Bayesian Optimisation for Planning under Uncertainty

RAFAEL DOS SANTOS DE OLIVEIRA

Supervisor: Prof. Fabio Ramos

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

Faculty of Engineering
The University of Sydney
Australia

June 2019

Declaration

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

Rafael dos Santos de Oliveira

June 2019

Abstract

Under an increasing demand for data to understand critical processes in our world, robots have become powerful tools to automatically gather data and interact with their environments. In this context, this thesis addresses planning problems where limited prior information leads to uncertainty about the outcomes of a robot's decisions. The methods are based on Bayesian optimisation (BO), which provides a framework to solve planning problems under uncertainty by means of probabilistic modelling.

As a first contribution, the thesis provides a method to find energy-efficient paths over unknown terrains. The method applies a Gaussian process (GP) model to learn online how a robot's power consumption varies as a function of its configuration while moving over the terrain. BO is applied to optimise trajectories over the GP model being learnt so that they are informative and energetically efficient. The method was tested in experiments on simulated and physical environments.

A second contribution addresses the problem of policy search in high-dimensional parameter spaces. To deal with high dimensionality the method combines BO with a coordinate-descent scheme that greatly improves BO's performance when compared to conventional approaches. The method was applied to optimise a control policy for a race car in a simulated environment and shown to outperform other optimisation approaches.

Finally, the thesis provides two methods to address planning problems involving uncertainty in the inputs space. The first method is applied to actively learn terrain roughness models via proprioceptive sensing with a mobile robot under localisation uncertainty. Experiments demonstrate the method's performance in both simulations and a physical environment. The second method is derived for more general optimisation problems. In particular, this method is provided with theoretical guarantees and empirical performance comparisons against other approaches in simulated environments.

Acknowledgements

This thesis is the result of work that could not be done without the support of many others that were present during the four years of the Ph. D. program. First and foremost, I would like to thank God for opening up this opportunity and for sustaining and enabling me during the course of the program.

Prof. Fabio Ramos has been an excellent supervisor. His support and ideas guided me throughout the Ph. D. Even at very busy times, Fabio would always find time to meet with his students and help us solve issues related to our research.

Dr. Lionel Ott, though not officially, has been a great co-supervisor during the course of the Ph. D. Lionel's shared experience and practical advice helped shaping my understanding of research.

I would also like to thank other people that I collaborated with in papers, such as Dr. Vitor Guizlini, Prof. Fernando Rocha and Prof. Valdir Grassi Jr. Their input and combined effort helped me to understand that research is usually not done alone, but as a result of different people with their perspectives, knowledge and ideas working together for a common goal.

Conversations, ideas and support from my fellow students, post-docs and other researchers have been interesting, helpful and fun during the whole course. I would like to thank all of them for their support.

Thanks also go to my home country, Brazil, whose government provided me with a scholarship to pursue these studies in Australia and to *Coordenção de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES), which managed the scholarship.

Lastly, I would like to thank my family. They have been my greatest supporters. Without them, this thesis would not have been possible.

CONTENTS

List of Figures

List of Tables

Nomenclature

Abbreviations

i.i.d.	independent and identically distributed
a.e.	almost everywhere
a.s.	almost surely
BO	Bayesian optimisation
CDBO	coordinate-descent Bayesian optimisation
CMA-ES	Covariance Matrix Adaptation Evolution Strategy
DUCB	distance-penalised upper confidence bound
EI	expected improvement
GP	Gaussian process
GPS	Global Positioning System
IMU	inertial measurement unit
RKHS	reproducing kernel Hilbert space
RMSE	root mean square error
RTK	real-time kinematics
UCB	upper confidence bound
WRMSE	weighted root mean square error

Basic fonts

a	a scalar
$\mathbf{a}$	a vector
$\mathbf{A}$	a matrix
$\mathcal{A}$	a set
$\mathfrak{A}$	a collection of sets, such as a σ-algebra (sigma-algebra) or a topology.
A	a functional or a measure

Matrix operations

$[\mathbf{A}]_{ij}$	the element in the i'th row and j'th column of $\mathbf{A}$
$\mathbf{A}^\mathsf{T}$	transpose

$\mathbf{A}^{-1}$	inverse
$\mathrm{tr}(\mathbf{A})$	trace
$\lvert\mathbf{A}\rvert$	determinant

Basic operations

$\mathcal{U}\backslash\mathcal{V}$	the complement of set $\mathcal{V}$ in $\mathcal{U}$
$\langle f, g\rangle_{\mathcal{F}}$	the inner product between $f, g \in \mathcal{F}$
$\limsup\limits_{n\to\infty} a_n$	the limit superior of a real-valued sequence, i.e. $\limsup_{n\to\infty} a_n := \lim_{n\to\infty}\left(\sup_{t\geq n} a_t\right)$
$\lVert f\rVert_{\mathcal{F}}$	the norm of $f \in \mathcal{F}$
$\overline{\mathcal{U}}$	the closure of the set $\mathcal{U}$, i.e. the smallest closed set containing $\mathcal{U}$.
$\mathbf{x}\cdot\mathbf{y}$	the dot product, i.e. the standard inner product in Euclidean vector spaces
$f(\mathcal{U})$	The image of set $\mathcal{U}$ under the mapping f
$f^{-1}(\mathcal{V})$	the pre-image of set $\mathcal{V}$ under the mapping f
$\lvert\mathcal{F}\rvert$	the cardinality of a set, i.e. the number of elements in set $\mathcal{F}$, or ∞ if $\mathcal{F}$ is not finite
$\lvert a\rvert$	the absolute value of $a \in \mathbb{R}$

Special symbols

$\mathfrak{B}(\mathcal{X})$	the Borel σ-algebra of a topological space $\mathcal{X}$. Case $\mathcal{X} = \mathbb{R}^d$, $d \in \mathbb{N}$, $\mathfrak{B}^d := \mathfrak{B}(\mathbb{R}^d)$.
$\mathcal{D}_n$	a dataset containing n entries
$\varnothing$	the empty set
$\mathbb{E}$	expected value
$\mu_n(\cdot)$	posterior mean function of a Gaussian process given a set of n observations
$\mathcal{H}$	a Hilbert space
$\mathcal{H}_k$	the reproducing kernel Hilbert space associated with a kernel k
$\mathbb{1}_{\mathcal{A}}(\cdot)$	the indicator function of a set $\mathcal{A}$
$\mathbb{N}$	the set of natural numbers, i.e. the positive integers, $\mathbb{N} = \{1, 2, 3, \dots\}$
$N(\hat{\mathbf{x}}, \mathbf{\Sigma})$	a normal distribution, i.e. a Gaussian probability measure, with mean $\hat{\mathbf{x}}$ and covariance $\mathbf{\Sigma}$
P	a probability measure
$\mathbb{P}\{\cdot\}$	probability of an event involving random variables, e.g. for a random variable ξ defined on a probability space $(\mathcal{X}, \mathfrak{A}, P)$ and valued in $(\mathcal{W}, \mathfrak{W})$, $\mathbb{P}\{\xi \in \mathcal{A}\} = P[\xi^{-1}(\mathcal{A})]$, $\forall\mathcal{A} \in \mathfrak{W}$.
$\mathbb{R}$	The field of real numbers, the real line
$\mathcal{O}$	big-O notation for the set of upper bounds valid up to a constant factor
$\sigma_n^2(\cdot)$	posterior variance function of a Gaussian process given a set of n observations
$\mathbb{V}$	variance

CHAPTER 1

Introduction

1.1 Motivation

As civilisation advances, there has been an increasing need to understand our world and the countless processes that take place within it. Environmental issues, planetary exploration, social media behaviours and many other related problems have the common characteristic of requiring data to be modelled and understood (Figure 1.1). In this sense, robots have become powerful tools to actively collect data from their environments (Marchant and Ramos, 2012; Candela et al., 2017; Bessi and Ferrara, 2016). Machine learning algorithms (Murphy, 2012) running on robotic platforms have the opportunity to be supplied with data in a self-supervised approach, using the robot to label new data points. This kind of automation requires the development of algorithms to actively guide the robot through the data collection process.

Planning algorithms search for strategies to take a system from a non-ideal initial state to a better, so called, goal state (LaValle, 2006). When applied to robotic systems, planners can guide the robot through the environment in order to accomplish a pre-defined task, such as learning an environmental model. Information from observed data allows us to validate and update theoretical models to explain different kinds of physical environments and their dynamics. At the same time, models aid the planning algorithm in guiding the robot, either to avoid hazards, such as obstacles and areas difficult to traverse, or to indicate areas in the model where more data is needed. However, each robot is limited by its sensors.

Sensors allow robots to perceive the world around them, but each sensor is subject to limitations. Cameras and laser ranging sensors cannot see through obstacles. Global positioning systems are subject to atmospheric interferences, multi-path reflections and satellite visibility. Orientation sensors, such as inertial measurement units, provide estimates that drift over time. All of these add uncertainty to models learnt from sensor data.

(A) Planetary exploration

(B) Air pollution

(C) Social media

FIGURE 1.1: Modern issues demanding increasing amounts of data: (a) Spacecrafts and robots have been sent to explore other planets in our solar system (Courtesy from NASA/JPL). (b) Increasing demands for energy and manufactured goods have been causing environmental issues, such as air pollution (Courtesy from *Pexels.com*). (c) Social media behaviours have interested both marketing companies and governments (Courtesy from *Pexels.com*).

Besides uncertainty from the sensors, planning algorithms usually assume that a pre-defined cost map is available to evaluate the feasibility of certain actions with a robot (Zucker et al., 2013). When traversing unknown environments, however, information about motion feasibility, as provided by traversability metrics (Martin et al., 2013), geometric maps and dynamical models (Liniger et al., 2015), can only be acquired online, breaking classical assumptions.

This thesis is motivated by planning problems where limited prior information leads to uncertainty about the outcomes of decisions. The motivation is mainly based on robotics problems, which present challenges that can illustrate well this thesis' concerns. However, the techniques are derived in a general approach, so that they can also be applied to areas outside of robotics.

1.2 Bayesian optimisation as a framework for planning under uncertainty

Planning actions for a robot can be formulated as an optimisation problem (Marchant and Ramos, 2012; Zucker et al., 2013; Marinho et al., 2016; Mukadam et al., 2017). In this case, one acts in order to optimise an objective function, which encodes goals and costs. In the case of motion planning, costs can usually be expressed as convex objectives, so that gradient-based approaches are well-suited (Zucker et al., 2013). However, other problems offer non-convex objectives. In this case, global optimisation is desired, such as finding:

$$\mathbf{x}^* \in \operatorname*{argmax}_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) , \tag{1.1}$$

where $\mathcal{S} \subset \mathbb{R}^d$ is a compact search space where the algorithm can choose its actions, or queries, $\mathbf{x}$ from, and f is the objective function. In practice, the objective may represent an environmental process, a terrain model, the performance of a given policy, or other objects of interest. Actions can correspond to spatial locations in an environmental process, parameters of a control policy, trajectories, etc. An optimisation algorithm can then use a robot to evaluate each action and find an optimal one.

For an arbitrary problem, the objective function is usually not known in closed form. Traditional optimisation methods that require gradients are therefore not applicable. In addition, evaluations of the objective function with a robot can be costly and noisy due to limitations on the robot, such as energy supply and sensor imperfections. Techniques that do not depend on gradients, such as evolutionary methods (Arnold and Hansen, 2010), and Lipschitz optimisation (Malherbe and Vayatis, 2017), usually require a large amount of function evaluations or assume noiseless observations.

Bayesian optimisation (BO) offers a principled approach to solve optimisation problems where objective function evaluations are noisy and expensive (Brochu et al., 2010; Shahriari et al., 2016). BO algorithms have found success in a variety of applications, such as tuning parameters for machine learning algorithms (Snoek et al., 2012), robotic environmental monitoring (Marchant and Ramos, 2012), policy search (Wilson et al., 2014), computational biology (Ulmasov et al., 2016), etc. Objectives can be considered as partially-observable functions of the algorithm's queries, i.e. the output is only observable through noisy samples. A BO algorithm then learns a probabilistic model of the objective function by observing the outcomes of its own actions. The model is combined with selection criteria to choose which actions should be tried next by the algorithm at each stage of the process. More details about the BO framework will be presented in Chapter 2.

1.3 Problem statement

Despite its success, many open problems are still present when applying Bayesian optimisation to robotics, a few of which this thesis considers. Firstly, BO has been applied to optimise continuous informative paths over environmental processes (Marchant and Ramos, 2014). Then it might be possible to apply BO as a method to optimise paths over unknown terrains that robots need to traverse. Secondly, dealing with the optimisation of paths and policies (Wilson et al., 2014) may also lead to high-dimensional parameter spaces, which leads to problems in BO that have only recently been explored (Chen et al., 2012; Wang et al., 2013; Kandasamy et al., 2015; Wang et al., 2018a). Lastly, a common assumption in BO approaches is that queries are executed precisely at where they were intended to be. When using a robot to execute queries, that assumption typically does not hold, and very few BO methods have been recently proposed to address this issue (Nogueira et al., 2016; Beland and Nair, 2017). In addition, estimates of the query location can be provided as probability distributions by localisation systems (Thrun et al., 2006). BO algorithms may be able to use probability distributions, instead of point estimates, to learn models of an objective function.

In a general, high-level description, this thesis addresses the problem of: *how to select actions to perform with a robot in order to optimise a partially-observable objective?* Partial observability means that the outcomes of the robot's actions can only be observed through samples subject to noise. It is also assumed that evaluations are costly and limited by a given budget.

1.4 Contributions

This thesis is composed of four contributing chapters, whose main contributions are summarised below.

1.4.1 Finding feasible paths

Chapter 3 addresses the possibility of using BO methods to learn feasible paths for a robot to navigate through unknown terrain. The chapter presents a BO algorithm to select energy-efficient paths between given start and goal locations while learning a model of the robot's power consumption over the terrain. The model is based on Gaussian process (GP) regression (Rasmussen and Williams, 2006), and paths are formulated as parametric curves. Experiments evaluate the approach both in simulation and with a physical robot.

1.4.2 BO in high-dimensional search spaces

Optimising paths and control policies leads to high-dimensional search spaces. Chapter 4 then proposes a BO method to optimise functions of many parameters by combining BO with coordinate-descent schemes. The method is applied to the problem of learning control policies for a robot race car with the goal of minimising its lapping time. The algorithm has no prior information about the track nor the robot's dynamical model, just an initial sub-optimal policy example. Policies are parameterised as functions in a reproducing kernel Hilbert space (RKHS) (Schölkopf and Smola, 2002). Experiments demonstrate the performance of the algorithm against other optimisation methods in a simulated racing environment.

1.4.3 Planning under localisation uncertainty

When dealing with physical robots, localisation uncertainty may corrupt models and mislead algorithms guiding the robot. The traditional approach is to either use mean estimates or to assume that the robot is following its intended trajectory. Chapter 5 investigates problems where location distributions provided by localisation systems (Thrun et al., 2006), instead of point estimates, can be used as inputs to models of the robot's environment. The chapter proposes a method using a Gaussian process model that takes probability distributions as inputs as a prior for BO. The proposed method also applies heuristics that consider the distance between the sampling locations when deciding where to send the robot next. An application is given to the problem of learning terrain roughness models while safely navigating the robot, and it is evaluated both in simulation and in an experiment with a physical robot.

1.4.4 Optimisation under uncertain inputs

The investigation in Chapter 5 opens up the question of whether a Bayesian optimisation algorithm can maintain theoretical convergence guarantees in problems involving uncertain inputs. Chapter 6 then presents a method for BO under uncertain inputs and theoretical guarantees for this method. The chapter also presents under which assumptions a conventional BO algorithm (Chowdhury and Gopalan, 2017) may converge despite the presence of noisy inputs. Experiments in simulation complement the results by evaluating the performance of the different BO approaches to input noise in a variety of scenarios.

1.5 Outline

After this introduction, this thesis' main components are a background chapter, followed by four contributing chapters, and a conclusion, which are outlined below.

Chapter 2 presents the theoretical background necessary to understand the main contributions of this thesis. Bayesian optimisation and Gaussian processes are formally introduced. The chapter also presents reproducing kernel Hilbert spaces, present in chapters 4 and 6, and other background necessary to understand the results in Chapter 6.

The four contributing chapters (3 through 6) follow the same basic structure. After a motivating introduction, each chapter describes related work, methodology and experiments. Chapter 3 presents the first contribution of this thesis: a method to find energy-efficient paths over unknown terrains. As a second contribution, Chapter 4 presents a BO method for policy-search problems involving high-dimensional parameter spaces. Chapter 5 addresses the problem of planning under localisation uncertainty, providing a method to learn terrain roughness models while safely navigating a mobile robot. Based on the investigation in the previous chapter, as a final contribution, Chapter 6 provides a method with theoretical guarantees to solve more general optimisation problems under uncertain inputs.

The thesis closes in Chapter 7 with a review of the contributions and directions for future research. In addition, Appendix A, at the end of the thesis, presents the derivation of proofs for auxiliary theoretical results, such as lemmas and propositions, which were presented in Chapter 6.

CHAPTER 2

Background

This chapter presents a review of concepts that form the basis for the methods this thesis proposes. Most of the problems faced by this work deal with variables assumed to be uncertain. In this case, a logical choice of approach is to apply probabilistic models and Bayesian reasoning. To form a basis for this approach, the chapter begins by reviewing some of the basic concepts of supervised learning in the Bayesian setting (Section 2.1). The focus is on regression models. In particular, Gaussian process (GP) regression (Section 2.2) is present throughout this thesis. Bayesian optimisation with GP priors is introduced in Section 2.3. GPs are also closely related to the topic of reproducing kernel Hilbert spaces (Section 2.4), which have their own variety of applications. The chapter also presents a review on some of the basics of topology (Section 2.5) and measure theory (Section 2.6), which play a role in most of the theoretical results in Chapter 6. Lastly, the chapter closes in Section 2.2.4 with a discussion on GP models accounting for noisy inputs. At the end of the chapter, a brief summary of the topics with their connections to the rest of the thesis and a few final bibliographical remarks are provided.

2.1 Bayesian learning

Let us begin by considering the case of supervised learning (Murphy, 2012). Consider a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ composed of n input-output pairs $(\mathbf{x}_i, y_i)$. Our task is to learn a function f mapping each input $\mathbf{x}_i$ to the corresponding output y_i, so that we can predict the function value at unobserved inputs. We can be general about the inputs, assuming they are d-dimensional vectors from a set $\mathcal{X} \subset \mathbb{R}^d$. The outputs, we will assume that they are real-valued scalars, $y \in \mathbb{R}$. In addition, we consider that observations are corrupted by noise, so that:

$$y_i = f(\mathbf{x}_i) + \nu_i \; , i = 1, \dots, n \; , \tag{2.1}$$

where ν_i is independent and identically distributed (i.i.d.) random noise drawn from a given probability distribution. In general, we will consider that the noise follows a zero-mean Gaussian distribution with variance σ_ν^2, i.e. $\nu_i \sim N(0, \sigma_\nu^2)$.

To solve the problem above, we consider that f comes from a known hypothesis space $\mathcal{H}$ and try to find the $f \in \mathcal{H}$ that best explains the data in $\mathcal{D}$. In the case of linear regression, for example, we assume that:

$$f(\mathbf{x}) = \mathbf{w}^\mathsf{T}\mathbf{x} \,, \tag{2.2}$$

where the vector $\mathbf{w}$ composes the set of parameters of our model. The hypothesis space in this case is simply $\mathcal{H} = \{f : \mathcal{X} \to \mathbb{R} \mid f(\mathbf{x}) = \mathbf{w}^\mathsf{T}\mathbf{x}, \mathbf{w} \in \mathbb{R}^d\}$. We can generalise that to the case of non-linear models by adopting a set of non-linear basis functions $\phi_i : \mathcal{X} \to \mathbb{R},\ i \in \{1, \dots, q\}$, $q \in \mathbb{N}$, and having:

$$f(\mathbf{x}) = \mathbf{w}^\mathsf{T}\boldsymbol{\phi}(\mathbf{x}) \,, \tag{2.3}$$

where $\boldsymbol{\phi}(\mathbf{x}) := [\phi_1(\mathbf{x}), \dots, \phi_q(\mathbf{x})]^\mathsf{T}$. In this case, the hypothesis space is given by $\mathcal{H} = \{f : \mathcal{X} \to \mathbb{R} \mid f(\mathbf{x}) = \mathbf{w}^\mathsf{T}\boldsymbol{\phi}(\mathbf{x}), \mathbf{w} \in \mathbb{R}^q\}$.

Given a set of parameterised functions, the task is to find the parameters $\mathbf{w}$ that best explain the data we observe. In general, this problem is formulated as minimising a data-dependent loss function over the hypothesis space, picking the $\mathbf{w}$ that best fits to it. However, there are a few drawbacks to that approach. The main issue here is that it becomes difficult to infer how confident the model is about its predictions given a limited amount of noise-corrupted data. Having models whose degree of uncertainty is quantifiable allows us to take such uncertainty into account when making decisions.

Bayesian learning quantifies model uncertainty by placing a belief distribution over the model we are trying to learn and updating that belief according to the observed data. Predictions are then performed via inference over the posterior distribution of the model conditioned on the data. By Bayes' rule, in the parametric case, the density of this posterior belief distribution at a given $\mathbf{w}$ is:

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathbf{w}, \mathcal{D})}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\mathbf{w})p(\mathbf{w})}{p(\mathcal{D})} \,, \tag{2.4}$$

where $p(\mathcal{D}|\mathbf{w})$ corresponds to the likelihood of the parameters, $p(\mathbf{w})$ is the prior, and $p(\mathcal{D})$ is called the evidence or the marginal likelihood (MacKay, 2003, Sec. 2.3). As a probability distribution, we can use the posterior $p(\mathbf{w}|\mathcal{D})$ to assess how probable each choice of $\mathbf{w}$ is to be the correct one. The posterior probability $p(\mathbf{w}|\mathcal{D})$ then encodes the model's confidence.

In the case of Bayesian linear regression (Murphy, 2012, Sec. 7.6), we can assume a Gaussian prior over the parameters vector, $\mathbf{w} \sim N(\mathbf{0}, \mathbf{\Sigma}_\mathbf{w})$. Assuming i.i.d. zero-mean Gaussian noise $\nu_i \sim N(0, \sigma_\nu^2)$, the joint distribution between the data and the parameters $p(\mathbf{w}, \mathcal{D})$ is then given by:

$$\begin{bmatrix} \mathbf{w} \\ \mathbf{y} \end{bmatrix} \sim N \left(\mathbf{0}, \begin{bmatrix} \mathbf{\Sigma}_\mathbf{w} & \mathbf{\Sigma}_\mathbf{w} \mathbf{\Phi}^\mathsf{T} \\ \mathbf{\Phi} \mathbf{\Sigma}_\mathbf{w} & \mathbf{\Phi} \mathbf{\Sigma}_\mathbf{w} \mathbf{\Phi}^\mathsf{T} + \sigma_\nu^2 \mathbf{I} \end{bmatrix} \right) , \tag{2.5}$$

where $\mathbf{y} = [y_1, \dots, y_n]^\mathsf{T}$ and $\mathbf{\Phi}$ is a n-by-q matrix with elements $\mathbf{\Phi}_{ij} = \phi_j(\mathbf{x}_i)$. Conditioning the joint (Equation 2.5) on the data $\mathcal{D}$, we get the posterior distribution (Equation 2.4) as:

$$\mathbf{w} | \mathcal{D} \sim N(\mathbf{\Sigma}_\mathbf{w} \mathbf{\Phi}^\mathsf{T} \mathbf{\Sigma}_\mathcal{D}^{-1} \mathbf{y},\ \mathbf{\Sigma}_\mathbf{w} - \mathbf{\Sigma}_\mathbf{w} \mathbf{\Phi}^\mathsf{T} \mathbf{\Sigma}_\mathcal{D}^{-1} \mathbf{\Phi} \mathbf{\Sigma}_\mathbf{w}) , \tag{2.6}$$

where $\mathbf{\Sigma}_\mathcal{D} := \mathbf{\Phi} \mathbf{\Sigma}_\mathbf{w} \mathbf{\Phi}^\mathsf{T} + \sigma_\nu^2 \mathbf{I}$. Now, given a query point $\mathbf{x}_* \in \mathcal{X}$, the value of $f(\mathbf{x}_*)$ follows:

$$f(\mathbf{x}_*) | \mathcal{D} \sim N(\boldsymbol{\phi}_*^\mathsf{T} \mathbf{\Sigma}_\mathbf{w} \mathbf{\Phi}^\mathsf{T} \mathbf{\Sigma}_\mathcal{D}^{-1} \mathbf{y},\ \boldsymbol{\phi}_*^\mathsf{T} \mathbf{\Sigma}_\mathbf{w} \boldsymbol{\phi}_* - \boldsymbol{\phi}_*^\mathsf{T} \mathbf{\Sigma}_\mathbf{w} \mathbf{\Phi}^\mathsf{T} \mathbf{\Sigma}_\mathcal{D}^{-1} \mathbf{\Phi} \mathbf{\Sigma}_\mathbf{w} \boldsymbol{\phi}_*) , \tag{2.7}$$

where $\boldsymbol{\phi}_* := \boldsymbol{\phi}(\mathbf{x}_*)$. The mean of the above distribution is our most likely prediction for $f(\mathbf{x}_*)$, and the variance quantifies the level of uncertainty about that prediction.

As discussed in Rasmussen and Williams (2006, Sec. 2.1.2), $\boldsymbol{\phi}(\mathbf{x})$ maps the d-dimensional vector $\mathbf{x}$ into a q-dimensional feature space. These maps appear in the predictive distribution (Equation 2.7) in the general form of $(\mathbf{x}, \mathbf{x}') \mapsto \boldsymbol{\phi}(\mathbf{x})^\mathsf{T} \mathbf{\Sigma}_\mathbf{w} \boldsymbol{\phi}(\mathbf{x}')$. The latter defines a positive-definite kernel or covariance function $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\mathsf{T} \mathbf{\Sigma}_\mathbf{w} \boldsymbol{\phi}(\mathbf{x}')$. Kernels allow abstracting feature maps to even infinite-dimensional feature spaces without having to deal with the elements of such spaces directly, leading to non-parametric methods. Kernel methods form an essential part of this thesis and will be further reviewed in the following sections.

The next two sections present kernel methods for Bayesian learning. Section 2.2 introduces Gaussian process regression, which arises as a natural extension of the parametric framework in Bayesian linear regression to the non-parametric setting. Section 2.4 presents reproducing kernel Hilbert spaces, which connects kernel methods to functional analysis through the framework of Hilbert spaces.

2.2 Gaussian processes

Gaussian process (GP) models (Rasmussen and Williams, 2006) provide a non-parametric framework for Bayesian inference over function spaces. Non-parametric here means that a model's parameters are not explicitly taken into account, but are inferred from the data, instead. This characteristic allows a model to be updated by simply adding a new data point without the need for retraining. As probabilistic models, GP predictions are probability distributions. In particular, a GP directly models a distribution over a space of functions, enabling Bayesian reasoning in that space.

By definition, a Gaussian process is a stochastic process (Bauer, 1981), i.e. a collection of random variables, whose finite sub-collections follow a joint Gaussian distribution (Rasmussen and Williams, 2006, Def. 2.1). Stochastic processes can also be interpreted as random functions (Berlinet and Thomas-Agnan, 2004; Rasmussen and Williams, 2006), which is the approach taken here to describe GPs.

Analogously to conventional multivariate Gaussian distributions, which are specified by a mean vector and a covariance matrix, a GP is completely specified by a *mean function* and a *covariance function*. Let's say we want to model a function $f : \mathcal{X} \to \mathbb{R}$, considering an arbitrary domain $\mathcal{X} \subset \mathbb{R}^d$. A GP model with mean function $m : \mathcal{X} \to \mathbb{R}$ and positive-definite covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ places a prior on f so that, for any finite collection of query points $\{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{X}$, the vector $\mathbf{f}_n = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)]^\mathsf{T}$ is distributed according to a Gaussian:

$$\mathbf{f}_n \sim N(\mathbf{m}_n, \mathbf{K}_n) \; , \tag{2.8}$$

where $\mathbf{m}_n := [m(\mathbf{x}_1), \dots, m(\mathbf{x}_n)]^\mathsf{T}$ and $[\mathbf{K}_n]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Now assume we are given a set of observations $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $y_i = f(\mathbf{x}_i) + \nu_i$, and $\nu_i \sim N(0, \sigma_\nu^2)$. Suppose we want to infer the value of the function at a given query point $\mathbf{x}_* \in \mathcal{X}$. According to the Gaussian process model, the vector of observations $\mathbf{y}_n := [y_1, \dots, y_n]^\mathsf{T}$ and the function value at the query location $f(\mathbf{x}_*)$ are joint normally distributed:

$$\begin{bmatrix} \mathbf{y}_n \\ f(\mathbf{x}_*) \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{m}_n \\ m(\mathbf{x}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}_n + \sigma_\nu^2 \mathbf{I} & \mathbf{k}_n(\mathbf{x}_*) \\ \mathbf{k}_n(\mathbf{x}_*)^\mathsf{T} & k(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix} \right) \; . \tag{2.9}$$

where $\mathbf{k}_n(\mathbf{x}_*) := [k(\mathbf{x}_1, \mathbf{x}_*), \dots, k(\mathbf{x}_n, \mathbf{x})]^\mathsf{T}$. Conditioning the joint distribution in Equation 2.9 on the observations, we have that:

$$f(\mathbf{x}_*) | \mathcal{D}_n \sim N(\mu_n(\mathbf{x}_*), \sigma_n^2(\mathbf{x}_*)) \; , \tag{2.10}$$

where:

$$\mu_n(\mathbf{x}_*) = m(\mathbf{x}_*) + \mathbf{k}_n(\mathbf{x}_*)^\mathsf{T}(\mathbf{K}_n + \sigma_\nu^2\mathbf{I})^{-1}(\mathbf{y}_n - \mathbf{m}_n) \;, \tag{2.11}$$

$$k_n(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k}_n(\mathbf{x})^\mathsf{T}(\mathbf{K}_n + \sigma_\nu^2\mathbf{I})^{-1}\mathbf{k}_n(\mathbf{x}') \;, \tag{2.12}$$

$$\sigma_n^2(\mathbf{x}_*) = k_n(\mathbf{x}_*, \mathbf{x}_*) \;. \tag{2.13}$$

This predictive distribution allows us to infer function values at unobserved locations. Also note that k_n in Equation 2.12 defines a valid covariance function, so that $\mathrm{GP}(\mu_n, k_n)$ is a Gaussian process, often referred to as the posterior GP (Srinivas et al., 2010).

2.2.1 Covariance functions

The GP covariance function determines how the values of random functions at different query points co-vary, i.e.:

$$\mathrm{Cov}_{f\sim\mathrm{GP}(m,k)}(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') \;, \tag{2.14}$$

where $\mathrm{GP}(m, k)$ represents a GP prior with mean m and covariance k. To define a valid covariance function, $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ needs to be positive definite, i.e. for all $n \in \mathbb{N}$, $\alpha_1, \dots, \alpha_n \in \mathbb{R}$ and $\{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{X}$, k satisfies:

$$\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \;. \tag{2.15}$$

In particular, if equality only holds for $\alpha_1 = \cdots = \alpha_n = 0$, k is called strictly positive definite [1].

Considering the Bayesian linear regression example in Equation 2.7, we have that:

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\mathsf{T}\boldsymbol{\Sigma}_\mathbf{w}\boldsymbol{\phi}(\mathbf{x}') \;, \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X} \;, \tag{2.16}$$

defines a positive-definite covariance function. If we rewrite Equation 2.7 in terms of k, we obtain exactly the GP predictive equations (Equation 2.11 and Equation 2.13) for the zero-mean case. Hence, a Bayesian linear regression model with Gaussian prior on the weights can be interpreted as a particular type of GP model.

[1] Note that this definition can be confused with the case of matrices. Some authors actually refer to what was defined as a positive-definite kernel in Equation 2.15 as positive *semi-definite* (Rasmussen and Williams, 2006). That terminology only calls positive definite the kernels that were defined as *strictly* positive definite in the description above. This thesis, instead, adopts the terminology above, since most of the theory holds if the kernels are simply positive definite, as defined.

The GP literature is filled with several other types of covariance functions for GP models. The remainder of this section presents a few covariance functions used throughout the work in this thesis. For more examples, the reader is referred to Rasmussen and Williams (2006, Ch. 4).

Linear covariance functions: One of the simplest cases is given by the *linear* covariance function:

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x}' + b\,, \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d \tag{2.17}$$

where $\mathbf{A}$ is any positive-definite matrix and $b \geq 0$. This covariance function allows us to model linear functions of the form $f(\mathbf{x}) = \mathbf{a}^\mathsf{T}\mathbf{x} + c$, where $\mathbf{a} \in \mathbb{R}^d$ and $c \in \mathbb{R}$.

Translation-invariant covariance functions: Another set of covariance functions is given by those with a translational-invariance property, i.e. $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}', \mathbf{0})$. For instance, many types of covariance functions can be defined as a radial function in terms of some distance metric ρ between the data points, i.e.:

$$k(\mathbf{x}, \mathbf{x}') = k(\rho(\mathbf{x}, \mathbf{x}'))\ . \tag{2.18}$$

Translation invariance leads to a *stationary* Gaussian process, i.e. the distribution of the process is not affected by a translation of the input space (Ledoux and Talagrand, 1991, p. 368). Hence, translation-invariant covariance functions are also referred to as *stationary*.

Stationary covariance functions: One of the most popular stationary covariance functions is the *squared exponential*, which is commonly given by:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\rho_{\mathbf{L}}^2(\mathbf{x}, \mathbf{x}')\right)\ , \tag{2.19}$$

where σ_f^2 is called the signal variance parameter, and:

$$\rho_{\mathbf{L}}(\mathbf{x}, \mathbf{x}') := \sqrt{(\mathbf{x} - \mathbf{x}')^\mathsf{T}\mathbf{L}^{-1}(\mathbf{x} - \mathbf{x}')} \tag{2.20}$$

is a common distance function for vectorial inputs, where $\mathbf{L}$ is a positive-definite matrix used for scaling. For being infinitely many times differentiable with respect to its inputs (Rasmussen and Williams, 2006), the squared exponential covariance function models very smooth functions with a GP. However, in most applications, such a level of smoothness is unrealistic. Therefore, another popular family of covariance functions is the *Matérn* class, which provides a controllable smoothness parameter ν in its construction (Rasmussen and Williams, 2006, p. 84). In particular, for a half-integer $\nu = p + 1/2$, where p is a non-negative integer, the Matérn covariance functions are given by:

$$k(\mathbf{x},\mathbf{x}') = \sigma_f^2 \exp\left(-\sqrt{2\nu}\rho_{\mathbf{L}}(\mathbf{x},\mathbf{x}')\right) \frac{p!}{(2p)!} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\sqrt{8\nu}\rho_{\mathbf{L}}(\mathbf{x},\mathbf{x}')\right)^{p-i} , \tag{2.21}$$

In the later chapters, we will specially make use of Matérn covariance functions with $\nu = \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$, commonly referring to them by the numerator (e.g. Matérn 1 refers to $\nu = 1/2$).

Hyper-parameters: The parameters of the covariance function, the noise model and the mean function of the GP model are referred to as the hyper-parameters of the model. In the case of the stationary covariance functions presented above, the noise model variance σ_ν^2, the signal variance σ_f^2 and $\mathbf{L}$ compose the set of hyper-parameters for a stationary zero-mean GP model. Commonly $\mathbf{L}$ is chosen as a diagonal matrix with $[\mathbf{L}]_{ii} = \ell_i^2$, $\ell_i > 0$, $i \in \{1, \dots, d\}$. Each ℓ_i is called a *length-scale*, as it controls the scaling of the i'th coordinate in the input space. The larger ℓ_i is, the lower is the sensitivity of the covariance function to changes in the i'th coordinate. In the special case when $\mathbf{L} = \ell^2\mathbf{I}$, having a single length-scale, the covariance function is said to be *isotropic*, since it varies equally across all directions.

2.2.2 Hyper-parameter learning

The hyper-parameters of a GP play an important role in determining the capability of a model to explain real data. Ideally one should know the hyper-parameters of a GP model in advance, as they are part of the prior over the function space being modelled. However, that is rarely the case in practice.

The machine learning literature provides a few approaches to handle unknown hyper-parameters by learning them from data (Rasmussen and Williams, 2006; Snoek et al., 2012; Wang et al., 2018b; Berkenkamp et al., 2019). The most popular approaches are based on type-II maximum likelihood (MLE) or maximum a posteriori (MAP) estimation (Rasmussen and Williams, 2006). Given a dataset with n observations $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the marginal likelihood $p(\mathcal{D}_n|\boldsymbol{\theta})$ basically determines how well GP hyper-parameters $\boldsymbol{\theta}$ explain the observed data $\mathcal{D}_n$ by marginalising $\mathbf{f}_n$ out of the joint probability density $p(\mathbf{y}_n, \mathbf{f}_n|\boldsymbol{\theta})$ given by Equation 2.9. As a result, we obtain:

$$\log p(\mathcal{D}_n|\boldsymbol{\theta}) = \log p(\mathbf{y}_n|\boldsymbol{\theta}) = -\frac{1}{2}(\mathbf{y}_n - \mathbf{m}_n^{\boldsymbol{\theta}})^\mathsf{T}(\mathbf{K}_n^{\boldsymbol{\theta}} + \sigma_\nu^2\mathbf{I})^{-1}(\mathbf{y}_n - \mathbf{m}_n) - \frac{1}{2}\log|\mathbf{K}_n^{\boldsymbol{\theta}} + \sigma_\nu^2\mathbf{I}| - \frac{n}{2}\log 2\pi , \tag{2.22}$$

where $|\mathbf{K}_n^{\boldsymbol{\theta}} + \sigma_\nu^2\mathbf{I}|$ represents the determinant of the matrix $\mathbf{K}_n^{\boldsymbol{\theta}} + \sigma_\nu^2\mathbf{I}$, and the superscripts with $\boldsymbol{\theta}$ denote dependence on the hyper-parameters.

Equation 2.22 can be maximised with respect to the hyper-parameters by gradient-based methods, when derivatives of the covariance functions with respect to the hyper-parameters are available, or by derivative-free methods, otherwise. In the works presented by this thesis, in particular, we applied local optimisation algorithms, such as the derivative-free *Constrained Optimization by Linear Approximations* (COBYLA) (Powell, 1998) and the sub-gradient-based RPROP algorithm (Riedmiller and Braun, 1993), depending on whether or not derivatives of the covariance function with respect to its hyper-parameters were available.

Optimising Equation 2.22 with respect to hyper-parameters within a bounded search space is equivalent to placing a uniform prior $p(\boldsymbol{\theta})$ over the hyper-parameters, so that MLE is equivalent to a MAP solution in this case. To perform MAP estimation with more informative priors (see Lizotte, 2008, e.g.), one can add $\log p(\boldsymbol{\theta})$ to Equation 2.22, which then acts as a regularisation term in the optimisation process.

2.2.3 Computational complexity and sparse approximations

One of main sources of computational complexity in traditional GP approaches is the inversion of the kernel matrix $\mathbf{K}_n$ (see Equation 2.9). This operation scales as $\mathcal{O}(n^3)$ with the number of data points. Although not applied in this thesis, the GP literature is filled with approximation methods to address this issue. One could use, for example, sparse approximations based on Fourier features (Rahimi and Recht, 2007), which lead to sparse spectrum Gaussian processes (Lázaro-Gredilla et al., 2010). Sparse spectrum models can achieve low computational cost $\mathcal{O}(nq^2)$, where q is the number of Fourier features, when using incremental updates (Gijsberts and Metta, 2013), making them suitable for online learning applications, as in Bayesian optimisation (Mutný and Krause, 2018). Another popular approach is to use sparse GP models based on variational inference approaches (Hensman et al., 2013), which usually require batch data. Adapting these models to online learning for Bayesian optimisation has also been done (Mcintire et al., 2016). However, such approach leads to costly operations in the updates of the variational parameters.

2.2.4 GP models accounting for input noise

Consider that measurement and execution noise affect the evaluation of a function $f : \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} \subset \mathbb{R}^d$. In this case, given a desired query input $\mathbf{x}_* \in \mathcal{X}$, the actual location where the observation is collected at, i.e. $\tilde{\mathbf{x}}_* | \mathbf{x}_* \sim P_*$, is not directly observable. Assuming that $f \sim \mathrm{GP}(m, k)$, the distribution of $f(\tilde{\mathbf{x}}_*) | \mathbf{x}_*$ is no longer Gaussian in general, due to the usually non-linear input dependencies in the GP

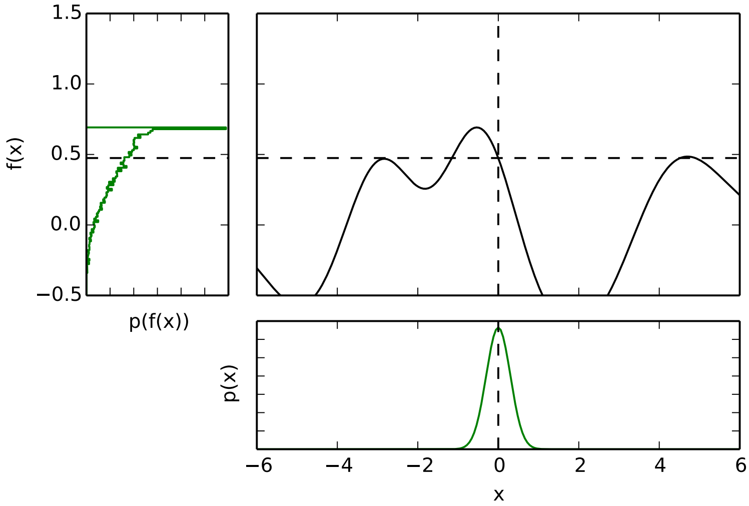


FIGURE 2.1: The effects of input noise on queries to a non-linear function. The function in the middle plot has been sampled from a squared-exponential GP model. The bottom plot presents a Gaussian query distribution. The top-left plot shows the distribution of output values. As seen, the resulting distribution of function values is no longer Gaussian due to the function's non-linearity.

posterior (Equation 2.11 and Equation 2.13). Therefore, the resulting stochastic process that represents f under noisy inputs is no longer a Gaussian process and lacks an analytic formulation (Girard, 2004; Damianou et al., 2016). Figure 2.1 presents an example where Gaussian noise affects the query of a function sampled from a GP.

The expected Gaussian process

In the machine learning literature, different methods have been proposed to learn Gaussian process models over functions with noisy inputs (Girard, 2004; Mchutchon and Rasmussen, 2011; Dallaire et al., 2011; Damianou et al., 2016). As demonstrated by Girard (2004), one can formulate a Gaussian process approximation for f by using the iterated mean and covariance of the resulting stochastic process under the influence of input noise. In particular, in the case of a constant deterministic mean function $m(\mathbf{x}) = m_0,\ \forall \mathbf{x} \in \mathcal{X}$, the mean and the covariance of this noisy process are given by:

$$\mathbb{E}_{\tilde{\mathbf{x}}}[\mathbb{E}_f[f(\tilde{\mathbf{x}})]] = m_0 \tag{2.23}$$

$$\mathrm{Cov}_{f,\tilde{\mathbf{x}},\tilde{\mathbf{x}}'}[f(\tilde{\mathbf{x}}), f(\tilde{\mathbf{x}}')] = \mathbb{E}_{\tilde{\mathbf{x}},\tilde{\mathbf{x}}'}[k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')]\ . \tag{2.24}$$

Therefore, for a set of independent random variables $\{\tilde{\mathbf{x}}_i\}_{i=1}^n$ with corresponding distributions $\{P_i\}_{i=1}^n$, we can define the *expected* covariance function (Dallaire et al., 2011) as:

$$\hat{k}(P_i, P_j) := \mathbb{E}[k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)] = \begin{cases} \mathbb{E}[k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)] = \int_{\mathcal{X}} \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}') \, \mathrm{d}P_i(\mathbf{x}) \, \mathrm{d}P_j(\mathbf{x}') \; , & i \neq j \\ \mathbb{E}[k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_i)] = \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}) \, \mathrm{d}P_i(\mathbf{x}) \; , & i = j \; . \end{cases} \tag{2.25}$$

Notice that, when $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ represent the same random variable, the above equation becomes a single expectation under the same input distribution.

Depending on the type of input distributions and the original kernel for deterministic inputs k, approximate and analytical solutions for Equation 2.25 are available in the literature (Girard, 2004; Dallaire et al., 2011; O'Callaghan and Ramos, 2012). An example is the squared exponential kernel in Equation 2.19. In this case, if the input distributions are Gaussian, the resulting covariance function, according to Dallaire et al. (2011), is given by:

$$\hat{k}(N(\hat{\mathbf{x}}_i, \boldsymbol{\Sigma}_i), N(\hat{\mathbf{x}}_j, \boldsymbol{\Sigma}_j)) = \frac{\sigma_f^2 \exp\left(-\frac{1}{2}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)^\mathsf{T}(\mathbf{L} + \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j)^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)\right)}{|\mathbf{I} + \mathbf{L}^{-1}(\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j)(1 - \delta_{ij})|^{1/2}} \; , \tag{2.26}$$

where σ_f^2 and $\mathbf{L}$ are the same hyper-parameters as described for the standard squared exponential kernel in Equation 2.19 and δ_{ij} denotes the Kronecker delta. As seen, though probability measures are abstract objects, $\hat{k}$ can be formulated as a function of the parameters of the input distributions, making them tractable, in practice.

The posterior of the stochastic process representing $f \sim \mathrm{GP}(m, k)$ under input noise is not Gaussian due to the non-linear form of Equation 2.11 and Equation 2.13 with respect to the query $\mathbf{x}$ and the data points $\mathbf{x}_i$. Yet one can obtain a suitable approximation for the original f in the noisy input setting by doing inference over a GP with mean m and covariance function $\hat{k}$, as defined in Equation 2.25. Such model is referred to as the *expected Gaussian process* model by Dallaire et al. (2011).

Despite the name, notice that the *expected* GP model does not truly represent the expected process. The true expected GP would require to compute expectations of the posterior mean and posterior variance with respect to the distributions of both the query point and the data points, which can become easily intractable. Following the terminology adopted by Dallaire et al. (2011), this thesis refers to *expected Gaussian process* as simply the model obtained by using the expected covariance function, as defined in Equation 2.25.

GP predictions under uncertain inputs

Equation 2.25 defines a covariance function over the set $\mathcal{P}$ of all probability measures on $\mathcal{X}$. A GP model with zero mean and covariance function $\hat{k} : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ places a prior over the space composed of functions $\hat{g} : \mathcal{P} \to \mathbb{R}$. Inference can then be done similarly to the way it is done with conventional GP models (Section 2.2). Given a set of observations $\mathcal{D}_n = \{(P_i, y_i)\}_{i=1}^n$, where $y_i = \hat{g}(P_i) + \nu_i$ and $P_i \in \mathcal{P}$, under the assumption that $\nu_i \sim N(0, \sigma_\nu^2)$, we have that the GP posterior mean $\hat{\mu}_n$ and variance $\hat{\sigma}_n^2$, at a given $P_* \in \mathcal{P}$, are:

$$\hat{\mu}_n(P_*) = \hat{\mathbf{k}}_n(P_*)^\mathsf{T}(\hat{\mathbf{K}}_n + \sigma_\nu^2\mathbf{I})^{-1}\mathbf{y}_n \ , \tag{2.27}$$

$$\hat{\sigma}_n^2(P_*) = \hat{k}(P_*, P_*) - \hat{\mathbf{k}}_n(P_*)^\mathsf{T}(\hat{\mathbf{K}}_n + \sigma_\nu^2\mathbf{I})^{-1}\hat{\mathbf{k}}_n(P_*) \ , \tag{2.28}$$

with $\hat{\mathbf{k}}_n(P_*) := [\hat{k}(P_*, P_1), \dots, \hat{k}(P_*, P_n)]^\mathsf{T}$ and $[\hat{\mathbf{K}}_n]_{ij} = \hat{k}(P_i, P_j)$. If $\tilde{\mathbf{x}}_*|\mathbf{x}_* \sim P_*$, the expected GP formulation approximates the posterior over $f(\tilde{\mathbf{x}}_*)$ as:

$$p(f(\tilde{\mathbf{x}}_*)|\mathbf{x}_*, \mathcal{D}_n) \approx N(\hat{\mu}_n(P_*), \hat{\sigma}_n^2(P_*)) \ . \tag{2.29}$$

This model is applied in Chapter 5 as a prior for a Bayesian optimisation algorithm to safely navigate a mobile robot over rough terrain under localisation uncertainty.

