Prospection for Mobile Robots in Unknown Environments

by

Ki Myung Brian Lee

A thesis submitted in partial fulfilment of the
requirements for the degree of Doctor of Philosophy

at the

School of Mechanical and Mechatronic Engineering
Faculty of Engineering and Information Technology
University of Technology Sydney

January 2023

Certificate of Original Authorship

I, Ki Myung Brian Lee, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in this thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signed: Production Note:
Signature removed prior to publication.

Date: 24/12/2022

Prospection for Mobile Robots in Unknown Environments

by

Ki Myung Brian Lee

Abstract

Modern applications require robots to operate with a greater degree of autonomy in dynamic and unknown environments. This problem of robot autonomy has been traditionally posed as that of improving adaptivity to support wider ranges of environments or inputs. We argue that adaptivity alone is insufficient for realising robots that are truly autonomous, because the resulting behaviour is inherently limited to be merely reactive to external stimuli.

This thesis proposes *robotic prospection* as a new framework for autonomy that enables proactive behaviours. Prospection is a concept from cognitive psychology that refers to the ability to generate and evaluate possible future scenarios and outcomes. We mathematically formulate the notion of robotic prospection from a Bayesian perspective, and define the problems of prospective perception and planning. The prospective perception problem asks to design predictive priors that expedite Bayesian estimation for environmental perception. The prospective planning problem aims to design a strategy for choosing control actions that maximise a task reward, by, for example, balancing exploration and exploitation.

Under this formulation, we contribute a suite of algorithmic tools for prospective perception and planning in both general and specialised problem settings. In the most general problem setting, we present the mutual information upper confidence bound (MI-UCB) strategy for prospective planning. The MI-UCB is an exploration-exploitation trade-off strategy that approximately maximises the posterior expected reward using information-theoretic quantities. The generality of the MI-UCB also allows scaling prospective planning to a heterogeneous multi-robot system consisting of scout- and task-robots, thereby distributing exploration and exploitation workloads. We apply the MI-UCB to a scout-task robot team performing target search, and demonstrate that it outperforms the conventional expected reward maximisation strategy.

We then demonstrate prospective planning and perception in tandem in the special case of environmental process monitoring. The goal is for the robot to seek the source of an environmental process under the influence of an ambient flow field based on sparse concentration measurements. As a predictive prior for prospective perception, we propose advection-diffusion Gaussian process (ADGP) by augmenting Gaussian process (GP) regression with the advection-diffusion partial differential equation (PDE) from fluid dynamics. Based on the ADGP, actions are chosen to balance exploration and exploitation using the Gaussian process upper confidence bound (GP-UCB) strategy. The results demonstrate intelligent behaviours arise from prospection, such as prioritising down- or cross-stream regions if the concentration measurement is low.

Subsequently, we consider prospective planning for complex semantic tasks specified using formal methods. To enable prospection, we propose random signal temporal logic (RSTL) to capture the uncertain semantics of the environment. Planning is achieved with gradient-based synthesis, and we present examples involving uncertain targets and obstacles. The examples demonstrate benefits of prospection, such as prioritising targets that are increasingly uncertain over those that are proximate, and reducing the uncertainty of targets that are currently relevant to the task.

Finally, we conduct case studies to identify desirable properties of predictive priors for prospection. We consider two problem settings: oceanic and indoor navigation. In both cases, we derive predictive priors by imposing relevant PDE constraints on GP regression for physical fidelity. Both case studies demonstrate that planning performance can be significantly improved through prospective perception. Based on these examples, we conclude that generative models serve as suitable predictive priors for prospective perception.

The contribution of this thesis represents a significant leap in progress toward robot autonomy. The proposed perspective of robotic prospection extends autonomy beyond traditional adaptivity, and unifies several previous formulations that partially address the problem of proactivity. This theoretical perspective is supported by the algorithmic tools presented in this thesis, which allow implementation in practical robotics application. We hope that the formulations and solutions presented in this thesis serve as a methodical recipe for robot autonomy in the future.

Acknowledgements

I would firstly like to thank my supervisory panel, Prof Robert Fitch, Prof Shoudong Huang, and Dr Chanyeol Yoo. Standing on Rob's shoulders I saw what a vision looks like. He also taught me how to show others a vision. Shoudong offered me clarity of thought when I lacked it. Chanyeol taught me to write from the reader's view, no matter how much of his energy it took.

I had the privilege to work with many industry partners without whose practical insight much of this work would not have been conceived or completed. To name a few, I would like to thank Dr Simon Ng, Dr Jennifer Palmer, Ricacardo Cannizzaro from DST Melbourne, Dr Jijoong Kim, Dr Carmine Pontecorvo and Rhett Hull from DST Adelaide, Dr David Johnson, Dr David Battle, Dave Jennings, Aspen Eyers, and Will Talbot from Mission Systems, Christian Bruwer, Katrin Schmid, and Rajanish Calisa from UTS Rapido, and the folks from DST Sydney and Blue Ocean Monitoring.

I'd like to thank my previous supervisors, Dr Xiaofeng Wu, Prof Ian Manchester, and Dr David Johnson. Xiaofeng gave me the 'red pill' for research. Ian taught me the technical foundations on which this thesis is built. David taught me the practice of field robotics.

I am grateful to have worked with many wonderful co-authors and colleagues, who have been endless sources of ideas and inspirations. I am fortunate to have co-authored with (in a rough reverse-temporal order): Jen Wakulicz, Dr Felix Kong, Dr James Lee, Cadmus To, Lan Wu, A/Prof Teresa Vidal-Calleja, Dr Wolfram Martens, Jayant Khatkar, Prof Ramgopal Mettu, Dr Yongbo Chen, and many more. Thank you all for tolerating my tendency to pull last-minute crunches. Thank you, Bang, Michael, and Eddy, for your hard work on hardware and the banter. Thank you, members of CDMRG past and present, including Fred, Gio, Stefan, Tim, Jesse, Mitchell, Clyde, Vera, Anthony, Mahdi, and many more, for preserving my sanity (to some degree, though not to their fault). Thank you, Dr Graeme Best and Dr Oliver Cliff, for introducing me to this group in the first place.

Finally, I thank my family and friends for all their support, both emotional and material. 엄마, 아빠, 기정, 할머니, 할아버지, relatives, and friends from Seoul, Brisbane, and Sydney (you know who you are). I love you all!

Contents

List of Figures

List of Tables

List of Acronyms

ADGP advection-diffusion Gaussian process vi, 71, 72, 82, 85, 136

AI artificial intelligence 15, 139

AUV autonomous underwater vehicle 12

CGF cumulant generating function 53, 56, 60

CI conditionally independent xiii, 91–95, 99, 100, 159

CRF conditional random field 23, 24

Dec-MCTS Decentralised Monte Carlo tree search xiii, 58–60, 66

DESPOT determinised sparse partially observable tree 27

EDF Euclidean distance field 125–129, 131–133, 136

EKF extended Kalman filter 25

EM expectation-maximisation 24, 116–118

ESDF Euclidean signed distance field 22

EXP3 exponential weight algorithm for exploration and exploitation 56

FMT fast marching tree 71, 78, 80, 81, 85

GAN generative adversarial network 23

GP Gaussian process vi, xiii, xiv, 10, 11, 17, 22–24, 28–30, 36, 37, 40, 41, 45, 71, 75–79, 84, 85, 96, 112–121, 125–130, 133, 134, 136, 138

GP-UCB Gaussian process upper confidence bound vi, x, 10, 30, 46, 71, 77–82, 84, 85, 136

GPIS Gaussian process implicit surface 96, 126, 127, 129–133, 136

GPS global positioning system 13, 14, 25, 107–110, 114, 115, 117, 119, 121, 122, 146–148, 154

GPU graphics processing unit 159

GT ground truth 131

HIL hardware in the loop 151

ISR intelligence, surveillance and reconnaissance xiii, 69, 145

KL Kullback-Leibler 42, 53, 54

KL-UCB Kullback-Leibler upper confidence bound (UCB) 30

LHS left-hand side 53

LTL linear temporal logic 16, 33

MAB multi-armed bandit 29

MAP maximum a posteriori 98, 99, 104, 109, 110

MC Monte Carlo xiii, 94, 95

MCTS Monte Carlo tree search xiii, 27–29, 55, 56, 59, 98

MDP Markov decision process 26, 33, 54

ME mutually exclusive xiii, 94, 95, 99, 159, 160

MI-UCB mutual information upper confidence bound v, xii, xiii, 10, 17, 46, 51–57, 59, 60, 62, 63, 65–68, 97, 135, 136, 138, 140–142

MILP mixed integer linear programming 16, 34

MOMDP mixed observability Markov decision process 27, 46–48, 138

MOOS Mission-Oriented Operating Suite 149

MORSE Modular Open Robotics Simulation Engine 151

MRF Markov random field 24

OCEAN online current estimation and navigation 107, 121, 133

PDE partial differential equation vi, x, 10, 11, 17, 41, 71, 73–75, 82, 85, 129, 130, 133, 136, 138

PDF probability distribution function 97

PHD probability hypothesis density 23, 24, 33

POMCP partially observable Monte Carlo planning 27

POMDP partially observable Markov decision process 26, 27, 29, 34, 35, 51, 52, 66, 138

QGC QGroundControl 149, 150

RF radio frequency 23

RFS random finite set 24, 33, 35

RHS right-hand side 53, 54

ROS Robot Operating System 147–150

RRT rapidly exploring random tree 28, 55

RSSI received signal strength indicator 23

RSTL random signal temporal logic vi, xiii, 11, 16, 17, 89–95, 99–104, 136, 139, 159

SDF signed distance field 22, 132

SE squared exponential 75, 76

SLAM simultaneous localisation and mapping 32, 147, 149, 151, 152, 154

SMC sequential Monte Carlo 23, 24

SPN sum-product network 23, 45, 140

STL signal temporal logic 11, 16, 17, 33, 36, 87, 89, 93, 102, 104, 136, 139

SVM support vector machine 23, 146

TSDF truncated signed distance field 22, 131, 132

UART universal asynchronous receive-transmit 149

UAV uninhabited aerial vehicle xiv, 28, 32, 145, 153

UCB upper confidence bound xviii, xx, 27, 30, 52–54, 56, 82

UCT UCB on trees 27, 30, 56

UGV uninhabited ground vehicle 32

UT unscented transform 23

Chapter 1

Introduction

Greater autonomy is being asked of robots than ever before. Applications such as cobots [1] and self-driving cars [2] increasingly demand robots to 'just work' in dynamic and unknown environments. This new generation of robots must be *autonomous*, and must operate without intervention given only high-level commands, by interrogating its surroundings with limited onboard sensors. We are interested in developing algorithmic tools that enable such autonomy in robots that they seem to 'just work'.

The current developments in robot autonomy have focused on enhancing adaptivity to the environment. This has been achieved through the use of the *separation principle*, which views perception and planning as separate and independent modules. In this view, perception and planning may regard each other as black boxes. The approach is not without merits. It has been beneficial thus far, as it permits fast, modular development of perception and planning algorithms with acceptable system performance overall.

However, we assert that the adaptivity approach will fall short of the requirements of modern applications, as it inherently produces behaviours that are merely reactive. Because planning treats perception as a black box, even non-myopic planners are rendered myopic[1], in the sense that it does not account for the evolution of uncertainty over time, or the resolution thereof that results from future measurements. Similarly, as perception

[1]Here, our notion of "myopic" extends the traditional ones. Although non-myopic planners are capable of selecting actions that prove optimal in a long time horizon, they are still myopic because the changes in environmental uncertainty are typically unaccounted for.

is unaware of planning, the processing of sensory information cannot discern what information in particular is relevant to the task at hand. These limitations are fundamental in nature, and will be increasingly conspicuous in modern robotics applications.

This thesis introduces *robotic prospection* to address such limitations of the current paradigm. Prospection is a concept from cognitive psychology [3] that refers to the prediction and evaluation of future scenarios against hedonic motives in humans and animals. In other words, prospection equips an agent with foresight on possible future environmental states and outcomes. We argue that prospection is necessary for achieving the next frontier of robot autonomy. The information afforded by foresight and future-oriented decision-making will be of greater value as robots start to operate in increasingly dynamic and uncertain environments,

This thesis presents a mathematical formulation and recipe for robotic prospection from a Bayesian perspective, as well as a suite of algorithmic tools that enable prospection in practical robotics application. We demonstrate the utility of prospection across many application domains, including ocean monitoring, indoor navigation, semantic planning, and multi-robot systems.

In the rest of this chapter, we establish the context and direction of this thesis, and introduce relevant concepts. We first discuss what constitutes robot autonomy, and argue that adaptivity is only a part of autonomy. We then review the current paradigm of separation principle, and highlight its limitations in meeting the modern demands. To address these limitations, we devise a recipe for robotic prospection based on insights from cognitive psychology. Finally, we lay out the specific objectives of this thesis, identify relevant applications, and briefly summarise the main contributions and results.

1.1 Autonomy Beyond Adaptivity

Significant progress has been made towards developing autonomous robots. Robots are no longer constrained to safety barricades in factories and conveyor belts. They are now pervasive across all economic sectors: producing resources and food [4] in the primary

sector, manufacturing products [5] in the secondary, providing daily transport [2] in the tertiary, and exploring the unknown [6] in the quaternary.

These developments have been guided by the vision of robot autonomy. An autonomous robot 'just works', without human intervention. However, one could argue the previous generation of automated robots also operate without human intervention to some degree. After all, automation, by definition, implies that the robot operates automatically. What, then, distinguishes autonomy from automation?

A popular distinction is that an autonomous robot can adapt under uncertainty, by sensing the surrounding environment and deliberating its actions. In contrast, an automated robot only knows how to act, without deliberation. It is therefore 'blind', and requires an operator to prescribe precisely what needs to be done, and any changes in the environment will lead to undesirable behaviours.

Let us consider some examples to scrutinise this definition. Manufacturing robots such as the one shown in Fig. 1.1a are considered a prominent example of automated robots. The practitioner precisely pre-programs their motion, and obviate any deviations by installing them behind a fence. Although they have contributed significant cost and lead-time reduction [5], their prospects seem to have hit a limit. Electric car manufacturer Tesla, for example, found that over-automation is costly, because some processes, such as the final assembly of automobiles, are still better suited for humans than robots [11].

Collaborative robots, or cobots [1], is an emerging alternative paradigm that addresses the shortcoming of automated manufacturing robots. A cobot refers to a robot that interacts with a human worker to collaboratively complete tasks, as shown in Fig. 1.1b. Doing so synergistically combines the strengths of robot and human. Cobots must necessarily be autonomous, because safety barricades no longer exist, and they must account for humans that work in proximity. Therefore, cobots can no longer rely on pre-programming and control of the environment, and the algorithms for a cobot must account for changes and uncertainties in the environment.

Similarly, a self-driving car [2] is a clear example of an autonomous system. These systems encapsulate well the spirit of robots that 'just work': the passenger simply inputs the

(A) [7]

(B) [8]

(C) [9]

(D) [10]

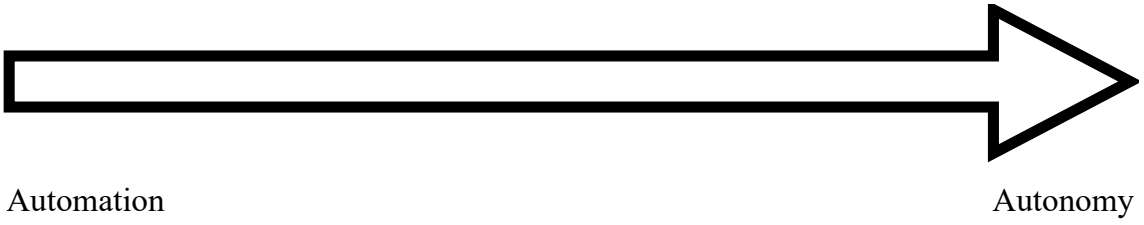


FIGURE 1.1: The spectrum from automation (a, c) to autonomy (b, d).

destination, to which the car drives itself, while abiding by the traffic rules. These self-driving cars are intended for everyday people, not those highly trained, and must therefore require minimal or no pre-programming at all. To this end, Google Firefly [10] completely removed the interface for human control, as can be seen in Fig. 1.1d.

A perhaps less obvious example is the case of aircraft autopilots such as the one shown in Fig. 1.1c. One could argue aircraft autopilots are automated in the sense that they are programmed to do specific tasks well, such as cruising and landing. Meanwhile, high-level decisions are deferred to highly trained human pilots, who command the aircraft through numerous controls shown in Fig. 1.1c. Moreover, the surrounding environment

is judiciously controlled by the control tower operators, unlike self-driving cars. However, one may argue that aircraft autopilots are autonomous in the sense that they adapt to the surroundings, for example, the wind gust during flight.

The examples of existing systems illustrate that, under the adaptivity definition, automation and autonomy lie on a spectrum. A robot may be considered 'more autonomous' than automated if it can adapt to a wider range of environments, or a greater variety of stimuli. In fact, one may even argue that the factory robots shown in Fig. 1.1a are autonomous in that they sense their joint angles and apply voltage accordingly. Consequently, in this perspective, to improve robot autonomy is to incrementally enhance adaptivity.

A natural question to ask, then, is if we can eventually develop a robot that is truly autonomous in the literal sense of 'just working', by considering all possible stimuli. We believe this is not the case. While adaptivity is an important part of autonomy, much greater capabilities are required to realise a truly autonomous robot. Adaptivity implies reactivity; the system passively reacts to changes in the environment to produce actions. Therefore, an adaptive robot's operation remains dependent on external entities.

In contrast, we envision that an autonomous robot should be proactive. It must predict ahead what the environment looks like or will look like, and identify what it needs to know about the environment in order to achieve its objectives. For example, a truly autonomous cobot would identify the intentions of its human co-worker without explicit specification. Similarly, a self-driving car must predict the surrounding traffic to operate reliably. In the rest of this chapter, we discuss how such proactivity can be implemented.

1.2 Rethinking the Separation Principle

The question of robot autonomy has been traditionally posed as that of improving adaptivity to uncertainty. To this end, a prominent approach is to use the *separation principle* [12], an idea borrowed from control theory. In a narrow technical sense, the separation principle states that, for linear systems, one can design an optimal system as a whole, by separately designing 1) an optimal state estimator (i.e., a Kalman filter) that produces an estimate of the system state from measurements; and 2) an optimal state-feedback controller (i.e.,

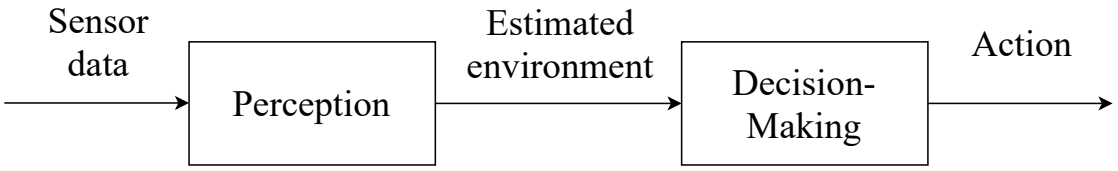


FIGURE 1.2: A typical perception-action pipeline under the separation principle.

a linear-quadratic regulator) that computes optimal actions given the system state estimate [12, 13]. In a broader sense in robotic software architecture, the separation principle is extended as a basis for a dichotomy between perception and decision making modules, as illustrated in Fig. 1.2. Sensor measurements from the environment are processed by a perception module to produce an estimate of environment and the robot itself. The estimated environment is then consumed by a planning module to produce the action. This pattern is ubiquitous at all levels of robotic software, from low-level control and collision avoidance to high-level task and motion planning. The widespread adoption of the separation principle is perhaps owing to the encapsulation of perception and decision-making algorithms in independent modules, which allows modular re-use of different algorithms in different applications.

However, the separation principle appears to be the barrier hindering a higher level of autonomy. In particular, the separation principle is starting to prove ineffective as robots are starting to operate in increasingly uncertain and adverse environments. Most fundamentally, good decisions require good situational awareness of the environment. This, in turn, demands good sensory data that contains useful information about the environment, despite significant efforts and progress toward robustness by the perception research community. To gather good sensory data, the decision-making algorithm must account for the quality of potential measurements that may result from the robot's actions.

An autonomous being does not passively process information. Instead, it actively seeks relevant information that helps achieve its task. However, the separation principle fails to capture the quality of measurements and how the robot's belief will be shaped from its own actions. This is because, under the separation principle, the environmental belief produced by a perception module must be treated as immutable by the planning module, whereas in reality the belief will change as the robot gathers more measurements. Instead,

the planning module simply employ conservatism in parts of the environment not yet observed, such as assuming unobserved regions as being occupied with low probability. In other words, the separation principle leads to *reactive* behaviours in the sense that the perception module passively processes incoming sensor measurements, and the planning module passively consumes the result of perception.

Such reactivity renders non-myopic decision-making algorithms myopic. If the environment is largely unknown, any non-myopic plan lacks relevance especially further along the planning horizon, because the actions in distant future either derive from inaccurate knowledge or rely on conservatism against uncertainty. Therefore, in practice, the performance of sophisticated non-myopic planning algorithms is limited by environmental uncertainty and the neglection of how the robot's actions can reduce the environmental uncertainty. To be truly non-myopic, then, requires consideration of what level of environmental knowledge will be afforded by the system in the future, and hence the quality of sensor data throughout.

The idea of quantifying and accounting for the effect of measurements is not new. A formulation to this end is *active perception* [14, 15], which aims to select actions such that the measurements gathered throughout operation provide the best situational awareness of the environment. Actions are explicitly valued according to the quality of measurements or viewpoints quantified in terms of geometric criteria such as coverage, or probabilistic criteria such as Shannon information gain [16, 17]. Improvement in situational awareness often leads to demonstrable improvement in task performance, for example, in locating objects of interest [18]. Similarly, some formulations incorporate the effect of measurements into non-myopic planning by explicitly simulating or enumerating possible measurements, and keeping track of how the robot's belief would evolve over time [19–22]. Solutions based on this perspective often demonstrate greater resilience against uncertainty. These developments support our claim that we must transcend the separation principle in order to achieve a higher level of autonomy. Motivated by these recent developments, this thesis aims to develop a principled algorithmic framework for predicting possible measurements and evaluating their quality in non-myopic planning.

1.3 Robotic Prospection

1.3.1 Prospection in Cognitive Psychology

We look to theories of intelligence in cognitive science for inspiration on the mechanism for robot autonomy. Earlier theories of intelligence was dominated by the *behaviourist theory* [23], which strongly resembles the separation principle. Behaviourists believed that all actions are 'responses' to external 'stimuli', much like in the well known Pavlov's dog experiment [24]. In this setting, the role of intelligence is to convert 'sensory signals' into 'action signals'. Reward-seeking behaviours are instilled through reinforcement and punishment from past experiences. In other words, actions are habitual and reflexive, which is analogous to the perception-action pipeline in the separation principle in the sense that incoming sensory signals are simply processed by a 'filter' to produce action.

Much like the shortcoming of the separation principle in realising true autonomy beyond reactivity, the behaviourist thesis is now abandoned because of the necessity of a higher degree of cognition and foresight in explaining intelligence. For example, in [25] the authors conditioned a rat to turn left for food, and subsequently disabled turning left by severing its motor cortex. The behaviourist thesis would dictate the rat can no longer reach the food. However, the rat instead turned 270° right. This implies that the rat foresaw turning right would lead to food, beyond simple conditioning and reflexes to its sensory data at present, as we would like to instill in robots.

The concept of *prospection* has been proposed to explain such future- and goal-oriented behaviours [3]. Prospection refers to episodic foresight in humans and other animals. Just as retrospection reviews past events to evaluate the current ones, prospection previews the future [3, 26]. The main claim is that an important element of intelligence is the ability to generate and evaluate possible future scenarios and outcomes based on the present state. Past experiences are used to build a predictive model of what the environment looks like, or what future measurements or outcomes are [26]. It has been shown that prospection is critical in motivation and planning in non-myopic tasks with delayed rewards [27]. Further, prospection explains exploratory behaviours because executing an action offers not only reward, but also information from verifying hypothesised outcomes [26, 28]. In doing

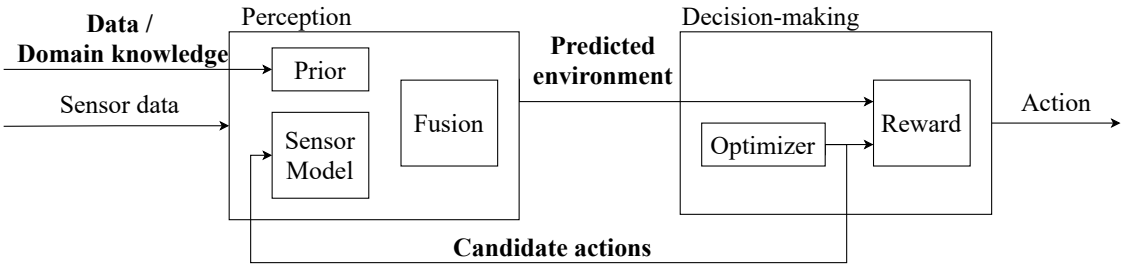


FIGURE 1.3: An architecture diagram for robotic prospection.

so, exploration is guided toward likely scenarios based on inductive biases [29], or what information is actionable [30].

1.3.2 A Recipe for Robotic Prospection

We are interested in translating the insights from prospection from cognitive science to algorithms and software for autonomous robots. A prospecting robot will *predict* and act in a dynamic and uncertain environment, rather than simply adapting to it. To realise such prospecting robots, the separation principle must be modified in two main ways.

R1 Prospective Perception: the perception module should predict possible environmental states based on onboard sensor measurements.

R2 Prospective Planning: the planning module should evaluate the evolution of environmental uncertainty, and the improvement in situational awareness from taking a sequence of actions.

These modifications are illustrated in Fig. 1.3.

To achieve **R1**, the perception algorithm must make provision for non-trivial prediction of the environmental state based on onboard sensor measurements. Inductive bias is crucial to this end, because we must reduce the scope of possible environments. This implies a departure from the conventional conservatism of assuming the most general model possible. In practice, **R1** can be achieved by using prior knowledge about the environment, or by learning a model based on data. Additionally, the model must allow online updates

from online sensor measurements. In Ch. 7, we investigate what models are suitable for incorporating both prior knowledge and online sensor measurements via case studies.

For **R2**, we must devise a method for incorporating the evolution of environmental uncertainty into non-myopic planning, especially the effect of measurements. A possible solution is to enumerate all possible measurements, and grow a search tree to track the possible paths of how a robot's belief might evolve. However, this necessitates an exhaustive simulation. Instead, a more efficient alternative would be to abstract the simulated measurements in a form that is relevant to the task at hand. This thesis will investigate methods for incorporating the effects of measurement into planning without explicit simulation of the sensor model.

1.3.3 Technical Approach

The rest of this thesis is dedicated to implementing the extensions **R1** and **R2** as an algorithmic framework. We take a Bayesian perspective of general robotic decision making problems, where the robot is assumed to perceive the environment from sensor measurements via Bayesian inference. The problem of prospective perception is then to design effective predictive priors that expedite Bayesian perception. We formulate the prospective planning problem as that of designing acquisition strategies reminiscent of Bayesian optimisation [31].

Under this formulation, we first derive a universal prospective planning strategy called the mutual information upper confidence bound (MI-UCB) using information-theoretic tools in Ch. 4. MI-UCB states that the posterior expected reward is probabilistically upper bounded by a weighted sum of information gain and prior expected reward. Therefore, choosing actions that maximise MI-UCB approximately maximises the posterior expected reward based on the principle of optimism under uncertainty. In Ch. 5, we demonstrate both prospective perception and planning in tandem in the special problem instance of environmental monitoring. Prospective perception is realised by augmenting Gaussian process (GP) regression with a partial differential equation (PDE) from fluid dynamics. The augmented GP is used with an existing Bayesian optimisation strategy called Gaussian

process upper confidence bound (GP-UCB), which realises prospective planning. Chapter 6 considers prospective planning with complex semantic tasks. To model such tasks in uncertain environments, we derive a new specification language called random signal temporal logic (RSTL) as an extension of signal temporal logic (STL). Prospective planning is achieved with gradient-based planning in the special case of linear Gaussian targets. Finally, Ch. 7 presents two case studies on predictive priors for prospective perception in order to identify common desirable characteristics. In both cases, we incorporate physical PDE constraints into GP regression.

1.4 Scope of this Thesis

1.4.1 Prospective Planning Under Environmental Uncertainty

The primary objective of this thesis is to develop a mathematical framework for prospective planning and provide a general solution to this problem. In particular, we are interested in handling environmental uncertainty, because the emerging applications of robotics place far greater importance on environmental uncertainty than others. Based on a principled framework, we analyse how the measurements generated by different actions provide varying degrees of improvement in situational awareness and task performance of the system as a whole.

Doing so requires an understanding of the perception algorithm in use. We consider both general and problem-specific settings. Chapter 4 presents a general algorithmic framework for the former case, where we regard the perception algorithm as a generic Bayesian estimator that produces an environmental belief given sensor measurements. The latter case is considered in Chapters 5 and 6, where we consider representative perception problems such as target tracking and environmental monitoring, and exploit the problem structure.

1.4.2 Predictive Priors for Prospective Perception

Another important aspect of robotic prospection is to predict future scenarios based on current knowledge. In this thesis, we focus on predicting the unseen parts of the environment given limited sensor measurements. We emphasise that the representation of the environment varies significantly with the environmental phenomenon being modelled. Therefore, this thesis will focus on identifying general characteristics of perception algorithms that are suitable for robotic prospection, through a set of case studies on practical applications in Ch. 7.

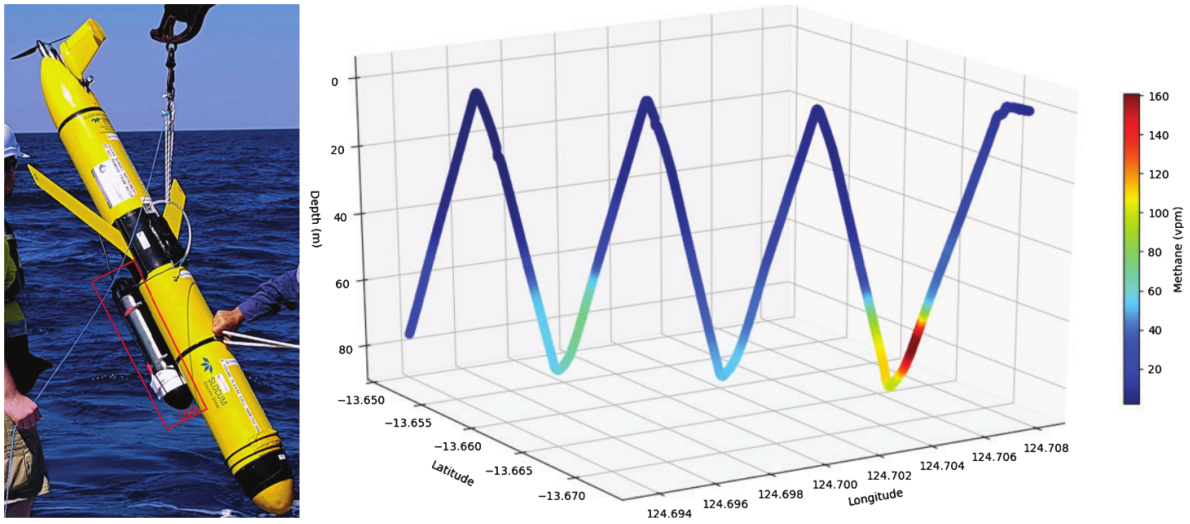
1.4.3 Practical Applications of Prospective Planning

We consider subproblems and specific instances of prospective planning in practical robotics applications. This section introduces and motivates these problem instances.

Autonomous Ocean Monitoring

Ocean monitoring offers tremendous economic value with various applications such as oceanographic research [32, 33] and military surveillance [34]. Various autonomous platforms are being introduced for such tasks, including autonomous underwater vehicles (AUVs) [35], underwater gliders [36], and even passive platforms without actuation [37–39].

An important example of ocean monitoring task that can significantly benefit from autonomous robots is source localisation, where the goal is to find the source of a substance that is suspended an ambient flow field. There are various circumstances in which it would be beneficial to track substances diffusing in water, such as chemical warfare protection [40]. and oil and gas source localisation [33]. One interesting recent development is a sensor that can measure methane concentration and that is suitable for use with underwater robots [33], as illustrated in Fig. 1.4. It is imperative to develop algorithms that make use of such sensors for effective application over large spatial scales of tens to hundreds of kilometres. This problem is a prominent instance of prospective planning, where the robot must simultaneously seek the source of a plume while simultaneously improving the estimate of its location. Chapter 5 solves this problem from the prospection perspective.



(A) (B)

FIGURE 1.4: Autonomous monitoring of oceanic plumes using underwater gliders. (a) shows An underwater glider equipped with a laser methane sensor (red box). (b) shows the methane concentration measurement over trajectory. Images reproduced from [33].

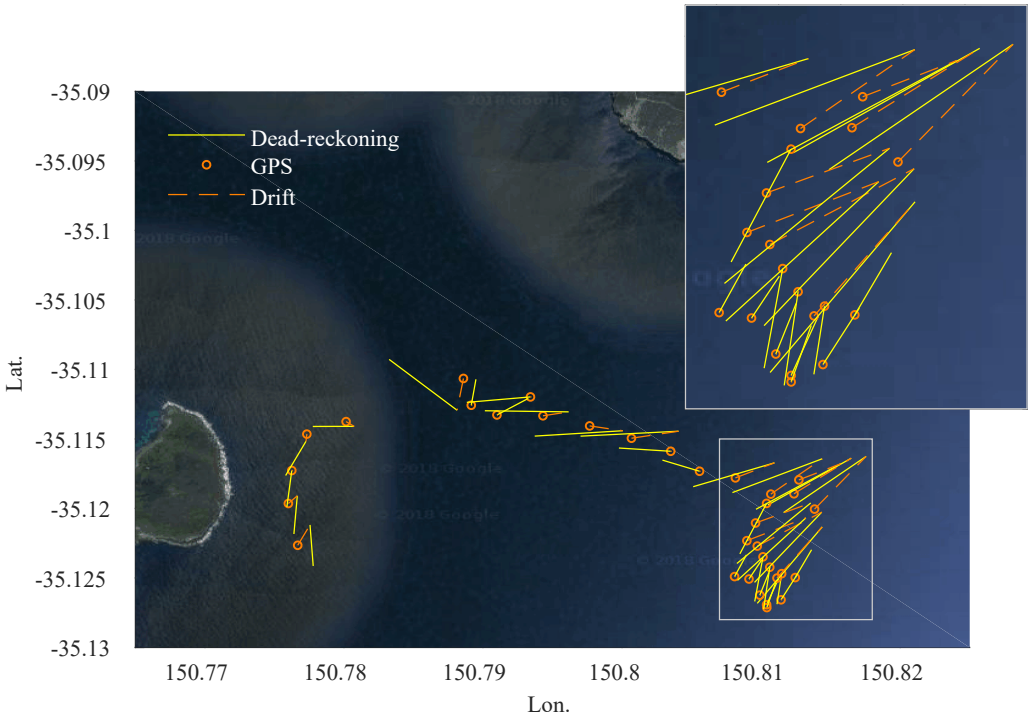


FIGURE 1.5: Result of a Slocum G3 underwater glider operation in Jervis Bay, Australia. The dead-reckoned position (yellow solid line) differ substantially from global positioning system (GPS) measurements on surface (orange circle) due to ambient current. Orange dashed line shows the drift, which we use for estimating ocean current. Inset: trajectory inside the white box.

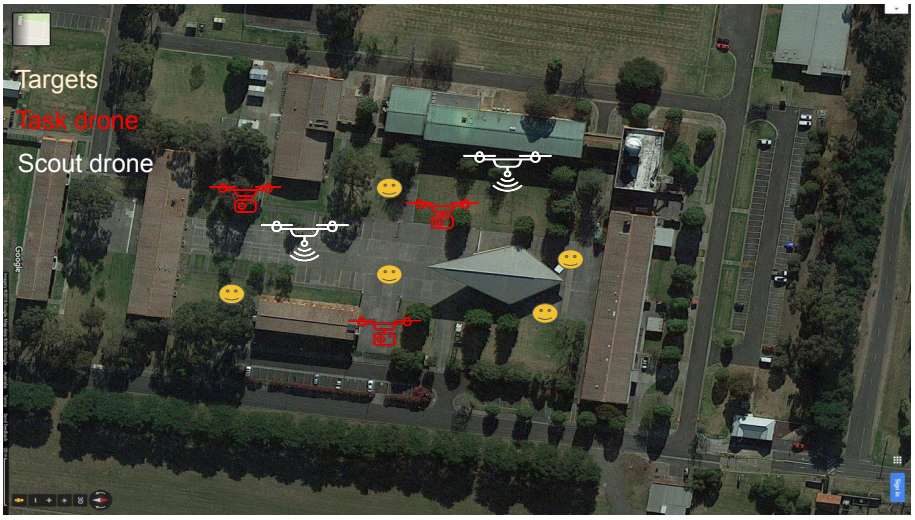


FIGURE 1.6: An example application for multi-drone surveillance. The task is to confirm all targets (yellow) with the task drones (red). Scout drones (white) support the process by sensing targets (yellow) from a distance, at low resolution, and cueing for possible target presence.

In deploying such algorithms to the ocean, a crucial challenge is the effect of ambient ocean current. Due to lack of GPS while submerged, ocean currents can cause considerable position drift. In the case of underwater gliders, this disparity is typically in the order of a few hundred metres, as illustrated in Fig. 1.5. This has a strong impact on the utility of the gathered data, as most ocean monitoring tasks concern spatial phenomena [41]. Thus, there has been substantial work on navigation and planning in flow fields [42–45]. However, most of this work assumes that the flow field is given *a priori*, e.g., from an external database [46–48]. Unfortunately, the spatiotemporal resolution or accuracy provided by most databases is insufficient for the purpose of navigation, as noted in [44]. Inspired by the idea of prospection, we are interested in estimating and predicting the ocean currents using limited onboard sensor measurements. In particular, we examine how the result of an action as measured by GPS can be used to estimate and predict the ocean currents in Ch. 7.

Heterogeneous Multi-Robot Systems

Decentralised multi-robot systems are a prominent approach to ensuring reliability and robustness in unknown environments, because many inexpensive, possibly disposable robots

can be composed into an effective team while ensuring redundancy. Heterogeneity allows further improvements via functional specialisation, because different robots can be equipped with different payloads to enhance the team's overall capability. A subset of the team could focus on gathering information, while the rest perform tasks.

In the later parts of Ch. 4, we define and study the *scout–task robot architecture*, a team composition in which some robots (i.e., *task robots*) are equipped to perform a particular task while others (i.e., *scout robots*) are equipped with sensors to rapidly acquire knowledge about the environment. There are many compelling applications of this idea. For example, it may be desirable to deploy disposable scout robots to ensure safe operation of a high-value task robot, as in the case of a Mars rover-copter team [49–51]. We consider the multi-drone surveillance application illustrated in Fig. 1.6, where a limited number of scout robots equipped with long-range sensors cue task robots for the presence of targets.

Scout–task robot architecture can be seen as an extension of prospective planning that side-steps the trade-off between exploration and reward seeking through heterogeneity. Whereas the two objectives must be balanced in the single-robot setting, a scout–task robot team permits simultaneous exploration and exploitation. The challenge is how to guide exploration in a way that is most relevant to exploitation. Scout robots should provide information about the environment that allows task robots to improve their plans and thus find higher-quality solutions. Chapter 3 presents a general formulation and solution of the scout-task coordination problem.

Semantic Task Planning in Unknown Environments

As robot autonomy continues to evolve, an imperative question is how untrained human operators will interact and command the robots. Temporal logic is a promising tool for robotics applications and explainable artificial intelligence (AI) in that it can be used to represent rich, complex task objectives in the form of human-readable logical specifications. Robotic systems equipped with the capability to perform temporal logic synthesis can implement powerful, intuitive command interfaces. We are interested in developing temporal logic synthesis for practical, real-world autonomous systems.

More widespread adoption of temporal logic in practical autonomous systems, in our view, is limited by two main factors. First, the computational requirements of existing synthesis methods are seen as prohibitive. The logical nature of temporal logic often leads to problem formulations that are NP-hard. For example, the complexity of linear temporal logic (LTL) synthesis grows doubly exponentially with the formula size [52], and STL synthesis requires mixed integer linear programming (MILP) of which the complexity grows exponentially with search horizon.

More importantly, the deterministic nature of STL limits reasoning over uncertain environments, because the logical predicates must evaluate to either true or false. This is particularly detrimental in autonomous robots, which must operate in uncertain environments. Although it is possible to impose deterministic conditions on uncertain semantics such as chance constraints [53] or bounds on Gaussian variance or entropy [54, 55], such predicates limit the expressivity and often requires the operator to have an intimate understanding of the algorithms onboard the robot. Furthermore, reasoning over uncertainty is crucial for prospection, as it allows evaluating how information gathering aids satisfying task specification. Chapter 6 presents a probabilistic extension of STL called RSTL that allows reasoning over uncertainty.

1.5 Contributions

The main contribution of this thesis is the introduction of prospection as a new mechanism for robot autonomy. This novel perspective illuminates the importance of proactivity and foresight, and unifies several previous formulations that partially address the issue. We present a suite of algorithmic tools for realising robotic prospection. Each of the tools solves the prospective planning problem in numerous settings. Chapter 4 considers a generalised prospective planning problem and its extension in heterogeneous multi-robot systems. Chapter 5 focuses on source localisation in environmental processes, a prominent practical application of prospective planning. Chapter 6 extends formal methods-based semantic planning to enable robotic prospection. Chapter 7 presents case studies on predictive priors for robotic prospection, and investigates the properties of good models for this purpose. More specifically, the contributions of this thesis are as follows:

- **Robotic prospection in uncertain environments:** This thesis presents a mathematical formulation of robotic prospection in uncertain environments. In particular, we present several general and specialised formulations of prospective perception and planning under environmental uncertainty. The formulation is general and applies to most modern robotic systems that perceive the environment using Bayesian principles. Based on the general formulation, we consider specialised problem instances in order to exploit the problem structure wherever available.

- **Mutual information upper confidence bound (MI-UCB)** for generalised prospective planning and simultaneous exploration and exploitation: We develop MI-UCB as a general solution for prospective planning under environmental uncertainty. The MI-UCB strategy allows selecting actions that approximately maximise the expected reward conditioned on measurements, without actually taking the measurements. Further, we show that MI-UCB can be scaled up to a scout-task team consisting of scout robots that gather information and task robots that seek reward. The significance of MI-UCB is that it provides a methodical recipe for practical implementation of prospection in general Bayesian robots, with a mild assumption that the environment is static.

- **Physics-based predictive priors** for robotic prospection: This thesis presents several predictive priors for robotic prospection that employ inductive biases based on physical knowledge. These models are derived by incorporating the governing PDE models into GP regression. Through case studies, we show that these predictive priors significantly improve planning performance in practical problem instances.

- **Random signal temporal logic (RSTL)** for uncertainty- and perception-aware semantic planning: We present a probabilistic formulation of STL called RSTL, which is designed to support synthesis of robot trajectories that satisfy specifications defined over uncertain events in the environment. The quality of robot's actions are evaluated in terms of probability of success, rather than a binary measure of task compliance. The inclusion of uncertain semantics is key to prospection, as it allows evaluating how the robot's measurements improve its situational awareness and hence the probability of mission success. Concurrently, this novel probabilistic formulation leads to a 'soft' metric for evaluating mission success, and hence allows efficient planning via gradient-based methods.

1.6 Publications

The work presented in this thesis resulted in a total of 11 publications, consisting of two journal articles, seven conference proceedings, and two refereed workshop papers. Of those, C7 won the best student paper award, and W1 won the best poster award.

The contents of this thesis are largely based on these publications. Specifically, Ch. 3 is based on C1. Chapter 4 is based on C1 and W1. Chapter 5 is based on C7. Chapter. 6 is based on C2. Chapter 7 is based on C5 and J1. Appendix A is based on C1 and W1. Appendix B is based on C5. Appendix C is based on C2. Ki Myung Brian Lee is the primary contributing author in these publications, except in J1. Other publications do not directly appear in this thesis, but support and complement the results.

Journal Articles

J1 L. Wu, **K. M. B. Lee**, L. Liu, T. Vidal-Calleja. 2021. 'Faithful Euclidean Distance Field from Log-Gaussian Process Implicit Surfaces'. *Robotics and Automation Letters*, vol. 6, no. 2, pp. 2461-2468. [56].

J2 Y. Chen, L. Zhao, **K. M. B. Lee**, C. Yoo, S. Huang, and R. Fitch. 2020. 'Broadcast Your Weaknesses: Cooperative Active Pose-Graph SLAM for Multiple Robots'. *Robotics and Automation Letters*, vol. 5, no. 2, pp. 2200-2207. [57].

Conference Papers

C1 **K. M. B. Lee**, F. H. Kong, R. Cannizzaro, J. L. Palmer, D. Johnson, C. Yoo, and R. Fitch. 2021. 'An Upper Confidence Bound for Simultaneous Exploration and Exploitation in Heterogeneous Multi-Robot Systems'. *International Conference on Robotics and Automation (ICRA)*, Xi'an, China. [58].

C2 **K. M. B. Lee**, C. Yoo, and R. Fitch. 2021. 'Signal Temporal Logic Synthesis as Probabilistic Inference'. *International Conference on Robotics and Automation (ICRA)*, Xi'an, China. [59].

C3 K. Y. C. To, **K. M. B. Lee**, C. Yoo, F. H. Kong, S. Anstee, and R. Fitch. 2021. 'Estimation of Spatially Correlated Ocean Currents from Ensemble Forecasts and Online Measurements'. *International Conference on Robotics and Automation (ICRA)*, Xi'an, China. [60].

C4 **K. M. B. Lee**, W. Martens, J. Khatkar, R. Fitch, and R. Mettu. 2020. 'Efficient Updates for Data Association with Mixture of Gaussian Processes'. *International Conference on Robotics and Automation (ICRA)*, Paris, France. [61].

C5 **K. M. B. Lee**, C. Yoo, B. Hollings, S. Anstee, S. Huang & R. Fitch. 2019. 'Online Estimation of Ocean Currents from Sparse GPS Data'. *International Conference on Robotics and Automation (ICRA)*, Montreal, Canada. [62].

C6 K. Y. C. To, **K. M. B. Lee**, C. Yoo, S. Anstee & R. Fitch. 2019. 'Streamlines for Motion Planning in Underwater Currents'. *International Conference on Robotics and Automation (ICRA)*, Montreal, Canada. [63].

C7 **K. M. B. Lee**, J. J. H. Lee, C. Yoo, B. Hollings & R. Fitch. 2018. 'Active Perception for Plume Source Localisation with Underwater Gliders'. *Australasian Conference on Robotics and Automation (ACRA)*, Lincoln, New Zealand. **Best Student Paper Award Winner**. [64].

Refereed Workshop Papers

W1 **K. M. B. Lee**, F. H. Kong, R. Cannizzaro, J. L. Palmer, D. Johnson, C. Yoo, and R. Fitch. 2021 'Decentralised Intelligence, Surveillance, and Reconnaissance in Unknown Environments with Heterogeneous Multi-Robot Systems" *ICRA2021 Robot Swarms in the Real World: From Design to Deployment*, **Best Poster Award**, Xi'an, China. [65].

W2 G. D'urso, J. J. H. Lee, **K. M. B. Lee**, J. Shields, B. Leighton, O. Pizarro, C. Yoo, R. Fitch 'Field Trial on Ocean Estimation for Multi-Vessel Multi-Float-based Active Perception' *ICRA2021 1st Advanced Marine Robotics TC Workshop: Active Perception.* [39].

1.7 Thesis Outline

The remainder of this thesis is organised as follows.

Chapter 2 surveys previous work related to the prospective planning problem.

Chapter 3 mathematically formulates robotic prospection from a Bayesian perspective.

Chapter 4 presents a general solution to the prospective planning problem using information-theoretic tools. The solution is used to scale up the problem to simultaneous exploration and exploitation in heterogeneous multi-robot systems.

Chapter 5 demonstrates prospection in both perception and planning in the special problem instance of environmental monitoring.

Chapter 6 presents a solution for prospective planning problem in complex semantic tasks.

Chapter 7 presents case studies on how predictive priors can significantly improve planning, with examples in navigation in oceanic and indoor environments.

Chapter 8 concludes the thesis with important avenues for future work.

