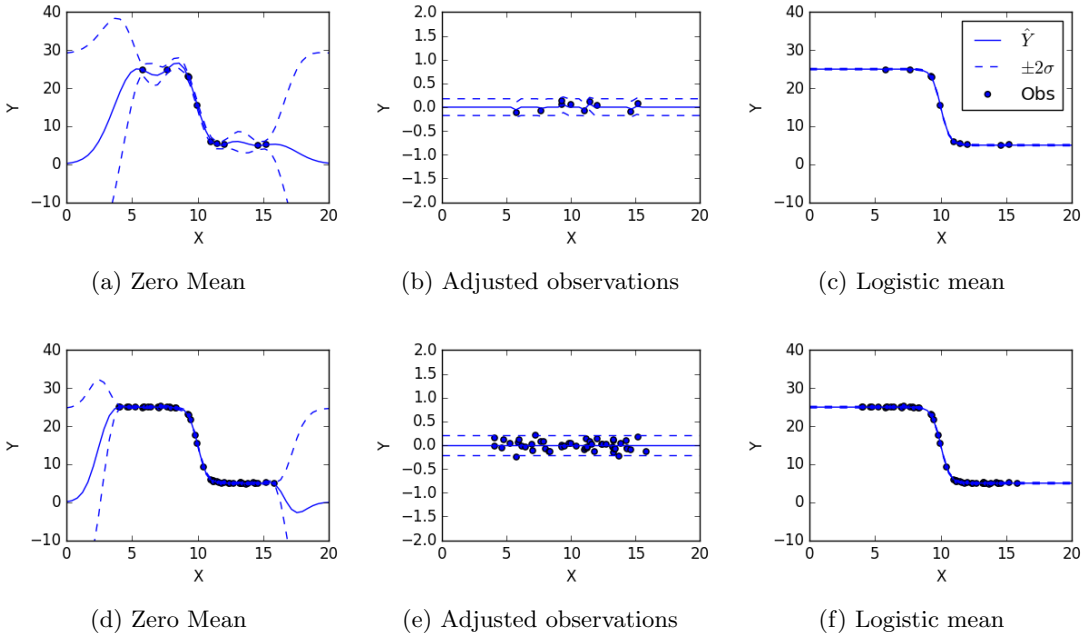
$$\mathbf{m} = \left(a + \frac{b}{(1 + \mathbf{e}^{-f(\mathbf{p}-\mathbf{x_1})})}\right) \tag{5.3}$$

where a, b and f define the minimum value, range and slope of the logistic function. The predicted mean from the adjusted model is:

$$\hat{\mathbf{y}}_* = K_* K_y^{-1} \mathbf{y_{adj}} + \mathbf{m} \tag{5.4}$$

where $\mathbf{y_{adj}} = \mathbf{y} - \mathbf{m}$. The derivatives of the LML w.r.t the HPs of this model can again be analytically derived and are presented in Appendix B.2.2.

This can be demonstrated in a simple 1D example with homoscedastic noise where $\mathbf{p}$ is just a scalar representing the midpoint of the logistic function as seen in Figure 5.1. Samples are drawn from a logistic function with homoscedastic noise. Figure 5.1 (a) shows a standard homoscedastic GP with a squared exponential kernel with HPs optimised by maximising the Log Marginal Likelihood fit to 10 observations. This displays a number of issues. With the mean function set to zero, movement away from the data to either side results in the predicted mean value $\hat{Y}$ tending to zero. The uncertainty also expands to a very large number as shown by the $\pm 2\sigma$ bands as the learned σ_f is large. This also leads to large uncertainties within the range of the data where there are some spaces between the data points. This last characteristic of the model can be addressed with increased data as seen in Figure 5.1 (d). There are still issues with the predicted mean reverting to zero outside the prediction area and the large uncertainty bounds here. Figure 5.1 (b) shows the adjusted data after the mean function has been removed (in this case the exact source function is used). In the next section, LML gradients are provided so this can be jointly optimised, and the resulting predicted mean and $\pm 2\sigma$ bands. Figure 5.1 (c) shows the resulting predictions with the mean function added back in. It can be seen that with the addition of the mean function the model no longer reverts to zero outside the predicted area and also



(a) Zero Mean (b) Adjusted observations (c) Logistic mean

(d) Zero Mean (e) Adjusted observations (f) Logistic mean

Figure 5.1: Simple 1D example of the impact of a mean function on GP predicted mean and variance for 10 (a,b,c) and 50 (d,e,f) sample points drawn from a logistic function with homosecdastic noise. This shows the impact of using a mean function with the zero mean model (left) requiring more observations to accurately predict near observed points and still reverting to the arbitrary level of zero with wide uncertainty away from them due to the short length scale learnt.

the large increase in predicted variance disappears. The less well the mean function matches the data, the more work the GP has to do in fitting the adjusted observations. It can also be seen that the model fits the data very will with even the 10 data points as after the mean function was taken out, the resulting adjusted observations produced a much simpler surface. An animation of this figure can also be seen online [1].

5.4 Adaptive Path Planning

Metrics for comparing the informativeness of potential sampling points were presented in Section 4.6. The greedy approximation to MI presented by [73] required discretisation of the sampling space for the calculation of $\bar{X} \backslash X_+$. As the input dimensions increase the curse of dimensionality presents itself. It is proposed to calculate this only for coverage in the spatial domain. As the phenomenon of interest is moving, the location of the observed points is adjusted relative to this front. Thus $X_+ = X_{adj} \cup X_*$ where

[1] `https://youtu.be/VHouvMOa9Cs`

Initial Obs	15
Planning Obs	1000
Test Obs	1000
Random Paths	100
MI $\bar{X}$ obs	2500
Simulations	25
σ_{n1} (g/Kg)	0.1
σ_{loc} (m)	1
Δt (s)	1
Front Velocity (m/s)	0.05
Input Space (m)	[0,0,100,40]
Front Location (m)	[[0,5],[25,20],[50,5]]
Start position (m)	[15,20]
Start Heading (rad)	0
Turn Limit (rad/s)	$\pi/2$
ASV Velocity (m/s)	1.0

Table 5.1: Parameter settings for simulation of dynamic tidal front and ASV dynamics

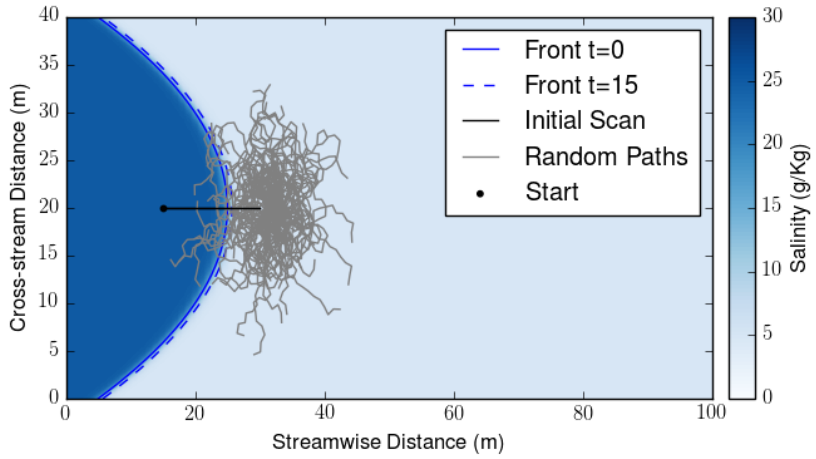
the previously sampled locations are adjusted in the $\mathbf{x_1}$ dimension by the estimated drift p_{dt} multiplied by the time difference between the observation time and the current period.

5.5 Experiments

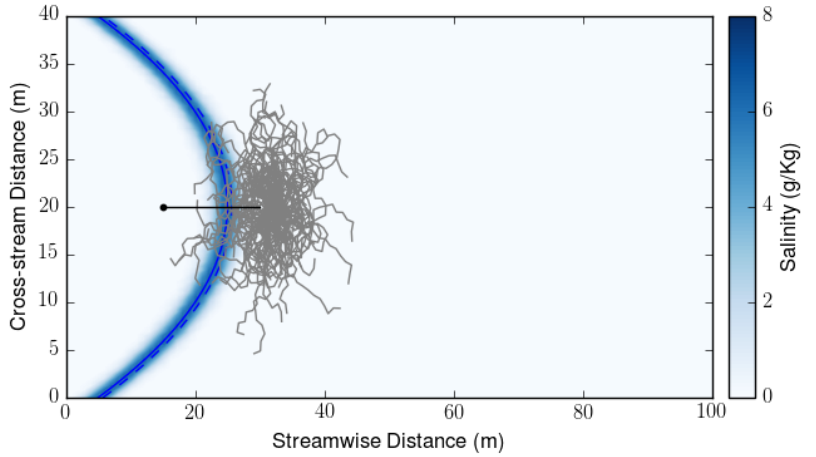
The ability to produce an accurate model of a moving scalar process under the PHGP model is now tested using a simulation of the surface salinity of a moving tidal front. Similar to the simulation used in Chapter 4, a front is defined in two spatial dimensions with heteroscedastic noise centered around the front. In this case a front velocity parameter is also defined to simulate a slowly moving Class II tidal front as discussed in Chapter 2 and detailed in Figure 2.3 (b). The parameters of the simulation are listed in Table 5.1.

5.5.1 Impact of Mean Function

The impact of using the Logistic Polynomial Mean Function as described in Section 5.3.2 is now tested. As in Chapter 4, the Root Mean Square Error (RMSE) and Root Mean Squared Standard Deviation Error (RMSSDE) are used to estimate accuracy of the models in predicting the first 2 moment of the source distribution. A set of



(a) Mean



(b) Standard Deviation

Figure 5.2: Simulated dynamic tidal front distribution of salinity. Mean (a) and standard deviation (b) overlaid with example paths, start point, initial scan and distance front has moved over the initial window

1000 random points are drawn from a uniform distribution across the 2 spatial and 1 time dimensions. The mean and standard deviation of these points are estimated from the source distribution, and these are compared to the predictions for the model. A random sample of 1000 observations in batches of 20 are drawn. After each resample, two PHGP models are fit, one with zero mean and the other with the Logistic Polynomial mean function with 10 restarts on the optimisation with the initial HPs drawn from the distributions shown in Appendix B, Table B.1. After the first step, one of these restarts will be seeded with the results from the previous optimisation. The GPs are then fit and the time taken for optimisation and fitting in both models, and the predictive power of the generated models in terms of RMSE and RMSSDE for

1000 randomly drawn points across the 2 spatial and 1 time dimensions is calculated. The results of this can be seen in Figure 5.3. Fitting time between the two models is relatively similar, the large difference in time is due to the HP optimisation. It can be seen that including the mean function, even though it introduces 3 more parameters, reduces the average time required to converge on a result. It also results in a lower standard deviation in the time. This is due to the mean and noise process using the same polynomial function parameters, which allows information from both the signal and the noise of the observations to drive the HP optimisation. It can also be seen that using the mean function also significantly reduces the RMSE and RMSSDE, and in the case of the RMSSDE also results in a much lower standard deviation of this value. One of the main reasons for this is that the GP with the mean function learns a longer characteristic length scale in the spatial dimension l_x. This allows information from the sampled data points to be projected over a larger area which improves the predictive performance. It also does not suffer from mean predictions reverting to an arbitrary zero as test points move away from sampled areas.

5.5.2 Path Planning Results

Whilst planning for sample selection is interesting, of greater relevance to sampling strategies for moving platforms is to analyse path planning taking into account the holonomic constraints in terms of velocities and turn rates. An ASV similar to that used in Chapter 3 is simulated. As such a velocity of 1m/s and limit the tun rate to $\pi/2$ rad/s is set. Sampling is conducted at 1Hz and the other relevant parameters defining the environment are shown in Table 5.1, and the salinity distribution i shown in Figure 5.2. An animation of the simulated tidal front can also be seen online [2]. The ASV is driven in a straight line for 15s from the starting position across the front to gather some initial data points, as shown in Figure 5.2. The HPs are then optimised and the GP model fit.