2.3 Bayesian optimisation

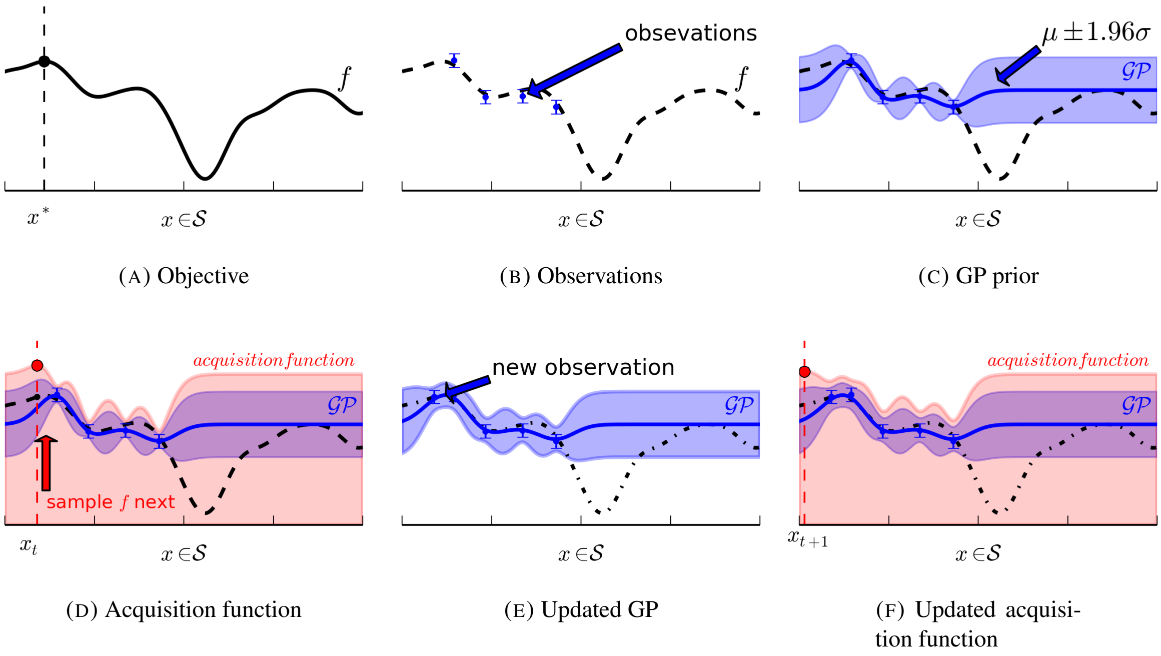
Consider the problem of searching for the global optimum of a function $f : \mathbb{R}^d \to \mathbb{R}$ within a given compact search space $\mathcal{S} \subset \mathbb{R}^d$, i.e. finding [2]:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{S}}{\operatorname{argmax}} f(\mathbf{x}) \ . \tag{2.30}$$

Assume that f is unknown to us and we only have access to noisy observations of its output $y = f(\mathbf{x}) + \nu$ with $\nu \sim N(0, \sigma_\nu^2)$. In addition, we are only allowed to sample the function up to n times.

Bayesian optimisation (Brochu et al., 2010) assumes that $f(\mathbf{x})$ is a random variable and applies a probabilistic model as a prior over f. Using a Gaussian process model, BO encodes prior assumptions about f via the mean m and the covariance function k of the GP.

[2] argmax corresponds to the set of arguments that maximise a function, which may contain multiple elements. The use of $\mathbf{x}^* = \operatorname{argmax}_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$, instead of $\mathbf{x}^* \in \operatorname{argmax}_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$, indicates the assumption that there is only one $\mathbf{x}^* \in \mathcal{S}$ for which $f(\mathbf{x}^*) = \max_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$.



(A) Objective

(B) Observations

(C) GP prior

(D) Acquisition function

(E) Updated GP

(F) Updated acquisition function

FIGURE 2.2: Illustration of the BO process: Consider an unknown objective function f and its global optimum x^* in a given search space $\mathcal{S}$ (a). Only noisy observations (b) are available. At each iteration t, BO fits a GP model over f with the current set of observations (c). Then BO maximises an acquisition function, which takes into account the GP model, to select the next query point x_t (d). The function is sampled at x_t, providing BO with a new observation to update the GP with (e). The algorithm then proceeds by selecting a new query point x_{t+1} with the updated acquisition function (f).

Rather than directly searching over f, BO uses an *acquisition function* $h(\mathbf{x})$ as a guide to sequentially select input locations at which to observe f. The acquisition function uses the information provided by the GP prior and the observations of f to estimate an utility value for sampling f at a given $\mathbf{x}$. So at each iteration t, BO queries the objective function f at the location of highest utility according to the acquisition function, considering the data observed so far $\mathcal{D}_{t-1}$, which is given by:

$$\mathbf{x}_t = \operatorname*{argmax}_{\mathbf{x} \in \mathcal{S}} h(\mathbf{x}|\mathcal{D}_{t-1}) \,. \tag{2.31}$$

After observing f at $\mathbf{x}_t$, BO updates the GP observations dataset with the new pair $(\mathbf{x}_t, y_t)$, which improves the belief about f. Then the algorithm proceeds to the next iteration, choosing an $\mathbf{x}_{t+1}$. The BO loop runs until a stopping criterion is satisfied, which is usually defined by a total number of iterations, n. At the end of this process, BO obtains a model that approximates the objective function $f|\mathcal{D}_n \sim \mathrm{GP}(\mu_n, \sigma_n^2)$ and an estimate of the optimum location $\mathbf{x}^*$, given by the best sample in the dataset. Algorithm 1 summarises BO in pseudo-code, and Figure 2.2 presents an illustration.

Algorithm 1: Bayesian optimisation

```
Input:
    $\mathcal{S}$: search space
    $n$: total number of iterations

for $t \in \{1, \dots, n\}$ do
    $\mathbf{x}_t = \operatorname*{argmax}_{\mathbf{x} \in \mathcal{S}} h(\mathbf{x}|\mathcal{D}_{t-1})$
    $y_t \leftarrow$ Sample $f(\mathbf{x}_t)$
    $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$
$t^* = \operatorname*{argmax}_{t \in \{1,\dots,n\}} \mu_n(\mathbf{x}_t)$
Result: $\mu_n, \sigma_n^2, \mathbf{x}_{t^*}$
```

2.3.1 Regret analysis

To derive theoretical guarantees for BO algorithms, a quantity that plays an important role in the theory is *regret*. BO can be classified as a multi-armed bandits algorithm (Srinivas et al., 2012). In the bandits literature (Arora et al., 2012), regret basically measures how good the algorithm's actions were when compared to optimal choices. The latter corresponds to the decision that an algorithm with perfect knowledge about the objective would make. There are many types of regret definitions available in the bandits literature (Arora et al., 2012). However, the work in this thesis is mainly concerned with *instant regret* and *cumulative regret*, which are defined as follows.

Consider a continuous function $f : \mathcal{S} \to \mathbb{R}$ defined over a compact set $\mathcal{S}$. At each round $t \in \mathbb{N}$, a maximisation algorithm over f selects query points $\mathbf{x}_t \in \mathcal{S}$. The instant regret suffered by the algorithm at round t is defined as:

$$r_t := \max_{\mathbf{x} \in \mathcal{S}} f(\mathbf{x}) - f(\mathbf{x}_t) \,, \tag{2.32}$$

while the cumulative regret after n rounds is defined as:

$$R_n := \sum_{t=1}^{n} r_t \,. \tag{2.33}$$

A desirable characteristic of an optimisation algorithm is to have no regret, at least asymptotically, i.e. $\lim\limits_{n\to\infty} \min_{t\leq n} r_t = 0$. With vanishing regret, as $n \to \infty$, we have that $\max_{t\leq n} f(\mathbf{x}_t) \to \max_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$. From the definitions above, one can see that theoretical results providing upper bounds for the cumulative regret of an optimisation algorithm also bound the instant regret, since $\min_{t\leq n} r_t \leq \frac{R_n}{n}$. Asymptotically vanishing regret then happens whenever the cumulative regret grows sub-linearly, yielding $\lim\limits_{n\to\infty} \frac{R_n}{n} = 0$.

These are the regret bounds usually sought after for BO algorithms, especially when based on upper confidence bound (UCB) approaches (Srinivas et al., 2012; Chowdhury and Gopalan, 2017). Other types of regret bounds include those based on the *simple regret*, i.e.:

$$s_t := \max_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) - \max_{i\leq t} f(\mathbf{x}_i) \,, \tag{2.34}$$

as derived for other BO approaches (Bull, 2011; Wang and Jegelka, 2017).

2.3.2 Acquisition functions

Acquisition functions define the selection criteria in determining BO queries. Sometimes they are also referred to as utility functions or search criteria. On one hand, these functions evaluate how useful collecting an observation at a given query location is for the optimisation process. On the other hand, they also encode criteria that quantify what is important for a query point to present, e.g. high predictive variance. The following presents the main types of acquisition function used throughout this thesis.

Upper confidence bound (UCB)

A commonly used acquisition function in the BO literature is the upper confidence bound (UCB). As proposed by Srinivas et al. (2012), the UCB utility is given by:

$$h_{\mathrm{UCB}}(\mathbf{x}|\mathcal{D}_{t-1}) := \mu_{t-1}(\mathbf{x}) + \beta_t\sigma_{t-1}(\mathbf{x}), \tag{2.35}$$

where $\beta_t > 0$ is a parameter of the algorithm and μ_{t-1} and σ_{t-1} are given by the GP posterior with the observations collected up to iteration $t-1$. Under certain assumptions, it is possible to properly set β_t as function of t to allow $h_{\mathrm{UCB}}(\mathbf{x})$ to behave as an upper bound for the value of $f(\mathbf{x})$ that is valid at every iteration and across the entire search space with high probability. In such settings, BO is able to attain asymptotic sub-linear regret bounds (Srinivas et al., 2012; Chowdhury and Gopalan, 2017). This parameter β_t controls the exploration-exploitation trade-off (Brochu et al., 2010), with higher values favouring areas of high uncertainty, while lower values favour areas of high mean value. Regret bounds are then guaranteed by setting β_t as a monotonically increasing function of t, ensuring continued exploration. In practice, however, β_t is usually set at a fixed value, which does not guarantee asymptotic regret bounds, but improves short-term performance.

Distance-penalised upper confidence bound (DUCB)

One example of acquisition function that can be applied to problems involving robotic navigation is the distance-penalised upper confidence bound (DUCB) (Marchant and Ramos, 2012):

$$h_{\mathrm{DUCB}}(\mathbf{x}|\mathcal{D}_{t-1}) := \mu_{t-1}(\mathbf{x}) + \beta\sigma_{t-1}(\mathbf{x}) - \zeta\|\mathbf{x} - \mathbf{x}_{t-1}\|_2 \ , \tag{2.36}$$

where $\|\mathbf{x} - \mathbf{x}_{t-1}\|_2$ corresponds to the Euclidean distance between the last sampled location and the candidate $\mathbf{x}$, μ_{t-1} and σ_{t-1} are given by the GP posterior with the observations in $\mathcal{D}_{t-1}$, and $\beta > 0$, $\zeta > 0$ are parameters to be set. β controls the exploration-exploitation trade-off, with higher values favouring areas of high uncertainty, while ζ penalises large jumps, allowing shorter paths between observations. In addition, notice that for minimisation objectives, DUCB can be applied by simply flipping the sign of the GP posterior mean to negative instead.

At the moment, there are no known regret bounds for BO algorithms based on DUCB in the literature. However, such bounds should be possible to derive. The distance-penalty term in Equation 2.36 defines the regret with respect to an adversary that remembers only the last action (i.e. $\mathbf{x}_{t-1}$) taken by the algorithm. Therefore, one should be able to apply the results in Arora et al. (2012) to derive DUCB algorithms with sub-linear regret bounds.

Expected improvement (EI)

Another classic type of acquisition function is the expected improvement (EI) (Bull, 2011). At iteration t, one can define $y_t^* := \max_{i<t} y_t$ as an optimal *incumbent*. The expected improvement is then defined as:

$$h_{\mathrm{EI}}(\mathbf{x}|\mathcal{D}_{t-1}) := \mathbb{E}[\max\{0, f(\mathbf{x}) - y_t^*\}\,|\mathcal{D}_{t-1}] \ . \tag{2.37}$$

In the case of a GP prior on f, the EI is given by:

$$h_{\mathrm{EI}}(\mathbf{x}|\mathcal{D}_{t-1}) = \begin{cases} (\mu_{t-1}(\mathbf{x}) - y_t^*)p(s_t; 0, 1) + \sigma_{t-1}(\mathbf{x})\varphi(s_t) \ , & \sigma_{t-1}(\mathbf{x}) > 0 \\ 0 \ , & \sigma_{t-1}(\mathbf{x}) = 0 \end{cases} \ , \tag{2.38}$$

where $s_t := \frac{\mu_{t-1}(\mathbf{x}) - y_t^*}{\sigma_{t-1}(\mathbf{x})}$, if $\sigma_{t-1}(\mathbf{x}) > 0$. Here $p(s_t; 0, 1)$ and $\varphi(s_t)$ denote, respectively, the cumulative density function and the probability density function of the standard normal distribution evaluated at s_t.

For the interested reader, theoretical results for EI are also available in the literature. Bull (2011), for instance, provides regret bounds for the optimisation of smooth functions with an EI-based algorithm.

2.3.3 Pure exploration

If one's objective is simply to learn a Gaussian process model, a valid strategy is to select $\mathbf{x}_t$ as:

$$\mathbf{x}_t = \operatorname*{argmax}_{\mathbf{x}\in\mathcal{S}} \sigma^2_{t-1}(\mathbf{x}) \,. \tag{2.39}$$

As discussed in Srinivas et al. (2012), this selection criterion is equivalent to choosing the location that provides the maximum amount of information about $f \sim \mathrm{GP}(m, k)$ given the previous observations. The information gain provided by n observations is given by the mutual information between the observations $\mathbf{y}_n$ and the corresponding $\mathbf{f}_n$ as:

$$I(\mathbf{y}_n; \mathbf{f}_n) = \frac{1}{2} \log |\mathbf{I} + \sigma_\nu^{-2} \mathbf{K}_n| \,, \tag{2.40}$$

which is upper-bounded by the maximum information gain, γ_n, defined as:

$$\gamma_n := \max_{\mathcal{Q}\subset\mathcal{X}: |\mathcal{Q}|=n} I(\mathbf{y}_\mathcal{Q}; \mathbf{f}_\mathcal{Q}) \,, \tag{2.41}$$

where $\mathbf{y}_\mathcal{Q} = \mathbf{f}_\mathcal{Q} + \boldsymbol{\nu}_\mathcal{Q}$, $\boldsymbol{\nu}_\mathcal{Q} \sim N(\mathbf{0}, \mathbf{I}\sigma_\nu^2)$, and $\mathbf{f}_\mathcal{Q} := [f(\mathbf{x})]_{\mathbf{x}\in\mathcal{Q}} \in \mathbb{R}^n$. The predictive variances and the information gain are related by (Srinivas et al., 2012, Lem. 5.3):

$$I(\mathbf{y}_n; \mathbf{f}_n) = \frac{1}{2} \sum_{t=1}^{n} \log(1 + \sigma_\nu^{-2} \sigma^2_{t-1}(\mathbf{x}_t)) \,. \tag{2.42}$$

Due to the sub-modularity of the information gain, following the strategy in Equation 2.39 then leads to a near-optimal information gain with $I(\mathbf{y}_n, \mathbf{f}_n) \geq (1 - 1/e)\gamma_n$. Besides that, general upper bounds for γ_n are available in the literature (see Srinivas et al., 2012, Theorem 5).

2.3.4 Practical considerations with model hyper-parameters

As discussed in Section 2.2.2, GP hyper-parameters are usually unknown a priori, so that one has to learn them from data. To account for uncertain hyper-parameters in the BO decision-making process, one can include the GP hyper-parameters $\boldsymbol{\theta}$ as latent variables in a joint distribution alongside the latent function:

$$p(f, \boldsymbol{\theta}|\mathcal{D}) = p(f|\mathcal{D}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) \,, \tag{2.43}$$

since the hyper-parameters do not depend on the latent function when conditioned on the data $\mathcal{D}$. Then one could work with the marginal posterior over f:

$$p(f|\mathcal{D}) = \int_{\boldsymbol{\theta}} p(f|\mathcal{D}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) \, \mathrm{d}\boldsymbol{\theta} \; . \tag{2.44}$$

In the BO context, the marginalisation above is applied by using an integrated acquisition function, as proposed by Snoek et al. (2012):

$$h(\mathbf{x}|\mathcal{D}_{t-1}) = \mathbb{E}_{\boldsymbol{\theta}}[h(\mathbf{x}|\mathcal{D}_{t-1}, \boldsymbol{\theta})] = \int_{\boldsymbol{\theta}} h(\mathbf{x}|\mathcal{D}_{t-1}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) \, \mathrm{d}\boldsymbol{\theta} \; . \tag{2.45}$$

However, the main difficulty in the former integral is in dealing with the hyper-parameters posterior:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} \; . \tag{2.46}$$

The evidence $p(\mathcal{D}) = \int p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}$ is usually intractable to compute, so that one has to apply numerical integration methods, such as Markov chain Monte Carlo (Murray and Adams, 2010), to solve Equation 2.45. A second approach is to predetermine the hyper-parameters via type-II maximum likelihood estimation (MLE):

$$\boldsymbol{\theta}^* \in \operatorname*{argmax}_{\boldsymbol{\theta}} p(\mathcal{D}_0|\boldsymbol{\theta}) \tag{2.47}$$

or via maximum a posteriori (MAP) estimation:

$$\boldsymbol{\theta}^* \in \operatorname*{argmax}_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) \; . \tag{2.48}$$

The posterior over the hyper-parameters is then replaced by a point-mass distribution on $\boldsymbol{\theta}^*$, so that:

$$p(f|\mathcal{D}) \approx p(f|\mathcal{D}, \boldsymbol{\theta}^*) \; . \tag{2.49}$$

The data for these estimates is usually the observations collected up to the current BO iteration or provided by an initial random sampling stage (McKay et al., 1979). Although computationally cheaper when compared to the full marginalisation approach, the main drawback in this case is that it is easy to over-fit the hyper-parameters to the data, introducing unwanted bias (Snoek et al., 2012). However, using point estimates for the hyper-parameters can still be a better approach than marginalisation if one considers online adaptation methods (Wang and de Freitas, 2014; Berkenkamp et al., 2019). The main idea behind adaptive methods is to anneal GP length-scales so that more complex objective functions are allowed by the GP prior as BO progresses. The practical effect is a gradual increase in exploration. For this thesis, the online adaptation method presented by Wang and de Freitas (2014) was applied in Chapter 4, while other chapters apply traditional MLE whenever hyper-parameters are unknown.

2.4 Reproducing kernel Hilbert spaces

Gaussian processes are directly related to the concept of a *reproducing kernel Hilbert space* (RKHS). In essence, a positive-definite covariance function also defines a so-called reproducing kernel and a unique Hilbert space associated with it. This section revises these concepts, which will play an important role in the derivation of models and theoretical results in the coming chapters. The section starts by revising the concept of linear spaces, also called vector spaces, which a Hilbert space is a subclass of. After that, reproducing kernels and RKHS's are defined in Section 2.4.2. The section closes in Section 2.4.3 with a few comments on the relationship between RKHSs and GPs. For a more in-depth review, among the abundant literature on the topic of RKHSs, readers are referred to Schölkopf and Smola (2002), Berlinet and Thomas-Agnan (2004) and Steinwart and Christmann (2008).

2.4.1 Linear spaces

As a single-sentence definition, a Hilbert space is an inner product space that is complete with respect to the norm induced by its inner product. Although Hilbert spaces can also be defined for vector-valued functions (Micchelli and Pontil, 2003), this section will be focusing on real, scalar-valued Hilbert spaces. The following briefly reviews some related concepts of mathematical spaces, which are necessary for a more complete definition of a Hilbert space and a RKHS. Books on functional analysis, such as Kreyszig (1978) and Lax (2002), provide further details on these topics.

Vector spaces: A vector space, also called a linear space, over a field $\mathbb{F}$ is basically a non-empty set $\mathcal{F}$ which is closed under the operations of addition $+ : \mathcal{F} \times \mathcal{F} \to \mathcal{F}$ and scalar multiplication $* : \mathbb{F} \times \mathcal{F} \to \mathcal{F}$. We say that a given set is *closed* under an operation if performing the operation on elements of the set always produces an element of the same set. For example, $\mathbb{R}^d,\ d \in \mathbb{N}$, the space of d-dimensional real-valued vectors, is a vector space. The space $\mathbb{R}^{\mathcal{X}}$ of all real-valued functions $f : \mathcal{X} \to \mathbb{R}$ on an arbitrary non-empty set $\mathcal{X}$ is also a vector space when equipped with the operations of addition and scalar multiplication defined by:

$$(f + g)(x) = f(x) + g(x) \ , \ \forall x \in \mathcal{X} \ , \tag{2.50}$$

$$(\alpha f)(x) = \alpha f(x) \ , \ \forall x \in \mathcal{X} \ , \tag{2.51}$$

for any $f, g \in \mathbb{R}^{\mathcal{X}}$ and any $\alpha \in \mathbb{R}$.

The following are definitions of other types of mathematical spaces related to the concept of vector spaces. For the axioms, consider the field of real numbers $\mathbb{F} = \mathbb{R}$, any $f, g, h \in \mathcal{F}$ and any $\alpha, \beta \in \mathbb{R}$.

Inner product spaces: An inner product space $(\mathcal{F}, \langle\cdot,\cdot\rangle_{\mathcal{F}})$ is a vector space $\mathcal{F}$ equipped with an additional structure defined by an inner product operator $\langle\cdot,\cdot\rangle_{\mathcal{F}} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$. The inner product operation satisfies the following axioms:

A1. Symmetry: $\langle f, g\rangle_{\mathcal{F}} = \langle g, f\rangle_{\mathcal{F}}$;

A2. Positive-definiteness: $\langle f, f\rangle_{\mathcal{F}} \geq 0$, $\langle f, f\rangle_{\mathcal{F}} = 0 \Longleftrightarrow f = 0$;

A3. Bilinearity: $\langle \alpha f + \beta g, h\rangle_{\mathcal{F}} = \langle \alpha f, h\rangle_{\mathcal{F}} + \langle \beta g, h\rangle_{\mathcal{F}} = \alpha\langle f, h\rangle_{\mathcal{F}} + \beta\langle g, h\rangle_{\mathcal{F}}$.

As a consequence of the inner product definition, we also have the following form of the *Cauchy-Schwartz* inequality:

$$|\langle f, g\rangle_{\mathcal{F}}| \leq \sqrt{\langle f, f\rangle_{\mathcal{F}}\langle g, g\rangle_{\mathcal{F}}}\ . \tag{2.52}$$

A simple example of inner product space is $\mathbb{R}^d$, $d \in \mathbb{N}$, equipped with the dot product operation:

$$\langle \mathbf{x}, \mathbf{y}\rangle_{\mathbb{R}^d} := \mathbf{x} \cdot \mathbf{y} := \sum_{i=1}^{d} x_i y_i\ ,\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d\ . \tag{2.53}$$

Normed vector spaces: In a vector space $\mathcal{F}$, we can define a norm $\|\cdot\|_{\mathcal{F}} : \mathcal{F} \to \mathbb{R}_0^+$, which is any function satisfying:

A1. Absolute homogeneity: $\|\alpha f\|_{\mathcal{F}} = |\alpha|\|f\|_{\mathcal{F}}$;

A2. Positive-definiteness: $\|f\|_{\mathcal{F}} \geq 0$, $\|f\|_{\mathcal{F}} = 0 \Longleftrightarrow f = 0$;

A3. Triangle inequality: $\|f + g\|_{\mathcal{F}} \leq \|f\|_{\mathcal{F}} + \|g\|_{\mathcal{F}}$.

For a given set $\mathcal{F}$ and norm $\|\cdot\|_{\mathcal{F}}$, the space $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$ is called a *normed vector space*.

Norms generalise the concept of length for 2- and 3-dimensional real vectors to arbitrary vector spaces. Examples of norms applied in this thesis are p-norms, such as:

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{d} x_i^p\right)^{\frac{1}{p}}\ ,\ \forall \mathbf{x} \in \mathbb{R}^d\ , \tag{2.54}$$

which are valid norms for $p \geq 1$, with $\|\mathbf{x}\|_\infty := \max_{i\in\{1,\dots,d\}} x_i$. Similarly, for real-valued functions over some domain $\mathcal{X} \subset \mathbb{R}^d$, i.e. $\mathcal{F} = \mathbb{R}^\mathcal{X}$, one can define:

$$\|f\|_p := \left(\int_\mathcal{X} |f(\mathbf{x})|^p \, \mathrm{d}\mathbf{x}\right)^{\frac{1}{p}} . \tag{2.55}$$

However, notice that this $\|\cdot\|_p$ may not satisfy the positive-definiteness axiom over the entire $\mathbb{R}^\mathcal{X}$. Any function $f \in \mathbb{R}^\mathcal{X}$ that is zero almost everywhere, except over a set of countably many points in $\mathcal{X}$, would still have $\|f\|_p = 0$, even though $f \neq 0$. In addition, any norm needs to be finite by definition, since $\infty \notin \mathbb{R}_0^+$. If we restrict ourselves to $\mathcal{F} = \mathcal{C}(\mathcal{X})$, the set of continuous real-valued functions on a compact $\mathcal{X} \subset \mathbb{R}^d$, we then have that $\|\cdot\|_p$ becomes a valid norm on elements of that set.

From an inner product, one can also straightforwardly define an induced norm by: $\|\cdot\|_\mathcal{F} := \sqrt{\langle\cdot,\cdot\rangle_\mathcal{F}}$. For instance, the 2-norm, as defined in Equation 2.54 with $p := 2$, is the norm induced by the inner product in $\mathbb{R}^d$ given by Equation 2.53.

Metric spaces. A metric on a set $\mathcal{F}$ is any function $\rho_\mathcal{F} : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_0^+$, where $\mathbb{R}_0^+$ denotes the set of non-negative real numbers, which satisfies the following axioms:

A1. Symmetry: $\rho_\mathcal{F}(f,g) = \rho_\mathcal{F}(g,f)$;

A2. Positive-definiteness: $\rho_\mathcal{F}(f,g) \geq 0$, $\rho_\mathcal{F}(f,g) = 0 \Longleftrightarrow f = g$;

A3. Triangle inequality: $\rho_\mathcal{F}(f,g) \leq \rho_\mathcal{F}(f,h) + \rho_\mathcal{F}(h,g)$.

A metric space $(\mathcal{F}, \rho_\mathcal{F})$ is simply a set $\mathcal{F}$ equipped with a metric $\rho_\mathcal{F}$ defined on it. Notice that $\mathcal{F}$ does not need to be a vector space for one to define a metric on it.

Both normed vector spaces and inner product spaces have metrics associated with them. For a given norm $\|\cdot\|_\mathcal{F}$, the induced metric is $\rho_\mathcal{F}(f,g) := \|f-g\|_\mathcal{F}$, $\forall f,g \in \mathcal{F}$. For instance, the 2-norm is the norm associated with the Euclidean distance metric:

$$\rho_{\mathbb{R}^d}(\mathbf{x},\mathbf{y}) := \|\mathbf{x}-\mathbf{y}\|_2 = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} = \sqrt{(\mathbf{x}-\mathbf{y})^\mathsf{T}(\mathbf{x}-\mathbf{y})} . \tag{2.56}$$

Therefore, the metric space defined by $\mathbb{R}^d$, $d \in \mathbb{N}$, and $\|\cdot\|_2$ is referred to as an *Euclidean vector space*. As the equation above still defines a valid metric if one scales the vectors by a positive-definite matrix $\mathbf{M}$, a non-Euclidean metric on $\mathbb{R}^d$ is the *Mahalanobis distance* and its associated norm, given by:

$$\|\mathbf{x}-\mathbf{y}\|_\mathbf{M} := \sqrt{(\mathbf{x}-\mathbf{y})^\mathsf{T}\mathbf{M}(\mathbf{x}-\mathbf{y})} . \tag{2.57}$$

This metric is the one applied in the definition of stationary covariance functions in Section 2.2.1.

Complete metric spaces: A metric space is complete if the limit of every Cauchy sequence in it is an element of the same space (Munkres, 1975). A sequence $\{f_i\}_{i\in\mathbb{N}}$ in a normed vector space $(\mathcal{F}, \|\cdot\|_{\mathcal{F}})$ is Cauchy if, and only if, for every $\epsilon > 0$, there is a $n_\epsilon \in \mathbb{N}$, such that:

$$\forall i, j > n_\epsilon,\ \|f_i - f_j\|_{\mathcal{F}} < \epsilon\ . \tag{2.58}$$

Without completeness it is not possible to ensure that the limit of any given convergent sequence is an element of the same metric space. That is why important concepts in calculus and functional analysis are only valid for complete metric spaces. As particular types of complete metric spaces, we have Banach spaces and Hilbert spaces. It is also known that every metric space has a completion (Kreyszig, 1978).

Banach spaces: A normed vector space that is complete with respect to the metric induced by its norm is called a Banach space.

Hilbert spaces: A Hilbert space is an inner product space that is complete with respect to the norm induced by its inner product.

From this last definition, we can see that any Hilbert space is also a Banach space with norm defined by the inner product. As a trivial example of a Hilbert space, we have Euclidean vector spaces $\mathbb{R}^d$. However, Hilbert spaces become a powerful tool in machine learning via the concept of a reproducing kernel, which is explained in the next section.

2.4.2 Reproducing kernels and their Hilbert spaces

We say that a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the reproducing kernel of a Hilbert space $\mathcal{H}$ composed of functions mapping $\mathcal{X}$ to $\mathbb{R}$, if and only if:

A1. $\forall x \in \mathcal{X},\ k(\cdot, x) \in \mathcal{H}$,
A2. $\forall x \in \mathcal{X},\ \forall f \in \mathcal{H},\ \langle f, k(\cdot, x)\rangle_{\mathcal{H}} = f(x)$.

Here $k(\cdot, x)$ denotes a function mapping $\mathcal{X}$ to $\mathbb{R}$ by $x' \mapsto k(x', x)$, $\forall x' \in \mathcal{X}$. The dot $(\cdot)$ notation in $k(\cdot, x)$ is used in place of a function argument to indicate that $k(\cdot, x)$ should be interpreted as a function of the that argument, the first argument in this case. Every Hilbert space which has a reproducing kernel is called a *reproducing kernel Hilbert space* (RKHS).

The relationship between positive-definite kernels and RKHSs is known to be two way. On the one hand, every Hilbert space of functions whose evaluation functional is continuous is a RKHS, having a unique reproducing kernel associated with it (Steinwart and Christmann, 2008, Thr. 4.20). Continuity of the evaluation functional, in this case, is equivalent to the same being bounded, i.e.:

$$\forall x \in \mathcal{X},\ \exists c_x < \infty : f(x) \leq c_x \|f\|_{\mathcal{H}},\ \forall f \in \mathcal{H}\ . \tag{2.59}$$

On the other hand, every symmetric, positive-definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the reproducing kernel of a unique RKHS (Berlinet and Thomas-Agnan, 2004; Steinwart and Christmann, 2008), which is discussed below.

The pre-Hilbert space: Given a positive-definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, one can define the set:

$$\mathcal{H}_k^0 := \operatorname{span}\{k(\cdot, x) \mid x \in \mathcal{X}\}\ , \tag{2.60}$$

where span denotes the linear span, i.e. the smallest linear space containing the set $\{k(\cdot, x) \mid x \in \mathcal{X}\}$, which is given by:

$$\operatorname{span}\{k(\cdot, x) \mid x \in \mathcal{X}\} = \left\{ \sum_{i=1}^{q} \alpha_i k(\cdot, x_i) \,\middle|\, q \in \mathbb{N}, \alpha_i \in \mathbb{R}, x_i \in \mathcal{X} \right\}\ . \tag{2.61}$$

The set $\mathcal{H}_k^0$ can then be equipped with an inner product structure by:

$$\forall f, f' \in \mathcal{H}_k^0, \quad \langle f, f' \rangle_k := \sum_{i=1}^{q} \sum_{j=1}^{q'} \alpha_i \alpha'_j k(x_i, x'_j)\ , \tag{2.62}$$

where $f := \sum_{i=1}^{q} \alpha_i k(\cdot, x_i) \in \mathcal{H}_k^0$ and $f' := \sum_{i=1}^{q'} \alpha'_i k(\cdot, x'_i) \in \mathcal{H}_k^0$. Equipping $\mathcal{H}_k^0$ with $\langle \cdot, \cdot \rangle_k$, we have that $(\mathcal{H}_k^0, \langle \cdot, \cdot \rangle_k)$ is an inner product space. However, $\mathcal{H}_k^0$ is not a Hilbert space, for it is not complete with respect to the norm $\|\cdot\|_k := \sqrt{\langle \cdot, \cdot \rangle_k}$. Including the limits of every Cauchy sequence in $\mathcal{H}_k^0$ then yields the set $\mathcal{H}_k$, which is the reproducing kernel Hilbert space associated with the kernel k. As $\mathcal{H}_k^0$ is isometrically embedded in $\mathcal{H}_k$ (Steinwart and Christmann, 2008, Thr. 4.21), the inner product $\langle \cdot, \cdot \rangle_k$ and the norm $\|\cdot\|_k$ are the same in $\mathcal{H}_k^0$ and $\mathcal{H}_k$. Therefore, $\mathcal{H}_k^0$ is referred to as a pre-Hilbert space (Berlinet and Thomas-Agnan, 2004).

Representing functions in a RKHS:. According to the definition of $\mathcal{H}_k^0$, any $f \in \mathcal{H}_k$ can be described as the limit of a Cauchy sequence $\{f_n\}_{n\in\mathbb{N}}$, $f_n \in \mathcal{H}_k^0$, converging in norm to f. Due to the continuity of the evaluation functional, we have that:

$$f, g \in \mathcal{H}_k, \quad \|f - g\|_k = 0 \Longleftrightarrow f = g \; . \tag{2.63}$$

To see that, one can consider that:

$$\forall x \in \mathcal{X}, |f(x) - g(x)| = |\langle f, k(\cdot, x)\rangle_k - \langle g, k(\cdot, x)\rangle_k| = |\langle f - g, k(\cdot, x)\rangle_k| \leq \sqrt{k(x,x)}\|f - g\|_k \; , \tag{2.64}$$

where the first equality comes from the reproducing property of k and the last inequality is by Cauchy-Schwartz, using $\|k(\cdot, x)\|_k = \sqrt{\langle k(\cdot, x), k(\cdot, x)\rangle_k} = \sqrt{k(x,x)} < \infty$. As a consequence, in an RKHS, convergence in norm implies point-wise convergence (Steinwart and Christmann, 2008, Sec. 4.2). By Equation 2.63, then we can write the given $f \in \mathcal{H}_k$ as:

$$f = \lim_{n\to\infty} f_n = \lim_{n\to\infty} \sum_{i=1}^{q_n} \alpha_{i,n} k(\cdot, x_{i,n}) \; , \tag{2.65}$$

where $\alpha_{i,n} \in \mathbb{R}$ and $x_{i,n} \in \mathcal{X}$, $\forall i \in \{1, \dots, q_n\}, n \in \mathbb{N}$. The drawback of using this representation for theoretical analysis is that, after performing an operation on the sequence elements f_n, it may be hard to prove that the sequence still converges to a function in $\mathcal{H}_k$.

Working with the pre-RKHS:. As an alternative, the pre-Hilbert space provides a useful framework to derive methods involving functions in the RKHS without having to deal with limits of Cauchy sequences. From the definition, $\mathcal{H}_k^0$ forms a *dense* subspace in $\mathcal{H}_k$, i.e. its closure $\overline{\mathcal{H}_k^0}$ is $\mathcal{H}_k$ itself. Consequently, any $f \in \mathcal{H}_k$ can be approximated by some $f^0 \in \mathcal{H}_k^0$ up to arbitrary precision, i.e.:

$$\forall f \in \mathcal{H}_k, \, \forall \epsilon > 0, \, \exists f^0 \in \mathcal{H}_k^0 : \quad \|f^0 - f\|_k \leq \epsilon \; . \tag{2.66}$$

The bounded linear extension theorem (Kreyszig, 1978, Thr. 2.7-11) is a well known result in functional analysis that allows extending linear operators defined on subspaces of normed vector spaces to their closure. In the RKHS case, any bounded linear map defined on $\mathcal{H}_k^0$ can be continuously extended to $\mathcal{H}_k$.

Feature maps. Given an arbitrary Hilbert space $\mathcal{H}$ and a feature map $\phi : \mathcal{X} \to \mathcal{H}$, mapping $\mathcal{X}$ into $\mathcal{H}$, we have that $k(x, x') := \langle \phi(x), \phi(x)\rangle_{\mathcal{H}}$, $\forall x, x' \in \mathcal{X}$ defines a reproducing kernel for the Hilbert space $\mathcal{H}_k$ of functions $f : \mathcal{X} \to \mathbb{R}$. Notice that the feature space $\mathcal{H}$ does not need to be $\mathcal{H}_k$ itself nor a function space. However, if $\mathcal{H} = \mathcal{H}_k$, ϕ is called the canonical feature map. In this case, it can be proved that ϕ is unique and given by $x \mapsto k(\cdot, x)$ (Steinwart and Christmann, 2008).

Restriction of a kernel. If $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel and $\mathcal{S} \subset \mathcal{X}$, the restriction of k to $\mathcal{S} \times \mathcal{S}$, denoted by $k|_{\mathcal{S}\times\mathcal{S}}$, is a kernel on $\mathcal{S}$ (Steinwart and Christmann, 2008, Lemma 4.3). Associated with $k|_{\mathcal{S}\times\mathcal{S}}$ is the RKHS $\mathcal{H}_k(\mathcal{S})$, which is the Hilbert space of functions $f : \mathcal{S} \to \mathbb{R}$ with norm given by (Aronszajn, 1950, p. 351):

$$\|f\|_{\mathcal{H}_k(\mathcal{S})} := \inf_{g \in \mathcal{H}_k : g|_{\mathcal{S}} = f} \|g\|_k \,. \tag{2.67}$$

2.4.3 RKHS's and Gaussian processes

The covariance function k of a Gaussian process defines a reproducing kernel and a corresponding RKHS $\mathcal{H}_k$. Therefore, all the covariance functions in Section 2.2.1 are reproducing kernels. One could also be led to think that a GP model defines a distribution over the function space given by $\mathcal{H}_k$. However, a function sampled from a GP is, in general, not an element of $\mathcal{H}_k$, as discussed in Wahba (1990, p. 5) and Rasmussen and Williams (2006, p. 131). On the other hand, the posterior mean of a zero-mean GP prior is always an element of $\mathcal{H}_k$. In this sense, a GP model can be applied to learn a function $f \in \mathcal{H}_k$ using a given set of observations $\mathcal{D}_n$ by approximating f with the GP posterior mean μ_n (Srinivas et al., 2012; Chowdhury and Gopalan, 2017).

2.5 Topology

Topology (Munkres, 1975) provides a common theoretical framework to analyse mathematical spaces and mappings between them based only on abstract concepts, such as neighbourhoods and continuity. This section provides a brief review of some of the topological concepts relevant to this thesis, such as continuity and compactness. For a better introduction to the topic, the reader is referred to any introductory book, such as Munkres (1975). In general, books on functional analysis (e.g. Kreyszig (1978)) and related areas also revise some of these concepts in a more applied way, connected to metric spaces, for example. Figure 2.3 presents a diagram summarising the relationships between most of the mathematical spaces described in this chapter.

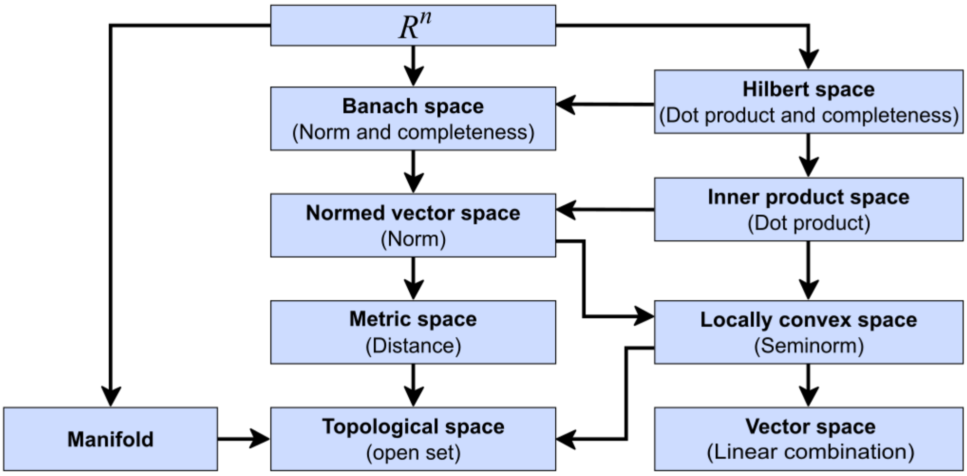


FIGURE 2.3: Diagram summarising the relationships between the different kinds of mathematical spaces reviewed in this background chapter (Source: Wikipedia.org).

2.5.1 Topological spaces

A topological space is the most general type of space in mathematics where a notion of neighbourhood can be defined. For example, any metric space is a topological space. The concept of neighbourhood is dependent on the definition of a topology, which is a collection of sets satisfying certain axioms. With a topology, one can define open and closed sets, which allow the definition of several important concepts, such as continuous functions, compactness, connectedness, etc., which are very relevant to other branches of mathematics.

Topology: Given an arbitrary non-empty set $\mathcal{X}$, a collection $\mathfrak{T}$ of subsets of $\mathcal{X}$ is called a topology if it satisfies the following axioms (Munkres, 1975, p. 76):

A1. $\varnothing$ and $\mathcal{X}$ are in $\mathfrak{T}$.

A2. The union of the elements of any sub-collection of $\mathfrak{T}$ is in $\mathfrak{T}$.

A3. The intersection of any finite sub-collection of $\mathfrak{T}$ is in $\mathfrak{T}$.

If these axioms are satisfied, the pair $(\mathcal{X}, \mathfrak{T})$ is called a topological space. It is common not to mention $\mathfrak{T}$ and refer to $\mathcal{X}$ as a topological space when confusion is unlikely to happen. As examples of topologies, we have:

E1. $\mathfrak{T} = \{\varnothing, \mathcal{X}\}$ is called the trivial topology, which is the smallest topology on $\mathcal{X}$.

E2. $\mathfrak{T} = 2^{\mathcal{X}}$, the power set of $\mathcal{X}$, i.e. the collection of every subset of $\mathcal{X}$, forms a topology on $\mathcal{X}$, which is called the discrete topology.

E3. Let $(\mathcal{X}, \rho)$ be a metric space. The default topology on $\mathcal{X}$ is induced by the metric ρ and is called the metric topology. Given $\delta > 0$, let $\mathcal{B}_\delta(x) := \{y \in \mathcal{X} \mid \rho(x, y) < \delta\}$ denote an open ball around a point $x \in \mathcal{X}$. A set $\mathcal{U} \subset \mathcal{X}$ is open in the metric topology if for every point $x \in \mathcal{U}$, there is a $\delta > 0$, such that $\mathcal{B}_\delta(x) \subset \mathcal{U}$.

When $\mathcal{X}$ is a vector space embedded with a topology, such as inner product spaces and normed spaces, $\mathcal{X}$ is commonly referred to as a *topological vector space*.

Open and closed sets: Given a topology $\mathfrak{T}$ on $\mathcal{X}$, any subset $\mathcal{U} \in \mathfrak{T}$ is called an *open* set, while an arbitrary subset $\mathcal{V} \subset \mathcal{X}$ is *closed* if its complement is in the topology, i.e. $\mathcal{X} \backslash \mathcal{V} \in \mathfrak{T}$. With this definition, given an arbitrary point $x \in \mathcal{X}$, any open set $\mathcal{U} \subset \mathcal{X}$ such that $x \in \mathcal{U}$ is called a *neighbourhood* of x.

Subspaces: Let $(\mathcal{X}, \mathfrak{T})$ be a topological space. For any $\mathcal{Y} \subset \mathcal{X}$, the collection:

$$\mathfrak{T}_{\mathcal{Y}} = \{\mathcal{Y} \cap \mathcal{U} \mid \mathcal{U} \in \mathfrak{T}\} \tag{2.68}$$

forms a topology on $\mathcal{Y}$ called the subspace topology. Then $\mathcal{Y}$ is called a subspace of $\mathcal{X}$.

2.5.2 Continuous functions

Continuity in calculus is usually dependent on differentiability. However, there are many other definitions of continuity, such as Lipschitz continuity, which do not imply differentiability and are still useful for analysis. Topology generalises the definition of continuity to its broadest sense.

Functions and their associated sets: Consider two arbitrary non-empty sets $\mathcal{X}$ and $\mathcal{Y}$. A function $f : \mathcal{X} \to \mathcal{Y}$ is any rule associating elements of $\mathcal{X}$ to elements of $\mathcal{Y}$, such that, for every $x \in \mathcal{X}$, there is exactly one $y \in \mathcal{Y}$ for which $f(x) = y$. Then $\mathcal{X}$ is called the *domain* of f, while $\mathcal{Y}$ is called its *co-domain*[3]. Given an arbitrary set $\mathcal{U} \subset \mathcal{X}$, the set:

$$f(\mathcal{U}) := \{f(x) \mid x \in \mathcal{U}\} \tag{2.69}$$

[3] Munkres (1975) refers to $\mathcal{Y}$ as the range of $f : \mathcal{X} \to \mathcal{Y}$. However, analysts usually define the range of a function, or an operator, as its image set, i.e. $f(\mathcal{X})$ (Lax, 2002). To avoid ambiguity, this thesis refers to $\mathcal{Y}$ as the co-domain of f.

is called the *image* of $\mathcal{U}$ under f. If $\mathcal{U} = \mathcal{X}$, $f(\mathcal{X})$ is simply called the image set of f. Similarly, we can define the *pre-image* of a set $\mathcal{V} \subset \mathcal{Y}$ under f as:

$$f^{-1}(\mathcal{V}) := \{x \in \mathcal{X} \mid f(x) \in \mathcal{V}\} \ , \tag{2.70}$$

which is the set of elements of $\mathcal{X}$ that are mapped to $\mathcal{V}$.

As an alternative notation, a function may also be defined using the symbol $\mapsto$ in terms of a formula or an explicit association rule, such as $f : x \mapsto x^2$, where $x \in \mathbb{R}$, defines the function $f : \mathbb{R} \to \mathbb{R}$, such that $f(x) = x^2$, $\forall x \in \mathbb{R}$. In general, given any $g : \mathcal{X} \to \mathcal{Y}$, the mapping $x \mapsto g(x)$, where $x \in \mathcal{X}$, defines a function mapping $\mathcal{X}$ into $\mathcal{Y}$.

Injective, surjective and bijective mappings: A function $f : \mathcal{X} \to \mathcal{Y}$ is said to be injective (or one-to-one) if images of distinct elements are distinct, i.e. $f(x) \neq f(y)$, for any $x, y \in \mathcal{X}$. The function is surjective (or onto) if its image set is the entire co-domain, i.e. $f(\mathcal{X}) = \mathcal{Y}$. Lastly, f is bijective if it is both injective and surjective.