Chapter 2

Related Work

This chapter reviews previous work relevant to robotic prospection. An important premise of prospection is that the environment and, hence, the outcomes are *uncertain*. We thus give an outline of classes of uncertainty in robotics, and their representation in Sec. 2.1. Then, we review general problem formulations and solution methods related to prospection in Sec. 2.2. Subsequently, Sec. 2.3 reviews relevant work on specific problem instances introduced in Ch. 1. Finally, Sec. 2.4 identifies limitations in previous work, and concludes the chapter.

2.1 Classes of Uncertainty Relevant to Prospection

We first review common classes of uncertainty in robotics that prospection must consider. We examine three major classes of uncertainty. First, we discuss how the environment, for examples, obstacles and general scenes, are represented and updated in Sec. 2.1.1. Then, we discuss how specific targets of interest, such as objects, are handled in Sec. 2.1.2. Finally, Sec. 2.1.3 presents how uncertainty in robot dynamics is considered.

2.1.1 Environment

A fundamental requirement for mobile robots is to avoid obstacles. If obstacles are unknown *a priori*, they must be mapped using onboard sensors, with appropriate representation of uncertainty. The simplest method for obstacle mapping is to consider a discrete grid on which the occupancy of obstacle is recorded [66]. The earlier work of [66, 67] implemented exactly this idea, by fusing 2D laser scan data onto a discrete grid for mapping. The key insight is that the 2D laser scans can themselves be represented as a grid by raycasting, which the authors refer to as inverse sensor model [67]. This can be augmented with a confidence value to provide a richer description [68]. The mapping can be made computationally efficient by the use of quad- or oc-tree data structures [69, 70], and can be scaled to 3D [71]. By tracking the probability of obstacle occupancy, the planning algorithms can plan conservative paths by computing and minimising the risk of collision (see [72]).

Continuous approaches such as [73, 74] use Gaussian processes (GPs) [75] to represent the occupancy. The spatial correlation between occupancy at different points are captured using a kernel function, so that it is more likely to observe an occupied region near another occupied region. This idea is developed further in [76] with a specialised kernel that captures sharpness in obstacles, so that the learnt obstacles are more realistic. Additionally, the uncertainty of the sensor location is considered in [77], and the authors of [78] combine continuous mapping with insights from discrete mapping. Using the continuous representation, collision-free paths can be planned with gradient descent [79].

Beyond occupancy, richer class of information such as distance to obstacle has proved useful in navigation [80–82]. The information is stored as a signed distance field (SDF), which records the distance to the closest obstacle. The SDF may be truncated to form a truncated signed distance field (TSDF) [80, 81, 83], or computed exactly to form an Euclidean signed distance field (ESDF) [82]. SDFs are useful because the planning algorithm can naturally compute how close a path is to the obstacle, and can be used for topological analysis such as computing the medial axis [84]. Further, SDFs are computationally advantageous in constructing a realistic 3D mesh of the environment using sensor data [83].

An even richer class of information is semantics which includes, for example, the category of objects or rooms. Earlier work uses tools such as support vector machines (SVMs) [85] or conditional random fields (CRFs) [86] to classify room category and build a semantic topological map consisting of room categories. More recent work such as [87, 88] uses deep learning-based classifiers to build a large-scale map with both object and room categories, possibly using GPs for mapping [88]. It is also possible to learn the map directly using generative models such as sum-product networks (SPNs) [89, 90] or generative adversarial networks (GANs) [91]. Interestingly, semantic mapping algorithms can often predict the unseen area using the semantics via Bayesian inference. Such capability is demonstrated at the semantic level in [85, 86, 89], and at the metric level in [89–91].

2.1.2 Target Localisation

Another important class of environmental uncertainty is the state of targets of interest, which robots may avoid [92], track [93, 94] or search for [93, 95]. This section reviews representations of belief over targets and the methods for updating them.

The most common representation of target belief is to represent the target position as a Gaussian distribution and use a Kalman filter [96]. The target's dynamic model is assumed to have constant velocity or acceleration. Nonlinear models can be used by linearising the dynamic model (see [97, Ch. 6]).

If the target dynamics or the measurement model is too nonlinear, unscented transform (UT) [98] or sequential Monte Carlo (SMC) methods [99] are used, where the belief is propagated by using a set of particles. Alternatively, similarly to the case of obstacles, target belief may be maintained using a discrete grid [100, 101] or a continuous function [102, 103]. For example, in [100], a grid-based filter is used to track the position of a radio frequency (RF) tag, in order to employ a complex measurement model based on received signal strength indicator (RSSI).

When there are multiple targets, the most immediate solution is to duplicate the filter for each target, and associate the measurements to individual targets [104]. More recent approaches such as the probability hypothesis density (PHD) filter [105, 106] alleviate the

need for association using the random finite set (RFS) statistics [107]. A RFS represents a finite set of unknown number of targets with unknown locations. A PHD filter is a probabilistic filter for multiple targets with RFS statistics, just as a Kalman filter is for a single target with Gaussian statistics. PHD filters can be implemented with a set of particles based on the SMC method [106], or a mixture of Gaussian distributions each maintained by a Kalman filter [105]. A hybrid method combining RFS filtering and discrete grid mapping is presented in [108] to represent dynamic obstacles.

Similar to the case of obstacle mapping, a recent trend is to learn semantic correlation or association between targets, such as objects that commonly occur together. This has been demonstrated in household objects using Markov random fields (MRFs) [109] and particle filtering augmented with a CRF [110, 111]. Learning such semantic correlation has been shown to yield substantial improvements in object search [111, 112]

2.1.3 Robot Dynamics

An important class of uncertainty is the dynamics of the robot itself. In the simplest case, the dynamics may be subject to parametric uncertainty, where, for example, the robot's inertial parameters are unknown. This is the system identification problem, which may be solved with least squares [113–115], or Kalman filtering with augmented states [116, 117]. When the dynamics is completely unknown, non-parametric methods are used. One non-parametric approach is to use a GP to learn the state-space model [118–120], using expectation-maximisation (EM) or variational inference. Another approach is to estimate the dynamics in terms of its Koopman operator, a 'lifted' representation of the dynamics in an infinite-dimensional feature space [121]. With the dynamics identified, one may design a controller with respect to the learnt model. This is referred to as *certainty equivalent* control.

Adaptive control techniques [122–124] combine estimation and control of the unknown dynamics with a prescribed convergence guarantee, for both parameteric [122] and non-parameteric uncertainty [123, 124]. In particular, recent work on adaptive control and system identification shows that certainty equivalent control may not be sufficient [125, 126], because it does not account for the uncertainty in the estimated dynamics, and

it does not excite the system trajectory to improve the estimated dynamics. To this end, optimal experimental design approaches generate trajectory that optimally excite the system [127, 128] so that least squares- or Kalman filter-based estimators perform well. *Dual control* approaches [129] directly design a controller that not only stabilises the system, but also steers the trajectory toward optimal excitation [126, 130].

An important special case considered in this thesis is when a part of the robot's dynamics is known, but there is a significant unknown residual drift. Such is the case for vehicles operating in oceanic currents or wind fields, such as underwater gliders [36, 62]. To estimate residual drifts online, one approach is to consider the drift field as a low-frequency disturbance and then apply an extended Kalman filter (EKF) [131] or nonlinear observer [132] in conjunction with acoustic sensors. However, modelling ocean currents or wind as a temporal phenomenon clearly overlooks its spatial structure, and acoustic sensors typically require a stationary reference (e.g., the seabed) [133].

An approach that does consider the spatial nature of drift is presented in [134], where the authors estimate ocean current by taking the average velocity between global positioning system (GPS) measurements. Due to its simplicity, the estimate is increasingly unreliable as distance between diving and surfacing locations grows, and lacks predictive power. An improvement on this concept is the 'motion tomography' algorithm [135, 136], which reconstructs the local ocean current from GPS measurements using techniques from the tomography literature. Similar techniques are developed in [137] and [44] with dense GPS measurements and direct ocean current measurements respectively. Using such estimates of the ocean current, approaches in [138, 139] plan efficient trajectories that exploit the drift field. Both these approaches take multiple samples from a probabilistic belief rather than a single most likely estimate, which shows the importance of considering the uncertain nature of the estimate. Concepts similar to dual control are presented in [140, 141], where the planning objective combines both navigation performance and system excitation.

2.2 General Problem Formulations and Planning Methods

This section reviews previous formulations of planning under uncertainty. We first discuss partially observable Markov decision process (POMDP) formulation and the corresponding solution techniques in Sec. 2.2.1. Then, we review active perception and information gathering formulations in Sec. 2.2.2. We then explore Bayesian optimisation formulation, and the concept of exploration-exploitation trade-off in Sec. 2.2.3.

2.2.1 Partially Observable Markov Decision Processes

Markov decision processes (MDPs) and POMDPs are fundamental frameworks that model sequential decision making. In this section, we present a brief overview of these concepts.

Following [142], in MDPs, a transition model $p(s_{t+1} \mid s_t, a_t)$ describes the 'dynamics' of the robot, where a_t is the robot's action at time t, and s_t and s_{t+1} represent the robot's state at times t and $t+1$ respectively. The aim is to maximise a sum of expected reward $\sum_t \mathbb{E}[R(s_t, a_t)]$ by either choosing a sequence of actions $a_1 \dots a_T$ over a finite time horizon T, or devising a 'policy' $p(a_{t+1} \mid s_t)$. POMDPs introduce an additional complication that the states s_t are not directly observable. Instead, only the measurements y_t are available, which are generated according to a sensor model $p(y_t \mid s_t, a_t)$. This is useful for modelling practical robotic systems, where the available information is limited to imperfect, noisy sensor measurements or restricted knowledge about the environment. Although the aim is still to maximise the sum of expected reward, the algorithm must account for the lack of a direct measurement of s_t. The policy generated must depend on the measurement only, and should follow the form of $p(a_{t+1} \mid y_t)$.

Both MDPs and POMDPs are solved using value iteration based on Bellman's principle of optimality (see [143, Ch. 2]). In particular, POMDP instances are converted to MDPs over the belief space, which necessitates the construction of a belief tree (see [19, 21, 144]). This is because the combination of different measurements lead to different beliefs, even if they share the same mean. For this reason, whereas MDPs can be solved in polynomial time through value iteration, solving POMDPs exactly is generally PSPACE-hard [145]. Thus, various POMDP solvers have been developed that approximate value iteration or speed up

the numerical operations involved in value iteration [20, 146]. Further, efficient algorithms have been proposed by considering the special case of POMDP instances with Gaussian belief space, such as maximum likelihood measurement assumption [22], feedback-based information roadmap [21], belief roadmap [147], and reduced value iteration [16, 148, 149].

An emerging trend in POMDP solvers in general belief space is the use of algorithms reminiscent of Monte Carlo tree search (MCTS). MCTS is a biased random search algorithm that asymptotically finds the optimal sequence of actions. In the reinforcement learning community, MCTS has been popularly used for solving problems that are too large or intractable. For example, AlphaGo [150] used MCTS in combination with a deep learning-based value function estimator to play the game of Go. [19] proposes partially observable Monte Carlo planning (POMCP), an MCTS-like algorithm for solving general POMDPs. Actions are searched using the UCB on trees (UCT) heuristic [151], and possible measurements are sampled according to the measurement model. This is later extended in [152], where the authors learn the measurement model online. The determinised sparse partially observable tree (DESPOT) algorithm [144] alleviates the computational complexity of POMCP-like approaches [19] by considering a carefully selected subset of scenarios. In [92], DESPOT is demonstrated in an autonomous navigation scenario comprising a large crowd through combination with deep-learned value function similar to the technique used in AlphaGo [150].

A variant of POMDP closely related to this work is the mixed observability Markov decision process (MOMDP) [153]. In a MOMDP, a portion of the state variables s_t are fully observable (e.g., robot's state) whereas the rest (e.g., the environment) is partially observable. The full observability of some state variables leads to considerable reduction in the search space, thereby improving the computational speed. Meanwhile, it is still required to construct a belief tree for all possible measurements of the partially observable state variables.

2.2.2 Active Perception and Information Gathering

Active perception approaches aim to overcome the challenge of perceptual uncertainty by selecting the robot's actions in a way that enhances its understanding of the environment [14, 15, 17]. In conventional passive perception systems, the perception algorithm simply processes given sensory information. In contrast, an active perception system will interactively choose what the sensor needs to see, closing the loop in the perception-action cycle [154].

One possible approach to active perception is to use geometric heuristics such as coverage of the environment. For instance, in [155], the authors propose to identify the frontiers of the mapped portion of an indoor environment, and subsequently select the largest frontiers for exploration. A similar simplification is as an orienteering problem [156], where the aim is to choose a set of viewpoints such that the number of covered features is maximised. Another simplification is the coverage formulation presented in [157], where the algorithm is given an approximate 3D model of the environment to survey, and viewpoints are generated and sequenced for maximal coverage.

More theoretically grounded approaches to active perception use concepts from information theory. A typical formulation is to maximise the mutual information between the measurements and the environment, or the *information gain* [158–160]. A fundamental property is that information gain is monotone submodular [158, 161], which can be exploited to design efficient algorithms. Although maximising a monotone submodular function is NP-hard [161], many efficient algorithms can yield approximately optimal solutions with known suboptimality bound, including recursive greedy [162], distributed sequential greedy [163], rapidly exploring random trees (RRTs) [159], and MCTS [164, 165].

Using information-theoretic tools, the authors of [166] consider informative path planning for weed classification in agriculture using uninhabited aerial vehicles (UAVs). Weed presence is represented using an occupancy grid, and a simple binary sensor model is used to model the misclassification rate of a more sophisticated classifier. A similar approach is presented in [167] for seafloor exploration, where the aim is to minimise the entropy of a GP-based classifier over space. [168] models the surface of a ship hull as a combination of 2.5D height maps represented as GPs. The authors use an adaptive variant of Chekuri's

recursive greedy algorithm [162] to plan a sequence of viewpoints that minimise the total variance, and report better performance compared to standard coverage or random viewpoints. In [169], an algorithmic framework is presented that enables active scientific sample selection for an interplanetary rover. A Bayesian network is trained to encode the scientific domain knowledge over different types of geological samples distributed over space. Notably, these formulations do not require explicit sampling to evaluate the utility of a possible measurement, unlike belief tree construction in POMDP solvers.

Meanwhile, more complicated formulations require sampling of possible measurements. Active classification problems such as [170] and [171] sample the possible measurements, and either compute the information gain or simulate the classifier to evaluate the viewpoints. Similarly, the Monte Carlo active perception approach [165] uses an MCTS-like search algorithm to maximise the information gain of a pointcloud-based GP classifier. In particular, it is worth noting that occlusion is considered by sampling possible measurements, and subsequently estimating the information gain during backpropagation. A similar approach is presented in [172] for active object recognition with tactile inputs. The sensor measurements are directly simulated, and MCTS is used to optimise for misclassification error directly. Approaches such as [18, 165] use GP to predict the unseen parts of the environment, and optimise the information gain using variants of MCTS.

2.2.3 Bayesian Optimisation and Exploration-Exploitation Trade-Off

The simplest instance of decision making under uncertainty with measurements is the multi-armed bandit (MAB) formulation [173–175]. In this formulation, a bandit is given a number of slot machines that can be played. Each machine has a varying payoff rate that is initially unknown to the bandit. The bandit's goal is to adaptively choose the best slot machine based on the payoff it observes.

The inherent challenge in the bandit problem is the *exploration-exploitation trade-off*. The bandit must, at some point, decide the best slot machine based on the payoffs observed, and *exploit* the knowledge to maximise the reward. Meanwhile, premature exploitation may lead to misidentifying a suboptimal slot machine. The bandit must also *explore*.

Numerous approaches have been proposed to solve the exploration-exploitation trade-off, including ϵ-greedy [176], Boltzmann exploration or Thompson sampling [177, 178], and principle of optimism under uncertainty [179]. Principle of optimism under uncertainty is one of the most prominent of these approaches. It asserts that, based on the statistics of the reward observed by the agent, the agent must act optimistically, and base its decision on the highest reward probable. In other words, the agent must pick an action whose upper confidence bound (UCB) of the reward is the highest. Here, UCB means the upper limit attainable by the reward subject to a confidence constraint, as per confidence intervals in statistics.

The principle of optimism under uncertainty is realised in multi-armed bandits by universal UCT [151] and Kullback-Leibler UCB (KL-UCB) [180]. In UCT, the agent keeps track of the mean of the reward and the number of times the action has been executed, with which the UCB can be calculated. These UCBs are based on a very relaxed assumption that the reward follows any distribution with a bounded range, and hence requires a large number of samples for convergence. Hence, these approaches are parallel to ours, since it is infeasible for a physical robot to gather sufficient data during its operation.

A more suitable class of algorithms for robotic applications in unknown environments is Bayesian optimisation [31]. A prominent example is Gaussian process upper confidence bound (GP-UCB) [181], which has been used in human–robot interaction [182] and environmental monitoring [183]. In this setting, the goal is to find the maximum of a function modelled as a GP. The agent cycles between gathering a new sample at a particular location, and updating the belief represented as a GP (see Sec. 3.1.2). Given the GP representation, the UCB is constructed as the sum of mean and a multiple of standard deviation, just as confidence intervals in Gaussian distributions are calculated. Concurrently, the mean and standard deviation terms represent exploitation and exploration respectively. Therefore, choosing points with higher UCB leads to convergence toward the optimum. Further, because the GP prior is stronger than the ones in multi-armed bandit formulations, the convergence is much faster.

At a more abstract level, exploration-exploitation trade-off has appeared as a heuristic in various robotics problem instances. In [18, 166], the current predictive mean is exploited

to focus exploratory efforts on specific areas of interest (e.g., high weed concentration or likelihood of apple presence, resp.). On the other hand, the approaches in [57, 184] toggle between exploratory (i.e., informative) and exploitative (i.e., task-oriented) actions using a pre-defined threshold, such as mission risk or localisation error. In the case of unknown dynamics, a balance can be achieved between exploration for better dynamic modelling and exploitation for minimising a control objective by considering a weighted sum of the information gain on dynamics with the control objective [140, 141]. An analogous approach for environmental uncertainty is presented in [185], which uses a weighted sum of information gain on environment and path distance as an objective for indoor navigation. Meanwhile, dual control approaches such as [126, 130] directly opt the for principle of optimism under uncertainty. In a multi-robot setting, exploration and exploitation is distributed across a heterogeneous team of a Mars rover and a helicopter in [49–51]. Although these approaches offer a heuristic for the desired behaviour of balancing exploration and exploitation, the performance implication is unclear, except in some dual control approaches [126, 130].

2.3 Applications

2.3.1 Monitoring Environmental Processes

An interesting problem instance considered in this thesis is environmental process monitoring, where the goal is to find or track the source of a plume given concentration measurements [186]. A traditional approach is to adopt an empirical parametric model referred to as the Gaussian plume [40], and to fit the observations to the model. However, using a parametric model may prove demanding in practical scenarios, because the data can be sparse or subject to noise. Adaptive heuristic strategies such as waiting time [187] and centroid tracking [188] have been proposed that demonstrate improvement in accuracy with additional data. A more principled approach is the 'infotaxi' strategy [189], where actions are chosen to maximise the information gain on the source location [190]. Although the infotaxi strategy has a theoretically justified background, a substantial limitation is that the information gain cannot be calculated in closed form for any flow fields more complex than the constant one.

2.3.2 Target Search and Localisation

An important application of robotics in uncertain environments is to find and possibly track targets of interest in an unknown environment. Active search algorithms [191] aim to detect the most targets within an environment, whereas active localisation algorithms [16] aim to plan a path that reduces the localisation uncertainty for the detected targets. Sensor measurements are fused into a probabilistic belief over target locations, represented as an occupancy grid [101] or a random finite set [93, 94, 192], as reviewed in Sec. 2.1.2. Subsequently, a plan is generated that either maximises the expected number or probability of detection [95, 103, 193, 194], or minimises the uncertainty of target locations [93, 100, 195].

A traditional approach to the active search problem is to model the problem as a *pursuit-evasion game*, where a number of 'pursuers' search an environment to capture all 'evaders' [191]. These approaches are typically formulated over a graph [95] or a grid map [193], and a Bayesian filter is often used to update the belief over target presence. Notably, the authors of [102] present a continuous, probabilistic approach to the target search problem, where the aim is to find a continuous path that maximises the detection likelihood. The continuous framework also allows inter-agent collision avoidance through distributed gradient descent [103].

Multi-robot systems have been considered with varying mobility [193, 196, 197] or sensing capability [198, 199]. Because solving the pursuit-evasion game over a general graph is NP-complete [191], centralised computation is often necessary to coordinate a multi-robot system in this case. Nonetheless, pursuit evasion-based frameworks have been demonstrated experimentally with multi-robot systems consisting of uninhabited ground vehicles (UGVs) [95], UAVs [200], and both [193].

Active localisation approaches aim to reduce the uncertainty of target position. For example, in [100], active localisation is demonstrated for a radio-tagged wildlife bird using a UAV. The belief over target presence is estimated using a grid Kalman filter, and the UAV plans the next best sensing location to reduce the entropy over target location. A closely related problem is the active simultaneous localisation and mapping (SLAM) problem. Assuming a Kalman filter-based SLAM system, the approaches in [21, 147] minimise the

position uncertainty of a single robot. Similarly, the target [148] and map uncertainty [16] can be minimised using a multi-robot system. Sparse pose graph-based approaches are presented in [57, 201] that reduce both the localisation and map uncertainty for single and multiple robots.

Recent work on active localisation based on multi-target tracking combine both search and localisation objectives. One such approach is presented in [93] based on the PHD filter [105]. The robots maximise information gain over both the number of targets and their respective positions, driven by analytical gradient of the RFS mutual information. The authors later extend to multi-robot systems by considering Voronoi partitioning, where each robot is only responsible for targets within its own Voronoi cell [202]. A similar approach is presented in [148] for a fixed number of linear Gaussian targets. A specialised tree search algorithm is used for planning, with a prescribed performance bound. Search is achieved by introducing fictitious targets.

2.3.3 Planning for Semantic Tasks

To model semantic tasks in uncertain environments, one approach is to model the robots' dynamics as a discrete MDP, where each state is assigned semantic labels with corresponding uncertainty [51, 203–205]. A natural task specification tool is linear temporal logic (LTL), given which a product MDP [203] is constructed from an automaton and a sequence of actions are found that maximize the probability of satisfaction. These ideas can be extended to the continuous case by judiciously partitioning the environment [45, 206, 207]. However, we find that partitioning is computationally prohibitive for online operations in uncertain environments, because any change in belief about the environment would lead to invalidation and expensive recomputation. Moreover, the construction of a product MDP is a computationally expensive operation, and improving its scalability remains an open problem.

A more suitable tool for continuous signals is signal temporal logic (STL) [208]. Unlike in LTL, satisfaction is determined using continuous-valued *robustness* [209]. Existing work uses STL to specify a task defined over a set of deterministic classes of conditions on the environment uncertainty, such as chance constraints or variance limits [53–55]. However,

since the robustness evaluation is not differentiable, a common approach is to synthesise solutions using mixed integer linear programming (MILP), which scales exponentially with the size of mission horizon [53, 210]. To address the inherent complexity, the robustness metric is often approximated to smooth the search space and find a solution using gradient ascent [211–213].

Most previous approaches on semantic planning under uncertainty do not take into account the effect of measurements on environmental uncertainty or task completion. A notable exception is [184], where a threshold-based information gathering strategy is used in conjunction with semantic task planning. In this approach, the robot continually computes the risk of mission failure due to uncertainty. If the risk is high, the robot visits an informative state. Another relevant work is [214], where the robot plans an informative path for surveying an area, similar to those described in Sec. 2.2.2, while subject to a temporal logic constraint.

2.4 Summary and Limitations

We reviewed relevant work on representations of environmental uncertainty, general problem formulations and specific problem instances considered in this thesis.

Environmental uncertainty can be largely categorised into that of obstacles, targets and robot's dynamics. To map obstacles, discrete and continuous representations of occupancy have been developed, and more recent approaches store distance to target. Targets can be estimated using classical filtering techniques, and in some cases, using discrete grids or a continuous function. In both cases, incorporating semantic information learnt from data allows 'imagining' the environment at varying scales. Parametric and nonparametric uncertainties in robot dynamics can be estimated using various tools. Notably, a good exciting trajectory is required to generate a good estimate.

General problem formulations include POMDPs, active perception, and Bayesian optimisation. POMDP is a general description of planning under uncertainty with measurements, and can be solved using approximate value iteration or tree search techniques. The effect

of measurements are considered, albeit through exhaustive simulation. Active perception approaches aim to maximise the information gain on environment. Planning can be achieved via submodular maximisation or tree search. In some cases, active perception problems can be solved without exhaustive simulation of sensor measurements and perception algorithms. Bayesian optimisation formulations consider maximising an *a priori* unknown function. A powerful insight is exploration-exploitation trade-off, which also seems to appear in problem-specific forms throughout robotics literature.

In this light, previous work on plume monitoring seems to lack diversity in possible plume models. Most previous work only consider parametric models derived from a particular solution of the governing physics. Target search and localisation seems divided between maximising the number of targets found, or minimising the error in position estimates. However, the two objectives seem to come together if both the quantity and location of targets are estimated using RFS statistics. Semantic task planning under uncertainty has been considered in discrete formulations, or in continuous formulations with deterministic predicates on the probabilistic belief. There is a need to extend the class of probabilistic predicates available in the continuous domain.

In the following, we summarise the findings from the literature review, and their relevance to the remainder of this thesis.

1. **Representation of uncertainty**: there are diverse representation of environmental uncertainty, with increasing complexity as semantic information is considered. Such a prior knowledge can be used to improve planning performance. The rest of this thesis will examine how to incorporate prior knowledge to improve planning.

2. **Comparison of previous formulations**: although we can pose the problem of planning under uncertainty as a POMDP, the corresponding solution methods are often computationally expensive because sensor measurements must be simulated. On the other hand, active perception algorithms only consider optimal reconstruction of the environment and does not allow for separate reward. However, they can often side-step the need to simulate sensor measurements using information-theoretic tools. Bayesian optimisation allows modelling an uncertain objective, and can be solved

via exploration-exploitation trade-off. Our approach unifies these formulations, and combines their advantages.

3. **Plume monitoring**: most previous methods consider a fixed, parametric model of plume with a spatially invariant flow field. There is a need to consider more realistic flow and concentration models. We derive a GP model that addresses this shortcoming.

4. **Target search and localisation**: most previous work considers search and localisation separately. Our approach combines the two objectives by explicitly modelling uncertainty in target location and quantity.

5. **Planning for semantic tasks**: most previous formulations either consider a discrete environment, or convert probabilistic belief into deterministic conditions through a careful choice of predicates. Moreover, the effect of measurements are not explicitly considered. We present a natural probabilistic extension of STL that permits uncertain semantics and the modelling of measurement effects.

Chapter 3

Background and Problem Formulation

Based on the ideas presented in the previous chapters, this chapter mathematically formulates robotic prospection from a Bayesian perspective. Firstly, in Sec. 3.1, we review background information on Bayesian approaches to robotic perception, with relevant example tools considered throughout this thesis. Subsequently, Sec. 3.2 formalises prospection in planning and perception. Finally, Sec. 3.3 compares prospection to previous formulations of planning and decision making under uncertainty.

3.1 Background

We first review Bayesian approaches to robotic perception. Section 3.1.1 presents general theory of Bayesian inference and relevant examples for robotic perception. Section 3.1.2 presents Gaussian process (GP), a Bayesian inference model used extensively throughout this thesis. Given the Bayesian tools for perception, Sec. 3.1.3 presents information theory, which allows quantifying environmental uncertainty and the utility of sensor measurements.

3.1.1 Bayesian Robotic Perception

Robotic perception problems are often formulated on the basis of Bayesian probability theory. In this section, we give a brief overview of Bayesian estimation, with relevant examples to robotics applications.

In the most basic sense, Bayesian estimation is formulated as follows. Suppose we are given two random variables, A and B. Then, the two random variables are described by a *joint* probability distribution $\mathcal{P}(A, B)$. The *marginal* probability of the two variables are given by integrating, or summing over the other variable. In other words, $\mathcal{P}(A) = \int \mathcal{P}(A, B)dB$ and $\mathcal{P}(B) = \int \mathcal{P}(A, B)dA$ respectively[1]. Bayesian estimation is concerned with estimating the value of A given B, or, conversely, B given A. Such a notion is described by the *conditional* probability distribution of A given B or B given A, which is defined as $\mathcal{P}(A \mid B) = \frac{\mathcal{P}(A,B)}{\mathcal{P}(B)}$ or $\mathcal{P}(B \mid A) = \frac{\mathcal{P}(A,B)}{\mathcal{P}(A)}$. Based on this definition, Bayes' theorem is given by a simple algebraic manipulation:

Theorem 3.1 (Bayes' Theorem [215, Sec. 6.6])**.** *Given two random variables A and B, the conditional distribution is given by:*

$$\mathcal{P}(A \mid B) = \frac{\mathcal{P}(B \mid A)\,\mathcal{P}(A)}{\mathcal{P}(B)}. \tag{3.1}$$

The significance of Bayes' theorem (3.1) is that it breaks down the Bayesian estimation problem into terms that can be modelled with engineering methods. In the robotics context, we are typically interested in estimating an environmental parameter E given some sensor measurements $\mathbf{Y}$. Thus, we would like to compute $\mathcal{P}(E \mid \mathbf{Y})$. Substituting E and $\mathbf{Y}$ into Theorem 3.1, we find that the relevant terms are $\mathcal{P}(\mathbf{Y} \mid E)$, $\mathcal{P}(E)$ and $\mathcal{P}(Y)$. The term $\mathcal{P}(\mathbf{Y} \mid E)$ is called the *likelihood* distribution, and describes the probability of obtaining a particular measurement $\mathbf{Y}$, had the environment been in a particular state E. For this reason, it is also called the sensor model because it describes what the sensor would output assuming some fixed environmental state. The term $\mathcal{P}(E)$ is called the *prior* distribution, which describes our *a priori* knowledge about the environment. As we will see later, the prior distribution not only concerns the particular state of the environment,

[1] For discrete random variables, the integral is replaced by a sum.

but also the correlation between the components that comprise the environment. The term $\mathcal{P}(\mathbf{Y})$ is called the measurement marginal. It is often simply treated as a normalising factor and ignored from computation, since it does not depend on the environmental parameter E. In this case, we make use of the fact that $\mathcal{P}(E \mid \mathbf{Y}) \propto \mathcal{P}(\mathbf{Y} \mid E)P(E)$, where $\propto$ means 'proportional to'.

In what follows, we give examples of how Bayesian estimation is applied in robotics applications.

Example 3.1 (Occupancy Mapping)**.** Robotic systems often build a map of the environmental obstacles in order to avoid them. Occupancy grids [67] are the most fundamental means of building such a map. In this case, the environmental parameter is a matrix of Bernoulli variables $E_{ij} \sim \text{Bernoulli}(p_{ij})$, where $E_{ij} = 0$ means cell i, j is free of obstacles and 1 means occupied. Similarly, p_{ij} is the probability of occupancy.

The measurements are also represented in terms of occupancy, such that $\mathbf{Y}_{ij} = \{0, 1\}$ means cell i, j being reported as free or occupied respectively. The sensor model $\mathcal{P}(\mathbf{Y} \mid E)$ is representative of a confusion matrix, giving the probabilities of true/false positives/negatives. Bayesian update can then be performed directly using (3.1) with the measurement marginal being computed as $\mathcal{P}(\mathbf{Y}) = \mathcal{P}(\mathbf{Y} \mid E = 0)(1 - p_{ij}) + \mathcal{P}(\mathbf{Y} \mid E = 1)p_{ij}$.

Example 3.2 (Linear Gaussian Systems)**.** Another common task of robotic systems is to estimate the position of a moving target, and possibly track it. In this case, the most common approach is to model the target as a multivariate Gaussian random variable, and fuse the measurements using Bayesian updates. Suppose the target's position and velocity are described by a multivariate Gaussian random variable $E \sim \mathcal{N}(\mu, \Sigma)$ with mean μ and covariance Σ. The robot makes measurements of the position of the target $\mathbf{Y} = CE + \epsilon_{\mathbf{Y}}$, where C is the measurement matrix that yields position and $\epsilon_{\mathbf{Y}} \sim \mathcal{N}(0, \sigma_\epsilon^2 I)$ is noise. Then, the target state and the measurement comprise a joint Gaussian random variable:

$$\mathcal{P}(E, \mathbf{Y}) = \mathcal{N}(\begin{bmatrix} \mu \\ C\mu \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma C^T \\ C\Sigma & C\Sigma C^T + \sigma_\epsilon^2 I \end{bmatrix}). \tag{3.2}$$

For joint Gaussian random variables, the Bayesian update (3.1) is given in closed form using linear algebraic operations [216]:

$$\mathcal{P}(E \mid \mathbf{Y}) = \mathcal{N}(\mu - \Sigma C^T (C\Sigma C^T + \sigma_\epsilon^2 I)^{-1}(C\mu - \mathbf{Y}), \Sigma - \Sigma C^T (C\Sigma C^T + \sigma_\epsilon^2 I)^{-1} C\Sigma). \quad (3.3)$$

3.1.2 Gaussian Processes (GPs)

GP regression [75] is a powerful Bayesian inference technique for modelling spatially correlated pheonomena. In this section, we give a brief description of GP regression, as it will be used substantially throughout the thesis.

GP regression is an extension of multivariate Gaussian random variables to *random functions*. Similar to Gaussian random variables, a GP $f(\mathbf{x}) \sim \mathrm{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ is characterised by the mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$:

$$\begin{aligned} \mathbb{E}[f(\mathbf{x})] &= m(\mathbf{x}), \\ \mathrm{Cov}(f(\mathbf{x}), f(\mathbf{x}')) &= k(\mathbf{x}, \mathbf{x}'). \end{aligned} \quad (3.4)$$

For simplicity, we only conisder the zero-mean case with $m(\mathbf{x}) = 0$.

Given a set of measurements $y_i = f(\mathbf{x}_i) + \epsilon_i$ with white noise $\epsilon_i \sim \mathcal{N}(0, \sigma_y^2)$, we would like to predict the value of the function at a remote location $f(\mathbf{x}^*)$. Let $\mathbf{Y}$ be a vector constructed by concatenating the measurements: $[\mathbf{Y}]_i = y_i$. Based on the definition (3.4), we can write $\mathbf{Y}$ and $f(\mathbf{x}^*)$ as a joint Gaussian random variable:

$$\mathcal{P}(\mathbf{Y}, f(\mathbf{x}^*)) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K + \sigma_y^2 I & \mathbf{k}^* \\ \mathbf{k}^{*T} & k^{**} \end{bmatrix}\right), \quad (3.5)$$

where $k^{**} = k(\mathbf{x}^*, \mathbf{x}^*)$ and $\mathbf{k}^* \in \mathbb{R}^N$ and $K \in \mathbb{R}^{N \times N}$ are constructed as $[\mathbf{k}^*]_i = k(\mathbf{x}_i, \mathbf{x}^*)$, and $[K]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Based on the joint random variable (3.5), the posterior distribution of $f(\mathbf{x}^*)$ given $\mathbf{Y}$ is given analogously to the linear Guassian estimation case in Example 3.2:

$$\mathcal{P}(f(\mathbf{x}^*) \mid \mathbf{Y}) = \mathcal{N}(\mathbf{k}^{*T}(K + \sigma^2 I)^{-1}\mathbf{Y}, k^{**} - \mathbf{k}^{*T}(K + \sigma^2 I)^{-1}\mathbf{k}^*). \quad (3.6)$$

An interesting extension considered throughout this thesis is when two functions described by a GP are related through a linear operator, i.e., $g(\mathbf{x}) = \mathcal{A}_{\mathbf{x}} f(\mathbf{x})$. The linear operator $\mathcal{A}_{\mathbf{x}}$ may be an integral or a partial derivative, or even a partial differential equation (PDE). In this case, the covariance between the two functions $f(\mathbf{x})$ and $g(\mathbf{x})$ are given by applying the linear operator to the covariance function [217]:

$$\begin{aligned} \mathrm{Cov}[g(\mathbf{x}), f(\mathbf{x})] &= \mathcal{A}_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}'), \\ \mathrm{Cov}[g(\mathbf{x}), g(\mathbf{x})] &= \mathcal{A}_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}') \mathcal{A}_{\mathbf{x}'}. \end{aligned} \tag{3.7}$$

The merit of the operator perspective is that the covariance functions (3.7) can be computed analytically given a choice of $k(\mathbf{x}, \mathbf{x}')$. Given the covariance relation (3.7), one can predict the value of $g(\mathbf{x})$ given noisy measurements of $f(\mathbf{x})$, and vice versa.

3.1.3 Information Theory

We are interested in how the uncertainty in environmental knowledge affects the planning performance. Information theory provides tools for quantifying uncertainty in general distributions. This section defines few fundamental information-theoretic quantities that are relevant to this thesis. For a more thorough treatise, the reader is directed to [218].

Firstly, environmental uncertainty can be quantified using the Shannon entropy $\mathbb{H}[E]$ [218, 219]:

$$\mathbb{H}[E] = -\int \mathcal{P}(E) \log \mathcal{P}(E) dE. \tag{3.8}$$

Higher entropy means higher uncertainty. In a Bernoulli variable, a probability of 0.5 yields the highest entropy, because the outcome is the most uncertain.

The uncertainty in posterior belief over E after measurements $\mathbf{Y}$ is defined similarly, and is referred to as conditional entropy or *posterior entropy*:

$$\mathbb{H}[E \mid \mathbf{Y}] = -\int \mathcal{P}(E \mid \mathbf{Y}) \log \mathcal{P}(E \mid \mathbf{Y}) \, \mathcal{P}(\mathbf{Y}) dE d\mathbf{Y}. \tag{3.9}$$

We emphasise that the posterior entropy is the *expected* value of entropy after possible measurements $\mathbf{Y}$ obtained by integrating over $\mathbf{Y}$. In effect, conditional entropy is the *expected* entropy of posterior, and is thus independent of the observed value of measurement.

The quality of measurement $\mathbf{Y}$ can be quantified as the expected reduction in uncertainty, which is referred to as *information gain*:

$$\mathbb{I}[E; \mathbf{Y}] = \mathbb{H}[E] - \mathbb{H}[E \mid \mathbf{Y}]. \tag{3.10}$$

It can be shown that information gain is symmetric, i.e., $\mathbb{I}[E; \mathbf{Y}] = \mathbb{I}[\mathbf{Y}; E]$.

The difference between two beliefs $\mathcal{P}(E)$ and $\mathcal{Q}(E)$ can be quantified using the Kullback-Leibler (KL) divergence:

$$\mathbb{D}_{KL}[\mathcal{P}, \mathcal{Q}] = \int \mathcal{P}(E)(\log \mathcal{P}(E) - \log \mathcal{Q}(E))dE. \tag{3.11}$$

A peculiar aspect of the KL divergence is that it is not symmetric, i.e., $\mathbb{D}_{KL}[\mathcal{P}, \mathcal{Q}] \neq \mathbb{D}_{KL}[\mathcal{Q}, \mathcal{P}]$.

A fundamental property of entropy is that it is always nonnegative for any probability distribution, and conditioning always decreases entropy i.e., $\mathbb{H}[E] \geq \mathbb{H}[E \mid \mathbf{Y}] \geq 0$. Therefore, the mutual information is also always nonnegative, i.e., $\mathbb{I}[E; \mathbf{Y}] \geq 0$. The KL divergence is also always nonnegative, i.e., $\mathbb{D}_{KL}[P, Q] \geq 0$.

Using Bayes' theorem (3.1), we can derive identities on the mutual information (3.10) that allow computation in various ways. These are listed below:

$$\begin{aligned} \mathbb{I}[E; \mathbf{Y}] &= \mathbb{H}[E, \mathbf{Y}] - \mathbb{H}[E] - \mathbb{H}[\mathbf{Y}] \\ &= \mathbb{H}[E] - \mathbb{H}[E \mid \mathbf{Y}] \\ &= \mathbb{H}[\mathbf{Y}] - \mathbb{H}[\mathbf{Y} \mid E] \\ &= \mathbb{E}_{\mathbf{Y}}[\mathbb{D}_{KL}[\mathcal{P}(E \mid \mathbf{Y}), \mathcal{P}(E)]]. \end{aligned} \tag{3.12}$$

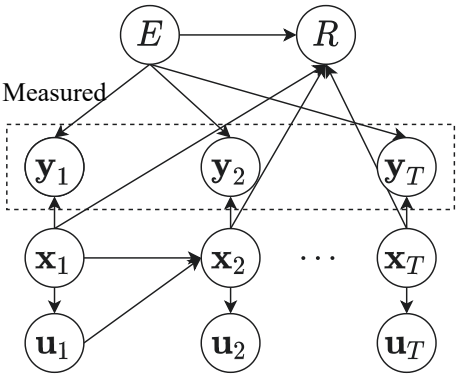


FIGURE 3.1: A probabilistic graphical model illustrating the prospective planning problem. Directed edges indicate availability of a generative model. A robot operates in an unknown environment E. The robot can visit different states, e.g., position, $\mathbf{x}_t$ with control input $\mathbf{u}_t$, e.g., velocity,. Depending on the state $\mathbf{x}_t$, the robot makes measurements $\mathbf{y}_t$ about the environment E. The robot's task is to maximise the reward R, which depends on both the environment E and the trajectory $\mathbf{X}$.

3.2 A Bayesian Formulation of Prospection

Using the tools presented in previous sections, we now formulate the problem of robotic prospection. This formulation serves as a general model for autonomous planning under environmental uncertainty considered in this thesis.

The prospective planning problem asks to find a sequence of control actions that maximise a task reward in an unknown environment. In doing so, the robot must take into account the effect of its actions on the quality of information revealed about the environment, and, in turn, its ability to successfully complete the task.

More concretely, the problem is illustrated as a graphical model in Fig. 3.1. We are given a robot, which operates in an unknown environment, denoted by E. E represents any arbitrary environmental parameters, and can follow any arbitrary distribution. It may represent, for example, the position of targets characterised by Gaussian distributions or even a random finite set, or it can represent an occupancy grid of obstacles characterised by a matrix of Bernoulli variables. We assume that the true environment is described by a distribution $\mathcal{P}^*(E)$. This encompasses the special case of a deterministic ground truth, where $\mathcal{P}^*(E)$ is Dirac delta.

We assume that the robot's dynamics is given by a deterministic model in the form of:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t), \tag{3.13}$$

where $\mathbf{x}_t$, $\mathbf{u}_t$ are the state (e.g., position, pose) and the control action (e.g., velocity, thrust) of the robot at time t.

Since E is unknown, it must be estimated from measurements. As the robot travels around, it generates measurements $\mathbf{y}_t$ that reveal information about E with a known sensor model:

$$\mathbf{y}_t \sim \mathcal{P}(\mathbf{y}_t \mid E, \mathbf{x}_t). \tag{3.14}$$

Importantly, the measurements depend on the state of the robot $\mathbf{x}_t$. For brevity, we omit the subscript to mean the set of states or measurements over time. I.e., we use $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_T\}$, and $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_T\}$.

The robot's task is modelled with a deterministic reward function $R(E, \mathbf{X})$, which depends on not only the robots state $\mathbf{X}$, but also the environmental parameter E. Our aim is to choose control inputs $\mathbf{u}$ maximising reward:

$$\max_{\mathbf{U}} R(E, \mathbf{X}). \tag{3.15}$$

We cannot solve (3.15) directly because E is a random variable whose value is unknown, and, consequently, so is $R(E, \mathbf{X})$. Therefore, it is necessary to first estimate E given the sensor measurements $\mathbf{y}_t$. We assume that this is achieved with Bayesian estimation:

$$\mathcal{P}(E \mid \mathbf{y}_{1:t}) = \mathcal{P}(E) \prod_{\tau=1}^{t} \frac{\mathcal{P}(\mathbf{y}_\tau \mid E, \mathbf{x}_\tau)}{\mathcal{P}(\mathbf{y}_\tau \mid \mathbf{x}_\tau)}, \tag{3.16}$$

where we used the conditional independence of each $\mathbf{y}_t$ given E, and the fact that the dynamics of $\mathbf{x}_t$ is deterministic.

The first problem of prospection is to accurately predict the environment E given the available measurements. Since the sensor model $\mathcal{P}(\mathbf{y}_t \mid E, \mathbf{x}_t)$ is given by the problem instance (and hence, the marginal $\mathcal{P}(\mathbf{y}_t \mid \mathbf{x}_t)$), the only available avenue for modification is the prior $\mathcal{P}(E)$. As the ground truth $\mathcal{P}(E)$ is unknown, we would like to design a prior

that works well for all possible ground truth. Thus, the general problem statement for prospective perception is as follows:

Problem 3.1 (Prospective perception)**.** For an arbitrary true environment distribution $\mathcal{P}^*(E)$, design a predictive prior $\mathcal{P}(E)$ that maximises the rate of convergence of the posterior distribution $\mathcal{P}(E \mid \mathbf{y}_{1:t})$.

We stress that the solution of Problem 3.1 is not to guess a prior $\mathcal{P}(E)$ that is already close to the ground truth $\mathcal{P}^*(E)$. Instead, the aim is to introduce an *inductive bias* so that the correlations in the elements of E are accurately modelled, based on previous data or domain knowledge. If, for example, E were multivariate Gaussian, we are more interested in accurately modelling the covariance, rather than the mean. Existing examples of such priors include GPs for scalar fields [220], which encode smoothness, and sum-product networks (SPNs) for indoor environments [90], which models common structural patterns.

Since the class of $\mathcal{P}(E)$ varies between problem instances (e.g., Gaussians or occupancy grid), we cannot have a universal prior for all possible classes of environments that appear in robotics applications. We thus focus on deriving suitable predictive priors for a few select case studies in Chapters. 5 and 7. This is achieved by incorporating physical knowledge into GPs.

Given the posterior estimate of the environment $\mathcal{P}(E \mid \mathbf{y}_{1:t})$, the remaining problem is to choose a sequence of control actions that maximise the reward upon execution. In addition to the reward function and the dynamic model as in a typical planning problem, we must consider the sensor model as the measurements depend on the control actions. As the value of reward is probabilistic due to environmental uncertainty, we aim to design an *acquisition strategy*, by which we mean any mapping that yields control actions $\mathbf{U}$ given the posterior $\mathcal{P}(E \mid \mathbf{y}_{1:t})$, and the sensor and dynamic models. The general problem statement is as follows:

Problem 3.2 (Prospective planning)**.** Design an acquisition strategy for choosing the control actions $\mathbf{U}$ that maximise the reward $R(E, \mathbf{X})$ over a time horizon T given the posterior distribution over the environment $\mathcal{P}(E \mid \mathbf{y}_{1:t})$, and the sensor (3.14) and dynamic (3.13) models.

A canonical example of an acquisition strategy is the expectimax approach [152], where the control actions are chosen to maximise the expected value of reward. Although commonly used, expectimax is not the only, nor the best strategy available. For example, the Gaussian process upper confidence bound (GP-UCB) strategy [181] uses a weighted sum of mean and standard deviation of the predicted reward to choose the control actions, which leads to better performance than expectimax.

In Ch. 4, we design a new strategy called mutual information upper confidence bound (MI-UCB) that enables prospective planning in the most general problem instances. We also use GP-UCB in Ch. 5 in conjunction with a predictive prior to realise prospection in the special problem instance of environmental monitoring.