A set of 100 random paths are generated by sampling a heading change uniformly within the turn rate limits specified in Table 5.1 at each time step Δ_t and applying these to a vehicle moving at the constant velocity of 1m/s. These random paths are

[2] `https://youtu.be/MiPZ8Aa7oR4`

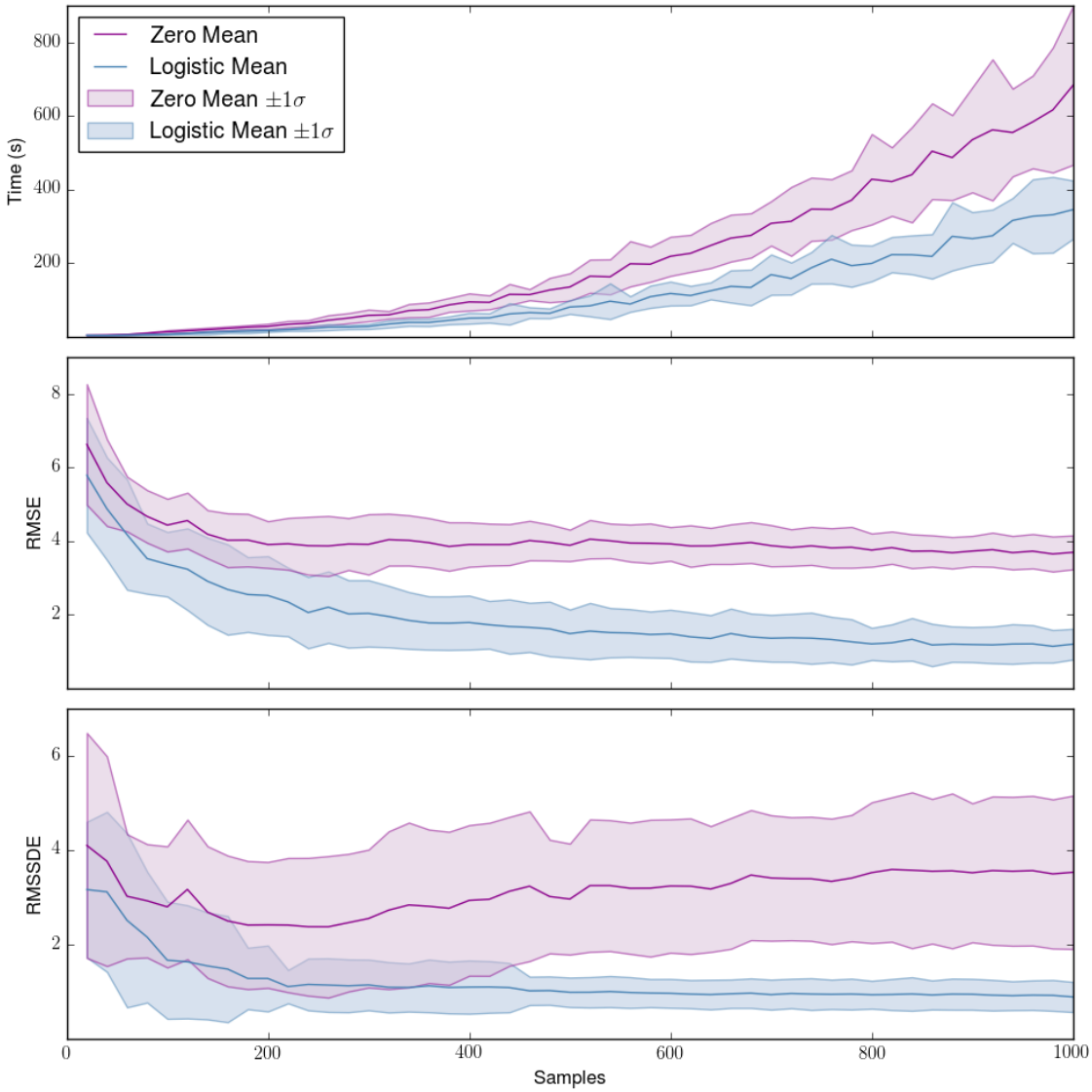


Figure 5.3: Comparison from 25 simulations of sequential sample selection in batches of 20 points for PHGP model on moving front for zero mean vs logistic mean function in terms of optimisation + fitting time (top), RMSE (middle) RMSSDE (bottom).

then evaluated by Entropy and MI as discussed in Section 5.4, the best path under each metric is chosen for the given model and one is randomly chosen for the random model. Three planning horizons of 10, 20 and 50s are tested. In each scenario the full path is followed and samples collected after which the HPs are re-optimised, model refit and path re-planned. Additionally for the horizon of 20 steps, the PHGP zero mean model is analysed. The majority of the time required to run these simulations comes from the HP optimisation step. Multiple random starts are implemented in the HP optimisation to help avoid converging on local minimums. The assumptions made in setting these random starting values are discussed in Appendix B. To keep the computational load

Mean Function	Horizon	Steps	Opt Restarts	Adaptive Obs	Simulations
Zero	20	50	10	1000	50
Logistic Polynomial	10	100	5	1000	50
Logistic Polynomial	20	50	10	1000	50
Logistic Polynomial	50	20	25	1000	50

Table 5.2: Path planning depth and model variations tested in simulation

Mean Function	Method	Horizon	RMSE	RMSSDE
Zero	Entropy	20	9.50 (4.16)	6.73 (5.11)
Zero	MI	20	6.86 (3.66)	3.76 (2.57)
Zero	Random	20	10.39 (2.83)	9.07 (5.25)
Logistic Polynomial	Entropy	10	2.33 (4.48)	2.79 (5.33)
Logistic Polynomial	MI	10	1.30 (1.75)	1.22 (1.89)
Logistic Polynomial	Random	10	2.75 (3.06)	1.67 (1.73)
Logistic Polynomial	Entropy	20	2.30 (2.53)	2.21 (1.79)
Logistic Polynomial	MI	20	1.42 (1.13)	1.00 (1.00)
Logistic Polynomial	Random	20	2.71 (2.66)	1.62 (1.65)
Logistic Polynomial	Entropy	50	1.47 (1.21)	1.27 (1.08)
Logistic Polynomial	MI	50	2.14 (2.27)	1.65 (2.61)
Logistic Polynomial	Random	50	3.89 (3.32)	2.64 (2.93)

Table 5.3: End of mission prediction errors, Mean (Standard Deviation), for RMSE and RMSSDE from 50 simulations for varying planning horizons and path evaluation methods.

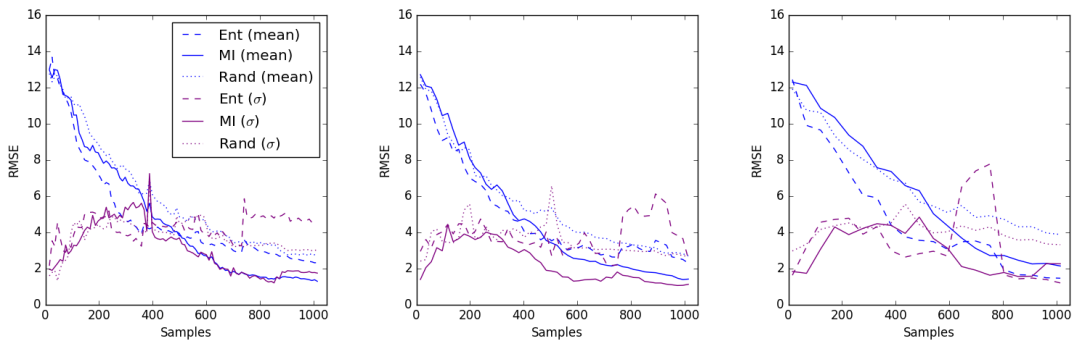
relatively constant and allow the differing scenarios to have the same number of HP optimisation restarts across the mission the number of optimisation restarts for the different planning horizons is adjusted as shown in Table 5.2.

Table 5.3 shows the prediction errors after all samples have been collected in a mission comparing planning horizon, planning method and the mean function used. It is clear that the zero mean model with a horizon of 20 steps performs poorly in prediction of both RMSE and RMSSDE across all planning methods. Thus is was not further tested at other planning horizons due to this poor performance combined with the much longer computational time required. Under the logistic polynomial mean model, random path selection performs poorly for RMSE, though is competitive in RMSSDE for some horizons. MI for the 20 step horizon arguably produces the best predictions across RMSE and RMSSDE, with lower standard deviations than the 10 step MI scenario. Entropy over 50 steps also performs to a similar level.

Examining the evolution of these values across the mission time in Figures 5.4 and 5.5 it becomes clear that the predictions from the models produced through MI planning

with the 20 step horizon are the most stable. The 50 step horizon entropy model which from Table 5.3 produced competitive results, show a large jump in standard deviation of both RMSE and RMSSDE around 600-800 samples, whilst the Horizon 20 MI driven model converges smoothly in both mean and standard deviation from a few hundred samples. The standard deviation of the results is important as whilst these results are aggregated over 50 simulations, the objective is to develop algorithms which can be deployed in the field producing accurate models from a single mission. The large standard deviation shown in the Entropy driven planning missions could lead to poor results when conducted in field tests which may be expensive to conduct.

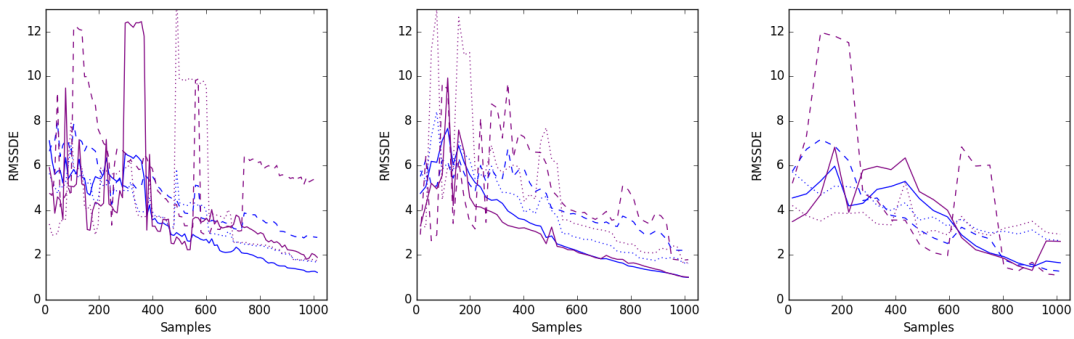
For the longer Horizon of 50 steps the MI driven model likely performs worse than in the shorter horizons as it wastes time exploring an incorrect model. Whereas the Entropy model performs better the longer 50 step horizon, a larger number of points are sampled before the model is refit resulting in more samples away from the front before front following behaviour dominates. This allows a more accurate estimation of the parameters relating to the mean function away from the front. Whilst this might work for this simulation, it would be dangerous to rely on the exploration of an incorrect model to provide adequate sampling away from the peak of the heteroscedastic noise in general and could easily break in another application. MI based planning would thus be expected to be a more robust exploration method.



(a) Horizon 10 (b) Horizon 20 (c) Horizon 50

Figure 5.4: Comparison of mean and standard deviations of errors in RMSE under the Logistic Polynomial Mean model across planning horizon and adaptive planning criteria.

Analysis of a representative run for each of these methods for the 20 step horizon is shown in Figure 5.6. An animation can be seen online which also shows the predicted



(a) Horizon 10 (b) Horizon 20 (c) Horizon 50

Figure 5.5: Comparison of mean and standard deviations of errors in RMSSDE under the Logistic Polynomial Mean model across planning horizon and adaptive planning criteria.

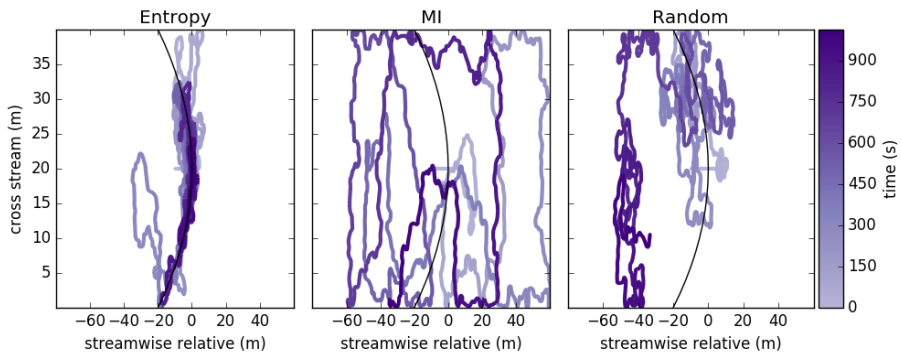


Figure 5.6: Example paths for the Logistic Polynomial Mean model with a planning horizon of 20 comparing Entropy, MI and Random path selection. Streamwise position shown relative to true front position to show spatial coverage of space relative to the front. Time indicated with shading of path.

standard deviation and the path vs the moving front [3]. The input space has been adjusted in the streamwise direction relative to the position of the tidal front at each time step. The colour of the path is graded with time, becoming dark as time progresses. Here it can be seen that under the entropy measure, the model relatively quickly finds the front and explores exclusively around it. This leads to a lack of observations in the area away from the front which impacts predictive power there. This may explain the poor performance of the Entropy measure on small horizons as it quickly finds the front and conducts little sampling away from it. This lack of breadth of coverage away from the front also explains the jumps in standard deviation seen in Figures 5.4 and 5.5 as this leads to a lack of diversity of samples resulting in unstable HP optimisation solutions. Random planning as expected wanders around the space with no direction.