Continuity: Let $(\mathcal{X}, \mathfrak{T}_\mathcal{X})$ and $(\mathcal{Y}, \mathfrak{T}_\mathcal{Y})$ be topological spaces. A function $f : \mathcal{X} \to \mathcal{Y}$ is said to be continuous if, for each open subset $\mathcal{V}$ of $\mathcal{Y}$, its pre-image is open in $\mathcal{X}$, i.e.:

$$\forall \mathcal{V} \in \mathfrak{T}_\mathcal{Y}, \quad f^{-1}(\mathcal{V}) \in \mathfrak{T}_\mathcal{X} \ . \tag{2.71}$$

Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be topological spaces. Regarding continuous functions, we have (Munkres, 1975):

(1) (Constants) Any $f : \mathcal{X} \to \mathcal{Y}$ that maps all of $\mathcal{X}$ to a single point in $\mathcal{Y}$ is continuous.

(2) (Compositions) If $f : \mathcal{X} \to \mathcal{Y}$ and $g : \mathcal{Y} \to \mathcal{Z}$ are continuous, their composition $g \circ f : \mathcal{X} \to \mathcal{Z}$ is continuous.

(3) (Mappings into products) Let $\{f_i : \mathcal{X} \to \mathcal{Y}\}_{i=1}^n$, $n \in \mathbb{N}$, be a set of continuous functions, then the function $f : \mathcal{X} \to \mathcal{Y}^n$, defined as $f(x) = [f_i(x)]_{i=1}^n$, $\forall x \in \mathcal{X}$, is continuous.

(4) (Pasting) Let $\mathcal{X} = \mathcal{U} \cup \mathcal{V}$, where $\mathcal{U}$ and $\mathcal{V}$ are closed in $\mathcal{X}$. If $f : \mathcal{V} \to \mathcal{Y}$ and $g : \mathcal{U} \to \mathcal{Y}$ are continuous, and $f(x) = g(x)$, $\forall x \in \mathcal{U} \cap \mathcal{V}$, then $h : \mathcal{X} \to \mathcal{Y}$, defined as $h(x) = f(x)$, if $x \in \mathcal{U}$, and $h(x) = g(x)$, if $x \in \mathcal{V}$, is continuous.

Observe that under the topological definition of continuity, any continuously-differentiable function as well as any Lipschitz continuous function is continuous. A Lipschitz continuous function is any function

$f : \mathcal{X} \to \mathcal{Y}$, where $(\mathcal{X}, \rho_\mathcal{X})$ and $(\mathcal{Y}, \rho_\mathcal{Y})$ are metrics spaces, such that:

$$\exists \ell_f \in \mathbb{R} : \rho_\mathcal{Y}(f(x), f(y)) \leq \ell_f \rho_\mathcal{X}(x, y), \quad \forall x, y \in \mathcal{X} \ . \tag{2.72}$$

2.5.3 Compact spaces

Compactness is an abstract concept in topology which allows deriving several useful results, such as the extreme value theorem (Munkres, 1975). There are different definitions of compactness, however, this thesis is mostly concerned with the one regarding metric spaces.

Compact metric spaces: A metric space $\mathcal{X}$ is said to be compact if every sequence in $\mathcal{X}$ has a convergent subsequence. A subset $\mathcal{U}$ of $\mathcal{X}$ is compact if it satisfies this condition when considered as a subspace of $\mathcal{X}$.

Properties of compact spaces: Consider an arbitrary metric space $\mathcal{X}$, and let $\mathcal{U} \subset \mathcal{X}$ be a compact subset of $\mathcal{X}$. As properties of compact spaces, we have (Munkres, 1975; Kreyszig, 1978):

P1. $\mathcal{U}$ is closed and bounded in $\mathcal{X}$.

P2. $\mathcal{U}$ is complete as a metric space.

P3. If $f : \mathcal{X} \to \mathcal{Y}$ is a continuous function and $\mathcal{Y}$ is a metric space, $f(\mathcal{U})$ is compact in $\mathcal{Y}$.

P4. If $f : \mathcal{X} \to \mathbb{R}$ is continuous, f attains a maximum and a minimum at some points in $\mathcal{U}$.

Notice that the definition of compactness implies that, if $\mathcal{U} \subset \mathcal{X}$ is compact, then $\mathcal{U}$ is closed and bounded in $\mathcal{X}$. However, the converse does not necessarily imply compactness, except when $\mathcal{X}$ is a finite-dimensional normed vector space, such as $\mathbb{R}^d$ (Kreyszig, 1978).

2.6 Measure theory

Probability, random variables and integration are intrinsically connected to measure theory, which provides a framework to formalise and analyse such concepts. This section briefly reviews some of the basics of measure theory connected to topics in this thesis, especially Chapter 6. For a more in-depth formal introduction to probability and measure theory, the reader is referred to books on the topic, such as Bauer (1981) and Durrett (2010). For vector-valued random variables, Li and Queffélec (2017) provides a formal introduction. Vector-valued integration is also in Mandrekar and Rüdiger (2015).

2.6.1 Measures, σ-algebras and measurable functions

Measures generalise physical measurement concepts such as mass, length and volume to arbitrary mathematical spaces. They are essentially functions which assign a number to a set.

σ-algebra: Given a non-empty set $\mathcal{X}$, a collection $\mathfrak{A}$ of subsets of $\mathcal{X}$ is called a σ-algebra (or sigma-algebra) if it meets the following requirements:

A1. $\mathcal{X} \in \mathfrak{A}$;

A2. $\mathcal{A} \in \mathfrak{A} \Longrightarrow \mathcal{X}\backslash\mathcal{A} \in \mathfrak{A}$

A3. for every sequence $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ of sets in $\mathfrak{A}$, $\bigcup_{n\in\mathbb{N}} \mathcal{A}_n \in \mathfrak{A}$.

Here $\mathcal{X}\backslash\mathcal{A}$ represents the complement of set $\mathcal{A}$ in $\mathcal{X}$. To exemplify, the smallest σ-algebra in a set $\mathcal{X}$ is simply the collection $\{\mathcal{X}, \varnothing\}$. The extreme opposite example is the power set, denoted as $2^{\mathcal{X}}$, which is the collection of all subsets of $\mathcal{X}$.

One can also define σ-algebras in terms of other sets and collections. Given a σ-algebra $\mathfrak{A}$ in $\mathcal{X}$ and any $\mathcal{U} \subset \mathcal{X}$, not necessarily in $\mathfrak{A}$, the collection:

$$\mathcal{U} \cap \mathfrak{A} := \{\mathcal{U} \cap \mathcal{V} \mid \mathcal{V} \in \mathfrak{A}\} \tag{2.73}$$

defines a σ-algebra in $\mathcal{U}$. Intersections of σ-algebras are also σ-algebras, so that, given any collection $\mathfrak{C}$ of subsets of $\mathcal{X}$, one can define $\mathfrak{A}(\mathfrak{C})$ as the intersection of all σ-algebras on $\mathcal{X}$ containing $\mathfrak{C}$. Then $\mathfrak{A}(\mathfrak{C})$ is the smallest σ-algebra on $\mathcal{X}$ containing $\mathfrak{C}$. With this definition, given a topological space $(\mathcal{X}, \mathfrak{T})$, one can define the Borel σ-algebra $\mathfrak{B}(\mathcal{X})$, which is the σ-algebra generated by the open sets of $\mathcal{X}$, i.e. $\mathfrak{B}(\mathcal{X}) = \mathfrak{A}(\mathfrak{T})$. It is common to assume this σ-algebra by default when dealing with topological spaces, such as metric spaces. For Euclidean vector spaces, i.e. $\mathcal{X} = \mathbb{R}^d$, $d \in \mathbb{N}$, the notation $\mathfrak{B}^d := \mathfrak{B}(\mathbb{R}^d)$ specifies the σ-algebra of Borel sets of $\mathbb{R}^d$ (Bauer, 1981).

Measure: A measure M is any set function defined on a σ-algebra $\mathfrak{A}$ and valued in $\mathbb{R} \cup \{-\infty, +\infty\}$ which satisfies the following axioms:

A1. $M[\varnothing] = 0$;

A2. $M[\mathcal{A}] \geq 0, \ \forall \mathcal{A} \in \mathfrak{A}$;

A3. for every sequence $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ of disjoint sets in $\mathfrak{A}$, $M\left[\bigcup_{n\in\mathbb{N}} \mathcal{A}_n\right] = \sum_{n\in\mathbb{N}} M[\mathcal{A}_n]$.

As a few examples of measures, we have Dirac measures, defined as:

$$x \in \mathcal{X},\ \forall \mathcal{A} \in \mathfrak{A},\ D_x[\mathcal{A}] = \begin{cases} 1, & x \in \mathcal{A} \\ 0, & x \notin \mathcal{A} \end{cases}, \tag{2.74}$$

counting measures, such as the cardinality of a set $C[\mathcal{A}] = |\mathcal{A}|$, i.e. the number of elements of $\mathcal{A}$, and probability measures.

Probability measures: Considering the previous settings, a probability measure is any measure on $\mathfrak{A}$ that satisfies the extra requirement that $P[\mathcal{X}] = 1$. Dirac measures, for example, are probability measures. Discrete probability measures can be directly formulated from them as weighted sums $P[\mathcal{A}] = \sum_{i=1}^{n} p_i D_{x_i}[\mathcal{A}],\ \forall \mathcal{A} \in \mathfrak{A}$, given that $\sum_{i=1}^{n} p_i = 1$, $p_i \geq 0$ and $\{x_i\}_{i=1}^{n} \subset \mathcal{X}$.

Related spaces: Given a set $\mathcal{X}$ and any σ-algebra $\mathfrak{A}$ of subsets of $\mathcal{X}$, the pair $(\mathcal{X}, \mathfrak{A})$ is called a *measurable space*. Adding a measure M on $\mathfrak{A}$, the triplet $(\mathcal{X}, \mathfrak{A}, M)$ is called a *measure space*. If the given measure is a probability measure, the space $(\mathcal{X}, \mathfrak{A}, P)$ is called a *probability space*. In this case, the set $\mathcal{X}$ is usually referred to as the event (or sample) space, $\mathfrak{A}$ is the σ-algebra of events in $\mathcal{X}$, and the elements $\omega \in \mathcal{X}$ are called elementary events.

Measurable functions: Given two measure spaces $(\mathcal{X}, \mathfrak{A})$ and $(\mathcal{X}', \mathfrak{A}')$, a function $f : \mathcal{X} \to \mathcal{X}'$ is said to be $\mathfrak{A}$-$\mathfrak{A}'$-measurable if:

$$f^{-1}(\mathcal{A}') \in \mathfrak{A},\ \forall \mathcal{A}' \in \mathfrak{A}' , \tag{2.75}$$

where $f^{-1}(\mathcal{A}')$ denotes the pre-image of set $\mathcal{A}'$ under f. If f is a real-valued function, i.e. $\mathcal{X}' = \mathbb{R}$, and $\mathfrak{A}'$ is the σ-algebra of Borel sets in $\mathbb{R}$, i.e. $\mathfrak{A}' = \mathfrak{B}(\mathbb{R})$, one can simply say that f is $\mathfrak{A}$-measurable. Related to measurable functions, we have the following properties and examples (Bauer, 1981):

(1) Every constant-valued mapping $f : \mathcal{X} \to \mathcal{X}'$ is $\mathfrak{A}$-$\mathfrak{A}'$-measurable.

(2) If $\mathcal{X}$ and $\mathcal{X}'$ are topological spaces, every continuous function $f : \mathcal{X} \to \mathcal{X}'$ is Borel-measurable, i.e. f is $\mathfrak{B}(\mathcal{X})$-$\mathfrak{B}(\mathcal{X}')$-measurable.

(3) If $f_1 : (\mathcal{X}_1, \mathfrak{A}_1) \to (\mathcal{X}_2, \mathfrak{A}_2)$ and $f_2 : (\mathcal{X}_2, \mathfrak{A}_2) \to (\mathcal{X}_3, \mathfrak{A}_3)$ are measurable functions, then the composed function $f_1 \circ f_2$ is $\mathfrak{A}_1$-$\mathfrak{A}_3$-measurable.

σ-algebra generated by a function: Consider a set of functions $\{f_i\}_{i\in\mathcal{I}}^n$, where $f_i : \mathcal{X} \to \mathcal{X}_i'$ and $(\mathcal{X}_i', \mathfrak{A}_i')$ is a measurable space, $i \in \mathcal{I}$, we have that any:

$$\mathfrak{A}(f_i) := f_i^{-1}(\mathfrak{A}_i') = \{f_i^{-1}(\mathcal{A}')|\mathcal{A}' \in \mathfrak{A}_i'\} , \tag{2.76}$$

defines a σ-algebra in $\mathcal{X}$. In particular, $\mathfrak{A}(f_i)$ is the smallest σ-algebra in $\mathcal{X}$ with respect to which f_i is measurable. Furthermore, we have that:

$$\mathfrak{A}(f_i; i \in \mathcal{I}) := \mathfrak{A}\left(\bigcup_{i\in\mathcal{I}} f_i^{-1}(\mathfrak{A}_i')\right) \tag{2.77}$$

is also a σ-algebra and is called the σ-algebra generated by the mappings f_i (and their respective measurable spaces $(\mathcal{X}_i', \mathfrak{A}_i')$).

Almost surely (a.s.) and almost everywhere (a.e.): These two terms are common in probability and measure theory. The first term concerns events in a probability space, while the second refers to functions on any measurable space. Consider a probability space $(\mathcal{X}, \mathfrak{A}, P)$. An event $\mathcal{A} \in \mathfrak{A}$ is said to happen almost surely (a.s.) if $P[\mathcal{A}] = 1$. Two $\mathfrak{A}$-measurable functions f and g agree almost everywhere (a.e.), if the set where they disagree has measure zero, i.e. $P[\{x \in \mathcal{X} \mid f(x) \neq g(x)\}] = 0$. As an example consider $\mathcal{X} := \mathbb{R}^d$, any σ-algebra $\mathfrak{A}$ on $\mathbb{R}^d$, and the function $f(\mathbf{x}) := \mathbf{a} \cdot \mathbf{x}$, $\mathbf{a} \in \mathbb{R}^d$. With respect to the Dirac measure at $\mathbf{0}$, $P := D_{\mathbf{0}}$, one can say that, given $\mathbf{x} \in \mathcal{X}$, $f(\mathbf{x}) = 0$ almost surely. Although f itself is not zero on most of the domain (for $\mathbf{a} \neq \mathbf{0}$), we have that $D_{\mathbf{0}}[\{\mathbf{x} \in \mathcal{X} \mid f(\mathbf{x}) = 0\}] = 1$, regardless. In this same example, one can also say that f is zero almost everywhere, i.e. $f = 0$ (a.e.).

2.6.2 Integration with respect to a measure

Measures allow generalising the concept of integration to arbitrary domains. For this thesis, an important characteristic is that measures allow unifying integration with respect to both discrete and continuous probability measures into a single operation.

To define the general concept of integration with respect to a measure, let us first consider scalar real-valued functions and then we can extend it to the case of vector-valued functions. We will first need to review a few fundamental concepts before proceeding. For the definitions, we will consider an arbitrary measurable space $(\mathcal{X}, \mathfrak{A})$ and any measure M on $\mathfrak{A}$. As a building block for the next steps, an elementary (or simple) function is any function $u : \mathcal{X} \to \mathbb{R}_0^+$, which is $\mathfrak{A}$-$\mathfrak{B}^1$-measurable, or simply $\mathfrak{A}$-measurable, and which can be described as:

$$u = \sum_{i=1}^{n} \alpha_i \mathbb{1}_{\mathcal{A}_i} \,, \tag{2.78}$$

where $n \in \mathbb{N}$, $\alpha_1 \ldots, \alpha_n \in \mathbb{R}_0^+$, $\{\mathcal{A}_i\}_{i=1}^n \subset \mathcal{X}$ is a collection of pairwise disjoint subsets of $\mathcal{X}$, and $\mathbb{1}_{\mathcal{A}_i}$ are indicator functions, i.e.:

$$\mathcal{A} \subset \mathcal{X}, \forall x \in \mathcal{X}, \quad \mathbb{1}_{\mathcal{A}}(x) := \begin{cases} 1, & x \in \mathcal{A} \\ 0, & x \notin \mathcal{A} \,. \end{cases} \tag{2.79}$$

Given a measure M on $\mathfrak{A}$, the integral of u with respect to M, or the M-integral of u, is then defined as:

$$\int u \, \mathrm{d}M := \sum_{i=1}^{n} \alpha_i M[\mathcal{A}_i] \,. \tag{2.80}$$

Now given any $\mathfrak{A}$-measurable non-negative function f, one can show that there exists an increasing sequence $\{u_n\}_{n\in\mathbb{N}}$ of elementary functions such that $f = \sup_n u_n$ (Bauer, 1981, Thr. 2.3.6). Then the M-integral of f over $\mathcal{X}$ is defined as:

$$\int f \, \mathrm{d}M := \sup_{n\in\mathbb{N}} \int u_n \, \mathrm{d}M \,. \tag{2.81}$$

Finally, the integral of any $\mathfrak{A}$-measurable function f can be defined in terms of the integral of the positive $f^+ := \sup(f, 0)$ and the negative $f^- := -\inf(f, 0)$ parts of f. A function f is (M-)integrable if it is $\mathfrak{A}$-measurable and both $\int f^+ \, \mathrm{d}M$ and $\int f^- \, \mathrm{d}M$ are finite. Then the M-integral of f is simply:

$$\int f \, \mathrm{d}M := \int f^+ \, \mathrm{d}M - \int f^- \, \mathrm{d}M \,. \tag{2.82}$$

Notice that this integral is over the entire $\mathcal{X}$. Given a set $\mathcal{A} \in \mathfrak{A}$, the integral of f over $\mathcal{A}$ is defined as:

$$\int_{\mathcal{A}} f \, \mathrm{d}M := \int f \mathbb{1}_{\mathcal{A}} \, \mathrm{d}M \tag{2.83}$$

To emphasise the dependence on the inputs $x \in \mathcal{X}$, one may also equivalently write:

$$\int_{\mathcal{A}} f(x) \, \mathrm{d}M(x) := \int_{\mathcal{A}} f \, \mathrm{d}M \,. \tag{2.84}$$

Properties: Let f and g be M-integrable functions. The following are properties of the integral defined above:

P1. Linearity: $\forall \alpha, \beta \in \mathbb{R}$, $\int \alpha f + \beta g \, \mathrm{d}M = \alpha \int f \, \mathrm{d}M + \beta \int g \, \mathrm{d}M$;

P2. Monotonicity: $f \leq g$ (a.e.) $\Longrightarrow$ $\int f \, \mathrm{d}M \leq \int g \, \mathrm{d}M$;

P3. $f = 0$ (a.e.) $\iff \int f \, \mathrm{d}M = 0$.

As an example of integration with respect to a measure, consider a discrete probability measure $P := \sum_{i=1}^{n} p_i D_{x_i}$, where $\sum_{i=1}^{n} p_i = 1$, $p_i \geq 0, \forall i \in \{1, \ldots, n\}$, defined on $(\mathcal{X}, \mathfrak{A})$. The P-integral of an integrable function $f : \mathcal{X} \to \mathbb{R}$ is then given by:

$$\int f \, \mathrm{d}P = \sum_{i=1}^{n} p_i f(x_i) \,, \tag{2.85}$$

which is equivalent to the expected value of f under P, discussed in the next section. However, if P was a continuous probability measure, one may need to define a density to compute the integral, which is discussed below.

Densities: For every non-negative measurable function f on $(\mathcal{X}, \mathfrak{A}, M)$, the map:

$$\mathcal{U} \mapsto \int_{\mathcal{U}} f \, \mathrm{d}M \,, \tag{2.86}$$

where $\mathcal{U} \in \mathfrak{A}$, defines a measure V on $\mathfrak{A}$ (Bauer, 1981, Thr. 2.9.1). Then the function f is called the *density* of V relative to M. In this case, one may write:

$$f = \frac{\mathrm{d}V}{\mathrm{d}M} \,, \tag{2.87}$$

which is a Radon-Nikodym derivative (Durrett, 2010). If $(\mathcal{X}, \mathfrak{A}) = (\mathbb{R}^d, \mathfrak{B}^d)$, $d \in \mathbb{N}$, and M is the Lebesgue measure on $\mathbb{R}^d$, densities of continuous probability measures P on $\mathcal{X}$ relative to the Lebesgue measure are simply the well-known *probability density functions*. Then the P-integral of a function $g : \mathbb{R}^d \to \mathbb{R}$ is simply given by:

$$\int g \, \mathrm{d}P = \int_{\mathbb{R}^d} g(\mathbf{x}) p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \,, \tag{2.88}$$

where $p = \frac{\mathrm{d}P}{\mathrm{d}\mathbf{x}}$ is the density of P relative to the Lebesgue measure on $\mathbb{R}^d$, i.e. the probability density function of P.

Lebesgue measure. The Lebesgue measure L^d is a measure on the σ-algebra $\mathfrak{B}^d$ of Borel sets of $\mathbb{R}^d$, which connects the concept of integration with respect to a measure back to the conventional (Riemann) integrals. Given a L^d-measurable real valued function, we have the equivalence:

$$\int_{\mathcal{A}} f \, \mathrm{d}L^d = \int_{\mathcal{A}} f(x) \, \mathrm{d}L^d(x) = \int_{\mathcal{A}} f(x) \, \mathrm{d}x \,, \ \forall \mathcal{A} \in \mathfrak{B}^d \,. \tag{2.89}$$

As seen, the Lebesgue measure corresponds to the measures of length, area and volume from 1-, 2- and 3-dimensional Euclidean spaces, generalised to arbitrary d-dimensional Euclidean spaces.

Vector-valued integrals

The previous presentation covered the case of integrals of real-valued functions with respect to measures. This section briefly extends integration to the vector-valued case. The extension is trivial in the finite-dimensional case. However, when functions are valued in a infinite-dimensional space, a special treatment is needed and different concepts of integral exist (see Mandrekar and Rüdiger, 2015, Ch. 2).

The finite-dimensional case: Let $\mathbf{f} : \mathcal{X} \to \mathbb{R}^d$, $d \in \mathbb{N}$, be a vector-valued function. If $\mathbf{f} := [f_1, \ldots, f_d]^{\mathsf{T}}$ is such that every $f_i : \mathcal{X} \to \mathbb{R}$, $i \in \{1, \ldots, d\}$, is M-integrable on $(\mathcal{X}, \mathfrak{A}, M)$, the M-integral of $\mathbf{f}$ is defined (Bauer, 1981, p. 255) as the vector:

$$\int \mathbf{f}\,\mathrm{d}M := \left[\int f_1\,\mathrm{d}M, \ldots, \int f_d\,\mathrm{d}M\right]^{\mathsf{T}} . \tag{2.90}$$

The Bochner integral: Consider an arbitrary measure space $(\mathcal{X}, \mathfrak{A}, M)$ and a Banach space $\mathcal{F}$ of arbitrary dimension, not necessarily finite. A function $\mathbf{f} : \mathcal{X} \to \mathcal{F}$ is said to be Bochner-integrable with respect to M if $\mathbf{f}$ is $\mathfrak{A}$-measurable and if $x \mapsto \|\mathbf{f}(x)\|_{\mathcal{F}}$ is M-integrable, i.e. $\int \|\mathbf{f}\|_{\mathcal{F}}\,\mathrm{d}M < \infty$. Denote by $\mathcal{L}^1_{\mathcal{F}}(M)$ the space containing all Bochner-integrable functions on $(\mathcal{X}, \mathfrak{A}, M)$. As in the real-valued case, let us first consider simple functions. In the Banach space case, a simple function is any function $\mathbf{u} : \mathcal{X} \to \mathcal{F}$ that can be described as $\mathbf{u} = \sum_{i=1}^{n} \mathbf{u}_i \mathbb{1}_{\mathcal{A}_i}$, where $\mathbf{u}_i \in \mathcal{F}$ and $\mathcal{A}_i \in \mathfrak{A}$ are mutually disjoint sets, i.e. $i \neq j \Longrightarrow \mathcal{A}_i \cap \mathcal{A}_j = \varnothing$, with $M[\mathcal{A}_i] < \infty$. The Bochner integral of $\mathbf{u}$ with respect to M is then defined as:

$$\int \mathbf{u}\,\mathrm{d}M := \sum_{i=1}^{n} \mathbf{u}_i M[\mathcal{A}_i] \in \mathcal{F} . \tag{2.91}$$

One can prove that $\mathbf{u} \mapsto \int \mathbf{u}\,\mathrm{d}M$ defines a bounded linear operator on the space of simple functions, which is dense in $\mathcal{L}^1_{\mathcal{F}}(M)$, and is therefore extensible to the entire space $\mathcal{L}^1_{\mathcal{F}}(M)$ of Bochner-integrable functions on $(\mathcal{X}, \mathfrak{A}, M)$ (Dinculeanu, 2000, p. 15). Given a function $\mathbf{f} \in \mathcal{L}^1_{\mathcal{F}}(M)$, the function $\int \mathbf{f}\,\mathrm{d}M \in \mathcal{F}$ is called the Bochner integral of $\mathbf{f}$ with respect to M. When the Banach space $\mathcal{F}$ is the finite-dimensional Euclidean space $\mathbb{R}^d$, $d \in \mathbb{N}$, the Bochner integral is equivalent to the integral presented before in Equation 2.90.

As in the real-valued case, the restriction of the Bochner integral to a measurable subset $\mathcal{A} \in \mathfrak{A}$ is given by:

$$\int_{\mathcal{A}} \mathbf{f} \, \mathrm{d}M := \int \mathbf{f} \mathbb{1}_{\mathcal{A}} \, \mathrm{d}M \ . \tag{2.92}$$

Among the properties of the Bochner integral, we have that the integral is linear and bounded by:

$$\left\| \int \mathbf{f} \, \mathrm{d}M \right\|_{\mathcal{F}} \leq \int \|\mathbf{f}\|_{\mathcal{F}} \, \mathrm{d}M < \infty \ . \tag{2.93}$$

Given another Banach space $\mathcal{G}$ and a continuous linear operator $\mathbf{L} : \mathcal{F} \to \mathcal{G}$, we also have that:

$$\int \mathbf{L}\mathbf{f} \, \mathrm{d}M = \mathbf{L} \int \mathbf{f} \, \mathrm{d}M \in \mathcal{G} \ . \tag{2.94}$$

2.6.3 Random variables

Using the tools from measure theory, the concept of a random variable can be generalised and formalised in a solid theoretical framework. Given a probability space $(\mathcal{E}, \mathfrak{E}, P)$ and a measurable space $(\mathcal{E}', \mathfrak{E}')$, a random variable is any function $\xi : \mathcal{E} \to \mathcal{E}'$ that is $\mathfrak{E}$-$\mathfrak{E}'$-measurable. In this case, ξ is also called a $\mathcal{E}'$-valued random variable or a $(\mathcal{E}', \mathfrak{E}')$-random variable on $(\mathcal{E}, \mathfrak{E}, P)$. For any $\omega \in \mathcal{E}$, $\xi(\omega)$ is called a realisation of ξ.

Distribution: Let $\mathbb{P}\{\xi \in \mathcal{A}'\} := P[\xi^{-1}(\mathcal{A}')],\ \mathcal{A}' \in \mathfrak{E}'$, denote the probability of the event ξ *lies in* $\mathcal{A}'$, i.e. the event that ξ is realised in $\mathcal{A}'$. The mapping $\mathcal{A}' \to \mathbb{P}\{\xi \in \mathcal{A}'\}$ is the image measure of P under ξ, denoted as P_ξ, which defines a measure on $(\mathcal{E}', \mathfrak{E}')$. P_ξ is referred to as the distribution of ξ with respect to P, so that one may write $\xi \sim P_\xi$.

Expectation: The expected value of a real-valued integrable random variable ξ on $(\mathcal{E}, \mathfrak{E}, P)$ is defined as:

$$\mathbb{E}[\xi] = \mathbb{E}_P[\xi] := \int \xi \, \mathrm{d}P \ . \tag{2.95}$$

Using the image measure P_ξ of ξ, the expected value of a real-valued function f of ξ is given by:

$$\mathbb{E}[f(\xi)] = \mathbb{E}_{P_\xi}[f] = \int f \, \mathrm{d}P_\xi \ . \tag{2.96}$$

Setting $f(x) = x, \forall x \in \mathbb{R}$, we also have that $\mathbb{E}[\xi] = \int_{\mathcal{E}} x \, \mathrm{d}P_\xi(x)$. Given a σ-sub-algebra $\mathfrak{Z} \subset \mathfrak{Z}$, ξ may not be measurable with respect to $\mathfrak{Z}$, but one can still compute expected values of ξ for every event set in $\mathfrak{Z}$. Define $\mathbb{E}[\xi|\mathfrak{Z}]$, the conditional expectation of ξ given $\mathfrak{Z}$. $\mathbb{E}[\xi|\mathfrak{Z}]$ is a $\mathfrak{Z}$-measurable random variable, such that:

$$\int_{\mathcal{Z}} \mathbb{E}[\xi|\mathfrak{Z}]\,\mathrm{d}P = \int_{\mathcal{Z}} \xi\,\mathrm{d}P,\ \forall \mathcal{Z} \in \mathfrak{Z}\ , \tag{2.97}$$

which means that every realisation of $\mathbb{E}[\xi|\mathfrak{Z}]$ is an expected value of ξ given an event $\mathcal{Z} \in \mathfrak{Z}$.

Considering a probability space $(\mathcal{E}, \mathfrak{E}, P)$, σ-sub-algebras $\mathfrak{Z} \subset \mathfrak{E}$ and $\mathfrak{U} \subset \mathfrak{E}$, and integrable real-valued random variables ξ and χ, the following are properties of the conditional expectation (Bauer, 1981, Sec. 10.1):

P1. $\mathbb{E}[\mathbb{E}[\xi|\mathfrak{Z}]] = \mathbb{E}[\xi]$;
P2. ξ is $\mathfrak{Z}$-measurable $\implies \mathbb{E}[\xi|\mathfrak{Z}] = \xi$ (a.s.);
P3. $\xi = \chi$ (a.s.) $\implies \mathbb{E}[\xi|\mathfrak{Z}] = \mathbb{E}[\chi|\mathfrak{Z}]$ (a.s.);
P4. $c \in \mathbb{R}$, $\xi = c$ (a.s.) $\implies \mathbb{E}[\xi|\mathfrak{Z}] = c$ (a.s.);
P5. $\forall \alpha, \beta \in \mathbb{R}$, $\mathbb{E}[\alpha\xi + \beta\chi|\mathfrak{Z}] = \alpha\mathbb{E}[\xi|\mathfrak{Z}] + \beta\mathbb{E}[\chi|\mathfrak{Z}]$ (a.s.);
P6. $\xi \leq \chi$ (a.s.) $\implies \mathbb{E}[\xi|\mathfrak{Z}] \leq \mathbb{E}[\chi|\mathfrak{Z}]$ (a.s.);
P7. ξ is $\mathfrak{Z}$-measurable $\implies \mathbb{E}[\xi\chi|\mathfrak{Z}] = \xi\mathbb{E}[\chi|\mathfrak{Z}]$ (a.s.);
P8. $\mathfrak{U} \subset \mathfrak{Z} \subset \mathfrak{E} \implies \mathbb{E}[\mathbb{E}[\xi|\mathfrak{Z}]|\mathfrak{U}] = \mathbb{E}[\mathbb{E}[\xi|\mathfrak{U}]|\mathfrak{Z}] = \mathbb{E}[\xi|\mathfrak{U}]$ (a.s.).

The extension of the concept of expectation to vector-valued random variables $\boldsymbol{\xi} : \mathcal{E} \to \mathcal{F}$, where $\mathcal{F}$ is a Banach space, follows via Bochner integration (see Section 2.6.2) and other types of integration, which may depend on weaker assumptions, such as the Pettis integral (Mandrekar and Rüdiger, 2015).

2.6.4 Stochastic processes

This section reviews some measure-theoretic concepts related to stochastic processes, which are formally defined below. Gaussian processes, as introduced in Section 2.2, are one example of such processes. The concept of a stochastic process, however, is abstract enough to analyse other types of processes, such as data streams and algorithmic choices. In the case of Bayesian optimisation, for example, Chowdhury and Gopalan (2017) treat both the algorithm's queries $\{\mathbf{x}_t\}_{t=1}^{\infty}$ and the observation noise $\{\nu_t\}_{t=1}^{\infty}$ sequences as stochastic processes.

This section starts by formally defining stochastic processes, and then introducing the related concepts of filtrations, martingales and predictability of random sequences, which are important for the theoretical results in Chapter 6. For the definitions below, consider an arbitrary probability space $(\mathcal{E}, \mathfrak{E}, P)$ and a non-empty set $\mathcal{I}$.

Processes: Any quadruple $(\mathcal{E}, \mathfrak{E}, P, \{\xi_t\}_{t\in\mathcal{I}})$, where $\{\xi_t\}_{t\in\mathcal{I}}$ is a family of random variables on $(\mathcal{E}, \mathfrak{E})$ with values all in the same measurable space $(\mathcal{W}, \mathfrak{W})$, is called a stochastic process. The space $(\mathcal{W}, \mathfrak{W})$ is referred to as the state space of the stochastic process, while the set $\mathcal{I}$ is called index set, parameter set or time set[4]. For every $\omega \in \mathcal{E}$, the mapping $t \mapsto \xi_t(\omega)$ can be referred to as a path, trajectory or realisation of the process. If $(\mathcal{E}, \mathfrak{E}, P)$ is already given or implicit, one may simply refer to $\{\xi_t\}_{t\in\mathcal{I}}$ as a stochastic process (Bauer, 1981).

Filtrations: Let $\mathcal{I}$ be a partially ordered set. A family $\{\mathfrak{F}_t\}_{t\in\mathcal{I}}$ of σ-sub-algebras of $\mathfrak{E}$ is called a filtration in $\mathfrak{E}$ if:

$$\forall i, j \in \mathcal{I}: \quad i \leq j \Longrightarrow \mathfrak{F}_i \subset \mathfrak{F}_j \; . \tag{2.98}$$

Given a stochastic process $\{\xi_t\}_{t\in\mathcal{I}}$, we say that $\{\xi_t\}_{t\in\mathcal{I}}$ is *adapted* to a filtration $\{\mathfrak{F}_t\}_{t\in\mathcal{I}}$ if every ξ_t is $\mathfrak{F}_t$-$\mathfrak{E}'$-measurable. As an example, if $\mathcal{I} = \mathbb{N}$, for any $t \in \mathbb{N}$, we have that $\mathfrak{F}_t^0 := \mathfrak{A}(\xi_i; i \in \{1, \ldots, t\})$ defines the smallest σ-algebra with respect to which all $\{\xi_i\}_{i=1}^t$ are measurable (see Equation 2.77). Therefore, $\{\xi_i\}_{i=1}^t$ is adapted to the filtration $\{\mathfrak{F}_t^0\}_{t\in\mathbb{N}}$, which is called the *canonical filtration*.

Predictability: Let $\mathcal{I} := \{0, 1, 2, \ldots\}$. Given a filtration $\{\mathfrak{F}_t\}_{t\in\mathcal{I}}$ in $\mathfrak{E}$, a sequence of random variables $\{\xi_t\}_{t\in\mathcal{I}}$ on $(\mathcal{E}, \mathfrak{E}, P)$ is said to be predictable with respect to $\{\mathfrak{F}_t\}_{t\in\mathcal{I}}$ if ξ_t is $\mathfrak{F}_{t-1}$-measurable, for all $t \geq 1$ (Durrett, 2010).

Martingales: Let $\mathcal{I}$ be a partially ordered set. Consider $\{\xi_t\}_{t\in\mathcal{I}}$ as a family of real-valued integrable, random variables, i.e. $\mathcal{E}' = \mathbb{R}$ and $\mathfrak{E}' = \mathfrak{B}^1$, adapted to the filtration $\{\mathfrak{F}_t\}_{t\in\mathcal{I}}$. Then $\{\xi_t\}_{t\in\mathcal{I}}$ is a super-martingale if, for all pairs $i, j \in \mathcal{I}$ with $i \leq j$, the following condition is satisfied:

$$\forall i, j \in \mathcal{I}: \quad i \leq j \Longrightarrow \mathbb{E}[\xi_j \mid \mathfrak{F}_i] \leq \xi_i \qquad \text{(a.s.)} \; . \tag{2.99}$$

If, instead, $\{-\xi_t\}_{t\in\mathcal{I}}$ satisfies the above condition, then $\{\xi_t\}_{t\in\mathcal{I}}$ is a sub-martingale. Lastly, if $\{\xi_t\}_{t\in\mathcal{I}}$ is both a super-martingale and a sub-martingale, meaning that equality holds in the above equation, $\{\xi_t\}_{t\in\mathcal{I}}$ is a martingale.

[4] In probability theory, the term *stochastic process* is commonly reserved to collections of random variables indexed by a uni-dimensional partially ordered set (Bauer, 1981), such as $\mathbb{N}$, while *random field* refers to cases with higher-dimensional index sets, such as $\mathbb{R}^d$ with $d > 1$ (Berlinet and Thomas-Agnan, 2004). However, this thesis adopts the broader definition.

2.7 Summary

This chapter introduced the theory behind the main concepts applied by the methods that this thesis proposes. Bayesian optimisation and Gaussian processes are omnipresent throughout the contributing chapters. Reproducing kernel Hilbert spaces, besides their connection to GPs, have been applied to formulate parametric models for control policies in Chapter 4. Optimisation problems affected by uncertain inputs are approached by Chapter 5 and Chapter 6, whose methods involve Section 2.2.4 and the background on topology and measure theory.

For more information on theoretical details, the machine learning literature is filled with reference material on the topics introduced here. Murphy (2012) presents a general overview on several probabilistic learning methods. For a better introduction to Gaussian processes, a popular textbook is Rasmussen and Williams (2006). Theory on reproducing kernel Hilbert spaces is found in Berlinet and Thomas-Agnan (2004) and Steinwart and Christmann (2008), and, for a more practical perspective on kernel methods, Schölkopf and Smola (2002) is recommended. Lastly, Bayesian optimisation has been a very active research area, so that material on it is mostly found in articles (Brochu et al., 2010; Shahriari et al., 2016), but a book (Mockus, 1989) and other theses (Lizotte, 2008; Marchant Matus, 2016) are also available.

CHAPTER 3

Active perception for modelling energy consumption

3.1 Introduction

Mobile robots have become successful tools for outdoor applications, e.g. planetary exploration (Carsten et al., 2007), mining operations (Brown, 2012), agriculture (Underwood et al., 2017) and many others. Contrasted with indoors, outdoor environments offer huge variability and challenges that the robot's navigation system may not be prepared for. Therefore, to enable robots to autonomously navigate through multiple types of environment, navigation systems are required to adapt to unforeseen challenges.

A critical factor that influences a robot's autonomy is energy consumption (Sun and Reif, 2005; Tokekar et al., 2011; Pereira et al., 2013; Morbidi et al., 2016). Mobile robots usually carry a limited amount of power supply on board, most of which is spent by the robot's propulsion system, whether it is aerial, terrestrial or aquatic. In addition, following paths that demand too much power from the robot's propulsion system can cause excessive stress to the platform. Navigation systems for autonomous robots should then be able to take energy consumption as a cost factor when planning trajectories.

Traversability models (Martin et al., 2013) that take into account power demands can aid robotic navigation systems to find energy-efficient paths. Underwater robots, for example, may be subject to strong ocean currents depending on the path they take (Pereira et al., 2013). Aerial robots can take advantage of the wind to facilitate their locomotion (Lawrance and Sukkarieh, 2009). In the case of ground robots, properties of the terrain can also directly influence power consumption. Most of these factors, however, are hard to model: variations in roughness (Molino et al., 2007), terrain slope (Sun and Reif, 2005), and interactions between the wheels and the soil (Ben Amar and Bidaud, 1995). Altogether, these environment-related complexities introduce uncertainty in the estimation of energy demands.

This chapter presents [1] a method to find energy-efficient paths over unknown terrains while learning a power-consumption model over the terrain. The method uses only information from estimates of the robot's pose and power consumption collected as the robot travels from start to goal locations. An active perception approach is applied to select informative paths between the two locations according to the estimates given by the model. Specifically, this work provides the following contributions:

(1) a Gaussian process model to estimate the power consumption of a robot moving at constant linear speed as a function of the robot's location and heading direction;

(2) a path-planning method that explores paths between two fixed locations building a model of the robot's power consumption over the terrain using Bayesian optimisation.

The remainder of this chapter is organised as follows. The next section reviews relevant prior work in this area. Section 3.3 specifies and formulates the problem of modelling energy consumption, which is followed by a description of the proposed path-planning algorithm in Section 3.4. Then, in Section 3.5 experimental results are presented both in simulation and on a physical robot to evaluate the proposed approach. Finally, Section 3.6 concludes and proposes some directions for future work.

3.2 Related work

This section reviews some relevant work from the area of energy-aware motion planning and Bayesian approaches applied to planning under uncertainty.

3.2.1 Vehicle-dynamics control

Heavy robots usually spend most of their energy supply on mobility. Therefore, research effort has been put into modelling the power demands of traction systems to provide cost models for energy-aware path planners. In this sense, energy efficiency can be achieved by motion control strategies that avoid costly movements like turning (Morales et al., 2006) and others demanding high torque (Salan et al., 2015), which can be estimated based on a dynamics model. Another approach is to directly optimise velocity profiles (Kim and Kim, 2007; Tokekar et al., 2011) for a given system. In general, these methods are dependent on parameters that require extensive evaluation and analysis of the robot in experiments and are highly vehicle dependent.

[1] This work has been partially presented at the 2016 Australasian Conference on Robotics and Automation (ACRA) as Oliveira et al. (2016)

3.2.2 Elevation-based strategies

Another strategy is to model power usage as a function of terrain properties, approaching it from a mechanical energy point of view. If the geometry of the terrain is known a priori, different techniques using grid-based elevation maps can be applied to estimate energy consumption along a path. This approach in general infers the power costs associated with individual actions over the terrain as a function of local slope angles and friction coefficients (Sun and Reif, 2005; Liu et al., 2008; Anuntachai et al., 2014; Ganganath et al., 2014). The energy expenditure of a path is then computed as the sum of the individual costs associated with each state transition on the path. However, these strategies require accurate information about the terrain type and its characteristics, besides the vast amounts of sensor data necessary to build the elevation models, which has its own challenges (Plagemann et al., 2008).

3.2.3 Bayesian approaches

Most of the approaches discussed so far depend on deterministic cost models that allow path planners to compute the cost of a path. Alternatively, Bayesian approaches allow deriving probabilistic cost maps to plan when uncertainty affects the estimation of cost factors.

Plonski et al. (2013) present a Bayesian approach to model energy. They use Gaussian process regression to build a solar-power map over an area, and then use this map to plan energy-optimal paths using empirical cost models to predict the energy spent on individual actions. Similarly, Martin and Corke (2014) present an approach to map the robot's power usage over the terrain via GP regression. In their case, at each iteration, the current GP mean is used as a cost map for planning using D* (Stentz, 1994) and then updated with new power-consumption measurements collected along the way. A limitation of their model is that it does not consider uneven terrains, where the robot's heading direction can affect power consumption. Additionally, the uncertainty of the model's predictions is not taken into account during the planning phase.

Marchant and Ramos (2014) present a Bayesian optimisation approach to plan continuous paths enabling robots to monitor spatial-temporal environmental processes. They use a GP to learn a model of the physical phenomena that the robot is monitoring, while BO is employed to select the most informative paths based on a path-integral acquisition function. In this problem, the main goal is to build a model of the environmental process throughout the whole area of the experiment, with less uncertainty in areas close to peaks of the process. There is no goal location for the robot to reach, and the robot's heading does not affect the readings. However, a similar framework can be applied to the problem in this chapter.

3.3 Problem formulation

Consider the problem of enabling a robot to learn a model that can be used to find energy-efficient paths in an obstacle-free area over a terrain with smooth elevation changes. The paths are taken from a family of parametric curves $\boldsymbol{\xi} \in \mathcal{C}$ and constrained by a given fixed starting pose (x_0, y_0, θ_0) and a goal location (x_f, y_f). Since the search is for energy-efficient paths, the model should provide less uncertainty in the predictions for paths that will yield better energy savings. In addition, assume that the only information the robot has access to are noisy estimates of its instantaneous power consumption w and its pose. The robot has no access to information about the terrain itself, such as slope, roughness, elevation maps, etc.

The total amount of energy E needed to execute a path $\boldsymbol{\xi} \in \mathcal{C}$ can be obtained by integrating the instantaneous power consumption w along the path over time, i.e.:

$$E[\boldsymbol{\xi}] = \int_{t_0}^{t_f} w(t)\,\mathrm{d}t\,, \tag{3.1}$$

where t_0 corresponds to the starting time, when the robot leaves the initial configuration (x_0, y_0, θ_0), and t_f corresponds to the final time, when the robot reaches the final location (x_f, y_f).

3.3.1 Assumptions

The modelling approach in this work assumes power consumption to be a function only of the robot's configuration along the path, which is a 2D pose, i.e. $\boldsymbol{\xi}(t) = (x(t), y(t), \theta(t)) \in \mathcal{X}$, where $\mathcal{X} = \mathbb{R}^2 \times [-\pi, \pi]$ is the robot's configuration space. The robot is assumed not to move backwards. Poses then inform the direction the robot is heading, which can affect power consumption over uneven terrains, depending on whether the robot is moving upwards or downwards. However, the pose alone does not provide enough information about other variables that can also affect the power consumption of the robot. Hence the proposed methodology considers a set of plausible assumptions. Firstly, the variations in the amount of power drawn by the traction system when the robot is in motion are usually much higher than the ones presented by the other on-board systems in any reasonably-sized robot. So one can assume that the power consumption of these other systems is approximately constant. Secondly, the terrain is assumed to be a static environment, so that its characteristics do not change over time.

Lastly, in experiments the robot will travel at a constant linear speed with angular speed low enough, so that turning movements do not significantly affect power consumption. In addition, a learning model that inherently considers noise in the underlying power-consumption sampling process should be able to adapt to the noise introduced by the former limitations.

3.3.2 The task

Considering the given modelling assumptions, predicting the amount of energy required to execute a particular path is equivalent to providing a model to predict the expected instantaneous power consumption $w(t) = w(\boldsymbol{\xi}(t))$ at each robot configuration $\boldsymbol{\xi}(t) \in \mathcal{X}$ along the path and then integrating over these estimates. The task is to learn a function $w : \mathcal{X} \to \mathbb{R}$ mapping robot configurations to power by using only pose estimates and power-consumption data collected along paths selected for execution. Prior knowledge about the environment or the robot's internal dynamics are considered unavailable.

3.4 Methodology

This section proposes an active perception approach to solve the task above. The method applies Bayesian optimisation to minimise the robot's integrated power consumption along paths while learning a model of the robot's power consumption over the terrain. *Active perception* here refers to the process of actively selecting locations where to sense the environment and collect information to update an internal model of it. The power-mapping function $w : \mathcal{X} \to \mathbb{R}$ is modelled as a sample from a Gaussian process. Given a family of parameterised curves, BO is then applied to select paths by maximising an acquisition function over GP estimates of the paths' energy requirements. Section 3.4.1 presents an overview of the system. The GP modelling approach is introduced in Section 3.4.2. Finally, Section 3.4.3 presents the BO method for path planning.