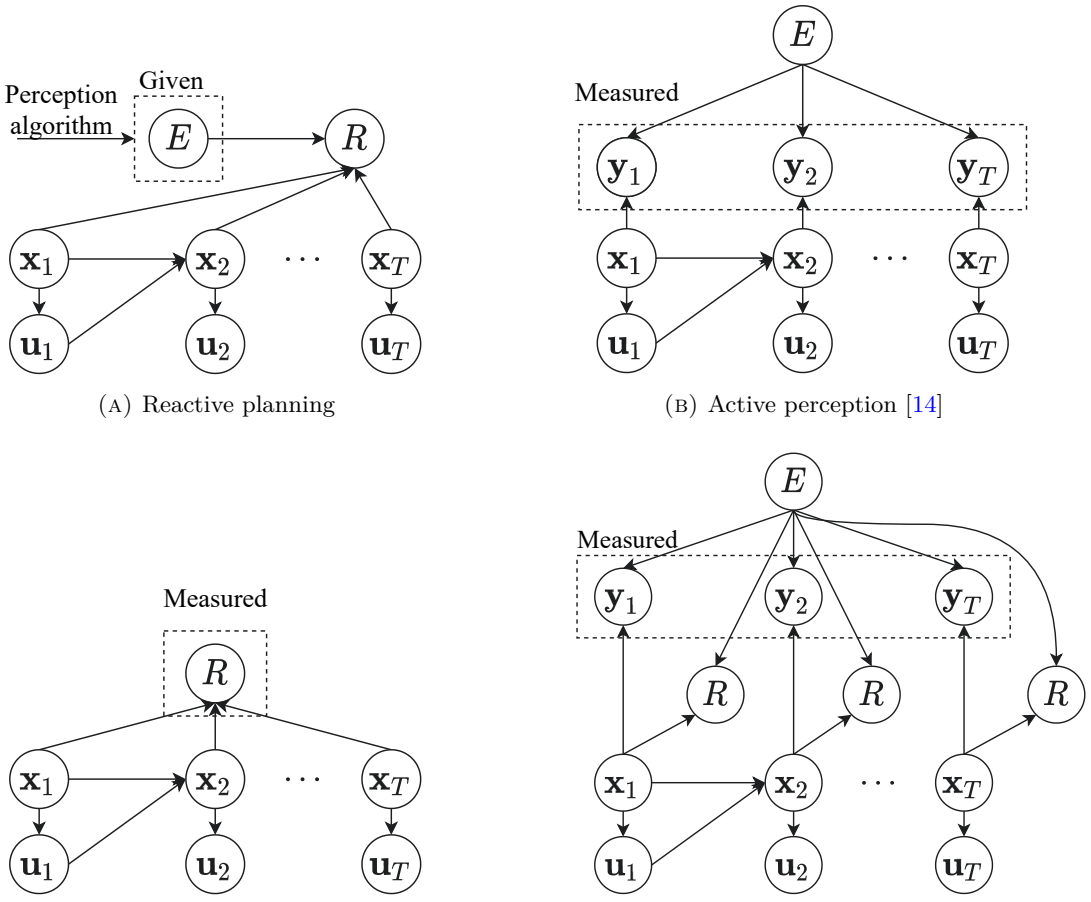
3.3 Comparison to Previous Formulations

In this section, we compare the prospective planning problem described in Sec. 3.2 to previous problem formulations for planning under uncertainty. We consider four representative formulations, which are reactive planning, active perception, MOMDP, and Bayesian optimisation, which are illustrated in Fig. 3.2. Each of these formulations shares a common element with prospective planning, while missing other important aspects. Prospective planning offers a unifying view of these formulations.

Reactive planning approaches (Fig. 3.2a) are a special case of prospective planning where the acquisition strategy is to take the environmental belief from the perception algorithm, and disregard the effect of future measurements. A common approach is *expectimax*, where one takes the expectation over the uncertain environment given current belief:

$$\max_{\mathbf{U}} \mathbb{E}[R(E, \mathbf{X})]. \tag{3.17}$$

In this case, the consideration of uncertainty may produce safe, conservative behaviours. However, reactive planning approaches treat the belief as immutable. Due to the exclusion of effect of measurements, the resulting behaviours are suboptimal when executed because the environmental belief changes with incoming measurements.



(A) Reactive planning

(B) Active perception [14]

(C) Bayesian Optimisation

(D) mixed observability Markov decision process (MOMDP) [153]

FIGURE 3.2: Comparison of related formulations as graphical models. Reactive planning formulations ignore the effect of measurements and regard environmental belief as immutable. Active perception formulations do not consider rewards other than reconstructing the environment. Bayesian optimisation is a special case of prospective planning, with no distinction between reward, environment and measurements. MOMDP formulations explicitly consider additive rewards, and planning methods typically require full simulation of perception algorithm.

An interesting formulation that does consider the effect of measurements is Bayesian optimisation, where the reward function R is unknown, and the agent tries to maximise it based on direct observations of R. This can be regarded as a special case of the prospective planning problem where the environmental parameter, reward and measurements are all merged into the reward itself. Based on this correspondence, we can borrow useful insights from the solution approaches for Bayesian optimisation problem.

Another special case that does consider the effect of measurements is MOMDP [153].

In this formulation, the robot's state $\mathbf{x}_t$ is observable, whereas the environment E is partially observable through the measurements $\mathbf{y}_t$. The MOMDP formulation explicitly adopts an acquisition strategy to maximise the expected reward of actions over possible measurements and environment instances, similar to the expectimax approach (3.17). In doing so, a technical difference to prospective planning is that the reward is explicitly additive over states, i.e., $R(E, \mathbf{X}) = \sum_t R(E, \mathbf{x}_t)$. As the acqusition strategy explicitly takes an expectation over possible measurements, the solution approaches for MOMDPs typically require eneumerating all possible measurements $\mathbf{y}_t$, and simulating the perception algorithm to construct a belief tree. Instead, prospective planning aims to design an acquisition strategy similar to Bayesian optimisation techniques, thereby avoiding the need for exhaustive sensor simulations.

The active perception formulation (Fig. 3.2b, [14, 15]) clearly differs from the prospective planning problem because there is no notion of reward. Instead, the only objective is to optimally reconstruct the environment E. Nonetheless, an encouraging observation is that many active perception approaches can side-step the need to simulate all possible measurements using information-theoretic tools described in Sec. 3.1.3.

3.4 Summary

In this chapter, we formulated the prospection problem based on a review of Bayesian perception techniques. The prospection problem comprises two subproblems: prospective perception and prospective planning. The prospective perception problem asks to find a predictive prior that speeds up Bayesian perception through accurate modelling of correlations in the environment. The prospective planning problem aims to design an acquisition strategy for choosing a sequence of control actions that yield high reward given the posterior environmental belief. This new formulation unifies several previous formulations including reactive planning, Bayesian optimisation, MOMDPs, and active perception.

In the remainder of this thesis, we solve the prospective perception and planning problems in a variety of settings. Chapter 4 offers a general solution strategy for the prospective

planning problem in isolation. Chapter 5 presents an example of both prospective perception and planning in the special problem instance of environmental monitoring. Chapter 6 considers prospective planning in complex semantic tasks. Chapter 7 presents case studies on predictive priors for prospective perception to identify what constitutes a good predictive prior.

Chapter 4

Generalised Prospective Planning with MI-UCB

The previous chapter mathematically formulated prospection in the context of perception and planning. In this chapter, we first focus on the prospective planning problem, and present the mutual information upper confidence bound (MI-UCB) as a general solution strategy.

MI-UCB states that, given a problem instance, the posterior expected reward after measurements of the environment is upper bounded by the weighted sum of information gain (i.e., exploration) and prior expectation of reward with respect to current environmental knowledge (i.e., exploitation). It significantly generalises the notion of exploration-exploitation trade-off from standard Bayesian optimisation on restricted class of objective functions to sequential decision-making agents with arbitrary environmental representations and task rewards. Inspired by active perception techniques, MI-UCB alleviates the requirement of exhaustive simulation of sensor measurements as in partially observable Markov decision processes (POMDPs) using information-theoretic tools, thereby decoupling exploratory and exploitative incentives. This allows scaling prospective planning to even heterogeneous multi-robot systems consisting of *scout* and *task* robots, so that the team can perform exploration and exploitation *simultaneously* and side-step the usual trade-off.

We first state and prove the MI-UCB in Sec. 4.1, and discuss its implication for online decision making under uncertainty. Then, in Sec. 4.2, we demonstrate how MI-UCB allows simultaneous exploration and exploitation in a scout-task team, and present a decentralised coordination algorithm for that purpose. Using these theoretical and algorithmic developments, Sec. 4.3 presents an application in a multi-drone surveillance scenario. Section 4.4 evaluates the performance characteristics of MI-UCB in the multi-drone surveillance scenario with photorealistic simulations, and Sec. 4.5 concludes the chapter. Appendix A describes practical implementation details in doing so.

This chapter builds upon our prior publications [58, 65]. In particular, the proof of MI-UCB in Sec. 4.1, and its application in a scout-task team was introduced in [58]. The multi-drone surveillance application presented in Secs. 4.3 and 4.4, as well as Appendix A, appeared in [58] and [65].

4.1 The Mutual Information Upper Confidence Bound

4.1.1 Statement and Proof

In this section, we derive MI-UCB by analysing the effect of improvement in environmental knowledge on the estimated reward. MI-UCB is motivated by attempting to maximise the *posterior* reward function:

$$\max_{\mathbf{U}} \mathbb{E}[R(E, \mathbf{X}) \mid \mathbf{Y}]. \tag{4.1}$$

Although (4.1) incorporates the effect of measurements, we cannot solve it directly because the measurements $\mathbf{Y}$ remain random variables whose values have not been observed. One may take an expectation over possible measurements similar to POMDP approaches [19, 144], but the fundamental challenge of enumerating possible measurements over possible trajectories remains.

Our finding is that we can solve (4.1) using the *principle of optimism under uncertainty* [179], by deriving an upper confidence bound (UCB) on the posterior expected reward. The exact statement is:

Theorem 4.1 (MI-UCB)**.** *Suppose $R(E, \mathbf{X})$ is a measurable function of E for all $\mathbf{X}$. With probability $\geq 1 - \delta$:*

$$\mathbb{E}[R(E, \mathbf{X}) \mid \mathbf{Y}] \leq \frac{1}{\delta}\mathbb{I}[\mathbf{Y}; E] + \log \mathbb{E}[\exp R(E, \mathbf{X}))]. \tag{4.2}$$

The most important merit of the MI-UCB (4.2) is that the posterior expected reward on the right-hand side (RHS) may be calculated before taking the measurements, whereas the left-hand side (LHS) of (4.2) cannot be calculated beforehand. Further, the UCB on the RHS of (4.2) *decouples* exploration and exploitation because it separates information gain and reward into a weighted sum. The term $\log \mathbb{E}[\exp R(E, \mathbf{X}))]$ is called the cumulant generating function (CGF) [221, Sec. 2.5] in probability theory. for which analytical expressions are often available.

Theorem 4.1 is proved by evaluating how the change from $\mathcal{P}(E)$ to $\mathcal{P}(E \mid \mathbf{Y})$ affects the estimated reward. This is captured by the seminal result of Donsker & Varadarhan on change of measure [222–224]:

Lemma 4.2 (Change of measure inequality [222])**.** *Given any measurable function $\phi : X \to \mathbb{R}$ and any two distributions $\mathcal{P}$ and $\mathcal{Q}$ on X, we have:*

$$\mathop{\mathbb{E}}_{x \sim P}[\phi(x)] \leq \mathbb{D}_{KL}[\mathcal{P}, \mathcal{Q}] + \log \mathop{\mathbb{E}}_{x \sim Q}[\exp \phi(x)], \tag{4.3}$$

where $\mathbb{D}_{KL}$ denotes the Kullback-Leibler (KL) divergence.

We also need the following fundamental inequality due to Markov, which provides a universal upper confidence bound for nonnegative random variables:

Lemma 4.3 (Markov's inequality)**.** *Given any nonnegative random variable X and a nonnegative real number $\delta > 0$, we have, with probability $\geq 1 - \delta$:*

$$X \leq \frac{1}{\delta}\mathbb{E}[X]. \tag{4.4}$$

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. Consider the change of measure inequality (Lemma 4.2, [222]) between $\mathcal{P}(E \mid \mathbf{Y})$ and $\mathcal{P}(E)$:

$$\mathbb{E}[R(E, \mathbf{X}) \mid \mathbf{Y}] \leq \mathbb{D}_{KL}[\mathcal{P}(E \mid \mathbf{Y}), \mathcal{P}(E)] + \log \mathbb{E}[\exp R(E, \mathbf{X})]. \tag{4.5}$$

The KL divergence term is still a random variable, as the measurements have not yet been taken. Applying Markov's inequality over $\mathbf{Y}$ to the KL divergence term:

$$\mathbb{E}[R(E, \mathbf{X}) \mid \mathbf{Y}] \leq \frac{1}{\delta} \mathbb{E}\left[\mathbb{D}_{KL}[\mathcal{P}(E \mid \mathbf{Y}), \mathcal{P}(E)]\right] + \log \mathbb{E}[\exp R(E, \mathbf{X})]. \tag{4.6}$$

Now, the claimed result is obtained by noting that the first term in the RHS is equivalent to mutual information using the fundamental identity (3.12). □

4.1.2 Prospective Planning via Submodular Maximisation

Based on the UCB in Theorem 4.1, the online planning framework cycles between updating the belief $\mathcal{P}(E)$ and maximising the MI-UCB (4.7). In other words, we consider the following surrogate problem:

$$\mathbf{U}^* = \arg\max_{\mathbf{U}} \mathbb{I}[E; \mathbf{Y}] + \delta \log \mathbb{E} \exp R(E, \mathbf{X}). \tag{4.7}$$

The principle of optimism under uncertainty [179] asserts that maximising the UCB (4.7) maximises the *posterior* reward function (4.1) when evaluated in hindsight. This is referred to as *regret minimisation*. In this context, [225, Lemma 3] establishes bounded regret for a similar strategy to the surrogate problem (4.7), based on a UCB that only holds in specific cases, such as tabular or linear Markov decision processes (MDPs). On the contrary, MI-UCB (4.2) is universally true, and applies to all decision-making problems of the form described in Ch. 3.

Maximising the MI-UCB (4.7) is much more computationally favourable than maximising the posterior reward function (4.1) or the expected value thereof. In addition to allowing computation without explicit sampling, MI-UCB can be maximised by using algorithms designed for active perception and information gathering discussed in Sec. 2.2.2, such

as recursive greedy [163], rapidly exploring random tree (RRT) [159], or Monte Carlo tree search (MCTS) [164, 165], without loss of their performance guarantees. This is because in many problem instances as we characterise below, the MI-UCB exhibits the property of *monotone submodularity* [161], which is the underlying assumption in active perception and information gathering algorithms. Recall that a set function $f : 2^X \to \mathbb{R}$ is monotone submodular, iff for all $A \subset B \subset 2^X$, it holds that $f(A \cup \{x\}) - f(A)) \geq f(B \cup \{x\}) - f(B)$ [161]. The problem instances where MI-UCB is monotone submodular are given as follows.

Corollary 4.4. *Suppose:*

1. *the reward function is monotone submodular; or*
2. *the reward function is additive over states, i.e.,* $R(E, \mathbf{X}) = \sum_t R(E, \mathbf{x}_t)$*, and is nonnegative* $R(E, \mathbf{x}_t) \geq 0$*.*

Then, the MI-UCB (4.7) *is monotone submodular.*

Proof. Since the measurements $\mathbf{y}_t$ are conditionally independent given the environment E and states $\mathbf{x}_t$, the information gain term $\mathbb{I}[E; \mathbf{Y}]$ is monotone submodular using [158, Proposition 1]. The claim then follows from noting that the linear combination of two monotone submodular functions remains monotone submodular, and that additive functions are a special case of monotone submodular functions. □

4.1.3 Online Planning

MCTS [226] is an increasingly popular planning approach for solving general reward maximisation problems such as submodular rewards in (4.7), owing to its asymptotic optimality and anytime property. We give a brief description of MCTS to illustrate how MI-UCB maximisation (4.7) can be solved with MCTS.

Each iteration of MCTS typically consists of four steps, which are depicted in Fig. 4.1. First, the selection step recursively chooses promising child nodes based on the sampling results so far. The selection step tries to account for future potential of each node, and tries to balance between *exploration and exploitation* of the search space. This is achieved using

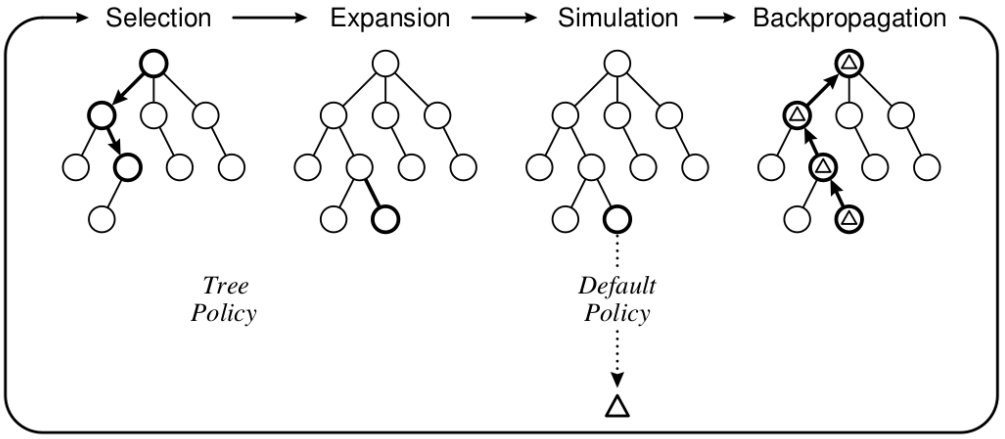


FIGURE 4.1: An iteration of MCTS. Originally appeared in [226]

selection policies such as UCB on trees (UCT) [151] and exponential weight algorithm for exploration and exploitation (EXP3) [177]. Although such selection policies also make use of UCBs, they are independent of the MI-UCB. In this case, the MI-UCB is just a reward function from the perspective of MCTS.

Subsequently, a new node is expanded, and a simulated rollout is carried out. A rollout refers to a full simulation of the problem of interest until termination under a 'default' action policy, which can be set as random, or biased towards greedy behaviour depending on the problem domain. The rollout returns the corresponding value of MI-UCB, which is backpropagated along the tree nodes. This means that MCTS can find the optimal plan that maximises the MI-UCB, as long as the value of MI-UCB can be computed for a given plan. Therefore, the algorithm designer simply needs to focus on evaluating the information gain and CGF terms in the MI-UCB (4.7) given a single path.

4.2 Simultaneous Exploration and Exploitation in a Scout-Task Team

An exciting benefit of MI-UCB is that exploration (i.e., the information gain term) and exploitation (i.e., the prior reward CGF term) are decoupled in the sense that they are simply summed together. An implication of such decoupling is that we can distribute exploration and exploitation across a functionally specialised team of robots. A subset of

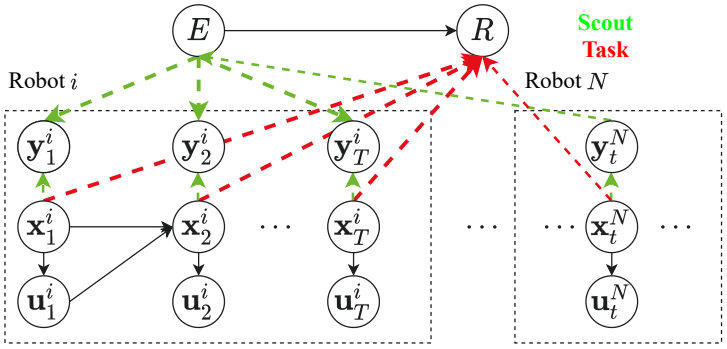


FIGURE 4.2: A probabilistic graphical model illustrating the scout–task coordination problem. Dashed connections depend on team composition. The control inputs $\mathbf{u}_t$ generates trajectory $\mathbf{x}_t$. A task robot (red) gains a reward R, depending on the trajectory $\mathbf{x}_t$ and the latent environment E. A scout robot (green) gathers measurements $\mathbf{y}_t$ at each state $\mathbf{x}_t$, revealing information about E.

the team could focus on gathering information, while the rest perform tasks. This section discusses how MI-UCB can be used to coordinate the behaviour of such heterogeneous teams that operate in unknown environments.

4.2.1 Scout-Task Coordination Problem

We define a team composition in which some robots (i.e., *task robots*) are equipped to perform a particular task whereas others (i.e., *scout robots*) are equipped with sensors to rapidly acquire knowledge about the environment as the *scout–task robot architecture*. Scout robots gathers measurements about the environment, while the task robots gain reward, as illustrated in Fig. 4.2. More concretely, we formulate the scout–task coordination problem by extending the formulation in Ch. 3 as follows.

Consider a team of N mobile robots, the dynamics of which are described by:

$$\mathbf{x}_{t+1}^r = \mathbf{f}(\mathbf{x}_t^r, \mathbf{u}_t^r), \tag{4.8}$$

where $\mathbf{x}_t^r$ and $\mathbf{u}_t^r$ are the state and control action of robot r at time t, respectively. The superscript $1 \leq r \leq N$ denotes the robot, the subscript t denotes time, and $\mathbf{u}_t^r$ is the control action applied to robot r at time t.

Each robot may belong to a set of scout robots $\mathcal{S} \subset [1, ..., N]$ or to a set of task robots $\mathcal{T} \subset [1, ..., N]$. $\mathcal{S}$ and $\mathcal{T}$ are not necessarily disjoint, and thus a robot may belong to

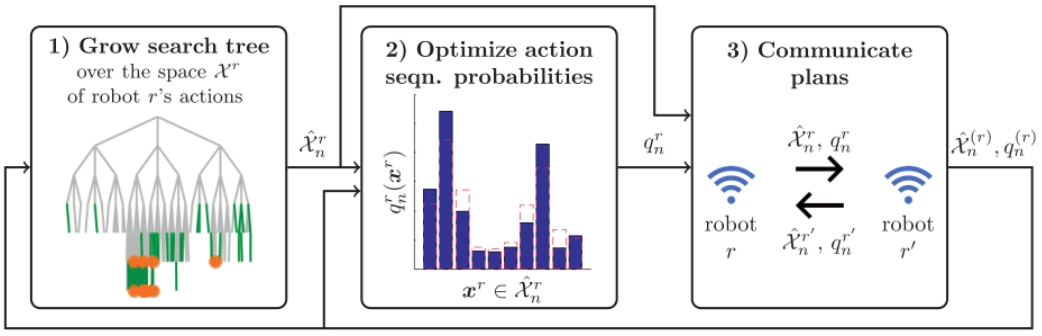


FIGURE 4.3: An illustration of Decentralised Monte Carlo tree search (Dec-MCTS). Originally appeared in [164]

both sets (i.e., it may be a scout-and-task robot). For brevity, we omit the subscript to mean the set of states over time, and omit the superscript to mean the set of states over different robots. That is, we use $\mathbf{X}^r = \{\mathbf{x}_1^r, ..., \mathbf{x}_T^r\}$, $\mathbf{X}_t = \{\mathbf{x}_t^1, ..., \mathbf{x}_t^N\}$. The omission of both subscript and superscript indicates the set of all robots' trajectories over time: i.e., $\mathbf{X} = \{\mathbf{X}^1, ..., \mathbf{X}^N\} = \{\mathbf{X}_1, ..., \mathbf{X}_T\}$. We also replace the superscript with $\mathcal{S}$ and $\mathcal{T}$ to mean the set of poses or trajectories of robots that belong to the set of scout or task robots, respectively. Hence, $\mathbf{X}^{\mathcal{T}} = \{\mathbf{X}^i \mid i \in \mathcal{T}\}$.

As in the general prospective planning problem, the robots operate in an unknown environment, denoted by E, which can follow any distribution. If $r \in \mathcal{S}$, robot r generates measurements $\mathbf{y}_t^r$ that reveal information about E:

$$\mathbf{y}_t^r \sim \mathcal{P}(\mathbf{y}_t^r \mid \mathbf{x}_t^r, E). \tag{4.9}$$

If $r \in \mathcal{T}$, robot r is equipped with a payload to perform an intended task, which depends on the environment E. We thus model the task completion by a deterministic reward function $R(E, \mathbf{X}^{\mathcal{T}})$. In doing so, the scout-only robots $r \in \mathcal{S} \setminus \mathcal{T}$ do not contribute directly to the reward function. Instead, they allow the task and scout-and-task robots $r \in \mathcal{T}$ to more effectively complete their tasks by gathering information on E.

4.2.2 Decentralised Coordination

Using the MI-UCB, we can obtain the surrogate objective as:

$$\mathbf{U}^* = \arg\max_{\mathbf{U}} \frac{1}{\delta}\mathbb{I}[\mathbf{Y}^{\mathcal{S}}; E] + \log \mathbb{E}[\exp R(E, \mathbf{X}^{\mathcal{T}})]. \tag{4.10}$$

This surrogate objective decouples scout-only and task-only robots because the respective objective terms appear as a weighted sum. As with the single-robot case in Sec. 4.1.2, for problem instances with additive or submodular rewards, MI-UCB can be maximised using multi-robot information gathering algorithms such as distributed sequential greedy assignment [227] or Dec-MCTS [164].

We use Dec-MCTS [164] to maximise the MI-UCB across the team. Dec-MCTS [164] is an asynchronous and decentralised extension of MCTS that inherits the desirable properties of MCTS while allowing scalable extension to multi-robot systems. We give a brief overview of Dec-MCTS to illustrate how the MI-UCB (4.10) can be maximised in a decentralised manner.

An iteration of Dec-MCTS consists of the following. First, each robot plans for their own actions through some iterations of a standard single-robot MCTS as described in Sec. 4.1.2 and Fig. 4.1. Then, the distribution over next possible action is compressed into a sparse representation appropriate for communication, based on product distribution approximation. Once the distribution is communicated, each robot 'simulates' other robots' action by sampling from the received probability distribution. Based on the sampled actions of other robots, each robot plans to maximise the MI-UCB (4.10) given other robots' actions. The probability distribution over action is further optimised through a variational method [228]. An illustration of these steps is shown in Fig. 4.3. As with the single-robot case in Sec. 4.1.2, the only requirement for implementation is that the MI-UCB can be evaluated given the robot team's trajectories.

4.3 Application to Multi-Drone Surveillance

We apply MI-UCB to a multi-drone surveillance problem. Targets of unknown quantity are at unknown locations, and the task is to maximise the number of targets that are visually confirmed with a short-range sensor, which thus serves as the task payload. Task drones are therefore equipped with short-range sensors only, and scout drones are equipped with long-range sensors that can rapidly provide knowledge about the environment. Drones may also be dual-equipped (i.e., they may be scout-and-task drones). Our focus will be on evaluating the information gain and prior CGF terms in the MI-UCB (4.10) so that Dec-MCTS can maximise it.

4.3.1 Reward Function

We represent the targets in a 2D occupancy grid, so that the environment is a Boolean matrix $E \in \mathbb{B}^{N_X \times N_Y}$, where N_X and N_Y are the number of cells in the X and Y directions, respectively. $E(i,j) = 1$ means cell (i,j) is occupied by a target, and 0 indicates otherwise.

We model the visibility of cell (i,j) from robot r at time t as a Bernoulli random variable $v_t^r(i,j;\mathbf{x}_t^r) \in \mathbb{B}$:

$$v_t^r(i,j;\mathbf{x}_t^r) \sim \mathcal{P}(v_t^r(i,j) \mid \mathbf{x}_t^r). \tag{4.11}$$

The visibility over a trajectory $\mathbf{X}^r$ is a disjunction $v^r(i,j;\mathbf{X}^r) = \vee_t v_t^r(i,j;\mathbf{x}_t^r)$. Similarly for the visibility over different robots, $v(i,j;\mathbf{X}) = \vee_r v^r(i,j;\mathbf{X}^r)$. Robot $r \in \mathcal{T}$ captures a target at cell (i,j) iff the target exists and is within a given radius. The reward is the number of targets captured:

$$R(E,\mathbf{X}^{\mathcal{T}}) = \sum_{ij} v(i,j;\mathbf{X}^{\mathcal{T}})E(i,j). \tag{4.12}$$

The reward function (4.12) is a sum of Bernoulli random variables, which is in turn a Poisson binomial random variable. Its CGF is given by:

$$\log \mathbb{E}[\exp R(E,\mathbf{X}^{\mathcal{T}})] = \sum_{ij} \log(1 + \mathcal{P}(d(i,j;\mathbf{X}^{\mathcal{T}}))(e-1)), \tag{4.13}$$

where $\mathcal{P}(d(i,j;\mathbf{X}^{\mathcal{T}})) = \mathcal{P}(v(i,j;\mathbf{X}^{\mathcal{T}}))\mathcal{P}(E(i,j))$.

4.3.2 Belief Update and Information Gain

We use a simple grid-based filter for decentralised data fusion of E. With the standard independence assumption, the belief over target occupancy decomposes as:

$$\mathcal{P}(E) = \prod_{i,j} \mathcal{P}(E(i,j)). \tag{4.14}$$

When a target is visible, a scout robot can measure its position. We adopt the inverse sensor model [67] approach to discretise the measurements and represent the measurement as a matrix of Bernoulli random variables:

$$\mathcal{P}(\mathbf{y}_t^r(i,j) \mid E(i,j), \mathbf{x}_t^r) = v_t^r(i,j;\mathbf{x}_t^r)\mathcal{P}(\mathbf{y}_t^r(i,j) \mid E(i,j)). \tag{4.15}$$

The sensor model $\mathcal{P}(\mathbf{y}_t^r(i,j) \mid E(i,j))$ is given by a confusion matrix between true and measured occupancy.

Each scout robot communicates its position and detected target locations (if any) at regular intervals. Measurements are fused with Bayes' rule:

$$\begin{aligned}\mathcal{P}(E(i,j) \mid \mathbf{y}_{1:t}^r(i,j)) = \Big((1 - v_t^r(i,j;\mathbf{x}_t^r)) \\ + v_t^r(i,j;\mathbf{x}_t^r)\frac{\mathcal{P}(\mathbf{y}_t^r(i,j) \mid E(i,j), \mathbf{y}_{1:t-1}^r(i,j))}{\mathcal{P}(\mathbf{y}_t^r(i,j) \mid \mathbf{y}_{1:t-1}^r(i,j))}\Big) \\ \times \mathcal{P}(E(i,j) \mid \mathbf{y}_{1:t-1}^r(i,j)).\end{aligned} \tag{4.16}$$

Information gain may be calculated as follows. For each cell, the information gain is:

$$\mathbb{I}[E(i,j); \mathbf{y}_t^r(i,j))] = H(\mathcal{P}(\mathbf{y}_t^r(i,j))) - \mathbb{E}_{E(i,j)} H(\mathcal{P}(\mathbf{y}_t^r(i,j) \mid E(i,j))), \tag{4.17}$$

where $H(p) = p\log p + (1-p)\log(1-p)$ is binary entropy and the measurement marginal is computed as $\mathcal{P}(\mathbf{y}_t^r(i,j)) = \mathbb{E}_{E(i,j)}\,\mathcal{P}(\mathbf{y}_t^r(i,j) \mid E(i,j))$. The information gain is summed

over the visible region:

$$\mathbb{I}[E; \mathbf{Y}^{\mathcal{S}}] = \sum_{ij} v(i,j; \mathbf{X}^{\mathcal{S}}) \mathbb{I}(E(i,j); \mathbf{y}_t^r(i,j)). \tag{4.18}$$

4.4 Results

We analyse the performance of MI-UCB in the context of the multi-drone surveillance problem. We first compare its performance in terms of ground-truth reward with that of a conventional expectimax approach in a simplified simulation. We then demonstrate the framework in two realistic simulated environments to examine the behaviour of MI-UCB in practical applications.

4.4.1 Comparison with Expectimax

We first compare the MI-UCB approach with the standard expectimax approach (3.17). Here, expectimax refers to maximising the expected reward, given the current belief at each stage, without accounting for information gain.

The comparison is set in the environment shown in Fig. 4.4a, where known obstacles and unknown targets are shown in black and yellow, respectively. The task is to capture targets within a given radius representing task payloads' field of effect. A scout robot may also reveal knowledge about the environment using longer-range sensors.

There are two robots: red and green. To make the comparison fair, the red robot is a scout-and-task robot, whereas the green robot is task-only. Thus, both are task robots, so the expectimax approach can generate a meaningful plan for each. If one robot were to be scout-only, the expectimax approach would not generate a plan for it, unlike MI-UCB. Intuitively, the expectimax approach simply reacts to the updates in belief, whereas MI-UCB proactively accounts for information gain associated with the belief update.

The robots start with a uniform prior, and the robots' trajectory length for each time step is fixed at 2.5 m. Each robot updates its environment belief after executing one time

step and re-plans its trajectory. We measure the performance in terms of the fraction of targets captured. In each simulation, the MI-UCB or the expectimax approach generates the robots' trajectories, while the environment is randomised by placing three targets in different locations for each run.

Combined results for ten runs of each simulation, provided in Fig. 4.4b, demonstrate that the MI-UCB approach outperforms the expectimax approach by $\sim 50\%$ in terms of the median fraction of targets captured. Examples that illustrate this trend are shown in Figs. 4.4c and 4.4d. In Fig. 4.4c, it may be observed that MI-UCB causes the red scout-and-task robot to (in effect) 'delegate' the task of capturing the target in the centre of the environment to the green task-only drone, unlike the expectimax approach used to generate the results shown in Fig. 4.4d. This is because information gain is considered in MI-UCB. Therefore, the team can maximise its utility if the red robot continues to explore and gather information, while the green robot captures the target. In contrast, in the expectimax approach, there is no incentive to do so. Thus, the red scout-and-task robot, being closer, captures the target in the centre. The higher variance of the MI-UCB approach is attributed to its optimism and shows that the upper limit of attainable reward is increased by valuing exploration.

We verify this trend in comparative performance with a four-robot experiment. As before, all drones are task drones, but the number of scout-and-task drones is varied from one to four. We fix the number of time steps and evaluate the reward per unit distance travelled by the drones in five runs for each approach, with five randomly placed targets.

As illustrated in Fig. 4.5, when there are *fewer* scout drones, MI-UCB provides greater performance benefit compared with expectimax. However, when all drones are scout-and-task drones, the approaches yield identical rewards. This implies that MI-UCB makes better use of *limited* information to provide consistent reward values with varying team composition. The performance improvement plateaus with two scouts, as shown in Fig. 4.5b, motivating the question of what team composition is *optimal* for a given problem.

4.4.2 Practical Demonstrations

Due to COVID-19 restrictions, which were enacted partway through the project, only a limited number of field trials were performed at the Royal Australian Air Force base in Point Cook, Victoria, as described in Appendix A. After restrictions were in place, experiments were performed in photorealistic simulation instead. A detailed description of the simulation framework is provided in Appendix A. To examine practical efficacy, we perturb the problem from the ideal by varying the belief over time, introducing obstacles with simultaneous mapping, and emulating sensor failure.

We implement the multi-drone surveillance framework in high-performance software based on the Robot Operating System [229]. Each drone builds its own map using Real-Time Appearance-Based Mapping [230]. The maps are combined to achieve inter-robot localisation and decentralised mapping. The grid-based filter for target estimation is implemented by use of the grid_map library [231]. A detailed description of the software framework is given in Appendix A.

The quadrotor simulation is based on the PX4 software-in-the-loop simulation [232], coupled with a modified version of the Modular Open Robotics Simulation Engine [233]. The simulation is distributed over two desktop computers, each equipped with an NVIDIA RTX2060 graphics card. The computations for each drone are executed in real-time on an NVIDIA Jetson AGX single-board computer.

We first demonstrate the framework with four drones in the environment pictured in Fig. 4.6a. It is based on an urban area near Roma Street Station, Brisbane, Australia. The 3D model is generated by use of high-altitude photogrammetry and contains a mixture of open and cluttered terrain.

Figs. 4.7a–4.7c show the target occupancy belief held by the green scout-and-task drone, as well as its intent. A target is identified at the start of the operation and is captured by the blue task drone (Fig. 4.7b). The yellow task drone captures another target in the upper-middle portion of the environment, where a parking lot is located. This behaviour is consistent with the simplified simulation, in which, when cued by a scout drone, a task drone undertakes the task instead, distributing the exploration–exploitation workload.

Throughout the mission, the blue and yellow (task) drones focus on smaller, geometrically complex areas (around the tall building and the parking lot), whereas the red and green scout-and-task drones jointly cover a larger area above the train tracks. This demonstrates MI-UCB's inheritance from and generalisation of heterogeneous information gathering.

The difference between the two approaches is that MI-UCB results in all targets eventually being captured, thus completing the intended task. A heterogeneous information-gathering approach would accept the long-range sensor's coverage of a target and not require that a short-range sensor capture it. On the other hand, our decision-making under uncertainty approach allows the practitioner to specify that it is imperative to capture targets with a task drone. This provides great flexibility, as one can easily replace the task of capturing targets with, e.g., payload delivery or casualty evacuation, each of which requires proximity.

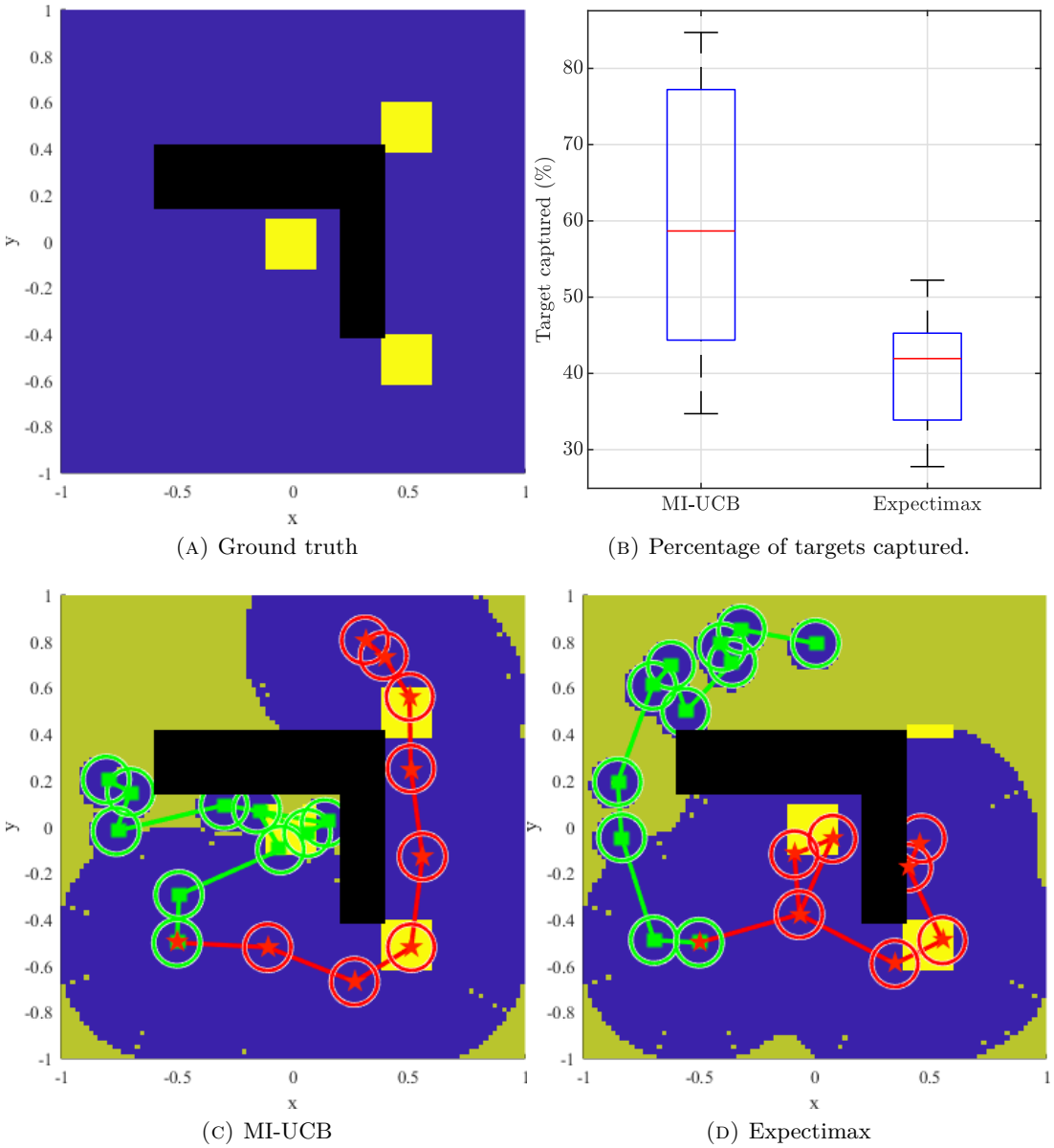
We also demonstrate the framework in the environment shown in Fig. 4.6b, which is modelled on Royal Australian Air Force Base, Point Cook, Australia. The environment is prepared by modelling the buildings from a satellite image. It creates an interesting scenario for low-altitude operations because of the alleyways and corners that limit full visibility. In this simulation, we emulate perception failures to examine their effect on the performance of the algorithm. For example, in Fig. 4.7d, the blue scout-and-task drone has capturing a target, but that is not reported to the other drones, whereas in Fig. 4.7e, the same occurs with the yellow drone. Despite these unmodelled failures, the algorithm adapts to the change and successfully captures all targets eventually, as illustrated in Fig. 4.7f.

4.5 Summary

This chapter presented MI-UCB as a general solution to the prospective planning problem. MI-UCB applies to any Bayesian agent that tries to maximise a reward based on an environmental belief maintained using sensor measurements. It states that one can approximately maximise the posterior reward after measurement, without knowing the value of measurements ahead, by selecting actions that yield the highest weighted sum of information gain and prior expected reward. MI-UCB is significantly more computationally

favourable than previous approaches for planning under uncertainty such as POMDPs, and can be solved via algorithms for active perception and information gathering in a wide class of problems. The decoupling of exploration and exploitation in MI-UCB allows distributing exploration and exploitation across a heterogeneous multi-robot team consisting of scout and task robots. The combination of MI-UCB and Dec-MCTS allows coordinating such a scout–task robot team to undertake exploration and exploitation simultaneously and synergistically. The experiments demonstrate that using MI-UCB leads to increased task performance in hindsight compared to the expectimax approach, where one simply maximises the expected reward given the current belief. These results indicate that prospective planning offers substantial performance improvements in practical problem instances where the surrounding environment is unknown.

The example presented in this chapter considers only a simple reward function, and also lacks prospective perception because only a simple prior is used. In the remainder of this thesis, we will examine prospective planning in tandem with predictive priors, and with more complex tasks.



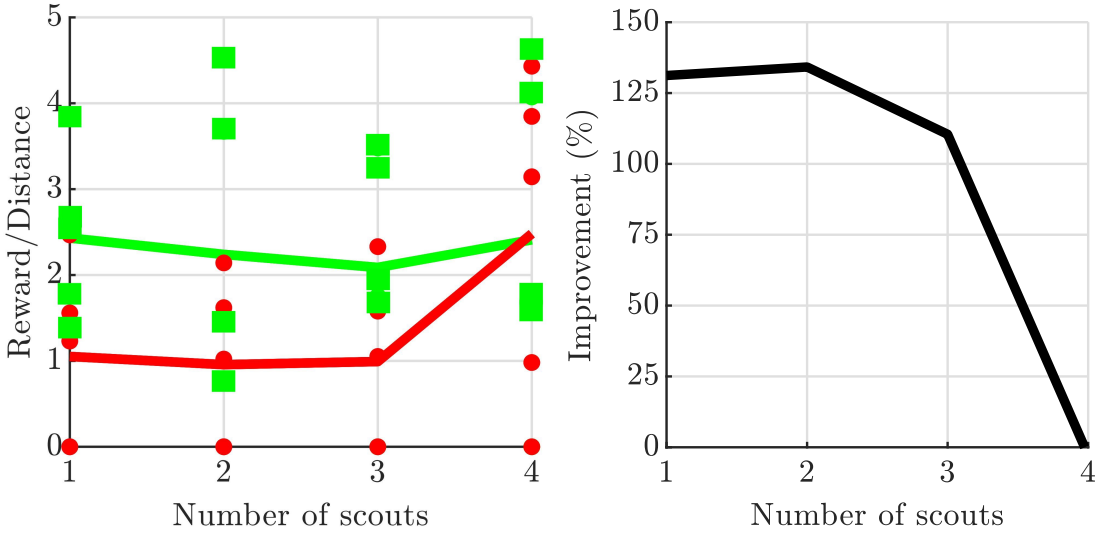
(A) Ground truth

(B) Percentage of targets captured.

(C) MI-UCB

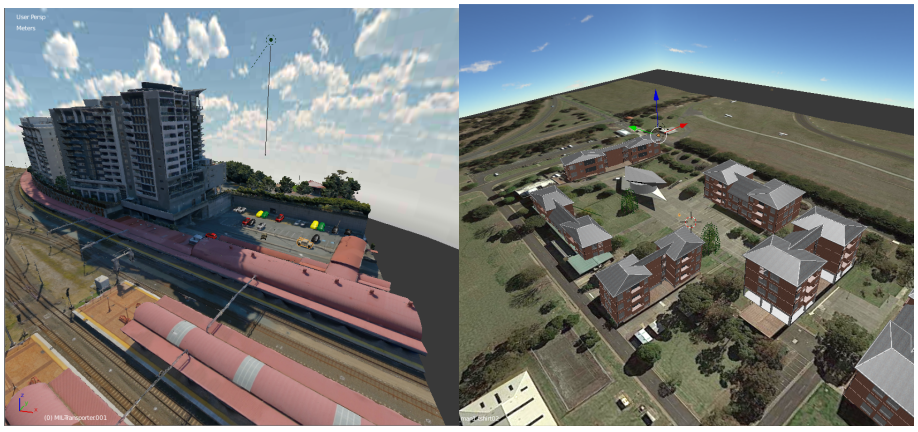
(D) Expectimax

FIGURE 4.4: Comparison between MI-UCB and expectimax in a simplified scenario. A robot captures a target (yellow) if it is within the circle representing its task payloads' field of effect (red stars for a scout-and-task robot equipped with task payload and long-range sensor and green squares for a task-only robot, equipped only with task payload). The colourmap shows the belief on target occupancy (increasing from blue to yellow), while black areas indicate obstacles and yellow-green areas are yet to be explored. MI-UCB (c) outperforms expectimax (d), because the former accounts for the fact that the task-only robot can provide greater information gain than the green robot.



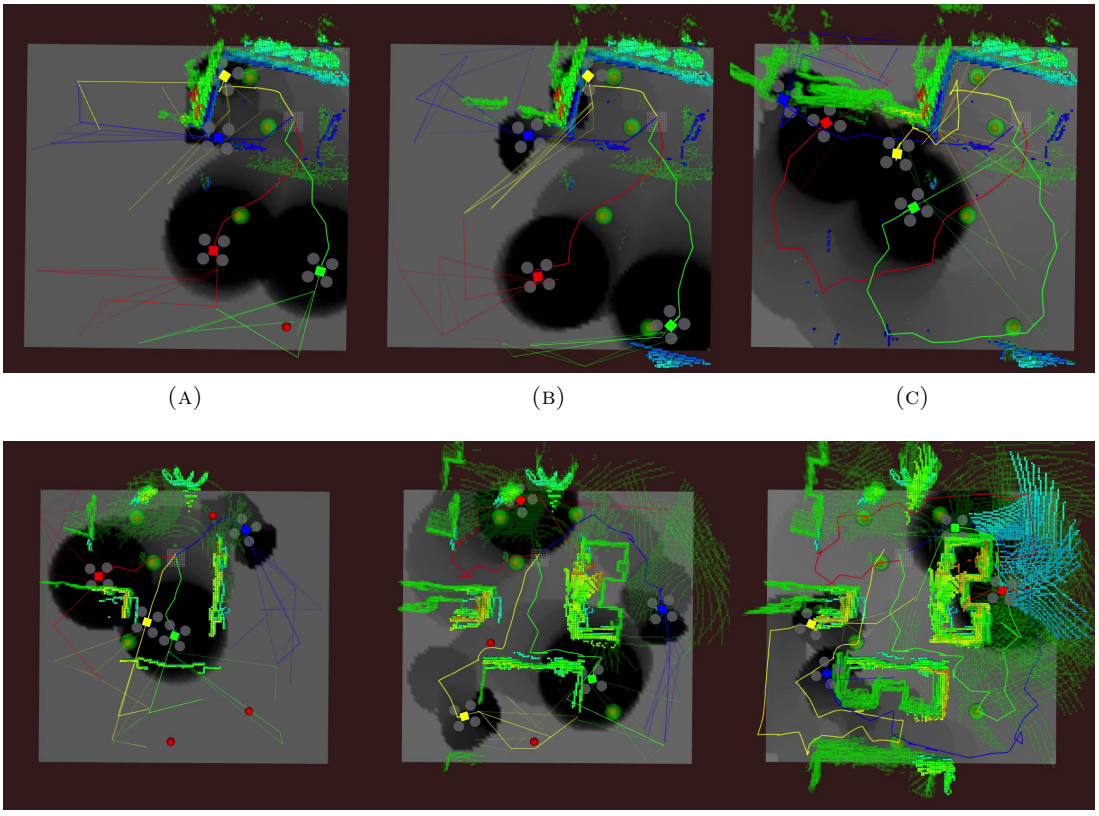
(A) Reward per unit distance (B) Performance improvement

FIGURE 4.5: Comparison of reward between MI-UCB and Expectimax. (a) Reward obtained with MI-UCB (green squares) and expectimax (red circles) as a function of the number of scouts in the four-robot scenario. (b) MI-UCB provides the most benefit compared with expectimax with two scouts, an improvement of 134%, and converges to equivalence with expectimax when four scouts are used.



(A) Brisbane (B) Point Cook

FIGURE 4.6: Environments used for high-fidelity simulation



(A) (B) (C)

(D) (E) (F)

FIGURE 4.7: High-fidelity simulation results for decentralised intelligence, surveillance and reconnaissance (ISR). Time runs horizontally. Red spheres are targets yet to be captured, and green are those captured. The grey colourmap shows the robot's belief over target occupancy.