[3] `https://youtu.be/8YvU-QkCdVo`

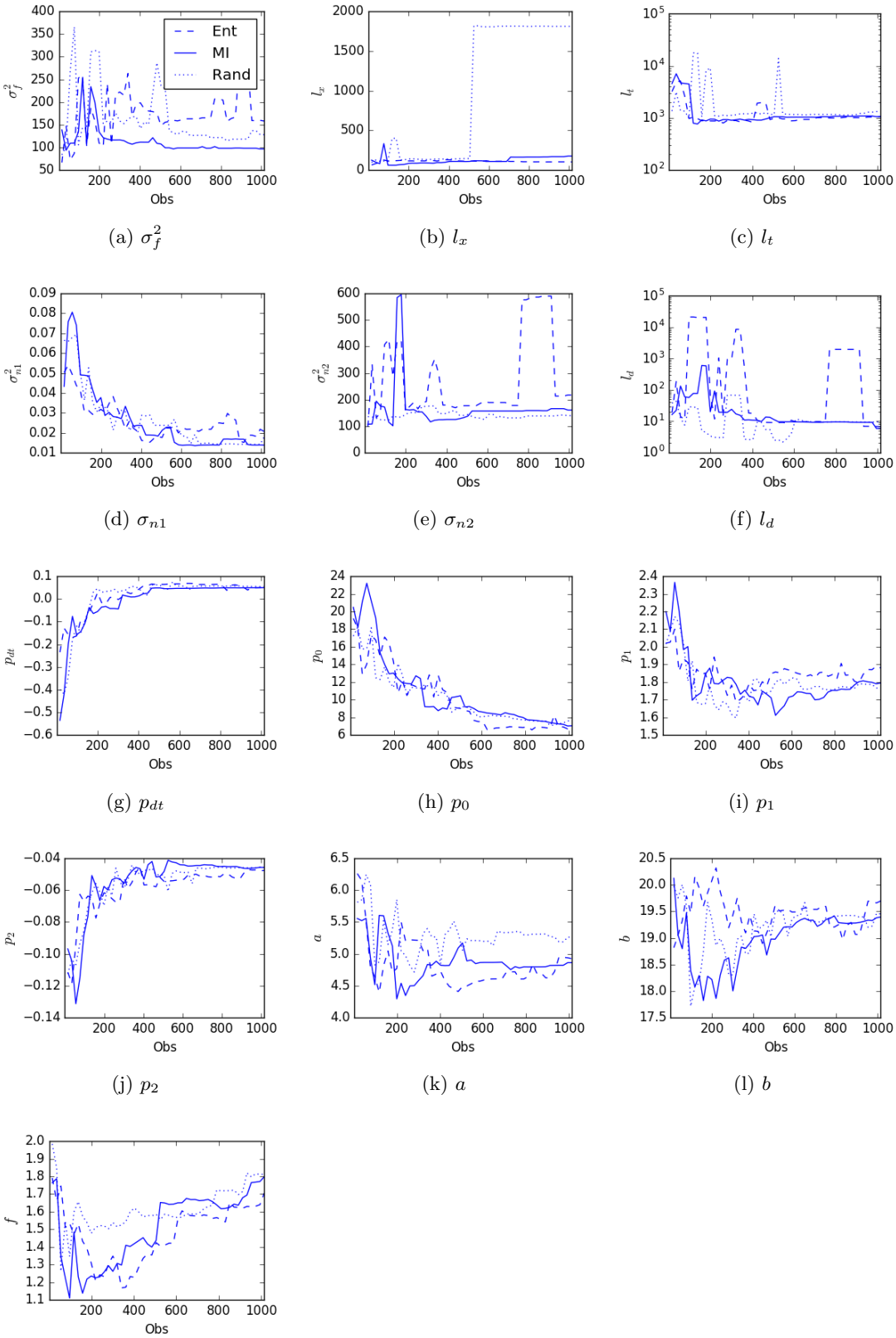
MI on the other hand relatively evenly covers the entire space, with slightly more coverage around the areas of the front. This agrees with the findings in Chapter 4 for a stationary front with sample selection.

The evolution of the mean values of the HPs for the 20 step planning horizon is shown in Figure 5.7. It can be seen that for the MI driven planning, the solved HP values tend to settle down by around 500 observations. This is not the case with the Entropy based planning which can have some large changes near the end of the period. This is due to the fact that the entropy based planning quickly focuses on sampling very near to the heteroscedastic noise. The resulting lack of samples away from the front can lead to instability in these HP values which are calculated from data points very close to the estimated center of the front. Instability in the HP estimates then leads to the large changes in model predictions which explains the jumps seen for MI based planning in the standard deviation of the RMSE and RMSSDE in figures 5.4 and 5.5.

5.6 Summary

In this chapter the work from Chapter 4 on Parametric Heteroscedastic Gaussian Processes has been extended to moving processes. A new kernel and its gradients are presented allowing fast optimisation of HPs with a gradient based solver. It has been shown how the use of a mean function in a GP can significantly increase the predictive power of the model and with the mean function sharing parameters with the kernel reduce the time required for HP optimisation. Adaptive planning is extended from sample selection to path planning on a simulated ASV and a number of planning horizons are compared in simulation of the surface salinity of a moving tidal front.

It has been shown that using MI to drive path selection produces the most accurate predictions of the first two moments of the simulated moving salinity field by sampling relatively evenly across the front, though the planning horizon must be set frequently enough, 20 steps in this case, to avoid spending resources minimising predicted MI on a model with incorrect HPs.



(a) σ_f^2 (b) l_x (c) l_t

(d) σ_{n1} (e) σ_{n2} (f) l_d

(g) p_{dt} (h) p_0 (i) p_1

(j) p_2 (k) a (l) b

(m) f

Figure 5.7: Mean HP evolution as number of observations increases under Entropy, Mutual Information and random planning for all 13 HPs from 50 Simulations for 20 step planning horizon for the logistic polynomial mean function PHGP

Chapter 6

Conclusion

The purpose of this thesis is to address problems which arise in using autonomous vehicles explore and collect data. To return the most useful data from a mission given limited time constraints these vehicles should be analysing the data they receive in real time and making planning or sampling decisions online in regards to the most informative samples to collect. Algorithms must be designed and implemented which can operate in real time with limited computational resources.

Two specific applications are considered. Firstly, this thesis examines the problem of planning for coverage to conduct bathymetric surveying in an unknown environment subject to minimum depth constraints and a bounding area. Secondly, this thesis addresses issues in planning sampling to model the surface salinity profile of a dynamic tidal front in an estuary. Whilst the focus is on Autonomous Surface Vessels, the issues which arise and solutions provided have applications to wider areas of robotic exploration and statistical modelling in general.

6.1 Summary of Contributions

6.1.1 Discrete Monotone Polygonal Partitioning

In planning for coverage, efficient methods are required to plan within a bounding region. Chapter 3 introduces Discrete Monotone Polygonal Partitioning (DMPP). This algorithm is an extension to the Boustrophedon Cellular Decomposition (BCD) [20].

Given a desired scanning path spacing and sweep direction, this algorithm allows a bounding non-convex polygon to be efficiently partitioned into cells which are monotone to that sweep direction. The resulting path length planned through these cells are shown to be shorter than that produced by the BCD. This is achieved by reducing the path overlap on adjacent polygons and planning coverage within each cell and transits between cells sequentially cell by cell.

6.1.2 Autonomous Depth Bounded Bathymetric Modelling

In Chapter 3 a suite of algorithms was presented to efficiently produce a full coverage bathymetric survey within a defined non-convex polygon whilst also subject to minimum depth constraint. With limited prior information it is shown in simulation and in the field that this implementation can in real time find and follow the intersection of a boundary and a bathymetric contour modelled as a Gaussian Process (GP). A path for coverage is planned using DMPP within the traced boundary and the batyhmetry is modelled. The Hyper Parameters (HPs) of the GP are optimised online. This is demonstrated both in simulation and on a platform built for the task in the field. Incremental Cholesky updates are implemented to allow fast fitting and prediction from the GP online as new data is collected. Analytical gradients of the Log Marginal Likelihood (LML) of the covariance kernel are used to enable fast estimation of the HPs with gradient descent solvers. Multi-threading is utilised to allow real time operation with control, planning and HP optimisation run on independent threads.

6.1.3 Parametric Heteroscedastic Gaussian Process regression

Standard GP regression assumes a noise process which is distributed uniformly across the input space. This assumption is not necessarily valid in all circumstances. It would be expected in the case of the turbulent interface between fluids of differring densities, as shown in [120] for this noise to be location dependent. Variational Heteroscedastic Gaussian Process regression addresses this problem by specifically modelling this noise as its own GP. It is shown how this method does not scale well with increasing data as the number of parameters grows with the number of observations and discuss how it is also not amenable to an incremental modelling approach as these variational parameters

are not transferable to new data points. In Chapter 4 it was proposed to model the noise as a parametric process, Parametric Heteroscedastic Gaussian Process regression (PHGP). Comparable performance was shown in terms of the first and second moments of the predicted distribution with lower computational cost and allowing incremental implementation.

6.1.4 Mean Functions in GPs

A GP is fully defined by its mean and covariance function. Generally the mean function is set at zero, sometimes explicitly, though often implicitly without discussion. In Chapter 5 it is shown that using a mean function can allow improved predictive performance. It is relatively obvious this will result in improved predictive performance with movement away from observed points as the prediction will now revert to the mean function rather than to the arbitrarily set zero value. It can also result in increasing the range of the predictive power of the observed points through increasing the length scale learnt in the covariance kernel due to the mean adjusted observations being more related to each other. The HPs of these mean functions can be learnt with the HPs of the covariance kernel and it is shown that this can actually result in faster HP optimisation. This is likely due to the simpler relationship between the observations once the mean is removed.

6.1.5 Active Sampling and Path Planning of Tidal Fronts

Covariance kernels are presented for modelling both bathymetrically arrested (Chapter 4) and moving (Chapter 5) tidal fronts. Analytical gradients to the LMLs are derived allowing efficient optimisation of the HPs. A number of information theoretic measures are implemented under these kernels for active sampling and active path planning. In Chapter 4 it is shown that Mutual Information (MI) is the optimal metric for planning for sample selection of a bathymetrically arrested tidal front modelled with PHGP to produce the model with the best predictive accuracy in terms of the first and second moments of the distribution. In Chapter 5 a moving tidal front is examined. Holonomically constrained pure random walks are generated to a given horizon and information theoretic measured used to decide the the most informative path segment. A number

of different planning horizons are also compared. To enable efficient application of MI with this extra time dimension an amendment is proposed whereby sampling locations are only considered in the spatial domain and prior observations for this purpose are considered in a space relative to their position to the estimate of the front.

6.2 Future Work

6.2.1 Field Work

The work on adaptive depth constrained bathymetric mapping in Chapter 3 was implemented on an ASV in the field. The work on tidal fronts presented in Chapters 4 and 5 was intended to study the phenomenon shown in Figure 2.4. Unfortunately it was decided that the platform which was intended to use for the purpose, a Clearpath Robotics Kingfisher ASV required redesign to allow it to be used. The vessel is only designed for use in fresh water and the technical staff decided that the connections for cables into the box were not rated highly enough to safely withstand exposure to small surface chop in saline environments. This is currently under way but the time frame for its completion did not allow it to be used for this thesis. As future work it would be interesting to implement the algorithms presented in Chapters 4 and 5 in the field.

6.2.2 Scalability

GPs in their standard from are known to suffer scalability issues. The main constraint is the size of the covariance matrix. Some measures were implemented to combat this in Chapter 3 in the form of incremental Cholesky matricies, batch prediction and multi-threaded HP optimisation. However there will still be issues as the size of the covariance matrix continues to grow. Implementing Chapters 4 and 5 in the field may encounter these issue depending on the sampling frequency of the sensor and the mission time. The solutions to this involve either sparseifying this covariance matrix [23] or reducing the number of points kept in the matrix. A simple way to do this is with a moving window [155]. This however throws away information which may be useful. A more principled approach would be to use representative points that

summarise the information in the real data such as that proposed in [59]. In Chapter 5 the most computationally intensive part of the algorithms is the optimisation of the HPs. Whilst for real time implementation this can be run in a separate thread as done in Chapter 3, and thus not impede the real time control and planning loops, it would be useful to examine methods to reduce the computational burden of this step. It would be interesting to investigate when HP optimisation should be conducted and/or how many random restarts should be used. The optimisation takes longer with more data points, but this also coincides with increased mission time when more optimisations have already been run and the HP optimisation may already be close to optimal. Local measures of HP uncertainty such as Fisher Information or analysis of the changes in HPs over the mission could potentially be used as a heuristics to control the amount of computational resources devoted to HP optimisation.