3.4.1 System overview

Figure 3.1 shows a simplified diagram of the proposed system. The GP model is designed to learn the expected power usage of the robot as it traverses the terrain. The model uses power-consumption measurements from traversed locations to infer the power necessary to traverse other locations, taking into account the heading direction as well. The BO algorithm queries the GP for individual estimates of power requirements along a candidate path. By maximising a utility function, the algorithm selects the

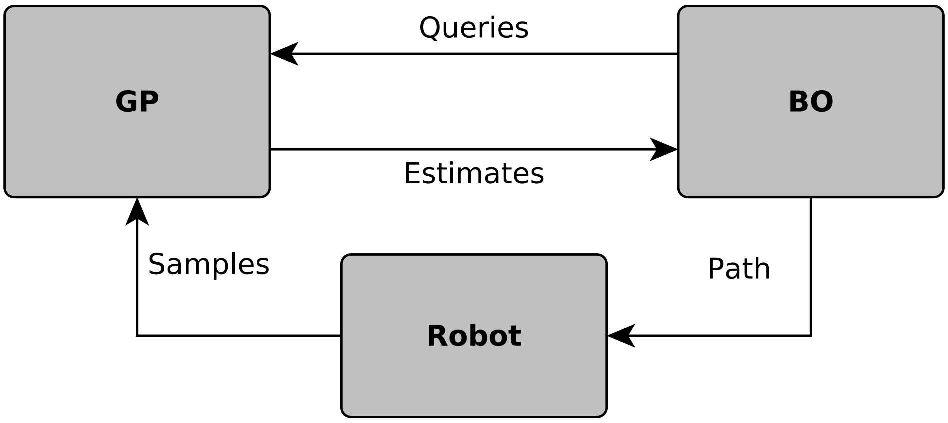


FIGURE 3.1: System diagram. The GP learns a model of the power consumption of the robot from the samples collected as it moves. The BO algorithm maximises a utility function over the GP estimates and selects the best candidate path. The robot executes the path, and the GP model is updated.

next path to follow. The paths are selected from a family of parametric curves connecting fixed start and goal locations. The selected path is given to the robot, which executes it, always starting from the same location, analogue to an episodic setting (Deisenroth, 2013). The power-consumption measurements collected along the path are fed back as observations to the GP, updating the model. This cycle repeats in each iteration of the BO algorithm. As iterations are performed, the expected amount of energy required to execute a path, as predicted by the GP, converges to the true amount of energy consumption.

3.4.2 GP model for power consumption

As a Bayesian non-parametric method, Gaussian process (GP) regression (Rasmussen and Williams, 2006) offers a powerful tool to model non-linear spatial processes. For this reason, a GP model is proposed to estimate the robot's instantaneous power consumption w over the configuration (or pose) space $\mathcal{X}$. With this model, a path's energy requirement can be predicted as an integral over the GP.

Power-consumption modelling

The GP power-consumption model is constructed with a constant prior mean:

$$m_w(\mathbf{x}) := m_s, \quad \forall\, \mathbf{x} \in \mathcal{X}\ , \tag{3.2}$$

where m_s is a hyper-parameter which in practice corresponds to the power consumed when the robot is stationary. The covariance between two power-consumption values $w(\mathbf{x})$ and $w(\mathbf{x}')$, where $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$

are given poses, is modelled using a version of the squared-exponential kernel:

$$k_w(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \exp\left(-\frac{1}{2}\rho_q^2(\mathbf{x}, \mathbf{x}')\right), \tag{3.3}$$

where $\rho_q(\mathbf{x}, \mathbf{x}')$ is a custom distance metric, defined below.

As orientation angles have a periodic nature, the Euclidean distance metric is not suitable to compute the distance between two different poses. For example, Euclidean distances assume that 0 and 2π are different numbers, while they mean the same as angles. To address this, a more suitable metric for the angular component is the following:

$$\forall \theta, \theta' \in [-\pi, \pi], \quad |\theta - \theta'|_{\cos} := \cos^{-1}(\cos(\theta - \theta')) \in [0, \pi]\,, \tag{3.4}$$

which corresponds to the shortest arc between the two given orientation angles. Using this angular metric, the distance between two poses $\mathbf{x} := (x, y, \theta)$ and $\mathbf{x}' := (x', y', \theta')$ becomes:

$$\rho_q(\mathbf{x}, \mathbf{x}') = \sqrt{\frac{(x - x')^2}{l_x^2} + \frac{(y - y')^2}{l_y^2} + \frac{|\theta - \theta'|_{\cos}^2}{l_\theta^2}}\,, \tag{3.5}$$

where l_x, l_y, l_θ are positive length-scale hyper-parameters also learnt from a training set.

With this kernel, the covariance between function values at two different points $\mathbf{x}$ and $\mathbf{x}'$ will smoothly decay according to their distance. Consequently, this covariance function is suitable for modelling power consumption as a Gaussian process on smooth natural terrains, i.e. terrains which do not have abrupt transitions of their surface characteristics.

All in all, the proposed GP model is fully determined by the hyper-parameters $(m_s, \sigma_f, l_x, l_y, l_\theta, \sigma_\nu)$. In experiments, these values were determined by maximising the log-marginal likelihood in a training phase on an initial set of observations. See Section 2.2.2 for more details about hyper-parameter learning.

Energy integration

Let μ_w and σ_w^2 denote the posterior mean and variance for the GP modelling w. Given a set of poses $\{\boldsymbol{\xi}(t_i)\}_{i=0}^n$ sampled along a candidate path $\boldsymbol{\xi} \in \mathcal{C}$, the total energy expenditure for that path can be modelled as $E[\boldsymbol{\xi}] \sim N(\mu_E[\boldsymbol{\xi}], \sigma_E^2[\boldsymbol{\xi}])$, where:

$$\mu_E[\boldsymbol{\xi}] = \sum_{i=0}^{n-1} \mu_w(\boldsymbol{\xi}(t_i))\Delta t_i \ , \tag{3.6}$$

$$\sigma_E^2[\boldsymbol{\xi}] = \sum_{i=0}^{n-1} \sigma_w^2(\boldsymbol{\xi}(t_i))\Delta t_i^2 \ . \tag{3.7}$$

The equation above provides us with an approximation of the true integrated energy consumption estimate from a GP, which would in fact be:

$$\mathbb{E}[E[\boldsymbol{\xi}]] = \int_t \mu_w(\boldsymbol{\xi}(t))\,\mathrm{d}t \ , \tag{3.8}$$

$$\mathbb{V}[E[\boldsymbol{\xi}]] = \int_t \int_{t'} k_{w|\mathcal{D}}(\boldsymbol{\xi}(t), \boldsymbol{\xi}(t'))\,\mathrm{d}t'\,\mathrm{d}t \ , \tag{3.9}$$

where $k_{w|\mathcal{D}}$ here denotes the posterior covariance function. By using only the predictive variance at the positions along the path, Equation 3.7 ignores the correlation between power-consumption values at different positions. However, the approximation in Equation 3.7 is computationally less expensive and allows for satisfactory results in practice, as the experiments (Section 3.5) will show.

Since the robot has no access to information about the terrain elevation, the time interval between two consecutive poses on a path can be estimated with a linear 2D motion model with constant speed v. Given two consecutive poses $\boldsymbol{\xi}(t_i) = (x(t_i), y(t_i), \theta(t_i))$ and $\boldsymbol{\xi}(t_{i+1}) = (x(t_{i+1}), y(t_{i+1}), \theta(t_{i+1}))$, we have that:

$$\Delta t_i \approx \frac{\sqrt{(x(t_{i+1}) - x(t_i))^2 + (y(t_{i+1}) - y(t_i))^2}}{v} \ , \tag{3.10}$$

where $v > 0$ is the robot's travelling speed, assumed constant. This approximation does not consider height differences, which could lead to estimation errors in areas of high slope. However, for typical smooth terrains, this should not lead to significant errors in the estimation of E.

3.4.3 Bayesian optimisation for path selection

Having a GP model for power consumption and a method to predict the total energy expenditure for a candidate path, this section proposes a BO algorithm for path selection. The first part presents a path acquisition function to select candidate paths with BO based on the proposed GP model (Section 3.4.2). As paths live in a function space, which is infinite dimensional, direct optimisation is not possible. So a parameterisation of the paths as splines is proposed. Lastly, the BO path-planning algorithm is described.

Acquisition function

The method proposed here applies BO to optimise over a model mapping paths to energy-consumption estimates. A function $R[\boldsymbol{\xi}]$ works as an acquisition function for BO based on a candidate path's expected energy consumption, modelled by $\mu_E[\boldsymbol{\xi}]$, and the uncertainty about it, σ_E^2. One of the criteria that allows BO to balance these two quantities, i.e. energy expenditure and uncertainty, is the upper confidence bound (UCB, see Section 2.3.2). Given the energy prediction model, an upper confidence bound over the negative of the energy estimate is given by:

$$R[\boldsymbol{\xi}] = -\mu_E[\boldsymbol{\xi}] + \lambda\sigma_E[\boldsymbol{\xi}] = -\sum_{i=0}^{n-1} \mu_w(\boldsymbol{\xi}(t_i))\Delta t_i + \lambda \left(\sum_{i=0}^{n-1} \sigma_w^2(\boldsymbol{\xi}(t_i))\Delta t_i^2 \right)^{1/2} , \tag{3.11}$$

where λ is a positive scalar which controls the exploration-exploitation trade-off. Higher values of λ increase exploration, giving high reward for selecting paths whose energy estimate has higher uncertainty. Observations along such paths might provide useful information to improve the accuracy of the GP estimates. On the other hand, lower values of λ favour exploitation, with the acquisition function preferring paths with low expected energy consumption.

Path parameterisation

In principle, the framework in this chapter could work with any class of parameterised curves to represent candidate paths. However, cubic splines provide a class of parametric curves $\mathcal{C}$ that is simple to work with and flexible enough to assess the capabilities of a learning algorithm for the problem in Section 3.3. Let $\boldsymbol{\xi}_{\boldsymbol{\alpha}} : [0,1] \to \mathcal{X}$ represent the path corresponding to a given setting of free spline parameters $\boldsymbol{\alpha}$, which are determined below. For any given $u \in [0,1]$, the corresponding pose $\boldsymbol{\xi}_{\boldsymbol{\alpha}}(u) = (x(u), y(u), \theta(u)) \in \mathcal{X}$ is given by the following equations:

$$x(u) = a_x u^3 + b_x u^2 + c_x u + d_x \tag{3.12}$$

$$y(u) = a_y u^3 + b_y u^2 + c_y u + d_y \tag{3.13}$$

$$\tan\theta(u) = \frac{\partial y(u)}{\partial x(u)} = \frac{3a_x u^2 + 2b_x u + c_x}{3a_y u^2 + 2b_y u + c_y} . \tag{3.14}$$

Algorithm 2: BO Path Planner

```
  Input:
        $\mathcal{S}$: spline coefficients search space
        $n_{\boldsymbol{\xi}}$: total number of paths
        $(x_0, y_0, \theta_0)$: starting pose
        $(x_f, y_f)$: goal location
1 for $i \in \{1, \dots, n_{\boldsymbol{\xi}}\}$ do
2     $\boldsymbol{\alpha}_i = \operatorname{argmax}_{\boldsymbol{\alpha} \in \mathcal{S}} R[\boldsymbol{\xi}_{\boldsymbol{\alpha}}]$
3     $\mathcal{Z}_i \leftarrow$ Sample along $\boldsymbol{\xi}_{\boldsymbol{\alpha}_i}$
4     $\mathcal{D}_i = \mathcal{D}_{i-1} \cup \mathcal{Z}_i$
  Result: $\mu_w, \sigma_w^2$
```

Since the starting pose at $u = 0$ and the final position of the robot at $u = 1$ are fixed, only (a_x, a_y, c_x) remain as free spline parameters, with the others given by:

$$d_x = x_0 \tag{3.15}$$

$$d_y = y_0 \tag{3.16}$$

$$c_y = c_x \tan \theta_0 \tag{3.17}$$

$$b_x = x_f - x_0 - c_x - a_x \tag{3.18}$$

$$b_y = y_f - y_0 - c_x \tan \theta_0 - a_y \ , \tag{3.19}$$

and thus $\boldsymbol{\alpha} = (a_x, a_y, c_x)$.

The algorithm

With the above path parameterisation, Algorithm 2 outlines the proposed BO path planner. At each iteration, the algorithm starts in line 2 by choosing spline parameters $\boldsymbol{\alpha}_i$ whose corresponding path $\boldsymbol{\xi}_{\boldsymbol{\alpha}_i}$ maximises the acquisition function $R[\boldsymbol{\xi}_{\boldsymbol{\alpha}_i}]$. Any global optimisation algorithm can be applied to maximise $R[\boldsymbol{\xi}_{\boldsymbol{\alpha}}]$ as a function of the spline parameters $\boldsymbol{\alpha}$ given a compact search space $\mathcal{S}$. Then, the robot executes the path (line 3), collecting a set of observations $\mathcal{Z}_i = \{(\boldsymbol{\xi}_{\boldsymbol{\alpha}_i}(t_j), z_j)\}_{i=1}^{n}$, where $z_j = w(\boldsymbol{\xi}_{\boldsymbol{\alpha}_i}(t_j)) + \nu_j$ and $\nu_j \sim N(0, \sigma_\nu^2)$, and updates the GP, appending the new samples to its observations dataset $\mathcal{D}$ in line 4. After running a total of $n_{\boldsymbol{\xi}}$ paths, the algorithm finishes, providing a model for the power required to traverse the terrain via the GP posterior mean μ_w and variance σ_w^2.

Feasibility constraints

To avoid infeasible paths, path selection was constrained. The constraint functions check if the candidate path stays within the area specified for the experiment and if the path's maximum curvature does not exceed a specific limit. To constrain candidate paths to a pre-specified area $\mathcal{A}$, a constraint functional was set as:

$$C[\boldsymbol{\xi}] = \sum_{i=1}^{n} c_v(1 - \mathbb{1}_{\mathcal{A}}(\boldsymbol{\xi}(u_i))) \;, \tag{3.20}$$

where $\{u_i\}_{i=1}^{n} \subset [0, 1]$ are uniformly spaced and c_v is a constant cost factor for going outside of the experiments area. Curvature is computed as the inverse of the radius of curvature. Given two consecutive poses (x_i, y_i, θ_i) and (x_j, y_j, θ_j) along a path, the curvature at pose i looking ahead to pose j is computed as:

$$\kappa_i = \frac{2}{\rho_{ij}} \sin\left(\frac{\theta_j - \theta_i}{2}\right) \;, \tag{3.21}$$

where ρ_{ij} denotes the planar Euclidean distance between (x_i, y_i) and (x_j, y_j). Maximum curvature limits were set accordingly for each experiment.

The constraints above are checked for each setting of spline parameters evaluated during the acquisition function optimisation in Algorithm 2 using algorithms that support non-linear inequality constraints, such as the augmented Lagrangian method in Johnson (2014).

3.5 Experiments

This section presents experiments to assess how accurate the GP power-consumption model can be and how energy-efficient the paths found by the proposed BO algorithm are. Simulations evaluated the accuracy of the GP model estimates, assessed the effects of the UCB parameter λ on the BO planner and compared BO to other optimisation algorithms in the task of finding energy-efficient paths. After that, tests with a physical robot were performed to verify the BO planner in a real environment.

3.5.1 Simulation

This section presents the simulation model, followed by GP accuracy results and tests of the BO path-planning algorithm.

Simulation setup

Terrains were modelled as continuous height maps generated by means of Perlin noise (Perlin, 1985). The following equation computes synthetic ground-truth values for the robot's power consumption:

$$w(\mathbf{x}) = \begin{cases} (f_c + m_r g \sin s(\mathbf{x}))v, & s(\mathbf{x}) \geq 0, \\ f_c(1 + \sin s(\mathbf{x}))v, & s(\mathbf{x}) < 0 \end{cases} \tag{3.22}$$

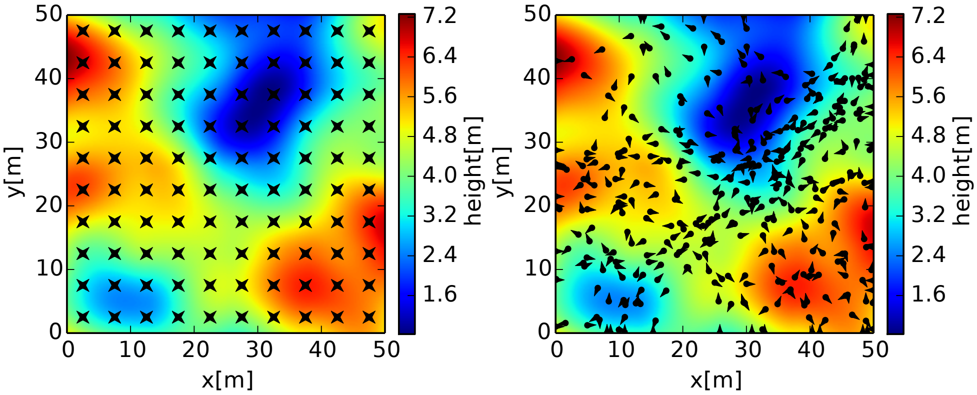
where m_r corresponds to the robot's mass, g is the gravity acceleration magnitude, $s(\mathbf{x})$ is the local terrain slope angle, and f_c is a constant factor to simulate the effect of forces independent of the robot's pose on the terrain. Overall, this formulation simulates large efforts when the robot is moving uphill, having to overcome its own weight, and lower, but non-negative, efforts when it is moving downhill. Zero-mean Gaussian noise with standard deviation σ_ν was added to the power-consumption values provided as observations to the GP. Table 3.1 presents the settings used for the tests.

f_c	m_r	g	v	σ_ν
200 N	100 kg	10 m/s^2	1 m/s	20 W

TABLE 3.1: Simulation settings

GP model accuracy

The first tests evaluated the quality of the GP power-consumption model predictions using two types of training sets. Figure 3.2a shows the first type, where sample poses are uniformly spaced 5 metres apart on an (x, y)-grid pattern and have orientation angles $\theta \in \{-135^o, -45^o, 45^o, 135^o\}$, yielding a total of 400 samples in the 50-by-50 metres terrain. Figure 3.2b shows the second type of training set, containing random-walk samples with consecutive poses separated by 5 metres and with turns sampled from a uniform distribution limited between -30^o and 30^o. This set also contains the same amount of poses as the previous one, 400 in total. In both cases, the GP was queried with a test-set at a higher spatial resolution, with points spaced by 1 m over a grid, and at the same angular resolution of the first case, but with an offset, such that $\theta \in \{-90^o, 0^o, 90^o, 180^o\}$ for the queries. In total, the query set contains 10,000 query poses. The accuracy in the predictions was evaluated by computing the relative error between the predicted and the ground-truth values for each query pose. The root mean square (RMS) value of the relative error for the case in Figure 3.2a was 14.5%, and for the case in Figure 3.2b, it was 17.2%. In both cases, the same GP hyper-parameter setting was used. Since the difference in the error levels between the ideal lawn-mower path of Figure 3.2a and the completely random path in



(A) Uniformly-spaced poses

(B) Random path

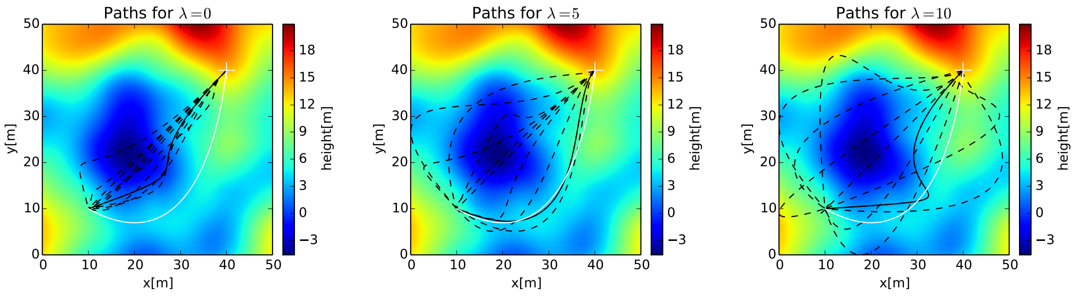
FIGURE 3.2: Example sets of sample poses used in tests with the GP model: on the left, a uniformly-spaced set, corresponding to a sweep pattern over the simulated terrain in 4 orientations; on the right, a random path where consecutive poses are separated from each other by a fixed-length segment and a random turn angle.

Figure 3.2b is relatively small (2.7%), the GP power-consumption model provides enough generalisation capabilities and is suitable for integration into the BO framework.

Impact of λ on the path planner

The algorithm's exploration-exploitation trade-off is controlled by the parameter λ of the acquisition function (see Equation 3.11). Too much exploration may lead to excessive energy expenditure, while too much exploitation may get the algorithm stuck at local minima, missing the most energy-efficient paths. Therefore, the BO path-planning algorithm was tested under different settings for λ across multiple simulated terrains.

Figure 3.3 presents a few test cases with the robot travelling between two diagonally-opposed locations over the terrain. The shortest path for this case is a straight line. However, this path goes through a valley, requiring a high amount of power draw for the robot to climb uphill in the area close to the goal. Consequently, in this scenario, the lowest-energy paths should not follow a straight line, but instead go around the valley to reach the goal location. The BO method should then be able to select paths to collect samples for the GP around that area, yielding better estimates for low energy-consumption paths. For comparisons, the optimal spline path was computed using simulated annealing on the noise-free cost function.



(A) $\lambda = 0$ (B) $\lambda = 5$ (C) $\lambda = 10$

FIGURE 3.3: Height map of the simulated terrain with executed paths for different settings of the UCB parameter λ (see Equation 3.11): Each candidate path is plotted with a dashed line, with the lowest-energy path found by the BO algorithm represented by a solid black line in each case. The optimum path, found by simulated annealing over the noise-free cost function, is also plotted in a solid white line. The robot's start location $(x_0, y_0) = (40, 40)$ is marked with a white cross (+) symbol. In all the cases, the robot's initial heading is $\theta_0 = -135^o$, i.e. the robot is headed downwards along the diagonal. The goal point is also fixed, located at $(x_f, y_f) = (10, 10)$.

In all the test cases in Figure 3.3, the algorithm was run for 10 iterations. For lower values of λ, the planner performs more exploitation than exploration. Consequently, the paths are usually very concentrated around a local low-energy region. These paths can also vary a lot among different runs of the test, depending on which low-energy path the algorithm finds first. However, for high values of λ the algorithm explores using longer paths, which can accumulate higher variance in the energy predictions. After 10 iterations, the estimates and the noise-free values for the lowest-energy paths found by the algorithm in each setting in Figure 3.3 are shown in Table 3.2. Among these test cases, the setting with $\lambda = 5$ found the best path in terms of energy consumption. Due to noise and significant differences in height along the paths, the estimated energy value was noticed to be slightly lower than its true value. However, for the case with $\lambda = 5$, the algorithm was still able to find a path close to the true optimum in terms of energy requirements, which requires 12396 J.

λ	0	5	10
$E[\boldsymbol{\xi}^*]$	14038 J	12663 J	16052.8 J
$\mu_E[\boldsymbol{\xi}^*]$	13437 J	12313 J	14584 J
$\sigma_E[\boldsymbol{\xi}^*]$	41 J	56 J	78 J

TABLE 3.2: True energy and its estimates for the lowest-energy paths found by the BO path planner in the test cases in Figure 3.3

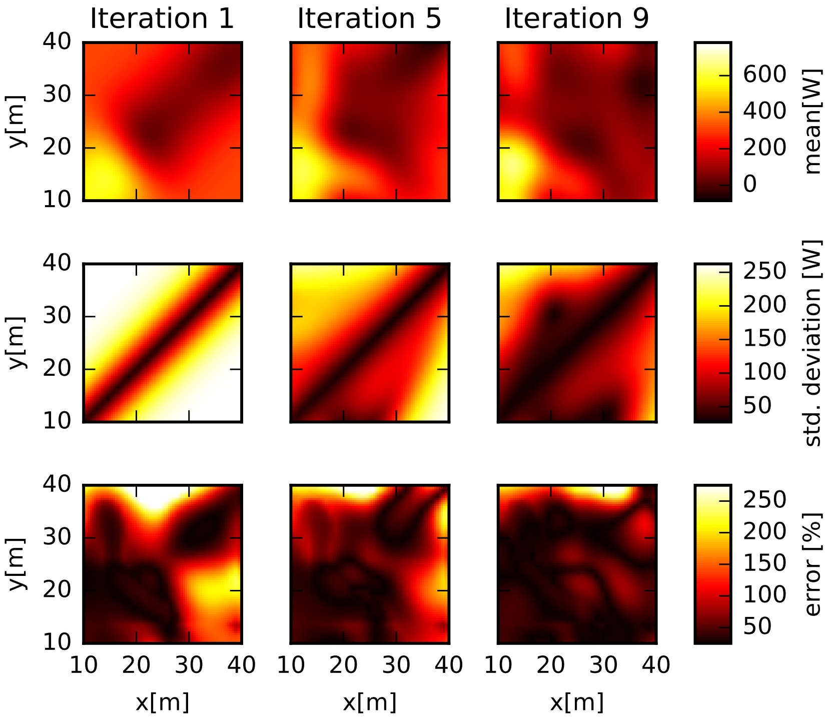


FIGURE 3.4: Evolution of the GP estimates for the robot's power consumption on poses with $\theta = -135^o$, main direction of movement, after iterations 1, 5 and 9 for the test case in Figure 3.3b. The upper plots show the GP predictive mean for the power consumption level, $\mu_w(\mathbf{x})$. The middle plots show the GP predictive variance (in terms of standard deviation, i.e. $\sigma_w(\mathbf{x})$). The lower plots show the relative error between the GP predictive mean and the computed ground-truth costs, i.e. $|1 - \mu_w(\mathbf{x})/w(\mathbf{x})|$.

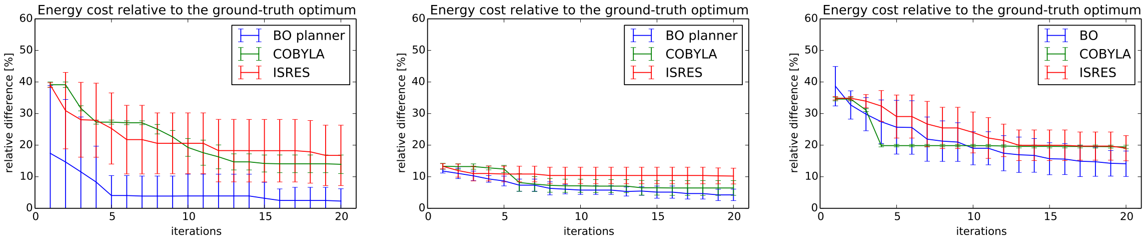
Though choosing a proper value for λ can be a trial and error process, in practice, a few points can be considered. For example, setting $\lambda = 2$ allows the algorithm to consider that a candidate path $\boldsymbol{\xi}$ could require an amount of energy in between $\mu_E[\boldsymbol{\xi}] \pm 2\sigma_E[\boldsymbol{\xi}]$ to be executed. This would correspond to a 95% confidence interval, if the GP model's hyper-parameters are properly set and the terrain slopes are not too steep for the approximation in the energy integration to hold (Section 3.4.2). On the other hand, if these assumptions are not valid, the true energy value can be quite far from the estimate, then a higher λ could allow the algorithm to perform better.

Figure 3.4 presents how the GP estimates evolve with the number of iterations in the paths for the above test case with $\lambda = 5$ (Figure 3.3b). As it can be seen in the error plots, the mean gradually converges to the ground-truth distribution in the area where the terrain is being explored. At the same time, the uncertainty about the estimations also reduces in the explored area. These results and the differences in paths seen for each setting in Figure 3.3 demonstrate that the proposed BO algorithm is effectively exploring the terrain to learn a power-consumption model and select paths based on their reward, which involves both cost and information gathered.

Comparisons

In the task of finding energy-efficient spline paths, the BO path planner was compared against the following non-linear optimisation algorithms:

- *Improved Stochastic Ranking Evolution Strategy* (ISRES) (Runarsson and Yao, 2005): a global optimisation algorithm; and
- *Constrained Optimization by Linear Approximations* (COBYLA) (Powell, 1998): a local optimisation algorithm.



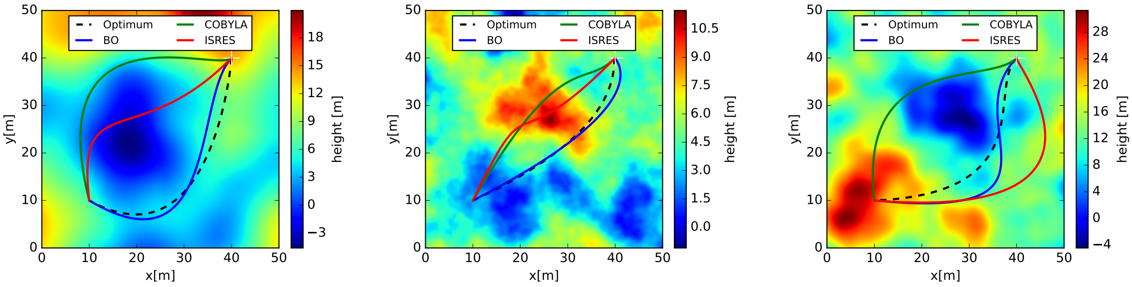
(A) Performance on Terrain 1

(B) Performance on Terrain 2

(C) Performance on Terrain 3

FIGURE 3.5: Averaged relative energy cost (%) between the current lowest-energy path found by each algorithm and the ground-truth optimum at each iteration. The error bars correspond to one standard deviation, computed over the re-runs of each algorithm. With only 5 iterations, the best paths found by the proposed BO planner are within 10% of the optimum. For these comparisons, $\lambda = 7$ was used (see Equation 3.11).

Publicly-available implementations of these algorithms, provided by the package NLopt (Johnson, 2014), were employed for the experiments. These algorithms directly support non-linear inequality constraints, which were used to constrain the curvature of the paths and to limit them to the experimental area. The algorithms were set to optimise spline parameters based only on the total energy spent for executing each path. Since both ISRES and COBYLA would have no prior information about the objective, the initial solution for the set of spline parameters was set to $\boldsymbol{\alpha}_0 = (0, 0, 0)$, which corresponds to a straight line from start to goal. Simulated annealing was used in these tests to find the true optimum spline path directly over the noise-free cost function for comparisons.



(A) Terrain 1

(B) Terrain 2

(C) Terrain 3

FIGURE 3.6: Examples of the lowest-energy paths found by each algorithm on the different terrains.

The performance of each method was evaluated on different terrains where the optimal spline path between the start and goal locations was non-trivial, i.e. not close to a straight line. This way, one may verify if the non-linear optimisation methods under comparison (COBYLA and ISRES) would be able to find paths close to the global optimum without the help of a GP model. Each method was configured to run for 20 iterations. The tests were repeated 10 times to ensure the consistency of the results. Figure 3.5 presents the performance of each algorithm in finding low-energy paths on 3 different terrain models. On average, the pure energy-based optimisation with COBYLA and ISRES seems to get stuck at a local minimum and, on average, is not able to find a better path within 20 iterations. On the other hand, since the BO planner builds a model of the terrain as it explores paths, it was able to find paths closer to the true optimum. Figure 3.6 presents some of the best paths found by each algorithm on each of those terrains.

3.5.2 Experiments with a physical robot

This section presents results obtained in experiments with a physical robot. A small four-wheeled skid-steer robot, shown in Figure 3.7a, was used as a test platform. The robot is equipped with an on-board computer running ROS[2]. To measure power consumption, the robot carries sensors measuring battery voltage and total current drain. For localisation, wheel odometry, IMU and GPS data are fused using an extended Kalman filter (EKF) providing full 6D pose estimates.

The experiments were performed on an open grass field in an urban park, shown in Figure 3.8a. For the start poses, the robot was positioned so that it was facing the north-east direction from within the area in the lower-left corner of the image. The goal position was located in the upper-right corner of the image

[2] Robot Operating System: `http://www.ros.org`

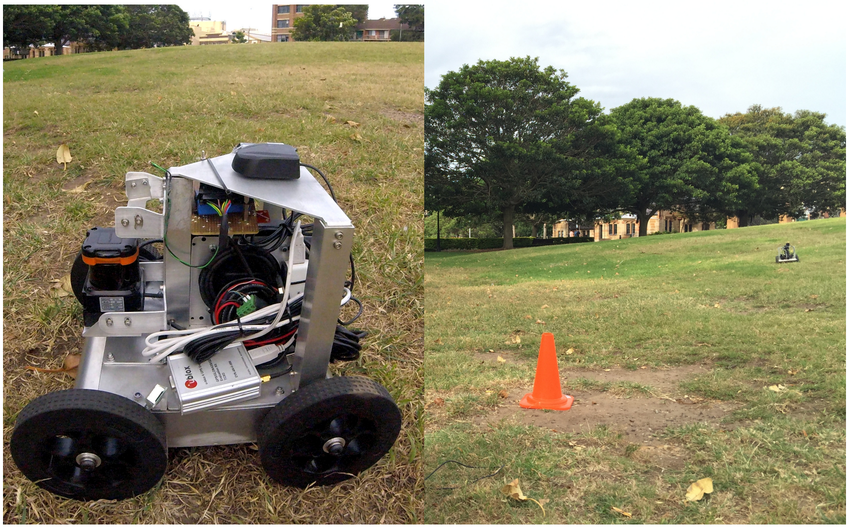
at a distance of roughly 60 metres from the starting position. This region had a small hill, partly seen in Figure 3.7b, between the robot's starting and end positions, so that paths going over the hill and paths going around it require different amounts of energy expenditure. The BO path-planning algorithm was run for 12 iterations with the robot moving at a constant 0.7 m/s linear speed and at a maximum of 0.25 rad/s angular speed. The acquisition function was set with $\lambda = 5$. Power-consumption measurements were collected at a rate of 2 Hz, with totals varying around 150 observations per path.

For this experiment, GP hyper-parameters, which include noise and signal variance, length-scales and the mean function's constant value, were learnt via maximum likelihood estimation using data from previous test runs and kept fixed during the BO process. Table 3.3 presents the hyper-parameter estimates. Although the noise level σ_ν in the power-consumption measurements is relatively large when compared to the signal level σ_f, the signal component is still twice as large allowing for meaningful estimates and planning. The length-scales reveal that terrain should allow for enough variation in power demands for paths stretching over tens of metres. The mean function parameter m_s indicates the average power consumption for the robot.

m_s	l_x	l_y	l_θ	σ_f	σ_ν
22.7 W	1.8 m	1.8 m	0.46 rad	2.9 W	1.4 W

TABLE 3.3: Field experiment hyper-parameters

Figure 3.8c shows the predictions of the GP model and the actual measured energy consumption for each path planned by BO, which are shown in Figure 3.8b. The measurement of energy consumption for an executed path was made by directly integrating the instantaneous power-consumption measurements given by the robot's power sensors over time. As the plot in Figure 3.8c shows, initially the energy requirement for the selected path is highly over-estimated, even though this first path goes over the hill in Figure 3.7b. The initial estimate comes from simply integrating the GP prior mean function m over the path, since the GP has no observations yet. Consequently, an overestimate for the energy in this case means that the average power consumption given by the mean function was too high. However, as the GP model is updated, the plot shows that the predictions gradually approximate the actual energy-consumption values, indicating that the model is improving over time, as seen previously in simulation. In addition, despite not much improvement due to the relatively simple terrain, towards the end, the paths' energy consumption are lower, demonstrating that the algorithm is able to find energy-efficient paths in the real-world scenario. This shows that the proposed method can adapt to multiple sources of noise and uncertainty in the physical world.



(A) Robot (B) Experiments area

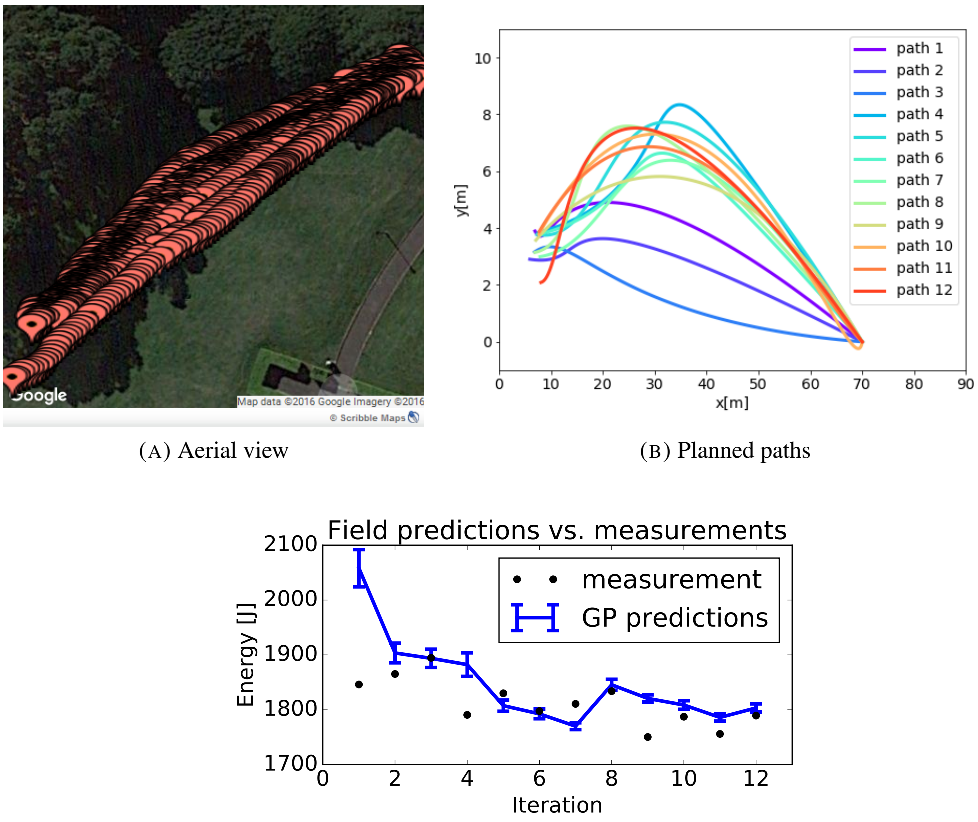
FIGURE 3.7: Robot used and partial view of the area where the field experiments were performed

3.6 Summary

This chapter presented a Bayesian optimisation method to actively learn a model capable of predicting energy costs for paths between a start and a goal location in off-road terrains. The method does not require information about the terrain's characteristics, requiring only access to the robot's localisation and power-consumption measurements as it executes a path. As the model is built, the method is able to find energy-efficient paths, using a utility function designed to balance the trade-off between exploration and exploitation. The method differs from previous BO approaches by considering an objective that can be expressed as an integral over a point-wise GP cost model between two fixed locations, instead of using a GP to model the integrated cost directly. In this case, both paths and the point-wise GP model are optimised simultaneously as the algorithm runs.

In simulation, results demonstrated the performance of the method in building an accurate model of the robot's power-consumption over the terrain. Then, in comparison with other non-linear optimisation algorithms, results showcase its ability to find low-energy paths close to the global optimum within a given family of parametric curves.

A path-planning experiment with a physical robot demonstrated the applicability of the proposed method to real-world environments, where a lot of the terrain and the vehicle's effects on power consumption are hard to model. Having a framework that does not depend on predefined models of these processes



(A) Aerial view

(B) Planned paths

(C) Energy consumption

FIGURE 3.8: Experimental results: (a) Aerial view of the experiments area overlaid with the paths executed by the robot running the BO planner. (b) Paths planned by the algorithm, where the start positions are located on the left side of the plot, while the goal is on the right. (c) Predictions and measurements for the energy required to execute each path attempted by the algorithm in the field experiments.

offers a great advantage in designing energy-efficient navigation systems. Therefore, the contributions in this chapter can provide effective adaptive methods for a variety of outdoor applications where energy consumption is a major concern.

As future work, the framework presented in this chapter can be adapted to include other variables not considered by the proposed power-consumption model, such as velocities. Other avenues include more complex path parameterisations and the integration of other constraints, such as obstacle avoidance, during planning. The method can also contribute to functional optimisation (Vien and Toussaint, 2018).

CHAPTER 4

Model-free optimisation of control policies

4.1 Introduction

The last chapter presented a problem involving optimisation in function space, a space of trajectories in that case. The planner had access to a Gaussian process model that allowed forward simulating the cost of trajectories and a parameterised search space which was only three-dimensional. In other instantiations of functional optimisation, however, there may be no pre-defined model available nor an efficient approach to learn one, and the parameter space can be very high-dimensional. In robotics, examples include tasks such as ball-throwing, where one could learn a model (Kupcsik et al., 2013), trying to hit a target on a wall with a dart, where learning a model may not be possible (Kober et al., 2010), etc. (see Mahmood et al., 2018, for additional examples). In other cases, individual costs or rewards might be quantifiable for each action, but a global approach, considering a whole episode of execution, could yield better results, such as in autonomous racing (Rizano et al., 2013). These problems lead to a need for policy-search methods that can deal with sparse rewards and high-dimensional parameter spaces.

This chapter addresses the problem of optimising functions of many parameters, under a limited evaluation budget, considering an application to policy search (Deisenroth, 2013). In these problems, the number of policy evaluations is limited due to the need to interact with a physical system or expensive physics simulations. Bayesian optimisation has shown promising results in policy-search problems involving expensive policy evaluations (Martinez-Cantin et al., 2007; Wilson et al., 2014), however, it still suffers with high dimensionality, where *high* for BO can be anything greater than 10 dimensions (Wang et al., 2013). Although methods which apply BO in high-dimensional search spaces have been proposed (Wang et al., 2013; Kandasamy et al., 2015; Li et al., 2016; Rolland et al., 2018), how to do so without making restrictive assumptions about the objective function remains an open question.

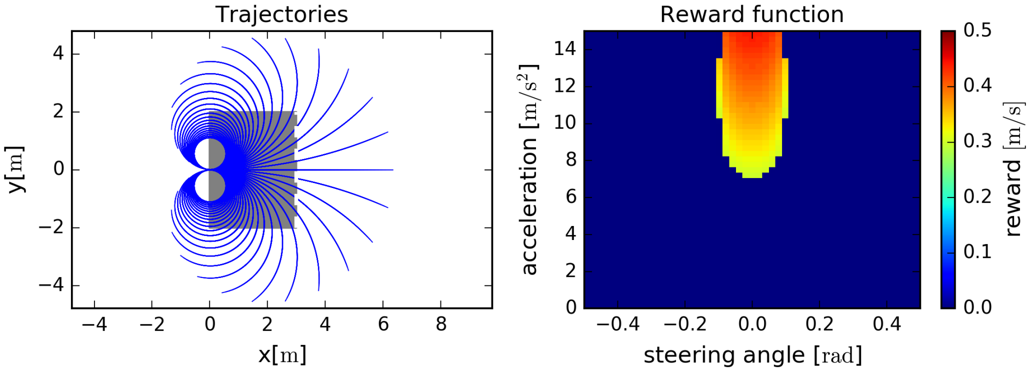


FIGURE 4.1: Restriction of the valid search space under track constraints: The plot on the left presents trajectory roll-outs for a robot car, starting at location $(0, 0)$, for a range of acceleration and steering angle values. The track is 3 metres long and 4 metres wide and is represented by the shaded area in the left-side plot. The robot's reward is either zero if it does not complete the track or its final speed, case it completes the track. The resulting reward surface is shown on the right-hand side.

The contribution in this chapter [1] is a method that applies Bayesian optimisation to optimise control policies with high-dimensional parameter spaces. The method is applied to the problem of minimising the lap time of a robot race car around a given track. No prior information about the track nor the robot's dynamical model is given, just an initial sub-optimal driving example. We formulate policies as functions in a reproducing kernel Hilbert space (RKHS) (Schölkopf and Smola, 2002) and parameterise them using finite-dimensional feature maps.

The method proposed in this chapter tackles the issues that arise in high-dimensional BO from the search perspective, while still using all available information to build a model of the objective function. Planning problems in scenarios like racing are heavily constrained by the environment, such as the track and the dynamical limits of the car. As a consequence, the mass of the reward function tends to be highly concentrated within a particular region of the search space, as most of it is composed of invalid examples. An illustrative example is shown in Figure 4.1. Therefore, the main argument for the methodology proposed in this chapter is that, by applying a relatively simple method, such as randomised coordinate descent (Nesterov, 2012), and starting the search from a valid initial solution, one can optimise over a high-dimensional GP prior efficiently. Coordinate descent relies on decomposing a high-dimensional problem into simpler, lower-dimensional sub-problems, which are solved more easily. Valid initial solutions for each BO iteration can be determined from the previous observations, such as

[1] The work in this chapter has been previously presented at the 2018 IEEE International Conference on Robotics and Automation (ICRA) as Oliveira et al. (2018b)

the current policy with the highest reward. Experiments empirically demonstrate that this simple search strategy can be effective when combined with BO in the particular class of problems addressed here.

The remainder of this chapter is organised as follows. The next section reviews related work in the areas of Bayesian optimisation, reinforcement learning and autonomous racing. Section 4.3 follows with a formal presentation of the optimisation problem approached by this chapter and some background information about BO. Then, in Section 4.4, the proposed BO method is presented. Section 4.5 reports the results of experimental evaluations using a race car simulator, comparing the proposed method against other optimisation algorithms. Finally, a summary and some directions for future work are provided in Section 4.6.

4.2 Related work

This section reviews relevant prior work contrasting them to the purposes of the contribution in this chapter. Firstly, a presentation of how the racing problem has been previously approached illustrates some of the issues to be addressed. Then the relation to reinforcement learning is discussed. Lastly, the section reviews Bayesian optimisation approaches to policy search and other high-dimensional search problems.

4.2.1 Autonomous racing

The problem of finding a control policy that will allow a robot racer to finish a given track in minimal time has been approached in many ways. Model predictive control (MPC) techniques have been applied to locally optimise driving policies over receding-horizons based on external sensor data and internal dynamical models, which are either pre-designed (Liniger et al., 2015) or learnt (Williams et al., 2017). From a global optimisation perspective, the same problem has been approached as racing line optimisation (Rizano et al., 2013), i.e. finding the path within the track that will allow the car to finish the race in minimal time, given a map of the race track and a kinematic model of the vehicle. When global information about the racing environment is available through external images, deep reinforcement learning architectures have also been applied to fine-tune control policies (Sallab et al., 2017). Lastly, this problem has also been approached by using evolutionary algorithms to optimise parameters for policies that either combine different pre-designed heuristics (Butz and Lonneker, 2009), or directly map the state of the car and the opponents to control actions via a gene regulatory network (Sanchez and Cussat-Blanc, 2014).

4.2.2 Reinforcement learning

Learning a control policy that maximises a reward that depends on the combined effect of its actions can also be approached as policy search in reinforcement learning (RL) (Deisenroth, 2013). Both model-based and model-free techniques have been proposed in the literature. In Kupcsik et al. (2013), for example, a framework is proposed to allow for algorithms that learn policies using an internal, also learned, GP-based forward model of the robot to simulate trajectory roll-outs in an internal optimisation step before each episode of policy execution. Neural-network learning models have also been applied to policy search, including applications to autonomous racing (Williams et al., 2017), in a receding horizon setting. Model-based techniques encode prior assumptions about the physical system where the policy runs. Such assumptions limit the applicability of the technique and require prior knowledge about the system, at least at a high level, which is not always available. In terms of model-free approaches, in Kober et al. (2010) a robot learns how to throw a dart to hit a target with an algorithm that learns a model of the optimal policy-parameters distribution. They apply a Gaussian process prior over the space of contexts for each single parameter and employ no model of the robot itself. Model-free approaches have also been successfully applied to a number of other robotics tasks, reaching, speed driving and position control (Mahmood et al., 2018). In contrast, the method this chapter proposes directly learns a GP model of the reward function over a policy parameters and performs BO to optimise them.

4.2.3 Bayesian optimisation approaches

Bayesian optimisation has been previously applied to policy-search problems with some success. However, the applications usually had parameter spaces whose dimensionality was low enough. The following reviews a few of these cases and discusses current BO approaches to high-dimensional problems.

BO for policy search

Bayesian optimisation has also been applied to problems involving policy search. A similar approach to the one in this chapter is presented in Martinez-Cantin et al. (2007), where the authors model cost as a direct function of policy parameters. In that paper, the robot's task was to reduce uncertainty about its location and its surroundings, and the policies were parameterised by a small number of variables. Since the policy-execution outcome/reward is usually more-closely a function of the resulting behaviour of the robot than of the policy parameters themselves, another approach to this problem is to apply a GP prior over this mapping from behaviours to rewards (Wilson et al., 2014), while still doing BO over the policy-parameters space. That approach, however, requires that enough information can be observed from the system. More recently, other BO approaches to direct policy search have been proposed (Martinez-Cantin, 2017; Chatzilygeroudis et al., 2017), dealing with different aspects of the policy-search problem, which involve low-dimensional parameter spaces. Our aim in this chapter is to learn policies that can have enough representation power, while requiring the least amount of information from the system. High representation power generally implies high-dimensional parameter spaces, which can be an issue for traditional BO methods.