Chapter 5

Prospection for Environmental Process Monitoring

In the previous chapter, we considered prospective planning in isolation in the most general setting. This chapter demonstrates a complete example of prospection in both planning and perception, restricting our attention to the special case of environmental monitoring. We consider an underwater robot surveying an environmental process under the influence of an ambient flow field, such as chlorophyll in water [234], as well as temperature [235] and methane [236] in air and water. The aim is for the robot to reach the source of the environmental activity.

To realise prospective perception, we develop the advection-diffusion Gaussian process (ADGP), a specialised Gaussian process (GP) model that incorporates the advection-diffusion partial differential equation (PDE) on chemical plumes from fluid dynamics. ADGP allows the estimation of source location and strength given sparse measurements of the environmental process. In this problem instance, prospective planning can be achieved with Bayesian optimisation discussed in Sec. 2.2.3. We combine a Bayesian optimisation strategy called Gaussian process upper confidence bound (GP-UCB) with a trim-based fast marching tree (FMT)* planner [42, 237] for use in underwater gliders.

Empirical results on simulated and experimental datasets illustrate that the use of both prospective perception and planning leads to efficient behaviour, such as prioritising cross-

or downstream regions if the current concentration measurement (i.e. the source strength of upstream region) is low. We attribute such an efficient behaviour to the combination of prospective planning and perception. Prospective planning with Bayesian optimisation permits reaching the source through balance of exploration and exploitation. Prospective perception with ADGP expedites the process through the use of inductive bias afforded by physical knowledge. Theoretical analysis also shows that the robot should reach the source efficiently with high probability. This chapter builds upon our prior publication in [64].

5.1 Problem Formulation

We consider a marine robot under the influence of a fully known, spatially varying flow field $\mathbf{w}(\mathbf{x})$. The dynamics of such a robot is given in the form of:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{w}(\mathbf{x}(t), t), \tag{5.1}$$

where $\mathbf{f}$ represents the mechanical dynamics of the glider relative to water, $\mathbf{x}(t)$ is the state at time t, $\mathbf{u}(t)$ is the control vector, and $\mathbf{w}(\mathbf{x}, t)$ is the ocean current at the glider's position.

The robot surveys an environmental process, such as a methane plume [33], quantified in terms of its concentration $c(\mathbf{x}, t)$. Concentration is driven by a source strength function $s(\mathbf{x}, t)$, which describes the amount of chemical substance entering or leaving the environment at position $\mathbf{x}$ and time t. The robot measures the concentration $c(\mathbf{x}, t)$ at a point using an onboard sensor, modelled as:

$$y = c(\mathbf{x}, t) + \epsilon, \tag{5.2}$$

where $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$ is white Gaussian noise with known variance σ_y^2. We will denote a set of observations as $\mathcal{O} = \{\mathbf{x}_i, y_i\}_{i=1}^N$.

Given the source strength function, we model the source location by:

$$\mathbf{x}_s = \arg\max_{\mathbf{x} \in \mathcal{E}} s(\mathbf{x}, t), \tag{5.3}$$

where $\mathcal{E}$ is the operating environment of the glider.

As the source strength over the environment is initially unknown, the overall problem necessitates two main sub-problems: *estimation* and *planning*. The estimation problem is to predict the source strength over the environment given sparse measurements of the plume concentration. To do so, we exploit the physical relationship between source strength and plume concentration, namely, the advection-diffusion PDE.

Based on the uncertain estimate of the source strength, the planning problem is to find a sequence of control actions that is energy-optimal and leads the glider to the source in a *probabilistic sense*. Because the true source strength is unknown, and the estimation of the source strength also depends on the actions taken, the planner must also consider the uncertainty of the estimated source strength, and encourage increase in information gain.

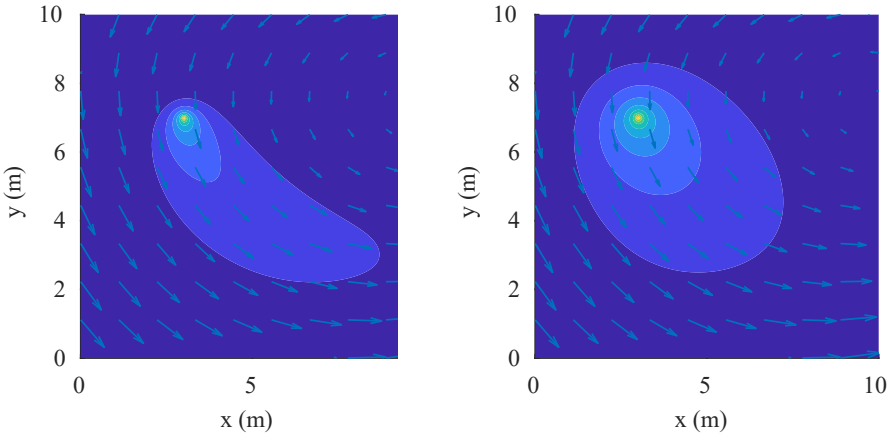
5.2 Advection-Diffusion Gaussian Process for Environmental Processes

The estimation part of the framework infers the source location by predicting the source strength given concentration measurements. It consists of two components: *concentration estimation* and *source localisation*.

5.2.1 Advection-Diffusion PDE

Advection-diffusion equation [238] describes the relation between source and plume under the influence of a flow field. Advection refers to the transport of plume particles by the flow field, and diffusion describes the natural dispersion in still water.

The addition or removal of a plume substance into the environment is modelled by a scalar function $s(\mathbf{x}, t)$, referred to as the *source strength*. The source strength $s(\mathbf{x}, t)$ is positive if the substance is being introduced into the environment at position $\mathbf{x}$ and time t, and negative if being removed. Given the source strength, $s(\mathbf{x}, t)$, the advection-diffusion PDE



(A) Advection-dominant (B) Diffusion-dominant

FIGURE 5.1: Examples of 2D advection-diffusion process

Examples of 2D advection-diffusion process, showing different components of the advection-diffusion PDE. The advection-dominant example (a) shows greater alignment between the directions of the flow and the plume than the diffusion-dominant one (b).

describes the evolution of concentration of a substance, $c(\mathbf{x}, t)$ as:

$$\frac{\partial c(\mathbf{x}, t)}{\partial t} + \mathbf{w}(\mathbf{x}, t) \cdot \nabla c(\mathbf{x}, t) = D\Delta^2 c(\mathbf{x}, t) + s(\mathbf{x}, t), \tag{5.4}$$

where ∇ is the gradient operator, Δ^2 is the Laplacian operator, and $\mathbf{w}$ is the flow field. D is the diffusion coefficient, which, for example, ranges between $D = 1 - 5m^2/s$ for the methane-water system depending on the temperature [236].

Assuming $s(\mathbf{x}, t)$ and $\mathbf{w}(\mathbf{x}, t)$ are time invariant, the plume process reaches its steady-state. Based on (5.4), the steady-state equation is obtained by setting the time derivative to zero:

$$\mathbf{w}(\mathbf{x}) \cdot \nabla c(\mathbf{x}) = D\Delta^2 c(\mathbf{x}) + s(\mathbf{x}). \tag{5.5}$$

where we omit time as $c(\mathbf{x}) = \lim_{t\to\infty} c(\mathbf{x}, t)$ with a slight abuse of notation.

If $\mathbf{w}(\mathbf{x})$ does not vary across space, it is well-known that (5.5) admits the so-called Gaussian plume solution [238]. However, it is practically impossible to find an analytical solution in the general case when $\mathbf{w}(\mathbf{x})$ varies spatially, and (5.5) needs to be solved numerically using, e.g., a finite difference method. Examples of such a solution in 2D are shown in

Fig. 5.1. In Fig. 5.1, it can be seen that the chemical substance is not only advected along the flow, but also diffused in directions against the flow.

5.2.2 Advection-Diffusion Gaussian Process

The plume concentration at point $\mathbf{x}$ modelled as a GP is denoted as $c(\mathbf{x})$. Formally,

$$c(\mathbf{x}) \sim GP(0, k_{cc}(\mathbf{x}, \mathbf{x}')), \tag{5.6}$$

where $k_{cc}(\mathbf{x}, \mathbf{x}')$ is a kernel function. Since the fundamental solution to (5.5) is given by a Gaussian plume [238], we use the squared exponential (SE) kernel to model the concentration function:

$$k_{cc}(\mathbf{x}, \mathbf{x}') = k_{SE}(\mathbf{x}, \mathbf{x}') \equiv \sigma_c^2 \exp\left(-\frac{\gamma}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right), \tag{5.7}$$

where σ_c^2 is a self-variance and γ is the length-scale parameter. Using the covariance function (5.7), the concentration at an arbitrary query point $c(\mathbf{x}^\star)$ can be predicted using the standard equation (3.6).

Although we can now estimate concentration at an arbitrary point, our main interest is in estimating the *source* term, given concentration measurements. To do so, we exploit the advection-diffusion PDE (5.5) to estimate the source strength. This can be achieved by considering the PDE as an operator on the vector space of functions, as is considered in [239] for other PDEs. The advection-diffusion PDE (5.5) can also be written as an operator:

$$\mathcal{D}_{\mathbf{x}} \equiv (\mathbf{w}(\mathbf{x}) \cdot \nabla - D\Delta^2). \tag{5.8}$$

By rearranging (5.5), we find that $s(\mathbf{x}) = \mathcal{D}_{\mathbf{x}} c(\mathbf{x})$. In other words, we can recover the source strength from the concentration by applying $\mathcal{D}_{\mathbf{x}}$ on $c(\mathbf{x})$. Using this property of $\mathcal{D}_{\mathbf{x}}$, it can be shown that the kernel function for $s(\mathbf{x})$ and the cross-covariance functions are

given by applying $\mathcal{D}_{\mathbf{x}}$ as:

$$k_{sc}(\mathbf{x}, \mathbf{x}') = Cov(s(\mathbf{x}), c(\mathbf{x}')) = \mathcal{D}_{\mathbf{x}} k_{cc}(\mathbf{x}, \mathbf{x}'), \tag{5.9}$$

$$k_{cs}(\mathbf{x}, \mathbf{x}') = Cov(c(\mathbf{x}), s(\mathbf{x}')) = k_{cc}(\mathbf{x}, \mathbf{x}')\mathcal{D}_{\mathbf{x}'}, \tag{5.10}$$

$$k_{ss}(\mathbf{x}, \mathbf{x}') = Cov(s(\mathbf{x}), s(\mathbf{x}')) = \mathcal{D}_{\mathbf{x}} k_{cc}(\mathbf{x}, \mathbf{x}')\mathcal{D}_{\mathbf{x}'}. \tag{5.11}$$

A significant merit of using the cross-covariance kernels in (5.9), (5.10) and (5.11) is that they can be calculated analytically for a given choice of kernel $k_{cc}(\mathbf{x}, \mathbf{x}')$ for concentration. For instance, assuming SE kernel (5.7), the cross-covariance function is given by:

$$k_{sc}(\mathbf{x}, \mathbf{x}') = (D(2\gamma - \gamma^2 \|\mathbf{x}\|^2) + \gamma \mathbf{w}(\mathbf{x})^T (\mathbf{x} - \mathbf{x}'))k_{SE}(\mathbf{x}, \mathbf{x}'). \tag{5.12}$$

Using the cross-covariance kernel functions (5.9), (5.10) and (5.11), we can rewrite the concentration $c(\mathbf{x})$ and source strength $s(\mathbf{x})$ as a joint GP [239]:

$$\begin{bmatrix} c(\mathbf{x}) \\ s(\mathbf{x}) \end{bmatrix} \sim GP\left(\mathbf{0}, \begin{bmatrix} k_{cc}(\mathbf{x}, \mathbf{x}') & k_{cs}(\mathbf{x}, \mathbf{x}') \\ k_{sc}(\mathbf{x}, \mathbf{x}') & k_{ss}(\mathbf{x}, \mathbf{x}') \end{bmatrix}\right). \tag{5.13}$$

Using (5.13), we get a joint random variable for observation $\mathbf{y}_{\mathcal{O}}$ and the source strength $s(\mathbf{x}^*)$ at a query point $\mathbf{x}^*$ denoted as

$$\begin{bmatrix} \mathbf{y}_{\mathcal{O}} \\ s(\mathbf{x}^*) \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K_{cc} + \sigma_y^2 I & \mathbf{k}_{cs}^* \\ \mathbf{k}_{sc}^{*T} & k_{ss}^{**} \end{bmatrix}\right). \tag{5.14}$$

where $k_{ss}^{**} \in \mathbb{R}$, $\mathbf{k}_{sc}^* \in \mathbb{R}^N$, and $\mathbf{k}_{cs}^* \in \mathbb{R}^N$ are given by:

$$\begin{aligned} k_{ss}^{**} &= k_{ss}(\mathbf{x}^*, \mathbf{x}^*), \\ \mathbf{k}_{sc}^* &= \begin{bmatrix} k_{sc}(\mathbf{x}^*, \mathbf{x}_1) & \dots & k_{sc}(\mathbf{x}^*, \mathbf{x}_N) \end{bmatrix}^T, \\ \mathbf{k}_{cs}^* &= \begin{bmatrix} k_{cs}(\mathbf{x}_1, \mathbf{x}^*) & \dots & k_{cs}(\mathbf{x}_N, \mathbf{x}^*) \end{bmatrix}^T. \end{aligned}$$

The source strength at the query point can then be predicted as the conditional distribution [75]:

$$\mathcal{P}(s(\mathbf{x}^*) \mid \mathcal{O}) = \mathcal{N}(\mu_s(\mathbf{x}^*), \sigma_s^2(\mathbf{x}^*)), \tag{5.15}$$

where the mean and variance are

$$\begin{aligned} \mu_s(\mathbf{x}^*) &= \mathbf{k}_{sc}^{*T}(K_{cc} + \sigma_y^2 I)^{-1}\mathbf{y}_{\mathcal{O}}, \\ \sigma_s^2(\mathbf{x}^*) &= k_{ss}^{**} - \mathbf{k}_{sc}^{*T}(K_{cc} + \sigma_y^2 I)^{-1}\mathbf{k}_{cs}^{*}. \end{aligned} \tag{5.16}$$

5.3 Prospective Planning for Source Seeking

5.3.1 GP-UCB Strategy

Given the GP model of source strength in Sec. 5.2, we find control vectors for finding the plume source. We first find the best next sampling waypoint given the source strength estimate, and find a continuous energy-optimal trajectory under the influence of ocean currents.

Although the knowledge about the source is updated with concentration measurements over time, the knowledge is still uncertain. This leads to the *exploration-exploitation dilemma*. When there is not enough information about the source, the glider must *explore* the environment before it *exploits* the information.

In this section, we present a solution to the exploration-exploitation dilemma with GP-UCB. The GP-UCB strategy adopts the principle of optimism under uncertainty, and picks the points that maximise the greatest value possible given the measurements so far (i.e., the upper confidence bound from posterior distribution). Another interpretation of the GP-UCB is as a weighted sum of predictive mean and standard deviation, which achieves a balance between exploitation (i.e., the mean) and the exploration (i.e., the variance).

In our case, we pick the next waypoint $\mathbf{x}_{k+1}$ as the one that maximises the following acquisition function:

$$\mathbf{x}_{k+1} = \underset{\mathbf{x}\in\mathcal{N}(\mathbf{x}_k)}{\arg\max}[\mu_s(\mathbf{x}) + \beta_k \sigma_s(\mathbf{x})], \tag{5.17}$$

Algorithm 1 GP-UCB-based Hierarchical Planner

1: GP ← InitialiseEmptyGP()
2: **while** Glider is operational **do**
3: **if** Previous dive $\mathbf{x}_k$ completed **then**
4: Update GP with measurements $\mathcal{O}_k$.
5: $\mathcal{N}_k \leftarrow$ GetNeighbours($\mathbf{x}_k$)
6: **for** $\mathbf{x} \in \mathcal{N}_k$ **do**
7: $\{\mu_s(\mathbf{x}), \sigma_s(\mathbf{x})^2\} \leftarrow$ GP.PredictSource($\mathbf{x}$)
8: **repeat**
9: $\mathbf{x}_{k+1} \leftarrow \arg\max_{\mathbf{x} \in \mathcal{N}_k} \mu_s(\mathbf{x}) + \beta_k \sigma_s(\mathbf{x})$
10: **until** $\mathbf{x}_{k+1}$ is feasible
11: $\mathbf{u}(t) \leftarrow \text{FMT}^*(\mathbf{x}_k, \mathbf{x}_{k+1})$.

where $\mathcal{N}(\mathbf{x}_k)$ is a finite set of potential next sampling locations, μ_s and σ_s are the predictive mean and standard deviation of the source term obtained from the GP after k samples, and β_k is a tuning parameter that balances between exploration and exploitation. As we will see later, β_k plays an important role in guaranteeing finding the source. Because we consider a finite set of potential next sampling location, the acquisition function can be maximised by direct sampling.

The pseudocode for GP-UCB-based hierarchical planner is shown in Alg. 1. We first sample the candidate dive locations in line 5, and predict the source strength at these candidate points (line 7). We pick the next dive location as the one that is feasible and maximises the GP-UCB acquisition function. After we find the best next sampling waypoint $\mathbf{x}_{k+1}$, we find the glider controls from $\mathbf{x}_k$ using the FMT* algorithm [42, 237].

5.3.2 Regret Analysis

In an active perception task where a decision is made based on uncertain information, the decision may not be correct. The performance loss due to making a wrong decision is referred to as *regret* [181].

In the context of our source localisation task, regret is the disparity between sampling locations $\mathbf{x}_{k+1}$ and the true source location $\mathbf{x}_s$ in terms of source strength. We consider

the *average regret* $\bar{R}_k$ over the sequence of waypoints up to the k-th time step:

$$\bar{R}_k = \frac{1}{k}\sum_{i=1}^{k} s(\mathbf{x}_s) - s(\mathbf{x}_i). \tag{5.18}$$

The average regret approaches zero asymptotically with high probability. Intuitively, this is because the GP describes the correlation between observations and source strength, and ensures that the estimated source strength converges to the ground truth. In other words, using the source GP to estimate the source location becomes increasingly accurate. The convergence behaviour is satisfied with a particular choice of tuning parameter β_k. The following theorem provides the formal statement.

Theorem 5.1. *Pick $\delta \in (0, 1)$, and let*

$$\beta_k = 2\log(\frac{\pi^2 k^2 N_s}{6\delta}), \tag{5.19}$$

where N_s is the size of potential sampling locations. Suppose we pick the sampling locations as per (5.17)*. Then,*

$$\lim_{k\to\infty} \bar{R}_k = 0, \tag{5.20}$$

holds with probability greater than $1 - \delta$.

Proof. The theorem is an application of Theorems 1 and 5 in [181]. By Theorems 1 and 5 in [181], we have $\bar{R}_k \leq \sqrt{\frac{C_1\beta_k\gamma_k}{k}}$ for C_1 a constant with probability $> 1 - \delta$, and $\gamma_k = O(\log(k)^3)$. In other words, $\exists k_0 \in \mathbb{N}$ and $\exists M_1 > 0$ such that $\forall k > k_0$, we have $\gamma_k < M\log(k)^3$. Thus, it holds that $\bar{R}_k \leq M_2\sqrt{\frac{\beta_k \log k^3}{k}}$. Because $\sqrt{\frac{\beta_k \log k^3}{k}} \to 0$ as $k \to \infty$, the claim holds by the sandwich theorem [240, Lemma 2.2.1] □

Intuitively, by Theorem 5.1, the GP-UCB-based planner guarantees that the true source location is eventually visited.

5.4 Realisation in Underwater Gliders

Underwater gliders [36] are an excellent platform for large-scale survey of environmental processes owing to their unparalleled power efficiency, which allows extended mission duration and coverage of wide areas. The efficiency is attributed to their dynamics; underwater gliders generate forward velocity by using the lift and drag forces from the diving and surfacing motions. These forces are dependent on the glider's angle of attack, which is controlled by a ballast mass and a buoyancy pump.

Although such a dynamics allows power efficient operation, the nonlinear coupling between the vertical and forward motions makes it difficult to directly solve for a sequence of controls that complies with the GP-UCB strategy. Thus, we solve the planning problem in a hierarchical manner by introducing an additional trajectory planner. Taking the 2D waypoints generated by the GP-UCB strategy as the dive-location, the trajectory planner plans an energy-optimal path through these waypoints using the trim-based FMT* framework [42, 237]. A brief summary of the trim-based planning framework is presented in the following.

A *trim-state* is a dynamic equilibrium of the vehicle dynamics (5.1) in the absence of disturbances or variations in control input [241]. Because an underwater glider covers a large area over a long period of time, we pose a computationally convenient approximation that the glider operates in a sequence of trim states with negligible transition times. For an underwater glider, the trim-state is characterised by:

$$\mathbf{u}(t) = \begin{bmatrix} V(t) & \gamma(t) & \delta(t) & m_b(t) \end{bmatrix}^T, \tag{5.21}$$

where $V(t)$ is the speed, $\gamma(t)$ is the glide angle, $\delta(t)$ is the heading angle, and $m_b(t)$ is the ballast mass (i.e., amount of water in the ballast tank). Within each trim state, the dynamic model (5.1) reduces to a kinematic model:

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \left(\begin{bmatrix} V_t \cos\gamma_t \cos\delta_t \\ V_t \cos\gamma_t \sin\delta_t \\ V_t \sin\gamma_t \end{bmatrix} + \mathbf{w}(\mathbf{x}(t)) \right) \Delta t. \tag{5.22}$$

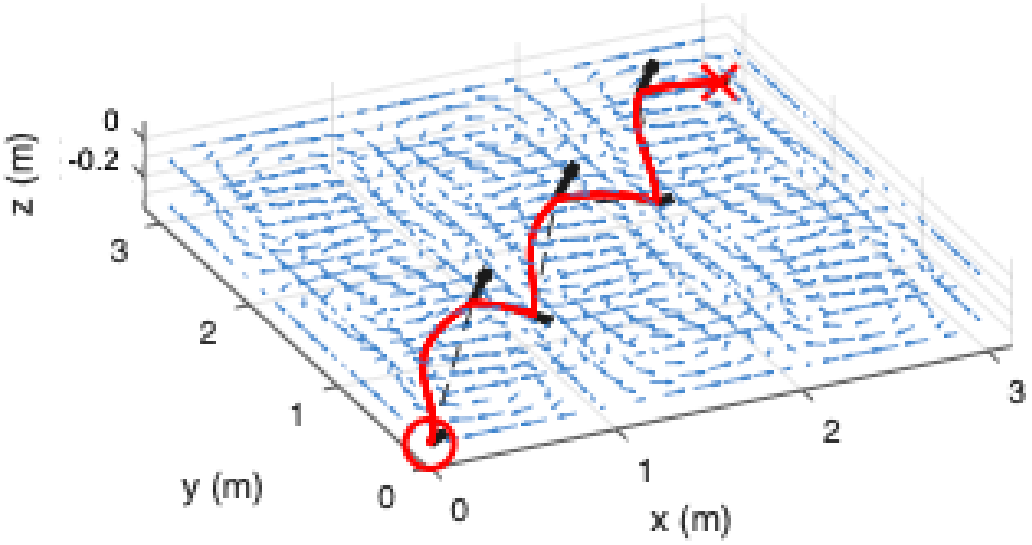


FIGURE 5.2: Trim-based glider manoeuvre over 3D environment under the influence of ocean currents

The FMT* planner proposed in [42] generates a continuous, energy optimal trajectory for following the 2D waypoint from the GP-UCB strategy by finding a sequence of trim states. In other words, given each 2D waypoint $\mathbf{x}_{k+1}$, the FMT* planner solves the following problem:

$$\begin{aligned} &\underset{u(t)}{\text{minimise}} \quad \sum_{t=1}^{T} E(\mathbf{x}(t), \mathbf{u}(t)), \\ &\text{subject to} \quad \mathbf{x}(1) = \mathbf{x}_k, \mathbf{x}(T) = \mathbf{x}_{k+1} \text{ and (5.22).} \end{aligned} \tag{5.23}$$

where $E(\mathbf{x}(t), \mathbf{u}(t))$ is the energy cost function, accounting for the sum of the ballast pump cost, moving mass re-position cost, turning cost, and the hotel cost. An example trajectory is shown in Fig. 5.2. Details on the FMT* planning algorithm for solving (5.23) is presented in [42].

5.5 Case Studies

We present case studies to demonstrate the behaviour of the GP-UCB strategy in simulated environment and real dataset. In both studies, the robot is initially given no prior knowledge about the true source location $\mathbf{x}_s$. We show that the glider actively balances between exploration and exploitation to reach the source location.

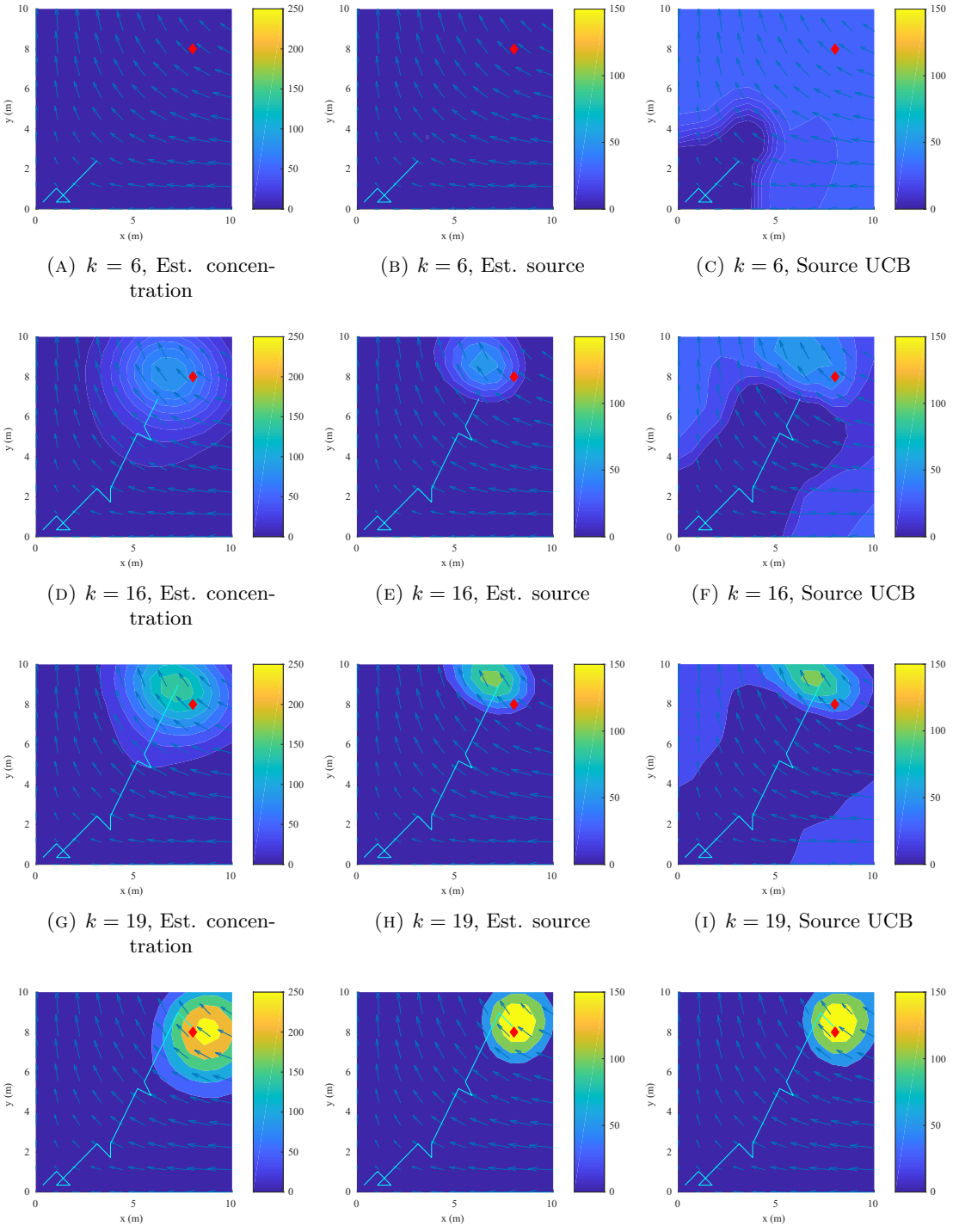
5.5.1 Simulated 2D Source

In this section, we test the proposed GP-UCB algorithm on an simulated 2-dimensional environment with a double gyre-flow field and plume. The plume with respect to the flow field is generated using the advection-diffusion PDE in (5.5).

The results are illustrated in Fig. 5.3. The glider initially has no prior knowledge of the source location as shown in Fig. 5.3a. Still, the upper confidence bound (UCB) map allows the glider to choose the next waypoint effectively, because the proposed ADGP can exploit the information about the flow field to compute a low uncertainty upstream of the glider. Therefore, low concentration at the present location indicates low likelihood of a source upstream. After the initial exploratory behaviour, the glider gets close to the source (i.e., around $k = 16$), and starts to develop a meaningful belief over the concentration and over the source location. The UCB map starts to show more interesting areas to visit. As the glider moves further, the waypoint from UCB approaches the true source location and the glider finally reaches the source. At this point, source localization is accurate up to 1-2 distance units.

An interesting observation is that the UCB of the source strength becomes almost indistinguishable to the source strength when the estimate is converged. It is thus evident that the upper confidence bound correctly balances between exploration and exploitation. When the estimate is uncertain, the glider explores the areas where the uncertainty is higher. When the estimate is certain, the glider exploits the estimate and converges toward the true source location.

Meanwhile, the estimate of the concentration does not necessary converge to the ground truth. This is because our objective is not to estimate the overall concentration of the entire environment, but to simply reach the source location. Consequently, the algorithm does not expend effort on trying to estimate the concentration correctly, unless it is necessary for estimating and converging toward the source location.



(A) $k = 6$, Est. concentration (B) $k = 6$, Est. source (C) $k = 6$, Source UCB

(D) $k = 16$, Est. concentration (E) $k = 16$, Est. source (F) $k = 16$, Source UCB

(G) $k = 19$, Est. concentration (H) $k = 19$, Est. source (I) $k = 19$, Source UCB

(J) $k = 20$, Est. concentration (K) $k = 20$, Est. source (L) $k = 20$, Source UCB

FIGURE 5.3: 2D simulation of source seeking. The vehicle (light blue) manoeuvres through ocean currents (in pale blue) to find the source of plume (red diamond)

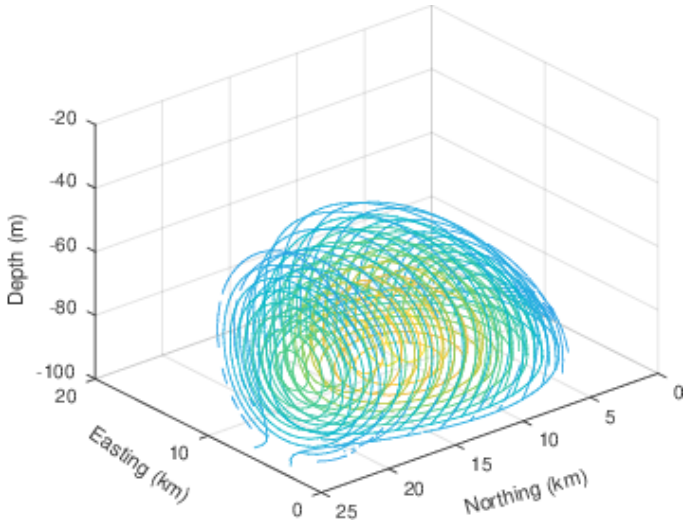
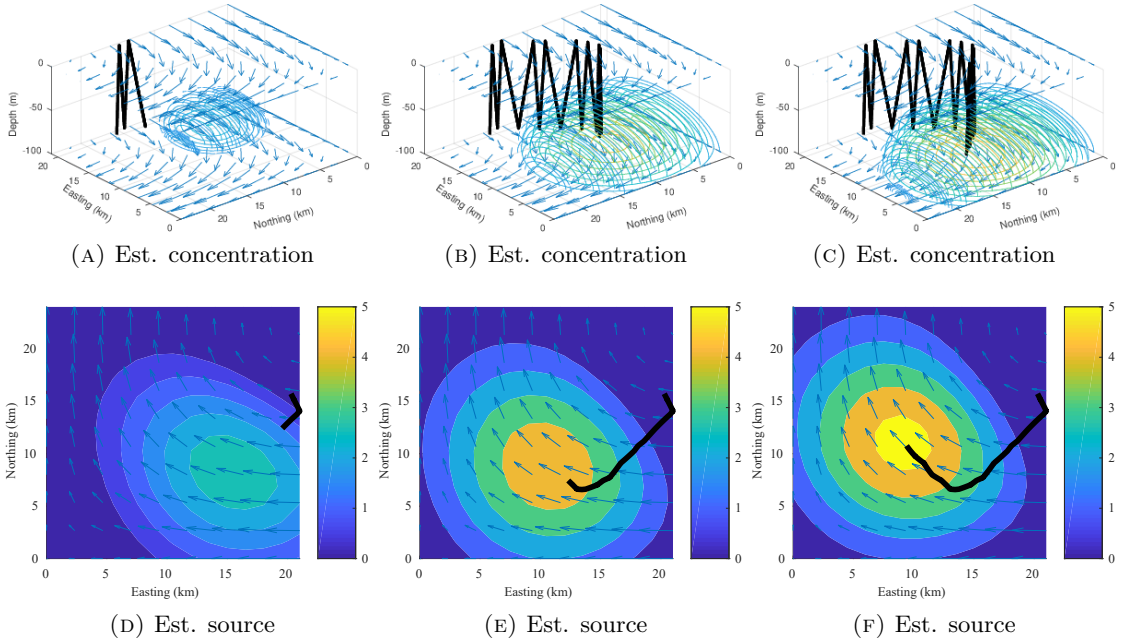


FIGURE 5.4: Ground truth concentration from experimental dataset. Iso-value contours are shown.

5.5.2 3D Experimental Dataset

We present a realistic 3D experiment using a methane plume dataset collected around Perth, Australia by Blue Ocean Monitoring [33] with an optical methane gas sensor, and the model of a Teledyne-Webb G2 Slocum glider [36]. The glider surveyed a region known to exhibit methane concentration. We pre-processed the collected data with GP regression to produce smooth 'ground truth' concentration, which is shown in Fig. 5.4. During the validation, measurements are simulated using the pre-processed concentration data through linear interpolation. As the data is from the real ocean, it is unknown where the true source location is. However, it is intuitive to expect the source to be at the centre of the plume shown in Fig. 5.4.

The result of the proposed algorithm is shown in Fig. 5.5. It can be seen that the dive- and surface-locations are chosen sequentially to eventually reach the centre of the plume, by updating the concentration and source estimation. Our analysis using GP-UCB shows that there is a high chance of the source being located around the centre of the plume. This is further ascertained by the convergence of the estimated source strength around the glider's final position.



(A) Est. concentration (B) Est. concentration (C) Est. concentration

(D) Est. source (E) Est. source (F) Est. source

FIGURE 5.5: Planning over the real measurement dataset

5.6 Summary

We presented an example of full prospection in both planning and perception in the special case of environmental monitoring. Prospection was used to enable a robot to reach the source of an environmental process based on sparse concentration measurements. A specialised GP regression scheme called ADGP was developed by incorporating the physical model of advection diffusion equation. By combining ADGP with the GP-UCB strategy and the FMT* planner [42, 237], we demonstrated that the robot localises the source of an underwater plume in an energy-efficient manner. Owing to prospective perception enabled by the physical prior of advection-diffusion PDE, the algorithm exhibits intelligent behaviour such as assigning low priority to the downstream region if the concentration measurement is low. This led to good empirical performance algorithms supported by a theoretical guarantee.

Although the example presented in this chapter demonstrates both prospective perception and planning, it is difficult to capture the relevance of environmental knowledge to task completion due to the simplicity of the task considered. In the following chapter, we

consider prospective planning in more complex tasks in order to explore how to capture task-relevance of knowledge.

Chapter 6

Prospective Planning for Complex Semantic Tasks

The previous chapters explored how prospection can be achieved in planning problems whose objective is given by a simple numerical function, such as maximising the number of targets captured (Sec. 4.3) or the magnitude of an environmental process (Ch. 5). In this chapter, we consider prospective planning for complex semantic tasks that feature, for example, ordinal constraints (i.e., task A must be completed before or after task B) or ambiguity (i.e., tasks A and B must be completed regardless of order). The semantic planning problem poses an interesting challenge for prospection as the relevance of environmental features to mission success varies significantly throughout the progress. For instance, an environmental feature relevant to a particular task will no longer be of interest once the task has been completed. We formulate and solve this problem by extending signal temporal logic (STL) synthesis from formal methods [208] with uncertain semantics. The results illuminate the importance of prospection. By accounting for the environmental uncertainty and the resolution thereof, desirable behaviours arise such as deliberately reducing the location uncertainty of a relevant target, and prioritising targets whose uncertainty is increasing over those that are physically proximate. Such behaviours illustrate the benefits of prospection.

This chapter builds upon our previous work in [59]. Further empirical analysis of performance characteristics are presented in Appendix C.

6.1 The Semantic Planning Problem

We first define the semantic planning considered in this chapter. We consider a robot with N-dimensional state $\mathbf{x}_t \in \mathbb{R}^N$ and control actions $\mathbf{u}_t \in \mathbb{U}$, where $\mathbb{U}$ is a continuous set of admissible control actions. The robot's dynamics are uncertain, and is modelled by a discrete-time, continuous-space Markov chain $\mathcal{P}(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_t)$ between t and $t+1$, so that the trajectory distribution over a horizon T is given by:

$$\mathcal{P}(\mathbf{X} \mid \mathbf{U}) = \mathcal{P}(\mathbf{x}_1) \prod_{t=1}^{T} \mathcal{P}(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{u}_t), \tag{6.1}$$

where $\mathbf{X} \equiv \mathbf{x}_1 ... \mathbf{x}_T$ and $\mathbf{U} \equiv \mathbf{u}_1 ... \mathbf{u}_T$.

The robot encounters a finite set of random events $\mathcal{E} = \{E^1, ..., E^M\}$ (e.g., 'object detected'), whose probability of occurrence depends on robot's state $\mathbf{x}_t$ and time t. We are interested in finding control actions $\mathbf{U}^*$ that maximises the probability of satisfying a task specification Φ defined over $\mathcal{E}$ (e.g., 'detect all objects'). This constitutes a synthesis problem:

Problem 6.1 (Synthesis)**.** Given the uncertain dynamics (6.1), and a temporal logic task specification Φ over a set of random events $\mathcal{E}$ with probability of satisfaction $\mathcal{P}\left((\mathbf{X}, t) \models \Phi\right)$, find an optimal sequence of controls $\mathbf{U}^*$ such that the trajectory $\mathbf{X}$ maximises the probability of satisfying Φ over time horizon T:

$$\mathbf{U}^* = \underset{\mathbf{U} \in \mathbb{U}^T}{\arg\max} \ \mathcal{P}\left((\mathbf{X}, t) \models \Phi\right), \tag{6.2}$$

with respect to time $t = 1$. We will shortly define the probability of satisfaction $\mathcal{P}\left((\mathbf{X}, t) \models \Phi\right)$.

6.2 Random Signal Temporal Logic (RSTL)

While STL is a prominent tool for task specification, it cannot be used directly for prospective planning because it is naturally deterministic and cannot reason over environmental uncertainty. For this reason, we first propose random signal temporal logic (RSTL) as an extension of STL with uncertain semantics. Our core insight is to consider the *probability* of satisfaction, rather than a deterministic outcome.

6.2.1 Definition and Semantics

We model the random events $\mathcal{E} = \{E^1, ..., E^M\}$ as (not necessarily independent) Bernoulli random variables that are dependent on robot's state and time, with conditional probability of occurrence:

$$\mathcal{P}(E^i = 1 \mid \mathbf{x}_t, t) = \mathcal{P}^i(\mathbf{x}_t, t). \tag{6.3}$$

In other words, each E^i is a Bernoulli *random field* over $\mathbb{R}^N \times \mathbb{R}^+$. Given a set of random events $\mathcal{E}$, the syntax of an RSTL formula Φ is given by:

$$\Phi := E \mid \neg\Phi \mid \Phi \wedge \Psi \mid \Phi \mathcal{U}_{[t_1,t_2]} \Psi, \tag{6.4}$$

where $E \in \mathcal{E}$, and Ψ, Φ are RSTL formulae. $\neg$ is logical negation, $\wedge$ is logical conjunction. $\mathcal{U}$ is the temporal operator 'Until', and $\Phi \mathcal{U}_{[t_1,t_2]} \Psi$ means Φ must hold true between time $[t_1, t_2]$ until Ψ. Other operators such as $\vee$ (disjunction), $\mathcal{F}_{[t_1,t_2]}$ ('in **F**uture', i.e., eventually) and $\mathcal{G}_{[t_1,t_2]}$ ('**G**lobally', i.e., always) can be derived from the syntax the same way as deterministic STL [208]. The events E will be referred to as 'event predicates'.

RSTL is *random* in the sense that the satisfaction of an RSTL formula Φ is a Bernoulli random event for a given trajectory $\mathbf{X}$. The probability of satisfaction is the *expected* rate of satisfaction computed over sampled instances of the event predicates $\hat{\mathcal{E}} = \{e_1, ..., e_M\} \sim \mathcal{E}$:

$$\mathcal{P}((\mathbf{X}, t) \models \Phi) = \mathop{\mathbb{E}}_{\hat{\mathcal{E}} \sim \mathcal{E}} \mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi). \tag{6.5}$$

Here, $\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi)$ is a deterministic function evaluated recursively as:

$$\begin{aligned}
\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, E^i) &\equiv e^i(\mathbf{x}_t, t) \sim \mathcal{P}^i(\mathbf{x}_t, t), \\
\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \neg\Phi) &\equiv \neg\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi), \\
\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi \wedge \Psi) &\equiv \mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi) \wedge \mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Psi), \\
\mathbf{Sat}(\mathbf{X}, t, \hat{\mathcal{E}}, \Phi \mathcal{U}_{[t_1,t_2]} \Psi) &\equiv \\
\bigvee_{\tau_1 \in t+[t_1,t_2]} \bigwedge_{\tau_2 \in t_1+[0,\tau_1]} & \mathbf{Sat}(\mathbf{X}, \tau_2, \hat{\mathcal{E}}, \Phi) \vee \mathbf{Sat}(\mathbf{X}, \tau_1, \hat{\mathcal{E}}, \Psi),
\end{aligned} \tag{6.6}$$

where $e^i(\mathbf{x}_t, t) \in \{0, 1\}$ is a sample from E^i at robot state $\mathbf{x}_t$ and time t.

For brevity, we use a shorthand $\mathcal{P}(\Phi_t \mid \mathbf{X}) \equiv \mathcal{P}((\mathbf{X}, t) \models \Phi)$ in the rest of this chapter.

6.2.2 Examples

In this section, we give motivating examples of RSTL formulae. The simplest example we consider is that of simply visiting a number of targets:

Example 6.1 (Visiting Multiple Targets)**.** Suppose we are given target predicates D^i, $i \in [1, N]$, that represent visiting target i. A search formula can be defined as:

$$\Phi_{\text{search}} = \bigwedge_i \mathcal{F} D^i, \tag{6.7}$$

which, when read literally, means "eventually visit target 1 and eventually visit target 2, and so forth".

Building on this simple example, we can add a target to avoid:

Example 6.2 (Reach-Avoid Task)**.** Suppose we are given a target predicate D^r to reach, and D^a to avoid. A reach-avoid task can be written in RSTL as:

$$\Phi_{\text{reach-avoid}} = \mathcal{F} D^r \wedge \mathcal{G} \neg D^a, \tag{6.8}$$

which, when read literally, means "eventually visit D^r and always do not (i.e. never) visit D^a".

We can further add temporal relations between tasks using the temporal operators of RSTL. One such example is to define precedence relations between tasks as follows:

Example 6.3 (Precedence)**.** Suppose we are given two tasks, Φ_1 and Φ_2, which we would like to complete in the order of Φ_1 then Φ_2. This precedence relation can be written in RSTL as:

$$\Phi_{\text{precedence}} = \neg\Phi_2 \mathcal{U} \Phi_1 \wedge \mathcal{F}\Phi_2, \tag{6.9}$$

which, when read literally, means "do not complete Φ_2 until Φ_1 is complete, and eventually complete Φ_2".

6.2.3 Approximate Analytical Semantics with Conditional Independence

To synthesise trajectories that maximise the probability of task satisfaction, we must first be able to evaluate the probability of task satisfaction (6.5). However, computing the expectation in (6.5) is intractable in general, and would require Monte Carlo sampling, which is unappealing for use in trajectory synthesis. To compute (6.5) analytically without sampling, we observe that (6.5) applies logical operations to samples from Bernoulli random variables. A convenient approximation is the product relation for independent Bernoulli random variables A and B:

$$\mathcal{P}(A \wedge B) = \mathcal{P}(A)\mathcal{P}(B). \tag{6.10}$$

Technically, the product relation holds true if the operands are *conditionally independent* given $\mathbf{X}$. The conditionally independent (CI)-approximation was introduced in [242] as follows:

Definition 6.1 (CI-approximation [242])**.** Given an RSTL formula Φ, the CI-approximation of $\mathcal{P}(\Phi_t \mid x)$ is defined by:

$$\begin{aligned}
\mathcal{P}(E_t^i \mid \mathbf{X}) &\equiv \mathcal{P}^i(\mathbf{X}, t), \\
\mathcal{P}(\neg\Phi_t \mid \mathbf{X}) &\equiv 1 - \mathcal{P}(\Phi_t \mid \mathbf{X}), \\
\mathcal{P}(\bigwedge_i \Phi_t^i \mid \mathbf{X}) &\equiv \prod_i \mathcal{P}(\Phi_t^i \mid \mathbf{X}), \\
\mathcal{P}((\mathcal{G}_{[t_1,t_2]}\Phi)_t \mid \mathbf{X}) &\equiv \prod_{\tau \in t+[t_1,t_2]} \mathcal{P}(\Phi_\tau \mid \mathbf{X}).
\end{aligned} \tag{6.11}$$

6.2.4 Recovering Deterministic Semantics with Log-Odds Transform

The output range for CI-approximation of $\mathcal{P}$ (6.11) is $[0, 1]$ since it computes probability. This can lead to numerical instability, and gradient ascent often leads to poor convergence. A natural re-parameterisation for Bernoulli random variables is the *log-odds*:

$$\mathcal{L}(A) = \log \frac{\mathcal{P}(A)}{\mathcal{P}(\neg A)}. \tag{6.12}$$

It can be shown with some algebraic manipulations that re-writing the CI rule for pairwise disjunction $\vee$ in (6.11) in terms of log-odds leads to:

$$\begin{aligned}
\mathcal{L}(A \vee B) &= \log \frac{\mathcal{P}(A)\mathcal{P}(B) + \mathcal{P}(\neg A)\mathcal{P}(B) + \mathcal{P}(A)\mathcal{P}(\neg B)}{\mathcal{P}(\neg A)\mathcal{P}(\neg B)} \\
&= \mathfrak{lse}(\mathcal{L}(A), \mathcal{L}(B), \mathcal{L}(A) + \mathcal{L}(B)),
\end{aligned} \tag{6.13}$$

where A and B are independent Bernoulli random variables, and $\mathfrak{lse}$ is the *log-sum-exp* function:

$$\mathfrak{lse}(\mathcal{L}_1, ..., \mathcal{L}_N) = \log \sum_i \exp \mathcal{L}_i. \tag{6.14}$$

A series of disjunction operations (6.13) is then:

$$\mathcal{L}\left(\bigvee_{i \in I} A_i\right) = \log \sum_{J \in 2^I} \exp \sum_{j \in J} \mathcal{L}(A_j), \tag{6.15}$$

where 2^I denotes the power set of I.