6.2.3 Hyper Parameter Uncertainty

As briefly discussed in Section 6.2.2 with reference to scalabiltiy, HP uncertainty is something to consider. In Chapter 4 the effect of using Fisher Information was analysed, which represents the local uncertainty of the HPs as a metric for active sampling. It was shown this was not useful in producing models with high predictive accuracy. An interesting avenue for future research is to combine a measure of HP uncertainty into the measure of kernel uncertainty. A linearised version of this has been used in [170] for sample selection based on the work in [57], though no analysis was done showing its performance against other measures. Some preliminary work was done for this thesis on the simulations in Chapter 4. Whilst in 1D some positive results were found, in 2D the magnitude of the effect on total uncertainty from parameter uncertainty quickly became negligible as the number of observations grew. It is suspected this may be due to the loss of information in the linearisation but further research is required.

6.2.4 Incorporating Other Sensors

The modelling of tidal fronts in Chapters 4 and 5 focused on surface salinity. It would be interesting to use more information to assist in modelling this phenomenon. For instance for a bathymetrically arrested tidal front, the surface, and subsurface mani-

festations are in part driven by the bathymetry itself. Thus sonar readings could be used to build a model of the bathymetry and use this information to help build a model of the salinity front. An Acoustic Doppler Current Profiler (ADCP) could be used to measure the currents or densities as various depths below the surface to build a 3D spatial model of the front. Finally it would also be interesting to include an Underwater Autonomous Vehicle (UAV) and jointly plan and sample the salinity on the surface and underwater.

6.2.5 Parametric Heteroscedastic Noise Functions

This thesis presented one main version of a parametric form for heteroscedastic noise. This was Gaussian weighted in one dimension. When extended to two spatial dimensions a polynomial was fit to model the mean of this Gaussian in the second spatial dimension, which was also then extended to a moving front by adding a drift term with constant velocity to this polynomial. For Longer time frames, a constant drift would not be appropriate. In the case of diurnal tides, it may be useful to model the velocity of this front with a periodic function such as a sine wave. In a different application the heteroscedastic noise associated with a point source of turbulence could be modelled by a Gaussian function in 2 spatial dimensions. The parametric noise itself may exhibit perodicity and hence be usefully modelled as a sinewave. A simple linear relation may be appropriate in some domains. For instance the linear relation between pixel intensity and noise has been used in [145] to differentiate camera models whilst [138] show a linear relation between noise in and accumulative irradiation in solar cells. Thus the framework of approximating the Heteroscedastic noise with a parametric function has wider uses than the application examined in this thesis. Use of these parametric functions if appropriately chosen produces computational efficiencies over the more general variational methods whilst also enabling the noise model to be simply applied to new data.

Appendix A

Development of a Small Autonomous Surface Vessel

This Appendix describes a small low cost Autonomous Surface Vessel (ASV) which was developed for Chapter 3 of this thesis. This craft has the advantage of being able to conduct estuarine surveys for lower cost than manual surveys and is also able to access shallower areas and with less disturbance than a larger vehicle due to reduced displacement and thrust. It has the advantage of being easy to deploy allowing rapid of testing of algorithms in real world environments. Being on the surface allows high bandwidth electronic communications to be maintained with the vessel, and access to a Global Navigation Satellite Systems (GNSS) for localisation facilitates algorithmic development and computational resources to focus on other tasks such as planning. Working on the sea-air interface it can also act as a useful bridge with underwater vehicles.

A.1 Hardware

The Autonomous Surface Vessel (ASV) is a twin hull differential thrust design. The vessel has a footprint of 1.2m x 0.85m and in its current set up weighs approximately 10kg. The twin hulls are made of medium density Polyurethane foam with a resin coating. The hulls are joined by 2 aluminium crossbars. There are 2 waterproof electronics enclosures, one mounted in each hull. One box contains the batteries, battery

Figure A.1: Autonomous Surface Vessel

controller and leak sensor. The rest of the electronics are in the other box, as can be seen in Fig. A.2. Propulsion is provided by 2 Seabotix BTD150 thrusters each able to continuously output 21.5 N of force, at 19V and drawing 4.25A. Each of these thrusters is controlled by a Pololu 18v7 simple motor controller, which can handle 24V at 7A. An ADLN2000PC from Embedded solutions which contains an Intel Atom N2600 1.6Ghz dual core processor with 2GB RAM provides computational capabilities. A VN-100 Rugged IMU from VectorNav Technologies provides attitude, velocity and acceleration data at 50Hz, and a Flex-pac G6 GPS from Novatel provides localisation capability at 1Hz. An Airmar D800 Single Beam Sonar running at 1Hz with a 12° beam width provides bathymetric sensing capability. Power is supplied via an Ocean server BBDC-02R dual battery controller connected to 2 BA-95HC 6.6Ah Li-Ion battery packs providing 6hrs of operational time. The 12 volt output is used to power the GPS, Single Beam Sonar and the thrusters (thus they are not able to reach their maximum designed output) via the motor controllers utilising pulse width modulation (PWM), with the 5V output powering the other electronics. The battery controller also relays information on battery status and temperature over a serial port to the CPU in the main box. There is a leak sensor in each box, and a temperature sensor in the main box connected to an Arduino Nano which relays this information to the main CPU. A USB Wi-Fi dongle is used at close range to interface with the CPU. Long range manual control and data transfer is provided by an XBee Pro 2.4GHz RF module with an external aerial, with a similar XBee module attached to a laptop on-shore.

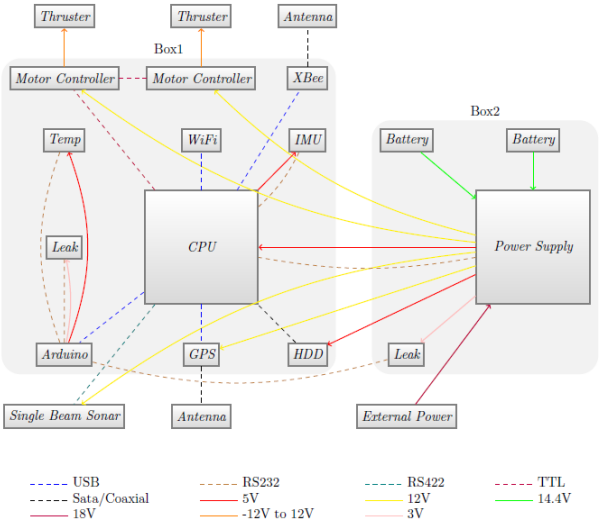


Figure A.2: ASV layout

A.2 Software

A.2.1 ROS

The framework used for the code running the ASV is Robot Operating System (ROS) Indigo [122], installed on Ubuntu(14.04). ROS inherently supports multi-process operation with a single launch file initiating the various modules called nodes. Communication between the processes is handled by both a parameter server to read/write global variables (which can be initiated from the launch file), and a publish/subscribe messaging system.

The main nodes can be seen in Fig. A.3. The Kalman filter node combines data

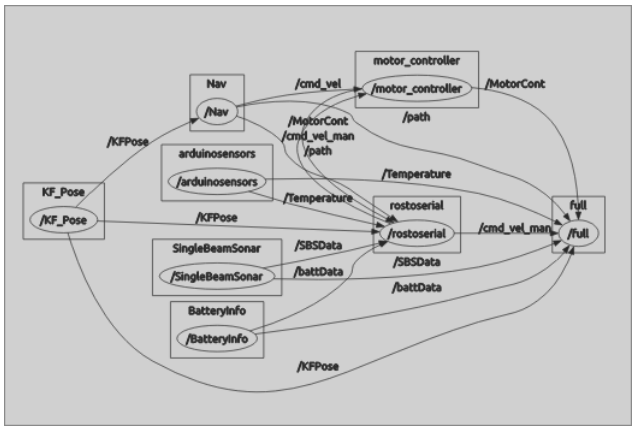


Figure A.3: ROS nodes, messages and data flow

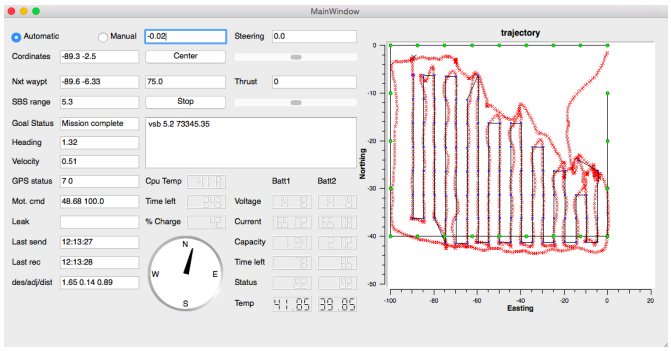


Figure A.4: Graphical User Interface on remote machine

from individual threads reading data from the IMU and the GPS and publishes a pose. There are individual nodes for publishing battery data, Sonar data and leak and internal temperature data. The Navigation module subscribes to the pose and sonar messages and reads mission settings from the parameter server. It publishes steering and thrust information for autonomous control. The RosToSerial node subscribes to all published data broadcasts a subset of it over the XBee network for monitoring purposes on the remote machine. It also listens to the XBee network, publishes a heartbeat and any manual controls given and sets a flag in the parameter server setting the motor state to manual or autonomous control. The motor controller node subscribes to both steering and thrust data published by both the navigation module for autonomous control and the RosToSerial node which publishes manual control data. It is overridden by a leak detected in either box or loss of a heartbeat over the XBee network via the parameter server. Additionally the motor controllers will switch off power to the thruster after 2 seconds of not receiving any commands.

A separate Graphical User Interface has been written in Python and QT which runs on a laptop (currently running OSX, but it is platform independent) and interfaces with the ASV over the Xbee link. This provides both information on the ASV, a map of the ASV path/way points and manual control of the ASV (see Fig. A.4 for an example). The ROS processes are launched from the laptop over the WiFi network via a tmux SSH session to ensure persistence of the processes in the case of Wi-Fi dropout. Whilst some debugging information is returned over this connection, it is not required for autonomous or manual operation of the ASV once the processes are started.

A.2.2 Heading Control and Sensor Fusion

It was determined that due to the high maneuverability of the vessel in yaw, for the purposes of this work a simple proportional controller was sufficient for heading control. This is detailed below:

$$s_d = K_p * \delta\theta/\pi \tag{A.1}$$

$$t_p = clamp(t - s_d) \tag{A.2}$$

$$t_s = clamp(t + s_d) \tag{A.3}$$

where t is percentage average thrust, t_p and t_s are percentage thrust for the port and starboard motors respectively, $\delta\theta$ is heading change in radians, s_d is the steering differential,K_p is the proportional gain coefficient and $clamp(x)$ enforces -1 $\leqslant x \leqslant$ 1.

The proportional gain coefficient was set at 6 and the average thrust level set at 75%. This results in the thrust response shown in Fig. A.5.

A Kalman filter has been implemented to fuse the data coming from the GPS and the IMU. As per [121], the GPS and IMU data are pre-processed internally on their respective units and then this data is fed to Kalman filter to fuse it together. The GPS data arrives at 1Hz, the IMU data at 50hz. With the update rate of the IMU being significantly higher than the GPS, we have followed the methodology of [121]

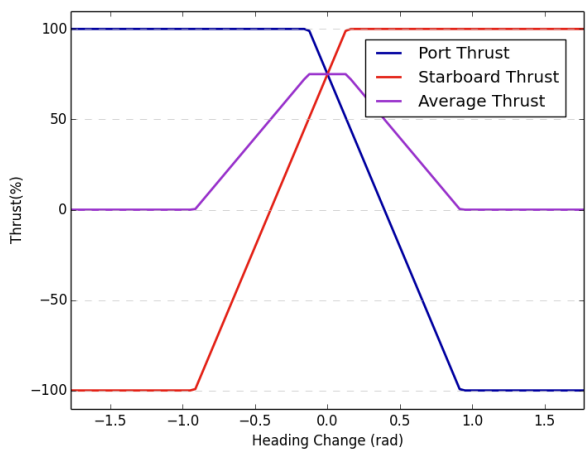


Figure A.5: Controller thrust response, $K_p = 6, t = 75\%$

in updating on the time cycles of the IMU, rather than interpolating between these when a GPS measurement arrives. A constant acceleration model has been chosen for the Kalman filter. As such we are estimating position, velocity and acceleration in 3 dimensions, leading to 9 states, $\hat{X} = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z})^T$.