BO in high-dimensional problems

One common issue with the presented applications of BO is the curse of dimensionality. The great majority of BO algorithms use GPs to learn and model the objective function. One of the core issues is that GP modelling does not scale well with high dimensions and/or large amounts of data, degrading the performance of BO in such settings. Several approaches have been recently developed to tackle the scaling of GP/BO to high dimensions. One could, for example, assume that the objective function only depends on a small subset of input coordinates and do variable selection to find these (Chen et al., 2012). Ulmasov et al. (2016), for instance, applies principal component analysis (Murphy, 2012) to determine a probability distribution over possible subsets of variables and keeps a GP model over each possible decomposition, one of which should prevail as the most sampled at the end of the process.

Another approach is to assume there are lower-dimensional linear embeddings containing most of the variation of the function. On the one hand, a single linear embedding might include the optimum of the objective function in a dimensionality reduction approach (Wang et al., 2013; Djolonga et al., 2013). On the other hand, one could assume that the objective function is additively composed of several low-dimensional functions either disjoint (Kandasamy et al., 2015; Li et al., 2016) or inter-dependent (Rolland et al., 2018).

Modelling the objective function as a sum of unidimensional components relates to the work in this chapter. However, the proposed method does not require such assumption to be true and places a full GP prior on the objective function, instead of modelling each component individually as Kandasamy et al. (2015). Determining the correlations between each dimension, as in Rolland et al. (2018), is also not a concern, but simply making the search for candidates more efficient by applying a randomised coordinate-wise optimisation scheme.

4.3 Preliminaries

This section provides a formal description of the problem setup, followed by the formulation of a class of parameterised policies.

4.3.1 Problem statement

Let $\mathcal{X}$ denote the set of positions along a race track, and let $\mathcal{U}$ denote an action space. Both $\mathcal{X}$ and $\mathcal{U}$ are continuous spaces. Consider policies mapping the position $x \in \mathcal{X}$ of the robot along the track to a corresponding control action $u \in \mathcal{U}$. The goal is to find the control policy $\pi : \mathcal{X} \to \mathcal{U}$ that minimises the time τ taken to complete the track under constraints imposed by the track itself and the robot dynamics.

The reward R is defined as:

$$R = \begin{cases} \ell/\tau \ , & \text{if track completed} \\ 0 \ , & \text{if failed to complete the track} \end{cases} \tag{4.1}$$

where ℓ is the length of the track. Therefore, for success cases, the reward is equivalent to the average linear speed of the robot. In this sense, minimising τ is equivalent to maximising R. Given a class of policies $\mathcal{H}$, we search for:

$$\pi^* = \underset{\pi \in \mathcal{H}}{\operatorname{argmax}} \ R[\pi] \ . \tag{4.2}$$

We seek a method that solves the above problem for any kind of mobile robot and without needing a map of the track. Therefore, model-based solutions are inapplicable, for they would have to learn an approximate transition model of the robot, whose representation varies among different driving mechanisms, and use this model to simulate trajectories over the track map.

4.3.2 Policy parameterisation

The optimal policy π^* is assumed to belong to a reproducing kernel Hilbert space $\mathcal{H}$ (Section 2.4). RKHS-based methods has been relatively successful in modelling trajectories for motion planning (Marinho et al., 2016) and components of stochastic control policies for reinforcement learning (Bagnell and

Schneider, 2003; Vien et al., 2016). The RKHS formulation allows the policy to assume a variety of shapes, depending on the choice of kernel function. Prior knowledge about optimal policy characteristics, such as the degree of smoothness, can be encoded in the class via the kernel so that every policy in the class presents them.

Considering a one-dimensional action space $\mathcal{U} \subset \mathbb{R}$, evaluating a policy $\pi \in \mathcal{H}$ at a point $x \in \mathcal{X}$ is equivalent to an inner product:

$$\pi(x) = \langle \pi, k(\cdot, x) \rangle_{\mathcal{H}} \,, \tag{4.3}$$

where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the reproducing kernel of $\mathcal{H}$. In a RKHS, kernels are equivalent to inner products between feature maps $\phi : \mathcal{X} \to \mathcal{H}$ in the corresponding Hilbert space, such that $k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ and $\pi(x) = \langle \pi, \phi(x) \rangle_{\mathcal{H}}$. The features $\phi(x)$ are usually infinite-dimensional vectors, making the direct optimisation of $\pi \in \mathcal{H}$ computationally intractable. However, several techniques in the kernel machines literature (Rahimi and Recht, 2007; Le et al., 2013; Avron et al., 2016) propose approximating $\phi(x)$ by a finite-dimensional vector $\hat{\boldsymbol{\phi}}(x) \in \mathbb{R}^q$, $q < \infty$, such that, for any $x, x' \in \mathcal{X}$, $k(x, x') \approx \hat{\boldsymbol{\phi}}(x)^\mathsf{T} \hat{\boldsymbol{\phi}}(x')$. As a consequence, we have that:

$$\pi(x) \approx \pi_{\mathbf{w}}(x) = \mathbf{w}^\mathsf{T} \hat{\boldsymbol{\phi}}(x) \,, \tag{4.4}$$

where $\mathbf{w} \in \mathbb{R}^q$ is a vector of weights, which uniquely determines the approximate policy $\pi_{\mathbf{w}}$ for a given finite feature map $\hat{\boldsymbol{\phi}}$. Therefore, we can replace the objective in Equation 4.2 by the search for an approximate solution $\pi_{\mathbf{w}^*}$ with:

$$\mathbf{w}^* = \operatorname*{argmax}_{\mathbf{w} \in \mathcal{S}} R[\pi_{\mathbf{w}}] \,, \tag{4.5}$$

where $\mathcal{S} \subset \mathbb{R}^q$ is a compact search space.

Relation to linear policies: The formulation in Equation 4.4 relates to linear policies with non-linear basis functions (Deisenroth, 2013). However, by making the connection to RKHS's allows a more flexible problem formulation which can incorporate feature maps from a variety of kernel approximation techniques, such as the Nyström method (Yang et al., 2012) and Fourier features (Rahimi and Recht, 2007), which are flexible enough for a variety of modelling problems (Tompkins et al., 2019).

Extension to multivariate control spaces: Although the problem has been formulated for 1-D actions in this section, it could be easily extended to actions composed of multiple independent controls. In that case, the resulting policy parameters would correspond to the concatenation of the weights vector for each independent control function (Alvarez et al., 2011).

4.4 Coordinate-descent Bayesian optimisation for policy search

This section presents a method that approaches the problem in Equation 4.5 from a Bayesian optimisation perspective. The method places a Gaussian process prior over the objective function in Equation 4.5, i.e. $R[\pi_{\mathbf{w}}] = f(\mathbf{w})$, and employs an efficient approach to search in high dimensions. The first part of this section presents how to apply coordinate-descent methods to handle issues that arise when doing BO in high-dimensional search spaces. The second part provides a full description of the policy-search algorithm.

4.4.1 Acquisition function optimisation

Although using finite dimensional approximations for feature maps in the policies representation, the dimensionality of the search space can be still quite large. A Bayesian optimisation method to to solve Equation 4.5 then faces of optimising high-dimensional parameter vectors. Several methods have been proposed in the BO literature to deal with high-dimensional search spaces (Wang et al., 2013; Li et al., 2016). However, those methods usually require a few possibly strong assumptions about the objective function. The framework proposed here instead employs a very simple method, Stochastic Coordinate Ascent (Algorithm 3). This method utilises a random axis selection scheme to optimise the AF over each axis individually, starting from the current best set of parameters. Such an approach biases the search locally, avoiding excessive exploration in a high-dimensional space, and reduces the simultaneous search over several coordinates to several single-coordinate line searches. Despite the bias, problems like racing offer a space of valid solutions that is heavily constrained by factors such as the shape of the track and the car's dynamical limits. Therefore, the reward function's mass is forced to be concentrated in a single island within the search space, which is shared by the global optimum and all the other valid solutions.

Stochastic coordinate ascent (Algorithm 3) was inspired by coordinate descent (CD), a class of algorithms that are one of the oldest in the optimisation literature (Rosenbrock, 1960). CD is based on the idea that a n-dimensional problem can be decomposed into n one-dimensional sub-problems, which makes it suitable for large-scale optimisation problems (Nesterov, 2012). In the CD strategy, each coordinate is updated sequentially by solving the one-dimensional problem using a suitable optimisation algorithm, while all other coordinates are kept fixed. The methods vary in the way the sequence of coordinates is chosen, whether only one coordinate or a block of them is optimised at a time, and whether derivatives are needed or not. Several works focused on characterising the convergence behaviour of these algorithms (Nesterov, 1983; Bertsekas, 2015; Nesterov, 2012). Under some assumptions (e.g. Lipschitz continuity, strong convexity) CD methods achieve linear convergence rate. The most intuitive scheme for this technique would be to optimise the coordinates in an ordered cyclical fashion. However, Powell (1973) showed an example of a non-convex function where this ordered sequential optimisation process cycles without convergence. To avoid this kind of issue, Algorithm 3 adopts a randomised sequence that changes once every coordinate has been optimised.

```
Algorithm 3: StochasticCoordinateAscent
  Input: $h, \mathbf{w}^*, \mathcal{D}$
1 $\mathbf{w} \leftarrow \mathbf{w}^*$
2 $\mathcal{I} = \texttt{randomShuffle}(\{1, \ldots, q\})$
3 for $i \in \mathcal{I}$ do
4     $w_i \leftarrow \operatorname{argmax}_{v \in \mathcal{W}_i} h(w_0, \ldots, w_{i-1}, v, w_{i+1}, \ldots, w_q | \mathcal{D})$
5 return $\mathbf{w}$
```

Algorithm 3 starts the search from the current optimum location. Then it randomly shuffles the sequence of coordinate indices $\{1, \ldots, q\}$ in line 2. In a loop (line 3), the algorithm follows the shuffled sequence $\mathcal{I}$ to optimise one axis after the other (line 4) within its corresponding (bounded) search space $\mathcal{W}_i$. Combined with an acquisition function that avoids excessive exploration and starting the search from the current optimum estimate, this step should ensure that the search for policies remains within a mostly valid neighbourhood. After passing through all the coordinates, it returns the corresponding optimised vector of weights.

Although not using derivatives, as in standard coordinate descent methods (Nesterov, 2012), in practice the approach in Algorithm 3 is still able to achieve similar results. In the development of this framework, preliminary tests tried other CD methods, e.g. the randomised accelerated coordinate descent method (RACDM) (Nesterov, 2012). The preliminary results revealed that their performance is usually not as good as the simple scheme in Algorithm 3. Reasons for that involve issues with escaping saddle points, such as when starting from a previously-observed weights setting, and restrictions in the choice of GP covariance function, which needs to be differentiable for the method to be applicable. For instance, the latter is not the case for Matérn 1 (see Equation 2.21), the best performing covariance function chosen for the experiments in Section 4.5.

4.4.2 The policy-search algorithm

This section proposes an algorithm that improves a valid initial policy so that the robot racer can finish the track in minimal time. The initial policy can be obtained by recording the actions of a simple controller or a human driver. Although related to imitation learning (Sun et al., 2017), there is no need for the initial policy to be optimal or near-optimal, only to be able to successfully complete the race track with the car.

Recording a driving example yields an initial set of points $\mathcal{Q}_s = \{x_i\}_{i=1}^s \subset \mathcal{X}$ and the corresponding vector of observed actions $\mathbf{u}_s = [u_i]_{i=1}^s$, with $u_i \in \mathcal{U},\ \forall i$. The initial weights $\mathbf{w}_0$ can then be determined by minimising a regularised least-squares loss function:

$$\mathbf{w}_0 = \underset{\mathbf{w}\in\mathbb{R}^q}{\operatorname{argmin}} \|\mathbf{u}_s - \hat{\boldsymbol{\phi}}(\mathcal{Q}_s)\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2 \,, \tag{4.6}$$

where $\hat{\boldsymbol{\phi}}(\mathcal{Q}_s) = [\hat{\boldsymbol{\phi}}(x_1), \dots, \hat{\boldsymbol{\phi}}(x_s)]^\mathsf{T}$ is a s-by-q matrix with the features for each x_i on the corresponding row and λ is a regularisation factor, to avoid extreme values for the weights. The solution to this convex optimisation problem can be found analytically by zeroing out the gradient of the loss with respect to $\mathbf{w}$, yielding:

$$\mathbf{w}_0 = [\hat{\boldsymbol{\phi}}(\mathcal{Q}_s)^\mathsf{T}\hat{\boldsymbol{\phi}}(\mathcal{Q}_s) + \lambda\mathbf{I}]^{-1}\hat{\boldsymbol{\phi}}(\mathcal{Q}_s)^\mathsf{T}\mathbf{u}. \tag{4.7}$$

Before starting the policy search with BO, an initial training of the GP is needed to provide an estimate of its hyper-parameters, i.e. the noise variance and the parameters of the covariance function. Due to the vast majority of the search space being, in general, composed of invalid policies, uniform or Latin hypercube random samples could provide too many observations with zero reward, leading to an uninformative GP prior. To avoid this issue, a better initialisation strategy is to sample and execute a set of s samples from a normal distribution $N(\mathbf{w}_0, \mathbf{I}\sigma_0^2)$, with a sufficiently small σ_0^2. This way, the initial observations dataset $\mathcal{D}_s = \{\mathbf{w}_i, R_i\}_{i=1}^s$ can provide a more informative prior over the reward function. In addition, the hyper-parameters can also be re-estimated online during the BO process, as in Wang et al. (2013).

The proposed method, *coordinate-descent Bayesian optimisation* (CDBO), is summarised in Algorithm 4. In lines 1 through 4, it collects the initial set of observations for the GP. In lines 5 through 14, the search for the optimal policy is performed. Line 9 executes the policy parameterised by the current weights. Lines 10 through 12 keep track of the optimum estimate. Line 13 updates the GP dataset.

Algorithm 4: CDBO

```
   Input: $\mathbf{w}_0$: weights of the initial policy
          $\sigma_0^2$: initial samples variance
          $s$: number of initial samples
          $n$: number of laps, $n > s$
1  for $i \in \{1, \dots, s\}$ do
2      Sample $\mathbf{w}_i \sim N(\mathbf{w}_0, \mathbf{I}\sigma_0^2)$
3      $R_i \leftarrow$ Execute $\pi_{\mathbf{w}_i}$
4  $\mathcal{D}_s = \{(\mathbf{w}_i, R_i)\}_{i=1}^{s}$
5  $R^* \leftarrow \max_{i \in \{1,\dots,s\}} R_i$
6  $\mathbf{w}^* \leftarrow \mathrm{argmax}_{i \in \{1,\dots,s\}} R_i$
7  $\mathbf{w}_{s+1} \leftarrow \mathbf{w}_0$
8  for $t \in \{s+1, \dots, n\}$ do
9      $R_t \leftarrow$ Execute $\pi_{\mathbf{w}_t}$
10     if $R_t > R^*$ then
11         $R^* \leftarrow R_t$
12         $\mathbf{w}^* \leftarrow \mathbf{w}_t$
13     $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{\mathbf{w}_t, R_t\}$
14     $\mathbf{w}_{t+1} \leftarrow \texttt{StochasticCoordinateAscent}(h, \mathbf{w}^*, \mathcal{D})$
15 return $\mathbf{w}^*, R^*$
```

Line 14 performs the maximisation of the acquisition function to select the next vector of weights to evaluate. The algorithm proceeds until its budget of n function evaluations is exhausted.

4.5 Experiments

This section presents the experimental evaluation of CDBO and other methods for optimisation in high dimensions. The experiments apply policies to control a race car in simulations using the physics engine of an open-source game, called *Speed Dreams*[2]. This game is based on TORCS, which has been a popular simulated racing environment in the research literature (Wymann et al., 2014). Figure 4.2 presents a screen-shot of the car model used in experiments, a *Spirit 300.* All the tests compared the performance of CDBO against the following methods:

- CMA-ES: Covariance Matrix Adaptation Evolution Strategy (CMA-ES) has been previously applied to reinforcement learning problems involving policy search (Rückstieß et al., 2010).

[2] Speed Dreams: https://sourceforge.net/projects/speed-dreams/

In particular, the experiments here use active CMA-ES(Arnold and Hansen, 2010), with an implementation provided by an open-source library[3].

- REMBO: Random Embeddings Bayesian Optimisation (REMBO) (Wang et al., 2013) uses random embeddings to deal with high-dimensional BO problems via dimensionality reduction. The comparisons involved REMBO with both 5 and 10-dimensional random embeddings.
- Standard BO: The tests apply plain BO using CMA-ES to optimise the acquisition function.

The internal optimisation along each coordinate in CDBO used COBYLA (Powell, 2007), a local derivative-free optimisation algorithm. The implementation was provided by a popular non-linear optimisation library, named *NLopt* (Johnson, 2014).

4.5.1 Setup

The state space of the robot is represented by its position along a given race line normalised by its length, i.e. $x \in [0, 1]$, where 0 corresponds to the start line, increasing to 1 when the robot crosses the finish line. The race line is the centre line of the track, but it could be any other valid trajectory, such as an optimised race line (Xiong, 2010). The control policy actuates the car's acceleration by optimising throttle and braking, which are combined into a single scalar output $u \in [-1, 1]$. Positive values command throttle, while negative ones brake. The car steering was not controlled by the policy, but by an external controller. Optimising only the acceleration control of the car is already a challenging problem and sufficient to assess the capabilities of the optimisation methods in dealing with high-dimensional problems.

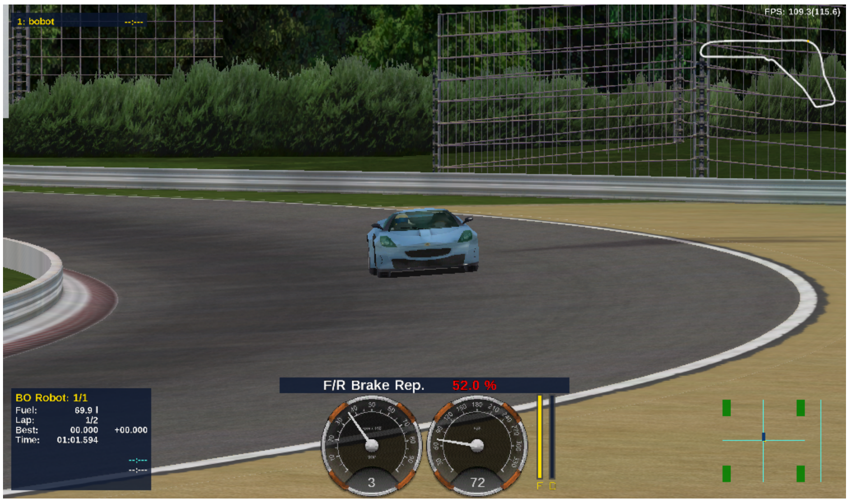
Steering control was performed by a simple proportional-integral (PI) controller, which tries to minimise the distance between the car and the racing line. In this setup, the policy-search algorithm needed to be cautious not to set huge acceleration values for the car at critical parts of the track. Going too fast in curves could destabilise the steering controller and violate the friction limits of the tires.

For features $\hat{\phi}$, the experiments utilised an array of q kernels, i.e.:

$$\hat{\phi}(x) = [k_u(x, \hat{x}_1), \dots, k_u(x, \hat{x}_q)]^\mathsf{T} , \tag{4.8}$$

where the points $\hat{x}_i = \frac{i-1}{q-1}$, $i \in \{1, \dots, q\}$, form a set of q regularly-spaced locations for the basis functions along the track. In this sense, no prior information about where the critical parts of the track

[3] libcmaes: `https://github.com/beniz/libcmaes`



BO Robot: 1/1
Fuel: 69.9 l
Lap: 1/2
Best: 00.000 +00.000
Time: 01:01.594

FIGURE 4.2: Screen-shot of the race car used in the experiments.

are was assumed. A possibly better design for the basis function locations would be to concentrate them around critical portions of the track. Driving through curves, for example, requires significant changes in acceleration. However, knowing where curves are would require a map of the track. These critical points could also be learnt by analysing the initial recorded actions, searching for areas of high variation, but that is left for future work.

Setting k_u to be the RBF squared exponential kernel would lead to the RBF network policy parameterisation, which has already been applied to some reinforcement learning problems (Deisenroth, 2013). The RBF kernel yields very smooth curves and would not be capable of providing fast transitions in the commands profile. Therefore, all the experiments used the *Matérn* class of kernel functions (Equation 2.21) for the policies. In particular, k_u is set as a Matérn 3 kernel:

$$k_u(x, x') = \left(1 + \frac{\sqrt{3}}{l_u}|x - x'|\right) \exp\left(-\frac{\sqrt{3}}{l_u}|x - x'|\right), \tag{4.9}$$

where l_u is a length-scale parameter controlling the smoothness of the curve. The values for l_u were set as approximately the spacing between the basis functions, i.e. $l_u \approx \frac{1}{q-1}$.

Figure 4.3 demonstrates the effect that different length-scales have on the shape of the curve. The length-scale could be numerically optimised before starting the race by minimising the approximation error with respect to the data given by the initial policy. However, fixing the length-scale according to the space between the kernels ensures that the gaps corresponding gaps can be filled without compromising the flexibility of the policy, and avoids over-fitting the length-scale to the initial policy data, which can be

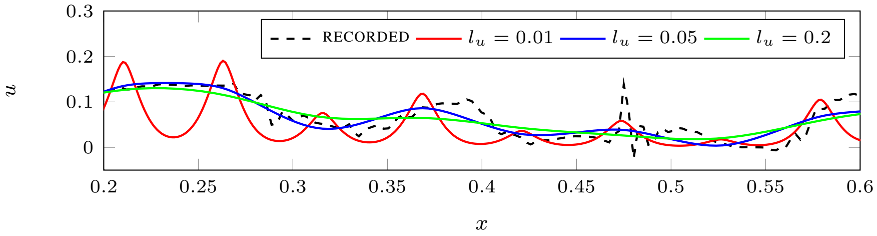


FIGURE 4.3: Detail of the effect of different lengths scales on the fitting of the initial policy for the same kernel placement. The recorded actions are shown in a dashed line. Shorter length scales allow sharper transition, but might compromise interpolation. Longer length scales yield smoother curves, but might compromise flexibility

very flat in some portions. Observing the figure, we also see that a non-stationary kernel could provide better approximations by allowing varying degrees of smoothness along the curve. Despite that, models based on flexible non-stationary kernels, such as the neural network kernel (Rasmussen and Williams, 2006, Ch. 4), usually require large amounts of data (Vasudevan et al., 2009), making them less suitable to represent policies using a relatively small amount of basis functions.

The GP model of the reward function used the Matérn 1 kernel (Equation 2.21) as covariance function. This kernel provides some flexibility to model sharp transitions in the reward function, such as the ones on the borders of the valid region in Figure 4.1, where the reward suddenly drops to zero. The Matérn 1 kernel is also equivalent to the exponential covariance function, with:

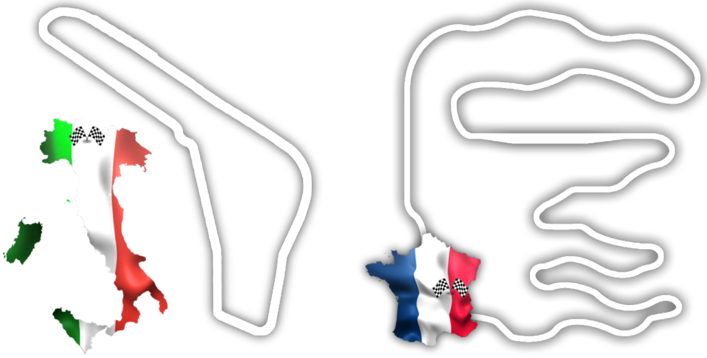
$$k_R(\mathbf{w}, \mathbf{w}') = \sigma_f^2 \exp\left(-\|\mathbf{w} - \mathbf{w}'\|_{\mathbf{\Lambda}}\right), \tag{4.10}$$

where σ_f^2 is a signal variance parameter and $\|\cdot\|_{\mathbf{\Lambda}}$ is the Mahalanobis distance norm (Equation 2.57), with $\mathbf{\Lambda} = \operatorname{diag}(l_i^{-2}), i = 1, \dots, q$, as a length-scales matrix for automatic relevance determination (ARD). The same GP hyper-parameters adaptation scheme proposed in Wang et al. (2013) was applied to all BO methods. In the case of the GP noise model, since the simulations of the physics engine in the game are deterministic, noise variance bounds were set very close to 0. [4]

CDBO and standard BO used the upper confidence bound criterion (see Section 2.3.2) as acquisition function. At each BO round t, candidate policy parameters $\mathbf{w}$ are evaluated by:

$$h(\mathbf{w}|\mathcal{D}_{t-1}) = \mu_{t-1}(\mathbf{w}) + \beta\sigma_{t-1}(\mathbf{w}), \tag{4.11}$$

[4] Small positive values (e.g. 10^{-3}) are allowed, even in a no-noise setting, only to ensure the positive-definiteness of the GP covariance matrix, so that it is always invertible.



(A) Forza (B) Allondaz

FIGURE 4.4: The race tracks for the experiments (Source: *Speed Dreams*)

where $\mu_{t-1}(\mathbf{w})$ is the mean of the GP posterior at $\mathbf{w}$, $\sigma_{t-1}(\mathbf{w})$ is the square root of the GP posterior variance, and β is a parameter controlling the exploration-exploitation trade-off. Performance results were best with small constant values for $\beta \in [0.5, 2]$ with the setting for the experimental comparisons fixed at $\beta = 1$. Due to the high-dimensional search space, a low value for β prevents excessive exploration, improving short-term performance.[5] Experiments with REMBO, however, used the expected improvement acquisition function, since the results in Wang et al. (2013) were achieved using that criterion.

The initial policy demonstration is given by a proportional-integral (PI) controller, whose only task is to drive the car at a constant speed of 15 m/s along the track. An example of such a demonstration for one of the test tracks is shown in Figure 4.3. The initial set of weights $\mathbf{w}_0$ is learnt from the policy given by this initial controller, as in Section 4.4.2. All methods under comparison are informed with the same $\mathbf{w}_0$ as an initial solution.

4.5.2 Results

Experiments were run on two different tracks. Each method under comparison was given a budget of 300 policy evaluations, each corresponding to one lap. In the case of BO and REMBO, the first 10 laps correspond to the initial samples (Section 4.4.2). Before each policy evaluation, to optimise the acquisition function, all versions of BO were allowed a maximum of 50,000 acquisition function evaluations, and had as starting point the best weights found up to that lap. To minimise stochastic effects in the algorithms, each run of 300 laps was repeated 4 times, and the results were averaged.

[5] Similar to the previous chapter, here again the UCB parameter was kept fixed throughout the BO process because prior knowledge about the parameters necessary to keep UCB's theoretical guarantees (Chowdhury and Gopalan, 2017), such as upper bounds on the RKHS norm of the objective function, was not assumed to be available.

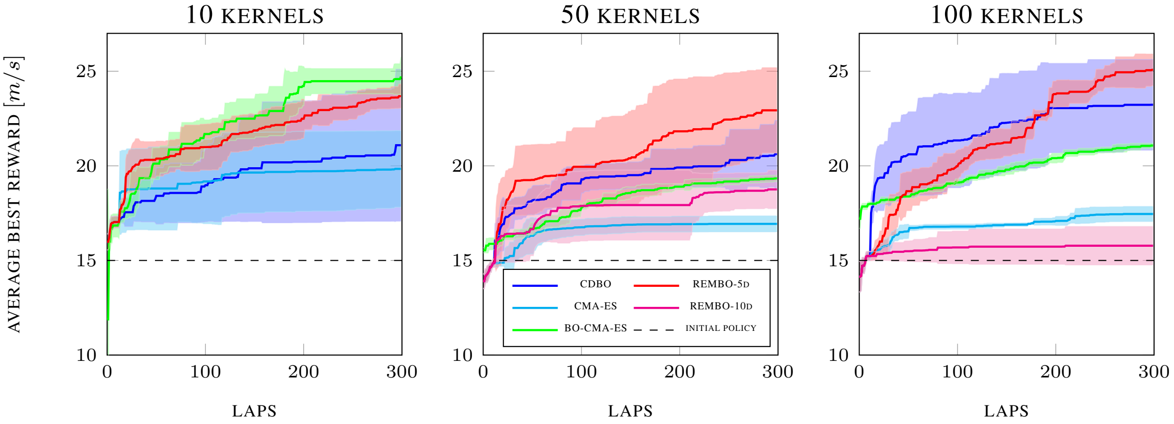
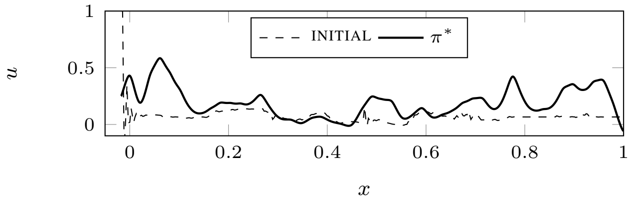


FIGURE 4.5: Performance vs. dimensionality on *Forza*: More kernels allow for better policies, but make the optimisation problem harder. Results were averaged over 4 trials. Shaded areas correspond to 1 standard deviation.

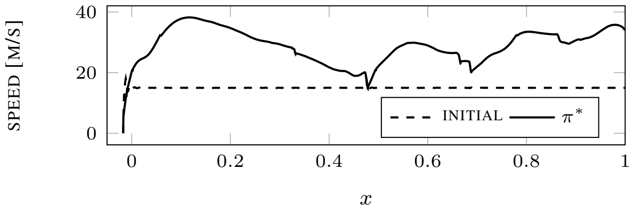
The first track, called *Forza* by the game, was a relatively simple circuit inspired by a real race track in Monza, Italy. This track is 5,784 metres long and 11 metres wide over flat asphalted terrain. The critical parts of this track are the sharp curves at the bottom of Figure 4.4a, which in terms of x value, happen around 40% of the track.

Figure 4.5 presents the overall performance of the analysed methods on *Forza* for policies with different numbers of kernels. The number of kernels corresponds to the number of policy parameters, since each kernel is a basis function, having a weight to be optimised. These experimental results then allow assessing how each method handles the increase in dimensionality, which adds flexibility to the control policy, but makes its optimisation harder. As the plots show, although finding better solutions in the 10 kernels setting, CMA-ES's performance severely degrades with the increase in dimensionality. When combined with BO (in BO-CMA-ES), CMA-ES helps it to improve performance on average. Across individual runs, however, BO-CMAE-ES's behaviour was actually bimodal: sometimes very good, and other times very bad. REMBO wasn't able to achieve good results for any number of kernels in this track. For the 100 kernel test, one of the problems with REMBO is clearly visible: if the used random embedding is not able to capture a (or if there isn't any) relevant subspace, the performance is poor. Overall, it is possible to see that CDBO's performance remains relatively stable with the increase in dimensionality and it is able to find better solutions for this policy search problem.

Figure 4.6a presents the best policy with 50 kernels obtained by CDBO and the resulting speeds the car achieved along the *Forza* track. From Figure 4.6b, it is possible to see that the algorithm adapts itself to speed up on the straight portions of the track and reduce speed close to curves, reducing the lap time.



(A) Control policy



(B) Resulting speeds

FIGURE 4.6: Best policy obtained for track *Forza*

The second test track, *Allondaz*, shown in Figure 4.4b, is a road track with varying elevation along the path and filled with portions of complex geometry. It is 6,356 metres long and 12 metres wide, around the same dimensions as *Forza*. The overall performance of the methods is presented in Figure 4.7. In this track no algorithm was able to achieve average speeds as high as in *Forza*. As increasing the number of kernels beyond 50 did not significantly improve performance, experiments in *Allondaz* used 50 kernels at most.

Similarly to the previous track, CMA-ES is not able to handle the high-dimensionality, and CDBO still maintains a consistent performance throughout the increase in dimensionality, demonstrating the capabilities of the method in high dimensions. Also, it's possible to see in all the tests that CDBO can achieve an even better result, if it is allowed to run for more iterations. REMBO achieved better results than the previous track, but it is still outperformed by BO-CMA-ES and CDBO. One interesting detail about REMBO's performance is that, after some iterations it does not improve any more, which means that it reached the optimum for the subspace used, and the global optimum is not in that subspace.

Compared to standard BO, another interesting feature of CDBO can be seen in Table 4.1, which shows the runtime for each experiment (300 laps) on *Forza*. It is possible to see that with the increase in dimensionality, standard BO with CMA-ES (BO-CMA-ES) significantly increases in runtime when compared to the other methods. REMBO and CDBO maintain a low runtime through all the different problem dimensions. In particular, for the 100-kernels case, each internal acquisition function optimisation made by CMA-ES for BO-CMA-ES was taking seconds to complete, while the much simpler CD method

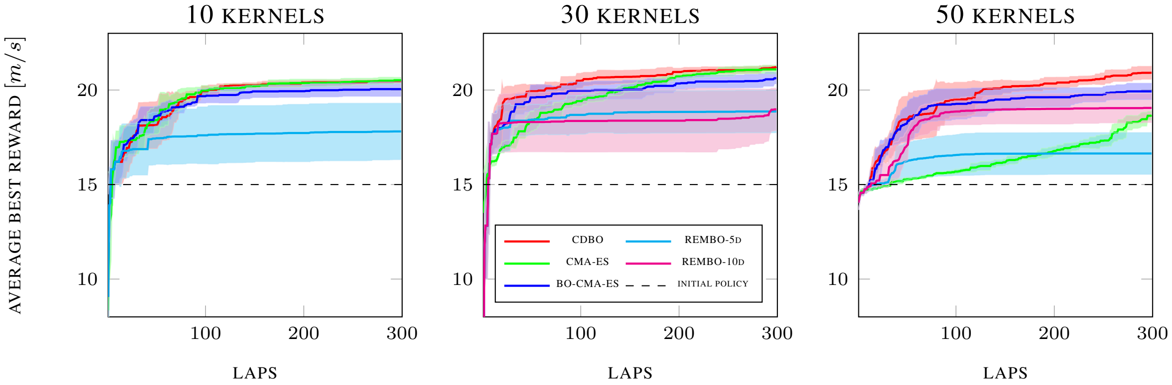


FIGURE 4.7: Performance vs. dimensionality on *Allondaz*: The plots show how the performance of each algorithm is affected by the increase in dimensionality. Results were averaged over 4 trials. Shaded areas correspond to 1 standard deviation.

TABLE 4.1: Average runtime (in seconds) on *Forza*

Method/Dimensions	10	50	100
CDBO	127	281	318
CMA-ES	173	234	252
BO-CMA-ES	553	680	3048
REMBO-5d	154	242	257
REMBO-10d	-	244	299

would do the same in fractions of a second. So, even if running standard BO using CMA-ES internally to optimise the acquisition function leads to similar average performance when compared to CDBO, the runtime for the standard BO method is about 10 times longer for the 100-dimensional case, which highlights the efficiency of CDBO for high-dimensional problems.

4.6 Summary

This chapter presented a method to perform Bayesian optimisation in high-dimensional search spaces. The proposed method was applied to optimise control policies to allow a robot to complete a given race track faster. This problem is an instantiation of a more general class of problems involving delayed rewards and costly policy evaluations. The proposed coordinate-descent Bayesian optimisation method guides the exploration of the parameter space towards the optimal policy. By making use of ideas from randomised coordinate descent methods, optimising a function one dimension at a time in a randomised sequence, and by starting the search from a valid, though sub-optimal, initial solution, CDBO is able to be effective in the high-dimensional acquisition function optimisation sub-problem. Experiments with a car racing simulator demonstrated that this relatively simple approach is able to outperform other state-of-the-art black-box optimisation and BO methods in complex scenarios, sometimes at a fraction of the runtime.

As future work, the modelling approach can be improved to deal with policies over higher-dimensional state and action spaces and to work together with other conventional motion planning algorithms. Adaptive sampling schemes for coordinate-wise optimisation (Namkoong et al., 2017), which go beyond the uniform coordinate sampling approach taken in this chapter, can also be investigated. Lastly, the game physics engine does not simulate different sources of randomness in the dynamics of a real environment, which could affect the performance of control policies. Therefore, robustness to uncertainty in control and localisation is a desirable goal to be incorporated into the policy optimisation. The next chapter presents a step towards that direction.

CHAPTER 5

Smooth navigation under localisation uncertainty

5.1 Introduction

Mobile robots have been successful in many field applications, such as mining (Maekawa et al., 2010; Brown, 2012), planetary exploration (Lang et al., 2007; Carsten et al., 2007), agriculture (Underwood et al., 2017), and environmental monitoring (Marchant and Ramos, 2012), to name a few. In all these applications, robots face environments with physical characteristics that are *a priori* unknown and can heavily affect performance. In the case of ground robots, terrain roughness can affect the ability of a robot to navigate and even cause damage to its on-board hardware due to excessive vibration (Souza et al., 2014). To aid in these problems, methods to enable the robot to automatically learn terrain properties from its sensory data have been proposed in the literature (Lang et al., 2007; Souza et al., 2014). However, such methods usually assume that localisation is accurate enough, without dealing with its inherent uncertainty.

Uncertainty in localisation can mislead learning algorithms with noise-corrupted estimates of the location where measurements are taken. Robot navigation heavily depends on knowing where the robot is with respect to a given reference path and its surroundings. As a result, localisation inaccuracies can lead a robot into areas that are unsafe for navigation. In the case of rough terrain modelling (Souza et al., 2014), for instance, the robot might be led into areas of excessively high roughness.

This chapter presents a method that allows a robot to actively learn terrain traversability (Martin et al., 2013) models from proprioceptive data while keeping itself safe by considering both model and localisation uncertainty. The particular contributions[1] are the following:

[1] The contributions in this chapter have been previously presented at the 18th International Symposium on Robotics Research as Oliveira et al. (2017b). At the time of this thesis writing, the final proceedings of the event are still being edited by Springer, the publisher, but a preprint of the paper is available as Oliveira et al. (2017a).

- a framework to account for input uncertainty in Bayesian optimisation for mission planning; and
- an adaptation of the DUCB acquisition function (Section 2.3.2) to the context of exploration under uncertain localisation.

The proposed method applies the expected Gaussian process model, discussed in Section 2.2.4, as a prior for the BO framework. This kind of GP model allows BO to take into account noise in both the execution of a query and in the location estimates in the observations dataset. As an application, experiments evaluate the proposed method in the task of learning a model of terrain roughness from experienced vibration data (Souza et al., 2014).

The remainder of this chapter is organised as follows. The next section reviews relevant prior work in the areas of terrain modelling and Bayesian optimisation. Section 5.3 presents the proposed methodology for Bayesian optimisation under localisation uncertainty. Results for experiments in simulation and with a physical robot are presented in Section 5.4. Finally, Section 5.5 concludes the chapter and proposes a few directions for future work.

5.2 Related Work

Traversability metrics estimate how hard it is for a vehicle to traverse a certain terrain. In robotics, terrain traversability is usually estimated from exteroceptive sensor data, such as LIDAR range measurements (Nordin, 2012) and stereo vision information (Ho et al., 2013). Among the different kinds of metrics, terrain roughness can play an important role in planning how a robot should navigate on the terrain. Methods to learn terrain roughness using image data usually rely on learning image classification models (Komma et al., 2009), which generally do not provide a measure of how uncertain they are about their estimates and are computationally intensive. Martin et al. (2013), however, propose using vehicle experience data to learn a Gaussian process regression model to predict terrain traversability with uncertainty estimates. Though restrictive when compared to exteroceptive sensing information (Ho et al., 2013), proprioceptive sensing can still be a very viable option for traversability estimation. For instance, many robot platforms are not equipped with stereo vision or 3D LIDAR sensors, or do not possess the computational resources to process this kind of information on-board online.

Souza et al. (2014) presented a method to learn a GP model for terrain roughness from vehicle experience. The authors applied Bayesian optimisation in an active perception approach to reduce experienced vibration during navigation while learning the model from acceleration measurements. In the BO framework, the terrain roughness model is learnt online as the algorithm drives the robot around selecting locations to visit by balancing a trade-off between exploration and exploitation (Brochu et al., 2010; Souza et al., 2014). Nevertheless, BO usually considers deterministic query locations within its search space, as BO is typically used in problems where a given set of parameters have to be optimised (Snoek et al., 2012; Wilson et al., 2014).

Nogueira et al. (2016) presented a method to apply BO to problems where the execution of a query is noisy, such as robotic grasping, and one seeks a solution that is robust under repeatable trials. The authors propose querying BO's surrogate model using a Gaussian distribution by applying the unscented transform (Wan and van der Merwe, 2000). In this, even with uncertainty in execution, the algorithm chooses grasp configurations that are both optimal and robust to execution noise. Despite that, Nogueira et al. (2016) still apply a conventional GP model as a prior for BO. In navigation problems, the robot is usually able to obtain a probability distribution estimating its location, which is given by a localisation system. Making use of a GP model that takes into account such distributions as inputs should then allow BO to learn a better model of the true underlying objective function.

To deal with uncertain location estimates in observations, besides the usual noisy outputs, BO needs a statistical model that also considers partial observations of the query inputs. In BO the most common approach to modelling the objective function is using Gaussian process regression (Section 2.2). In the case of observations at uncertain inputs, Girard (2004) proposed methods to propagate input uncertainty through a GP model, developing analytical approximations to compute covariance function values between inputs represented by Gaussian distributions. Another approach is to assume that location estimates are corrupted by i.i.d. Gaussian noise, similar to usual assumptions on output observations in standard GP regression. That is the approach taken by Mchutchon and Rasmussen (2011) with a method that linearises the effect of input noise on the output of the model via a first-order Taylor expansion. A final possible approach is to use GP latent variable models (Titsias and Lawrence, 2010) to consider the true unknown inputs of the GP model as latent variables. Damianou et al. (2016) proposed a variational inference framework to compute the posterior of the GP model under a set of assumptions about the input distributions. None of the mentioned GP methods, however, has been previously applied to the BO context, where the GP model is built in an online fashion as the algorithm proceeds.

5.3 Bayesian optimisation under localisation uncertainty

This section presents a framework for Bayesian optimisation under uncertain inputs. Here *inputs* refer to points in the objective function $f : \mathbb{R}^d \to \mathbb{R}$ (Equation 2.30) input space, i.e. any $\mathbf{x} \in \mathbb{R}^d$. Input noise can affect both the execution of a query to the objective function and the estimates of the locations where these observations are taken, as further explained below. Hence one first needs to extend BO's surrogate model to incorporate uncertainty in the inputs. The method proposed here uses a Gaussian process regression model with uncertain inputs (Section 2.2.4) as a prior. Using this GP model as a prior allows BO to take into account localisation uncertainty affecting both the next query location and the past locations where previous observations were collected. The remainder of this section starts by describing the different sources of location noise that affect the BO process, followed by a review of the GP modelling approach and ending with the proposed acquisition function to guide BO.

5.3.1 The effects of input noise into the BO process

To extend BO to the problems involving uncertain inputs in robotics, let us first consider how input noise affects the BO process. As an illustration, consider the example in Figure 5.1. In robotics problems involving location estimation, input noise basically falls into two categories: *localisation* noise and *execution* noise. The first refers to noise affecting the location estimate provided by the robot's localisation system. This noise is usually caused by imperfections in motion estimation and unknown disturbances affecting other kinds of sensors, such as global positioning system (GPS) devices. The second type, execution noise, is the combined effect of everything affecting the execution of the robot's path to a given target location, such as localisation noise, uncertain motion dynamics, etc.

Recalling Section 2.3, when a BO algorithm selects a query location $\mathbf{x}_t$, the usual assumption is that it will observe $y_t = f(\mathbf{x}_t) + \nu_t$, where $\nu_t \sim N(0, \sigma_\nu^2)$ represents Gaussian observation noise. However, in robotics applications, the actual location where the observation will be taken is:

$$\tilde{\mathbf{x}}_t = \mathbf{x}_t + \boldsymbol{\epsilon}_t \ , \tag{5.1}$$

where $\boldsymbol{\epsilon}_t \sim P_{\boldsymbol{\epsilon}|\mathbf{x}_t}$ represents execution noise following a (usually unknown) distribution $P_{\boldsymbol{\epsilon}|\mathbf{x}_t}$. Then, instead, we have:

$$y_t = f(\tilde{\mathbf{x}}_t) + \nu_t = f(\mathbf{x}_t + \boldsymbol{\epsilon}_t) + \nu_t \ , \tag{5.2}$$

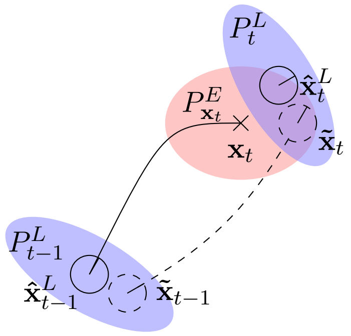


FIGURE 5.1: At time $t-1$, the robot is estimated to be at some $\tilde{\mathbf{x}}_{t-1} \sim P^L_{t-1}$ with mean $\hat{\mathbf{x}}^L_{t-1}$. The robot is then sent to target location $\mathbf{x}_t$. However, due to uncertainty in the query execution, represented by $P^E_{\mathbf{x}_t}$, the robot actually ends up at another location $\tilde{\mathbf{x}}_t$, whose belief distribution, according to the localisation system, is represented by P^L_t. The robot's true locations and true path are indicated by the dashed lines.

as the observation the algorithm receives. In addition, the true query location $\tilde{\mathbf{x}}_t$ cannot be estimated exactly. After the query, a localisation system, instead, provides an independent estimate $\hat{\mathbf{x}}^L_t$ for the true query location $\tilde{\mathbf{x}}_t$ from observed sensor data $\mathbf{z}_t$ (e.g. GPS and odometry estimates), which is such that:

$$\hat{\mathbf{x}}^L_t = \tilde{\mathbf{x}}_t + \boldsymbol{\eta}_t\,, \tag{5.3}$$

where $\boldsymbol{\eta}_t \sim P_{\boldsymbol{\eta}|\mathbf{z}_t}$ represents localisation noise. Unlike the case for the execution noise, it is plausible to assume that the localisation noise distribution $P_{\boldsymbol{\eta}|\mathbf{z}_t}$ is known, since classical localisation systems are based on Bayesian filtering approaches (Thrun et al., 2006).

To account for the different sources of noise mentioned above, the true query location $\tilde{\mathbf{x}}_t$ can be interpreted as a random variable, which is distributed as $\tilde{\mathbf{x}}_t|\mathbf{x}_t \sim P^E_{\mathbf{x}_t}$, when conditioned on the target location $\mathbf{x}_t$ chosen by the BO algorithm. For example, if execution noise follows a zero-mean Gaussian $N(\mathbf{0}, \boldsymbol{\Sigma}_\mathbf{x})$ with location-dependent covariance matrix $\boldsymbol{\Sigma}_\mathbf{x}$, we have $P^E_\mathbf{x} = N(\mathbf{x}, \boldsymbol{\Sigma}_\mathbf{x})$. After the query, the localisation system provides another probability distribution P^L_t estimating the robot's true query location $\mathbf{x}_t$. Then, when conditioned on the sensor data $\mathbf{z}_t$ observed by the localisation system, we can model the true query location as $\tilde{\mathbf{x}}_t|\mathbf{z}_t \sim P^L_t$. If, for example, the localisation system is an extended Kalman filter (Moore and Stouch, 2014), we have $P^L_t = N(\hat{\mathbf{x}}^L_t, \boldsymbol{\Sigma}^L_t)$, for some known covariance matrix $\boldsymbol{\Sigma}^L_t$ informed by the system.