Computing log-sum-exp over sum of all subsets is clearly cumbersome. We avoid such computation by using the relationship between elementary symmetric polynomials and monic polynomials. Observe that the summations over $j \in J$ can be taken out of the exponential as products. Then, we have all elementary symmetric polynomials over A_i less 1. We thus arrive at a more compact expression:

$$\mathcal{L}\left(\bigvee_i A_i\right) = \log\left(\prod_i \left(1 + \exp \mathcal{L}(A_i)\right) - 1\right). \tag{6.16}$$

Now, the CI computation rules in the log-odds domain are given as follows:

Definition 6.2 (CI-approximate log-odds)**.** Given an RSTL formula Φ, the CI-approximation of log-odds of satisfaction $\mathcal{L}(\Phi_t \mid \mathbf{X})$ is calculated by:

$$\begin{aligned}
\mathcal{L}(E_t^i \mid \mathbf{X}) &\equiv \log \mathcal{P}^i(\mathbf{X}, t) - \log(1 - \mathcal{P}^i(\mathbf{X}, t)), \\
\mathcal{L}(\neg\Phi_t \mid \mathbf{X}) &\equiv -\mathcal{L}(\Phi_t \mid \mathbf{X}), \\
\mathcal{L}\left(\bigvee_i \Phi_t^i \mid \mathbf{X}\right) &\equiv \log\left(\prod_i (1 + \exp \mathcal{L}(\Phi_t^i \mid \mathbf{X})) - 1\right), \\
\mathcal{L}((\mathcal{F}_{[t_1,t_2]}\Phi)_t \mid \mathbf{X}) &\equiv \log\Big(\prod_{\tau \in t + [t_1,t_2]} \left(1 + \exp \mathcal{L}(\Phi_\tau \mid \mathbf{X}\right) - 1\Big).
\end{aligned} \tag{6.17}$$

Interestingly, the proposed computation rules for probability of satisfaction exhibits strong similarities to existing work on deterministic STL synthesis and model checking [208, 211, 212, 243, 244]. In the log-odds domain, certain satisfaction (i.e., probability of 1) translates to ∞, certain dissatisfaction is $-\infty$, and absolute uncertainty (i.e., probability of 0.5) is 0, which are the behaviours of spatial robustness measure for deterministic STL introduced in [208]. Further, the log-sum-exp function has been used in deterministic STL synthesis [211, 243] as a smooth approximation of the maximum function. Finally, a similar expression to (6.16) was presented in [244] as an alternative robustness measure for deterministic STL. The authors report encouragement of repeated satisfaction, which is consistent with the probability of disjunction increasing with increasing probability of the disjuncts.

Given that the approaches that do not encourage repeated satisfaction [211, 243] still report acceptable results, we consider the following approximation for disjunction:

$$\begin{aligned}\mathcal{L}(A \vee B) &\approx \mathfrak{lse}(\mathcal{L}(A), \mathcal{L}(B))) \\ &= \log \frac{\mathcal{P}(A)\mathcal{P}(\neg B) + \mathcal{P}(\neg A)\mathcal{P}(B)}{\mathcal{P}(\neg A)\mathcal{P}(\neg B)}.\end{aligned} \tag{6.18}$$

This ignores repeated satisfaction by omitting the $\mathcal{P}(A)\mathcal{P}(B)$ term from the numerator of (6.13). Meanwhile, there is a potential numerical benefit that log-sum-exp can be computed numerically stably with the so-called *log-sum-exp trick*, whereas the product in (6.16) may underflow. We thus define the *mutually exclusive (ME)* approximation as follows.

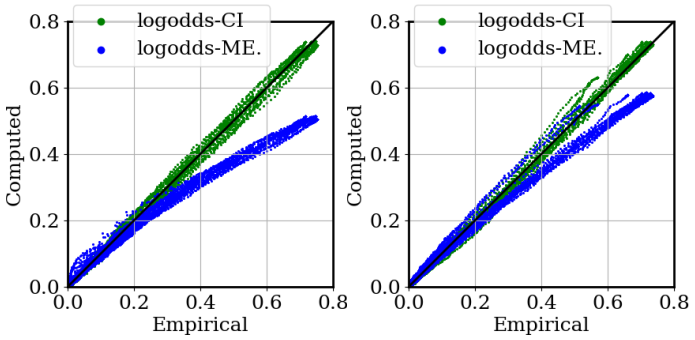
Definition 6.3 (ME approximation)**.** Given an RSTL formula Φ, the ME approximation of $\mathcal{L}(\Phi_t \mid \mathbf{X})$ is calculated by:

$$\begin{aligned}\mathcal{L}(E^i_t \mid \mathbf{X}) &\equiv \log \mathcal{P}^i(\mathbf{X}, t) - \log(1 - \mathcal{P}^i(\mathbf{X}, t)), \\ \mathcal{L}(\neg\Phi_t \mid \mathbf{X}) &\equiv -\mathcal{L}(\Phi_t \mid \mathbf{X}), \\ \mathcal{L}(\Phi_t \vee \Psi_t \mid \mathbf{X}) &\equiv \mathfrak{lse}(\mathcal{L}(\Phi_t \mid x), \mathcal{L}(\Psi_t \mid \mathbf{X})), \\ \mathcal{L}((\mathcal{F}_{[t_1,t_2]}\Phi)_t \mid \mathbf{X}) &\equiv \log \sum_{\tau \in t + [t_1,t_2]} \exp \mathcal{L}(\Phi_\tau \mid \mathbf{X}).\end{aligned} \tag{6.19}$$

6.2.5 Empirical Analysis of Approximation Quality

In this section, we evaluate whether the CI and ME approximations compute the probability of satisfaction accurately. Because the CI computation rule assumes independence amongst operands, we expect it to be exact if 1) all predicates are conditionally independent; 2) each predicate is independent across time and space; and 3) only one operator uses each predicate. For example, assuming 1) and 2) hold, we expect the CI rule to be exact on $\mathcal{F}A \wedge \mathcal{F}B$, but not $\mathcal{F}(A \wedge \mathcal{F}B)$, because the disjuncts of the outer $\mathcal{F}$ operator are not independent. The ME computation rules (6.19) will not be exact in any case.

We validate these hypotheses by comparing against a 1000-sample Monte Carlo (MC) approximation of RSTL probability of satisfaction (6.5). We used the trajectories from the first 2000 gradient ascent steps generated from the target search scenario (Fig. 6.2).



(A) $\mathcal{F}(A) \wedge \mathcal{F}(B)$ (B) $\mathcal{F}(A \wedge \mathcal{F}(B))$

FIGURE 6.1: Comparison of the CI (green) and ME (blue) approximation results against MC estimates ('Empirical'). CI is exact for $\mathcal{F}(A) \wedge \mathcal{F}(B)$, whereas ME underestimates. For $\mathcal{F}(A \wedge \mathcal{F}(B))$, the error increases, but not significantly. Naive method's result showed numerically insignificant difference to CI, and is omitted.

For simplicity, we evaluated each predicates' marginal probability independently before sampling, so that the first two conditions of exactness hold.

In Fig. 6.1a, CI (green) and ME (blue) results are compared against the MC estimate for $\mathcal{F}A \wedge \mathcal{F}B$. It can be seen that the CI method matches the MC result as expected, whereas ME consistently underestimates. This is expected, because ME does not account for multiple satisfaction.

Fig. 6.1b shows comparison for $\mathcal{F}(A \wedge \mathcal{F}B)$. As $\mathcal{F}B$ is double-counted by the outer $\mathcal{F}$, CI and ME tend to over-estimate, but not by much. ME continues to underestimate, and CI matches the MC closely, showing that CI and ME are reasonable approximations for practical applications.

6.3 Event Predicates for Robotics Applications

RSTL can be used with a variety of event predicates to model a wide variety of uncertain semantics. In this section, we present useful examples of event predicates for modelling common tasks in robotics applications.

6.3.1 Collision Avoidance

The most common representation of obstacles is to use an occupancy grid O, such that $O_{i,j}$ is the probability of obstacle occupancy [67]. A desirable property of the event predicate is to ensure differentiability, because it allows efficient gradient-based synthesis. For simplicity, we model the probability of collision with an obstacle by an interpolation, using `batch_interp_regular_nd_grid` in Tensorflow [245], which is a differentiable operation. Formally, the model can be written as:

$$\mathcal{P}(O \mid \mathbf{x}_t) = \sum_{i,j} K_{ij}(\mathbf{x}_t) O_{i,j}, \tag{6.20}$$

where $K_{ij}(\mathbf{x})$ denotes the interpolant.

Alternatively, continuous obstacle representations, such as Gaussian process (GP) occupancy maps [73] or Log-Gaussian process implicit surface (GPIS) (Sec. 7.2) are also differentiable and allows gradient-based synthesis. The incorporation of these representations are deferred to future work for the sake of simplicity.

6.3.2 Target Detection

Another common class of robotic task is to visit or detect targets of interest. Such tasks are modelled as follows. If the target's position is known deterministically, the robot detects the target with likelihood modelled by:

$$\mathcal{P}(D_t \mid \mathbf{x}_t, \mathbf{z}_t) = P_D \exp\left(\frac{||\mathbf{x}_t - \mathbf{z}_t||^2}{2r_D^2}\right), \tag{6.21}$$

where $\mathbf{z}_t$ is the target location, r_D is the radius of detection, and P_D controls the peak.

Meanwhile, a target's position is usually unknown or uncertain. A common approach is to model the target's position as a linear Gaussian system. Given a linear system model $\dot{\mathbf{z}} = A\mathbf{z}$, the mean $\bar{\mathbf{z}}_t$ and covariance Σ_t^z of target position can be propagated from the robot's belief as follows:

$$\begin{aligned} \dot{\bar{\mathbf{z}}} &= A\bar{\mathbf{z}}, \\ \dot{\Sigma_t^z} &= A\Sigma_t^z + \Sigma_t^z A^T. \end{aligned} \tag{6.22}$$

Given the mean $\bar{\mathbf{z}}_t$ and covariance Σ_t^z of target position, the marginal probability of detection accounting for their uncertainty is given by:

$$\begin{aligned}\mathcal{P}(D_t \mid \mathbf{x}_t) &= \int \exp\left(\frac{||\mathbf{x}_t - \bar{\mathbf{z}}_t||^2}{2r_d^2}\right)\mathcal{N}(\bar{\mathbf{z}}_t, \Sigma_t^z)d\mathbf{z}_t \\ &= \sqrt{2\pi^N} r_d^N \mathcal{N}(\bar{\mathbf{z}}_t, \Sigma_t^z + r_d^2 I),\end{aligned} \tag{6.23}$$

where $\mathcal{N}(\bar{\mathbf{z}}_t, \Sigma_t^z)$ is the multivariate Gaussian probability distribution function (PDF).

6.3.3 Prospecting for Measurements

For Guassian targets, it is possible to model the effect of measurements directly using tools from Gaussian belief space planning [22], without requiring the use of mutual information upper confidence bound (MI-UCB) in Ch. 4. This is achieved by using the maximum likelihood observation assumption [22] and the information filter equations.

For simplicity, we consider a static target modelled by a Gaussian distribution with mean and covariance $\bar{\mathbf{z}}$ and Σ_0^z. The robot makes a direct observation of the position of the target, with measurement noise $\sigma^2(\mathbf{x}_t)$. The measurement noise is used to model the robot's field of view. For example, for a drone equipped with a downward looking camera, the field of view is circular, and the corresponding measurement noise can be modelled as:

$$\sigma^{-2}(\mathbf{x}_t) = \sigma_0^{-2}\exp(\frac{||\bar{\mathbf{z}} - \mathbf{x}_t||}{2r_D^2}). \tag{6.24}$$

Given the noise model, the posterior target covariance after accounting for measurements is given by:

$$(\Sigma_t^z)^{-1} = (\Sigma_0^z)^{-1} + \sigma^{-2}(\mathbf{x}_t)I, \tag{6.25}$$

which can be combined with the detection model (6.23).

The merit of (6.25) is that there are no explicit inversions when computing the likelihood of detection (6.25) given the robot's position $\mathbf{x}_t$. This allows efficient and stable gradient computation using autograd frameworks such as Tensorflow [245] or PyTorch [246]. The static target assumption is important in ensuring this property, but may be relaxed through

the use of sampling-based algorithms such as Monte Carlo tree search (MCTS) [164, 165], or by differentiating the continuous-time Ricatti equation as done in [247]. For simplicity, we restrict our attention to static targets.

6.4 Synthesis via Gradient-Based MAP Inference

6.4.1 Reformulation as MAP Inference

Since both task satisfaction and robot dynamics are probabilistic, it is natural to ask if Problem 6.1 can be solved solely within the realm of probability theory. This is achieved by the control-as-inference paradigm [248, 249], which has been shown to not only encompass existing optimal control problems, but also to enable new approaches. We follow a similar development and present an inference formulation of Problem 6.1.

In this formulation, the problem is modelled by the joint distribution among task satisfaction, robot trajectory, and control actions:

$$\mathcal{P}(\Phi_t, \mathbf{X}, \mathbf{U}) = \mathcal{P}(\Phi_t \mid \mathbf{X})\mathcal{P}(\mathbf{X} \mid \mathbf{U})\mathcal{P}(\mathbf{U}). \tag{6.26}$$

Here, $\mathcal{P}(\mathbf{U})$ is our prior belief on what the control actions should be, and is representative of the admissible control space $\mathbb{U}$ in the synthesis formulation. For example, if $\mathcal{P}(\mathbf{U})$ is a zero-mean Gaussian prior, it is equivalent to penalising quadratic control cost. The prior can derive from other knowledge, e.g., an imitation-learnt prior as [250] does for optimal control.

A balance between admissibility and probability of satisfaction is captured by the *posterior* probability of control actions given that the task is satisfied:

$$\begin{aligned} \mathcal{P}(\mathbf{U} \mid \Phi_t) &\propto \mathcal{P}(\mathbf{U})\mathcal{P}(\Phi_t \mid \mathbf{U}) \\ &= \mathcal{P}(\mathbf{U}) \int \mathcal{P}(\Phi_t \mid \mathbf{X})\mathcal{P}(\mathbf{X} \mid \mathbf{U}) d\mathbf{X}. \end{aligned} \tag{6.27}$$

We thus pose Problem 6.1 as a maximum a posteriori (MAP) inference problem:

Problem 6.2 (Inference)**.** Given the dynamic model (6.1) and an RSTL task specification Φ, find the MAP control actions $\mathbf{U}^*$ given the robot's trajectory satisfies Φ with respect to t:

$$\mathbf{U}^* = \arg\max_{\mathbf{U}} \mathcal{P}(\mathbf{U} \mid \Phi_t). \tag{6.28}$$

6.4.2 Approximate Gradient Ascent

The MAP control sequence $\mathbf{U}^*$ that maximises the posterior probability is computed using gradient ascent. We use Jensen's inequality to bound the log of posterior probability (6.28):

$$\log \mathcal{P}(\mathbf{U} \mid \Phi_t) \geq \mathop{\mathbb{E}}_{\mathbf{X} \sim \mathcal{P}(\mathbf{X}|\mathbf{U})} [\log \mathcal{P}(\Phi_t \mid \mathbf{X}))] + \log \mathcal{P}(\mathbf{U}). \tag{6.29}$$

Subsequently, we maximize the lower bound:

$$\mathbf{U}^* = \arg\max_{\mathbf{U}} \mathop{\mathbb{E}}_{\mathbf{X} \sim \mathcal{P}(\mathbf{X}|\mathbf{U})} [\log \mathcal{P}(\Phi_t \mid \mathbf{U})] + \log \mathcal{P}(\mathbf{U}). \tag{6.30}$$

The maximisation is done by gradient ascent on (6.30). Because the expectation in (6.30) is intractable, we replace it with an empirical mean over a N_s number of trajectory samples, so that the i-th gradient ascent step is:

$$\hat{\mathbf{U}}^{i+1} = \hat{\mathbf{U}}^i + \frac{1}{N_s} \sum_j \frac{\partial}{\partial \hat{\mathbf{U}}^i} [\log \mathcal{P}(\Phi_t \mid \mathbf{X}^j_{1:T}(\hat{\mathbf{U}}^i)) + \log \mathcal{P}(\hat{\mathbf{U}}^i)], \tag{6.31}$$

where $\hat{\mathbf{U}}^{i+1} = \left[\hat{\mathbf{u}}^i_1 ... \hat{\mathbf{u}}^i_T\right]$. Each trajectory sample $\mathbf{X}^j(\hat{\mathbf{U}}^i)$ is obtained by propagating the dynamic model (6.1) forward in time including actuation uncertainty. Throughout this process, the objective (6.31) remains differentiable as long as the predicates' distributions and the dynamic model are differentiable. If so, analytical gradients can be computed easily using autograd frameworks such as Tensorflow [245] or PyTorch [213, 246] for both CI and ME approximations. Therefore, we do not present the expressions here.

6.5 Demonstrations

We demonstrate three example cases that illustrate how RSTL allows prospective planning in uncertain environments. The proposed gradient ascent method was implemented in Tensorflow [245]. For all examples, we consider a robot described by a bicycle dynamic model:

$$\dot{\mathbf{x}}_t = \begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} V_t \cos\theta_t \\ V_t \sin\theta_t \\ \omega_t + \epsilon_t \end{bmatrix}, \tag{6.32}$$

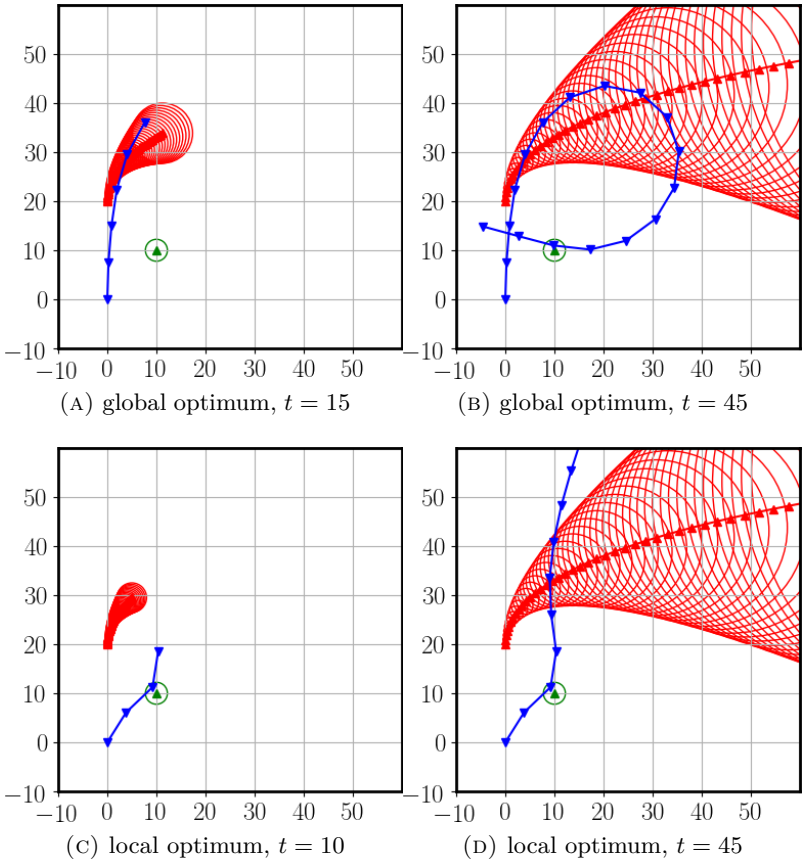
where $\epsilon_t \sim \mathcal{N}(0, \sigma_u)$ is white Gaussian noise. The control inputs are $\mathbf{u}_t = \begin{bmatrix} V_t & \omega_t \end{bmatrix}^\top$.

6.5.1 2D Target Search

Motivated by Example 6.1, we consider a 2D target search scenario, where a robot is tasked with detecting possibly moving targets in the environment: *Tom* and *Jerry*. Jerry, as usual, is moving with increasing uncertainty, whereas Tom is stationary with high certainty. Fig. 6.2 depicts an example where the mean paths for Tom and Jerry are shown in green and red. The growing uncertainty over time is shown around the mean. Since Tom (in green) is known to be stationary, its uncertainty does not grow over time. The robot starts at $\mathbf{x}_1 = [0, 0]^\top$.

As per Example 6.1, the task of finding Tom and Jerry can be expressed using RSTL as $\Phi = \mathcal{F}(D^{\text{Tom}}) \wedge \mathcal{F}(D^{\text{Jerry}})$ (i.e., 'eventually detect Tom and eventually detect Jerry'). The event predicates D^{Tom} and D^{Jerry} are modelled as per Sec. 6.3.2. Tom and Jerry are modelled by a constant acceleration model, which is a linear Gaussian system.

Fig. 6.2 shows global and local optima found using the log-odds CI computation rule (6.17). It can be seen that the global optimum (Figs. 6.2a and 6.2b) is to detect Jerry first at $t = 15$ (Fig. 6.2a), and to return to Tom at $t = 45$ (Fig. 6.2b), whereas the local optimum is to detect Tom first at $t = 10$ (Fig. 6.2c) and then Jerry later. This is because the uncertainty of Jerry grows unlike Tom, and the optimal trajectory should detect Jerry



(A) global optimum, $t = 15$ (B) global optimum, $t = 45$

(C) local optimum, $t = 10$ (D) local optimum, $t = 45$

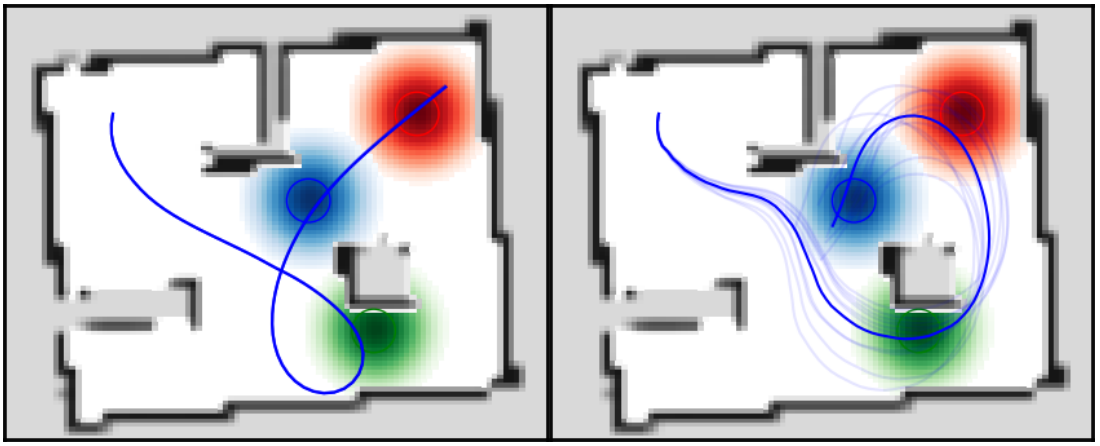
FIGURE 6.2: A 2D target search scenario using RSTL. The robot (blue) is tasked with detecting both Tom (Green) and Jerry (Red). The global optimum with $\mathcal{P}(\Phi \mid \mathbf{x}) \approx 0.7$ (left column) is to detect Jerry before uncertainty grows. A local optimum with $\mathcal{P}(\Phi \mid \mathbf{x}) \approx 0.5$ (right column) prefers Tom, who is closer. Circles show 1-covariance bound.

first before its uncertainty grows. This demonstrates that RSTL allows evaluating the effect of uncertainty on mission success.

6.5.2 Complex Missions in an Indoor Environment

Based on Example 6.3, we consider a nursing robot with a complex task. The robot operates in an indoor environment, modelled as an occupancy grid O. The robot cares for two patients, Rob and Bob. The doctor asks the robot to avoid obstacles, and to visit the supply station before visiting any of the patients, which can be written as an RSTL formula using Example 6.3:

$$\Phi_1 = (\neg(D^{Rob} \vee D^{Bob})\mathcal{U}D^{Sup}) \wedge \mathcal{G}(\neg O), \tag{6.33}$$



(A) $\sigma_u = 10^{-4}$ (B) $\sigma_u = 0.1$

FIGURE 6.3: Results for complex indoor mission using RSTL. The robot's (blue) task is a conjunction of 'never visit Rob (red) or Bob (blue) before visiting supply station (green), and avoid obstacles (grey colormap)', and 'visit Rob (red) and Bob (blue)'. With higher actuation uncertainty σ_u, the trajectory becomes further from the walls. Solid lines are the robot's nominal trajectory in the absence of noise. Transparent blue lines are the noised samples used during synthesis. The start location is top-left corner.

where D^{Rob}, D^{Bob}, and D^{Sup} are distributed as per (6.23).

Now, in addition to the previous command, the doctor asks the robot to visit the two patients:

$$\Phi_2 = \mathcal{F}(D^{Rob}) \wedge \mathcal{F}(D^{Bob}) \wedge \Phi_1. \tag{6.34}$$

We created an occupancy grid from a realistic dataset commonly used in perception research [56, 251], and compared the results with low ($10^{-4}\text{rad}s^{-1}$) and high ($\sigma_u = 0.1\text{rad}s^{-1}$) actuation uncertainty. The results are shown in Fig. 6.3. In both cases, the generated trajectory is correct, visiting the supply station first, and then the two patients. Interestingly, the trajectory changes drastically when control noise increases. The path with small control noise in Fig. 6.3a is aggressively close to the wall, whereas the path with control noise in Fig. 6.3b is more conservative in that the robot keeps distance away from the wall by manoeuvring around the obstacle. This is because, with greater control noise, more trajectory samples collide into obstacles as can be seen in Fig. 6.3b. This demonstrates that the proposed probabilistic formulation encompasses risk-averse behaviour in STL synthesis, a crucial property for practical applications.

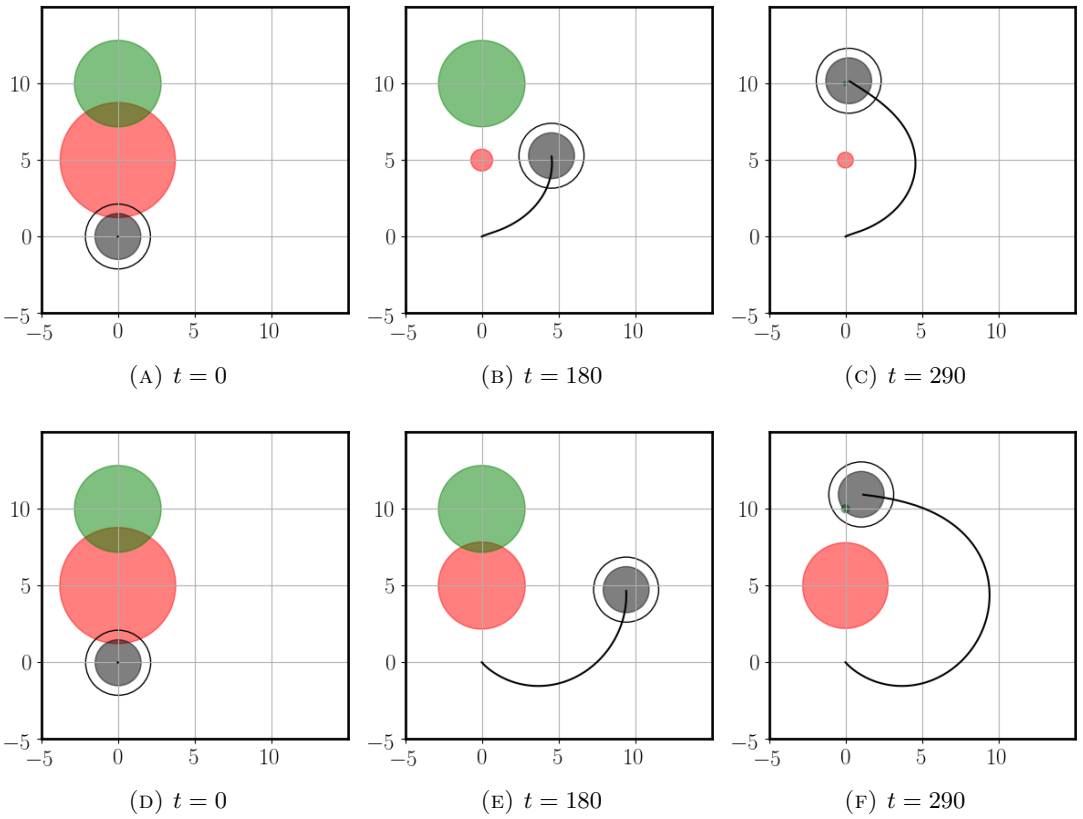
6.5.3 Reach-Avoid Task with Measurement Prospection

We now examine the effect of measurement prospection on the synthesised behaviours with a simple reach-avoid task based on Example 6.2, as illustrated in Fig. 6.4. There are two stationary targets, red and green. The robot's task is to reach the green target while avoiding the red target, which can be written in RSTL as $\Phi = \mathcal{F}(D^{Green}) \wedge \mathcal{G}(D^{Red})$. For both targets, the robot can visit or avoid the target with a fixed range, which will be referred to as the visit range.

The positions of both targets are uncertain. The uncertainty of the red target is especially large and intrudes not only into the confidence bound of the green target, but also into the visit range of the robot. Nonetheless, the uncertainties can be reduced by position measurement during operation as per Sec. 6.3.3. We examine the effect of considering uncertainty reduction by comparing the plans generated with and without measurement prospection. The results are shown in Fig. 6.4.

It can be seen that the robot travels closer to the mean position of the red target when measurement propsection is enabled (Figs. 6.4a- 6.4c) than when it is not. This is because the plan takes into account the uncertainty reduction in red target. It can be seen that the plan without measurement prospection (Figs. 6.4d-6.4f) takes a large detour in order to avoid the red target with respect to the prior belief. This demonstrates that measurement prospection leads to efficient behaviours without unnecessary conservatism by accounting for uncertainty reduction. An interesting future work would be to examine the performance benefits of such efficiency in a closed-loop setting with replanning.

Another interesting observation is that, although there is a moderate probability of violating the avoid the red ($\mathcal{G}(D^{Red})$)) specification, both variants successfully find the best possible trajectory. This is in contrast to deterministic approaches, e.g., [53, 210], where the initial violation would automatically lead to infeasibility. This highlights the fact that explicit consideration of uncertainties without determinisation [53] allows flexible handling probability of failure.



(A) $t = 0$ (B) $t = 180$ (C) $t = 290$

(D) $t = 0$ (E) $t = 180$ (F) $t = 290$

FIGURE 6.4: Result of reach-avoid task using RSTL with measurement prospection. Top row: RSTL with measurement prospection Bottom row: RSTL without measurement prospection. Green and red circles show confidence bounds of the targets. The inner grey circle shows the range of visit, and the outer circle shows the measurement radius. Solid black line represents trajectory. The task is to reach the green target while avoiding the red target $(\mathcal{F}(D^{Green}) \wedge \mathcal{G}(D^{Red}))$. When considering the measurement effects, the generated plan is closer to the red target because the uncertainty can be reduced, whereas the non-prospective plan takes an overly conservative path.

6.6 Summary

We presented how to perform prospective planning for complex semantic tasks, focusing on occupancy grids and targets described by a linear Gaussian system. To do so, we derived RSTL as a natural probabilistic extension of STL by considering the probability of satisfaction over possible realisations of the environment. This allows semantic planning with uncertain dynamics and environments, in addition to efficient gradient-based synthesis posed as MAP inference. The probabilistic semantics enables prospective planning by describing how uncertainty grows, or can be resolved.

Examples of prospective planning presented throughout this thesis consistently illustrate that a better description of environmental uncertainty leads to desirable behaviours. The remainder of this thesis will examine what constitutes a good description of environmental uncertainty.

Chapter 7

Case Studies on Predictive Priors

The previous chapters demonstrated instances of prospection where the planning takes into account the effect of measurements on the plan performance. Meanwhile, another core element of prospection is to devise predictive priors on the environment given prior knowledge and data, as was evident in the intelligent behaviours that arise from the inclusion of physical models in Ch. 5. This chapter presents two case studies where a predictive prior can significantly improve planning performance for navigation in oceanic and indoor environments. The purpose is to identify the characteristics of a good predictive prior for prospection.

This chapter builds upon our prior publications in [56, 62]. In particular, Sec. 7.1 is based on [62], and Sec. 7.2 on [56].

7.1 Online Current Estimation and Navigation (OCEAN)

We first consider the online current estimation and navigation (OCEAN) problem. This problem considers an underwater robot with limited actuation navigating in ocean currents without prior data. While submerged, global positioning system (GPS) is unavailable, and the robot has no position feedback. Due to limited actuation, it must exploit the surrounding ocean currents, which is initially unknown. The robot must therefore estimate the surrounding ocean currents from motion data. In doing so, we incorporate

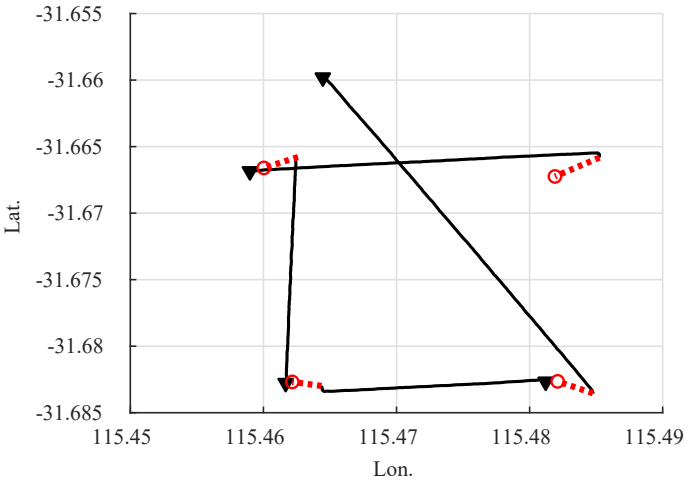


FIGURE 7.1: An illustrative example data from a Slocum G3 glider operation near Perth, Australia. Black bottom-facing triangles: dive-in points. Black solid line: dead-reckoned estimates ($\hat{\mathbf{x}}_t$). Red circles: GPS measurements ($\mathbf{y}_{\tau_k}$). Red dashed lines: the drift ($\Delta\mathbf{x}_{\tau_k}$)

the physical knowledge that ocean currents are planar and incompressible. The incompressibility assumption leads to substantial improvement in the quality of predictions at remote locations, thereby assisting nonmyopic planning. The rest of this section describes the theoretical algorithmic framework to this end. Practical implementation details are deferred to Appendix B.

7.1.1 Problem Statement

We consider an underwater vehicle whose dynamics can be modelled as:

$$\dot{\mathbf{x}}(t) = \mathbf{u}(t) + \mathbf{w}(\mathbf{x}(t)), \tag{7.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^2$ is the position of the vehicle at time t, $\mathbf{u}(t) \in \mathbb{R}^2$ is the velocity through water at time t, and $\mathbf{w} \in \mathcal{C}^\infty(\mathbb{R}^2)$ is the ocean current modelled as a smooth 2D vector field. For simplicity, we do not take into account the vertical motion of the vehicle. We approximate the continuous time dynamics (7.1) with a discrete-time one as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + (\mathbf{u}_t + \mathbf{w}(\mathbf{x}_t))\Delta t, \tag{7.2}$$

where Δt is the sampling time.

The vehicle's velocity through water $\mathbf{u}_t$ is known, whereas the current $\mathbf{w}(\mathbf{x})$ is unknown. The vehicle estimates its own position based on *dead-reckoning* assuming zero ocean current while submerged. The initial estimate derives from the last GPS measurement prior to dive-in:

$$\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t + \mathbf{u}_t \Delta t. \tag{7.3}$$

The vehicle uses the dead-reckoned estimate to attempt to reach a target point. When the vehicle's estimate is within a pre-set tolerance range from the target point, the vehicle climbs up to the surface and updates its position using GPS. We denote the time of surfacing events as $\tau_k \in [1, T]$, $\tau_k < \tau_{k+1}$. The GPS measurements are assumed to have i.i.d. Gaussian measurement error:

$$\mathbf{y}_{\tau_k} = \mathbf{x}_{\tau_k} + \epsilon_k, \tag{7.4}$$

$$\epsilon_k \sim \mathcal{N}(\mathbf{0}, \sigma_y^2 I). \tag{7.5}$$

For the periods in between each GPS measurement, we use shorthand notation $\mathbf{X}_k = \mathbf{X}_{\tau_{k-1}:\tau_k}$ for true trajectory, $\hat{\mathbf{X}}_k = \hat{\mathbf{x}}_{\tau_{k-1}:\tau_k}$ for dead-reckoned trajectory, $\mathbf{y}_k = \mathbf{y}_{\tau_k}$ for GPS measurements, and $\mathbf{W}_k = \{\mathbf{w}(\mathbf{x}_{\tau_{k-1}}), \dots \mathbf{w}(\mathbf{x}_{\tau_k})\}$ for current along trajectory respectively. As the dead-reckoned estimates do not take ocean current into account, there is a substantial disparity between the dead-reckoned estimate and the GPS measurement. Throughout the rest of the paper, we refer to this disparity as *drift*, and denote it by $\Delta\mathbf{x}_k$. In other words:

$$\Delta\mathbf{x}_k = \mathbf{y}_{\tau_k} - \hat{\mathbf{x}}_{\tau_k}. \tag{7.6}$$

We will use the concept of a *cycle* to describe the three behaviours: 1) dive-in, 2) manoeuvre and 3) surfacing, as depicted in Fig. 7.1. A cycle $\mathbf{c}_k = \{\Delta\mathbf{x}_k, \mathbf{X}_k\}$ is a tuple containing the dead-reckoned trajectory, $\mathbf{X}_k$, and the measured drift $\Delta\mathbf{x}_k$. Although the current $\mathbf{w}(\mathbf{x})$ throughout each cycle is unknown, we know that the drift measurements are related to the current. Thus, we estimate the underlying ocean current given the cycles. The estimation problem is posed as a maximum a posteriori (MAP) inference problem:

Problem 7.1 (Ocean current estimation)**.** Suppose we have a sequence of cycles $\mathbf{C}_{1:k} = \mathbf{c}_1\mathbf{c}_2 \cdots \mathbf{c}_k$. Find an optimal estimate for ocean current $\mathbf{w}^*(\mathbf{x})$ over the space of 2D smooth

vector fields $\mathcal{C}^\infty(\mathbb{R}^2)$ that maximises the posterior probability:

$$\mathbf{w}^*(\mathbf{x}) = \underset{\mathbf{w}(\mathbf{x})\in\mathcal{C}^\infty(\mathbb{R}^2)}{\arg\max}\ \mathcal{P}(\mathbf{w}(\mathbf{x}) \mid \mathbf{C}_{1:k}). \tag{7.7}$$

Intuitively, solving the MAP problem implies that we find the ocean current $\mathbf{w}(\mathbf{x})$ that is the best trade-off between fitting 1) the drift measurements, $\Delta\mathbf{x}_k$ and 2) a constraint on the general behaviour of ocean current, which we will discuss in Sec. 7.1.3. The constraint is necessary because there are infinitely many possibilities of ocean current vectors that sum up to the drift measurement $\Delta\mathbf{x}_k$. In other words, the problem is *underdetermined*.

Given the online estimate of the ocean current, the aim of navigation problem is to visit a set of target points $\mathbf{X}^d = \{\mathbf{x}_1^d, ..., \mathbf{x}_K^d\}$. To do so, we assume that the vehicle's velocity through water $\mathbf{u}_t$ can be controlled on-board. However, since the vehicle does not have GPS position feedback while underwater, we impose a constraint that the control stays constant in each cycle, i.e., $\forall t \in [\tau_k, \tau_{k+1})$, $\mathbf{u}_t = \mathbf{u}_k$. We wish to find a sequence of control actions that minimises the position error on surfacing:

Problem 7.2 (Navigation)**.** Given the vehicles initial position $\mathbf{x}_{\tau_k}$, estimated flow field $\mathbf{w}^*(\mathbf{x})$, and a target point $\mathbf{x}_k^d$, find the control $\mathbf{u}_k$ that minimises the position error:

$$\begin{aligned} \min_{\mathbf{u}_k} \epsilon_{\mathbf{x}} &\equiv ||\mathbf{x}_k^d - \mathbf{x}_{\tau_k}||, \\ \text{s.t. } \mathbf{x}_{t+1} &= \mathbf{x}_t + (\mathbf{u}_k + \mathbf{w}^*(\mathbf{x}))\Delta t. \end{aligned} \tag{7.8}$$

7.1.2 Planar Incompressible Flows

As aforementioned, solving the estimation problem (7.7) is difficult because the problem is underdetermined, and there are infinitely many possible solutions for the ocean currents $\mathbf{w}(\mathbf{x})$. To this end, we impose a stronger prior by modelling the ocean current as a planar, time-invariant and incompressible flow field. Planarity implies that the ocean current has no z-component, which describes the horizontal stratification of oceanic flow well (see [252, 253]).

A flow field is *incompressible* [254] if its divergence vanishes at all points:

$$\nabla \cdot \mathbf{w}(\mathbf{x}) = 0, \tag{7.9}$$

where $\nabla\cdot$ is the divergence operator. Intuitively, incompressibility implies that 'the amount of water entering a point is equal to the amount exiting the area'. As we do not expect to see surplus or deficit of water entering an area in the ocean, incompressibility is an appropriate description. In fact, incompressibility of water is a common assumption made universally in fluid dynamics [254]. Further, ocean current is almost planar, because the vertical component is much weaker than the horizontal one [253]. We thus make the following assumption about the ocean current $\mathbf{w}(\mathbf{x})$:

Assumption 7.1 (Incompressibility)**.** The ocean current $\mathbf{w}(\mathbf{x})$ is planar incompressible (i.e., satisfies (7.9)).

An important property of planar incompressible flow fields is that it can be fully represented by a scalar-valued *streamfunction* $\phi : \mathbb{R}^2 \to \mathbb{R}$, which can be constructed as a line integral:

$$\phi(\mathbf{x}) = \int_{\mathbf{x}_0}^{\mathbf{x}} (\mathbf{w}(\mathbf{x}) \times d\mathbf{x}), \tag{7.10}$$

where $\times$ denotes the vector cross product and $\mathbf{x}_0 \in \mathbb{R}^2$ is an arbitrary reference point. If a flow field is incompressible, the line integral (7.10) is well defined in the sense that the result is the same for any curve connecting $\mathbf{x}_0$ and $\mathbf{x}$. Therefore, a change in reference point $\mathbf{x}_0$ simply leads to a change in the stream function by a constant offset.

Conversely, given a streamfunction $\phi(\mathbf{x})$, one can compute the current $\mathbf{w}(\mathbf{x})$ as partial derivatives:

$$\mathbf{w}(\mathbf{x}) = \begin{bmatrix} \frac{\partial \phi(\mathbf{x})}{\delta y} & -\frac{\partial \phi(\mathbf{x})}{\delta x} \end{bmatrix}^T. \tag{7.11}$$

7.1.3 Online Current Estimation

We now present an algorithm for estimating the ocean current online. To do so, we first make an observation that the drift measurements are not directly related to the current vectors at all points. This is because drift is the product of current vectors along the

trajectory, but not necessarily the current vectors at other points. Formally, this implies *conditional independence*:

Assumption 7.2 (Conditional independence)**.** For all $\mathbf{x} \in \mathbb{R}^2$ such that $\mathbf{x} \neq \mathbf{x}_t$, $\mathbf{w}(\mathbf{x})$ is conditionally independent of $\mathbf{C}_{1:k}$ given the current along trajectory, $\mathbf{W}_{1:k} = \{\mathbf{W}_1, \cdots, \mathbf{W}_k\}$. In other words, $\mathbf{w}(\mathbf{x})$ is indirectly related to $\mathbf{C}_{1:k}$ through $\mathbf{W}_{1:k}$.

With the assumption, the overall problem can be re-written in a form that reveals two sub-problems:

$$\mathcal{P}(\mathbf{w}(\mathbf{x}) \mid \mathbf{C}_{1:k}) = \int \mathcal{P}(\mathbf{w}(\mathbf{x}) \mid \mathbf{W}_{1:k}) \mathcal{P}(\mathbf{W}_{1:k} \mid \mathbf{C}_{1:k}) d\mathbf{W}_{1:k}. \tag{7.12}$$

Namely, the two sub-problems are as follows:

Problem 7.3 (Drift-based Current Estimation)**.** Estimate the current experienced along the vehicle's trajectory given drift measurements. In other words, find $\mathcal{P}(\mathbf{W}_{1:k} \mid \mathbf{C}_{1:k}))$ in (7.12).

Problem 7.4 (Current Extrapolation)**.** Estimate the current at a remote location given the current along the trajectory. In other words, find $\mathcal{P}(\mathbf{w}(\mathbf{x}) \mid \mathbf{W}_{1:k})$ in (7.12).

Although the two subproblems may seem hierarchical at first (i.e. the solution of Problem 7.3 directly feeds into Problem 7.4), the two are in fact intertwined, and it is necessary to address Problem 7.4 before Problem 7.3. This is because, in order to estimate the current given purely the drift, one must model the correlations between the current vectors at each point along the trajectory in order to reduce the range of plausible solutions. In the following section, we derive a suitable Gaussian process (GP) model for this purpose, based on the physical assumption of *incompressibility*.

7.1.3.1 Incompressible Gaussian Process

In this section, we show how to enforce the incompressibility condition in a GP model using a streamfunction. This addresses the problem of extrapolating the current vectors at a remote location given the current vectors along the trajectory. The incompressibility

assumption improves the extrapolation because we can model the correlation between current vectors with greater accuracy.

As described in Sec. 7.1.2, all incompressible flowas can be represented by a streamfunction. Therefore, we first consider a streamfunction modelled as a GP:

$$\phi(\mathbf{x}) \sim GP(0, k(\|\mathbf{x} - \mathbf{x}'\|)), \tag{7.13}$$

where $k(\|\mathbf{x} - \mathbf{x}'\|)$ is a kernel function.

Because the derivative of a streamfunction ϕ is flow field $\mathbf{w}$ as shown in (7.11) and the derivative of a GP is another GP [255], our flow field can be represented by a GP. To follow the notations in Sec. 3.1.2, the derivative operators can be written as $\mathcal{D} = \begin{bmatrix} \frac{\partial}{\partial y} & -\frac{\partial}{\partial x} \end{bmatrix}^T$ and $\mathcal{D}' = \begin{bmatrix} \frac{\partial}{\partial y'} & -\frac{\partial}{\partial x'} \end{bmatrix}$. Then, the flow field can be represented as:

$$\mathbf{w}(\mathbf{x}) = \mathcal{D}\phi(\mathbf{x}) \sim GP(\mathbf{0}, \mathbf{K}(\mathbf{x}, \mathbf{x}')), \tag{7.14}$$

where the kernel function $\mathbf{K}$ is given by:

$$\begin{aligned} \mathbf{K}(\mathbf{x}, \mathbf{x}') &= \mathcal{D}k(\|\mathbf{x} - \mathbf{x}'\|)\mathcal{D}' \\ &= \begin{bmatrix} \frac{\partial^2 k}{\partial y^2} & -\frac{\partial^2 k}{\partial x \partial y} \\ -\frac{\partial^2 k}{\partial x \partial y} & \frac{\partial^2 k}{\partial x^2} \end{bmatrix}. \end{aligned} \tag{7.15}$$

From the first line to the second line in (7.15), we used the stationarity of the kernel. The merit of applying the derivative operator to the kernel is that (7.15) can be computed analytically given a choice of kernel for the streamfunction [255, 256].