The Novatel GPS unit publishes variance numbers in real time, which are used as input to the Kalman filter. The VectorNav IMU does not. The variance of this unit has been estimated experimentally at $\hat{\sigma_i^2} = 0.2m/s^2$.

Whenever we wish an estimate of the state of the Kalman filter $\hat{X}$ we run the prediction step. When a new IMU data point arrives we first run the prediction then the update step. When a new GPS data point arrives we wait until the next IMU data point arrives and run the prediction and updates steps for both data at once (using R_{ig}, Z_{ig} and H_{ig}). In addition to the state of the filer we can also return the current uncertainty of this state from the variable P. The variance of the process is σ_q^2. The equations below are based on those contained in [35, 165].

Prediction Step:

$$\hat{X}_t^- = F_t \hat{X}_{t-1}^+$$
$$P_t^- = F_t P_{t-1}^+ F_t^T + Q_t$$

Update Step:

$$K_t = P_t^- H^T (H P_t^- H^T + R)^{-1}$$
$$\hat{X}_t^+ = \hat{X}_{t-1}^- + K(Z - (H\hat{X}_t^-))$$
$$P_t^+ = (I - KH)P_{t-1}^-$$

where:

$$F = \begin{pmatrix} F_0 & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & F_0 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & F_0 \end{pmatrix} \qquad F_0 = \begin{pmatrix} 1 & dt & dt^2 \\ 0 & 1 & dt \\ 0 & 0 & 1 \end{pmatrix}$$

$$Q = \begin{pmatrix} Q_0 & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & Q_0 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & Q_0 \end{pmatrix} \sigma_q^2 \qquad Q_0 = \begin{pmatrix} dt^5/20 & dt^4/8 & dt^3/6 \\ dt^4/8 & dt^3/3 & dt^2/2 \\ dt^3/6 & dt^2/2 & dt \end{pmatrix}$$

$$H_i = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad H_{ig} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_i = I_{3x3}\sigma_i \qquad R_{ig} = diag(\sigma_{gx}^2, \hat{\sigma_i^2}, \sigma_{gy}^2, \hat{\sigma_i^2}, \sigma_{gz}^2, \hat{\sigma_i^2})$$

$$Z_i = (\ddot{x}, \ddot{y}, \ddot{z})^T \qquad Z_{ig} = (\dot{x}, \ddot{x}, \dot{y}, \ddot{y}, \dot{z}, \ddot{z})^T$$

Appendix B

Hyper Parameter Optimisation

This Appendix presents some information relevant to HPs in as applied in this thesis. The prior distributions used and bounding constraints used when solving the HPs with gradient descent are also discussed. The analytical derivatives of the LML w.r.t the HPs for the heteroscedastic models in Chapters 4 and 5 are presented. These allow efficient gradient descent algorithms to be implemented.

B.1 Priors

Hyper Parameter (HP) optimisation requires starting values. Choosing these as distributions rather than scalars helps the optimisation cover the potential parameter space and avoid converging on local minimums. Whilst these prior distributions are purely for the starting values for the gradient descent optimiser, and thus the results are not highly sensitive to choices here, starting values in the correct order of magnitude to the results will result in faster convergence whilst local minimums in the LML surface could result in some solutions not being found if the entire range of possible values i snot covered. Gradient based solvers such as L_BFGS_B [167] also allow hard bounds to be added. If limits can be set on the bounds of any HPs, implementing this prior knowledge will increase convergence speed by restricting the optimiser from searching outside these areas. This can also be a useful way to stop unrealistic solutions which may occur early in a mission with a small amount of data. Care should be taken to analyse the resulting HP evolution. If any of the HPs are converging hard on these lim-

HP	Distribution	Parameters	bounds
σ_f^2	exponential	100	$[1e^{-1}, 1e^4]$
l_x	exponential	range(X)	$[1e^{-1}, \infty]$
l_t	exponential	max(T)	$[1e^{-1}, \infty]$
σ_{n1}^2	exponential	0.01	$[1e^{-8}, 0.25]$
σ_{n2}^2	exponential	100	$[1e^{-8}, 1e^4]$
l_d	exponential	range(X)/5	$[1e^{-8}, \infty]$
p_{dt}	gaussian	[0,0.5]	$[-1, 1]$
p_0	exponential	min(X1)+range(X1)/4	$[1e^{-8}, 75]$
p_1	gaussian	[2,1]	$[1e^{-8}, 5]$
p_2	exponential.(-1)	[1]	$[-0.5, -1e^{-4}]$
a^*	gaussian	[min(Y),1]	$[1e^{-8}, 30]$
b^*	gaussian	[range(Y),1]	$[1e^{-8}, 30]$
f^*	exponential	1	$[1e^{-8}, 100]$

Table B.1: Hyper Parameter prior distributions. Exponential distribution have 1 scale parameter, gaussian distribution has 2 parameters mean and variance. The starting value for m_2 is drawn from an exponential distribution and then multiplied by -1 to convert it to a negative bounded value. * not used for Zero mean PHGP.

its, especially as the number of observations increases, this could indicate these bounds have been set too tightly.

The HP prior distributions and boundary values chosen in Chapter 5 and shown in Table B.1 will now be justified. To keep the number of parameters down the single parameter exponential distribution is chosen for most HP prior distributions. The scale parameter of the exponential distribution equals the mean of this distribution as well as controlling the dispersion. As there is no separate control of the variance of this distribution and it is bounded between 0 and ∞ this does not work for all the parameters which may require negative values or mean values away from zero with a tight dispersion around this value. As such for a number of HPs, a Gaussian distribution which is defined by a mean and variance is used, and for p_2 the negative value of the exponential distribution is taken to gain a single parameter distribution of negative values. Kernel uncertainty is controlled by σ_f^2, which also controls the uncertainty predicted as the distance between the prediction point and the observations, divided by the relevant length scales (l_x, l_t), increases. The uncertainty from the heteroscedastic noise process is controlled by σ_{n2}^2. The mean and bounds of these have been set based on observation of prior experiments. The lower bound of σ_f^2 is higher than that of σ_{n2}^2 as a value close to zero for σ_f^2 would indicate the kernel has no predictive power, whereas for σ_{n2}^2, it would indicate the model sees no heteroscedastic noise, which should be allowed. The characteristic length scales

in the spatial and time domains are set by l_x and l_t. The mean values are set at the maximum distance that could be seen between points. The upper bound as infinity reflects no bound in the optimiser as large values in either of these indicate fitting a flat plane through the data. The lower bounds have been set at $1e-1$ based on physical bounds of the sensing platform. If the process being sensed requires length scales smaller than this then it would not be possible to model it accurately with a platform moving at 1m/s and sensing at 1Hz. The characteristic length scale of the heteroscedastic noise is represented by l_d. This has been given a lower mean value at 20% of the range in the spatial scale as it is expected the noise process will be more localised. Noise which is evenly distributed across the input space is represented by σ_{n1}^2. Physically this could manifest itself though inaccuracies in the physical sensor and localisation. This is upper bounded at 0.25 which equates to a standard deviation of 0.5 on measurements of an underlying process which can range from 0 to 30 g/Kg of salt for fresh to ocean water. Bounding this value relatively low forces the GP to model the data not just as noise which can sometime be a problem with a small number of observations. p_{dt} controls the velocity of the tidal front and its parameters are set in ranges expected to be seen from physical observations. Again if it were outside the bounds given the ASV would not be able to follow it. The position of the polynomial is determined by p_0, p_1 and p_2, and the parameters are again set by the expectations of possible curves which could be seen. Bounding p_2 negative causes the curve to be bent with the center further to the right than the top and bottom as would be expected on an incoming tide for a class II tidal front where the tide is flowing from left to right. The minimum value and range of the logistic mean function, a and b, and are bounded by the physical water properties and the scale parameter is set by the minimum value observed. The slope of the curve, f, is bounded both positive to reflect the expectation of the fresh water being upstream to the right of the sea water as the experiment is designed. The upper bound and mean were set based on prior experiments. It can be seen from Figure 5.7 that the bounds chosen were not constraining the average HP values in that case.

B.2 Analytical Derivatives of Log Marginal Likelihood

The derivatives for the LML w.r.t the HPs in the GP models in Chapters 4 and 5 can be calculated analytically. Whilst it can result in a large number of parameters, they are relatively straight forward to derive and the results are presented here with minimal working for completeness.

The Log Marginal Likelihood (LML) of the observations $\mathbf{y}$ given the sampling locations X and HPs θ is:

$$\log p(\mathbf{y}|X,\theta) = -\frac{1}{2}\mathbf{y_{adj}}^T K_R^{-1}\mathbf{y_{adj}} - \frac{1}{2}\log|K_R| - \frac{n}{2}\log 2\pi \tag{B.1}$$

The partial derivatives of this LML are:

$$\frac{\partial}{\partial\theta_i}\log p(\mathbf{y}|X,\theta) = \frac{1}{2}\mathrm{tr}\left((\alpha\alpha^T - K_R^{-1})\frac{\partial K_R}{\partial\theta_i}\right) \tag{B.2}$$

where $K_R = K + R$, $\mathbf{y_{adj}} = \mathbf{y} - \mathbf{m}$ and $\alpha = K^{-1}\mathbf{y_{adj}}$.

For each model there is a given mean, kernel and noise function, leading to a different set of partial derivatives. The squared exponential function is used for the kernel in both cases:

$$K(X, X_*) = \sigma_f^2 \exp\left(-\frac{1}{2}||X, X_*||^T\Sigma^{-1}||X, X_*||\right) \tag{B.3}$$

where Σ is a diagonal matrix populated by the square of the characteristic length scale on each input dimension.

B.2.1 Stationary Logistic Polynomial Heteroscedastic Noise Gaussian Process with Zero Mean

The analytical derivatives of the LML w.r.t the HPs for the heteroscedastic GP model in Chapter 4 are now presented. As this model has a zero mean function $\mathbf{y_{adj}} = \mathbf{y}$. The heteroscedastic noise function R is defined by the following equations:

$$R = \sigma_{n1}^2 I + \sigma_{n2}^2 \text{diag}(D(X)) \tag{B.4}$$

$$D(\mathbf{x_1}, \mathbf{x_2}) = \frac{1}{l_d\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{\mathbf{x_1} - \mathbf{p}}{l_d}\right)^2\right) \tag{B.5}$$

where $\mathbf{p} = p_0 + p_1\mathbf{x_2} + p_2\mathbf{x_2}^2$. The Kernel in this case becomes:

$$K_* = \sigma_f^2 \exp\left(-\frac{1}{2}||X, X_*||^T (l_x^2 I)^{-1} ||X, X_*||\right) \tag{B.6}$$

where the same length scale l_x is used for both spatial dimensions. The following HPs need to be optimised:

$$\theta = [\sigma_f^2, l_x, \sigma_{n1}^2, \sigma_{n2}^2, l_d^2, p_0, p_1, p_2] \tag{B.7}$$

This leads to the following partial derivatives of the kernel + noise function K_R, which have been simplified where possible:

$$\frac{\partial K_R}{\partial \sigma_f^2} = \exp\left(-\frac{1}{2}||X, X_*||^T (l_x^2 I)^{-1} ||X, X_*||\right) = \frac{K}{\sigma_f^2}$$

$$\frac{\partial K_R}{\partial l_x} = \frac{(\mathbf{x_1} - \mathbf{x_{1*}})^2 + (\mathbf{x_2} - \mathbf{x_{2*}})^2}{l_x^3} K$$

$$\frac{\partial K_R}{\partial \sigma_{n1}^2} = I \qquad \frac{\partial K_R}{\partial \sigma_{n2}^2} = \text{diag}(D(\mathbf{x_1}, \mathbf{x_2}))$$

$$\frac{\partial K_R}{\partial l_d^2} = \text{diag}\left(\frac{\sigma_{n2}^2\left((\mathbf{x_1} - \mathbf{p})^2 - l_d^2\right)}{2l_d^4} D(\mathbf{x_1}, \mathbf{x_2})\right)$$

$$\frac{\partial K_R}{\partial p_0} = \frac{\sigma_{n2}^2(\mathbf{x_1} - \mathbf{p})}{l_d^2} \text{diag}(D(\mathbf{x_1}, \mathbf{x_2}))$$

$$\frac{\partial K_R}{\partial p_{dt}} = \mathbf{t}\frac{\partial K_R}{\partial p_0} \quad \frac{\partial K_R}{\partial p_1} = \mathbf{x_2}\frac{\partial K_R}{\partial p_0} \quad \frac{\partial K_R}{\partial p_2} = \mathbf{x_2^2}\frac{\partial K_R}{\partial p_0}$$

B.2.2 Drifting Logistic Polynomial Heteroscedastic Noise Gaussian Process with Parametric Mean

The analytical derivatives of the LML w.r.t the HPs for the heteroscedastic GP model in Chapter 5 are now presented. The two main differences with the model in Chapter 4 are the introduction of a mean function which share parameters with the noise function, and the addition of the time domain. This time domain both appears in the Kernel and in the noise and mean functions though the drift term which controls the location of the front in the time dimension. The heteroscedastic noise function now becomes:

$$D(\mathbf{x_1}, \mathbf{x_2}, \mathbf{t}) = \frac{1}{l_d\sqrt{2\pi}} exp\left(-\frac{1}{2}\left(\frac{\mathbf{x_1} - \mathbf{p}}{l_d}\right)^2\right) \tag{B.8}$$

where $\mathbf{p} = p_{dt}\mathbf{t} + p_0 + p_1\mathbf{x_2} + p_2\mathbf{x_2}^2$.