In a setting where execution noise is significant, standard BO may utilise $\hat{\mathbf{x}}^L_t$ as the input to be added with the corresponding outcome y_t into the observations dataset $\mathcal{D}_t$, instead of the target location $\mathbf{x}_t$.

However, if the localisation noise level is also significant, or if $P_{\boldsymbol{\eta}|\mathbf{z}_t}$, and consequently P_t^L, is multi-modal, $\hat{\mathbf{x}}_t^L$ by itself might not be a good estimator for $\tilde{\mathbf{x}}_t$. The next section describes a principled GP modelling approach to take input noise into account when performing inference.

5.3.2 Using GP models taking probability measures

The expected GP model, as defined in Section 2.2.4, allows deriving a BO framework that takes into account both localisation and execution noise in the decision-making process. In that model, given any two random locations $\tilde{\mathbf{x}} \sim P$ and $\tilde{\mathbf{x}}' \sim P'$, the expected GP covariance is defined as a function of each variable's probability distribution by:

$$\hat{k}(P, P') := \mathbb{E}[k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')] = \int\int k(\mathbf{x}, \mathbf{x}') \, \mathrm{d}P(\mathbf{x}) \, \mathrm{d}P(\mathbf{x}') \,, \tag{5.4}$$

where k is the original GP covariance function, which assumes noise-free inputs. If the two inputs correspond to the same random variable $\tilde{\mathbf{x}} \sim P$, the expected covariance is then given by:

$$\hat{k}(P, P) := \mathbb{E}[k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})] = \int k(\mathbf{x}, \mathbf{x}) \, \mathrm{d}P(\mathbf{x}) \,. \tag{5.5}$$

In this framework, one can use the full information of the localisation distributions by adding them directly to the GP dataset $\mathcal{D}_n = \{(P_t^L, y_t)\}_{t=1}^n$.[2] Yet querying the GP model with the true $P_\mathbf{x}^E$ for each candidate target location $\mathbf{x}$ is not possible. The mapping $\mathbf{x} \mapsto P_\mathbf{x}^E$ is generally unknown for it is hard to estimate the execution noise distribution. Instead, a model $\hat{P}_\mathbf{x}$ can be used as a replacement for $P_\mathbf{x}^E$ when querying the GP prior during the acquisition function optimisation. For instance, one may assume $\hat{P}_\mathbf{x} = N(\mathbf{x}, \hat{\mathbf{\Sigma}}_\mathbf{x})$ for a given model mapping $\mathbf{x} \mapsto \hat{\mathbf{\Sigma}}_\mathbf{x}$, or simply $\hat{\mathbf{\Sigma}}_\mathbf{x} = \hat{\mathbf{\Sigma}}$ as a constant. Hence, for a given target location $\mathbf{x}$, the GP can approximate the distribution of $f(\tilde{\mathbf{x}})$, where $\tilde{\mathbf{x}}|\mathbf{x} \sim P_\mathbf{x}^E$, as:

$$f(\tilde{\mathbf{x}})|\mathcal{D}_n, \mathbf{x} \overset{\text{approx.}}{\sim} N(\hat{\mu}_n(\hat{P}_\mathbf{x}), \hat{\sigma}_n^2(\hat{P}_\mathbf{x})) \,, \tag{5.6}$$

where:

$$\hat{\mu}_n(\hat{P}_\mathbf{x}) = \hat{\mathbf{k}}_n(\hat{P}_\mathbf{x})^\mathsf{T} (\hat{\mathbf{K}}_n + \sigma_\nu^2 \mathbf{I})^{-1} \mathbf{y}_n \,, \tag{5.7}$$

$$\hat{\sigma}_n^2(\hat{P}_\mathbf{x}) = \hat{k}(\hat{P}_\mathbf{x}, \hat{P}_\mathbf{x}) - \hat{\mathbf{k}}_n(\hat{P}_\mathbf{x})^\mathsf{T} (\hat{\mathbf{K}}_n + \sigma_\nu^2 \mathbf{I})^{-1} \hat{\mathbf{k}}_n(\hat{P}_\mathbf{x}) \,, \tag{5.8}$$

[2] A dataset with distributions as elements is only an abstract representation which allows for a general treatment of different types of location distributions. In practice, having a dataset comprised of, for example, Gaussian distributions $P_t^L = N(\hat{\mathbf{x}}_t^L, \mathbf{\Sigma}_t^L)$ and observation values y_t simply translates to having $\mathcal{D}_n = \{(\hat{\mathbf{x}}_t^L, \mathbf{\Sigma}_t^L, y_t)\}_{t=1}^n$.

with $\hat{\mathbf{k}}_n(\hat{P}_{\mathbf{x}}) := [\hat{k}(\hat{P}_{\mathbf{x}}, P_1^L), \dots, \hat{k}(\hat{P}_{\mathbf{x}}, P_n^L)]^\mathsf{T}$ and $[\hat{\mathbf{K}}_n]_{ij} = \hat{k}(P_i^L, P_j^L)$. Considering $\hat{P}_{\mathbf{x}}$, at each iteration t, the resulting BO loop should select as target location:

$$\mathbf{x}_t = \operatorname*{argmax}_{\mathbf{x} \in \mathcal{S}} h(\hat{P}_{\mathbf{x}} | \mathcal{D}_{t-1}) \ , \tag{5.9}$$

which is computed over the GP posterior using equations 5.7 and 5.8. Under this formulation, $\hat{P}_{\mathbf{x}}$ provides BO with an estimate of how execution noise affects the utility of a query at a given target location.

```
Algorithm 5: UIBO
  Input:
      $\mathcal{S}$: search space
      $n$: total number of iterations

1 for $t \in \{1, \dots, n\}$ do
2     $\mathbf{x}_t = \operatorname*{argmax}_{\mathbf{x} \in \mathcal{S}} h(\hat{P}_{\mathbf{x}} | \mathcal{D}_{t-1})$
3     $y_t \leftarrow$ Observe $f$ at $\tilde{\mathbf{x}}_t | \mathbf{x}_t \sim P^E_{\mathbf{x}_t}$
4     $P^L_t \leftarrow$ Estimate $\tilde{\mathbf{x}}_t$ from sensor data $\mathbf{z}_t$
5     $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(P^L_t, y_t)\}$
  Result: $\hat{\mu}_n, \hat{\sigma}^2_n$
```

```
Algorithm 6: Standard BO
  Input:
      $\mathcal{S}$: search space
      $n$: total number of iterations

1 for $t \in \{1, \dots, n\}$ do
2     $\mathbf{x}_t = \operatorname*{argmax}_{\mathbf{x} \in \mathcal{S}} h(\mathbf{x} | \mathcal{D}_{t-1})$
3     $y_t \leftarrow$ Sample $f(\mathbf{x}_t)$
4
5     $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$
  Result: $\mu_n, \sigma^2_n$
```

The proposed uncertain-inputs Bayesian optimisation (UIBO) algorithmic framework is summarised in Algorithm 5 with a comparison to standard BO on the side (Algorithm 6). As a first difference, UIBO selects target locations $\mathbf{x} \in \mathcal{S}$ by maximising an acquisition function using the predictions of the expected GP for $\hat{P}_{\mathbf{x}}$ in line 2. The main difference, however, is in line 4. UIBO takes the estimate given by P^L_t alongside the observation y_t as an entry to the expected GP's dataset $\mathcal{D}_t$. Standard BO, instead, uses the target location $\mathbf{x}_t$ as an input to its GP dataset (line 5). Both algorithms run for a given total amount of iterations n. In the end, each method obtains a GP model that can be used for future tasks. By Equation 5.4, the expected covariance function is also valid for deterministic inputs, so that the final expected GP model can also be queried with point estimates by conventional planning algorithms.

Notice that $\hat{P}_{\mathbf{x}}$ is only applied to query the acquisition function. Determining the true $P^E_{\mathbf{x}}$ is a challenging task, beyond the scope of this work. However, the next chapter will present theoretical guarantees for a uncertain-inputs BO algorithm using an imperfect model for $\hat{P}_{\mathbf{x}}$.

5.3.3 A DUCB acquisition function for BO under localisation uncertainty

As formulated in Equation 5.9, the acquisition function needs to be able to handle probability distributions as inputs by using the expected GP model. For some acquisition functions, such as UCB (Srinivas et al., 2012), it can be as straight forward as replacing $\mathbf{x}$ by $\hat{P}_{\mathbf{x}}$ and using the corresponding GP posterior mean $\hat{\mu}_{t-1}(\hat{P}_{\mathbf{x}})$ (Equation 5.7) and variance $\hat{\sigma}^2_{t-1}(\hat{P}_{\mathbf{x}})$ (Equation 5.8). For others, some modifications need to be made. Recall that the GP model in Section 5.3.2 provides a Gaussian approximation for $p(f(\tilde{\mathbf{x}})|\mathcal{D}_{t-1}, \mathbf{x})$, which is actually not Gaussian. The UCB criterion only needs that $\beta\hat{\sigma}_{t-1}(\hat{P}_{\mathbf{x}})$ remains as a valid confidence bound for $|f(\tilde{\mathbf{x}}) - \hat{\mu}_{t-1}(\hat{P}_{\mathbf{x}})|$, which can be tuned by properly selecting large enough values for β. On the other hand, if one were to try expected improvement (see Section 2.3.2) as an acquisition function for uncertain-inputs BO, the Gaussian assumption might be far enough from truth so that one would have to develop an appropriate approach to tune the acquisition function.

To guide a robot using BO, the distance between query locations should be considered, so that BO does not take the robot too far through unknown regions and also so that other budget constraints are factored in, such as a limited energy supply. The DUCB acquisition function then provides a simple but effective approach for BO in this case. A sensible adaptation for the distance-penalty term to the uncertain-inputs case is to use the distance between the candidate target location and the mean of the distribution of the last sampled location ($\mathbb{E}_{P_{t-1}}[\tilde{\mathbf{x}}^L_{t-1}] = \hat{\mathbf{x}}^L_{t-1}$). The resulting DUCB acquisition function is given by:

$$h_{\text{DUCB}}(\hat{P}_{\mathbf{x}}|\mathcal{D}_{t-1}) = -\hat{\mu}_{t-1}(\hat{P}_{\mathbf{x}}) + \beta\hat{\sigma}_{t-1}(\hat{P}_{\mathbf{x}}) - \zeta\|\mathbf{x} - \hat{\mathbf{x}}^L_{t-1}\|_2 \tag{5.10}$$

for minimisation objectives. In the proposed BO framework, this acquisition function can guide a robot learning models from noisy observations and localisation data, while taking into account both sources of uncertainty. The experiments in the next section present a practical application in the case of terrain traversability modelling.

5.4 Experiments

This section presents experimental results obtained with the proposed method for uncertain-inputs Bayesian optimisation (UIBO) in the presence of localisation uncertainty. In all cases, the DUCB acquisition function guided the BO exploration process. For framework comparisons, two other approaches were considered. The first is standard BO, which does not consider any uncertainty in the location estimates.

The second approach is unscented Bayesian optimisation (UBO) (Nogueira et al., 2016), which considers execution noise by means of the unscented transform (Wan and van der Merwe, 2000), but uses a point estimate for the location of each observation. Experiments were performed both in simulation and with a physical robot in an outdoor scenario.

In terms of GP settings, although the *expected GP* formulation (Section 5.3.2) is general enough to use different types of covariance functions and input distributions, both simulation and physical experiments employed the squared-exponential covariance function and assumed Gaussian noise. The squared-exponential kernel, also known as the *Gaussian kernel*, is a universal approximator for continuous functions (Sriperumbudur et al., 2011), so that GP models with this kernel are general enough. In addition, Dallaire et al. (2011) provide an analytical derivation for the expected squared-exponential covariance function in the case of Gaussian inputs. The standard squared exponential function is given by:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^\mathsf{T}\mathbf{L}^{-1}(\mathbf{x}-\mathbf{x}')\right) , \tag{5.11}$$

where σ_f^2 is a signal-variance parameter and $\mathbf{L}$ is a squared length-scales diagonal matrix. In this case, the *expected* squared-exponential is derived as:

$$\hat{k}(N(\hat{\mathbf{x}}_i, \mathbf{\Sigma}_i), N(\hat{\mathbf{x}}_j, \mathbf{\Sigma}_j)) = \frac{\sigma_f^2 \exp\left(-\frac{1}{2}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)^\mathsf{T}(\mathbf{L}+\mathbf{\Sigma}_i+\mathbf{\Sigma}_j)^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)\right)}{|\mathbf{I}+\mathbf{L}^{-1}(\mathbf{\Sigma}_i+\mathbf{\Sigma}_j)(1-\delta_{ij})|^{1/2}} , \tag{5.12}$$

where δ_{ij} denotes the Kronecker delta, and σ_f^2 and $\mathbf{L}$ are the same parameters as described above for the original kernel. These GP hyper-parameters were fixed in simulations, and learnt online in the experiments with a physical robot.

As model evaluation metrics, the root mean square error (RMSE) and weighted RMSE (WRMSE) (Marchant and Ramos, 2012) values were computed from each method's GP posterior mean (μ) and the ground-truth values of f deterministically queried over a uniform grid covering the entire search space. In these experiments, the WRMSE emphasises the reconstruction error in areas of lower vibration, being computed as:

$$\text{WRMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left((\mu(\mathbf{x}_i) - f(\mathbf{x}_i))\frac{\max_j f(\mathbf{x}_j) - f(\mathbf{x}_i)}{\max_j f(\mathbf{x}_j) - \min_j f(\mathbf{x}_j)}\right)^2} . \tag{5.13}$$

This metric is a better performance indicator for this chapter's task than the plain RMSE, since the focus is on finding areas that cause lower vibration to the robot's body.

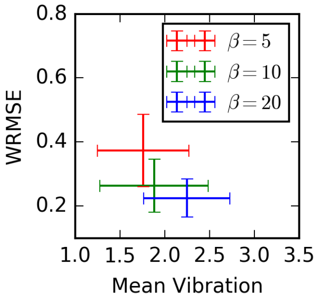


FIGURE 5.2: Effect of different values for the DUCB parameter β using $\zeta = 1$ on UIBO's performance.

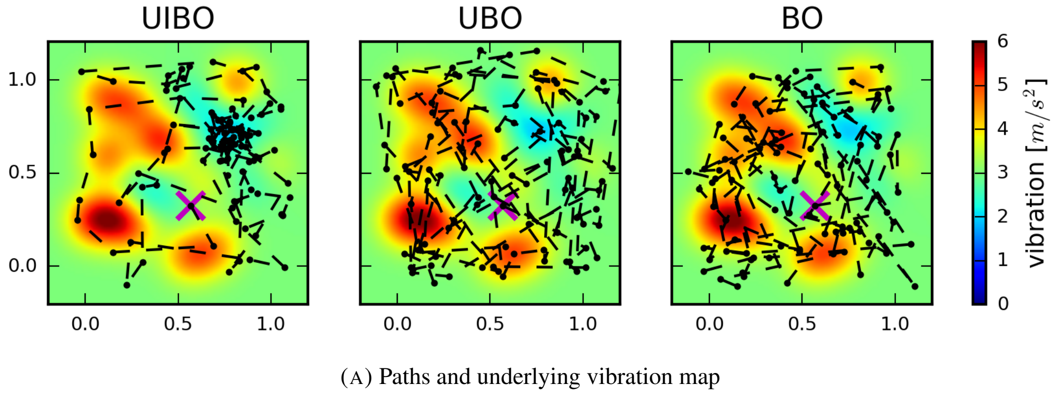
5.4.1 Simulations

Simulations used synthetic functions as a model for terrain roughness to be learnt by BO. The task is to find areas of low roughness while avoiding areas that cause excessive vibration, which could damage the robot in reality. The learnt GP model should then be more accurate over areas of lower terrain roughness, which are more interesting in practice.

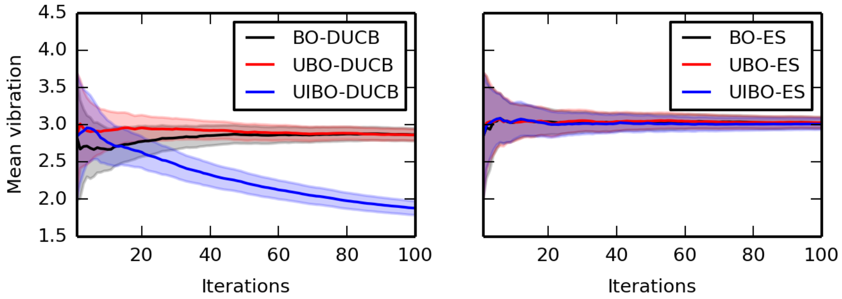
To simulate a variety of terrains, each test run randomly generates a different synthetic function to provide ground-truth vibration levels at each location on the terrain. The function is generated by using an element from the reproducing kernel Hilbert space (Section 2.4) associated with the noise-free GP covariance k, which is combined with a constant mean to ensure that vibration values are positive.

To simulate noise, execution noise is sampled from a Gaussian $N(0, \sigma_E^2 \mathbf{I})$ and localisation noise from another Gaussian $N(0, \sigma_L^2 \mathbf{I})$. The approximate querying distribution $\hat{P}_{\mathbf{x}}$ applied by UIBO is set also as a Gaussian $N(\mathbf{x}, \sigma_{\mathbf{x}}^2 \mathbf{I})$ and is the same distribution that is applied by unscented BO (UBO). The experiments were set with $\sigma_E = \sigma_L = 0.07$ and $\sigma_{\mathbf{x}} = 0.1$. Besides input noise, observation noise was generated with $\sigma_\nu = 0.1$.

As BO settings, DUCB's parameters were set as $\beta = 10$ and $\zeta = 1$, which were manually tuned to minimise both the mean experienced vibration and the final model's WRMSE, for the three versions of BO (see Figure 5.2 for a few results). The hyper-parameters for the GP models were fixed and identical for each BO method. For fair comparisons, the methods based on deterministic-inputs GP models used the equivalent noise-free version of the expected GP covariance function and the same DUCB parameters, since these do not affect the GP model.



(A) Paths and underlying vibration map



(B) Mean experienced vibration

FIGURE 5.3: Simulation results for standard BO, Unscented BO (UBO) and Uncertain-Inputs BO (UIBO) methods for random functions modelling terrain-induced vibration. (a) presents the true paths taken by each BO method using *DUCB* over the true vibration map in one of the test trials. The markers along the paths indicate the locations where observations were taken. The big "X" mark (in magenta) indicates the starting location, which is the same for the three methods at each trial. The plots in (b) present performance results in terms of mean intensity of the experienced vibration. The results were averaged over 30 trials, each with different maps, and the shaded areas correspond to one standard deviation.

The simulations also assessed the performance of maximum entropy search (ES) as heuristics for BO. These are not to be confused with methods based on the entropy of the optimum location distribution, such as Hennig and Schuler (2012). Maximum entropy search in this chapter refers to heuristics seeking only to reduce the entropy of the GP posterior as a whole, choosing to visit areas of high uncertainty, as in Section 2.3.3. ES methods should provide a baseline for comparisons, since they do not consider any estimate of the expected vibration nor distance penalties in their search. Following ES guidance, the robot should obtain an accurate model, but experience high amounts of vibration and long paths.

Figure 5.3 presents results obtained in simulation. As seen in the test case in Figure 5.3a, standard BO and UBO using DUCB end up having a much more exploratory behaviour due to noise. Excessive exploration leads the robot through possibly-damaging long paths over the terrain. UIBO, on the other hand, is able to focus its exploration on areas that induce less vibration. Overall, Figure 5.3b shows that UIBO's performance significantly improves over iterations, finding areas of low vibration for the robot to navigate through. Both the standard and the unscented versions of BO ended up having an average performance much more similar to pure entropy search (ES). As entropy-based methods are purely exploratory, there is no improvement in terms of mean experienced vibration, since the algorithm is not optimising for that. In the case of BO-DUCB and UBO-DUCB, the uncertainty in localisation corrupts the location estimates in the observations dataset, and this effect is not taken into account by the GP model. As a result, both methods are misled into areas close to locations that they previously observed, ending up into this more exploratory behaviour, instead of exploiting previous information.

Table 5.1 presents a summary of the final performance in terms of different metrics for each method.

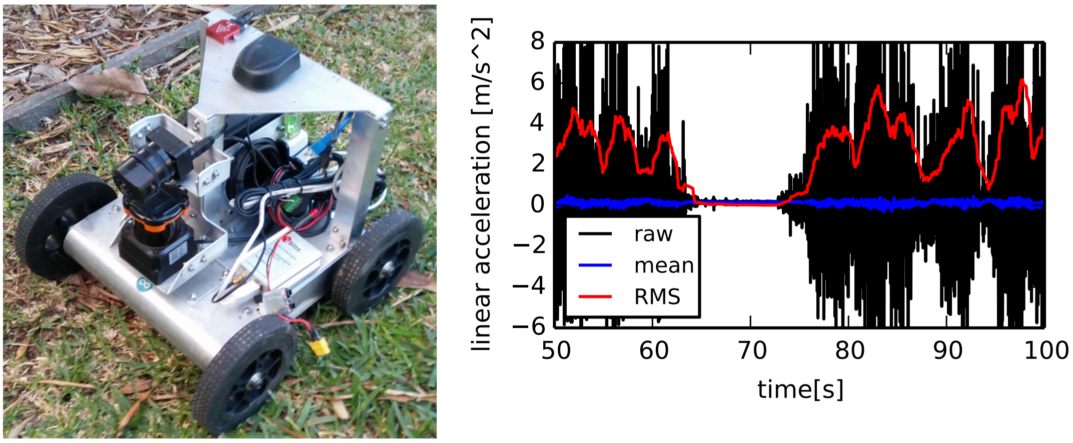
As seen in Table 5.1, UIBO is able to outperform the other methods when using DUCB, obtaining a good model, in terms of WRMSE, of the terrain roughness while travelling the shortest distance and experiencing the least amount of vibration. Entropy search using the expected GP was able to obtain the smallest RMSE and WRMSE values, but required a much longer distance to be travelled under the cost of high vibration. In general, in terms of relative experienced vibration, BO and UBO methods with DUCB performed worse due to the high amounts of localisation noise, which led them to behave similarly to BO-ES. These results indicate that UIBO is able to effectively take into account the uncertainty in localisation, keeping the robot safe while exploring an unknown environment.

Method	RMSE	WRMSE	Distance Travelled	Relative Vibration
BO-DUCB	0.59 ± 0.09	0.31 ± 0.05	31.25 ± 1.98	0.95 ± 0.04
UBO-DUCB	0.64 ± 0.12	0.34 ± 0.08	35.49 ± 1.23	0.95 ± 0.04
UIBO-DUCB	0.59 ± 0.15	0.26 ± 0.08	$\mathbf{16.12 \pm 2.04}$	$\mathbf{0.61 \pm 0.17}$
BO-ES	0.58 ± 0.10	0.32 ± 0.07	57.07 ± 2.73	1.00
UBO-ES	0.64 ± 0.12	0.34 ± 0.08	67.15 ± 2.56	1.00 ± 0.03
UIBO-ES	$\mathbf{0.38 \pm 0.05}$	$\mathbf{0.22 \pm 0.05}$	62.85 ± 2.06	0.99 ± 0.03

TABLE 5.1: Simulation results: average performance comparisons for each BO approach under different metrics with the corresponding standard deviation (30 trials).

5.4.2 Experiment with a real robot

An experiment with a physical robot outdoors tested the performance of the proposed uncertain-inputs BO approach against standard BO with DUCB, as in Souza et al. (2014). The purpose of this experiment



(A) Robot platform

(B) Vibration measurements

FIGURE 5.4: Robot experiment details. (a) presents the experiments' robotic platform. (b) compares different methods to measure vibration using an IMU: *raw* corresponds to the raw vertical acceleration readings (discounting gravity), *mean* corresponds to the mean value of the raw measurements, considering a moving window of size 100, and *RMS* corresponds to the root-mean-square value of the raw measurements in the same window.

is to highlight the differences in performance and in behaviour between the two BO approaches. This experiment presents localisation uncertainty in a real-world scenario, where the robot is tasked with learning terrain roughness.

Experimental setup

The test platform was a small four-wheeled skid-steer robot, depicted in Figure 5.4a. The robot is equipped with an on-board computer running ROS [3]. Tests were performed in an area with terrain covered by grass and with some portions of harder ground exposed. The goal of the robot is to find areas of low terrain roughness, while avoiding areas that induce excessive vibration to the platform. In this kind of scenario, exposure to high amounts of vibration can cause damage to the robot's hardware, while interfering with readings from a few sensors, such as cameras and LIDAR's.

The robot was equipped with an inertial measurement unit (IMU) sensor, placed on top of its chassis. With the robot moving, vibration shows up mostly as linear acceleration along the robot's vertical axis, bouncing the robot up and down off the ground. Different methods, such as the mean and the root mean square (RMS) values, were considered to measure vibration from a fixed moving window of

[3] The Robot Operating System: `www.ros.org`

raw acceleration measurements. As the robot should avoid areas of high roughness, the RMS value demonstrated itself to be a better estimator, since it grows with the amplitude of the vibration, as pictured in Figure 5.4b. The mean, on the other hand, should be always around zero, as each upwards acceleration is immediately compensated by a subsequent fall.

As observations for the BO algorithms, the RMS vibration estimates were computed over a sliding window of 100 IMU measurements and updated at a rate of 2 Hz as the robot is driving. To reduce the effect of different driving speeds on the readings, the vibration estimates were posted whenever the robot was driving at speeds between 0.4 and 0.7 m/s, where 0.7 m/s was the maximum speed allowed for the path following control. These observations were combined with location estimates from an extended Kalman filter (EKF) (Moore and Stouch, 2014), which was configured to fuse wheel odometry, IMU and GPS estimates.

Both plain BO and UIBO were run for a fixed budget of 30 iterations, i.e. each algorithm was allowed to choose 30 target locations to take the robot to. At each iteration, the robot drove autonomously attempting to follow straight paths from its previous goal to the next goal, as given by the BO planner. The iteration was signalled as finished whenever the robot arrived within a given radius from the goal. After that, the GP model is updated with the new observations, the hyper-parameters are re-learnt by maximising the GP log-marginal likelihood using COBYLA (Powell, 2007), and the planner selects a new target. As the sample rate of observations is relatively low, both BO methods were able to repeatedly execute this process online in close to real time.

As in the simulations, both GP models were configured with the squared-exponential covariance function. However, hyper-parameters were learnt online by maximising marginal likelihood (Section 2.2.2). The DUCB acquisition function was configured with an uncertainty factor $\beta = 10$ and a distance factor $\zeta = 0.5$. UIBO applied as query distribution model an isotropic Gaussian with variance $\sigma_{\mathbf{x}}^2 = 4$ for each coordinate. The choice of this value for $\sigma_{\mathbf{x}}$ was mainly based on a minimum distance-to-goal tolerance for the robot's path follower, which was set to 1.3 metres to cope with execution noise. Ideally, $\sigma_{\mathbf{x}}$ could be estimated from data, which is left for future work.

Performance

Figure 5.5 presents the GP model obtained by each BO method and their paths, according to noisy EKF location estimates. For both methods, the robot was placed at an initial location at the bottom left area of

Method	WRMSE [m/s^2]	RMSE [m/s^2]	Distance Travelled [m]	Mean Vibration [m/s^2]
BO	1.04	1.31	247	3.0
UIBO	0.83	1.33	92	2.1

TABLE 5.2: Field results: performance comparisons for each BO approach under different metrics. The WRMSE was computed between the posterior mean of the final GP model at the locations in the validation dataset and the corresponding vibration measurements at those locations.

the search space. As Figure 5.5 shows, initially the robot guided by UIBO goes through areas that cause high vibration, possibly due to high initial GP uncertainty. However, UIBO is able to quickly adapt to the terrain and concentrate its search over areas of low vibration. On the other hand, standard BO ends up performing too much exploration passing more often over areas that cause excessive vibration.

For performance comparisons, a set of vibration measurements located using a real time kinematic (RTK) GPS [4] device was collected to obtain high-precision location estimates. These measurements were only collected for the purposes of validation, and not passed on to the BO planners, which relied solely on a conventional GPS device fused into the EKF estimates. The magnitude of the error in the conventional GPS location estimates varied around 3 metres with respect to the RTK GPS estimates, which is able to provide centimetre accuracy. The posterior mean of a GP model built directly from the validation data is shown in Figure 5.6. This model is presented only for visualisation purposes and was not used to compute performance metrics, which were solely based on raw validation data. The model's hyper-parameters were determined as $\{m_0, \ell, \sigma_f, \sigma_\nu\} = \{2.7, 2.8, 0.8, 1.0\}$ by maximising the GP log-marginal likelihood, where ℓ is the kernel length-scale and m_0 is the GP mean function parameter.

Table 5.2 presents a performance summary. The WRMSE value for this experiment was computed between the posterior mean from each method's final GP model, built from noisy EKF location estimates, and the raw vibration measurements present in the validation dataset, collected using precise RTK GPS location estimates. One can see that, for this experiment, UIBO is able to outperform standard BO by producing a better model over areas of low roughness while experiencing less vibration and following a much shorter path, confirming facts previously observed in simulations.

[4] Emlid Reach RTK: `https://emlid.com/reach/`

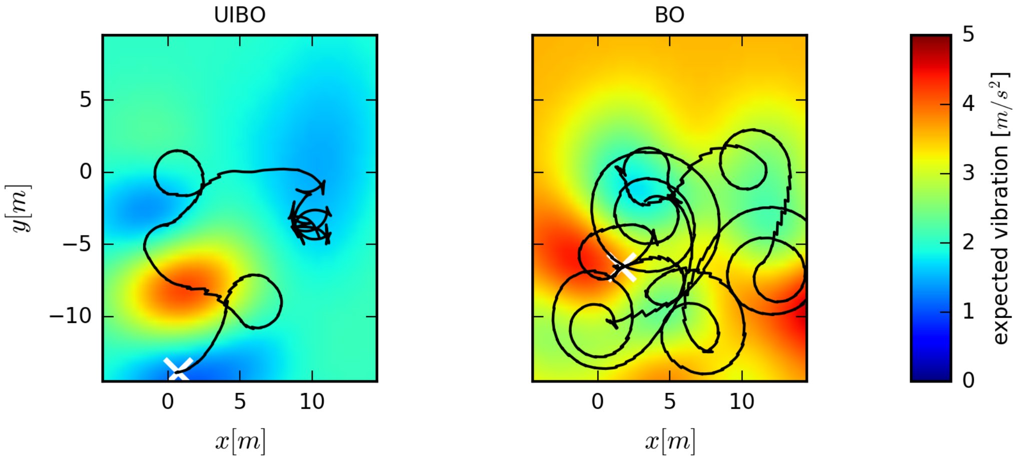


FIGURE 5.5: Posterior mean of the GP model built by each BO method overlaid with their respective paths according to locations estimated by an EKF fusing conventional GPS, IMU and odometry. In both cases, the robot started at the lower left corner of the map at the locations marked by a white cross.

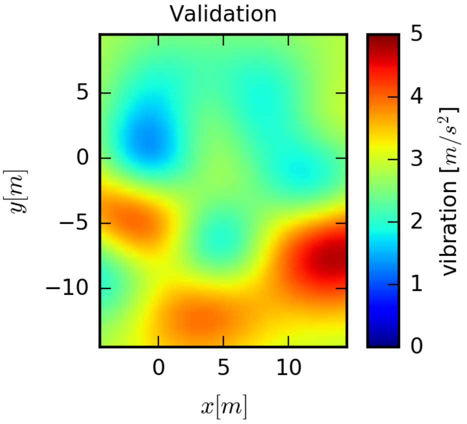


FIGURE 5.6: For visualisation purposes, posterior mean of a GP model built from the validation data taken with RTK GPS high-precision location estimates.

5.5 Summary

This chapter presented a new Bayesian optimisation method for active learning problems in robotics in the cases where uncertainty in the location estimates is significant. The proposed method provides a principled approach to account for this uncertainty affecting the inputs of a Gaussian process model, which is applied as a prior by the BO algorithm. Execution noise is also considered by means of a query distribution when optimising BO's acquisition function. The method was proved to outperform other BO approaches in simulation and in an experiment with a physical robot navigating over rough terrain.

Therefore, the proposed method can be applied to problems where one is interested in mapping the traversability of a terrain, while keeping the robot safe, in the midst of localisation and model uncertainty.

Some topics were not addressed by this chapter, but are worthy of future research work. One of them is the estimation of execution noise, which could be done in an online way by other statistical methods, for example, using maximum likelihood estimates for the parameters of a given distribution. To perform such estimates, however, would require ground-truth data, which might be unavailable, leading to the need for adaptive methods that can work with noisy localisation estimates, instead. Another topic is the extension of the DUCB-based exploration to informative continuous path planning (Marchant and Ramos, 2014) or to trajectory optimisation in goal-directed navigation, as in Chapter 3. Such extension could allow BO to handle obstacles by integrating it with a path planner over a GP-based cost function and to safely plan paths over longer distances by integrating predictions and the uncertainty of the model along the path. For that, however, a suitable method to propagate execution noise through a candidate path would have to be developed. This requires taking into account hard-to-model factors from both the robot and the environment, such as stochastic motion dynamics and imperfect sensor information, to derive a model of the paths distribution. An approach with similar issues is to derive transition models which can be used for trajectory roll outs, as in stochastic motion planing. Then one could use the roll outs as an approximate empirical distribution over the true paths.

CHAPTER 6

Bayesian optimisation under uncertain inputs

6.1 Introduction

The previous chapter investigated a planning problem where localisation uncertainty affects the guidance of a robot and the location of measurements taken by the robot. Although applied to the task of learning a terrain roughness model, the BO framework proposed in that chapter could also be applied to more general optimisation problems. A question that comes alongside, however, is whether Bayesian optimisation algorithms can still find the global optimum of a function in situations involving input noise. In addition, most theoretical guarantees given to BO algorithms are derived for a GP model whose domain is an Euclidean vector space, which is no longer the case when inputs are given as probability distributions, as proposed in Chapter 5.

Classical BO algorithms account only for the noise affecting samples of the objective function for a given query location (Brochu et al., 2010; Snoek et al., 2012; Chowdhury and Gopalan, 2017). Most algorithms then assume that the function has been sampled precisely at the specified query location within the given search space. However, there are certain problems, especially in the areas of robotics and control, where this assumption typically does not hold. Chapter 5 presented one such problem in the context of traversability estimation for robotics.

This chapter investigates optimisation problems where noise affects both the execution of a query and the estimation of the actual query location. The investigation in this chapter provides a generalisation and a more formal analysis of the framework proposed in Chapter 5. In particular, the main contributions [1] are theoretical guarantees, for a standard BO approach (Chowdhury and Gopalan, 2017) and a

[1] This work has been partially presented at the Workshop on Informative Path Planning and Adaptive Sampling (WIPPAS) at the 2018 IEEE International Conference on Robotics and Automation (ICRA) as Oliveira et al. (2018a). A more up-to-date version of this work and its results are also available in the proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS) (Oliveira et al., 2019).

novel BO algorithm, in problems involving noisy inputs. The proposed algorithm, named *uncertain-inputs Gaussian process upper confidence bound* (uGP-UCB), employs a GP model for uncertain inputs that allows extending the theoretical results in Chowdhury and Gopalan (2017) to the uncertain-inputs setting. Experiments in simulation present empirical performance comparisons against the standard BO approach and the unscented BO method (Nogueira et al., 2016).

6.2 Related work

A few BO approaches have been recently proposed to deal with problems where the execution of queries to an objective function is affected by uncertainty. As mentioned in Chapter 5, Nogueira et al. (2016) presented a method that applies the unscented transform (Wan and van der Merwe, 2000) to query BO's acquisition function. By considering a stochastic query execution process, the method is able to find robust solutions to robotics problems such as grasping. Another approach to handle query uncertainty is presented in Pearce and Branke (2017) to optimise stochastic simulations. In that case, query uncertainty refers to imperfect knowledge about input variates for a simulation model (Lam, 2016). Pearce and Branke use a GP model that takes in the full input variates vector and then apply Monte Carlo integration to marginalise out the unknown input variates. In a similar context, Beland and Nair (2017) consider a robust optimisation problem under stochastic constraints. However, their method takes the expected value of the GP covariance function when computing estimates of an objective defined as an expected loss function. Contrasted to the method in this chapter, all the approaches mentioned above only deal with independent and identically distributed input noise and offer no known theoretical guarantees. The data points in their GP datasets are also only point estimates, instead of full distributions as proposed here.

This chapter adopts a BO framework similar to Chapter 5, with an important modification to the formulation of the uncertain-inputs GP model, which leads to the first theoretical guarantees of BO algorithms for uncertain-inputs optimisation problems. The work in this chapter sheds new light on the GP models in Girard (2004) and Dallaire et al. (2011), connecting them to the theory of kernel embeddings of probability distributions (Muandet et al., 2016). The latter allows for upper bounds on the approximation error of BO in the uncertain-inputs setting when running an extension of the GP-UCB algorithm (Srinivas et al., 2012; Chowdhury and Gopalan, 2017).

Problems like the one illustrated in Figure 5.1 can be found whenever the objective is a function of an agent's unknown state, which is only partially controllable through a continuous set of actions. Therefore, it is possible to relate the work in this chapter to partially-observable Markov decision processes (POMDPs), which have been previously connected to BO (Marchant and Ramos, 2014; Ling et al., 2016). POMDP problems are usually concerned with selecting actions that lead to optimal outcomes over future evaluations. The work in this chapter, however, is concerned with a general optimisation setup. In this setting, an agent is only concerned with maximising its immediate expected reward.

6.3 Problem formulation

Consider an optimisation problem where an algorithm sequentially selects target locations $\mathbf{x}_t$ within a compact search space $\mathcal{S} \subset \mathcal{X}$ at which to query a function $f : \mathcal{X} \to \mathbb{R}$ seeking a global optimum. In addition, the query execution itself is a stochastic process, leading the query to be made at some $\tilde{\mathbf{x}}_t \neq \mathbf{x}_t$, instead. Hence, the goal is defined as finding:

$$\mathbf{x}^* \in \operatorname*{argmax}_{\mathbf{x}\in\mathcal{S}} \mathbb{E}[f(\tilde{\mathbf{x}})|\mathbf{x}] , \tag{6.1}$$

which defines a target location that maximises the expected value of the function f under the querying process noise. This execution noise could be the outcome of a sequence of actions taken by the agent to reach the target location $\mathbf{x}_t$, which would then result in dependence on the starting point $\tilde{\mathbf{x}}_{t-1}$ and the actions themselves. However, assume that $\tilde{\mathbf{x}}_t|\mathbf{x}_t \sim P^E_{\mathbf{x}_t}$, i.e.:

$$\mathbb{E}[f(\tilde{\mathbf{x}}_t)|\mathbf{x}_t] = \mathbb{E}_{\tilde{\mathbf{x}}^E_t \sim P^E_{\mathbf{x}_t}}[f(\tilde{\mathbf{x}}^E_t)] =: \mathbb{E}_{P^E_{\mathbf{x}_t}}[f] , \tag{6.2}$$

where $P^E_{\mathbf{x}_t}$ accounts for all possible starting points and actions taken to reach the given target location $\mathbf{x}_t$, i.e. an average case scenario. Also assume that, *after* each query, the algorithm is provided with a distribution P^L_t representing an estimate of the actual query location $\tilde{\mathbf{x}}_t$ and observations $y_t = f(\tilde{\mathbf{x}}_t)+\zeta_t$, where ζ_t is σ_ζ-sub-Gaussian noise, for some $\sigma_\zeta \geq 0$. Notice that the sub-Gaussian assumption on the noise is more general than usual BO assumptions. Both Gaussian and bounded random variables meet the sub-Gaussian assumption, for example (Boucheron et al., 2013).

Regret: In the design of a global optimisation algorithm, the main goal is to find the location of the global optimum $\mathbf{x}_*$. A measure of how close the algorithm got to the global optimum at round t is the instant regret $r_t := \max_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) - f(\mathbf{x}_t)$, as presented in Section 2.3.1. In the uncertain-inputs case, however, the noise affecting query execution might render $\max_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$ a very unlikely outcome and possibly not robust in certain situations (Nogueira et al., 2016). Therefore, a better performance measure is the *expected regret* under uncertain inputs, defined as:

$$\hat{r}_t := \max_{\mathbf{x}\in\mathcal{S}} \mathbb{E}_{P^E_{\mathbf{x}}}[f] - \mathbb{E}_{P^E_{\mathbf{x}_t}}[f] \,. \tag{6.3}$$

As mentioned in Section 2.3.1, a desirable property of an optimisation algorithm is to have no regret, meaning that $\lim_{n\to\infty} \min_{t\leq n} \hat{r}_t = 0$. The *no regret* property is analogous to the concept of convergence in local gradient-based optimisation, but it does not indicate the same, as the global optimisation algorithm may keep exploring the search space, never actually converging. Instead, a bound on the *cumulative expected regret* of the algorithm, $\hat{R}_n := \sum_{t=1}^n \hat{r}_t$, indicates whether the algorithm can eventually get arbitrarily close to the global optimum as iterations go by, since $\min_{t\leq n} \hat{r}_t \leq \hat{R}_n/n$. Therefore, a goal for the algorithm's design can be defined as minimising the cumulative expected regret.

Regularity assumptions: The problem in Equation 6.1 is still very general and may not be solvable if considered for an arbitrary function f. Therefore, as a regularity assumption, $f : \mathcal{X} \to \mathbb{R}$ is assumed to be an element of $\mathcal{H}_k$, which is a reproducing kernel Hilbert space (RKHS, see Section 2.4), and $\|f\|_k \leq b$, where $b > 0$ is known.[2] The kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is assumed to be continuous and bounded,[3] with $k(\mathbf{x}, \mathbf{x}) \leq 1, \forall \mathbf{x} \in \mathcal{X}$. For the derivations, assume f's domain $\mathcal{X}$ to be an Euclidean vector space $\mathcal{X} \subseteq \mathbb{R}^d$, $d \in \mathbb{N}$, though some of the theoretical results remain valid for other types of topological vector spaces.

6.4 The uncertain-inputs Gaussian process upper confidence bound

This section describes a method for Bayesian optimisation under uncertain inputs. The section starts by presenting a Gaussian process that allows modelling objectives defined in terms of expectations directly. This GP approach is then applied to derive a BO algorithm named *uncertain-inputs Gaussian process upper confidence bound* (uGP-UCB), presented in the second part of the section.

[2] In general, for $k(\mathbf{x}, \mathbf{x}) \leq 1, \forall \mathbf{x} \in \mathcal{X}$, one can show that $\sup_{\mathbf{x}\in\mathcal{X}} |f(\mathbf{x})| \leq \|f\|_k$, so that the norm bound corresponds to a supremum bound. The RKHS norm is also related to the Lipschitz constant by Steinwart and Christmann (2008, Cor. 4.36).

[3] This is the case for most of the popular kernels in the machine learning literature, such as the squared exponential and the ones in the Matérn class (see Section 2.2.1). The maximum value of 1 is adopted only for convenience in deriving the theoretical results. If the kernel maximum is given instead by some $c > 0$, the results are still valid by scaling the terms dependent on b by c. This bounded variation assumption on k is common in the regret bounds literature for BO methods (Srinivas et al., 2012; Chowdhury and Gopalan, 2017).

6.4.1 A Gaussian process model for the expected function

To apply BO to the problem defined in Equation 6.1, one needs a model for the given objective. Let $\mathcal{P} := \mathcal{P}(\mathcal{X}, \mathfrak{X})$ denote the set containing all probability measures on $(\mathcal{X}, \mathfrak{X})$, where $\mathfrak{X}$ is the Borel σ-algebra of $\mathcal{X}$ (see Section 2.6). We can then define an objective $\hat{f}$ as a function over $\mathcal{P}$ as:

$$\begin{aligned} \hat{f} : \mathcal{P} &\to \mathbb{R} \\ P &\mapsto \mathbb{E}_{\tilde{\mathbf{x}}\sim P}[f(\tilde{\mathbf{x}})] \,. \end{aligned} \tag{6.4}$$

Considering the given regularity assumptions on the kernel k, we can define the mean map:

$$\begin{aligned} \psi : \mathcal{P} &\to \mathcal{H}_k \\ P &\mapsto \int_{\mathcal{X}} k(\cdot, \mathbf{x}) \, \mathrm{d}P(\mathbf{x}) \,. \end{aligned} \tag{6.5}$$

For a $\mathcal{X}$-valued random variable $\tilde{\mathbf{x}}$ distributed according to $P \in \mathcal{P}$, we then have that:

$$\mathbb{E}_{\tilde{\mathbf{x}}\sim P}[f(\tilde{\mathbf{x}})] = \langle \psi_P, f \rangle_k, \ \forall f \in \mathcal{H}_k \,, \tag{6.6}$$

where $\psi_P := \psi(P)$. If the kernel k is characteristic, such as radial kernels on $\mathbb{R}^d$ (Sriperumbudur et al., 2011), ψ is injective, defining a one-to-one relationship between measures in $\mathcal{P}$ and elements of $\mathcal{H}_k$. Therefore, ψ_P is referred to as the kernel mean embedding of P (Muandet et al., 2016).

Using ψ as defined in Equation 6.5, one can construct kernels over the set of probability measures $\mathcal{P}$. In particular, for any $P, P' \in \mathcal{P}$, we have that:

$$\hat{k}(P, P') := \langle \psi_P, \psi_{P'} \rangle_k = \int_{\mathcal{X}} \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}') \, \mathrm{d}P(\mathbf{x}) \, \mathrm{d}P'(\mathbf{x}') \tag{6.7}$$

defines a positive-definite kernel over $\mathcal{P}$ (Muandet et al., 2012). This kernel is associated with a RKHS $\mathcal{H}_{\hat{k}}$ containing functions over the space of probability measures $\mathcal{P}$. Besides the linear kernel in Equation 6.7, many other kernels on $\mathcal{P}$ can be defined, e.g. radial kernels using $\|\psi_P - \psi_{P'}\|_k$ as a distance metric. However, the simple kernel in Equation 6.7 provides a useful property to model the objective function $\hat{f}$ in Equation 6.4, as presented next.

LEMMA 6.1 (Expected function). *Any $f \in \mathcal{H}_k$ is continuously mapped to a corresponding $\hat{f} \in \mathcal{H}_{\hat{k}}$, which is such that:*

$$\begin{aligned} \forall P \in \mathcal{P}, \quad \hat{f}(P) &= \mathbb{E}_P[f] \\ \|\hat{f}\|_{\hat{k}} &= \|f\|_k \,. \end{aligned} \tag{6.8}$$

Under the additional assumption that k is characteristic, we have that the mapping $f \mapsto \hat{f}$ constitutes an isometry between $\mathcal{H}_k$ and $\mathcal{H}_{\hat{k}}$ and is also invertible.

PROOF SKETCH. The proof for the first statement follows from the fact that Dirac measures are members of $\mathcal{P}$, since they are defined on any σ-algebra. The second statement then follows from the injectivity of the mean map ψ when k is characteristic. The full proof is presented in the appendix (Section A.2). □

As a positive-definite kernel, $\hat{k}$ defines the covariance function of a Gaussian process $\mathrm{GP}(0, \hat{k})$ modelling functions over $\mathcal{P}$. This GP model can then be applied to learn $\hat{f}$ from a given set of past observations $\mathcal{D}_{t-1} = \{(P_i, y_i)\}_{i=1}^{t-1}$. Under a zero-mean GP assumption, the value of $\hat{f}(P_*)$ for a given $P_* \in \mathcal{P}$ follows a Gaussian posterior distribution with mean and variance given by:

$$\hat{\mu}_{t-1}(P_*) = \hat{\mathbf{k}}_{t-1}(P_*)^\mathsf{T}(\hat{\mathbf{K}}_{t-1} + \lambda\mathbf{I})^{-1}\mathbf{y}_{t-1} \;, \tag{6.9}$$

$$\hat{\sigma}^2_{t-1}(P_*) = \hat{k}(P_*, P_*) - \hat{\mathbf{k}}_{t-1}(P_*)^\mathsf{T}(\hat{\mathbf{K}}_{t-1} + \lambda\mathbf{I})^{-1}\hat{\mathbf{k}}_{t-1}(P_*) \;, \tag{6.10}$$

where $\hat{\mathbf{k}}_{t-1}(P_*) := [\hat{k}(P_*, P_1), \dots, \hat{k}(P_*, P_{t-1})]^\mathsf{T}$ and $[\hat{\mathbf{K}}_{t-1}]_{ij} = \hat{k}(P_i, P_j)$. Recall that for a $\hat{f} \in \mathcal{H}_{\hat{k}}$, $\hat{f}$ is generally not a sample from the GP (see Section 2.4.3). However, we always have that $\hat{\mu}_{t-1} \in \mathcal{H}_{\hat{k}}$, allowing the GP to learn an approximation for $\hat{f}$. Therefore, in these equations, $\lambda \geq 0$ is simply a parameter that can be set independently from the true observation noise variance (cf. Section 2.2).