Using the GP representation of flow field $\mathbf{w}$ with the kernel function in (7.15), we can predict a set of current vectors $\mathbf{W}(\mathbf{X}^Q) = \begin{bmatrix} \mathbf{w}(\mathbf{x}_1^Q) & \dots & \mathbf{w}(\mathbf{x}_N^Q) \end{bmatrix}$ given previous measurement data $\mathbf{W}(\mathbf{X}^D) = \begin{bmatrix} \mathbf{w}(\mathbf{x}_1^D) & \dots & \mathbf{w}(\mathbf{x}_M^D) \end{bmatrix}$. The predictions are given as a set of normal random variables:

$$\mathcal{P}(\mathbf{W}(\mathbf{X}^Q) \mid \mathbf{W}(\mathbf{X}^D)) = \mathcal{N}(\mu(\mathbf{X}^Q),\ (\mathbf{X}^Q)), \tag{7.16}$$

with mean and covariance:

$$\mu(\mathbf{X}^Q) = \mathbf{K}_{DQ}^T K_{DD}^{-1} \mathbf{W}_D, \tag{7.17}$$

$$(\mathbf{X}^Q) = \mathbf{K}_{QQ} - \mathbf{K}_{DQ}^T K_{DD}^{-1} \mathbf{K}_{DQ}, \tag{7.18}$$

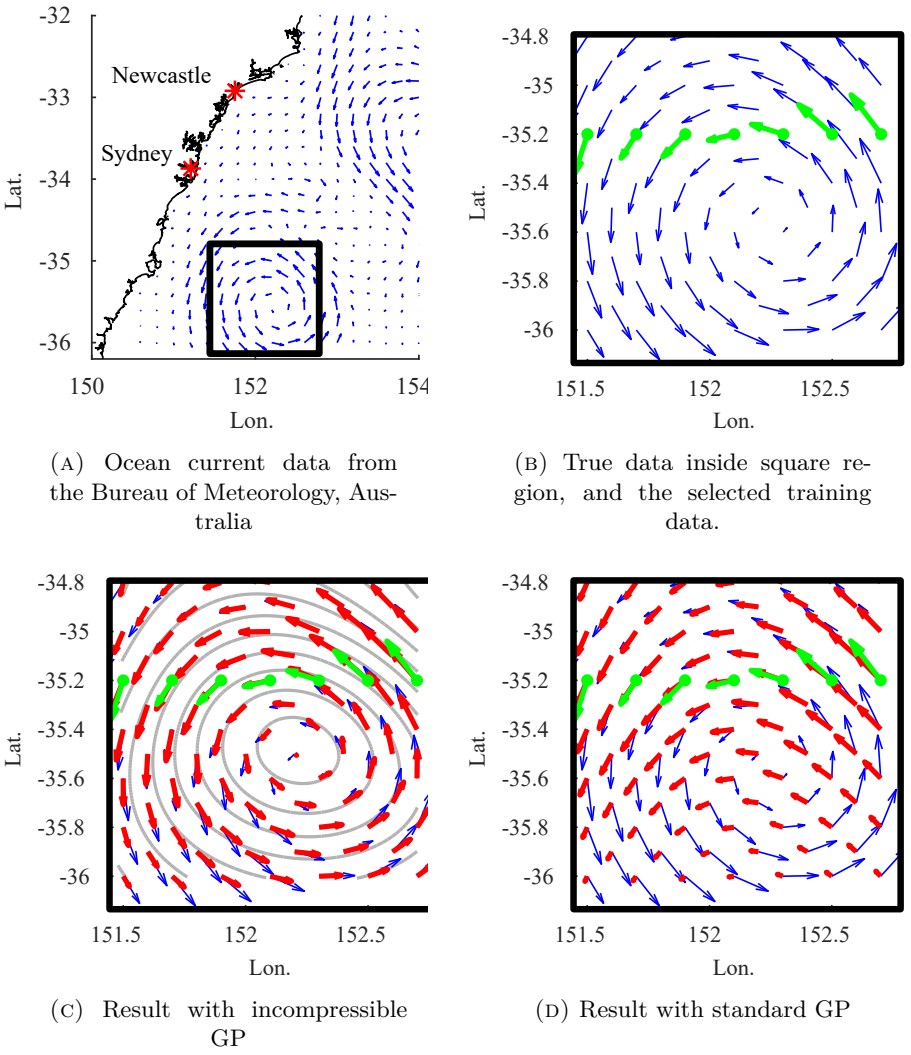
where the matrices $\mathbf{K}_{DD}^{(i,j)} = \left[\mathbf{K}(\mathbf{x}_i^D, \mathbf{x}_j^D)\right]$, $\mathbf{K}_{DQ}^{(i,j)} = \left[\mathbf{K}(\mathbf{x}_i^D, \mathbf{x}_j^Q)\right]$, and $\mathbf{K}_{QQ}^{(i,j)} = \left[\mathbf{K}(\mathbf{x}_i^Q, \mathbf{x}_j^Q)\right]$ are constructed blockwise. Because GP regression yields a continuous estimate, the predicted mean (7.17) and covariance (7.18) do not rely on discretisation; the prediction can be made at any arbitrary point. Further, GP regression allows prediction of ocean current distant from the GPS measurements, depending on the lengthscale used for GP.

We now investigate if the incompressible GP is a better representation of the oceanic currents using a real dataset in Fig. 7.2. We selected a representative eddy from the east Australian current data provided by the Australian Bureau of Meteorology. Then, we selected training samples along a line to emulate the current estimated along a trajectory. These training samples were extrapolated with GPs having the proposed incompressible kernel and the standard squared-exponential kernel $\mathbf{K}_{SE}(\mathbf{x}, \mathbf{x}') = \mathbf{diag}(k_{SE}(\mathbf{x}, \mathbf{x}'), k_{SE}(\mathbf{x}, \mathbf{x}'))$.

The standard kernel only fits a smooth vector field to the training samples. Meanwhile, the proposed incompressible GP extrapolates the ocean current much more accurately even with the limited training samples. An apparent benefit is that we can reconstruct eddy-like patterns [47], which leads to a better extrapolation for the flow along a future trajectory given the estimate along the present trajectory.

7.1.3.2 Expectation-Maximisation for Drift-based Estimation

In this section, we solve Problem 7.3, which concerns estimating the current along trajectory. For simplicity, we will focus on estimating the flow along trajectory sequentially, given each incoming GPS measurements. In doing so, we are making a Markov assumption, where we fix the estimate of current along previous trajectories, $\mathbf{W}_{1:k-1}$, when estimating the current along the present trajectory, $\mathbf{W}_k$. It substantially reduces the computational complexity of the problem, as the algorithm is *incremental*. This is the key enabler for online implementation of the algorithm, because the key computational bottleneck of GP



(A) Ocean current data from the Bureau of Meteorology, Australia

(B) True data inside square region, and the selected training data.

(C) Result with incompressible GP

(D) Result with standard GP

FIGURE 7.2: Comparison between incompressible GP and the standard multi-output GP. Blue: true data. Green: the training data used for regression. Red: estimated current. Gray: reconstructed streamline (only available with the proposed incompressible GP). Data were selected to emulate current estimated along a trajectory at each cycle. The proposed incompressible GP is capable of identifying large-scale eddy-like patterns, and hence offers better extrapolation.

inference is significantly reduced by only considering the current vectors along the present trajectory. More precisely, we assume:

$$\mathcal{P}(\mathbf{W}_{1:k} \mid \mathbf{C}_{1:k}) = \mathcal{P}(\mathbf{W}_k \mid \mathbf{C}_k, \mathbf{W}_{1:k-1})\mathcal{P}(\mathbf{W}_{1:k-1} \mid \mathbf{C}_{1:k-1}),$$

which shows the problem reduces to estimating the current along present trajectory, given GPS measurements and previous current estimates (i.e., $\mathcal{P}(\mathbf{W}_k \mid \mathbf{C}_k, \mathbf{W}_{1:k-1})$).

The main challenge in estimating the current along trajectory arises from the strong causality between the trajectory itself and the current along the trajectory. To predict the current along trajectory, we must know the trajectory beforehand, and to predict the trajectory, we must know the current along trajectory. The challenge is solved through an expectation-maximisation (EM) algorithm. In each iteration, i, the EM algorithm alternates between 1) estimating the trajectory $\mathbf{X}_k^i$, called the *expectation step* (E-step) 6; and 2) estimating the current along trajectory $\mathbf{W}_k^i$ given $\mathbf{X}_k^i$, called the *maximisation step* (M-step). In effect, we iteratively 'guess' the true trajectory, estimate the flow using the guess, and refine the guess on trajectory using the estimated flow.

Problem 7.3 is re-written in the EM formulation:

$$\begin{aligned}\mathcal{P}(\mathbf{W}_k \mid \mathbf{C}_k, \mathbf{W}_{1:k-1}) \propto &\int \mathcal{P}(\mathbf{W}_k \mid \mathbf{X}_k, \mathbf{C}_k, \mathbf{W}_{1:k-1})\mathcal{P}(\mathbf{X}_k \mid \mathbf{W}_k, \mathbf{C}_k)d\mathbf{X}_k \\ =&\mathbb{E}_{\mathbf{X}_k|\mathbf{W}_k,\mathbf{C}_k}[\mathcal{P}(\mathbf{W}_k \mid \mathbf{X}_k, \mathbf{C}_k, \mathbf{W}_{1:k-1})].\end{aligned} \tag{7.19}$$

In the E-step, we find $\mathcal{P}(\mathbf{X}_k^i \mid \mathbf{W}_k^{i-1}, \mathbf{C}_k)$ and evaluate the expectation (7.19). In the M-step, we find the posterior estimate of the currents along the trajectory:

$$\mathbf{W}_k^i = \arg\max_{\mathbf{W}} \mathbb{E}_{\mathbf{X}_k^i|\mathbf{W}_k^{i-1},\mathbf{C}_k}[\mathcal{P}(\mathbf{W} \mid \mathbf{X}_k^i, \mathbf{C}_k, \mathbf{W}_{1:k-1})]. \tag{7.20}$$

E-step: In the E-step, we need to compute the expectation in (7.19). In (7.2), notice that the only source of uncertainty derives from the fact that $\mathbf{w}(\mathbf{x})$ is a GP. Therefore, given current, the trajectory is fully known. Formally, the conditional distribution of the trajectory becomes a Dirac delta distribution:

$$\mathcal{P}(\mathbf{X}_k^i \mid \mathbf{W}_k^{i-1}, \mathbf{C}_k) = \delta(\mathbf{X}_k^i - (\mathbf{X}_k + B\mathbf{W}_k^{i-1})), \tag{7.21}$$

where $B^{(i,j)} = \left[\Delta t \mathbf{I}_2\right]$ if $i \leq j$, and $B^{(i,j)} = \left[\mathbf{0}_{2\times 2}\right]$ otherwise. $\mathbf{I}_2$ and $\mathbf{0}_{2\times 2}$ denote identity and zero matrices.

Consequently, the expectation integral collapses to a mere evaluation at:

$$\mathbf{X}_k^i = \mathbf{X}_k + B\mathbf{W}_k^{i-1}. \tag{7.22}$$

M-step: In the M-step, we maximise the expectation taken in the E-step. From the E-step, the expectation (7.19) is a simple evaluation at (7.22). As $\mathbf{w}(\mathbf{x})$ is a GP, this yields a prior given by a Gaussian random variable:

$$\mathbb{E}_{\mathbf{X}_k^i|\mathbf{W}_k^{i-1},\mathbf{C}_k}\left[\mathcal{P}(\mathbf{W}_k^i \mid \mathbf{X}_k^i, \mathbf{W}_{1:k-1})\right] = \mathcal{N}(\mu(\mathbf{X}_k^i),\ (\mathbf{X}_k^i)), \tag{7.23}$$

where $\mu(\mathbf{X}_k^i)$ and $(\mathbf{X}_k^i)$ are calculated using GP prediction equations (7.17) and (7.18) given the current estimated with previous drift measurements, $\mathbf{W}_{1:k-1}$.

Notice that we can write $\Delta\mathbf{x}_k$ in terms of $\mathbf{W}_k$ as:

$$\Delta\mathbf{x}_k = C\mathbf{W}_k + \epsilon_k, \tag{7.24}$$

where $C = \Delta t \begin{bmatrix}\mathbf{I}_2 & \mathbf{I}_2 & \dots & \mathbf{I}_2\end{bmatrix}$. Thus, $\mathbf{W}_k^i$ and $\Delta\mathbf{x}_k$ are joint normal random variables:

$$\begin{aligned}&\mathbb{E}_{\mathbf{X}_k^i|\mathbf{W}_k^{i-1},\mathbf{C}_k}\left[\mathcal{P}(\mathbf{W}_k^i, \Delta\mathbf{x}_k \mid \mathbf{X}_k^i, \mathbf{W}_{1:k-1})\right] \\ &\qquad = \mathcal{N}\left(\begin{bmatrix}\mu(\mathbf{X}_k^i) \\ C\mu(\mathbf{X}_k^i)\end{bmatrix}, \begin{bmatrix}\Sigma(\mathbf{X}_k^i) & \Sigma(\mathbf{X}_k^i)C^T \\ C\Sigma(\mathbf{X}_k^i) & C\Sigma(\mathbf{X}_k^i)C^T + \sigma_y^2 I_{2\times 2}\end{bmatrix}\right).\end{aligned} \tag{7.25}$$

The merit of this formulation is that the maximising solution is now given in closed form, because maximising the posterior (7.20) is equivalent to finding the conditional mean using (7.25). This is given by [216]:

$$\mathbf{W}_k^i = \mu + \Sigma C^T\left(C\Sigma C^T + \sigma_y^2 I\right)^{-1}(\Delta\mathbf{x}_{T_k} - C\mu), \tag{7.26}$$

where $\mu = \mu(\mathbf{X}_k^i)$ and $\Sigma = \Sigma(\mathbf{X}_k^i)$.

7.1.3.3 Implementation

The algorithm for solving Problem 7.1 is shown in Alg. 2. We initialise the algorithm with a zero-mean GP without any measurements (Alg. 2 line 1). From lines 4 to 7, we estimate the true trajectory and the current along the trajectory, $\mathbf{X}_k$ and $\mathbf{W}_k$ using our EM algorithm based on incoming GPS measurement. The current along trajectory is then

Algorithm 2 GP-EM algorithm for ocean current estimation

$GP \leftarrow$ InitialiseEmptyGP
while Vehicle is operational **do**
  if Drift measurement $\Delta\mathbf{x}_k$ available **then**
    Initialise $\mathbf{X}_k^0 \leftarrow \mathbf{X}_k$
    for i=1 …N **do**
      Update estimate of $\mathbf{W}_k^i$ using (7.26) with $\mathbf{X}_k^{i-1}$
      Update estimate of $\mathbf{X}_k^i$ using (7.22) with $\mathbf{W}_k^i$
    $GP \leftarrow \text{UpdateGP}(GP, \mathbf{W}_k^N, \mathbf{X}_k^N)$

added to the measurement set of the GP, as 'pseudo-target' [120]. As more measurements become available, the GP produces better prior for the iteration.

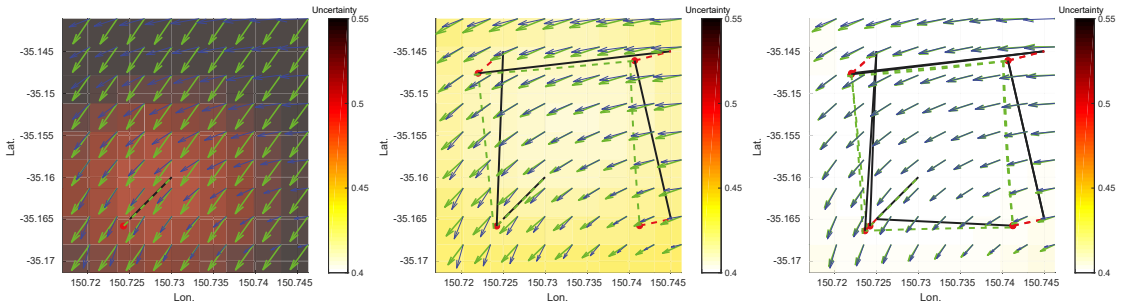
7.1.4 Control Generation for Navigation

Given the estimated ocean current $\mathbf{w}^*(\mathbf{x})$, we use a simple model-predictive control algorithm to generate the control action $\mathbf{u}_k$ for navigation. This is achieved by directly simulating the dynamic model (7.1) using the estimated current $\mathbf{w}^*(\mathbf{x})$ for a given value of $\mathbf{u}_k$, and evaluating the position error (7.8). For simplicity, we used MATLAB's `fminsearch`[1] function to find the optimal control action $\mathbf{u}_k$. Although it may be possible to improve the computational speed, it was found that the speed of `fminsearch` is sufficient for underwater glider applications, where the typical dive duration is on the order of hours.

7.1.5 Results

In this section, we empirically analyse our framework in three settings: 1) simulation; 2) offline evaluation; and 3) online evaluation. The simulation experiment evaluates the performance of the online current estimation algorithm (Sec. 7.1.3) in isolation, with randomised flow fields. The offline evaluation experiment examines the behaviour of the online current estimation algorithm in isolation with a dataset collected from a field trial. The online evaluation experiment tests the entire framework including control generation in the open ocean.

[1] https://www.mathworks.com/help/matlab/ref/fminsearch.html



(A) Estimated flow field after 1 waypoint

(B) Estimated flow field after 4 waypoints

(C) Estimated flow field after 8 waypoints

FIGURE 7.3: Simulation results for online current estimation. Trajectory converted to lat-long for end-to-end testing. Black solid line: dead-reckoned trajectory. Green dashed line: reconstructed trajectory. Red markers: GPS. Red dashed line: drift. Blue arrows: true flow field. Green arrows: estimated flow field. Uncertainty refers to trace of covariance.

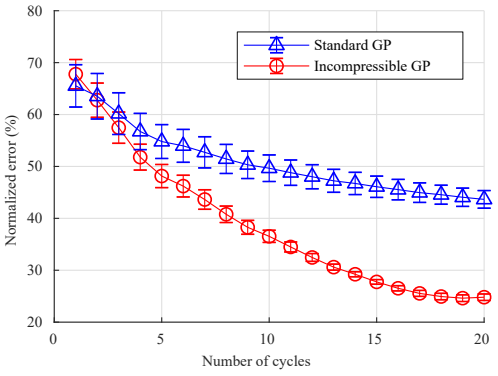


FIGURE 7.4: Comparison of convergence between standard (blue, triangle) and incompressible GPs (red, circle) for 100 different gyre patterns. Error normalized by the magnitude of the true current. 99% confidence interval is shown.

We used the squared exponential kernel [75] for the streamfunction, having lengthscale of $\ell = 35km$, and self-variance of $\sigma_W^2 = 0.5m^2s^{-2}$. These hyperparameters were found by maximising the likelihood of a dataset from the Australian Bureau of Meteorology using a standard procedure [75]. The kernel for the current vectors was computed analytically with (7.15).

7.1.5.1 Simulation Results

We validate the performance of the proposed framework with simulated data. We simulated an underwater vehicle travelling in a flow field, with true and dead-reckoned estimates evolving as (7.1) and (7.3). The vehicle is given four waypoints, and surfaces when

its *dead-reckoned* position estimate obtained using (7.3) is within 100 metres from the current target waypoint. For the purpose of validation, we ran the proposed algorithm on a double-gyre model. The results are shown in Fig. 7.3.

From Figs. 7.3a to 7.3c, it can be seen that the algorithm actively improves the estimate of current as the mission progresses. In Fig. 7.3a, it can be seen that the estimate after only one cycle is as good as the average current method in [134]. However, by the fourth cycle, it can be seen that the estimated and the true flow fields are already in good agreement, with minor disparity. By the eighth cycle, the estimated and the true flow fields are almost indistinguishable. The uncertainty of the estimated current also decreases.

To examine the convergence of the proposed algorithm further, we performed a Monte Carlo simulation with 100 randomly generated double gyre patterns. We took the error between predicted and true currents, normalised by the magnitude of the true current. The convergence was compared with a standard GP with squared-exponential kernel function $\mathbf{K}(\mathbf{x}, \mathbf{x}') = \mathbf{diag}(k_{SE}(\mathbf{x}, \mathbf{x}'), k_{SE}(\mathbf{x}, \mathbf{x}'))$ having identical parameters, but without incompressibility assumption. The result is shown in Fig. 7.4.

Fig. 7.4 shows that the algorithm gradually learns any randomly generated flow field, which is demonstrated by the decrease in normalised error for both standard and the proposed incompressible GP. However, the incompressible GP shows a much faster rate of convergence and a lower steady-state error than the standard. The result clearly indicates that our incompressible GP outperforms over the standard in describing oceanic flows as shown in Fig. 7.2.

7.1.5.2 Offline Field Evaluation

We test the proposed algorithm in an offline setting with a drift dataset from a field operation near Jervis Bay, Australia. Given a drift dataset, the current is reconstructed offline, and we qualitatively examine its fidelity. The dataset and the results are shown in Fig. 7.5. The glider was tasked to visit designated waypoints using dead-reckoning only, and communicate the current cycle $\mathbf{c}_k$ when on surface. For the purpose of experiment,

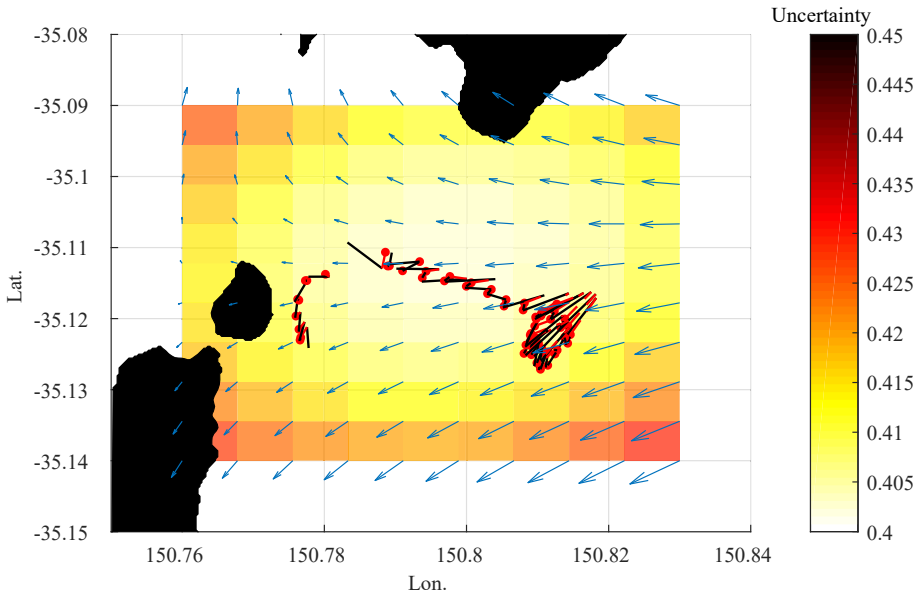


FIGURE 7.5: Offline evaluation with Jervis Bay field trial data. Blue arrows: estimated current. Red circles: position from GPS, Solid black lines: dead-reckoned trajectory.

we disabled the onboard *average current correction* [134] to ensure the dead-reckoned estimates evolve as (7.3).

For the Jervis Bay trial in Fig. 7.5, it can be seen that the estimated current is remarkably consistent with what is expected near a bay: ocean flow comes in from the open ocean, and the majority enters the bay, with only a small perpendicular component to the bay's shoreline. This indicates that the proposed incompressible GP accurately models the correlation between ocean currents at different points, as incompressibility is a physical attribute of the real ocean. The improvement in prediction at a remote location, in turn, leads to improvement in non-myopic planning.

7.1.5.3 Online Field Evaluation

We tested the entire closed-loop OCEAN framework in the open ocean in an online setting. A field trial was conducted between 20th August 2018 and 9th October 2018 including three days of preparation. The glider was deployed on 23rd August around Perth, Australia and successfully recovered on 9th October. During this time, the glider was tasked to visit a 2km square sequentially, using the OCEAN framework to update estimate of ocean current

and to generate control actions. Details on the software implementation is described in Appendix B.

The control command $\mathbf{u}_k$ is sent to the glider as a *virtual waypoint*. A virtual waypoint is defined as the position the glider will reach if it executes action $\mathbf{u}_k$ in the absence of ocean currents:

$$\hat{\mathbf{x}}_k^d = \mathbf{x}_{\tau_{k-1}} + \mathbf{u}_k(\tau_k - \tau_{k-1}). \tag{7.27}$$

If the estimated current is accurate, then dead-reckoning toward a virtual waypoint will allow the glider to correctly reach the desired target point.

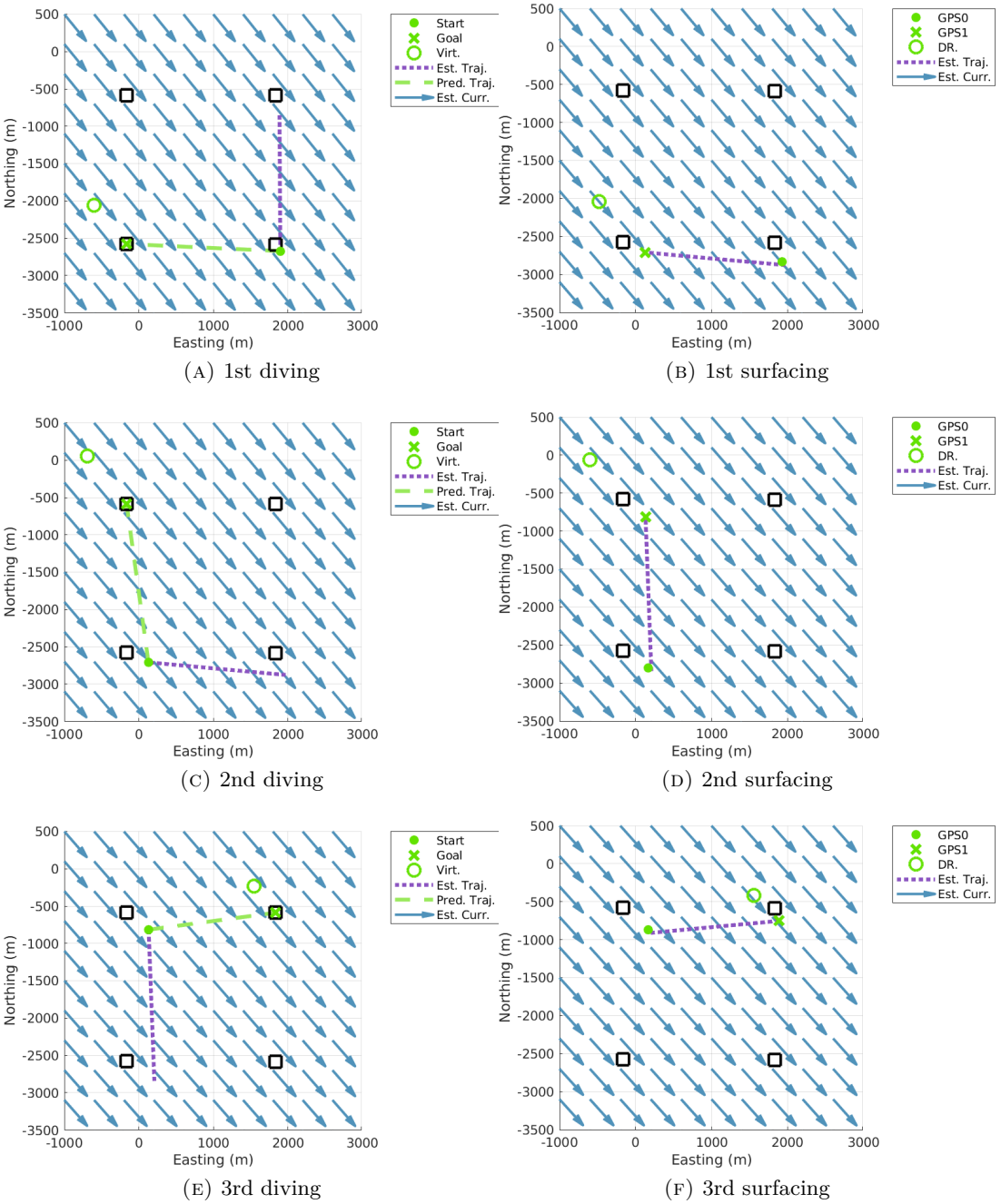
We examine the performance in terms of position error (7.8) as well as *control error*. Control error arises because the Slocum G2 glider's operation is constrained by safety features such as maximum diving time, which prevents dead-reckoning towards the virtual waypoint, and causes premature surfacing while en route. The control error ϵ_c is thus defined by the distance between the virtual waypoint and the dead-reckoning position:

$$\epsilon_c = \|\hat{\mathbf{x}}_k^d - \hat{\mathbf{x}}_{\tau_k}\|. \tag{7.28}$$

Typical behaviours from the trial is shown in Fig. 7.6. In Fig. 7.6b, 7.6d, 7.6f, the glider surfaces and updates the ocean current estimation. Prior to diving, the glider computes a virtual waypoint. Ideally, we expect to see that the virtual waypoint overlaps with the dead-reckoned position, and the desired target point with the GPS position.

In Fig. 7.6, it can be seen that the virtual waypoints computed based on the estimated ocean current generally steers the glider towards the desired target point. Using the virtual waypoint reduces the position error below 500m in most cases, as shown in Fig. 7.7. Yet, there are cases with up to 2km position error, which is substantial.

A detailed analysis reveals that large position error is mostly attributed to control error (i.e., premature surfacing triggered by glider). An instance of this failure mode is shown in Fig. 7.8. Although the glider is commanded to a virtual waypoint at around $(x, y) = (-500, 500)$, it surfaces prematurely at $(0, -500)$, which leads to a large position error. This occurs because the glider is travelling directly against the current, and is thus slowed



(A) 1st diving (B) 1st surfacing

(C) 2nd diving (D) 2nd surfacing

(E) 3rd diving (F) 3rd surfacing

FIGURE 7.6: Typical behaviours during online evaluation experiments. Using updated ocean current after surfacing, virtual waypoints are generated (green circle). The target position (green cross) is met by the true position (green cross in 2nd column)

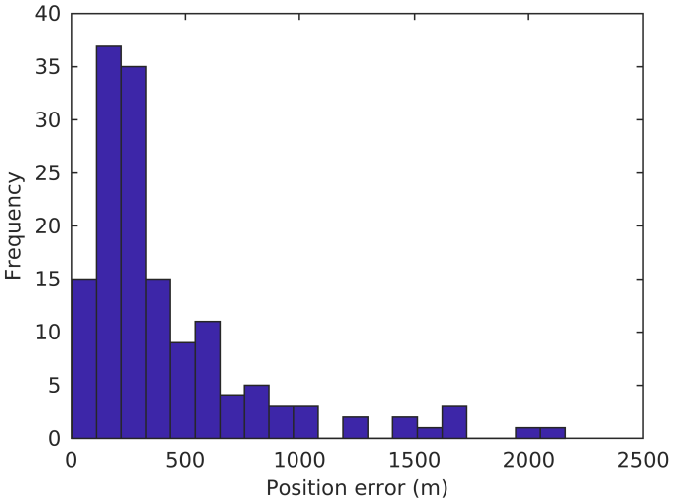
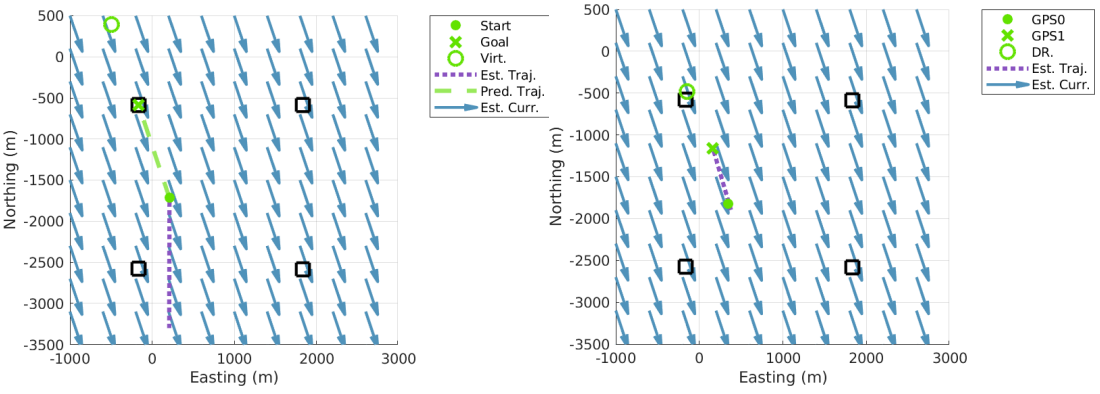


FIGURE 7.7: Distribution of position error



(A) Diving (B) Surfacing

FIGURE 7.8: Instance of failure due to control error. The glider does not correctly dead-reckons to virtual waypoint because of dive-time safety feature.

down as a result. The maximum dive-time fail-safe is then activated which causes the glider to surface.

Fig. 7.9 supports our hypothesis on correlation between control and position error. The scatter plot suggests that there is a strong linear correlation between control and position error, meaning that occasional large current estimation error is caused by glider's inevitable internal surfacing mechanism (i.e., control), not our framework. The strong linear correlation appears because although the glider is driven to the virtual waypoint, it does not in fact surface at the commanded virtual waypoint due to internal logic. In

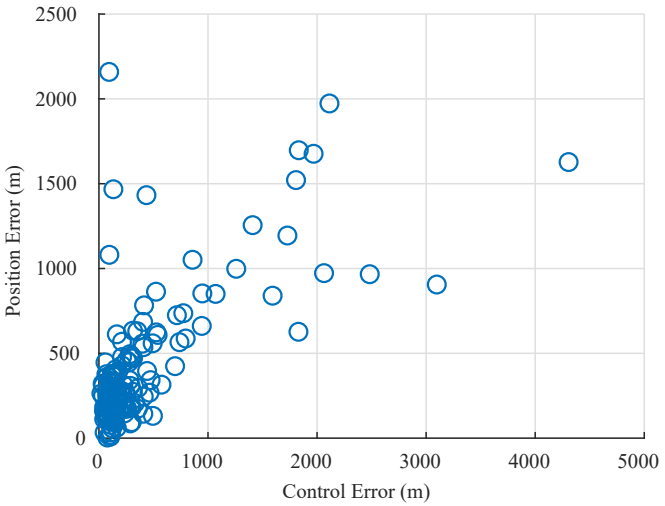


FIGURE 7.9: Correlation between control error (between the compensated virtual waypoint and dead-reckoning position) and position error (between the true desired waypoint and the actual surfacing position). Control error must be zero, so that the glider dead-reckons toward virtual waypoints, but it is not due to the dive-time safety feature. Position error is largely explained by control error.

other words, if control error is reduced (i.e., dead reckoning works well), there is a strong chance that the glider surfaces closer to the desired location.

7.2 Distance Field Estimation with Log-GPIS

We now consider the general problem of autonomous navigation in an indoor environment. In particular, we focus on the obstacle representation, and aim to provide an accurate continuous representation using an Euclidean distance field (EDF). In constructing an EDF, a fundamental challenge is representing the (unbounded) increase in distance values across space as we move further from the obstacles. Discrete techniques [80–82, 257] handle this issue by restriction to a finite grid, whereas no clear solutions have been presented for continuous approaches [251, 258]. We focus on the fact that the behaviour of an EDF is elegantly described by the Eikonal equation, and combine it with GPs to improve the mathematical fidelity of the prediction. This allows a more accurate prediction of distance values further from the points measured by the sensor.

7.2.1 Background

7.2.1.1 Euclidean Distance Field

Consider an obstacle represented as a manifold $S \subset \mathbb{R}^D$ with boundary ∂S, of which the orientation is given by the normal $\mathbf{n}$. For all $\mathbf{x} \in \mathbb{R}^D$, the EDF $d(\mathbf{x})$ is the closest distance to the boundary ∂S:

$$d(\mathbf{x}) = \min_{\mathbf{y} \in \partial S} |\mathbf{x} - \mathbf{y}|. \tag{7.29}$$

Consider the scenario where the boundary ∂S dilates along its normal direction with unit speed. Then, the arrival time will be equal to the distance $d(\mathbf{x})$. This intuition is captured by the Eikonal equation [259], which is given by

$$|\nabla d(\mathbf{x})| = 1 \quad \mathbf{x} \in \mathbb{R}^D, \tag{7.30}$$

with boundary constraints,

$$d(\mathbf{x}) = 0 \quad \text{and} \quad \partial d(\mathbf{x})/\partial \boldsymbol{n} = 1 \quad \text{on } \mathbf{x} \in \partial S. \tag{7.31}$$

Suppose we are given sparse measurements of the points on surface, $\mathcal{X} = \{\mathbf{x}_i\} \subset \partial S$, $i = 1 \dots N$. Our aim of is to estimate $d(\mathbf{x})$ given $\mathcal{X}$, thereby reconstructing S.

7.2.1.2 Gaussian Process Implicit Surfaces

Gaussian process implicit surface (GPIS) techniques [251, 258, 260, 261] use GP regression to estimate the distance field of the surface. Consequently, the surface is given by the zero-level set of the distance field. Most standard GPIS techniques directly model the distance field as a GP [258], and often use normal measurements to ensure correct results [251]. However, although standard GPIS allows continuous and probabilistic representation of the surface with high precision and continuity of the distance field near the surface, it is often highly inaccurate further from the surface. For example, using a zero-mean GP implies that the prediction will include false-positive artifacts as the predicted function

value falls back to zero. Our method, *log-GPIS*, alleviates this issue, by elegantly enforcing the Eikonal equation (7.30) into GP regression.

7.2.2 Varadhan's Formula

A major challenge in exploiting the Eikonal equation (7.30) as we did in previous chapters is that it is non-linear and hyperbolic. It is therefore difficult to solve directly, and special numerical approaches are often required, including discrete and combinatorial methods such as fast marching or label-correcting to propagate the distance field through a grid (e.g., [257] [262]).

A recent work in the computer graphics literature [263] presents a smooth alternative based on Varadhan's distance formula [264] [265], which approximates the EDF $d(\mathbf{x})$ using the heat kernel on the manifold S. The physical intuition behind Varadhan's formula is as follows. Imagine that the target surface S is hot and emanates heat. The conduction of heat can be viewed as particles taking a random walk starting from the boundary ∂S. If we restrict the duration of the random walks to be short (i.e., as time $t \to 0$), the paths taken will be close to the shortest possible one.

Heat conduction on S is modelled by the heat kernel, denoted by $v(\mathbf{x})$. The heat kernel, which we denote by $v(\mathbf{x})$, is the solution of the homogeneous screened Poisson equation.

$$\begin{aligned} &(1 - t\Delta)v = (1/t - \Delta)v = 0 \quad \text{in } S, \\ &v = 1 \quad \text{on } \partial S, \end{aligned} \tag{7.32}$$

where $\Delta = (\sum_i \frac{\partial^2}{\partial x_i^2})$ is the Laplace operator and $1/t$ is a positive parameter that controls 'screening'. An apparent merit of the screened Poisson equation (7.32) is that it is linear, unlike the Eikonal equation (7.30).

The celebrated result of Varadhan (Theorem 2.3, [264]) is that the heat kernel $v(\mathbf{x})$ and the EDF $d(\mathbf{x})$ are related as follows:

$$d(\mathbf{x}) = \lim_{t \to 0} \{-\sqrt{t} \ln[v(\mathbf{x})]\}, \tag{7.33}$$

where it can be seen that the approximation gets better as t approaches zero. Consequently, we define the approximation of $d(\mathbf{x})$ as $u(\mathbf{x})$:

$$u(\mathbf{x}) = -\sqrt{t} \ln v(\mathbf{x}). \tag{7.34}$$

In this case, t serves as a smoothing factor of $u(\mathbf{x})$. Accordingly, rewriting $v(\mathbf{x})$ in terms of $u(\mathbf{x})$,

$$v(\mathbf{x}) = \exp\{-u(\mathbf{x})/\sqrt{t}\}. \tag{7.35}$$

Substituting $v(\mathbf{x})$ from (7.35) in (7.32) leads to:

$$v - t\Delta v = v\left[\left(1 - |\nabla u|^2\right) + \sqrt{t}\Delta u\right] = 0, \tag{7.36}$$

which shows that we effectively solve the regularised form of the Eikonal equation.

Although the theory described thus far only applies to the case of $\mathbf{x} \in S$ (due to the domain constraint in (7.32)), we can easily extend it to $\mathbf{x} \in \mathbb{R}^D$. To do so, consider the EDF of the complement, S^c. The corresponding heat equation (7.32) is then defined in S^c. Meanwhile, the boundary condition remains the same, because $\partial S^c = \partial S$. Combining the two, we have:

$$\begin{aligned} v - t\Delta v &= 0 \quad \text{in } \mathbb{R}^D, \\ v &= 1 \quad \text{on } \partial S, \end{aligned} \tag{7.37}$$

in place of (7.32).

In doing so, we have lost the information on whether $\mathbf{x}$ is inside or outside S (i.e., the sign is lost), because we have 'stitched' the EDFs on S and S^c. In our view, the loss of sign is a small price to pay compared to the convenience of using (7.37) to approximate the (unsigned) distance field everywhere on $\mathbb{R}^D$.

7.2.3 Log-Gaussian Process Implicit Surfaces (Log-GPIS)

In this section, we describe how to exploit Varadhan's formula in GP regression. Based on Varadhan's formula, we find that it is advantageous to model the heat kernel $v(\mathbf{x})$ (i.e., the

exponential of the EDF) as a GP, instead of the EDF $d(\mathbf{x})$. This is because $v(\mathbf{x})$ is governed by the screened Poisson partial differential equation (PDE) (7.37), which is linear unlike the Eikonal (7.30) or the regularised Eikonal (7.36) equations. Linear PDE constraints can be naturally incorporated into GP regression by ensuring that the covariance function follows the linear PDE [217], which we will describe shortly.

Assuming that the choice of covariance function respects (7.37), we estimate $v(\mathbf{x})$ by simply using the GP regression equation (3.6) with target measurements set to $y_i = 1$ at locations $\mathbf{x}_i$ on the surface. Intuitively, this implies the target values in terms of the EDF is $\log(1) = 0$ on the surface boundary. Similarly, this enforces the boundary condition in (7.37).

Subsequently, we use Varadhan's formula (7.34) to recover the EDF from the predictive mean and covariance function:

$$\bar{d}_* = -\sqrt{t} \ln \bar{f}_*. \tag{7.38}$$

Remarkably, (7.38) shows that we can incorporate the Eikonal equation (7.30) by a careful choice of covariance and a simple log-transformation, which justifies the name of log-GPIS. Despite its simplicity, the log-transformation has a non-trivial benefit that the predicted distance will approach infinity as we query points further away from the surface unlike the standard GPIS, which will predict distance value of zero. In other words, log-GPIS solves the issue of undesirable artifacts in the predicted surface geometry.

The log-transformation also affects the gradient. Taking the gradient of both sides of (7.38) respect to the distance,

$$\nabla \bar{d}_* = \frac{-\sqrt{t}}{\bar{f}_*} \nabla \bar{f}_*, \tag{7.39}$$

where it can be seen that the gradient of $\bar{d}_*$ is in the opposite direction as the gradient of $\bar{f}_*$, subject to a scaling factor $\sqrt{t}/\bar{f}_*$. However, because the Eikonal equation (7.30) requires that the magnitude of the gradient is normalised to 1, the scaling factor is unimportant, and we simply need to normalise the gradient of $\bar{f}_*$, and invert the sign.

Based on the gradient (7.39), the predictive variance can be calculated using a first-order approximation:

$$\mathbb{V}\left[d_*\right] = \frac{-\sqrt{t}}{\bar{f}_*} \mathbb{V}\left[f_*\right] \frac{-\sqrt{t}}{\bar{f}_*}^\top. \tag{7.40}$$

7.2.4 Choice of Covariance Function

In this section, we derive a covariance function that satisfies the screened Poisson PDE (7.32), inspired by Sarkka's work [217]. The main result is that the *Whittle kernel* [266], which is a special case of the Matérn family of order 1 [75], satisfies the screened Poisson PDE (7.37) exactly. The Whittle kernel is given by:

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \frac{\left|\mathbf{x} - \mathbf{x}'\right|}{2\lambda} K_\nu\left(\lambda \left|\mathbf{x} - \mathbf{x}'\right|\right). \tag{7.41}$$

Here, K_ν is the modified Bessel function of the second kind of order $\nu = 1$. λ is the characteristic length scale hyperparameter, and is set as $\lambda = 1/\sqrt{t}$. In choosing the hyperparameter λ, two considerations need to be made. The first is that as $\lambda \to \infty$, the log-GPIS inference equation (7.38) will satisfy the Eikonal equation (7.30) more faithfully. This is because (7.33) shows that when $t \to 0$ it will produce a better distance approximation, which means the value of λ gets bigger. Meanwhile, the choice of hyperparameter also affects the computational stability of GP inference, and also the reconstruction accuracy for the heat kernel $v(\mathbf{x})$. Therefore, λ must be set as large as possible, while ensuring computational stability and reconstruction accuracy are not affected.

We present why the Whittle kernel (7.41) satisfies the screened Poisson PDE (7.32). For brevity, we only consider the 2D case, where we may expand (7.32) as:

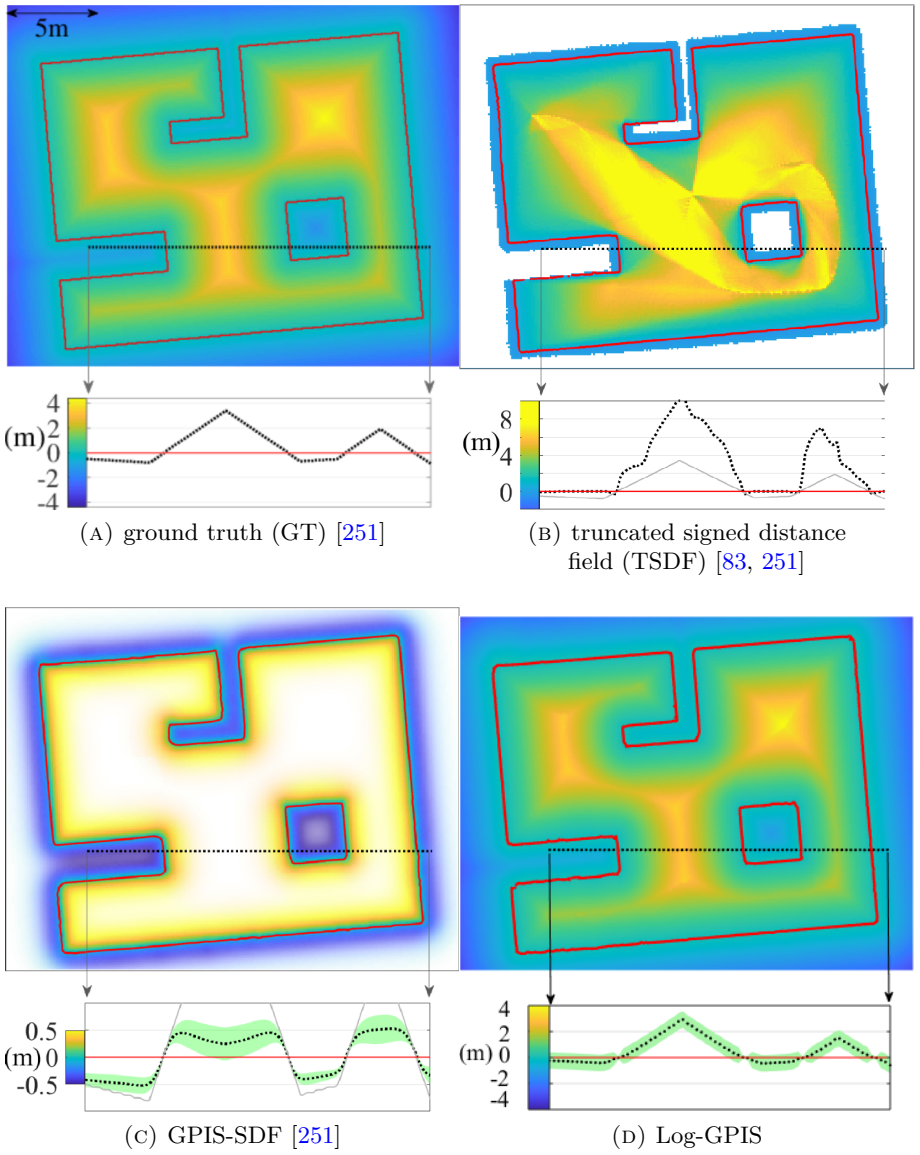
$$\frac{\partial^2 v\left(x_1, x_2\right)}{\partial x_1^2} + \frac{\partial^2 v\left(x_1, x_2\right)}{\partial x_2^2} - \lambda^2 v\left(x_1, x_2\right) = w\left(x_1, x_2\right), \tag{7.42}$$

where $w(x, y)$ is the white noise. To exploit the Wiener-Khinchin theorem [75], we take the Fourier transform of (7.42) and obtain the spectral density:

$$S\left(\omega_1, \omega_2\right) = \frac{1}{\left(\omega_1^2 + \omega_2^2 + \lambda^2\right)^2}. \tag{7.43}$$

Now, the Wiener-Khinchin theorem [75] provides that the covariance function is given by the inverse Fourier transform of the spectral density (7.43), which is the Whittle kernel (7.41).

To summarise, log-GPIS is capable of producing EDF estimates that faithfully satisfy the Eikonal equation (7.30) by simply applying a log-transform to a GPIS with a covariance of the Matérn family. The implication is that there is no need to provide sample points inside and outside the surface. Our algorithm only requires the measurements around the surface, and the EDF will be predicted accurately throughout the entire domain.



(A) ground truth (GT) [251]

(B) truncated signed distance field (TSDF) [83, 251]

(C) GPIS-SDF [251]

(D) Log-GPIS

FIGURE 7.10: Comparison of inferred distance fields in a 2D dataset. a) The ground truth. b) TSDF [83, 251]. c) GPIS-SDF [251]. d) Log-GPIS (proposed). All methods accurately reconstruct the obstacle boundary (solid red line), but only log-GPIS infers the EDF accurately over the whole field. The bottom subplots show the EDF values along the black line. N.b., the colormap is differs among different maps for visualisation purposes, except for the GT and ours. In the subplots of c) and d), the green area is variance.

7.2.5 Evaluation

We compare the performance of log-GPIS against state of the art techniques including GPIS-signed distance field (SDF) [251] and TSDF [83] in a 2D navigation setting. For fairness, we use the same dataset as [251], which consists of 28 LIDAR scans from different robot poses with angular range of $-135°$ to $+135°$, sensing resolution of $1°$, and noise $\sigma = 0.01m$.