The mean function is defined as:

$$\mathbf{m} = \left(a + \frac{b}{(1 + \mathbf{e}^{-f(\mathbf{p}-\mathbf{x_1})})}\right) \tag{B.9}$$

where a, b and f define the minimum value, range and slope of the logistic function. The Kernel in this case becomes:

$$K_* = \sigma_f^2 \exp\left(-\frac{1}{2}||X, X_*||^T \Sigma^{-1} ||X, X_*||\right) \tag{B.10}$$

where $\Sigma = \text{diag}(l_x^2, l_x^2, l_t^2)$. Here the length scales on the spatial domain are controlled by one parameter l_x, and the time domain by a second length scale l_t. The following HPs need to be optimised:

$$\theta = [\sigma_f^2, l_x, l_t, \sigma_{n1}^2, \sigma_{n2}^2, l_d^2, a, b, f, p_{dt}, p_0, p_1, p_2] \tag{B.11}$$

This leads to the following partial derivatives of the kernel + noise function K_R, which

have been simplified where possible:

$$\frac{\partial K_R}{\partial \sigma_f^2} = \exp\left(-\frac{1}{2}||X, X_*||^T \Sigma^{-1} ||X, X_*||\right) = \frac{K}{\sigma_f^2}$$

$$\frac{\partial K_R}{\partial l_x} = \frac{(\mathbf{x_1} - \mathbf{x_{1*}})^2 + (\mathbf{x_2} - \mathbf{x_{2*}})^2}{l_x^3} K$$

$$\frac{\partial K_R}{\partial l_t} = \frac{(\mathbf{t} - \mathbf{t}_*)^2}{l_t^3} K$$

$$\frac{\partial K_R}{\partial \sigma_{n1}^2} = I$$

$$\frac{\partial K_R}{\partial \sigma_{n2}^2} = \mathrm{diag}(D(\mathbf{x_1}, \mathbf{x_2}, \mathbf{t}))$$

$$\frac{\partial K_R}{\partial l_d^2} = \mathrm{diag}\left(\frac{\sigma_{n2}^2 \left((\mathbf{x_1} - \mathbf{p})^2 - l_d^2\right)}{2 l_d^4} D(\mathbf{x_1}, \mathbf{x_2}, \mathbf{t})\right)$$

$$\frac{\partial K_R}{\partial p_0} = -\frac{\sigma_{n2}^2 (\mathbf{p} - \mathbf{x_1})}{l_d^2} \mathrm{diag}(D(\mathbf{x_1}, \mathbf{x_2}, \mathbf{t}))$$

$$\frac{\partial K_R}{\partial p_{dt}} = \mathbf{t} \frac{\partial K_R}{\partial p_0} \qquad \frac{\partial K_R}{\partial p_1} = \mathbf{x_2} \frac{\partial K_R}{\partial p_0} \qquad \frac{\partial K_R}{\partial p_2} = \mathbf{x_2^2} \frac{\partial K_R}{\partial p_0}$$

$$\frac{\partial K_R}{\partial a} = 0 \qquad \frac{\partial K_R}{\partial b} = 0 \qquad \frac{\partial K_R}{\partial f} = 0$$

$$\frac{\partial \mathbf{y_{adj}}}{\partial a} = -\mathbf{1} \qquad \frac{\partial \mathbf{y_{adj}}}{\partial b} = -\frac{1}{1 + \exp(-f(\mathbf{p} - \mathbf{x}))}$$

$$\frac{\partial \mathbf{y_{adj}}}{\partial f} = -\frac{b(\mathbf{p} - \mathbf{x}) \exp(-f(\mathbf{p} - \mathbf{x})}{(1 + \exp(-f(\mathbf{p} - \mathbf{x})))^2}$$

$$\frac{\partial \mathbf{y_{adj}}}{\partial p_0} = -\frac{(\mathbf{p}-\mathbf{x})\exp(-f(\mathbf{p}-\mathbf{x})}{(1+\exp(-f(\mathbf{p}-\mathbf{x})))^2} = f\frac{\partial \mathbf{y_{adj}}}{\partial f}$$

$$\frac{\partial \mathbf{y_{adj}}}{\partial p_{dt}} = f\mathbf{t}\frac{\partial \mathbf{y_{adj}}}{\partial f} \qquad \frac{\partial \mathbf{y_{adj}}}{\partial p_1} = f\mathbf{x_2}\frac{\partial \mathbf{y_{adj}}}{\partial f} \qquad \frac{\partial \mathbf{y_{adj}}}{\partial p_2} = f\mathbf{x_2^2}\frac{\partial \mathbf{y_{adj}}}{\partial f}$$

$$\frac{\partial \mathbf{y_{adj}}}{\partial \sigma_f^2} = 0 \quad \frac{\partial \mathbf{y_{adj}}}{\partial l_x} = 0 \quad \frac{\partial \mathbf{y_{adj}}}{\partial l_t} = 0 \quad \frac{\partial \mathbf{y_{adj}}}{\partial \sigma_{n1}^2} = 0 \quad \frac{\partial \mathbf{y_{adj}}}{\partial \sigma_{n2}^2} = 0 \quad \frac{\partial \mathbf{y_{adj}}}{\partial l_d^2} = 0$$

Appendix C

Films and Animations

A number of animated figures and videos have been referenced in the text. These are listed here for convenience.

- A short video of the ASV conducting the field trial
  `https://youtu.be/YH2nymgKXws`
- An animation of the data behind Figure 3.7 of the GP predictions and path planning whilst conducting the field trial for autonomous bathymetric mapping
  `https://youtu.be/G88L7FATtKQ`
- A timelaspe film of a bathymetrically arrested tidal front at Lilli Pilli Point, Port Hacking river, Sydney, Australia
  `https://youtu.be/id1YWLujGX8`
- An animation of the data behind Figure 4.5 showing the evolution of the different information metrics as more data is randomly sampled from the 1D bathymetrically arrested salinity front
  `https://youtu.be/zR4abnICCjM`
- An animation an examples simulation of the evolution of the predicted standard deviation under sample selection driven by different information metrics, similar to Figure 4.7
  `https://youtu.be/lwyaeyZOR-M`
- An animation of Figure 5.1 showing the impact of including a mean function in

GPs on their predictive power

https://youtu.be/VHouvMOa9Cs

- An animation of the simulated dynamic tidal front used in Chapter 5

  https://youtu.be/MiPZ8Aa7oR4

- An animation of an example simulation of the path taken under Entropy vs Mutual Information. Similar to Figure 5.6, however also showing the moving front and the predicted standard deviation of the model

  https://youtu.be/8YvU-QkCdVo

Bibliography

[1] E. U. Acar and H. Choset. Robust sensor-based coverage of unstructured environments. In *International Conference on Intelligent Robots and Systems*, volume 1, pages 61–68. IEEE, 2001.

[2] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull. Morse decompositions for coverage tasks. *The International Journal of Robotics Research*, 21(4):331–344, 2002.

[3] V. J. Aidala. Kalman filter behavior in bearings-only tracking applications. *Transactions on Aerospace and Electronic Systems*, AES-15(1):29–39, 1979.

[4] E. M. Arkin, M. Held, and C. L. Smith. Optimization problems related to zigzag pocket machining. *Algorithmica*, 26(2):197–236, 2000.

[5] D. Ashlock and J. Freeman. A pure finite state baseline for Tartarus. In *Congress on Evolutionary Computation*, volume 2, pages 1223–1230. IEEE, 2000.

[6] A. Baranes, P.-Y. Oudeyer, P.-y. O. Inria, and F. January. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73, January 2013.

[7] G. Batchelor. *The theory of homogenous turbulence*. Cambridge University Press, 1953.

[8] H. Baumann, R. B. Wallace, T. Tagliaferri, and C. J. Gobler. Large Natural pH, CO2 and O2 Fluctuations in a Temperate Tidal Salt Marsh on Diel, Seasonal, and Interannual Time Scales. *Estuaries and Coasts*, pages 220–231, 2014.

[9] A. Bender, S. B. Williams, O. Pizarro, and M. V. Jakuba. Adaptive exploration of benthic habitats using gaussian processes. In *OCEANS*, pages 1–10. IEEE, 2010.

[10] A. Bender, S. B. Williams, and O. Pizarro. Autonomous exploration of large-scale benthic environments. In *International Conference on Robotics and Automation*, 2013.

[11] J. Binney, A. Krause, and G. S. Sukhatme. Informative path planning for an autonomous underwater vehicle. In *International Conference on Robotics and Automation*, pages 4791–4796. IEEE, 2010.

[12] F. A. Bombardelli, M. I. Cantero, M. H. Garcia, and G. C. Buscaglia. Numerical aspects of the simulation of discontinuous saline underflows: the lock-exchange problem. *Journal of Hydraulic Research*, 47(6):777–789, 2009.

[13] P. Boyle and M. Frean. Dependent gaussian processes. *In Advances in Neural Information Processing Systems 17*, pages 217–224, 2005.

[14] M. B. Brown and A. B. Forsythe. Robust tests for the equality of variances. *Journal of the American Statistical Association*, 69(346):364–367, 1974.

[15] L. C. Bruce, P. L. M. Cook, I. Teakle, and M. R. Hipsey. Hydrodynamic controls on oxygen dynamics in a riverine salt wedge estuary, the Yarra River estuary, Australia. *Hydrology and Earth System Sciences*, 18(4):1397–1411, 2014.

[16] D. M. Burrage. Evolution and dynamics of tropical river plumes in the Great Barrier Reef: An integrated remote sensing and in situ study. *Journal of Geophysical Research*, 107(C12), 2002.

[17] J. A. Caley and G. A. Hollinger. Data-driven comparison of spatio-temporal monitoring techniques. In *OCEANS*, pages 1–7. IEEE, 2015.

[18] R. Castano, T. Estlin, R. C. Anderson, D. M. Gaines, and C. Chouinard. OASIS : Onboard Autonomous Science Investigation System for Opportunistic Rover Science. *Journal of Field Robotics*, 24(5):379–397, 2007.

[19] C. F. Cerco and T. Cole. Three-dimensional eutrophication model of Chesapeake Bay. *Journal of Environmental Enginineering*, 119(6):1006–1025, 1993.

[20] H. Choset and P. Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Field and Service Robotics*, pages 203–209. Springer, 1998.

[21] C. M. Clark, K. Hancke, A. Xydes, K. Hall, F. Schreiber, J. Klemme, J. Lehr, and M. Moline. Estimation of volumetric oxygen concentration in a marine environment with an autonomous underwater vehicle. *Journal of Field Robotics*, 30(1):1–16, 2013.

[22] N. Cressie and H.-C. Huang. Classes of nonseparable, spatio-temporal stationary covariance functions. *Journal of the American Statistical Association*, 94(448):1330–1339, 1999.

[23] L. Csató and M. Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.

[24] G. Cuccu and F. Gomez. When novelty is not enough. In *Applications of Evolutionary Computation*, pages 234–243. Springer, 2011.

[25] D. Damian, P. D. Sampson, and P. Guttorp. Bayesian estimation of semi-parametric non-stationary spatial covariance structures. *Environmetrics*, 12(2):161–178, 2001.