Contrast with the expected GP model: Although very similar, the GP model proposed above provides useful properties for learning $\hat{f}$ from data when compared to the expected GP (Section 2.2.4). In the latter, Dallaire et al. (2011) formulate $\hat{k}$, instead, as the expected covariance function under the joint distribution of both random inputs. As a result, any diagonal entry in the expected GP covariance matrix has the form:

$$\mathbb{E}_{\tilde{\mathbf{x}} \sim P_i}[k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})] = \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}) \,\mathrm{d}P_i(\mathbf{x}) \neq \int_{\mathcal{X}}\int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}') \,\mathrm{d}P_i(\mathbf{x}) \,\mathrm{d}P_i(\mathbf{x}') = \langle \psi_{P_i}, \psi_{P_i} \rangle_k \;, \tag{6.11}$$

since both inputs correspond to the same random variable. Another difference is that, for any given $f \in \mathcal{H}_k$, we have that f's variance under an input distribution P is given by:

$$\begin{aligned} \mathbb{V}_P[f] = \mathbb{E}_{\tilde{\mathbf{x}} \sim P}[(f(\tilde{\mathbf{x}}) - \mathbb{E}_P[f])^2] &= \mathbb{E}_{\tilde{\mathbf{x}} \sim P}[(\langle f, k(\cdot, \tilde{\mathbf{x}}) - \psi_P \rangle_k)^2] \\ &\leq \|f\|_k^2 \mathbb{E}_{\tilde{\mathbf{x}} \sim P}[\|k(\cdot, \tilde{\mathbf{x}}) - \psi_P\|_k^2] \\ &= \|f\|_k^2 \mathbb{E}_{\tilde{\mathbf{x}} \sim P}[k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) - 2\psi_P(\tilde{\mathbf{x}}) + \|\psi_P\|_k^2] \\ &= \|f\|_k^2 (\mathbb{E}_{\tilde{\mathbf{x}} \sim P}[k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})] - \|\psi_P\|_k^2) , \end{aligned} \tag{6.12}$$

where the second line follows by the Cauchy-Schwartz inequality, and the last equality follows from the linearity of the expectation and $\mathbb{E}_{\tilde{\mathbf{x}} \sim P}[\psi_P(\tilde{\mathbf{x}})] = \langle \psi_P, \psi_P \rangle_k = \|\psi_P\|_k^2$. Then, considering the inequality above, if we had that $\hat{k}(P, P) = \langle \psi_P, \psi_P \rangle_k = \|\psi_P\|_k^2$ was equivalent to $\mathbb{E}_{\tilde{\mathbf{x}} \sim P}[k(\tilde{\mathbf{x}}, \tilde{\mathbf{x}})]$, as is the case in the expected GP formulation, $\mathbb{V}_P[f]$ would be zero, which is not true in general. Therefore, the expected GP covariance function is not directly related to $\hat{f}$ and differs from the kernel in Equation 6.7. Consider, for example, a set of Gaussian measures $P_i = N(\hat{\mathbf{x}}_i, \mathbf{\Sigma}_i)$, $i = 1, \dots, n$. In the case of a squared-exponential kernel with signal variance σ_f^2 and length-scales matrix $\mathbf{L}$, Equation 6.7 resolves to (Girard, 2004, Eq. 3.53):

$$\hat{k}(N(\hat{\mathbf{x}}_i, \mathbf{\Sigma}_i), N(\hat{\mathbf{x}}_j, \mathbf{\Sigma}_j)) = \frac{\sigma_f^2 \exp\left(-\frac{1}{2}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)^\mathsf{T}(\mathbf{L} + \mathbf{\Sigma}_i + \mathbf{\Sigma}_j)^{-1}(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)\right)}{|\mathbf{I} + \mathbf{L}^{-1}(\mathbf{\Sigma}_i + \mathbf{\Sigma}_j)|^{1/2}} , \tag{6.13}$$

which differs from the expected covariance function formulation by the lack of a term in the denominator that accounts for inputs corresponding to the same random variable (cf. Equation 2.26).

When to use the expected GP model: For a query at $\tilde{\mathbf{x}}|\mathbf{x} \sim P$, the *expected* GP in Section 2.2.4 actually proposes a Gaussian approximation to the distribution over the *function value* $p(f(\tilde{\mathbf{x}})|\mathbf{x}, \mathcal{D}_n)$, when both f and $\tilde{\mathbf{x}}$ are random variables. When $f \sim \mathrm{GP}(0, k)$, the uncertain-inputs GP model (this chapter), instead, provides a model over the *expected value* distribution $\mathbb{E}_{\tilde{\mathbf{x}}}[f(\tilde{\mathbf{x}})|\mathbf{x}, \mathcal{D}_n] \sim N(\hat{\mu}_n(P), \hat{\sigma}_n^2(P))$, which is exact due to the linearity of the expectation and the invariance of Gaussians under linear transformations. Therefore, one should use Dallaire et al. (2011)'s model when concerned with inference over the value of $f(\tilde{\mathbf{x}})$, while the uncertain-inputs GP provides a better alternative when the concern is over the value of $\mathbb{E}_{\tilde{\mathbf{x}}}[f(\tilde{\mathbf{x}})|\mathbf{x}]$, as in this chapter. Moreover, due to the lack of the adjustment for the case when both inputs correspond to the same random variable, which causes the collapse of the covariance function into a single integral (see Equation 6.11), the uncertain-inputs GP predictive variance is generally lower than the expected GP's. Using the uncertain-inputs GP to infer over $f(\tilde{\mathbf{x}})$, instead of $\mathbb{E}_{\tilde{\mathbf{x}}}[f(\tilde{\mathbf{x}})|\mathbf{x}]$, might then mislead BO's exploration of the search space due to the underestimated uncertainty.

Remark: As a final remark, using a GP model with covariance function as defined in Equation 6.7 is not a new idea. Similar GP formulations have been previously proposed in the literature to account for Gaussian input noise (Rasmussen and Ghahramani, 2003; Girard, 2004). Under the RKHS mean embeddings approach, the kernel in Equation 6.7 has also been applied to learn GP models over bag-of-words data (Yoshikawa et al., 2015). The novelty here, instead, lies in the reinterpretation of this GP as a model for an objective function defined in terms of expected values of another function, as in the formulation of the uncertain-inputs optimisation problem (Section 6.3).

6.4.2 The uGP-UCB algorithm

The GP model proposed in the previous section allows deriving a BO algorithm to solve the problem in Equation 6.1. Given a set of past observations $\mathcal{D}_{t-1} = \{(P_i, y_i)\}_{i=1}^{t-1}$, the following defines an upper confidence bound (UCB) acquisition function:

$$h(P|\mathcal{D}_{t-1}) = \hat{\mu}_{t-1}(P) + \beta_t \hat{\sigma}_{t-1}(P) \ , \tag{6.14}$$

where β_t is a parameter controlling the exploration-exploitation trade-off. The theoretical results in the next section will show that β_t can be set accordingly to maintain a high-probability upper bound on $\hat{f}$. Since the upper confidence bound in Equation 6.14 is defined over a GP model for uncertain inputs, this acquisition function and the corresponding BO algorithm are named *uncertain-inputs Gaussian process upper confidence bound* (uGP-UCB).

Querying the GP model with $\mathbf{x} \mapsto P^E_{\mathbf{x}}$ would allow selecting points based on an estimate for $\mathbb{E}_{P^E_{\mathbf{x}}}[f]$. However, in general, the true mapping $\mathbf{x} \mapsto P^E_{\mathbf{x}}$ is unknown, as discussed in Section 5.3.2. Instead, one can use a model $\mathbf{x} \mapsto \hat{P}_{\mathbf{x}}$ whose approximation error $|\mathbb{E}_{P^E_{\mathbf{x}}}[f] - \mathbb{E}_{\hat{P}_{\mathbf{x}}}[f]|$ is small.

Algorithm 7 presents the uGP-UCB algorithm. Equipped with the acquisition function in Equation 6.14, at each iteration t, the algorithm selects the target location $\mathbf{x}_t$ that maximises $h(\hat{P}_{\mathbf{x}}|\mathcal{D}_{t-1})$ (line 2). In line 3, the function f is queried at some location $\tilde{\mathbf{x}}_t|\mathbf{x}_t \sim P^E_{\mathbf{x}_t}$. After the query is done, the algorithm is provided with an observation $y_t = f(\tilde{\mathbf{x}}_t) + \zeta_t$ and an independent estimate for $\tilde{\mathbf{x}}_t$ given by P^L_t (line 4), as described in Section 6.3. In line 5, the GP model is updated with the new observation pair (P^L_t, y_t). This process then repeats for a given number of iterations n. As a result, the algorithm provides an estimate of the optimum location $\mathbf{x}^*$ given as the target location with the best estimated outcome $\mathbf{x}^*_n$ (line 6).

Algorithm 7: uGP-UCB

```
Input:
    $\mathcal{S}$: search space
    $n$: total number of iterations
for $t \in \{1, \dots, n\}$ do
    $\mathbf{x}_t = \operatorname*{argmax}_{\mathbf{x} \in \mathcal{S}} \hat{\mu}_{t-1}(\hat{P}_{\mathbf{x}}) + \beta_t \hat{\sigma}_{t-1}(\hat{P}_{\mathbf{x}})$
    $y_t \leftarrow$ Observe $f$ at $\tilde{\mathbf{x}}_t | \mathbf{x}_t \sim P^E_{\mathbf{x}_t}$
    $P^L_t \leftarrow$ Estimate $\tilde{\mathbf{x}}_t$
    $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(P^L_t, y_t)\}$
$\mathbf{x}^*_n = \operatorname*{argmax}_{t \in \{1,\dots,n\}} \hat{\mu}_n(\hat{P}_{\mathbf{x}_t})$
Result: $\mathbf{x}^*_n$
```

When compared to Chapter 5, the method proposed here differs in both purpose and properties. The DUCB method in Section 5.3.3 aims at exploring while minimising travelled distance. The uGP-UCB method, instead, deals with a more general optimisation scenario, where distances are not considered, and may not even make sense, depending on the application. In terms of properties, DUCB methods do not provide any known theoretical guarantees. The expected GP model used in Chapter 5 also makes it difficult to analyse the theoretical properties of the method in that chapter. As discussed in Section 6.4.1, the expected GP is not a suitable model for the expected value of f, relying only on an approximate covariance function. The uncertain-inputs GP in Section 6.4.1, however, allows for a proper derivation of the resulting approximation error between the model and the true $\hat{f}$. This fact leads to the theoretical results presented in the next section.

6.5 Theoretical analysis

This section presents theoretical results bounding the expected regret (Section 6.3) of the uGP-UCB algorithm and a standard BO approach proposed by Chowdhury and Gopalan (2017), which was not originally designed to handle input noise. Before the main theoretical results are presented, the section introduces the results obtained by Chowdhury and Gopalan for their method and the definitions related to their results. The theoretical analysis presented in this chapter is mainly based on their results, but it also brings new insights into BO methods for optimisation under uncertain inputs. Section 6.5.2 presents bounds obtained for the expected regret of IGP-UCB in the noisy-inputs setting, and Section 6.5.3 follows with the regret bounds for uGP-UCB.

6.5.1 Regret in the deterministic-inputs case

The main theoretical foundations for the regret of GP-UCB algorithms have been presented in Srinivas et al. (2010) and several works followed after that. Recently, Chowdhury and Gopalan (2017) proposed a version of the algorithm, named *improved Gaussian process upper confidence bound* (IGP-UCB), which attains no regret under very general observation noise assumptions. More specifically, they assume only additive conditionally sub-Gaussian noise affecting the observations of f, an assumption which is advantageous for the analysis of the uncertain-inputs BO case. The definition of *conditionally sub-Gaussian observation noise* is based on an abstraction of the BO algorithm in terms of the variables involved in the optimisation process. This general abstraction is presented next, followed by a restatement of the main result in Chowdhury and Gopalan (2017).

For an asymptotic analysis of BO, both the queries $\{\mathbf{x}_t\}_{t=1}^{\infty}$ and the observation noise $\{\nu_t\}_{t=1}^{\infty}$ can be treated as sequences of random variables. At a given round $t \geq 1$, the current history $\{\mathbf{x}_i, \nu_i\}_{i=1}^{t}$ generates a σ-algebra $\mathfrak{F}_t := \mathfrak{A}(\mathbf{x}_i, \nu_i; i \in \{1, \dots, t\})$. The history also determines the next query location $\mathbf{x}_{t+1}$, which is deterministically selected based on the UCB given by the GP posterior. Considering an abstract probability space $(\mathcal{E}, \mathfrak{E}, P_{\mathcal{E}})$ where all the random variables in $\{\mathbf{x}_t\}_{t=1}^{\infty}$ and $\{\nu_t\}_{t=1}^{\infty}$ are defined, the sequence $\{\mathfrak{F}_t\}_{t=0}^{\infty}$, where $\mathfrak{F}_0 := \{\varnothing, \mathcal{E}\}$, defines a filtration in $\mathfrak{E}$ (see Section 2.6). Both $\{\mathbf{x}_t\}_{t=1}^{\infty}$ and $\{\nu_t\}_{t=1}^{\infty}$ are adapted to the filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$, and $\{\mathbf{x}_t\}_{t=1}^{\infty}$ is also predictable with respect to it. Under a setting such as this, the sub-Gaussian condition on the sequence $\{\nu_t\}_{t=1}^{\infty}$ is formally defined as:

DEFINITION 6.2. *A sequence of real-valued random variables $\{\nu_t\}_{t=1}^{\infty}$ adapted to a filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$ is said to be conditionally σ_ν-sub-Gaussian, $\sigma_\nu > 0$, if:*

$$\forall t \geq 1,\ \forall \lambda \in \mathbb{R}, \quad \mathbb{E}[e^{\lambda \nu_t} | \mathfrak{F}_{t-1}] \leq e^{\lambda^2 \sigma_\nu^2 / 2} \qquad \text{(a.s.)}\ . \tag{6.15}$$

The notation $\mathbb{E}[e^{\lambda \nu_t} | \mathfrak{F}_{t-1}]$ denotes a conditional expectation given the σ-algebra $\mathfrak{F}_{t-1}$ (see Section 2.6.3). The conditional expectation given a σ-algebra is also a random value, so that the inequality above is defined as holding *almost surely* (a.s.), i.e. with probability one. As an example, any zero-mean random variable that is Gaussian with variance at most σ_ν^2 or bounded with values lying in $[-\sigma_\nu, \sigma_\nu]$ is σ_ν-sub-Gaussian (Boucheron et al., 2013).

Besides the definitions above, the theoretical results in Chowdhury and Gopalan (2017) are also dependent on the maximum information gain γ_n, which is defined as:

$$\gamma_n := \max_{\mathcal{Q} \subset \mathcal{X} : |\mathcal{Q}| = n} I(\mathbf{y}_{\mathcal{Q}}, \mathbf{f}_{\mathcal{Q}}) \ , \tag{6.16}$$

The mutual information $I(\mathbf{y}_{\mathcal{Q}}, \mathbf{f}_{\mathcal{Q}})$ above is given by:

$$I(\mathbf{y}_{\mathcal{Q}}, \mathbf{f}_{\mathcal{Q}}) = \frac{1}{2} \log |\mathbf{I} + \lambda^{-1} \mathbf{K}_{\mathcal{Q}}| \ , \tag{6.17}$$

under the assumption that $\mathbf{y}_{\mathcal{Q}} = \mathbf{f}_{\mathcal{Q}} + \boldsymbol{\nu}_{\mathcal{Q}}$, with $\boldsymbol{\nu}_{\mathcal{Q}} \sim N(\mathbf{0}, \lambda \mathbf{I})$, and $\mathbf{f}_{\mathcal{Q}} := [f(\mathbf{x})]_{\mathbf{x} \in \mathcal{Q}} \in \mathbb{R}^n$. Similar to the regularity assumptions in this chapter (Section 6.3), IGP-UCB also considers $f \in \mathcal{H}_k$, and consequently not a sample from $\mathrm{GP}(0, k)$ (see discussion in Section 2.4.3). For IGP-UCB, the GP is only a model to approximate f. Considering the definitions above, the following theorem upper bounds the cumulative regret of IGP-UCB in the deterministic-inputs setting.

THEOREM 6.3 (Chowdhury and Gopalan (2017, Theorem 3)). *Let $\delta \in (0,1)$, $\|f\|_k \leq b$, and ν_t be conditionally σ_ν-sub-Gaussian noise. Then, running IGP-UCB with $\beta_t = b + \sigma_\nu \sqrt{2(\gamma_{t-1} + 1 + \log(1/\delta))}$ for $f \in \mathcal{H}_k(\mathcal{S})$, and a compact $\mathcal{S} \subset \mathbb{R}^d$, the cumulative regret of the algorithm is bounded by $\mathcal{O}(\sqrt{n}(b\sqrt{\gamma_n} + \gamma_n))$ with high probability. Specifically, we have that:*

$$\mathbb{P}\left\{ R_n \in \mathcal{O}\left(b\sqrt{n\gamma_n} + \sqrt{n(\gamma_n + \log(1/\delta))} \right) \right\} \geq 1 - \delta \ . \tag{6.18}$$

The result above states that the cumulative regret of IGP-UCB is bounded by a quantity dependent on the maximum information gain γ_n. Bounds for γ_n are available in the literature (see Srinivas et al., 2012, Theorem 5) depending on the choice of covariance function. For common choices, such as the squared exponential and the Matérn class of functions (Section 2.2.1), γ_n grows sub-linearly with n. In particular, for the squared-exponential kernel, γ_n is $\mathcal{O}((\log n)^{d+1})$ (Srinivas et al., 2010, Thr. 5). In this case, the cumulative regret of IGP-UCB grows sub-linearly, so that the algorithm attains asymptotically no regret. The result in Theorem 6.3 is mainly based on the following noise concentration bound.

THEOREM 6.4 (Theorem 1 in Chowdhury and Gopalan (2017)). *Let $\{\mathbf{x}_t\}_{t=1}^{\infty}$ be a $\mathbb{R}^d$-valued discrete-time stochastic process predictable with respect to the filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$, i.e. $\mathbf{x}_t$ is $\mathfrak{F}_{t-1}$-measurable for all $t \geq 1$. Let $\{\nu_t\}_{t=1}^{\infty}$ be a real-valued stochastic process such that, for some $\sigma_\nu \geq 0$ and for all $t \geq 1$, ν_t is (a) $\mathfrak{F}_t$-measurable, and (b) σ_ν-sub-Gaussian conditionally on $\mathfrak{F}_{t-1}$. Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a symmetric, positive-definite kernel, and let $0 < \delta \leq 1$. For a given $\eta > 0$, with probability at least $1 - \delta$,* *the following holds simultaneously for all* $t \geq 0$*:*

$$\|\boldsymbol{\nu}_t\|^2_{((\mathbf{K}_t+\eta\mathbf{I})^{-1}+\mathbf{I})^{-1}} \leq 2\sigma_\nu^2 \log \frac{\sqrt{|(1+\eta)\mathbf{I}+\mathbf{K}_t|}}{\delta} \ , \tag{6.19}$$

where $\mathbf{K}_t$ *is the* t*-by-*t *kernel matrix with elements* $[\mathbf{K}_t]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$*, and* $\|\boldsymbol{\alpha}\|^2_{\mathbf{M}} := \boldsymbol{\alpha}^\mathsf{T}\mathbf{M}\boldsymbol{\alpha}$*, for a positive-definite matrix* $\mathbf{M} \in \mathbb{R}^{t\times t}$ *and a vector* $\boldsymbol{\alpha} \in \mathbb{R}^t$*. We also have that, if* $\mathbf{K}_t$ *is almost surely positive definite for all* $t \geq 1$*, the result above also holds for* $\eta = 0$*.*

The result in Theorem 6.4 bounds the growth of the observation noise vector with respect to the GP kernel matrix. This norm bound allows showing that with high-probability:

$$\forall \mathbf{x} \in \mathcal{S}, \quad |f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq \beta_t \sigma_{t-1}(\mathbf{x}) \ , \tag{6.20}$$

when β_t has the form in Theorem 6.3 (see Chowdhury and Gopalan, 2017, Thr. 2). This derivation follows from the decomposition:

$$\begin{aligned} \forall \mathbf{x} \in \mathcal{S}, \quad |f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| &= |f(\mathbf{x}) - \mathbf{k}_{t-1}(\mathbf{x})^\mathsf{T}\mathbf{K}_{t-1}\mathbf{y}_{t-1}| \\ &= |f(\mathbf{x}) - \mathbf{k}_{t-1}(\mathbf{x})^\mathsf{T}\mathbf{K}_{t-1}(\mathbf{f}_{t-1} + \boldsymbol{\nu}_{t-1})| \\ &\leq |f(\mathbf{x}) - \mathbf{k}_{t-1}(\mathbf{x})^\mathsf{T}\mathbf{K}_{t-1}\mathbf{f}_{t-1}| + |\mathbf{k}_{t-1}(\mathbf{x})^\mathsf{T}\mathbf{K}_{t-1}\boldsymbol{\nu}_{t-1}| \ . \end{aligned} \tag{6.21}$$

Looking at the last line, one can shown that the first term on the right-hand side is bound by $b\sigma_{t-1}(\mathbf{x})$, while a bound for the second term can be derived from the result in Theorem 6.4 and its relation to the information gain (Equation 6.17). For more details on this derivation, the reader is referred to Chowdhury and Gopalan (2017, App. C). Since IGP-UCB selects query locations $\mathbf{x}_t$ that maximise $\mu_{t-1}(\mathbf{x}) + \beta_t\sigma_{t-1}(\mathbf{x})$, the instant regret can be bound with high-probability as:

$$\begin{aligned} \forall t \geq 1, \quad r_t &:= \max_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) - f(\mathbf{x}_t) \\ &\leq \mu_{t-1}(\mathbf{x}_t) + \beta_t\sigma_{t-1}(\mathbf{x}_t) - f(\mathbf{x}_t) \\ &\leq 2\beta_t\sigma_{t-1}(\mathbf{x}_t) \ . \end{aligned} \tag{6.22}$$

Then we can bound the cumulative regret in terms of β_t and the GP predictive variances, which determine the information gain (recall Equation 2.42), leading to the result in Theorem 6.3.

To bound the expected regret of uGP-UCB, Section 6.5.3 will derive a result analogous to the one in Theorem 6.4. However, since the query location estimate P_t^L is independently provided, and not necessarily predictable at $t-1$, the analogous of Theorem 6.4 should not depend on the predictability assumption, besides having to be applicable under non-Euclidean domains, such as $\mathcal{P}$. Firstly, the next section will present theoretical results for the expected regret of IGP-UCB under query execution noise.

6.5.2 Bounding the expected regret of IGP-UCB

This section derives theoretical guarantees for Chowdhury and Gopalan's IGP-UCB when applied to the uncertain-inputs optimisation problem (Equation 6.1). This kind of result allows analysing how is the performance of IGP-UCB affected by input noise and how one should adapt the algorithm to that. The derivation starts by analysing the assumptions necessary for the result in Theorem 6.3 to hold in the uncertain-inputs setting. The main result in this section then basically follows as a re-statement of Theorem 6.3 in terms of the cumulative expected regret. The remainder of the section then presents cases where the necessary assumptions for the main result hold in practice.

The main assumptions of Theorem 6.3 are that the objective function is a member of the RKHS $\mathcal{H}_k(\mathcal{S})$ (as defined in Section 2.4.2) and that the observation noise ν_t is conditionally sub-Gaussian. To analyse IGP-UCB in the uncertain-inputs setting, firstly, one can define an objective function g as:

$$\begin{aligned} g : \mathcal{S} &\to \mathbb{R} \\ \mathbf{x} &\mapsto \mathbb{E}_{P^E_{\mathbf{x}}}[f] \,. \end{aligned} \tag{6.23}$$

However, there is no guarantee that such g is a member of $\mathcal{H}_k(\mathcal{S})$ for a general mapping $\mathbf{x} \mapsto P^E_{\mathbf{x}}$ and kernel k. Secondly, even if $g \in \mathcal{H}_k(\mathcal{S})$ in some special case, the resulting observation noise $\nu_t = y_t - g(\mathbf{x}_t)$ is not necessarily conditionally σ_ν-sub-Gaussian for a given $\sigma_\nu > 0$. What is observable is only the value of $y_t = f(\tilde{\mathbf{x}}_t) + \zeta_t$, where f is the original deterministic-inputs objective, $\tilde{\mathbf{x}}_t|\mathbf{x}_t \sim P^E_{\mathbf{x}_t}$, and $\zeta_t \neq \nu_t$ is conditionally sub-Gaussian. The following theorem summarises the conditions under which IGP-UCB in the noisy-inputs setting attains bounded expected regret.

THEOREM 6.5 (Noisy-inputs IGP-UCB regret). *Let $f \in \mathcal{H}_k$, where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive-definite kernel on $\mathcal{X} \subseteq \mathbb{R}^d$. Assume that:*

(1) the mapping $\mathbf{x} \mapsto \mathbb{E}_{P^E_{\mathbf{x}}}[f]$ defines a function $g \in \mathcal{H}_k(\mathcal{S})$ and $\|g\|_k \leq b$;

(2) for a given $\sigma_F > 0$, the random variable defined by $\Delta f_{P^E_{\mathbf{x}}} := f(\tilde{\mathbf{x}}) - \mathbb{E}_{P^E_{\mathbf{x}}}[f]$, where $\tilde{\mathbf{x}}|\mathbf{x} \sim P^E_{\mathbf{x}}$, is σ_F-sub-Gaussian for all $\mathbf{x} \in \mathcal{S}$; and

(3) ζ_t is conditionally σ_ζ-sub-Gaussian for all $t \geq 1$.

Then setting the observation noise parameter as $\sigma_\nu := \sqrt{\sigma_F^2 + \sigma_\zeta^2}$ *and running IGP-UCB with* $\beta_t := b + \sigma_\nu\sqrt{2(\gamma_{t-1} + 1 + \log(1/\delta))}$ *leads to the cumulative expected regret* $\hat{R}_n$ *being bound by:*

$$\mathbb{P}\left\{\hat{R}_n \in \mathcal{O}\left(b\sqrt{n\gamma_n} + \sqrt{n(\gamma_n + \log(1/\delta))}\right)\right\} \geq 1 - \delta\ . \tag{6.24}$$

PROOF. Theorem 6.5 establishes sufficient conditions for Theorem 6.3 to be applicable to the noisy-inputs settings. The observation noise, as perceived by the GP model, is $\nu_t := y_t - g(\mathbf{x}_t)$, where g follows the definition in Equation 6.23 and $\mathbf{x}_t$ is the location selected by IGP-UCB according to the setting for β_t in Theorem 6.5. Observations y_t are taken at $\tilde{\mathbf{x}}_t^E \sim P_{\mathbf{x}_t}^E$, instead, yielding:

$$\nu_t = y_t - g(\mathbf{x}_t) = \zeta_t + f(\tilde{\mathbf{x}}_t^E) - \mathbb{E}_{P_{\mathbf{x}_t}^E}[f] = \zeta_t + \Delta f_{P_{\mathbf{x}_t}^E}\ . \tag{6.25}$$

Given that $\mathbf{x}_t$ is $\mathfrak{F}_{t-1}$-measurable, as it is predictable given $\{\mathbf{x}_i, \nu_i\}_{i=1}^{t-1}$, we have that $\Delta f_{P_{\mathbf{x}_t}^E}$ is σ_F-sub-Gaussian when conditioned on $\mathfrak{F}_{t-1}$. By assumption 3, ζ_t is conditionally sub-Gaussian. Since ζ_t and $\Delta f_{P_{\mathbf{x}_t}^E}$ are independent given $\mathfrak{F}_{t-1}$, we have that:

$$\begin{aligned}\forall\lambda \in \mathbb{R},\ \mathbb{E}[\exp(\lambda\nu_t)|\mathfrak{F}_{t-1}] &= \mathbb{E}\left[\exp\left(\lambda\left(\zeta_t + \Delta f_{P_{\mathbf{x}_t}^E}\right)\right)\middle|\mathfrak{F}_{t-1}\right] \\ &= \mathbb{E}\left[\exp(\lambda\zeta_t)\exp\left(\lambda\Delta f_{P_{\mathbf{x}_t}^E}\right)\middle|\mathfrak{F}_{t-1}\right] \\ &\leq e^{\lambda^2\sigma_\zeta^2/2}e^{\lambda^2\sigma_F^2/2} = e^{\lambda^2(\sigma_\zeta^2+\sigma_F^2)/2} = e^{\lambda^2\sigma_\nu^2/2} \qquad \text{(a.s.)}\ ,\end{aligned} \tag{6.26}$$

so that ν_t is conditionally σ_ν-sub-Gaussian by Definition 6.2.

Assumption 1 states that $g \in \mathcal{H}_k(\mathcal{S})$, meeting the remaining requirement for Theorem 6.3. Therefore, running IGP-UCB with σ_ν and $b \geq \|g\|_k$, following the settings in Theorem 6.3, leads to cumulative regret bounds for g as in Equation 6.18. From the definition in Equation 6.3, r_t for g is equivalent to the expected regret $\hat{r}_t$ for f, which leads to the conclusion in Theorem 6.5. □

The result above states that, as long as the observation noise sub-Gaussian parameter σ_ν is large enough to accommodate for the additional variance propagated to the output observations due to input noise, IGP-UCB should attain similar for the cumulative expected regret to those of Theorem 6.3. However, it is possible that the extra output noise variance σ_F^2 required to compensate for all the execution noise distributions at each target $\mathbf{x} \in \mathcal{S}$ is impractically large. For example, the RKHS norm bound $\|f\|_k \leq b$ and $k(\mathbf{x}, \mathbf{x}) \leq 1, \forall\mathbf{x} \in \mathcal{X}$, indicate that $|f(\mathbf{x})| \leq \|f\|_k = b, \forall\mathbf{x} \in \mathcal{X}$. Then, for any $\tilde{\mathbf{x}} \sim P$, $\Delta f_P := f(\tilde{\mathbf{x}}) - \mathbb{E}_P[f]$ is b-sub-Gaussian, but b, which bounds f itself, is too big as a noise parameter. The following result presents special cases where the output noise level may be more suitable in practice.

PROPOSITION 6.6. *Let $f \in \mathcal{H}_k$ and $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be an at least twice-differentiable positive-definite kernel with finite $\ell_k^2 \geq \sup_{\mathbf{x}\in\mathbb{R}^d} \sup_{i\in\{1,\dots,d\}} \frac{\partial^2 k(\mathbf{x},\mathbf{x}')}{\partial x_i \partial x_i'}|_{\mathbf{x}=\mathbf{x}'}$. Then, for a given $P \in \mathcal{P}$ and $\tilde{\mathbf{x}} \sim P$, we have that $\Delta f_P := f(\tilde{\mathbf{x}}) - \mathbb{E}_P[f(\tilde{\mathbf{x}})]$ is a σ_F-sub-Gaussian random variable in the following cases:*

(1) $\sigma_F = \|f\|_k \ell_k \operatorname{tr}(\mathbf{\Sigma})^{1/2}$, if P is Gaussian with covariance matrix $\mathbf{\Sigma}$;

(2) $\sigma_F = \frac{1}{2}\|f\|_k \ell_k \sqrt{\sum_{i=1}^d \sigma_i^2}$, if P has compact support, with $|\tilde{x}_i - \hat{x}_i| \leq \frac{1}{2}\sigma_i$, $i \in \{1,\dots,d\}$, where $\tilde{\mathbf{x}} \sim P$ and $\hat{\mathbf{x}} = \mathbb{E}[\tilde{\mathbf{x}}]$.

PROOF SKETCH. The proof follows by applying concentration inequalities for Lipschitz-continuous functions of random variables (Boucheron et al., 2013). Any $f \in \mathcal{H}_k$, as defined, is ℓ_f-Lispchitz continuous with $\ell_f = \ell_k \|f\|_k$ (Steinwart and Christmann, 2008, see Corollary 4.36). The full proof is presented in the appendix (Section A.3). □

The assumptions in the result above are met for $P_{\mathbf{x}}^E$ if the execution-noise is uniformly bounded or Gaussian. Therefore, Theorem 6.5 says that it is possible to guarantee asymptotic convergence for BO methods that can tolerate sub-Gaussian observation noise, such as IGP-UCB (Chowdhury and Gopalan, 2017), in the noisy-inputs setting.

Theorem 6.5 assumes that the function g, as defined in Equation 6.23, is a member of $\mathcal{H}_k(\mathcal{S})$. This assumption is equivalent to the common requirement that the mapping $\mathbf{x} \mapsto \mathbb{E}[f(\tilde{\mathbf{x}})|\mathbf{x}]$ defines an element of $\mathcal{H}_k$ for any given $f \in \mathcal{H}_k$ when working with conditional mean embeddings (Muandet et al., 2016). In fact, $P_{\mathbf{x}}^E$ is the conditional distribution of the executed query $\tilde{\mathbf{x}}$ given the target location $\mathbf{x}$. Most of the theoretical results in the kernel embeddings literature are based on discrete domains, where the former assumption typically holds. However, optimisation problems involving input noise are more common if considering functions with continuous domains. Hence, to verify whether g as assumed in Theorem 6.5 exists for any $f \in \mathcal{H}_k$, where the domain $\mathcal{X}$ is continuous, new results are required.

In the continuous domain case, translation-invariant kernels, such as the squared exponential, are popular in the kernel machines literature due to their modelling properties. For instance, Sriperumbudur et al. (2011) demonstrated that any radial kernel on $\mathbb{R}^d$ is a universal approximator for continuous functions. Therefore, one may wonder if the assumption that the mapping $\mathbf{x} \mapsto \mathbb{E}_{\tilde{\mathbf{x}}|\mathbf{x}}[f(\tilde{\mathbf{x}})|\mathbf{x}]$ is an element of $\mathcal{H}_k$, for any $f \in \mathcal{H}_k$, is valid for some special case involving translation-invariant kernels. The next result presents a case where such assumption holds.

PROPOSITION 6.7. *Let k be a bounded, measurable, translation-invariant kernel on $(\mathcal{X}, \mathfrak{X})$, where $\mathcal{X}$ is a topological vector space, and $\mathfrak{X}$ is its Borel σ-algebra. Assume that the mapping $\mathbf{x} \mapsto P_{\mathbf{x}} \in \mathcal{P}(\mathcal{X})$ is such that, for any $\mathbf{x} \in \mathcal{S} \subset \mathcal{X}$, $\tilde{\mathbf{x}} \sim P_{\mathbf{x}}$ is decomposable as $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is independent and identically distributed, i.e. $\boldsymbol{\epsilon} \sim P_E \in \mathcal{P}(\mathcal{X})$. Then we have that, for any $f \in \mathcal{H}_k$, the mapping $\mathbf{x} \mapsto \mathbb{E}_{P_{\mathbf{x}}}[f]$ defines a function $g \in \mathcal{H}_k(\mathcal{S})$, and $\|g\|_k \leq \|f\|_k$.*

PROOF SKETCH. The proof follows by interpreting $\boldsymbol{\epsilon}$ as a random translation on f. Since the kernel is translation invariant, it happens that the norm of any $\boldsymbol{\epsilon}$-shifted function $f^{\boldsymbol{\epsilon}}$ is equivalent to the norm of the original function. Then picking g as the restriction of $\mathbb{E}_{P_E}[f^{\boldsymbol{\epsilon}}] \in \mathcal{H}_k$ to $\mathcal{S} \subset \mathcal{X}$ leads to the conclusion. Section A.4 presents the full proof. □

Proposition 6.7 implies that Theorem 6.5 is applicable whenever the execution noise $\boldsymbol{\epsilon}^E$ is independent and identically distributed regardless of the target location $\mathbf{x}$. One can always make such assumption by considering expectations with respect to some marginal distribution of the execution noise. However, in cases where the distribution of $\boldsymbol{\epsilon}^E$ changes significantly from target to target, algorithms such as uGP-UCB, which allow querying the GP model with different distributions per target, can lead to better BO performance.

6.5.3 The expected regret of uGP-UCB

As presented in Theorem 6.5, guarantees for IGP-UCB in the noisy-inputs setting are limited by a few assumptions. Proposition 6.7 requires i.i.d. execution noise for the first assumption in Theorem 6.5 to hold. The cases where observation noise is guaranteed to be sub-Gaussian at a suitable level for practical applications are also limited by Proposition 6.6. Considering these restrictions, the proposed uGP-UCB method offers advantages for optimisation problems with uncertain inputs by allowing for an arbitrary execution noise distribution. In addition, uGP-UCB can also take advantage of the post-query location estimates $\tilde{\mathbf{x}}_t^L \sim P_t^L$, which can induce lesser output noise, i.e. $\Delta f_{P_t^L} := f(\tilde{\mathbf{x}}_t^L) - \mathbb{E}_{P_t^L}[f]$ has low variance. However, deriving regret bounds for uGP-UCB offers additional challenges when compared to the IGP-UCB case, as presented in this section.

Main result

The theoretical analysis of uGP-UCB will consider a general setting where $\hat{P}_{\mathbf{x}}^t \approx P_{\mathbf{x}}^E$ denotes a model that is allowed to change as the algorithm proceeds, such as an adaptive model that is learnt online.

Recall that the regret bounds presented so far depend on the maximum information gain γ_n. As an analogy, in the case of uGP-UCB, given an arbitrary set $\{P_t\}_{t=1}^n \subset \mathcal{P}$, we have:

$$I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{P_t\}_{t=1}^n) = \frac{1}{2} \log |\mathbf{I} + \lambda^{-1} \hat{\mathbf{K}}_n| \,, \tag{6.27}$$

where $[\hat{\mathbf{K}}_n]_{ij} = \hat{k}(P_i, P_j),\ i, j \in \{1, \dots, n\}$. The results presented next will assume the existence of upper bounds $\hat{\gamma}_n^L$ and $\hat{\gamma}_n^Q$ for any possible $I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{P_t^L\}_{t=1}^n)$ and $I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{\hat{P}_{\mathbf{x}_t}^t\}_{t=1}^n)$, respectively. This assumption is reasonable as the kernel is bounded. Deriving the exact formulation of these bounds for arbitrary kernels is out of the scope of this thesis. However, Proposition 6.10 will present one case where these quantities are upper-bounded by their deterministic-inputs counterparts, which have known upper bounds (Srinivas et al., 2010, Thr. 5). Considering these definitions, the following presents the main result regarding the expected regret of uGP-UCB. The proof will be presented in the second part of this section.

THEOREM 6.8. *Let $\delta \in (0, 1)$, $f \in \mathcal{H}_k$, and $b \geq \|f\|_k$. Consider ζ_t as σ_ζ-sub-Gaussian noise. Assume that both k and location estimates P_t^L satisfy the conditions for $\Delta f_{P_t^L}$ to be σ_F-sub-Gaussian, for a given $\sigma_F > 0$, for all $t \geq 1$. Then, running uGP-UCB with:*

$$\beta_t = b + \sqrt{2(\sigma_F^2 + \sigma_\zeta^2)(I(\mathbf{y}_{t-1}; \hat{\mathbf{f}}_{t-1} \mid \{P_i^L\}_{i=1}^t) + 1 + \log(1/\delta))} \tag{6.28}$$

and a model $\mathbf{x} \mapsto \hat{P}_\mathbf{x}^t$, such that:

$$\forall t \geq 1,\ \exists \rho_t \geq 0 : \max_{\mathbf{x} \in \mathcal{S}} \left| \mathbb{E}_{P_\mathbf{x}^E}[f] - \mathbb{E}_{\hat{P}_\mathbf{x}^t}[f] \right| \leq \rho_t \qquad \text{(a.s.)}\,, \tag{6.29}$$

the cumulative expected regret satisfies:

$$\mathbb{P}\left\{ \hat{R}_n \in \mathcal{O}\left(\sum_{t=1}^n \rho_t + \sqrt{n \hat{\gamma}_n^Q} \left(b + \sqrt{\hat{\gamma}_n^L + \log(1/\delta)} \right) \right) \right\} \geq 1 - \delta \,. \tag{6.30}$$

Theorem 6.8 states that the cumulative expected regret of uGP-UCB follows a similar pattern to that of IGP-UCB, but depends on the error between the query execution model and the true query distribution. In addition, the two information gain bounds, the one with respect to the location estimates in P_t^L and the other with respect to the query distributions $\hat{P}_{\mathbf{x}_t}^t$, affect differently the regret of the algorithm, with the latter having a larger impact on it. According to Theorem 6.8, assuming a sub-linear growth for $\hat{\gamma}_n^Q$ and $\hat{\gamma}_n^L$, the expected regret of uGP-UCB can asymptotically converge to zero if $\rho_t \to 0$ as $t \to \infty$, since:

$$\lim_{n \to \infty} \min_{t \leq n} \hat{r}_t \leq \limsup_{n \to \infty} \frac{\hat{R}_n}{n} \leq \limsup_{n \to \infty} \frac{1}{n} \sum_{t=1}^n \rho_t \leq \limsup_{t \to \infty} \rho_t \tag{6.31}$$

with probability at least $1-\delta$. Otherwise, even if $\rho_t \not\to 0$, but to a constant $c > 0$, i.e. $\rho_t \to c$, uGP-UCB still reaches the global optimum within a value difference given by c. Therefore, using a GP model for uncertain inputs is advantageous if one has a model $\hat{P}^t_{\mathbf{x}} \approx P^E_{\mathbf{x}}$ such that the modelling error ρ_t is small. The error term ρ_t is implementation dependent and can be bounded in terms of divergence measures via Pinsker's inequality (Boucheron et al., 2013) or by metrics such as the maximum mean discrepancy $\|\psi_{P^E_{\mathbf{x}}} - \psi_{\hat{P}^t_{\mathbf{x}}}\|_k$ (Muandet et al., 2016).

In practical applications, considering execution noise as marginally i.i.d. Gaussian has been a popular approach when dealing with problems involving uncertain inputs (Mchutchon and Rasmussen, 2011; Nogueira et al., 2016). In this case, the following result provides a constant upper bound for the approximation error ρ_t.

PROPOSITION 6.9. *Let $\mathcal{X} = \mathbb{R}^d$, $f \in \mathcal{H}_k$ and $\|f\|_k \leq b$. Assume that, for any $\mathbf{x} \in \mathcal{S}$, the query distribution $P^E_{\mathbf{x}}$ is Gaussian with mean $\mathbf{x}$ and positive-definite covariance $\mathbf{\Sigma}^E$. Then, using a Gaussian model $\hat{P}_{\mathbf{x}}$ with same mean and a given constant positive-definite covariance matrix $\hat{\mathbf{\Sigma}}$, we have that:*

$$\forall \mathbf{x} \in \mathcal{S}, \quad \left|\mathbb{E}_{P^E_{\mathbf{x}}}[f] - \mathbb{E}_{\hat{P}_{\mathbf{x}}}[f]\right| \leq \frac{b}{2}\left(\operatorname{tr}(\hat{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}^E) - d + \log\frac{|\hat{\mathbf{\Sigma}}|}{|\mathbf{\Sigma}^E|}\right)^{1/2} .$$

PROOF SKETCH. The result follows from Pinsker's inequality (Boucheron et al., 2013), which relates the total variation distance between two probability distributions to their Kullback-Leibler divergence. The reader is referred to Section A.6 for a complete proof. □

The remaining important terms that appear in Theorem 6.8 are $\hat{\gamma}^Q_n$ and $\hat{\gamma}^L_n$, the upper bounds on the uncertain-inputs GP information gain with respect to the query model and the estimated location distributions, respectively. Considering any set $\mathcal{P}_\mathcal{S} \subset \mathcal{P}$, define:

$$\hat{\gamma}_n(\mathcal{P}_\mathcal{S}) := \sup_{\mathcal{R} \subset \mathcal{P}_\mathcal{S} : |\mathcal{R}| = n} I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \mathcal{R}) , \tag{6.32}$$

where $\mathbf{y}_n = \hat{\mathbf{f}}_n + \boldsymbol{\nu}'_n$, $\hat{\mathbf{f}}_n \sim N(\mathbf{0}, \hat{\mathbf{K}}_n)$ and $\boldsymbol{\nu}'_n \sim N(\mathbf{0}, \lambda\mathbf{I})$, with $[\hat{\mathbf{K}}_n]_{ij} = \hat{k}(P_i, P_j)$, $P_i, P_j \in \mathcal{R}$. As the set $\mathcal{P}_\mathcal{S}$ is not necessarily compact, a maximum for $I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \mathcal{R})$ may not exist. However, in the case of location distributions affected by i.i.d. noise, this maximum exists and is equivalent to the deterministic-inputs maximum information gain.

PROPOSITION 6.10. *Consider a compact set $\mathcal{S} \subset \mathcal{X}$, a distribution $P_{\boldsymbol{\epsilon}} \in \mathcal{P}$, with $\mathbb{E}_{P_{\boldsymbol{\epsilon}}}[\boldsymbol{\epsilon}] = 0$, and a set:*

$$\mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S}) := \{P \in \mathcal{P} \mid \tilde{\mathbf{x}} \sim P \Longrightarrow \tilde{\mathbf{x}} = \hat{\mathbf{x}} + \boldsymbol{\epsilon},\, \boldsymbol{\epsilon} \sim P_{\boldsymbol{\epsilon}},\, \hat{\mathbf{x}} \in \mathcal{S}\} , \tag{6.33}$$

which is the set of location distributions with mean on $\mathcal{S}$ and affected by i.i.d. P_ϵ-noise. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a continuous translation-invariant kernel, and $\hat{k} : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ be according to Equation 6.7. Then we have that:

$$\forall n \geq 1, \quad \hat{\gamma}_n(\mathcal{P}_\epsilon(\mathcal{S})) \leq \gamma_n(\mathcal{S}) \; . \tag{6.34}$$

PROOF SKETCH. This result follows from an analysis similar to the applied in the proof of Proposition 6.7. One can prove that $\mathbf{K}_n - \hat{\mathbf{K}}_n$ is positive definite for $\mathbf{K}_n$ built with $\{\hat{\mathbf{x}}_t\}_{t=1}^n \subset \mathcal{S}$. As the information gain grows monotonically with the determinant of the matrices, the deterministic inputs information gain will always be higher for that choice of inputs, leading to the inequality in the statement. A complete proof is derived in Section A.7. □

The above proposition suggests that one can apply upper bounds available for the deterministic-inputs information gain γ_n (Srinivas et al., 2012, Theorem 5) to bound $\hat{\gamma}_n^Q$ and $\hat{\gamma}_n^L$. Moreover, since the regret bounds for IGP-UCB and uGP-UCB are a function of their respective maximum information gain, which is also true for other theoretically-proven GP-UCB strategies (Srinivas et al., 2012), Proposition 6.10 also implies that uGP-UCB should be able to obtain lower expected regret in optimisation problems with input noise than other GP-UCB methods, as Section 6.6 will show. Finally, notice that although β_t has a similar formulation for both IGP-UCB and uGP-UCB, σ_F is derived from P_t^L for the latter, while depending on $P_{\mathbf{x}}^E$ for the former. Hence, lower-noise location estimates from P_t^L can benefit uGP-UCB.