As aforementioned, our approach does not contain sign information. However, we can simply recover the sign of the log-GPIS by comparing the predicted gradient of each testing point with sensor position. If the gradient is in the opposite direction of the sensor, we flip the sign of distance value of the testing point. This recovery procedure was considered during comparison.

The results of TSDF [83], GPIS-SDF [251], and our method log-GPIS and the ground truth are shown in Fig. 7.10. The sub-figures at the bottom show the distance field sampled along the solid black line. The ground truth in Fig. 7.10a) shows that the distance field grows linearly away from the surface. Because the signed distance field is shown, the surface is exactly at the zero crossing. Fig. 7.10b) shows the distance field estimated by the TSDF method [83] with constant weights. Although TSDF still produces an accurate continuous implicit surface, it can be seen that TSDF overestimates the distance field due to the limited viewpoints, and only agrees with the true Euclidean distance if the sensor ray is in the direction of surface normal. Fig. 7.10c) is the result of what the authors in [251] called GPIS-SDF, which is an online continuous and probabilistic method to recover the implicit surface and the distance field. It can be seen that, although the distance value within a small region around the surface is accurate, if the testing points are far away from the measurements, the uncertainty increases and the distance values are not properly estimated. As we can see, the region without measurements nearby is covered by the white color with brighter being more uncertain. Fig. 7.10d) shows the surface and distance field estimated by our method log-GPIS. Here, it is evident that the EDF is accurately predicted even when far from the surface measurement with the same linear growth as in the ground truth, and the surface is accurately reconstructed. This shows that log-GPIS is advantageous in both mapping and accurate distance estimation.

The latter advantage is particularly useful for planning, because it allows advanced biasing such as the medial axis [84]. Thus far, doing so required a computationally intensive fast marching techniques [80–82, 257], but log-GPIS handles EDF estimation elegantly and efficiently by a simple log-transformation. For this reason, log-GPIS can significantly improve planning.

7.3 Summary and Evaluation

In summary, the two case studies demonstrate that planning performance can be significantly improved by incorporating predictive priors into perception algorithms. In both case studies, a PDE constraint was imposed on GP regression, thereby improving the physical fidelity of the predicted environments. This allows the robot to make better prediction of the environment at a remote location using the sensor data at the current location.

In the OCEAN problem, we exploited the fact that ocean currents are planar incompressible. Planar incompressibility implies the existence of a streamfunction, which is related to the ocean currents via partial derivatives. We presented incompressible GP, which makes use of this relationship. Incompressible GP can extrapolate the ocean currents with a greater accuracy than standard GPs. The extrapolative power enabled drift-based current estimation and navigation to goals.

In the EDF mapping case, we used the fact that EDFs must satisfy the Eikonal equation. Whereas the Eikonal equation is nonlinear and difficult to incorporate, Varadhan's formula allows approximate enforcement of the Eikonal equation through linear PDEs. We presented log-GPIS which enforces Eikonal equation through Varadhan's formula. We demonstrated that using log-GPIS allows accurate reconstruction of EDF especially further away from measurement points.

Overall, in both cases, the improvement in planning performance was driven by the predictive capability of the GP model of the environment. In particular, incorporating the PDE model improved the accuracy of covariance function that models correlation between the value of the underlying field. In the case of incompressible GPs, we obtained a more accurate and specific model of the correlation between current vectors. In the case of log-GPIS,

the combination of the heat kernel and log-transform yielded a linear behaviour in the distance field as we move away from the obstacles. In abstract terms, these observations indicate that the underlying GP became a better *generative model* of the environment, because correlation between values affect the joint distribution. Based on this observation, we hypothesise that generative modelling is the ideal method for devising predictive priors for prospection.

Chapter 8

Conclusion

8.1 Summary of Thesis

This thesis proposed prospection as a mechanism for autonomous mobile robots in uncertain environments. Prospection refers to predicting possible future environmental states and evaluating them with respect to the task at hand. This new perspective offers a unifying view of autonomous decision-making that combines insights from active perception [14] and planning under uncertainty [153, 267]. Specifically, robotic prospection comprises two main elements: prospective perception with predictive priors, and prospective planning with uncertainty awareness.

We considered general and specialised variants of prospective planning. Chapter 3 presented the most general instance of prospective planning under environmental uncertainty, comprising a Bayesian robot operating in an unknown environment using its onboard sensor. The aim of the problem is to find a strategy that maximises the reward gathered based on previous sensor measurements.

Chapter 4 presented mutual information upper confidence bound (MI-UCB) as a solution to this general problem formulation. MI-UCB is defined as the weighted sum of information gain and prior expected reward. The main result is that we can approximately maximise the posterior reward after measurements without knowing their values ahead, by selecting actions with the highest MI-UCB value. Under mild assumptions on the problem

instance, the MI-UCB exhibits submodularity, which permits the use of algorithms for active perception and information gathering. We also showed that MI-UCB can be used to distribute exploration and exploitation in a *scout–task robot team*, consisting of scout robots that gather information and task robots that take reward. This allows simultaneous and synergistic exploration and exploitation, as opposed to a trade-off in the single-robot setting, as we demonstrated experimentally in a multi-drone surveillance application.

Chapter 5 demonstrated prospective perception and planning in tandem in the case of environmental process monitoring, where the goal is for a robot to reach the source of an environmental process based on sparse concentration measurements. Prospective perception was realised with a specialised Gaussian process (GP) regression scheme called advection-diffusion Gaussian process (ADGP) which incorporates the physical model of advection diffusion partial differential equation (PDE) into GPs. The combination of ADGP with Gaussian process upper confidence bound (GP-UCB) [181] strategy led to theoretical guarantees on convergence and intelligent behaviour such as prioritising cross-stream regions if the concentration measurement (i.e., the advection from down-stream regions) is low.

Chapter 6 presented prospective planning for complex semantic tasks beyond simple scalar fields. To reason over uncertainty, we developed random signal temporal logic (RSTL), which extends signal temporal logic (STL) from binary evaluation of task satisfaction to continuous *probability* of task satisfaction. This uncertain semantics allows diverse and flexible reasoning over environmental and dynamic uncertainties, as well as the effect of future measurements, which was shown to resolve the issue of over-conservatism.

Finally, in Ch. 7, we investigated what constitutes good predictive priors for prospective perception through two case studies, and concluded that generative modelling is likely the most suitable. We considered two problem instances, of oceanic and indoor navigation. For the former, we developed the incompressible GP using the fact that ocean currents are planar incompressible. This physical knowledge granted sufficient extrapolation for the robot to simultaneously estimate ocean currents from motion data and navigate based on the estimated current. For the indoor navigation problem, we developed log-Gaussian process implicit surface (GPIS), which exploits the fact that Euclidean distance fields

(EDFs) must satisfy the Eikonal equation. Similar to the case of oceanic navigation, the incorporation of Eikonal equation improved accuracy further away from measurements. These observations support the relevance of generative models, as the improvement in prediction accuracy resulted from accurate modelling of correlation through the use of physical models.

In the remainder of this final chapter, we review and summarise the significance of these results. We then recommend several avenues for future research, and discuss the implications of robotic prospection towards autonomy.

8.2 Main Contributions

Most importantly, this thesis introduced robotic prospection, a novel mechanism for autonomy that extends the typical perception-planning pipeline with foresight and proactivity. Robotic prospection comprises prediction of possible environmental states given sensor measurements, and evaluation of these states for action selection. It combines insights from multiple formulations for planning, including active perception [14], decision-making under uncertainty [267], and Bayesian optimisation [31]. We presented a general mathematical formulation of robotic prospection in the case when the main source of uncertainty is the surrounding environment, as well as special cases with favourable problem structures that can be exploited. The significance of this new perspective is that it explicitly accounts for how the robot's possible future understanding of the environment should affect the present decision.

Based on the general and specialised problem formulations, we presented a suite of algorithmic tools, assuming that the environment is static. We believe that, together with these tools, prospection has the potential to shift the paradigm of robotic system architecture, from the conventional reactive perception-planning pipeline to a proactive loop of queries between perception and planning algorithms. The rest of this section reviews and summarises the significance of these tools relative to the vision of robot autonomy.

8.2.1 MI-UCB for General Prospective Planning

We proposed MI-UCB as a strategy for solving the generalised prospective planning problem, based on prior work in [58, 65]. Most importantly, MI-UCB shows that combining mutual information (i.e., exploration) and reward (i.e., exploitation) serves as a proxy for optimising posterior reward in hindsight. MI-UCB can be computed efficiently in many problem instances, and essentially circumvents the computational requirement of enumerating possible environmental states or sensor measurements as did previous formulations. In fact, we showed that MI-UCB reduces partially observable Markov decision process (POMDP) and mixed observability Markov decision process (MOMDP) formulations to submodular maximisation. Although submodular maximisation is still NP-hard, as are POMDP and MOMDP formulations, many computationally efficient, polynomial-time approximate solvers are available. Therefore, it enables prospective planning in practical robotic systems.

We defined the concept of scout-task architecture, in which some robots gather information and others take reward, and showed how such a team can be coordinated through the use of MI-UCB. The scout-task architecture provides a unifying view of ideas from many previous implementations, such as in [49–51, 193]. Further, we demonstrated how scout-task teams can be viewed as simultaneously performing exploration and exploitation, by extension of the classical trade-off. Significantly, this provides the grounds for generalising scout-task robot teams to other problem instances where selected subset of robots are specialised for implicitly aiding through observation.

8.2.2 Physics-based Priors

Throughout this thesis, we proposed several novel GP priors that faithfully satisfy physical models, including incompressibility (based on [62]), advection-diffusion PDE (based on [64]), and the Eikonal equation (based on [56]). These models are highly useful as predictive priors for prospection, as they offer excellent extrapolation with minimal data, while their correctness is guaranteed by our understanding of physics. Even more importantly, an exciting aspect of these models is that they each serve as powerful tool for

Bayesian data assimilation in various branches of natural sciences. Therefore, we anticipate that these models could serve as an intermediate layer between the natural science and robotics communities that enables both high-quality data assimilation and development of prospection or information gathering algorithms. Having such an intermediate layer will be instrumental in widespread adoption of robotics in scientific missions.

8.2.3 An Uncertainty-Aware Temporal Logic for Semantic Task Planning

We proposed RSTL to model uncertainty in semantic task planning based on [59]. Reasoning over uncertainty allows integration with practical robotic systems, and also with prospective planning. Further, because probability is continuous, as opposed to binary Boolean evaluations, it allows efficient gradient-based synthesis. For these reasons, we believe that RSTL will be pivotal in the adoption of formal methods in practical robotic systems.

In addition to the inherent benefits of incorporating uncertainty, RSTL leads to a natural synthesis-as-inference perspective. The reformulation of STL synthesis as probabilistic inference in Sec. 6.4.1 has important implications for future work. Independently, it has the potential to accelerate the development of explainable artificial intelligence (AI) techniques through seamless integration between formal methods and machine learning techniques as did the optimal control-as-inference paradigm [248, 249]. Further, under the synthesis-as-inference paradigm, multi-robot planning is equivalent to decentralised data fusion, which is well studied. These examples illustrate the opportunities offered by RSTL for designing new algorithms for STL synthesis by adopting tools from probabilistic inference.

8.3 Future Work

Prospection is a novel perspective on robot autonomy that significantly improves upon the previous one of adaptivity. Many important avenues for future work arise from this new perspective. This section summarises these areas of future work, and their impact on advancing robot autonomy.

8.3.1 Prospective Perception with Deep Generative Models

This thesis developed a methodical recipe for robotic prospection comprising prospective perception and planning. In particular, the practitioner may realise prospective planning by using MI-UCB, which only involves definition of the reward, and evaluation of he Shannon information gain between the measurements and the environment. As was seen in the case studies, predictive priors based on generative models can be used to significantly improve the performance of planning.

An important recent development is the advance of deep generative models, such as generative adversarial networks, variational autoencoders, and sum-product networks (SPNs) (see [90, 152]), which offer powerful extrapolation given limited data. The combination of prospective planning with such powerful predictive priors has significant potential for enhancing robot autonomy. In fact, the utility of such deep generative models has been already demonstrated in simple navigation tasks [90]. Further, prospection can naturally handle more complicated problem instances such as the Canadian traveller's problem with correlated edge costs [185, 268]. The remaining challenge is the computation of information-theoretic quantities for such models. The result would be a robot that not only reacts to changes in the predicted environment, but also actively selects actions that offer useful knowledge.

8.3.2 Communication and Composition Planning in Multi-Robot Systems via Prospection

The scout-task coordination problem considered in Sec. 4.2 stimulates much imagination for future work. The most immediate avenue based on the results are some fundamental questions in multi-robot coordination. Given a problem instance, can we postulate an optimal composition of scout and task robots? Can the composition be adapted dynamically depending on the task at hand? These problems allows operational optimisation in many practical applications in areas such as agriculture, infrastructure monitoring, construction, marine robotics, and others where there is value in separating scout and task robots.

Further, we conjecture that it will offer interesting insight into information-theoretic characterisation of difficulty of a problem instance [181].

Another exciting aspect is that prospective planning in a scout-task team can be extended to implement planning-aware communication [269, 270]. An entire scout-task robot team can be viewed as a system on its own with sensors and effectors that can be controlled. Measurements in this system are realised through communication, in the sense that communication actually delivers the value of measurement to other agents. In fact, Shannon [219] initially developed information theory to quantify the utility of communication packets. The extension to evaluation of measurements is owing to an analogy to communication in that both are conveyors of information. By extension, the scout-task coordination problem of selecting task-relevant measurements is equivalent to the planning-aware communication problem of choosing task-relevant communication packets.

8.3.3 Semantic Task as Predictive Prior or Environmental Parameter

The perspective of viewing a task as a Bayesian conditional [248, 249] offers useful insight for prospection. In a cooperative setting, a multi-robot system may share the knowledge about what task is to be conducted prior to mission. Using this knowledge, each robot can augment or replace explicit communication for behaviour coordination with trajectory predictions derived from specifications [58, 164], through integration with estimation methods [271, 272].

Another important extension is when the task itself is unknown or unclear. This occurs when the instruction contains ambiguities that must be grounded given context [273]. Under the prospection formulation, we can pose the task specification as an environmental parameter that must be estimated given observations, and employ MI-UCB as we do for any other environmental uncertainty. Indeed, the questions remain open, as to how task can be inferred given observation, and how we can quantify the information gain of measurements in this setting.

8.3.4 Extensions to Dynamic Environments

For the sake of simplicity, the formulation in this thesis exclusively considered situations where the environment is static. An important direction for future work is to handle dynamic environments.

In the simplest case, the environment may be varying over time, independently of the robot's action. In principle, MI-UCB can solve this issue via over-parameterisation, by redefining a new environmental parameter as concatenation over time. Nonetheless, more efficient treatise of temporal variation would be necessary in practice. To this end, recent work on transfer operators [121] appears to be a promising direction for scalar field problems such as the one considered in Ch. 5. More ambitiously, the dynamics itself could also be treated as a part of the environmental parameters to be identified, similar to the premise of dual control [129, 130].

A less trivial case is if the changes in the environment are the result of robot's actions. In this case, one must consider what the robot is allowed to change in the environment, or, aptly, the *affordance* [274]. As an extension of the results presented herein, we conjecture that the solution involves steering the environmental state to be favourable not only in terms of possible reward, but also in terms of uncertainty reduction. Even so, practical identification of affordance on its own remains the subject of active research, particularly in manipulator motion planning (see [154, 275]).

8.4 Concluding Remarks

A new era of robotics is coming. In this new era, robots will need to proactively postulate and query its surroundings, rather than simply reacting to the changes. A higher degree of autonomy beyond reactivity will be increasingly important.

Robotic prospection presented in this thesis offers a unifying view of such foresight. In this new setting, robots predict future measurements and environmental states, and evaluate the prospects with respect to the task at hand. The result is proactive robots that can predict beyond their sensing horizon and deliberately interrogate their surroundings to extract

task-relevant knowledge. Such an ability will be crucial in extending robotic applications to handle the extent of environmental uncertainty required in modern applications.

This thesis presented a practical recipe for realising robotic prospection. We hope that the results of this thesis serve as useful building blocks in the road towards autonomous robots that 'just work'.

Appendix A

Implementation Details for Decentralised ISR Application

This appendix presents the design and implementation for demonstration of the decentralised intelligence, surveillance and reconnaissance (ISR) task described in Sec. 4.4.2. We present a practical system implementation, including decentralised inter-robot localisation, mapping, data fusion and coordination. The system is demonstrated in an efficient distributed simulation. We also describe an uninhabited aerial vehicle (UAV) platform for hardware experiments, and the preliminary progress toward hardware verification. The contents of this Appendix is based on our previous publication in [65].

A.1 Software Architecture

In this section, we present the sensors and software components of the framework, as summarised in Fig. A.1.

A.1.1 Target Detection and Decentralised Fusion

To detect the presence of a target in a sensor measurement, classifiers were trained to segment humanoids from LIDAR and RGBD point clouds. The RGBD classifier is based on

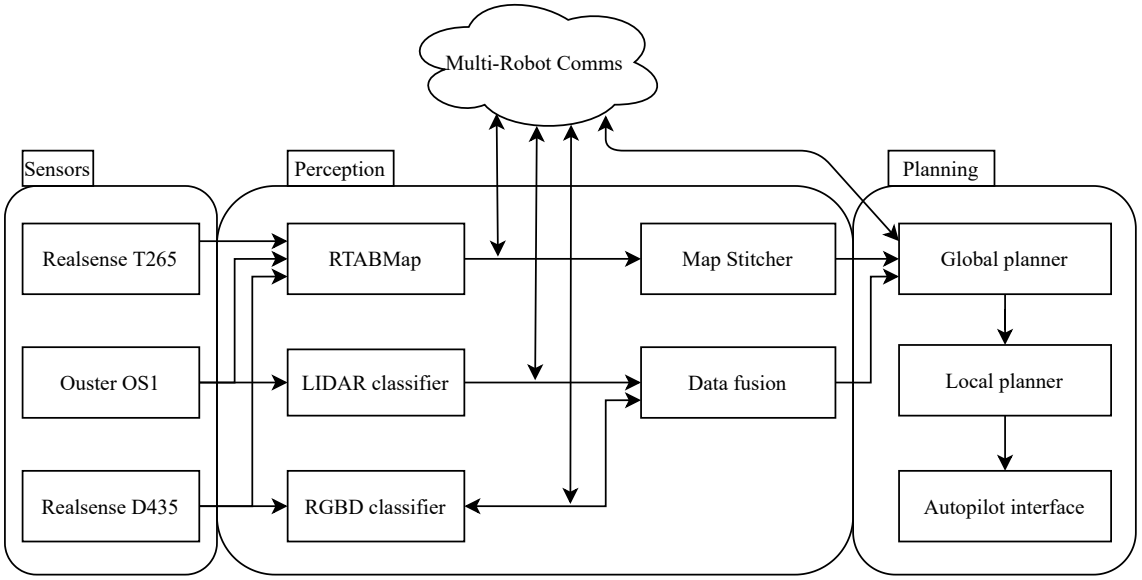


FIGURE A.1: Software system overview for a single LIDAR drone. RGBD drones do not have the LIDAR sensor onboard.

the `jetson-inference`[1] library, which was modified to correlated RGB and depth images from the RealSense D435 camera. We use the LIDAR classifier proposed in [276], which consists of a cluster extraction algorithm, an support vector machine (SVM) classifier, and a nearest-neighbour-based Kalman filter for data association and tracking. The LIDAR classifier required additional training data, which was sourced from some field trials.

The detection results from the RGBD and LIDAR classifiers are broadcast to the entire team. Each drone runs an independent occupancy grid filter described in [58] to fuse the detections generated by itself and the ones communicated by others.

A.1.2 Decentralised mapping and localisation

For a drone to perform its mission, it must be able to map its environment and localise itself in the map. Each drone runs real-time appearance-based mapping (RTABmap) [230] onboard using sensor measurements from the Realsense T265, D435, and if available, the Ouster OS1 and global positioning system (GPS) sensor. RTABmap enables each drone to build a map of its surroundings, assign a geographic location to detected targets, and to perform reliable path-planning and navigation. The C++ RTABmap package

[1] https://github.com/dusty-nv/jetson-inference

interfaces with Robot Operating System (ROS) via the `rtabmap_ros` node, which allows other processes (e.g., the global planner) access to the map and current drone pose estimate generated by RTABmap.

In addition to decentralised operation, another requirement of the system is that it can operate in GPS-denied scenarios. Since the system consists of multiple drones, a common geographic reference frame must be established online between drones without the use of GPS services. An online "map stitching" ROS node was developed to "stitch" maps of different drones together, thereby joining the maps between drones, and thus allowing a common reference frame. In particular, the relative 3D rotation and translation between two drones' individual simultaneous localisation and mapping (SLAM) maps is estimated periodically, facilitated by sharing of downsampled sensor data. This is the main idea behind the `map_stitcher` node, which simply performs loop closures *between* drones, as opposed to between poses of a single drone. Although this functionality exists in RTABmap during offline postprocessing, the `map_stitcher` node was developed for this project for online map stitching, with modifications for fast execution.

We present a high-level overview of the map stitching process (see Algorithm 3). Let Z_i be the set of all of its sensor measurements of the environment. The elements $z \in Z_i$ are "full" sensor measurements, with geometric information about the environment, in contrast to the "measurements" Y, which refer to target detections by the classifier. Then, given an initial pose, the internal SLAM algorithm of drone i will produce a pose graph G_i, which can be solved to estimate its trajectory T_i, i.e., the set of all of its poses. Drone i the receives compressed sensor measurements from other drones, and uses them to perform a map stitching.

Suppose drone i receives an incoming sensor measurement z_j at pose p_j from drone j and its pose graph G_j. The map stitching algorithm running on drone i compares z_j to all of its own measurements Z_i. The function SimilarityScore is a fast way to estimate the "similarity" between two RGB images; if the images are similar enough, relative pose estimation is attempted. If the relative pose between the two estimates is successful, then a the relative pose (along with covariance information) is added between the two pose graphs G_i and G_j. If multiple measurements have high similarity score and successful

Algorithm 3 Map stitching algorithm

procedure StitchMap($G_i, G_j, z_j(t)$)
  for $z_k \in Z_i$ **do**
    $s \leftarrow$ SimilarityScore(z_j, z_k)
    if $s > s^\star$ **then**
      $\hat{T} \leftarrow$EstimateRelativePose(z_j, z_k)
      if $\hat{T} \neq \emptyset$ **then**
        AddLoopClosure($G_i, G_j, p_k, p_j, \hat{T}$)
  $\mathcal{G} \leftarrow$ SolvePoseGraph($G_i \bigcup G_j$)
  return $\mathcal{G}$

relative pose estimates, multiple loop closures can be added between G_i and G_j from z_k. When all the measurements in Z_i have been checked against the incoming z_j, the joint pose graph is solved. Hence the maps of drones i and j are joined together in $\mathcal{G}$. This allows a shared geographic reference between the drones i and j which can operate in GPS-denied scenarios, facilitating the sharing of target beliefs among drones.

A.1.3 Local planner

We use the local obstacle avoidance planner proposed in [277]. Its role is to take the planned paths from the global planner and generate a local collision-free trajectory near the drone's current location for the low-level PX4Cube to follow.

The local planner uses the last few frames from the D435 to map the obstacles. This ensures two layers of redundancy against unmapped or moving obstacles. The global plan is collision-free with respect to the large-scale map built by the decentralised mapping module, and the local plan is collision-free with respect to the local RGBD measurements, which does not depend on the decentralised mapping module.

A.2 Communication Architecture

A.2.1 ROS and MOOS networks

ROS [229] is a software library that provides a common interface for different C++ and Python programs, called *nodes*, to communicate with one another. ROS is used onboard

each drone to manage communications within the drone's NVIDIA Jetson TX2 computer. However, ROS cannot be used in a decentralised way between multiple computers, as it requires a unique ROS "master" to be run on the network. The failure of this ROS "master" would bring down the entire team, which is undesirable. Hence, a different middleware solution called Mission-Oriented Operating Suite (MOOS) [278] is used for inter-vehicle communications. MOOS has the advantage that it can be used in a decentralised way, without a single "master" computer. With most of the algorithms being written in ROS, including the local and global planners, the classifiers, the SLAM algorithm, and the sensor drivers, bridge between ROS and MOOS was used, called `rospymoos`[2]. We developed a modified version of `rospymoos` to handle custom ROS messages required in this project. Any ROS messages containing information necessary to be sent offboard the drone are converted into a MOOS message, and sent to other drones over WiFi via MOOS. Similarly, messages from other drones are received by MOOS and converted into ROS-compatible messages via `rospymoos`.

A.2.2 Low-level communications

Within the ROS network running on the NVIDIA Jetson TX2 onboard each drone, the local planner calculates a nearby "local waypoint" for use as a setpoint for the PX4Cube. This local waypoint is communicated to the PX4 via the `mavros`[3] ROS node, which is translated into a MAVLINK message using `mavlink_router`[4], and then finally transmitted from the Jetson TX2 to the PX4 over a universal asynchronous receive-transmit (UART) connection.

A.2.3 Ground station

To facilitate mission monitoring by humans, we modified the popular open-source QGround-Control (QGC) software, which is commonly used to monitor and control hobby drones. The concept was to have one computer with QGC per drone to monitor its status, as

[2]https://github.com/SyllogismRXS/moos-ros-bridge
[3]https://github.com/mavlink/mavros
[4]https://github.com/mavlink-router/mavlink-router

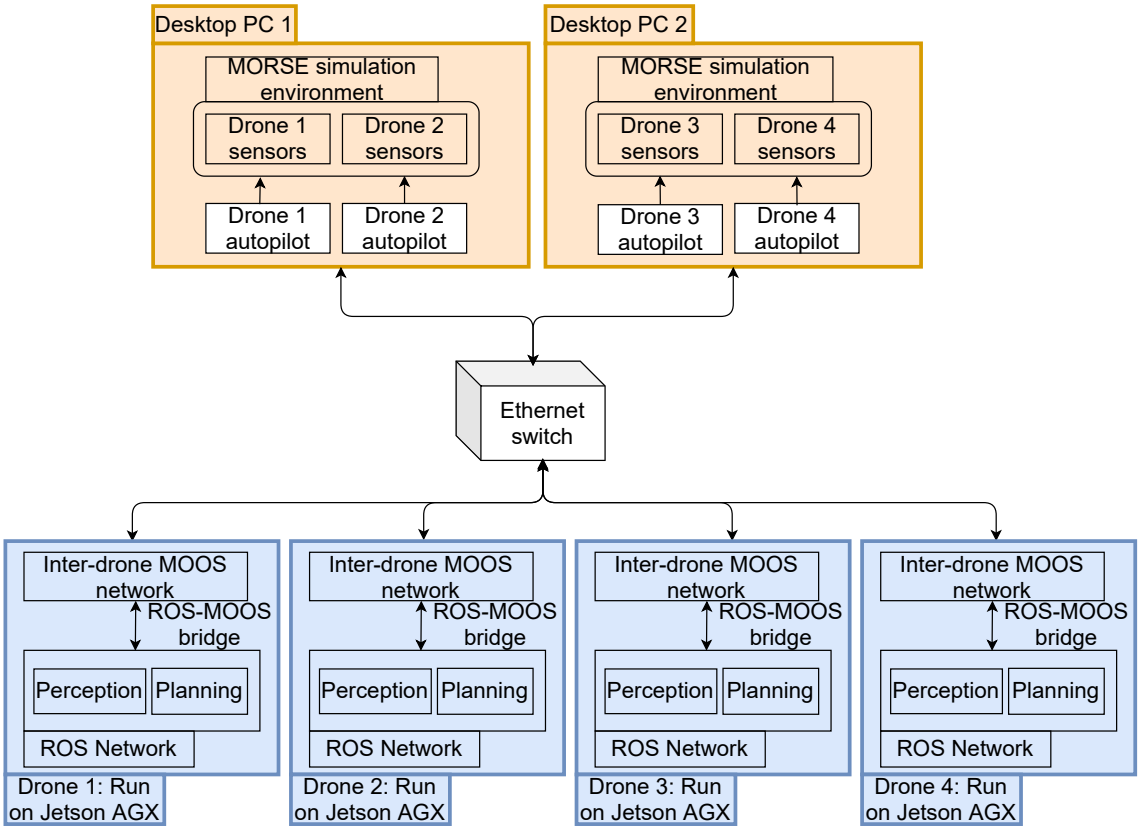


FIGURE A.2: Simulation architecture diagram.

well as a "team" QGC that displays higher-level information about all the drones in the team. The major change to QGC was the added capability to access information from each drone's onboard ROS network, which is absent from standard QGC. Important algorithmic diagnostic information, such as metrics about the "health" of mapping and localisation algorithm, local and global planners, detected targets, and more are now displayed on QGC.

Since a ROS subscriber cannot subscribe to topics managed by two different ROS "masters" (i.e., on two different drones) at once, a ROS node was implemented to relay ROS messages to QGC upon receiving an HTTP GET request from QGC. This allows each QGC to access ROS information from multiple drones, and for each drone to serve multiple QGC instances on the various ground stations.

A.3 Distributed Simulation Architecture

A distributed simulation framework was set up to perform hardware in the loop (HIL) testing after field trials became infeasible. The perception, mapping, and planning software described in Section A.1 was used on four NVIDIA Jetson AGX Xaviers, a more powerful but pin-compatible version of the Jetson TX2 onboard the drones. The Jetson AGX Xaviers are connected via an ethernet switch to two desktop computers with Core i7 processors and NVIDIA RTX2060 graphics cards. These desktops simulated the motion of the four drones using the open-source PX4 simulator [232] in simulated environments using a custom version of the Modular Open Robotics Simulation Engine (MORSE) [233]. The desktops were also responsible for the computationally expensive task of ray-tracing to simulate RGBD and LIDAR sensor measurements for each of the four drones, facilitated through MORSE and Blender. Fig. A.2 shows a graphical representation of the simulation architecture.

A.4 Hardware Experiments

The core unit of the multi-robot system is an individual drone (Fig. A.3). The team of drones consists of two types: drones that are task drones, and drones that are both scout and task drones. In this team, there are no drones that are "scout-only" drones. All drones share a common airframe and propulsion system, and differ only in their sensor payloads. The specification is summarised in Table A.1. In total, the team consists of 4 drones. Fig. A.4 shows an image taken during a field trial.

A.4.1 Sensors

The sensor payload for the RGBD drones consists of an Intel RealSense D435 RGBD camera mounted facing forward for target recognition, and a nadir-pointing Intel RealSense T265 RGBD camera for localisation and mapping. The T265 camera also includes an onboard SLAM implementation, and can used to produce a pose estimate of a drone

Table A.1: Specification of the drone platforms

Airframe	S500
Size	500mm wheelbase
Maximum take-off weight	1.8 kg
Motors	4x NTM PROPDRIVE v2 3536 910kV
Propellers	10-inch x 4.7
ESC	4x Flycolor X-cross BL-32 50A
PDB	Matek FCHUB-125 PDB
Battery	Zippy 4S LiPo, 40C, 6200mAh, 0.59 kg
FCS	Pixhawk Cube
GPS	Here2 GPS

relative to its own initial pose. The T265 camera SLAM solution is considered a source of odometry in a higher-level SLAM implementation (see Section A.1.2).

The D435 camera also contributes towards mapping and localisation; the RGBD images it takes are fed to the aforementioned higher-level SLAM implementation. In addition, the the RGBD images are also used as an input to a deep image classifier to recognise targets. Target detections from the RGB classifier are considered to be true detections. In ideal indoor conditions, the D435 is rated to give accurate depth readings at ranges up to 10m, though in our outdoor experiments, this was never realised.

The LIDAR drones also carry T265 and D435, but is augmented by an Ouster OS1 3D LIDAR. The OS1 LIDAR provides an omnidirectional field of view with an order of magnitude longer sensor range than the RealSense D435 (about 100m), allowing the robots carrying the OS1 to sense large areas quickly. However, a LIDAR scan was considered to have insufficient resolution to reliably decide if a segment of the scan is a target or not; especially at long ranges, the LIDAR point cloud can be very sparse. The LIDAR sensor data was used as an input to the higher-level SLAM algorithm, as well as to detect *potential* targets using the LIDAR classifier. Targets identified using the LIDAR classifier are considered potential targets, and require confirmation using the shorter-range D435.

A.4.2 Electronics, flight control hardware

Low-level control (e.g., attitude control) of each drone is handled by a PX4Cube flight control unit. Additionally, an NVIDIA Jetson TX2 is onboard each drone to perform

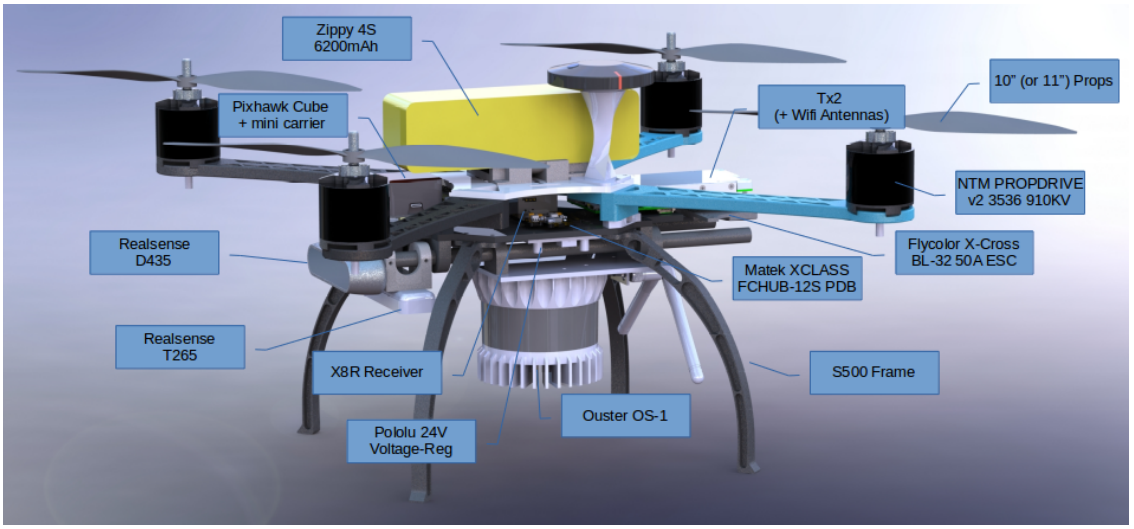


FIGURE A.3: A 3D render of the UAV platform with LIDAR

FIGURE A.4: Obstacle avoidance test at the RAAF Williams in Victoria, Australia.

higher-level computing tasks, including target recognition, mapping, localisation, planning, and also a "local planner" which gives commands to the PX4 flight computer.

A.4.3 Lessons learned from field trials

Because of COVID-19 restrictions, full-system field tests of the hardware were not performed. However, the field tests that were performed were invaluable in improving and debugging the system. We describe some lessons learned here.

During field tests, it was discovered that the RealSense D435 camera interfered with the GPS sensor module. When the D435 plugged into the Jetson TX2 computer onboard the drone, GPS connection would be intermittently lost, or the wrong number of satellites would be reported. Even when the connection seemed stable, the measurements from the GPS sensor module accumulated unacceptably large drift during test flights, with over 10m of drift in under 5 minutes. Since the interference was believed to be electromagnetic, this was remedied by changing the distance between the GPS module and the RealSense D435. The implemented solution was to be to 3D-print a taller "mast" for the GPS module to be raised about 10cm above the body of the drone, which was successful in recovering good GPS sensor performance. The shorter mast is shown in Fig. A.3.

Although the T265 camera performs SLAM as opposed to simply visual odometry (i.e., it attempts to detect loop closures), in practice this added functionality caused undesirable behavior. When flying the drones at low altitude over grass for extended periods of time, impossibly large jumps were detected from the T265 odometry output. This was caused by different patches of grass being identified as the same patch, resulting in incorrect loop closures and severe mapping and localisation failure. This issue affected only the task-only drones, which did not have LIDAR information. The problem was ameliorated by detecting the onset of this localisation failure in order to land the drone safely.

Appendix B

Implementation Details for OCEAN

In this appendix, we describe the details of implementation needed to perform our field trials. The overview of the closed-loop implementation is shown in Fig. B.1. For a proof-of-concept product, we decided to use MATLAB as a development platform for its builtin graphing capabilities and its quick development cycles.

When the glider is on a surface, it transmits the most recent glider data (.sbd) to Dock server and waits for a new plan. Our autonomous system then updates the ocean current estimates using drift between the dead reckoned and true locations. If the glider is close

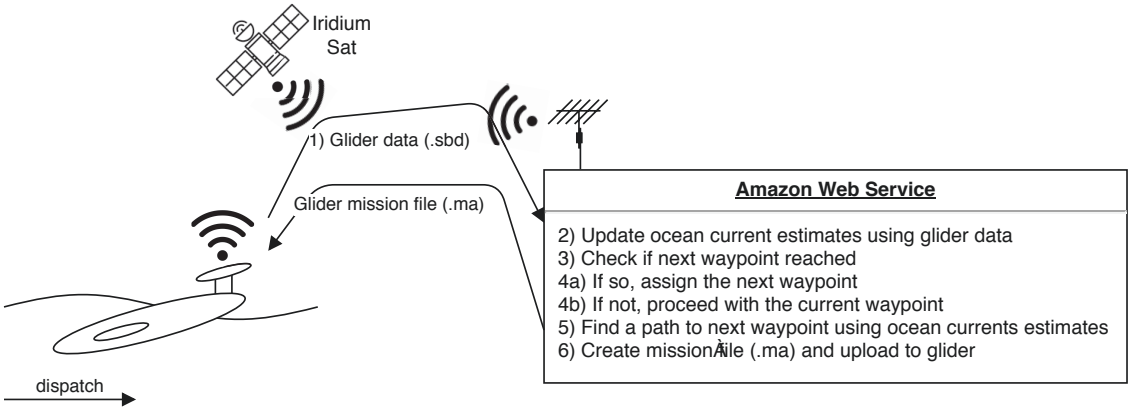


FIGURE B.1: Autonomous closed-loop estimation and re-planning workflow with Amazon Web Service

enough to target waypoint, then the system chooses the next waypoint. Given the target waypoint and the updated ocean current estimates, the system finds a glider control that drives the glider to the target waypoint under the influence of estimated ocean currents. Since the glider assumes idle ocean condition, the waypoint that the glider is assigned to follow is different to where we want it to go. We call such waypoint *virtual waypoint*. The virtual waypoint is computed given ocean currents estimation and a new mission file containing the virtual waypoint is created. The file is uploaded to the glider and the glider resumes its mission. The surface time required to process all mentioned are below a couple of minutes; the glider would only be drifted a minimal distance on the surface.

The modularity of this project allowed us to test each component independently before aggregating them to demonstrate how the objective of this project is satisfied. Fig. B.2 gives an overview of the structure of our software. For the class responsible for integrating the modules (`StatelessBOT` and `StreamBOT`), we maintain our policy of statelessness so that we are able to save working data and reload it after one surface iteration. This makes the class act as if it were a collection of functions as opposed to an object with state. The classes `StatelessBOT` and `StreamBOT` will be briefly covered in the following sections.

B.1 `StatelessBOT`

This class acts as a minimum working product to fulfil the requirements of this project. The second half of the name refers to the "Blue Ocean Top" level of the solution. `StatelessBOT` is responsible for interfacing between the components we have separately produced.

A typical usage of this class is outlined in Alg. 4. The data from the Slocum glider is parsed and filtered into data that is used in Line 1. Line 2 constructs `StatelessBOT` with the settings in the task config file. Line 3 uses the expected surfacing position of the glider (from dead reckoning) and the actual surface position simultaneously reconstruct the path and also estimate the current flow along the path. Line 4 uses an optimisation algorithm called the simplex search method (MATLAB's `fminsearch`) to find the best virtual waypoint to reach the desired position. A waypoint is considered "good", if the corresponding constant velocity vector it corresponds to produces the trajectory with a

Algorithm 4 Pseudocode for one surface iteration

Input: config, data_sbd, nextPos, cache
Output: nextPos, cache

```
1: gliderData = parse_sbd(data_sbd);
2: bot = StatelessBOT(config);
3: [cache.oceanCurrent,∼,∼,currPos] = …
        bot.update_oc(gliderData,cache.oceanCurrent,config.geoProjection);
4: [success,nextVirtPos] = bot.plan(cache.oceanCurrent,currPos,nextPos);
5: if success then
6:     nextPos = nextVirtPos;
7: return [nextPos,cache];
```

point that is closest to the desired position. Finally, Line 5-7 conditional block ensures that if we are unable to find a suitable waypoint, we use the desired position directly as a waypoint for robustness.

B.2 `StreamBOT`

This class improves upon `StatelessBOT`'s planning which can be prone to falling into local minima as there is no guarantee that the corresponding trajectory is optimal or complete in any sense. The `plan` method of `StatelessBOT` is overridden with the use of our streamline-based PRM* planner. Through inheritance, "`StatelessBOT`" can be replaced by "`StreamBOT`" in Line 2 of Alg. 4.

As a sampling-based algorithm, we are able to restrict the state space (2D position). We choose to sample within the ellipse with the start and goal points as focal points, and eccentricity customisable in the task config file. This means that we are able to dynamically scale the resolution of the path, depending on where the start and end goals are. Because the streamline-based method appears to be time-optimal, this approach should be complete and optimal.

It is important to note here, that without applying the idea of Sparse Gaussian Processes to the current estimation process, repeated queries of current estimation becomes more computationally intensive over mission time. To alleviate this effect, we equally sample the current field and use that grid to interpolate the current vectors used in the planner.

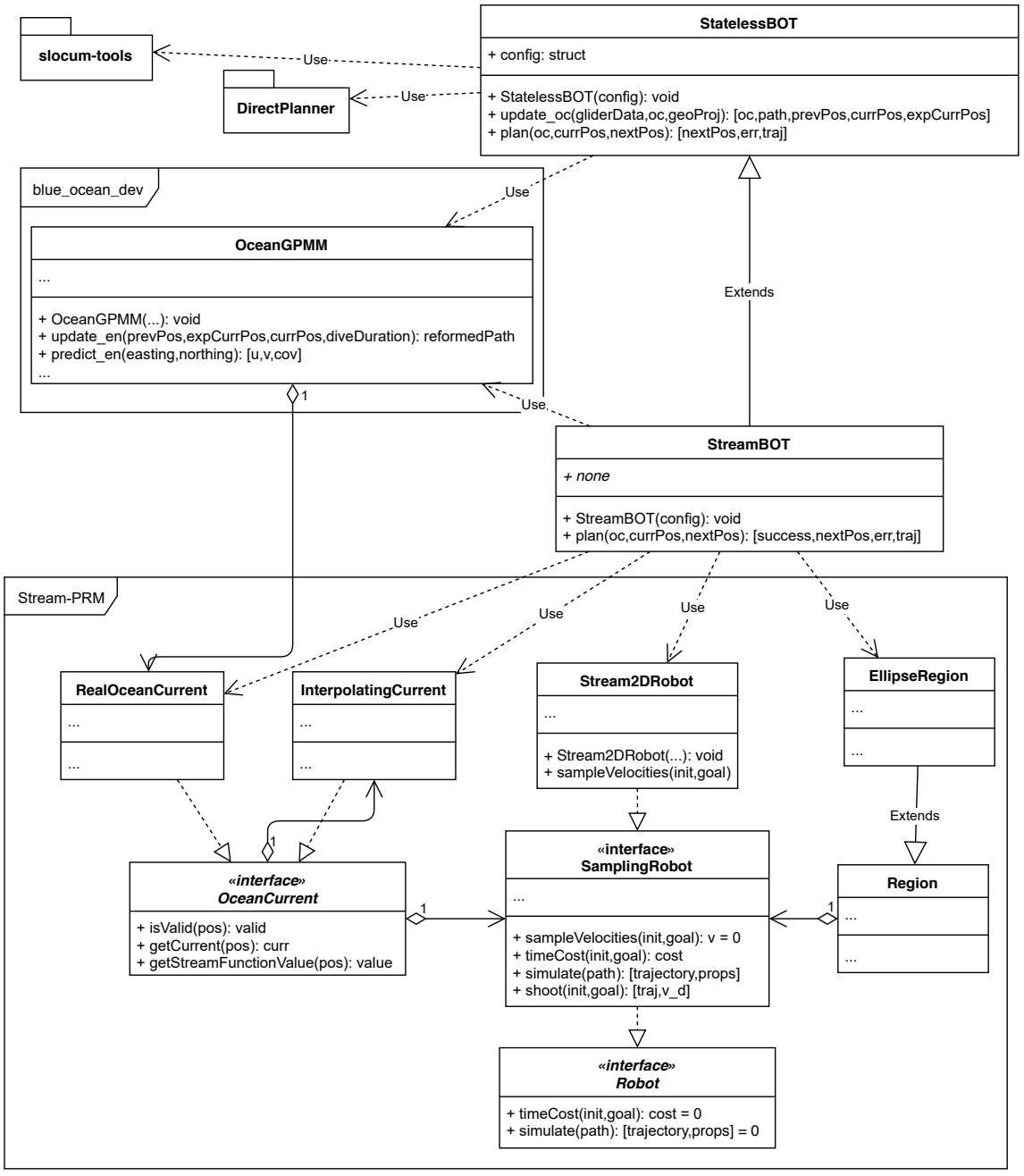


FIGURE B.2: Simplified class diagram showing the relationship between different modules to achieve glider navigation

The trade of accuracy for computation time appears to make little impact to the overall outcome of the planner, especially for higher interpolating resolutions.

Appendix C

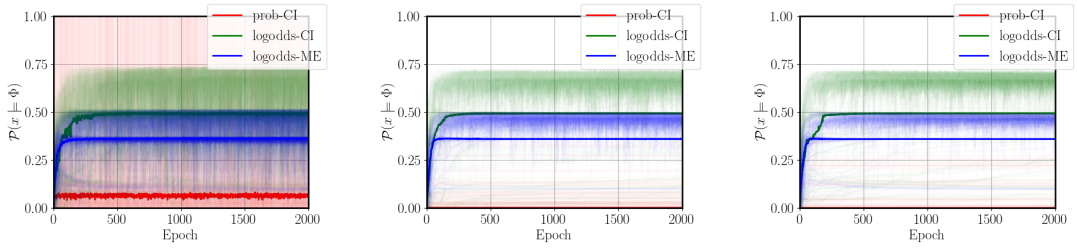
Empirical Analysis of RSTL

This appendix empirically analyses the performance characteristics of the gradient-based synthesis algorithm for random signal temporal logic (RSTL) described in Ch. 6. We first examine the convergence characteristics of the gradient ascent (6.31) solution, and demonstrate the computational benefits of graphics processing unit (GPU) acceleration. We then show that the framework is GPU-compatible and that it finds a better solution by concurrently running many initial conditions.

C.1 Convergence

Gradient-based methods cannot guarantee globally optimal solutions unless the objective is convex. We analyse the convergence characteristics of the proposed computation rules in the target search scenario (Fig. 6.2). We randomly generated 100 initial conditions from the control prior $\mathcal{P}(\mathbf{U})$, and ran the gradient ascent step for 2000 iterations, with $N_s = 1, 50, 100$ number of trajectory samples.

Figure C.1 shows the probability of satisfaction over gradient ascent steps for naive conditionally independent (CI) ((6.11), red), log-odds CI ((6.17), green) and log-odds mutually exclusive (ME) ((6.19), blue) methods with varying number of trajectory samples N_s. It can be seen that while log-odds CI and ME methods find global and local optima, the naive CI method does not find any. The naive CI method's failure is attributed to the



(A) $N_s = 1$ (B) $N_s = 50$ (C) $N_s = 100$

FIGURE C.1: Comparison of convergence with varying number of trajectory samples. Solid lines are medians.

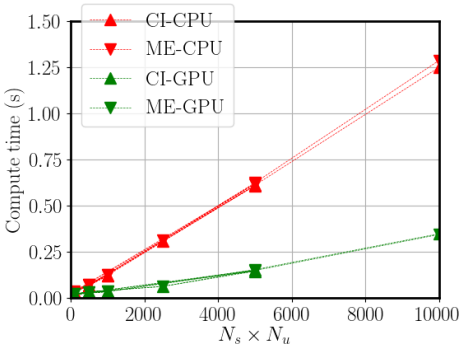


FIGURE C.2: Comparison of average computation time per gradient ascent step between combinations of CPU (red) and GPU (green) with CI (upward triangle and ME (downward triangle). GPU shows 4-fold improvement in scalability. Variance was in the order of 10^{-4} for all configurations.

computation rules (6.11) being bounded to $[0, 1]$ range, which causes numerical errors to build up. Note that log-odds ME reports lower probability of satisfaction due to its underestimation property, but we found that the resulting trajectories were still similar.