[26] N. Darbeheshti and W. E. Featherstone. Non-stationary covariance function modelling in 2D least-squares collocation. *Journal of Geodesy*, 83(6):495–508, 2009.

[27] J. R. Davis and K. Koop. Eutrophication in Australian rivers, reservoirs and estuaries A southern hemisphere perspective on the science and its implications. *Hydrobiologia*, 559(1):23–76, apr 2006.

[28] R. Dechter and J. Pearl. Generalized best-first search strategies and the optimality of A. *Journal of the ACM*, 32(3):505–536, 1985.

[29] R. J. Diaz. Overview of hypoxia around the world. *Journal of Environmental Quality*, 30(2):275–81, 2001.

[30] P. Dierckx. Algorithms for smoothing data with periodic and parametric splines. *Computer Graphics and Image Processing*, 20(2):171–184, 1982.

[31] D. S. dos Santos, C. L. Nascimento, and W. C. Cunha. Autonomous navigation of a small boat using IMU/GPS/digital compass integration. In *International Systems Conference*, pages 468–474. IEEE, 2013.

[32] M. Dunbabin, A. Grinham, and J. Udy. An autonomous surface vehicle for water quality monitoring. In *Australasian Conference on Robotics and Automation*, pages 2–4, 2009.

[33] N. Durrande, D. Ginsbourger, O. Roustant, and L. Carraro. ANOVA kernels and RKHS of zero mean functions for model-based sensitivity analysis. *Journal of Multivariate Analysis*, 115(0):57–67, 2013.

[34] N. Durrande, D. Ginsbourger, and O. Roustant. Additive covariance kernels for high-dimensional gaussian process modeling. *stat*, 27:1050, 2011.

[35] H. Durrant-Whyte. Introduction to estimation and the Kalman filter. *Australian Centre for Field Robotics*, 2001.

[36] P. Dutilleul and P. Legendre. Spatial heterogeneity against heteroscedasticity: an ecological paradigm versus a statistical concept. *Oikos*, pages 152–171, 1993.

[37] K. R. Dyer. *Estuaries: A physical introduction.* John Wiley and Sons, 1973.

[38] R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007, 1982.

[39] N. Fairfield and D. Wettergreen. Active localization on the ocean floor with multibeam sonar. In *Oceans*. IEEE, 2008.

[40] H. Ferreira, R. Martins, E. Marques, J. Pinto, A. Martins, J. Almeida, J. Sousa, and E. Silva. Swordfish: An autonomous surface vehicle for network centric operations. In *Oceans-Europe*, pages 1–6. IEEE, 2007.

[41] G. Ferri, M. V. Jakuba, and D. R. Yoerger. A novel method for hydrothermal vents prospecting using an autonomous underwater robot. In *International Conference on Robotics and Automation*, pages 1055–1060. IEEE, 2008.

[42] E. Frazzoli, M. A. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. In *American Control Conference*, volume 1, pages 43–49. IEEE, 2001.

[43] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013.

[44] A. García-olaya, F. Py, J. Das, and K. Rajan. An online utility-based approach for sampling dynamic ocean fields. *Journal of oceanic engineering*, 37(2): 185–203, 2012.

[45] S. Garg, A. Singh, and F. Ramos. Learning non-stationary space-time models for environmental monitoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2012.

[46] P. M. Gilbert, S. Seitzinger, C. A. Heil, J. M. Burkholder, M. W. Parrow, L. A. Codispoti, and V. Kelly. The Role of Eutrophication in the Global Proliferation of Harmful Algal Blooms. *Oceanography*, 18(2):198–209, 2005.

[47] Y. Girdhar, P. Giguere, and G. Dudek. Autonomous adaptive underwater exploration using online topic modeling. In *International Symposium on Experimental Robotics*, pages 1–14, 2012.

[48] J. Godfrey and J. Parslow. Description and preliminary theory of circulation in Port Hacking estuary. Technical report, CSIRO, 1976.

[49] P. W. Goldberg, C. K. Williams, and C. M. Bishop. Regression with input-dependent noise: A gaussian process treatment. *Advances in Neural Information Processing Systems*, 10:493–499, 1997.

[50] K. Goodwin, N. Caraco, and J. Cole. Temporal dynamics of dissolved oxygen in a floating-leaved macrophyte bed. *Freshwater Biology*, 53(8):1632–1641, 2008.

[51] A. Gotovos, N. Casati, G. Hitz, and A. Krause. Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence*, 2013.

[52] R. L. Graham, D. E. Knuth, O. Patashnik, and S. Liu. Concrete mathematics: A foundation for computer science. *Computers in Physics*, 3(5):106–107, 1989.

[53] M. Grasmueck, G. P. Eberli, D. A. Viggiano, T. Correa, G. Rathwell, and J. Luo. Autonomous underwater vehicle (AUV) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the Straits of Florida. *Geophysical Research Letters*, 33(23), 2006.

[54] G. M. Guerra, S. Zio, J. J. Camata, J. Dias, R. N. Elias, M. Mattoso, P. L. Paraizo, A. L. Coutinho, and F. A. Rochinha. Uncertainty quantification in numerical simulation of particle-laden flows. *Computational Geosciences*, 20(1): 265–281, 2016.

[55] D. V. Hansen and M. Rattray. New dimensions in estuary classification. *Limnology and Oceanography*, 11(3):319–326, 1966.

[56] C. Härtel, E. Meiburg, and F. Necker. Analysis and direct numerical simulation of the flow at a gravity-current head. Part 1. Flow topology and front speed for slip and no-slip boundaries. *Journal of Fluid Mechanics*, 418:189–212, 2000.

[57] D. A. Harville. Decomposition of prediction error. *Journal of the American Statistical Association*, 80(389):132–138, 1985.

[58] P. Hennig. Introduction to GPs II. `https://youtu.be/C9ZnpFG69KE`. University of Sheffield, 2014.

[59] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

[60] R. Hine, S. Willcox, G. Hine, and T. Richardson. The wave glider: A wave-powered autonomous marine vehicle. In *Oceans*, pages 1–6. IEEE, 2009.

[61] G. Hitz, F. Pomerleau, M.-E. Garneau, C. Pradalier, T. Posch, J. Pernthaler, and R. Y. Siegwart. Autonomous inland water monitoring: Design and application of a surface vessel. *Robotics & Automation Magazine*, 19(1):62–72, 2012.

[62] G. Hitz, F. Pomerleau, M.-E. Garneau, C. Pradalier, T. Posch, J. Pernthaler, and R. Y. Siegwart. Autonomous inland water monitoring: Design and application of a surface vessel. *Robotics & Automation Magazine*, 19(1):62–72, 2012.

[63] G. A. Hollinger and G. S. Sukhatme. Sampling-based motion planning for robotic information gathering. In *Robotics: Science and Systems*, volume 3, page 4. Citeseer, 2013.

[64] G. A. Hollinger, U. Mitra, and G. S. Sukhatme. Active and adaptive dive planning for dense bathymetric mapping. In *Experimental Robotics*, pages 803–817. Springer, 2013.

[65] R. Hudd. Springtime episodic acidification as a regulatory factor of estuary spawning fish recruitment. Technical report, Finnish Game and Fisheries Research Institute, 2000.

[66] L. M. Huzzey. The dynamics of a bathymetrically arrested estuarine front. *Estuarine, Coastal and Shelf Science*, 15(5):537–552, 1982.

[67] M. Johnson-Roberson, O. Pizarro, S. B. Williams, and I. Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51, 2010.

[68] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[69] I. Kamon and E. Rivlin. Sensory-based motion planning with global proofs. *Transactions on Robotics and Automation*, 13(6):814–822, 1997.

[70] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller. Anytime motion planning using the rrt. In *International Conference on Robotics and Automation*, pages 1478–1483. IEEE, 2011.

[71] T. Kärnä, A. M. Baptista, J. E. Lopez, P. J. Turner, C. McNeil, and T. B. Sanford. Numerical modeling of circulation in high-energy estuaries: A Columbia River estuary benchmark. *Ocean Modelling*, 88:54–71, 2015.

[72] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard. Most likely heteroscedastic gaussian process regression. In *International conference on Machine learning*, pages 393–400. ACM, 2007.

[73] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, Feb 2008.

[74] J. J. Kuffner and S. M. LaValle. Rrt-connect. an efficient approach to single-query path planning. In *International Conference on Robotics and Automation*, volume 2, pages 995–1001. IEEE, 2000.

[75] H. Kurniawati, D. Hsu, and W. S. Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and systems*, volume 2008. Zurich, Switzerland., 2008.

[76] J. L. Largier. Tidal intrusion fronts. *Estuaries*, 15(1):26–39, 1992.

[77] J.-C. Latombe. *Robot motion planning*, volume 124. Springer Science & Business Media, 2012.

[78] S. M. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University, Computer Science Department, 1998.

[79] N. R. Lawrance and S. Sukkarieh. Path planning for autonomous soaring flight in dynamic wind fields. In *International Conference on Robotics and Automation*, pages 2499–2505. IEEE, 2011.

[80] N. R. J. Lawrance. *Autonomous soaring flight for unmanned aerial vehicles.* PhD thesis, The University of Sydney, 2011.

[81] M. Lazaro-Gredilla and M. Titsias. Variational heteroscedastic gaussian process regression. In *International Conference on Machine Learning*, pages 841–848. ACM, 2011.

[82] O. Le Maître and O. M. Knio. *Spectral methods for uncertainty quantification: With applications to computational fluid dynamics.* Springer Science & Business Media, 2010.

[83] E. R. Levine, L. Goodman, and J. O'Donnell. Turbulence in coastal fronts near the mouths of Block Island and Long Island Sounds. *Journal of Marine Systems*, 78(3):476–488, 2009.

[84] B. R. Light and J. Beardall. Distribution and spatial variation of benthic microalgal biomass in a temperate, shallow-water marine system. *Aquatic Botany*, 61(1):39–54, 1998.

[85] D. G. MacDonald, L. Goodman, and R. D. Hetland. Turbulent dissipation in a near-field river plume: A comparison of control volume and microstructure observations with a numerical model. *Journal of Geophysical Research: Oceans*, 112(7):1–13, 2007.

[86] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-whyte. An experiment in integrated exploration. In *International Conference on Intelligent Robots and Systems, 2002.*, pages 534–539, 2002.

[87] A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-whyte. Information based adaptive robotic exploration. In *International Conference on Intelligent Robots and Systems, 2002.*, pages 540–545, 2002.

[88] J. E. Manley, A. Marsh, W. Cornforth, and C. Wiseman. Evolution of the autonomous surface craft AutoCat. In *Oceans*, volume 1, pages 403–408. IEEE, 2000.

[89] R. Marchant and F. Ramos. Bayesian optimisation for intelligent environmental monitoring. In *International Conference on Intelligent Robots and Systems*, pages 2242–2249. IEEE, 2012.

[90] R. Marchant, F. Ramos, S. Sanner, et al. Sequential bayesian optimisation for spatial-temporal monitoring. In *UAI*, pages 553–562, 2014.

[91] K. V. Mardia and R. Marshall. Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, 71(1):135–146, 1984.

[92] M. Martínez, A. Intralawan, G. Vázquez, O. Pérez-Maqueo, P. Sutton, and R. Landgrave. The coasts of our world: Ecological, economic and social importance. *Ecological Economics*, 63(2):254–272, 2007.

[93] J. Mathieu and J. Scott. *An introduction to turbulent flow.* Cambridge University Press, 2000.

[94] A. Matos and N. Cruz. Positioning control of an underactuated surface vessel. In *Oceans*, pages 1–5. IEEE, 2008.

[95] R. M. McCabe, B. M. Hickey, and P. MacCready. Observational estimates of entrainment and vertical salt flux in the interior of a spreading river plume. *Journal of Geophysical Research: Oceans*, 113(C8), 2008.

[96] T. McClimans. Estuarine fronts and river plumes. In *Physical processes in estuaries*, pages 55–69. Springer, 1988.

[97] C. McGann, F. Py, K. Rajan, J. Ryan, and R. Henthorn. Adaptive control for autonomous underwater vehicles. *National conference of Assoc. for the Advancement of Artificial Intelligence*, 2008.

[98] A. Melkumyan and F. Ramos. A sparse covariance function for exact gaussian process inference in large datasets. *International Joint Conference on Artificial Intelligence*, 9:1936–1942, 2009.

[99] A. M. Moore, H. G. Arango, G. Broquet, B. S. Powell, A. T. Weaver, and J. Zavala-Garay. The Regional Ocean Modeling System (ROMS) 4-dimensional variational data assimilation systems: Part I–System overview and formulation. *Progress in Oceanography*, 91(1):34–49, 2011.

[100] I. Murray and R. P. Adams. Slice sampling covariance hyperparameters of latent gaussian models. In *Advances in Neural Information Processing Systems*, pages 1732–1740, 2010.

[101] R. Murray-Smith and A. Girard. Gaussian process priors with ARMA noise models. In *Irish Signals and Systems Conference, Maynooth*, pages 147–152, 2001.

[102] W. Naeem, R. Sutton, and T. Xu. An integrated multi-sensor data fusion algorithm and autopilot implementation in an uninhabited surface craft. *Ocean Engineering*, 39:43–52, 2012.

[103] H. Nakhost and M. Müller. Monte-Carlo exploration for deterministic planning. In *International Joint Conference on Articifical Intelligence*, volume 9, pages 1766–1771, 2009.

[104] J. Nguyen, N. Lawrance, R. Fitch, and S. Sukkarieh. Energy-constrained motion planning for information gathering with autonomous aerial soaring. In *International Conference on Robotics and Automation*, pages 3825–3831. IEEE, 2013.

[105] D. Nguyen-Tuong, M. Seeger, and J. Peters. Model learning with local gaussian process regression. *Advanced Robotics*, 23(15):2015–2034, 2009.

[106] S. OBoyle, G. McDermott, and R. Wilkes. Dissolved oxygen levels in estuarine and coastal waters around ireland. *Marine pollution bulletin*, 58(11):1657–1663, 2009.

[107] S. O'Callaghan, F. T. Ramos, and H. Durrant-whyte. Contextual occupancy maps using gaussian processes. *International Conference on Robotics and Automation*, pages 1054–1060, 2009.

[108] J. ODonnell, G. O. Marmorino, and C. L. Trump. Convergence and downwelling at a river plume front. *Journal of Physical Oceanography*, 28(7): 1481–1495, 1998.

[109] J. O'Donnell, S. G. Ackleson, and E. R. Levine. On the spatial scales of a river plume. *Journal of Geophysical Research*, 113(C4):1–12, 2008.

[110] T. Oksanen and A. Visala. Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26(8):651–668, 2009.

[111] S. Ong and S. Png. Pomdps for robotic tasks with mixed observability. In *Robotics: Science and Systems*, 2009.

[112] M. Osborne, S. Roberts, A. Rogers, S. Ramchurn, and N. Jennings. Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes. In *International Conference on Information Processing in Sensor Networks*, pages 109–120. IEEE, 2008.

[113] C. J. Paciorek and M. J. Schervish. Nonstationary covariance functions for gaussian process regression. *Advances in neural information processing systems*, 16:273—-280, 2004.

[114] A. Pascoal, P. Oliveira, C. Silvestre, L. Sebastião, M. Rufino, V. Barroso, J. Gomes, N. Seube, J. Champeau, P. Dhaussy, V. Sauce, and R. Moitié. Robotic ocean vehicles for marine science applications : The european ASIMOV project. In *Oceans*. IEEE, 2000.

[115] A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme. Risk-aware path planning for autonomous underwater vehicles using predictive ocean models. *Journal of Field Robotics*, 30(5):741–762, 2013.

[116] T. Pfingsten, M. Kuss, and C. Rasmussen. Nonstationary gaussian process regression using a latent extension of the input space. In *Extended Abstract in Proc. of ISBA World Meeting on Bayesian Statistics*, January 2006.

[117] J. Pineau, G. Gordon, S. Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, pages 1025–1032, 2003.

[118] C. Plagemann, K. Kersting, and W. Burgard. Nonstationary gaussian process regression using point estimates of local smoothness. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5212 LNAI(PART 2):204–219, 2008.

[119] P. A. Plonski, P. Tokekar, and V. Isler. Energy-efficient path planning for solar-powered mobile robots. *Journal of Field Robotics*, 30(4):583–601, 2013.

[120] G. Poëtte, B. Després, and D. Lucor. Treatment of uncertain material interfaces in compressible flows. *Computer Methods in Applied Mechanics and Engineering*, 200(1):284–308, 2011.

[121] H. Qi and J. B. Moore. Direct Kalman filtering approach for GPS/INS integration. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2): 687–693, 2002.

[122] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: an open-source robot operating system. In *International Conference on Robotics and Automation workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[123] J. Quinonero-Candela, C. E. Rasmussen, and C. K. Williams. Approximation methods for gaussian process regression. *Large-scale kernel machines*, pages 203–224, 2007.

[124] G. Rabideau, B. Engelhardt, and S. Chien. Using generic preferences to incrementally improve plan quality. In *AIPS*, pages 236–245, 2000.

[125] M. Rahimi, R. Pon, W. J. Kaiser, G. S. Sukhatme, D. Estrin, and M. Srivastava. Adaptive sampling for environmental robotics. In *International Conference on Robotics and Automation*, volume 4, pages 3537–3544. IEEE, 2004.

[126] F. Ramos and L. Ott. Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent. *The International Journal of Robotics Research*, 35(14):1717–1730, 2016.

[127] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.

[128] P. Rigby and S. B. Williams. Adaptive sensing for localisation of an autonomous underwater vehicle. In *Australasian Conference on Robotics and Automation*, 2005.

[129] P. Rigby, O. Pizarro, and S. B. Williams. Toward adaptive benthic habitat mapping using gaussian process classification. *Journal of Field Robotics*, 27(6): 741–758, 2010.

[130] M. Rodrigues, G. Jesus, J. Rogeiro, A. Fortunato, J. Gomes, and A. Oliveira. RealQual: Towards real-time high-resolution monitoring and prediction of water quality in estuaries and coastal areas-Final report. Technical report, Laboratrio NacioNaL de eNgeNharia CiviL, 2015.

[131] A. C. Ross, R. G. Najjar, M. Li, M. E. Mann, S. E. Ford, and B. Katz. Sea-level rise and other influences on decadal-scale salinity variability in a coastal plain estuary. *Estuarine, Coastal and Shelf Science*, 157:79–92, 2015.

[132] Saildrone. Saildrone. `www.saildrone.com`. 2016. Accessed: 2016-01-18.

[133] J. Sattar, G. Dudek, O. Chiu, I. Rekleitis, P. Gigu, A. Mills, N. Plamondon, C. Prahacs, Y. Girdhar, M. Nahon, and J.-p. Lobos. Enabling autonomous capabilities in underwater robotics. In *International Conference on Intelligent Robots and Systems*, pages 3628—-3634, 2008.

[134] A. M. Schmidt and A. O'Hagan. Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 65(3):743–758, 2003.

[135] M. Seeger. Low rank updates for the Cholesky decomposition. Technical report, University of California, Berkeley, 2007.

[136] C. M. Senet, J. Seemann, S. Flampouris, and F. Ziemer. Determination of bathymetric and current maps by the method DiSC based on the analysis of nautical X-band radar image sequences of the sea surface. *Transactions on Geoscience and Remote Sensing*, 46(8):2267–2279, 2008.

[137] J. H. Sharp. Estuarine oxygen dynamics : What can we learn about hypoxia from long-time records in the Delaware Estuary ? *Limnology and Oceanography*, 55(2):535–548, 2010.

[138] Z. Sheng and R. Liang. Heteroscedastic linear model based reliability evaluation for solar cell degradation testing. In *Industrial Engineering and Engineering Management*, pages 890–893. IEEE, 2016.

[139] J. Simpson and R. Nunes. The tidal intrusion front: An estuarine convergence zone. *Estuarine, Coastal and Shelf Science*, 13(3):257IN1–266, 1981.

[140] A. Singh, F. Ramos, H. D. Whyte, and W. J. Kaiser. Modeling and decision making in spatio-temporal processes for environmental surveillance. In *International Conference on Robotics and Automation*, pages 5490–5497. IEEE, 2010.

[141] R. N. Smith, P. P. Li, D. a. Caron, B. H. Jones, and G. 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, August 2010.

[142] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer, 1990.

[143] T. Smith and R. Simmons. Point-based pomdp algorithms: Improved analysis and implementation. *arXiv preprint arXiv:1207.1412*, 2012.

[144] J. R. Souza, R. Marchant, L. Ott, D. F. Wolf, and F. Ramos. Bayesian optimisation for active perception and smooth navigation. In *International Conference on Robotics and Automation*, pages 4081–4087. IEEE, 2014.

[145] T. H. Thai, R. Cogranne, and F. Retraint. Camera model identification based on the heteroscedastic noise model. *Transactions on Image Processing*, 23(1): 250–263, 2014.

[146] D. R. Thompson and D. Wettergreen. Intelligent maps for autonomous kilometer-scale science survey. In *International Symposium on Artificial Intelligence Robotics and AUtomation in Space*, pages 1–8, feb 2008.

[147] S. Thrun. Monte carlo pomdps. In *Advances in neural information processing systems*, pages 1064–1070, 2000.

[148] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

[149] P. Tokekar, D. Bhadauria, A. Studenski, and V. Isler. A robotic system for monitoring carp in Minnesota lakes. *Journal of Field Robotics*, 27(6):779–789, 2010.

[150] V. Tresp. Mixtures of gaussian processes. *NIPS*, page 654=660, 2000.

[151] R. M. Tyler and T. E. Targett. Juvenile weakfish Cynoscion regalis distribution in relation to diel-cycling dissolved oxygen in an estuarine tributary. *Marine Ecology Progress Series*, 333(August 2015):257–269, 2007.

[152] R. M. Tyler, D. C. Brady, and T. E. Targett. Temporal and spatial dynamics of diel-cycling hypoxia in estuarine tributaries. *Estuaries and Coasts*, 32(1): 123–145, 2009.

[153] R. J. Uncles. Estuarine physical processes research: Some recent studies and progress. *Estuarine, Coastal and Shelf Science*, 55(6):829–856, 2002.

[154] S. Vasudevan. Data fusion with gaussian processes. *Robotics and Autonomous Systems*, 60(12):1528–1544, 2012.

[155] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte. Gaussian process modeling of large-scale terrain. *Journal of Field Robotics*, 26(10): 812–840, 2009.

[156] S. Vijayakumar, A. D'souza, and S. Schaal. Incremental online learning in high dimensions. *Neural computation*, 17(12):2602–2634, 2005.

[157] R. West and J. West. Cabbage Tree basin: Natural values and options for management. *Unpublished report for the Hacking Catchment Management Committee*, 79, 2000.

[158] P. Whaite and F. P. Ferrie. Autonomous exploration : Driven by uncertainty. *Transactions on Pattern Analysis and Machine Intelligence*, 19(3):193–205, 1997.

[159] T. K. Wilding, E. Brown, and K. J. Collier. Identifying dissolved oxygen variability and stress in tidal freshwater streams of northern New Zealand. *Environmental Monitoring and Assessment*, 184(10):6045–6060, 2012.

[160] T. Wilson and S. B. Williams. Active sample selection in scalar fields exhibiting non-stationary noise with parametric heteroscedastic gaussian process regression. In *International Conference on Robotics and Automation*. IEEE, 2017. To appear.

[161] T. Wilson and S. B. Williams. Adaptive path planning for depth-constrained bathymetric mapping with an autonomous surface vessel. *Journal of Field Robotics*, 00:1–14, 2017.

[162] Y. Xu and J. Choi. Adaptive sampling for learning gaussian processes using mobile sensor networks. *Sensors*, 11(3):3051–3066, 2011.

[163] B. Yamauchi. A frontier-based approach for autonomous exploration. In *International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151, Monterey, California, USA, 1997. IEEE Comput. Soc. Press.

[164] K. Yang, S. Keat Gan, and S. Sukkarieh. A gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV. *Advanced Robotics*, 27(6):431–443, 2013.

[165] P. Zarchan and H. Musoff. Fundamentals of Kalman filtering: A practical approach. *Progress in Astronautics and Aeronautics, AIAA, Virginia*, 232, 2009.

[166] Y. Zhang and A. M. Baptista. SELFE: a semi-implicit Eulerian–Lagrangian finite-element model for cross-scale ocean circulation. *Ocean modelling*, 21(3): 71–96, 2008.

[167] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[168] Z. Zhu and M. L. Stein. Spatial sampling design for parameter estimation of the covariance function. *Journal of Statistical Planning and Inference*, 134(2): 583–603, 2005.

[169] Z. Zhu and M. L. Stein. Spatial sampling design for prediction with estimated parameters. *Journal of Agricultural, Biological, and Environmental Statistics*, 11(1):24–44, March 2006.

[170] D. L. Zimmerman. Optimal network design for spatial prediction, covariance parameter estimation, and empirical prediction. *Environmetrics*, 17(6):635–652, 2006.