With increasing N_s, the variance in probability of satisfaction is decreased. In practice, this means the generated plan will more reliably account for state uncertainty, which is useful for, e.g., the collision avoidance scenario in Fig. 6.3.

C.2 Computation Time

The results in Sec. C.1 illustrate that it is important to use multiple initial conditions and more trajectory samples to ensure reliable operation. However, this would inevitably increase the computation time as well.

GPU acceleration is a prominent means to circumvent the issue of computation time, but not all algorithms benefit from GPU acceleration. To determine if our proposed methods benefit from GPU acceleration, we compare the computation time per gradient ascent step of our Tensorflow [245] implementation between GPU and CPU. We used all combinations between $N_s = 1, 10, 50, 100$ and $N_u = 1, 10, 50, 100$, and computed the mean over 1000 gradient ascent steps. We used a desktop with CPU (Intel i5-9500) and a GPU (NVIDIA RTX2060) to conduct the experiment.

Figure C.2 shows the computation time with varying number of initial conditions N_u and the number of state samples N_s. We found that the total number of samples $N_u \times N_s$ explains all changes. The result shows that the computation time with GPU is significantly lower than that with CPU, and that using a GPU leads to 4-fold improvement in scalability. This demonstrates that the proposed gradient ascent method benefits from GPU acceleration.

Bibliography

[1] J. E. Colgate, W. Wannasuphoprasit, and M. A. Peshkin, "Cobots: robots for collaboration with human operators," in *Proc. of ASME International Mechanical Engineering Congress and Exposition*, 1996, pp. 433–439.

[2] S. Thrun, "Toward robotic cars," *Communications of the ACM*, vol. 53, no. 4, p. 99–106, 2010.

[3] D. Gilbert and T. Wilson, "Prospection: Experiencing the future," *Science*, vol. 351, pp. 1351–1354, 2007.

[4] G. D'Urso, S. L. Smith, R. Mettu, T. Oksanen, and R. Fitch, "Multi-vehicle refill scheduling with queueing," *Comput. and Electron. in Agriculture*, vol. 144, pp. 44 – 57, 2018.

[5] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. V. Brussel, "Reconfigurable manufacturing systems," *Annals of the CIRP*, vol. 48, no. 2, p. Keynote Paper, 1999.

[6] K. L. Smith, A. D. Sherman, P. R. McGill, R. G. Henthorn, J. Ferreira, T. P. Connolly, and C. L. Huffard, "Abyssal benthic rover, an autonomous vehicle for long-term monitoring of deep-ocean processes," *Science Robotics*, vol. 6, no. 60, p. eabl4925, 2021.

[7] KUKA Roboter GmbH, "Factory automation with industrial robots for palletizing food products like bread and toast at a bakery in Germany," Online, public domain. Accessed 11 Dec 2021 at https://en.wikipedia.org/wiki/Industrial_robot#/media/File:Factory_Automation_Robotics_Palettizing_Bread.jpg, 2005.

[8] University of Technology Sydney, "Cobots the focus of new Advanced Manufacturing Centre," *Online Article*, 2020. [Online]. Available: https://www.uts.edu.au/about/faculty-engineering-and-information-technology/news/cobots-focus-new-advanced-manufacturing-centre

[9] K. A. Salom, "The modern flight control unit of an Airbus A340," Online, CC BY 2.0. Accessed 11 Dec 2021 at https://en.wikipedia.org/wiki/Autopilot#/media/File:A340_FCU.jpg, 2013.

[10] Y. Ahn and J. Waydo, "From post-it note to prototype: The journey of our Firefly," *Online Article*, 2017. [Online]. Available: https://blog.waymo.com/2019/08/from-post-it-note-to-prototype-journey.html

[11] L. Lopez, "The robots are killing Tesla," *The Business Insider*, 2018. [Online]. Available: https://www.businessinsider.com/tesla-robots-are-killing-it-2018-3

[12] K. J. Astrom, *Introduction to Stochastic Control Theory*, 1st ed. Elsevier, 1970.

[13] J. Hespanha, *Linear systems theory*, 2nd ed. Princeton University Press, 2009.

[14] R. Bajcsy, "Active perception," *Proc. of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[15] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Auton. Rob.*, vol. 42, no. 2, pp. 177–196, Feb 2018.

[16] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot slam," in *Proc. of IEEE ICRA*, May 2015, pp. 4775–4782.

[17] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The Int. J. of Rob. Res.*, vol. 30, no. 11, pp. 1343–1377, 2011.

[18] F. Sukkar, G. Best, C. Yoo, and R. Fitch, "Multi-robot region-of-interest reconstruction with dec-mcts," in *Proc. of IEEE ICRA*, 2019, pp. 9101–9107.

[19] D. Silver and J. Vennes, "Monte Carlo planning in large POMDPs," in *Proc. of NIPS*, 2012.

[20] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proc. of Robotics Science and Systems*, 2008.

[21] A. akbar Agha-mohammadi, S. Chakravorty, and N. Amato, "Firm: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements," *Int. J. of Rob. Res.*, vol. 33, pp. 268–304, 2014.

[22] R. Platt, R. Tedrake, L. Kaelbling, and T. Lorenzo-Perez, "Belief space planning assuming maximum likelihood observation," in *Proc. of Robotics: Science and Systems*, 2006.

[23] J. B. Watson, "Psychology as the behaviorist views it," *Psychological Review*, vol. 20, no. 2, p. 158, 1913.

[24] I. P. Pavlov, *Conditioned reflexes: An investigation of the physiological activity of the cerebral cortex*. Oxford University Press, 1903.

[25] K. S. Lashley, *Brain mechanisms and intelligence: A quantitative study of injuries to the brain*. University of Chicago Press, 1929.

[26] M. E. Seligman, P. Railton, R. F. Baumeister, and C. Sripada, "Navigating into the future or driven by the past," *Perspectives on psychological science*, vol. 8, no. 2, pp. 119–141, 2013.

[27] G. Pezzulo and F. Rigoli, "The value of foresight: how prospection affects decision-making," *Front. in Neuroscience*, vol. 5, no. 79, 2011.

[28] R. Wilson, A. Geana, J. White, E. Ludvig, and J. Cohen, "Humans use directed and random exploration to solve the explore-exploit dilemma." *J. of Experimental Psychology*, vol. 143, 2014.

[29] R. Carnap, *Logical Foundations of Probability*. The University of Chicago Press, 1950.

[30] B. M. Christian, L. K. Miles, F. H. K. Fung, S. Best, and C. N. Macrae, "The shape of things to come: Exploring goal-directed prospection," *Consciousness and Cognition*, vol. 22, pp. 471–478, 2013.

[31] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proc. of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[32] D. L. Rudnick, R. E. Davis, C. C. Eriksen, D. M. Fratantoni, and M. J. Perry, "Underwater gliders for ocean research," *Marine Technology Society J*, vol. 38, no. 2, pp. 73–84, 2004.

[33] L. M. Russell-Cargill, B. S. Craddock, R. B. Dinsdale, J. G. Doran, B. N. Hunt, and B. Hollings, "Using autonomous underwater gliders for geochemical exploration surveys," *The APPEA J*, vol. 58, pp. 367–380, 2018.

[34] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard, "Imaging sonar-aided navigation for autonomous underwater harbor surveillance," in *Proc. of IEEE/RSJ IROS*, 2010, pp. 4396–4403.

[35] R. P. Stokey, A. Roup, C. von Alt, B. Allen, N. Forrester, T. Austin, R. Goldsborough, M. Purcell, F. Jaffre, G. Packard, and A. Kukulya, "Development of the REMUS 600 autonomous underwater vehicle," in *Proc. of MTS/IEEE OCEANS*, 2005, pp. 1301–1304.

[36] D. C. Webb, P. J. Simonetti, and C. P. Jones, "SLOCUM: An underwater glider propelled by environmental energy," *IEEE J. Ocean. Eng.*, vol. 26, no. 4, pp. 447–452, 2001.

[37] Argo, "Argo float data and metadata from Global Data Assembly Centre (Argo GDAC)," 2000.

[38] G. D'Urso, J. J. H. Lee, O. Pizarro, C. Yoo, and R. Fitch, "Hierarchical MCTS for scalable multi-vessel multi-float systems," in *Proc. of IEEE ICRA*, 2021, pp. 8664–8670.

[39] G. D'Urso, J. J. H. Lee, K. M. B. Lee, J. Shields, B. Leighton, O. Pizarro, C. Yoo, and R. Fitch, "Field trial on ocean estimation for multi-vessel multi-float-based active perception," in *ICRA2021 1st Advanced Marine Robotics TC Workshop: Active Perception*, 2021.

[40] A. Gunatilaka, B. Ristic, A. Skvortsov, and M. Morelande, "Parameter estimation of a continuous chemical plume source," in *Proc. of IEEE ICIF*. IEEE, 2008, pp. 1–8.

[41] L. Paull, M. Seto, J. J. Leonard, and H. Li, "Probabilistic cooperative mobile robot area coverage and its application to autonomous seabed mapping," *Int. J. of Rob. Res.*, vol. 37, no. 1, pp. 21–45, 2018.

[42] J. J. H. Lee, C. Yoo, R. Hall, S. Anstee, and R. Fitch, "Energy-optimal kinodynamic planning for underwater gliders in flow fields," *Proc. of ACRA*, pp. 42–51, 2017.

[43] D. Kularatne, S. Bhattacharya, and M. A. Hsieh, "Going with the flow: A graph based approach to optimal path planning in general flows," *Auton. Rob.*, vol. 42, no. 7, pp. 1369–1387, 2018.

[44] G. A. Hollinger and T. Somers, "Learning uncertainty in ocean current predictions for safe and reliable navigation of underwater vehicles," *J of Field Robotics*, vol. 33, no. 1, pp. 47–66, 2015.

[45] C. Yoo, R. Fitch, and S. Sukkarieh, "Online task planning and control for fuel-constrained aerial robots in wind fields," *Int. J. of Robot. Res.*, vol. 35, no. 5, pp. 438–453, 2016.

[46] P. R. Oke, A. Schiller, D. A. Griffin, and G. B. Brassington, "Ensemble data assimilation for an eddy-resolving ocean model of the Australian region," *Quarterly J of the Royal Meteorological Society*, vol. 131, no. 613, pp. 3301–3311, 2005.

[47] P. R. Oke, D. A. Griffin, A. Schiller, R. J. Matear, R. Fiedler, J. Mansbridge, A. Lenton, M. Cahill, M. A. Chamberlain, and K. Ridgway, "Evaluation of a near-global eddy-resolving ocean model," *Geoscientific Model Development*, vol. 6, no. 3, p. 591, 2013.

[48] A. F. Shchepetkin and J. C. McWilliams, "The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model," *Ocean Modelling*, vol. 9, no. 4, pp. 347–404, 2005.

[49] T. Sasaki, K. Otsu, R. Thakker, S. Haesaert, and A. Agha-mohammadi, "Where to map? Iterative rover-copter path planning for Mars exploration," *Robot. and Automat. Lett.*, vol. 5, no. 2, pp. 2123–2130, 2020.

[50] K. Ebadi and A. Agha-Mohammadi, "Rover localization in Mars helicopter aerial maps: Experimental results in a Mars-analogue environment," in *Proc. of Int. Symp. on Exp. Robot. (ISER)*, 2018.

[51] S. Bharadwaj, M. Ahmadi, T. Tanaka, and U. Topcu, "Transfer entropy in MDPs with temporal logic specifications," in *Proc. of Conf. on Decis. and Contr. (CDC)*, 2018, pp. 4173–4180.

[52] O. Kupferman, "Automata theory and model checking," in *Handbook of Model Checking*, E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, Eds. Springer International Publishing, 2018.

[53] D. Sadigh and A. Kapoor, "Safe control under uncertainty with probabilistic signal temporal logic," in *Proc. of RSS*, AnnArbor, Michigan, June 2016.

[54] M. Tiger and F. Heintz, "Incremental reasoning in probabilistic signal temporal logic," *Int. J. Approx. Reason.*, vol. 119, pp. 325–352, April 2020.

[55] C.-I. Vasile, K. Leahy, E. Cristofalo, A. Jones, M. Schwager, and C. Belta, "Control in belief space with temporal logic specifications," in *Proc. of IEEE CDC*, 2016, pp. 7419–7424.

[56] L. Wu, K. M. B. Lee, L. Liu, and T. Vidal-Calleja, "Faithful Euclidean distance field from log-Gaussian process implicit surfaces," *IEEE Robot. and Automat. Lett.*, vol. 6, no. 2, pp. 2461–2468, 2021.

[57] Y. Chen, L. Zhao, K. M. B. Lee, C. Yoo, S. Huang, and R. Fitch, "Broadcast your weaknesses: cooperative active pose-graph slam for multiple robots," in *Proc. of IEEE ICRA*, 2020.

[58] K. M. B. Lee, F. H. Kong, R. Cannizzaro, J. L. Palmer, D. Johnson, C. Yoo, and R. Fitch, "An upper confidence bound for simultaneous exploration and exploitation in heterogeneous multi-robot systems," in *Proc. of IEEE ICRA*, 2021.

[59] K. M. B. Lee, C. Yoo, and R. Fitch, "Signal temporal logic synthesis as probabilistic inference," in *Proc. of IEEE ICRA*, 2021, pp. 5483–5489.

[60] K. Y. C. To, F. H. Kong, K. M. B. Lee, C. Yoo, S. Anstee, and R. Fitch, "Estimation of spatially-correlated ocean currents from ensemble forecasts and online measurements," in *Proc. of IEEE ICRA*, 2021, pp. 2301–2307.

[61] K. M. B. Lee, W. Martens, J. Khatkar, R. Fitch, and R. Mettu, "Efficient updates for data association with mixtures of gaussian processes," in *Proc. of IEEE ICRA*, 2020, pp. 335–341.

[62] K. M. B. Lee, C. Yoo, B. Hollings, S. Anstee, S. Huang, and R. Fitch, "Online estimation of ocean current from sparse GPS data for underwater vehicles," in *Proc. of IEEE ICRA*, 2019, pp. 3443–3449.

[63] K. Y. C. To, K. M. B. Lee, C. Yoo, S. Anstee, and R. Fitch, "Streamlines for motion planning in underwater currents," in *Proc. of IEEE ICRA*, 2019, pp. 4619–4625.

[64] K. M. B. Lee, J. J. H. Lee, C. Yoo, B. Hollings, and R. Fitch, "Active perception for plume source localisation with underwater gliders," in *Proc. of Australasian Conference on Robotics and Automation (ACRA)*, 2018, p. Best Student Paper Award.

[65] ——, "Decentralised intelligence, surveillance, and reconnaissance in unknown environments with heterogeneous multi-robot systems," in *ICRA2021 Robot Swarms in the Real World: From Design to Deployment*, 2021, p. Best Poster Award.

[66] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. of IEEE ICRA*. IEEE, 1985, pp. 116–121.

[67] S. Thrun, "Learning occupancy grids with forward models," in *Proc. of IROS*, vol. 3. IEEE, 2001, pp. 1676–1681.

[68] A. akbar Agha-mohammadi, E. Heiden, K. Hausman, and G. Sukhatme, "Confidence-rich grid mapping," *The Int. J. of Rob. Res.*, vol. 38, no. 12-13, pp. 1352–1374, 2019.

[69] R. A. Finkel and J. L. Bentley, "Quad trees a data structure for retrieval on composite keys," *Acta informatica*, no. 1, pp. 1–9, 1974.

[70] D. Meagher, "Geometric modeling using octree encoding," *Computer graphics and image processing*, no. 2, pp. 129–147, 1982.

[71] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Auton. Rob.*, pp. 189–206, 2013.

[72] M. Jun and R. D'Andrea, "Path planning for unmanned aerial vehicles in uncertain and adversarial environments," in *Cooperative Control: Models, Applications and Algorithms*. Springer US, 2003, p. Ch. 6.

[73] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *The Int. J. of Rob. Res.*, 2012.

[74] S. Kim and J. Kim, *GPmap: A Unified Framework for Robotic Mapping Based on Sparse Gaussian Processes*. Springer International Publishing, 2015, pp. 319–332.

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

[76] C. E. O. Vidos and F. Ramos, "From grids to continuous occupancy maps through area kernels," in *Proc. of IEEE ICRA*, 2016.

[77] M. G. Jadidi, J. V. Miro, and G. Dissanayake, "Warped Gaussian process occupancy mapping with uncertain inputs," *Rob. and Autom. Lett.*, vol. 2, no. 2, pp. 680–687, 2017.

[78] T. Duong, N. Das, M. Yip, and N. Atanasov, "Autonomous navigation in unknown environments using sparse kernel-based occupancy mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[79] G. Francis, L. Ott, and F. Ramos, "Fast stochastic functional path planning in occupancy maps," in *Proc. of IEEE ICRA*, 2019, pp. 929–935.

[80] R. Wagner, U. Frese, and B. Bäuml, "3D modeling, distance and gradient computation for motion planning: A direct GPGPU approach," in *Proc. of IEEE ICRA*. IEEE, 2013, pp. 3586–3592.

[81] R. Wagner, U. Frese, and B. Bauml, "Real-time dense multi-scale workspace modeling on a humanoid robot," in *Proc. of IROS*. IEEE, 2013.

[82] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart, "Signed distance fields: A natural representation for both mapping and planning," in *RSS 2016 Workshop: Geometry and Beyond-Representations, Physics, and Scene Understanding for Robotics*. University of Michigan, 2016.

[83] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. of the ACM Symp. on User Interface Software and Technology*, 2011, pp. 559–568.

[84] S. A. Wilmarth, N. M. Amato, and P. F. Stiller, "Maprm: A probabilistic roadmap planner with sampling on the medial axis of the free space," in *Proc. of IEEE ICRA*, vol. 2. IEEE, 1999, pp. 1024–1031.

[85] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Proc. of the International Conference on Robotics and Automation*, 2012.

[86] B. Douillard, D. Fox, F. Ramos, and H. Durrant-Whyte, "Classification and semantic mapping of urban environments," *Int. J. of Rob. Res.*, vol. 30, no. 1, pp. 5–32, 2011.

[87] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020. [Online]. Available: https://github.com/MIT-SPARK/Kimera

[88] E. Zobeidi, A. Koppel, and N. Atanasov, "Dense incremental metric-semantic mapping via sparse gaussian process regression," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[89] K. Zheng, A. Pronobis, and R. P. N. Rao, "Learning graph-structured sum-product networks for probabilistic semantic maps," in *Proc. of Conference on Artificial Intelligence (AAAI)*, 2018.

[90] A. Pronobis and R. P. N. Rao, "Learning deep generative spatial models for mobile robots," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots*, 2017.

[91] R. Shrestha, F.-P. Tian, W. Feng, P. Tan, and R. Vaughan, "Learned map prediction for enhanced mobile robot exploration," in *Proc. of International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1197–1204.

[92] P. Cai, Y. Luo, A. Saxena, D. Hsu, and W. S. Lee, "LeTS-Drive: Driving in a crowd by learning from tree search," in *Proc. of RSS*, 2019.

[93] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *Int. J. of Rob. Res.*, vol. 36, no. 13-14, pp. 1540–1553, 2017.

[94] Y. Sung and P. Tokekar, "Algorithm for searching and tracking an unknown and varying number of mobile targets using a limited FoV sensor," in *Proc. of IEEE ICRA*, 2017, pp. 6246–6252.

[95] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, "Efficient multi-robot search for a moving target," *The Int. J. of Robot. Res.*, vol. 28, no. 2, pp. 201–219, 2009.

[96] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–J of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[97] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. Elsevier, 1970.

[98] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[99] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.

[100] O. M. Cliff, D. L. Saunders, and R. Fitch, "Robotic ecology: Tracking small dynamic animals with an autonomous aerial vehicle," *Sci. Robot.*, vol. 3, 2018.

[101] R. Vidal, O. Shakernia, H. J. Kim, D. H. Shim, and S. Sastry, "Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation," *Trans. on Robot. and Automat.*, vol. 18, no. 5, pp. 662–669, 2002.

[102] S. K. Gan, R. Fitch, and S. Sukkarieh, "Online decentralized information gathering with spatial-temporal constraints," *Auton. Rob.*, vol. 37, 2014.

[103] S. K. Gan, R. Fitch, and S. Sukkarieh, "Real-time decentralized search with inter-agent collision avoidance," in *Proc. of IEEE ICRA*, 2012, pp. 504–510.

[104] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[105] B. N. Vo and W. K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, Nov 2006.

[106] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multitarget filtering with random finite sets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, 2005.

[107] R. Mahler, D. Hall, and J. Llinas, "Random set theory for target tracking and identification," p. 14, 2001.

[108] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer, "A random finite set approach for dynamic occupancy grid maps with real-time application," *The Int. J. of Rob. Res.*, vol. 37, no. 8, pp. 841–866, 2018.

[109] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *Proc. of International Conference on Robotics and Automation*, 2009, pp. 2168–2173.

[110] Z. Zeng, Y. Zhou, O. C. Jenkins, and K. Desingh, "Semantic mapping with simultaneous object detection and localization," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[111] Z. Zeng, A. Röfer, and O. C. Jenkins, "Semantic linking maps for active visual object search," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1984–1990.

[112] K. Zheng, R. Chitnis, Y. Sung, G. Konidaris, and S. Telex, "Towards optimal correlational object search," *arXiv preprint arXiv:2110.09991*, 2021.

[113] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The Int. J. of Rob. Res.*, vol. 5, no. 3, pp. 101–119, 1986.

[114] S. Tanygin and T. William, "Mass property estimation using coasting maneuvers," *J of Guidance, Control and Dynamics*, vol. 20, no. 4, 1997.

[115] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International J of Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[116] E. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. of the IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158.

[117] T. H. Kwan, K. M. B. Lee, J. Yan, and X. Wu, "An air bearing table for satellite attitude control simulation," in *Proc. of Conference on Industrial Electronics and Applications (ICIEA)*, 2015, pp. 1420–1425.

[118] S. Eleftheriadis, T. Nicholson, M. Deisenroth, and J. Hensman, "Identification of Gaussian process state space models," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[119] R. Frigola, Y. Chen, and C. E. Rasmussen, "Variational Gaussian process state-space models," in *Advances in Neural Information Processing Systems*, vol. 27. Curran Associates, Inc., 2014.

[120] R. Turner, M. P. Deisenroth, and C. E. Rasmussen, "State-space inference and learning with Gaussian processes," *J of Machine Learning Research*, vol. 9, pp. 868–875, 2009.

[121] A. Mauroy and J. Goncalves, "Koopman-based lifting techniques for nonlinear systems identification," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2550–2565, 2019.

[122] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *The Int. J. of Rob. Res.*, vol. 6, no. 3, pp. 49–59, 1987.

[123] R. M. Sanner and J.-J. E. Slotine, "Gaussian networks for direct adaptive control," in *Proc. of American Control Conference*, 1991, pp. 2153–2159.

[124] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, "Bayesian nonparametric adaptive control using gaussian processes," *Transactions on neural networks and learning systems*, vol. 26, no. 3, pp. 537–550, 2014.

[125] N. M. Boffi, S. Tu, and J.-J. E. Slotine, "Regret bounds for adaptive nonlinear control," in *Proc. of the Conf. on Learning for Dyn. and Cont.*, vol. 144, 2021, pp. 471–483.

[126] J. Umenberger and T. B. Schön, "Optimistic robust linear quadratic dual control," in *Proc. of the Conf. on Learning for Dyn. and Cont.*, vol. 120. PMLR, 10–11 Jun 2020, pp. 550–560.

[127] T. Lee, B. D. Lee, and F. C. Park, "Optimal excitation trajectories for mechanical systems identification," *Automatica*, vol. 131, p. 109773, 2021.

[128] P. Sekhavat, M. Karpenko, and I. Ross, "UKF-based spacecraft parameter estimation using optimal excitation," in *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 5786.

[129] A. A. Feldbaum, "Dual control theory. i," *Avtomat. i Telemekh.*, vol. 21, no. 9, 1960.

[130] S. M. Kakade, A. Krishnamurthy, K. Lowrey, M. Ohnishi, and W. Sun, "Information theoretic regret bounds for online nonlinear control," in *Proc. of Neural Information Processing Systems*, 2020.

[131] L. Medagoda, S. B. Williams, O. Pizarro, J. C. Kinsey, and M. V. Jakuba, "Mid-water current aided localization for autonomous underwater vehicles," *Auton. Rob.*, vol. 40, no. 7, pp. 1207–1227, 2016.

[132] S. Fan, W. Xu, Z. Chen, and F. Zhang, "Nonlinear observer design for current estimation based on underwater vehicle dynamic model," in *Proc. of MTS/IEEE OCEANS 2016*, 2016, pp. 1–5.

[133] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE J of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.

[134] L. M. Merckelbach, R. D. Briggs, D. A. Smeed, and G. Griffiths, "Current measurements from autonomous underwater gliders," in *Proc. of the IEEE Working Conference on Current Measurement Technology*, 2008, pp. 61–67.

[135] D. Chang, W. Wu, C. R. Edwards, and F. Zhang, "Motion tomography: Mapping flow fields using autonomous underwater vehicles," *Int. J. of Rob. Res.*, vol. 36, no. 3, pp. 320–336, 2017.

[136] B. P. Russo, R. Kamalapurka, D. Chang, and J. A. Rosenfeld, "Motion tomography via occupation kernels," *arXiv preprint arXiv:2101.02677*, 2021.

[137] S. Cho and F. Zhang, "Adaptive learning for controlled lagrangian particle tracking," in *Proc. of MTS/IEEE OCEANS*, 2016.

[138] C. Yoo, J. J. Heon Lee, S. Anstee, and R. Fitch, "Path planning in uncertain ocean currents using ensemble forecasts," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8323–8329.

[139] D. Dey, A. Kolobov, R. Caruana, E. Kamar, E. Horvitz, and A. Kapoor, "Gauss meets canadian traveler: Shortest-path problems with correlated natural dynamics," in *Proc. of the International Conference on Autonomous Agents and Multi-Agent Systems*, ser. AAMAS '14, Richland, SC, 2014, p. 1101–1108.

[140] N. Yang, D. Chang, M. Johnson-Roberson, and J. Sun, "Energy-optimal path planning with active flow perception for autonomous underwater vehicles," in *Proc. of International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9928–9934.

[141] I. Abraham and T. D. Murphey, "Active learning of dynamics for data-driven control using koopman operators," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1071–1083, 2019.

[142] H. Kurniawati, "Partially observable Markov decision processes and robotics," *Annual Review of Control, Rob. and Auton. Systems*, vol. 5, pp. 253–277, 2022.

[143] S. M. LaValle, *Planning Algorithms*. USA: Cambridge University Press, 2006.

[144] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP planning with regularization," in *Adv. in Neural Info. Process. Syst.*, C. J. C. Burges, L. Bottou,

M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[145] C. Lusena, J. Goldsmith, and M. Mundhenk, "Nonapproximability results for partially observable markov decision processes," *J of artificial intelligence research*, vol. 14, pp. 83–103, 2001.

[146] N. Roy, "Finding approximate pomdp solutions through belief compression," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, September 2003.

[147] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. of Rob. Res.*, vol. 28, pp. 1448–1465, 2009.

[148] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *Rob. and Autom. Lett.*, vol. 3, pp. 1025 – 1032, 2018.

[149] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *Proc. of IEEE ICRA*, 2014, pp. 6447–6454.

[150] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 2016.

[151] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[152] J. Veness, K. S. Ng, M. Hutter, W. Uther, and D. Silver, "A Monte-Carlo AIXI approximation," *J. of Artificial Intelligence Research*, vol. 40, pp. 95–142, 2011.

[153] S. C. W. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Planning under uncertainty for robotic tasks with mixed observability," *Int. J. of Rob. Res.*, vol. 29, no. 8, pp. 1053–1068, 2010.

[154] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.

[155] B. Yamauchi *et al.*, "Frontier-based exploration using multiple robots," in *Agents*, vol. 98, 1998, pp. 47–53.

[156] G. Best, J. Fiagl, and R. Fitch, "Online planning for multi-robot active peception with self-organising maps," *Auton. Rob.*, vol. 42, no. 4, pp. 715–738, 2018.

[157] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel, and R. Siegwart, "Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics," in *Proc. of IEEE ICRA*, 2015, pp. 6423–6430.

[158] A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *Proc. of Unvertainty in Artificial Intelligence*, 2005, pp. 324–331.

[159] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. J. of Rob. Res.*, vol. 33, no. 9, pp. 1271–1287, 2014.

[160] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3D object reconstruction," *Auton. Rob.*, vol. 42, p. 197–208, 2018.

[161] A. Krause and D. Golovin, "Submodular function maximization." *Tractability*, vol. 3, pp. 71–104, 2014.

[162] C. Chekuri and M. Pal, "A recursive greedy algorithm for walks in directed graphs," in *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, 2005, pp. 245–253.

[163] M. Corah and N. Michael, "Efficient online multi-robot exploration via distributed sequential greedy assignment," in *Proc. of RSS*, 2017.

[164] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-MCTS: Decentralized planning for multi-robot active perception," *The Int. J. of Robot. Res.*, vol. 38, no. 2-3, pp. 316–337, 2019.

[165] T. Patten, W. Martens, and R. Fitch, "Monte carlo planning for active object classification," *Auton. Rob.*, vol. 42, no. 2, pp. 391–421, 2017.

[166] M. Popović, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran, "Online informative path planning for active classification using uavs," in *Proc. of IEEE ICRA*, May 2017, pp. 5753–5758.

[167] K. Hsu, S. O'Callaghan, A. Reid, and S. Williams, "Informative seafloor exploration using the linearised differential entropy of Gaussian process classifiers," in *Proc. of ACRA*, 2015.

[168] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *Int. J. of Rob. Res.*, vol. 32, no. 1, pp. 3–18, 2013.

[169] A. Arora, R. Fitch, and S. Sukkarieh, "An approach to autonomous science by modeling geological knowledge in a bayesian framework," in *Proc. of IROS*, 2017.

[170] G. A. Hollinger, U. Mitra, and G. S. Sukhatme, "Active classification: Theory and application to underwater inspection," in *Proc. of the International Symposium on Robotics Research (ISRR)*. Springer, 2017, pp. 95–110.

[171] N. Atanasov, B. Sankaran, J. L. Ny, G. J. Pappas, and K. Daniilidis, "Nonmyopic view planning for active object classification and pose estimation," *Transactions on Robotics*, vol. 30, no. 5, 2014.

[172] M. M. Zhang, N. Atanasov, and K. Daniilidis, "Active tactile object recognition by monte carlo tree search," *ArXiv*, vol. abs/1703.00095, 2017.

[173] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," in *Proc. of ECML*, 2005, pp. 437–448.

[174] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletins of the American Mathematical Society*, vol. 55, pp. 527–535, 1952.

[175] A. Slivkins, "Introduction to multi-armed bandits," *arXiv preprint arXiv:1904.07272*, 2019.

[176] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Cambridge University, 1989.

[177] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[178] N. Cesa-Bianchi, C. Gentile, G. Lugosi, and G. Neu, "Boltzmann exploration done right," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[179] R. I. Brafman and M. Tennenholtz, "R-MAX: A general polynomial time algorithm for near-optimal reinforcement learning," *J. of Mach. Learn. Res.*, vol. 3, no. Oct, pp. 213–231, 2002.

[180] A. Garivier and O. Cappé, "The KL-UCB algorithm for bounded stochastic bandits and beyond," in *Proc. of Conf. Comput. Learn. Theory*, 2011, pp. 359–376.

[181] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proc. of ICML*, ser. ICML'10. USA: Omnipress, 2010, pp. 1015–1022.

[182] H. Ahn, Y. Oh, S. Choi, C. J. Tomlin, and S. Oh, "Online learning to approach a person with no regret," *Robot. and Automat. Lett.*, vol. 3, no. 1, pp. 52–59, 2018.

[183] R. Marchant and F. Ramos, "Bayesian optimisation for intelligent environmental monitoring," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 2242–2249.

[184] M. Ghasemi, E. A. Bulgur, and U. Topcu, "Task-oriented active perception and planning in environments with partially known semantics," in *Proc. of International Conference on Machine Learning*, 2020, pp. 3484–3493.

[185] R. A. MacDonald and S. L. Smith, "Active sensing for motion planning in uncertain environments via mutual information policies," *The Int. J. of Rob. Res.*, vol. 38, no. 2-3, pp. 146–161, 2019.

[186] G. Kowadlo and R. A. Russell, "Robot odor localization: A taxonomy and survey," *Int. J. of Rob. Res.*, vol. 27, no. 8, pp. 869–894, 2008.

[187] D. Chang, W. Wu, D. R. Webster, M. J. Weissburg, and F. Zhang, "A bio-inspired plume tracking algorithm for mobile sensing swarms in turbulent flow," in *Proc. of IEEE ICRA*. IEEE, 2013, pp. 921–926.

[188] R. Smith, Y. Chao, P. Li, D. A. Caron, B. Jones, and G. Sukhatme, "Planning and implementing trajectories for autonomous underwater vehicles to track evolving ocean processes based on predictions from a regional ocean model," *Int. J. of Rob. Res.*, vol. 29, pp. 1475–1497, 10 2010.

[189] M. Vergassola, E. Villermaux, and B. I. Shraiman, "'infotaxis' as a strategy for searching without gradients," *Nature*, vol. 445, p. 406, jan 2007.

[190] H. Hajieghrary, D. Mox, and M. A. Hsieh, "Information theoretic source seeking strategies for multiagent plume tracking in turbulent fields," *J. Mar. Sci. Eng.*, vol. 5, no. 1, 2017.

[191] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Auton. Rob.*, vol. 31, 2011.

[192] D. E. Clark, K. Panta, and B.-N. Vo, "The GM-PHD filter multiple target tracker," in *Int. Conf. on Info. Fusion*. IEEE, 2006, pp. 1–8.

[193] R. Vidal, S. Rashid, C. Sharp, O. Shakernia, Jin Kim, and S. Sastry, "Pursuit-evasion games with unmanned ground and aerial vehicles," in *Proc. of IEEE ICRA*, vol. 3, 2001, pp. 2948–2955 vol.3.

[194] F. Bourgault, T. Furukawa, and H. Durrant-Whyte, "Optimal search for a lost target in a Bayesian world," *Springer Tracts in Adv. Robot.*, vol. 24, pp. 209–222, 01 2003.

[195] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multi-robot information gathering," in *Proc. of RSS*, FreiburgimBreisgau, Germany, June 2019.

[196] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, "Cooperative air and ground surveillance," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 16–25, 2006.

[197] P. Rybski, N. Papanikolopoulos, S. Stoeter, D. Krantz, K. Yesin, M. Gini, R. Voyles, D. Hougen, B. Nelson, and M. Erickson, "Enlisting rangers and scouts for reconnaissance and surveillance," *IEEE Robot. Autom. Mag.*, vol. 7, no. 4, pp. 14–24, 2000.

[198] A. J. Smith, G. Best, J. Yu, and G. A. Hollinger, "Real-time distributed non-myopic task selection for heterogeneous robotic teams," *Auton. Robots*, vol. 43, no. 3, pp. 789–811, 2019.

[199] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, "A decentralized algorithm for area surveillance missions using a team of aerial robots with different sensing capabilities," in *Proc. of IEEE ICRA*, 2014, pp. 4735–4740.

[200] A. Alexopoulos, T. Schmidt, and E. Badreddin, "A pursuit-evasion game between unmanned aerial vehicles," in *Proc. of the International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, vol. 02, Sep. 2014, pp. 74–81.

[201] Y. Chen, S. Huang, R. Fitch, L. Zhao, H. Yu, and D. Yang, "On-line 3D active pose-graph slam based on key poses using graph topology and sub-maps," in *Proc. of IEEE ICRA*, 2019, pp. 169–175.

[202] P. Dames, "Distributed multi-target search and tracking using the phd filter," in *Proc. of the International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, Dec 2017, pp. 1–8.

[203] X. Ding, S. L. Smith, C. Belta, and D. Rus, "Optimal control of Markov decision processes with linear temporal logic constraints," *IEEE Trans. Automat. Contr.*, vol. 59, no. 5, pp. 1244–1257, 2014.

[204] C. Yoo, R. Fitch, and S. Sukkarieh, "Probabilistic temporal logic for motion planning with resource threshold constraints," in *Proc. of RSS*, 2012.

[205] ——, "Provably-correct stochastic motion planning with safety constraints," in *Proc. of IEEE ICRA*, 2013, pp. 981–986.

[206] T. Wongpiromsarn, U. Topcu, N. Ozay, H. Xu, and R. M. Murray, "TuLiP: A software toolbox for receding horizon temporal logic planning," in *Proc. of HSCC*, 2011, pp. 313–314.

[207] J. J. H. Lee, C. Yoo, S. Anstee, and R. Fitch, "Hierarchical planning in time-dependent flow fields for marine robots," in *Proc. of IEEE ICRA*, 2020, pp. 885–891.

[208] A. Donzé, "On signal temporal logic," in *Runtime Verification*, A. Legay and S. Bensalem, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 382–383.

[209] J. V. Deshmukh, A. Donzé, S. Ghosh, X. Jin, G. Juniwal, and S. A. Seshia, "Robust online monitoring of signal temporal logic," *Formal Methods in Syst. Des.*, pp. 5–30, 2017.

[210] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *Proc. of IEEE CDC*, 2014, pp. 81–87.

[211] I. Haghighi, N. Mehdipour, E. Bartocci, and C. Belta, "Control from signal temporal logic specifications with smooth cumulative quantitative semantics," in *Proc. of IEEE CDC*, 2019, pp. 4361–4366.

[212] Y. Gilpin, V. Kurtz, and H. Lin, "A smooth robustness measure of signal temporal logic for symbolic control," *Cont. Sys. Lett.*, vol. PP, pp. 1–1, 06 2020.

[213] K. Leung, N. Aréchiga, and M. Pavone, "Back-propagation through signal temporal logic specifications: Infusing logical structure into gradient-based methods," in *Algorithmic Foundation of Robotics XIV*, S. M. LaValle, M. Lin, T. Ojala, D. Shell, and J. Yu, Eds. Springer, 2021, vol. 17, pp. 432–449.

[214] K. J. Leahy, D. Aksaray, and C. Belta, "Informative path planning under temporal logic constraints with performance guarantees," in *Proc. of American Control Conference*, 2017.

[215] R. Peck, C. Olsen, and J. Devore, *Introduction to Statistics & Data Analysis*, 3rd ed. Thomson Higher Education, 2008.

[216] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," pp. 1–66, 2007.

[217] S. Särkkä, "Linear operators and stochastic partial differential equations in Gaussian process regression," in *International Conference on Artificial Neural Networks and*

Machine Learning, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 151–158.

[218] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Wiley, 2006.

[219] C. E. Shannon and W. Weaver, *The mathematical theory of communication*. University of Illinois Press, 1949.

[220] K.-C. Ma, L. Liu, H. K. Heidarsson, and G. S. Sukhatme, "Data-driven learning and planning for environmental sampling," *J. Field Rob.*, vol. 35, pp. 643–661, 2018.

[221] C. Walck, *Hand-book on statistical distributions for experimentalists*, 3rd ed. Internal Report SUF-PFY/96-01, 2007.

[222] M. D. Donsker and S. S. Varadhan, "Asymptotic evaluation of certain Markov process expectations for large time," *Commun. on Pure and Appl. Math.*, vol. 36, p. 183–212, 1983.

[223] P. Dupuis and R. S. Ellis, *A Weak Convergence Approach to the Theory of Large Deviations*. Wiley, 1997.

[224] Y. Seldin, F. Laviolette, N. Cesa-Bianchi, J. Shawe-Taylor, and P. Auer, "PAC-Bayesian inequalities for martingales," *IEEE Trans. on Info. Theory*, vol. 58, no. 12, pp. 7086–7093, 2012.

[225] X. Lu and B. V. Roy, "Information-theoretic confidence bounds for reinforcement learning," in *Adv. Neural Info. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[226] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, March 2012.

[227] M. Corah and N. Michael, "Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice," *Auton. Rob.*, vol. 43, no. 2, pp. 485–501, 2019.

[228] D. H. Wolpert and S. Bieniawski, "Distributed control by lagrangian steepest descent," in *Proc. of the IEEE Conference on Decision and Control (CDC)*, vol. 2, 2004, pp. 1562–1567 Vol.2.

[229] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system." [Online]. Available: https://www.ros.org

[230] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. of Field Robot.*, vol. 36, 10 2018.

[231] P. Fankhauser and M. Hutter, "A universal grid map library: Implementation and use case for rough terrain navigation," in *Robot Operating System (ROS) –The Complete Reference*, A. Koubaa, Ed. Springer, 2016, vol. 1, ch. 5.

[232] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, "RotorS—a modular Gazebo MAV simulator framework," in *Robot Operating System (ROS): The Complete Reference*, A. Koubaa, Ed. Springer, 2016, vol. 1, pp. 595–625.

[233] Mission Systems Pty. Ltd, "MORSE: the modular open robots simulator engine." [Online]. Available: https://github.com/mission-systems-pty-ltd/morse

[234] D. Valenti, G. Denaro, R. Ferreri, S. Genovese, S. Aronica, S. Mazzola, A. Bonano, G. Basilone, and B. Spagnolo, "Spatio-temporal dynamics of a planktonic system and chlorophyll distribution in a 2d spatial domain: matching model and data," *Scientific Reports*, vol. 7, no. 220, 2017.

[235] J. T. Holt and R. Proctor, "The role of advection in determining the temperature structure of the Irish Sea," *J of Physical Oceanography*, vol. 33, no. 11, pp. 2288–2306, 2003.

[236] S. M. Mrazovac, P. R. Milan, M. B. Vojinovic-Miloradov, and B. S. Tosic, "Dynamic model of methane–water diffusion," *Appl. Math. Model.*, vol. 36, no. 9, pp. 3985 – 3991, 2012.

[237] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The Int. J. of Rob. Res.*, vol. 34, no. 7, pp. 883–921, 2015.

[238] R. Lupini and T. Tirabassi, "Gaussian plume model and advection-diffusion equation: An attempt to connect the two approaches," *Atmos. Environ.*, vol. 13, no. 8, pp. 1169 – 1174, 1979.

[239] S. Särkkä and J. Hartikainen, "Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression," in *Proc. of AISTATS*, 2012, pp. 993–1001.

[240] J. Lebl, *Basic Analysis I*, version 5.6 ed. Creative Commons, 2022.

[241] N. E. Leonard and J. G. Graver, "Model-based feedback control of autonomous underwater gliders," *IEEE J. Ocean. Eng.*, vol. 26, no. 4, pp. 633–645, 2001.

[242] C. Yoo and C. Belta, "Control with probabilistic signal temporal logic," *arXiv preprint arXiv:1510.08474*, 2015.

[243] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Contr. Syst. Lett.*, vol. 3, no. 1, pp. 96–101, 2019.

[244] N. Mehdipour, C. Vasile, and C. Belta, "Arithmetic-geometric mean robustness for control from signal temporal logic specifications," in *Proc. of IEEE ACC*, 2019, pp. 1690–1695.

[245] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng,

"TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: http://tensorflow.org/

[246] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[247] M. Cognetti, P. Salaris, and P. R. Giordano, "Optimal active sensing with process and measurement noise," in *Proc. of IEEE ICRA*, 2018.

[248] H. J. Kappen, V. Gómez, and M. Opper, "Optimal control as a graphical model inference problem," *Mach. learn.*, vol. 87, no. 2, pp. 159–182, 2012.

[249] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," *arXiv preprint arXiv:1805.00909*, 2018.

[250] N. Rhinehart, R. McAllister, and S. Levine, "Deep imitative models for flexible inference, planning, and control," in *Proc. of ICLR*, April 2020.

[251] B. Lee, C. Zhang, Z. Huang, and D. D. Lee, "Online continuous mapping using Gaussian process implicit surfaces," in *Proc. of IEEE ICRA*, 2019, pp. 6884–6890.

[252] J. D. Zika, M. H. England, and W. P. Sijp, "The ocean circulation in thermohaline coordinates," *J of Physical Oceanography*, vol. 42, no. 5, pp. 708–724, 2012.

[253] C. Cenedese and A. L. Gordon, "Ocean current," in *Encyclopedia Britannica*, 2021. [Online]. Available: https://www.britannica.com/science/ocean-current

[254] P. J. Pritchard and J. W. Mitchell, *Fox and McDonald's Introduction to Fluid Mechanics*, 8th ed. Wiley, 2011.

[255] W. Martens, Y. Poffet, P. R. Soria, R. Fitch, and S. Sukkarieh, "Geometric Priors for Gaussian Process Implicit Surfaces," *Rob. and Autom. Lett.*, vol. 2, no. 2, pp. 373–380, 2017.

[256] E. Solak, R. Murray-Smith, W. Leithead, D. Leith, and C. Rasmussen, "Derivative observations in Gaussian process models of dynamic systems," in *Proc. of Advances in Neural Information Processing Systems 15*, 2002, p. 8.

[257] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D euclidean signed distance fields for on-board mav planning," in *Proc. of IROS*, 2017, pp. 1366–1373.

[258] O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," 2007.

[259] S. Mauch, "A fast algorithm for computing the closest point and distance transform," *Technical Report caltechASCI/2000.077, Applied and Computational Math., Calif. Inst. of Technology*, 2000.

[260] J. A. Stork and T. Stoyanov, "Ensemble of sparse Gaussian process experts for implicit surface mapping with streaming data," *2020 IEEE International Conference on Robotics and Automation(ICRA)*, 2020.

[261] L. Wu, R. Falque, V. Perez-Puchalt, L. Liu, N. Pietroni, and T. Vidal-Calleja, "Skeleton-based conditionally independent Gaussian process implicit surfaces for fusion in sparse to dense 3D reconstruction," *Rob. and Autom. Lett.*, no. 2, pp. 1532–1539, 2020.

[262] B. Lau, C. Sprunk, and W. Burgard, "Improved updating of euclidean distance maps and voronoi diagrams," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010.

[263] K. Crane, C. Weischedel, and M. Wardetzky, "Geodesics in heat," *ACM Transactions on Graphics*, 2012.

[264] S. R. S. Varadhan, "On the behavior of the fundamental solution of the heat equation with variable coefficients," *Communications on pure and applied mathematics*, pp. 431–455, 1967.

[265] A. G. Belyaev and P. A. Fayolle, "On variational and PDE-based distance function approximations," *COMPUTER GRAPHICS Forum*, pp. 1–16, 2015.

[266] P. Whittle, "On stationary processes in the plane," *Biometrika*, 1954.

[267] E. J. Sondik, "The optimal control of partial observable Markov processes," Ph.D. dissertation, Stanford University, 1971.

[268] G. H. Polychronopoulos and J. N. Tsitsiklis, "Stochastic shortest path problems with recourse," *Networks*, vol. 27, pp. 133–143, 1996.

[269] G. Best, M. Forrai, R. R. Mettu, and R. Fitch, "Planning-aware communication for decentralised multi-robot coordination," in *Proc. of IEEE ICRA*, 2018, pp. 1050–1057.

[270] S. Kemna and G. S. Sukhatme, "Surfacing strategies for multi-robot adaptive informative sampling with a surface-based data hub," in *Proc. of MTS/IEEE OCEANS*, 2018, pp. 1–10.

[271] G. Best and R. Fitch, "Bayesian intention inference for trajectory prediction with an unknown goal destination," in *Proc. of IROS*, 2015, pp. 5817–5823.

[272] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "PRECOG: Prediction conditioned on goals in visual multi-agent settings," in *Proc. of Int. Conf. on Comput. Vision*, October 2019.

[273] D. K. Misra, J. Sung, K. Lee, and A. Saxena, "Tell me dave: Context-sensitive grounding of natural language to manipulation instructions," *The Int. J. of Rob. Res.*, vol. 35, no. 1-3, pp. 281–300, 2016.

[274] J. J. Gibson, *The Theory of Affordances.* Lawrence Erlbaum Associates, Inc., 1979.

[275] T. Mar, V. Tikhanoff, G. Metta, and L. Natale, "Self-supervised learning of grasp dependent tool affordances on the icub humanoid robot," in *Proc. of IEEE ICRA*, 2015, pp. 3200–3206.

[276] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for 3D LiDAR-based huamn detection: experimental analysis of point cloud clustering and classification methods," *Auton. Robots*, vol. 44, pp. 147 – 164, 2020.

[277] T. Baumann, "Obstacle avoidance for drones using a 3DVFH* algorithm," Master's thesis, ETH, Spring 2018.

[278] P. M. Newman, "MOOS-Mission orientated operating suite," 2008.