The result in Proposition 6.10 also reveals that, under an increase in the input noise level, the optimisation problem as defined in Equation 6.1 becomes simpler. The magnitude of the expected value of the objective function decreases as the variance of the input distributions increases by a natural filtering effect that expectations have. Assuming $\sup_{\mathbf{x}\in\mathcal{X}} |f(\mathbf{x})| = \max_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x})$, we have that:

$$\max_{\mathbf{x}\in\mathcal{S}} \mathbb{E}_{P_{\mathbf{x}}^E}[f] = \max_{\mathbf{x}\in\mathcal{S}} \int_{\mathcal{X}} f(\mathbf{x}') \, \mathrm{d}P_{\mathbf{x}}^E(\mathbf{x}') \leq \sup_{\mathbf{x}\in\mathcal{X}} |f(\mathbf{x})| = \max_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) \; , \tag{6.35}$$

where equality holds for all $f \in \mathcal{H}_k$ if and only if $P_{\mathbf{x}}^E = D_{\mathbf{x}}$ for any $\mathbf{x} \in \mathcal{S}$, i.e. the noise-free case[4]. This fact confirms the practical intuition that optimising functions under the influence of query execution noise is only worthwhile if the noise levels are not excessively high.

[4] $D_{\mathbf{x}}$ denotes the Dirac measure on $\mathbf{x}$ (see Equation 2.74).

Proof of the main result

To derive similar results to Theorem 6.3 for uGP-UCB, the relevant stochastic processes and the associated filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$, which the processes are adapted to, need to be defined. Firstly, as a general abstraction, the sequence of true query locations at each round t forms a stochastic process $\{\tilde{\mathbf{x}}_t\}_{t=1}^{\infty}$. Both the execution noise and the noise affecting the observations in the GP dataset influence *where* the process $\{\tilde{\mathbf{x}}_t\}_{t=1}^{\infty}$ will be realised at round t. Secondly, as proposed, uGP-UCB uses the distributions P_t^L as entries in the GP dataset. These estimates are only available after $\tilde{\mathbf{x}}_t$ is realised and are subject to external factors. Therefore, it is reasonable to assume that P_t^L itself is a random variable or, more appropriately, a random measure.

The theory of random measures has a vast literature with different formal definitions of what a random measure is (Suquet, 2009; Kallenberg, 2017). For the theoretical analysis of uGP-UCB, the objective function is dependent on the probability measures only via expectations (Section 6.3). In addition, all the relevant operations involving measures can be defined in terms of inner products in the RKHS via the embeddings with ψ (Equation 6.5). For these purposes then, a useful definition of a random measure is given by Berlinet and Thomas-Agnan (2004, p. 233, Def. 36), and it is adapted here as:

DEFINITION 6.11. *Let $\mathcal{M}^+$ denote the set of all non-negative finite measures on $(\mathcal{X}, \mathfrak{X})$. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ denote a positive-definite characteristic kernel on $\mathcal{X}$, so that the map:*

$$\psi : \mathcal{M}^+ \to \mathcal{H}_k \tag{6.36}$$

$$M \mapsto \int_{\mathcal{X}} k(\cdot, \mathbf{x}) \, \mathrm{d}M(\mathbf{x}) \tag{6.37}$$

is injective. A random measure on $(\mathcal{X}, \mathfrak{X})$ based on a probability space $(\mathcal{E}, \mathfrak{E}, P)$ is a random variable defined on $(\mathcal{E}, \mathfrak{E}, P)$ and taking almost surely its values in $\psi(\mathcal{M}^+)$ equipped with the σ-algebra induced by $\mathfrak{B}_k := \mathfrak{B}(\mathcal{H}_k)$, which is the Borel σ-algebra of $\mathcal{H}_k$.

Under this definition, each random probability measure $P^{(\cdot)}$ is identified by the embedding $\psi_P^{(\cdot)}$, which is a $\mathcal{H}_k$-valued random variable, i.e.:

$$\begin{aligned} \psi_P^{(\cdot)} &: \mathcal{E} \to \mathcal{H}_k \\ \omega \mapsto \psi_P^{(\omega)} &= \int k(\cdot, \mathbf{x}) \, \mathrm{d}P^{(\omega)}(\mathbf{x}) \end{aligned} \tag{6.38}$$

One may question whether this map is measurable, since a random variable is by default a measurable map from a probability space into another measurable space (see Section 2.6.3). As presented by Berlinet and Thomas-Agnan (2004, p. 195, Cor. 12), a map $\xi^{(\cdot)} : \mathcal{E} \to \mathcal{H}_k$ is $\mathfrak{E}$-$\mathfrak{B}_k$-measurable if and only if, for any $\mathbf{x} \in \mathcal{X}$, the function $\omega \mapsto \xi^{(\omega)}(\mathbf{x}) = \langle \xi^{(\omega)}, k(\cdot, \mathbf{x}) \rangle_k$ is a real-valued random variable. In the case of uGP-UCB, this is implied by assuming that $\omega \mapsto \mathbb{E}_{P_t^{L(\omega)}}[f]$ defines a real-valued random variable for any $f \in \mathcal{H}_k$, which is a reasonable assumption to consider. Including $\psi^{(\cdot)}_{P_t^L}$ as one of the generators of the σ-algebra $\mathfrak{F}_t$ in $\mathcal{E}$ then makes these maps $\mathfrak{F}_t$-$\mathfrak{B}_k$-measurable.

Regarding the observation noise, only point estimates of $\mathbb{E}_{P_t^L}[f]$ are available with each pair (P_t^L, y_t) as $y_t = f(\tilde{\mathbf{x}}_t^L) + \zeta_t$, where $\tilde{\mathbf{x}}_t^L \sim P_t^L$. Here $\tilde{\mathbf{x}}_t^L$ denotes the random variable representing the distribution of the true query location $\tilde{\mathbf{x}}_t$ when P_t^L is known. To extend the results in Theorem 6.3, one needs the equivalent observation noise as perceived by the GP model, i.e. $\nu_t := y_t - \hat{f}(P_t^L)$, to be conditionally sub-Gaussian. This definition depends on the existence of a filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$. Considering the other random variables, $\mathfrak{F}_t$ can be defined as the σ-algebra generated by the random variables $\{\zeta_i, \tilde{\mathbf{x}}_i, P_i^L\}_{i=1}^{t}$ for all $t \geq 1$. For the particular case of $t = 0$, $\mathfrak{F}_0 := \{\mathcal{E}, \varnothing\}$ is a valid choice. For this choice, it is clear that $\{\mathfrak{F}_t\}_{t=0}^{\infty}$ defines a filtration, since $\mathfrak{F}_i \subset \mathfrak{F}_j$, for every $i \leq j$, $i, j \in \{0, 1, 2, \dots\}$. Furthermore, as ν_t is fully determined by ζ_t, $\tilde{\mathbf{x}}_t$ and P_t^L, this definition implies that ν_t is $\mathfrak{F}_t$-measurable, so that the process $\{\nu_t\}_{t=1}^{\infty}$ is adapted to $\{\mathfrak{F}_t\}_{t=0}^{\infty}$. At any round t, the observation noise ν_t is conditionally sub-Gaussian with respect to $\mathfrak{F}_{t-1}$, if it satisfies the condition in Definition 6.2 for some $\sigma_\nu > 0$. Therefore, the conditional sub-Gaussianity of ν_t depends on requirements on ζ_t and P_t^L.

Contrasting with IGP-UCB in terms of the assumptions in Theorem 6.4 on the stochastic processes, the t'th GP input P_t^L is not predictable at round $t - 1$ in the case of uGP-UCB. As a random variable, only $\mathbf{x}_t$ is $\mathfrak{F}_{t-1}$-measurable, since it is chosen as a deterministic function of the history $\{\zeta_i, \tilde{\mathbf{x}}_i, P_i^L\}_{i=1}^{t-1}$. In addition, the GP domain $\mathcal{P}$ is not Euclidean any more, since the set of Borel probability measures on $\mathcal{X}$ cannot be defined as a finite-dimensional vector space for an arbitrary $\mathcal{X} \subseteq \mathbb{R}^d$. Therefore, a new base result is needed to bound the expected regret of uGP-UCB. The following theorem generalises the result in Chowdhury and Gopalan (2017, Thr. 1) to arbitrary domains and removes the predictability assumption on the inputs in the GP dataset.

THEOREM 6.12. *Consider an arbitrary domain $\mathcal{W}$ and a σ-algebra $\mathfrak{W}$ on it. Let $\{\xi_t\}_{t=1}^{\infty}$ be a $\mathcal{W}$-valued discrete-time stochastic process adapted to the filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$. Let $\{\nu_t\}_{t=1}^{\infty}$ be a real-valued stochastic process such that, for some $\sigma_\nu \geq 0$ and for all $t \geq 1$, ν_t is (a) $\mathfrak{F}_t$-measurable, and (b) σ_ν-sub-Gaussian conditionally on $\mathfrak{F}_{t-1}$. Let $\hat{k} : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$ be a $\mathfrak{W}$-measurable, bounded positive-definite kernel, and let $0 < \delta \leq 1$. For a given $\eta > 0$, with probability at least $1 - \delta$, the following holds* *simultaneously for all $t \geq 0$:*

$$\|\boldsymbol{\nu}_t\|^2_{((\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1}+\mathbf{I})^{-1}} \leq 2\sigma_\nu^2 \log \frac{\sqrt{|(1+\eta)\mathbf{I}+\hat{\mathbf{K}}_t|}}{\delta}\ , \tag{6.39}$$

where $\|\mathbf{x}\|^2_{\mathbf{M}} := \mathbf{x}^\mathsf{T}\mathbf{M}\mathbf{x}$, for a positive-definite matrix $\mathbf{M}$. We also have that, if $\hat{\mathbf{K}}_t$ is almost-surely positive-definite for all $t \geq 1$, the above result holds for $\eta = 0$.

PROOF. Theorem 6.12 is analogous to Theorem 6.4. Therefore, this proof will follow similar steps to the ones taken in the proof of Theorem 6.4, i.e. the proof of Theorem 1 in Chowdhury and Gopalan (2017, App. A). The first part of the proof relies on showing that the following quantity defines a non-negative super-martingale (see Section 2.6.4) relative to the filtration $\{\mathfrak{F}_t\}_{t=0}^\infty$:

$$m_t^{u,s} := \exp\left(\boldsymbol{\nu}_t^\mathsf{T}\mathbf{u}_{t,s} - \frac{\sigma_\nu^2}{2}\|\mathbf{u}_{t,s}\|_2^2\right)\ , \tag{6.40}$$

where $u : \mathcal{W} \to \mathbb{R}$, $\mathbf{u}_{t,s} := [u(\xi_1)+s_1, \dots, u(\xi_t)+s_t]^\mathsf{T}$ with given $s := \{s_i\}_{i=1}^\infty$, $s_i \in \mathbb{R}$. Define:

$$\Delta_t^{u,s} := \exp\left(\nu_t(u(\xi_t)+s_t) - \frac{\sigma_\nu^2}{2}(u(\xi_t)+s_t)^2\right)\ . \tag{6.41}$$

By the sub-Gaussian inequality in Definition 6.2, we have that:

$$\forall t \geq 1, \forall \lambda \in \mathbb{R}, \quad \mathbb{E}\left[\exp\left(\nu_t\lambda - \frac{\sigma_\nu^2}{2}\lambda^2\right)\,\middle|\, \mathfrak{F}_{t-1}\right] \leq 1 \qquad \text{(a.s.)}\ . \tag{6.42}$$

The condition for $\{m_t^{u,s}\}_{t=1}^\infty$ to be a non-negative super-martingale is that:

$$1 \leq i \leq j \implies \mathbb{E}[m_j^{u,s}|\mathfrak{F}_i] \leq m_i^{u,s} \qquad \text{(a.s.)}\ . \tag{6.43}$$

Since all the random variables in $m_{t-1}^{u,s}$ are $\mathfrak{F}_{t-1}$-measurable, we have that:

$$\mathbb{E}[m_t^{u,s}|\mathfrak{F}_{t-1}] = \mathbb{E}[m_{t-1}^{u,s}\Delta_t^{u,s}|\mathfrak{F}_{t-1}] = m_{t-1}^{u,s}\mathbb{E}[\Delta_t^{u,s}|\mathfrak{F}_{t-1}]\ . \tag{6.44}$$

Then, if $\mathbb{E}[\Delta_t^{u,s}|\mathfrak{F}_{t-1}] \leq 1$ (a.s.) holds for all $t \geq 1$, the condition in Equation 6.43 is satisfied.

Compared to Lemma 2 in the proof of Theorem 1 in Chowdhury and Gopalan (2017, Appendix A), the predictability assumption is not required for $\{\xi_t\}_{t=1}^\infty$. Although $\mathbf{x}_t$, as selected by IGP-UCB, is $\mathfrak{F}_{t-1}$-measurable, making Equation 6.42 directly applicable, ξ_t is not $\mathfrak{F}_{t-1}$-measurable. However, we can take expectations of $\Delta_t^{u,s}$ with respect to a larger σ-algebra with respect to which ξ_t is measurable and that contains $\mathfrak{F}_{t-1}$. Let $\mathfrak{F}'_{t-1} := \mathfrak{A}(\mathfrak{F}_{t-1} \cup \mathfrak{A}(\xi_t))$, which is the σ-algebra generated by ξ_t and $\mathfrak{F}_{t-1}$, so that

$\mathfrak{F}_{t-1} \subset \mathfrak{F}'_{t-1}$ and ξ_t is $\mathfrak{F}'_{t-1}$-measurable. In this case, we can say that:

$$\mathbb{E}[\Delta_t^{u,s} \mid \mathfrak{F}'_{t-1}] = \mathbb{E}\left[\exp\left(\nu_t(u(\xi_t) + s_t) - \frac{\sigma_\nu^2}{2}(u(\xi_t) + s_t)^2\right) \middle| \xi_t, \mathfrak{F}_{t-1}\right] \leq 1 \qquad \text{(a.s.)} \qquad (6.45)$$

by taking $\lambda := u(\xi_t) + s_t$ in Equation 6.42. By iterating the conditional expectations (see Section 2.6.3), since $\mathfrak{F}_{t-1} \subset \mathfrak{F}'_{t-1}$, we then have that:

$$\mathbb{E}[\Delta_t^{u,s}|\mathfrak{F}_{t-1}] = \mathbb{E}[\mathbb{E}[\Delta_t^{u,s}|\mathfrak{F}'_{t-1}]|\mathfrak{F}_{t-1}] \leq 1 \qquad \text{(a.s.)}\ . \qquad (6.46)$$

From the latter, it follows that $\{m_t^{u,s}\}_{t=1}^{\infty}$ is a non-negative super-martingale relative to $\{\mathfrak{F}_t\}_{t=0}^{\infty}$.

The rest of the proof follows by repeating the same steps applied in the proof of Theorem 1 in Chowdhury and Gopalan (2017, Appendix A). The remaining steps in their proof are not dependent on the nature of the domain $\mathcal{W}$ nor on the measurability of $\{\xi_t\}_{t=1}^{\infty}$, so that they can be followed exactly. For completeness, the remainder of the proof of Theorem 6.12 is presented in Section A.5. □

As a first step to prove the main result for uGP-UCB (Theorem 6.8), the theorem above generalises Theorem 6.4 to bound the growth of the observation noise vector $\boldsymbol{\nu}_t$ in terms of the uncertain-inputs GP information gain, determined by the known quantity $|(1+\eta)\mathbf{I} + \hat{\mathbf{K}}_t|$ via Equation 6.27. To contrast, Theorem 6.4 was only valid for Euclidean input domains, while the uncertain-inputs GP model operates on a non-Euclidean domain, the space of probability measures $\mathcal{P}$. Another restriction is that Theorem 6.4 required the inputs sequence to be predictable with respect to past data, but the query location distribution estimates P_t^L's parameters can suffer from other sources of randomness, such as execution and localisation noise in robotics applications (see Section 5.3.1). Theorem 6.12 relaxes the predictability assumption, allowing one to bound the observation noise vector in uGP-UCB with an adapted, but not necessarily predictable, sequence of data inputs.

Defining $\mathcal{W} := \mathcal{P}$, the determinant $|(1+\eta)\mathbf{I} + \hat{\mathbf{K}}_t|$ is related, by a logarithmic factor, to the information gain of a uncertain-inputs GP model with covariance $\hat{k}$ and observation noise variance set as $1+\eta$. In both uGP-UCB and IGP-UCB, the GP is only used as a model for $\hat{f} \in \mathcal{H}_{\hat{k}}$, not as a prior distribution. This fact allows one to freely set the GP model's observation noise variance as $1+\eta \neq \sigma_\nu^2$, for some arbitrary $\eta > 0$. The following result then provides an upper confidence bound on the approximation error of this GP model with respect to the true $\hat{f} \in \mathcal{H}_{\hat{k}}$.

THEOREM 6.13. *Let k be a characteristic kernel on $\mathcal{X}$, $f \in \mathcal{H}_k$ and $\|f\|_k \leq b$. Under the assumptions of Theorem 6.12, let $\mathcal{W} := \mathcal{P}$ and $\hat{k}$ be defined as in Equation 6.7. Consider a GP model with covariance* *function $\hat{k}$ and observation noise model variance set to $1+\eta$, $\eta > 0$. Then, given $\mathcal{D}_t = \{(P_i, y_i)\}_{i=1}^t$, with probability at least $1-\delta$, the following holds:*

$$\forall P \in \mathcal{P},\ \forall t \geq 1, \quad |\hat{\mu}_{t-1}(P) - \hat{f}(P)| \leq \left(b + \sigma_\nu \sqrt{2(I(\mathbf{y}_{t-1}; \hat{\mathbf{f}}_t) + 1 + \log(1/\delta))}\right) \hat{\sigma}_{t-1}(P)\,, \tag{6.47}$$

where $I(\mathbf{y}_{t-1}; \hat{\mathbf{f}}_t)$ is the GP information gain, $\hat{\mu}_{t-1}$ and $\hat{\sigma}_{t-1}^2$ are the GP posterior mean and variance given $\mathcal{D}_t$, and $\eta := 2/n$.

PROOF. This result is based on Theorem 2 in Chowdhury and Gopalan (2017) with only the following modifications. The proof in Chowdhury and Gopalan (2017, App. C) is centred around linear-algebra operations with canonical feature maps. As the uncertain-inputs kernel $\hat{k}$ is such that $\hat{k}(P, P') = \langle \psi_P, \psi'_P \rangle_k$, we can use ψ, instead of the canonical feature map $\hat{\phi} : P \mapsto \hat{k}(\cdot, P)$ and still obtain similar results. In this case, the feature space is $\mathcal{H}_k \neq \mathcal{H}_{\hat{k}}$. However, observe that for any $f \in \mathcal{H}_k$, $\langle \psi_P, f \rangle_k = \mathbb{E}_P[f] = \hat{f}(P)$, where $\hat{f} \in \mathcal{H}_{\hat{k}}$ is unique by Lemma 6.1.

For any $f, g \in \mathcal{H}_k$, let $f^\mathsf{T} g := \langle f, g \rangle_k$, so that $f^\mathsf{T} \psi_P = \mathbb{E}_P[f] = \hat{f}(P)$ and $\psi_P^\mathsf{T} \psi_{P'} = \hat{k}(P, P')$ for any $P, P' \in \mathcal{P}$. Given $\{P_i\}_{i=1}^t \subset \mathcal{P}$, define $\mathbf{\Psi}_t := [\psi_{P_1}, \dots, \psi_{P_t}]^\mathsf{T}$, which is analogous to a t-by-∞-matrix or, more precisely, a linear operator from $\mathcal{H}_k$ to $\mathbb{R}^t$. With that, we have $\hat{\mathbf{K}}_t = \mathbf{\Psi}_t \mathbf{\Psi}_t^\mathsf{T}$, and $\hat{\mathbf{k}}_t(P) = \mathbf{\Psi}_t \psi_P$ for all $P \in \mathcal{P}$. For any $\lambda > 0$, the matrices $(\mathbf{\Psi}_t^\mathsf{T} \mathbf{\Psi}_t + \lambda \mathbf{I})$ and $(\mathbf{\Psi}_t \mathbf{\Psi}_t^\mathsf{T} + \lambda \mathbf{I})$ are positive definite, and $(\mathbf{\Psi}_t^\mathsf{T} \mathbf{\Psi}_t + \lambda \mathbf{I}) \mathbf{\Psi}_t^\mathsf{T} = \mathbf{\Psi}_t^\mathsf{T} (\mathbf{\Psi}_t \mathbf{\Psi}_t^\mathsf{T} + \lambda \mathbf{I})$. Then we have that:

$$\mathbf{\Psi}_t^\mathsf{T} (\mathbf{\Psi}_t \mathbf{\Psi}_t^\mathsf{T} + \lambda \mathbf{I})^{-1} = (\mathbf{\Psi}_t^\mathsf{T} \mathbf{\Psi}_t + \lambda \mathbf{I})^{-1} \mathbf{\Psi}_t^\mathsf{T}\,. \tag{6.48}$$

From the definitions, we also have that $(\mathbf{\Psi}_t^\mathsf{T} \mathbf{\Psi}_t + \lambda \mathbf{I}) \psi_P = \mathbf{\Psi}_t^\mathsf{T} \hat{\mathbf{k}}_t(P) + \lambda \psi_P$. Using Equation 6.48, we then get:

$$\psi_P = \mathbf{\Psi}_t^\mathsf{T} (\mathbf{\Psi}_t \mathbf{\Psi}_t^\mathsf{T} + \lambda \mathbf{I})^{-1} \hat{\mathbf{k}}_t(P) + \lambda (\mathbf{\Psi}_t^\mathsf{T} \mathbf{\Psi}_t + \lambda \mathbf{I})^{-1} \psi_P\,, \tag{6.49}$$

which leads to:

$$\lambda f^\mathsf{T} (\mathbf{\Psi}_t^\mathsf{T} \mathbf{\Psi}_t + \lambda \mathbf{I})^{-1} \psi_P = f^\mathsf{T} \psi_P - f^\mathsf{T} \mathbf{\Psi}_t^\mathsf{T} (\mathbf{\Psi}_t \mathbf{\Psi}_t^\mathsf{T} + \lambda \mathbf{I})^{-1} \hat{\mathbf{k}}_t(P) \tag{6.50}$$

for any $f \in \mathcal{H}_k$. In particular, taking $f = \psi_P$, yields:

$$\lambda \psi_P^\mathsf{T} (\mathbf{\Psi}_t^\mathsf{T} \mathbf{\Psi}_t + \lambda \mathbf{I})^{-1} \psi_P = \hat{k}(P, P) - \hat{\mathbf{k}}_t(P)^\mathsf{T} (\mathbf{\Psi}_t \mathbf{\Psi}_t^\mathsf{T} + \lambda \mathbf{I})^{-1} \hat{\mathbf{k}}_t(P) = \hat{\sigma}_t^2(P)\,, \tag{6.51}$$

which is the GP posterior variance.

Considering the formulation of the GP posterior mean $\hat{\mu}_t(P) = \hat{\mathbf{k}}_t(P)^\mathsf{T}(\hat{\mathbf{K}}_t + \lambda\mathbf{I})^{-1}\mathbf{y}_t$, where $\mathbf{y}_t = \hat{\mathbf{f}}_t + \boldsymbol{\nu}_t$ and $\hat{\mathbf{f}}_t := [\hat{f}(P_i)]_{i=1}^t$, we have that:

$$|\hat{f}(P) - \hat{\mu}_t(P)| \leq |\hat{f}(P) - \hat{\mathbf{k}}_t(P)^\mathsf{T}(\hat{\mathbf{K}}_t + \lambda\mathbf{I})^{-1}\hat{\mathbf{f}}_t| + |\hat{\mathbf{k}}_t(P)^\mathsf{T}(\hat{\mathbf{K}}_t + \lambda\mathbf{I})^{-1}\boldsymbol{\nu}_t| \,. \tag{6.52}$$

Observe that:

$$\begin{aligned}
|\hat{f}(P) - \hat{\mathbf{k}}_t(P)^\mathsf{T}(\hat{\mathbf{K}}_t + \lambda\mathbf{I})^{-1}\hat{\mathbf{f}}_t| &= |\psi_P^\mathsf{T} f - \psi_P^\mathsf{T}\boldsymbol{\Psi}_t^\mathsf{T}(\boldsymbol{\Psi}_t\boldsymbol{\Psi}_t^\mathsf{T} + \lambda\mathbf{I})^{-1}\boldsymbol{\Psi}_t f| \\
&= |\lambda\psi_P^\mathsf{T}(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1} f| \\
&\leq \|\lambda(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1}\psi_P\|_k \|f\|_k \\
&= \|f\|_k\sqrt{\lambda\psi_P^\mathsf{T}(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1}\lambda(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1}\psi_P} \\
&\leq b\sqrt{\lambda\psi_P^\mathsf{T}(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1}(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1}\psi_P} \\
&= b\hat{\sigma}_t(P) \,,
\end{aligned} \tag{6.53}$$

where the second equality follows from Equation 6.50, the first inequality is by Cauchy-Schwartz and the last equality is from Equation 6.51. Turning our attention to the second term in the right-hand side of Equation 6.52, we have:

$$\begin{aligned}
|\hat{\mathbf{k}}_t(P)^\mathsf{T}(\hat{\mathbf{K}}_t + \lambda\mathbf{I})^{-1}\boldsymbol{\nu}_t| &= |\psi_P^\mathsf{T}\boldsymbol{\Psi}_t^\mathsf{T}(\boldsymbol{\Psi}_t\boldsymbol{\Psi}_t^\mathsf{T} + \lambda\mathbf{I})^{-1}\hat{\boldsymbol{\nu}}_t| \\
&= |\psi_P^\mathsf{T}(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1}\boldsymbol{\Psi}_t^\mathsf{T}\hat{\boldsymbol{\nu}}_t| \\
&\leq \left\|(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1/2}\psi_P\right\|_k \left\|(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1/2}\boldsymbol{\Psi}_t^\mathsf{T}\hat{\boldsymbol{\nu}}_t\right\|_k \\
&= \sqrt{\psi_P^\mathsf{T}(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1}\psi_P}\sqrt{(\boldsymbol{\Psi}_t^\mathsf{T}\hat{\boldsymbol{\nu}}_t)^\mathsf{T}(\boldsymbol{\Psi}_t^\mathsf{T}\boldsymbol{\Psi}_t + \lambda\mathbf{I})^{-1}\boldsymbol{\Psi}_t^\mathsf{T}\hat{\boldsymbol{\nu}}_t} \\
&= \lambda^{-1/2}\hat{\sigma}_t(P)\sqrt{\hat{\boldsymbol{\nu}}_t^\mathsf{T}\boldsymbol{\Psi}_t\boldsymbol{\Psi}_t^\mathsf{T}(\boldsymbol{\Psi}_t\boldsymbol{\Psi}_t^\mathsf{T} + \lambda\mathbf{I})^{-1}\hat{\boldsymbol{\nu}}_t} \\
&= \lambda^{-1/2}\hat{\sigma}_t(P)\sqrt{\hat{\boldsymbol{\nu}}_t^\mathsf{T}\hat{\mathbf{K}}_t(\hat{\mathbf{K}}_t + \lambda\mathbf{I})^{-1}\hat{\boldsymbol{\nu}}_t} \,,
\end{aligned} \tag{6.54}$$

where the second equality uses Equation 6.48, the inequality is by Cauchy-Schwartz, and the fourth equality applies both Equation 6.51 and again Equation 6.48. Thus, combining the two last results into Equation 6.52, we have:

$$|\hat{f}(P) - \hat{\mu}_t(P)| \leq \hat{\sigma}_t(P)\left(b + (1+\eta)^{-1/2}\sqrt{\hat{\boldsymbol{\nu}}_t^\mathsf{T}\hat{\mathbf{K}}_t(\hat{\mathbf{K}}_t + (1+\eta)\mathbf{I})^{-1}\hat{\boldsymbol{\nu}}_t}\right) \,, \tag{6.55}$$

where $\lambda := 1 + \eta$, $\eta \geq 0$ as in Theorem 6.12. A few linear algebra manipulations yield:

$$(\hat{\mathbf{K}}_t + \eta \mathbf{I})(\hat{\mathbf{K}}_t + (1+\eta)\mathbf{I})^{-1} = ((\hat{\mathbf{K}}_t + \eta\mathbf{I})^{-1} + \mathbf{I})^{-1} \,. \tag{6.56}$$

With that, we obtain:

$$\hat{\boldsymbol{\nu}}_t^\mathsf{T} \hat{\mathbf{K}}_t (\hat{\mathbf{K}}_t + (1+\eta)\mathbf{I})^{-1} \hat{\boldsymbol{\nu}}_t \leq \hat{\boldsymbol{\nu}}_t^\mathsf{T} (\hat{\mathbf{K}}_t + \eta\mathbf{I})(\hat{\mathbf{K}}_t + (1+\eta)\mathbf{I})^{-1} \hat{\boldsymbol{\nu}}_t = \hat{\boldsymbol{\nu}}_t^\mathsf{T} ((\hat{\mathbf{K}}_t + \eta\mathbf{I})^{-1} + \mathbf{I})^{-1} \hat{\boldsymbol{\nu}}_t \,. \tag{6.57}$$

Now we can apply Theorem 6.12 to upper-bound this last term, which leads to:

$$\begin{aligned} \forall t \geq 0, \forall P \in \mathcal{P}, \quad |\hat{\mu}_t(P) - \hat{f}(P)| &\leq \hat{\sigma}_t(P) \left(b + \|\hat{\boldsymbol{\nu}}_t\|_{((\hat{\mathbf{K}}_t + \eta\mathbf{I})^{-1} + \mathbf{I})^{-1}} \right) \\ &\leq \hat{\sigma}_t(P) \left(b + \sigma_\nu \sqrt{2 \log \frac{|(1+\eta)\mathbf{I} + \hat{\mathbf{K}}_t|^{1/2}}{\delta}} \right) \end{aligned} \tag{6.58}$$

with probability[5] at least $1 - \delta$.

As a final step, observe that $|(1+\eta)\mathbf{I} + \hat{\mathbf{K}}_t| = |\mathbf{I} + (1+\eta)^{-1}\hat{\mathbf{K}}_t||(1+\eta)\mathbf{I}|$. Then we have:

$$\log|(1+\eta)\mathbf{I} + \hat{\mathbf{K}}_t| = \log|\mathbf{I} + (1+\eta)^{-1}\hat{\mathbf{K}}_t| + t\log(1+\eta) = 2I(\mathbf{y}_t; \hat{\mathbf{f}}_t) + \eta t \tag{6.59}$$

Setting $\eta := 2/n$, with n as the total number of iterations (see Algorithm 7), the result in Theorem 6.13 follows. □

One can use the error bounds from Theorem 6.13 to bound the cumulative expected regret $\hat{R}_n$ for uGP-UCB. However, there is a dependence on the information gain $I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{P_t^L\}_{t=1}^n)$ and on the predictive variances $\hat{\sigma}_{t-1}^2(\hat{P}_{\mathbf{x}_t}^t)$, which are also related to information gain (see Equation 2.42). As mentioned at the start of the section, a reasonable assumption is to consider the existence of an upper bound $\hat{\gamma}_n^L$ for the information gain which does not depend on a particular choice of distributions. Proposition 6.10 presented a special case where this assumption holds with the deterministic-inputs information gain γ_n as an upper bound. Consider then that:

$$\forall n \geq 1, \quad I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{P_t^L\}_{t=1}^n) \leq \hat{\gamma}_n^L \qquad \text{(a.s.)}\,, \tag{6.60}$$

where the notation $I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{P_t^L\}_{t=1}^n)$ emphasises the dependence on the given set of inputs $\{P_t^L\}_{t=1}^n$, i.e.:

$$I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{P_t^L\}_{t=1}^n) = \frac{1}{2}\log|\mathbf{I} + (1+\eta)^{-1}\hat{\mathbf{K}}_n| \,, \tag{6.61}$$

[5] Notice that for $t = 0$, we simply have $|\hat{\mu}_0(P) - \hat{f}(P)| = |\hat{f}(P)| \leq \|\hat{f}\|_{\hat{k}} \sqrt{\hat{k}(P,P)} = \|f\|_k \hat{\sigma}_0(P) \leq b\hat{\sigma}_0(P)$, which is always true, as it follows from the Cauchy-Schwartz inequality, for a zero-mean GP model and $b \geq \|f\|_k = \|\hat{f}\|_{\hat{k}}$.

with $[\hat{\mathbf{K}}_n]_{ij} = \hat{k}(P_i^L, P_j^L),\ i, j \in \{1, \dots, n\}$. One may also consider an adaptive model $\hat{P}_{\mathbf{x}}^t$ to learn $P_{\mathbf{x}}^E$ from the previous t observations. Therefore, consider that:

$$\forall n \geq 1, \quad \exists \hat{\gamma}_n^Q > 0 : \quad I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{\hat{P}_{\mathbf{x}_t}^t\}_{t=1}^n) \leq \hat{\gamma}_n^Q \qquad \text{(a.s.)} \ . \tag{6.62}$$

Now we are ready to prove the main result, which is restated below, followed by the proof.

THEOREM 6.8 (Restatement from p. 117). *Let $\delta \in (0,1)$, $f \in \mathcal{H}_k$, and $b \geq \|f\|_k$. Consider ζ_t as σ_ζ-sub-Gaussian noise. Assume that both k and location estimates P_t^L satisfy the conditions for $\Delta f_{P_t^L}$ to be σ_F-sub-Gaussian, for a given $\sigma_F > 0$, for all $t \geq 1$. Then, running uGP-UCB with:*

$$\beta_t = b + \sqrt{2(\sigma_F^2 + \sigma_\zeta^2)(I(\mathbf{y}_{t-1}; \hat{\mathbf{f}}_{t-1} \mid \{P_i^L\}_{i=1}^t) + 1 + \log(1/\delta))} \tag{6.28}$$

and a model $\mathbf{x} \mapsto \hat{P}_{\mathbf{x}}^t$, such that:

$$\forall t \geq 1, \exists \rho_t \geq 0 : \max_{\mathbf{x} \in \mathcal{S}} \left| \mathbb{E}_{P_{\mathbf{x}}^E}[f] - \mathbb{E}_{\hat{P}_{\mathbf{x}}^t}[f] \right| \leq \rho_t \qquad \text{(a.s.)} \ , \tag{6.29}$$

the cumulative expected regret satisfies:

$$\mathbb{P}\left\{ \hat{R}_n \in \mathcal{O}\left(\sum_{t=1}^n \rho_t + \sqrt{n \hat{\gamma}_n^Q} \left(b + \sqrt{\hat{\gamma}_n^L + \log(1/\delta)} \right) \right) \right\} \geq 1 - \delta \ . \tag{6.30}$$

PROOF. The following derivations will first apply Theorem 6.13 and the error bound ρ_t to upper-bound the expected regret $\hat{r}_t$ for each choice of uGP-UCB. With these bounds and the monotonicity of β_t, the cumulative expected regret can be upper bounded. A few algebraic manipulations will transform the dependence on the distributions P_t^L and $\hat{P}_{\mathbf{x}_t}^t$ into a dependence on the corresponding information gain upper bounds.

Assuming that the mapping $\mathbf{x} \to \hat{P}_{\mathbf{x}}^t$ satisfies Equation 6.29, by Lemma 6.1 and Theorem 6.13 with $\sigma_\nu^2 := \sigma_F^2 + \sigma_\zeta^2$, we have that:

$$\begin{aligned}\forall \mathbf{x} \in \mathcal{S}, \forall t \geq 1, \quad |\hat{\mu}_{t-1}(\hat{P}_{\mathbf{x}}^t) - \mathbb{E}_{P_{\mathbf{x}}^E}[f]| &\leq |\hat{\mu}_{t-1}(\hat{P}_{\mathbf{x}}^t) - \mathbb{E}_{\hat{P}_{\mathbf{x}}^t}[f]| + |\mathbb{E}_{\hat{P}_{\mathbf{x}}^t}[f] - \mathbb{E}_{P_{\mathbf{x}}^E}[f]| \\ &\leq \beta_t \hat{\sigma}_{t-1}(\hat{P}_{\mathbf{x}}^t) + \rho_t \ .\end{aligned} \tag{6.63}$$

with probability at least $1 - \delta$. The choice of $\mathbf{x}_t$ by uGP-UCB is such that:

$$\forall t \geq 1, \quad \max_{\mathbf{x} \in \mathcal{S}} \mathbb{E}_{\hat{P}_{\mathbf{x}}^t}[f] \leq \max_{\mathbf{x} \in \mathcal{S}} \hat{\mu}_{t-1}(\hat{P}_{\mathbf{x}}^t) + \beta_t \hat{\sigma}_{t-1}(\hat{P}_{\mathbf{x}}^t) = \hat{\mu}_{t-1}(\hat{P}_{\mathbf{x}_t}^t) + \beta_t \hat{\sigma}_{t-1}(\hat{P}_{\mathbf{x}_t}^t) \tag{6.64}$$

with high probability. We also have that:

$$\begin{aligned}\forall t \geq 1, \quad \max_{\mathbf{x}\in\mathcal{S}} \mathbb{E}_{P^E_{\mathbf{x}}}[f] &= \max_{\mathbf{x}\in\mathcal{S}} \mathbb{E}_{P^E_{\mathbf{x}}}[f] - \mathbb{E}_{\hat{P}^t_{\mathbf{x}}}[f] + \mathbb{E}_{\hat{P}^t_{\mathbf{x}}}[f] \\ &\leq \max_{\mathbf{x}\in\mathcal{S}} \left|\mathbb{E}_{P^E_{\mathbf{x}}}[f] - \mathbb{E}_{\hat{P}^t_{\mathbf{x}}}[f]\right| + \max_{\mathbf{x}\in\mathcal{S}} \mathbb{E}_{\hat{P}^t_{\mathbf{x}}}[f] \\ &\leq \rho_t + \max_{\mathbf{x}\in\mathcal{S}} \mathbb{E}_{\hat{P}^t_{\mathbf{x}}}[f] \qquad \text{(a.s.)}\end{aligned} \tag{6.65}$$

Consequently, applying Equation 6.63, we obtain:

$$\begin{aligned}\forall t \geq 1, \quad \hat{r}_t &= \max_{\mathbf{x}\in\mathcal{S}} \mathbb{E}_{P^E_{\mathbf{x}}}[f] - \mathbb{E}_{P^E_{\mathbf{x}_t}}[f] \\ &\leq \rho_t + \beta_t \hat{\sigma}_{t-1}(\hat{P}^t_{\mathbf{x}_t}) + \hat{\mu}_{t-1}(\hat{P}^t_{\mathbf{x}_t}) - \mathbb{E}_{P^E_{\mathbf{x}}}[f] \\ &\leq 2\rho_t + 2\beta_t \hat{\sigma}_{t-1}(\hat{P}^t_{\mathbf{x}_t})\end{aligned} \tag{6.66}$$

with high probability. As $\beta_n \geq \beta_t,\ \forall t \leq n$, it then follows that:

$$\sum_{t=1}^{n} \hat{r}_t \leq 2\sum_{t=1}^{n} (\beta_t \hat{\sigma}_{t-1}(\hat{P}^t_{\mathbf{x}_t}) + \rho_t) \leq 2\left(\beta_n \sum_{t=1}^{n} \hat{\sigma}_{t-1}(\hat{P}^t_{\mathbf{x}_t})\right) + 2\sum_{t=1}^{n} \rho_t \tag{6.67}$$

with probability at least $1-\delta$. Bounding the first term in the last inequality, we have that:

$$\sum_{t=1}^{n} \hat{\sigma}_{t-1}(\hat{P}^t_{\mathbf{x}_t}) \leq \left(n \sum_{t=1}^{n} \hat{\sigma}^2_{t-1}(\hat{P}^t_{\mathbf{x}_t})\right)^{1/2} \tag{6.68}$$

by the Cauchy-Schwartz inequality.

Under the general regularity assumptions in Section 6.3, $k(\mathbf{x},\mathbf{x}) \leq 1, \forall \mathbf{x} \in \mathcal{X}$, so that $\hat{k}(P,P) \leq 1$ and $0 \leq \hat{\sigma}^2_{t-1}(P) \leq 1, \forall P \in \mathcal{P}$. As $\log(1+\alpha) \geq \alpha/2$, for $0 \leq \alpha \leq 1$, using $\eta > 0$ as in Theorem 6.13, we have that:

$$(1+\eta)^{-1}\hat{\sigma}^2_{t-1}(P) \leq 2\log(1 + (1+\eta)^{-1}\hat{\sigma}^2_{t-1}(P))\ . \tag{6.69}$$

Therefore, the GP predictive variance is bounded by:

$$\hat{\sigma}^2_{t-1}(P) \leq 2(1+\eta)\log(1 + (1+\eta)^{-1}\hat{\sigma}^2_{t-1}(P))\ . \tag{6.70}$$

We can use Lemma 5.3 in (Srinivas et al., 2012) to relate the predictive variance to the information gain by:

$$I(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{\hat{P}^t_{\mathbf{x}_t}\}_{t=1}^n) = \frac{1}{2}\sum_{t=1}^{n} \log\left(1 + (1+\eta)^{-1}\hat{\sigma}^2_{t-1}(\hat{P}^t_{\mathbf{x}_t})\right)\ . \tag{6.71}$$

With this result, we obtain:

$$\begin{aligned}\sum_{t=1}^{n} \hat{\sigma}_{t-1}(\hat{P}^t_{\mathbf{x}_t}) &\leq \left(2(1+\eta)n \sum_{t=1}^{n} \log\left(1 + (1+\eta)^{-1}\hat{\sigma}^2_{t-1}(\hat{P}^t_{\mathbf{x}_t})\right)\right)^{1/2} \\ &= \sqrt{4(1+\eta)nI(\mathbf{y}_n; \hat{\mathbf{f}}_n \mid \{\hat{P}^t_{\mathbf{x}_t}\}^n_{t=1})} \\ &\leq \sqrt{4(1+\eta)n\hat{\gamma}^Q_n} \in \mathcal{O}\left(\sqrt{n\hat{\gamma}^Q_n}\right) .\end{aligned} \tag{6.72}$$

At the same time, considering β_n in Equation 6.67, we have:

$$\begin{aligned}\beta_n &= b + \sigma_\nu \sqrt{2\left(I(\mathbf{y}_{n-1}; \hat{\mathbf{f}}_{n-1} \mid \{P^L_t\}^{n-1}_{t=1}) + 1 + \log(1/\delta)\right)} \\ &\leq b + \sigma_\nu \sqrt{2(\hat{\gamma}^L_n + 1 + \log(1/\delta))} ,\end{aligned} \tag{6.73}$$

where $\hat{\mathbf{f}}_n \sim N(\mathbf{0}, \hat{\mathbf{K}}_n)$ with $[\hat{\mathbf{K}}_n]_{ij} = \hat{k}(P^L_i, P^L_j)$. Combining these results yields:

$$\begin{aligned}\hat{R}_n = \sum_{t=1}^{n} \hat{r}_t &\leq 2\left(\sqrt{4(1+\eta)n\hat{\gamma}^Q_n}\left(b + \sigma_\nu\sqrt{2(\hat{\gamma}^L_n + 1 + \log(1/\delta))}\right) + \sum_{t=1}^{n} \rho_t\right) \\ &\in \mathcal{O}\left(b\sqrt{n\hat{\gamma}^Q_n} + \sqrt{n\hat{\gamma}^Q_n(\hat{\gamma}^L_n + \log(1/\delta))} + \sum_{t=1}^{n}\rho_t\right)\end{aligned} \tag{6.74}$$

with probability at least $1 - \delta$, which concludes the proof. □

6.6 Experiments

This section presents experiments analysing the performance of uGP-UCB and IGP-UCB (adapted to input noise via Theorem 6.5) empirically. Comparisons with the unscented expected improvement (UEI) method, proposed in Nogueira et al. (2016), are also presented. The purpose of these experiments is to both verify the theoretical results in the previous section and to also test each BO approach in situations that do not meet their theoretical assumptions, including a case involving a robotic simulation.

6.6.1 Theory check

To verify the theoretical results, this experiment tested IGP-UCB and uGP-UCB under ideal settings where the theoretical assumptions are met. The objective function was set as $f = \sum_{i=1}^{q} \alpha_i k(\cdot, \mathbf{x}_i) \in \mathcal{H}_k$, where $\{\mathbf{x}_i\}_{i=1}^{n}$ are uniformly sampled from $[0,1]^2 \subset \mathbb{R}^2$ and $\boldsymbol{\alpha} \sim N(\mathbf{0}, \mathbf{K})$ with $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $i, j \in \{1, \dots, q\}$, and $q = 20$. The kernel was the squared exponential with length-scale set to 0.1.

The search space for IGP-UCB was set as a uniformly sampled subset of the unit box, i.e. $\mathcal{S} \subset [0,1]^2$ with $|\mathcal{S}| = 100$. For uGP-UCB, query distributions were picked from a finite set $\mathcal{R} = \{N(\mathbf{x}, \mathbf{\Sigma}) \mid \mathbf{x} \in \mathcal{S}\} \subset \mathcal{P}$, where $\mathbf{\Sigma} = \sigma_E^2 \mathbf{I}$ with $\sigma_E = 0.1$, and $\hat{P}_{\mathbf{x}} = P_{\mathbf{x}}^E$ for this experiment. Having the search space for both algorithms configured as finite sets enables fast recursive updates for the GP predictions (see Chowdhury and Gopalan, 2017, App. F), so that tests could be run for as many iterations as needed to verify asymptotic performance. The UCB parameters are set as $\delta = 0.1$, and the RKHS norm bound b was directly computed as $\|f\|_k = \sqrt{\boldsymbol{\alpha}^\mathsf{T} \mathbf{K} \boldsymbol{\alpha}}$. The observation noise parameter σ_F was computed according to Lemma 6.6 and white Gaussian noise $\zeta_t \sim N(0, \sigma_\zeta^2)$ was added to each observation. The location estimates P_t^L were provided simply using $P_t^L = P_{\mathbf{x}_t}^E$ for these tests.

The theoretical upper bound on the expected regret was computed from Theorem 6.13's result:

$$\forall \mathbf{x} \in \mathcal{S}, \quad |\mathbb{E}_{P_{\mathbf{x}}^E}[f] - \hat{\mu}_{t-1}(P_{\mathbf{x}}^E)| \leq \beta_t \hat{\sigma}_{t-1}(P_{\mathbf{x}}^E)\,, \quad \forall t \geq 1\,, \tag{6.75}$$

which should hold with probability greater than $1 - \delta$ for β_t set according to Theorem 6.8. Then, with the same probability, we should have that:

$$\forall t \geq 1, \quad \hat{r}_t = \max_{\mathbf{x} \in \mathcal{S}} \mathbb{E}_{P_{\mathbf{x}}^E}[f] - \mathbb{E}_{P_{\mathbf{x}_t}^E}[f] \leq 2\beta_t \hat{\sigma}_{t-1}(P_{\mathbf{x}_t}^E)\,, \tag{6.76}$$

and consequently,

$$\hat{R}_n = \sum_{t=1}^n \hat{r}_t \leq 2 \sum_{t=1}^n \beta_t \hat{\sigma}_{t-1}(P_{\mathbf{x}_t}^E)\,. \tag{6.77}$$

An analogue result can be derived for IGP-UCB by replacing the GP predictive mean and variance.

Results evaluating the expected regret and the upper bound above are presented in Figure 6.1. As the plot shows, the regret of each algorithm stays within the theoretical bounds with a gap of almost an order of magnitude. It's also noticeable that uGP-UCB's regret is lower both in theory and in practice. The plot also shows that each algorithm's mean regret tends to zero with a well-defined downward slope. Recall that the mean expected regret is an upper bound on the minimum expected regret up to a given iteration, as $\min_{t \leq n} \hat{r}_t \leq \hat{R}_n/n$, so that a vanishing mean expected regret $\hat{R}_n/n \to 0$ as $n \to \infty$ indicates that the algorithm eventually reaches the global optimum.

6.6.2 Objective functions in the same RKHS

In this experiment, for each trial a different objective function was generated by uniformly sampling a set of 30 weights α_i and support points $\mathbf{x}_i$ for the kernels, with $f = \sum_{i=1}^q \alpha_i k(\cdot, \mathbf{x}_i) \in \mathcal{H}_k$. The search

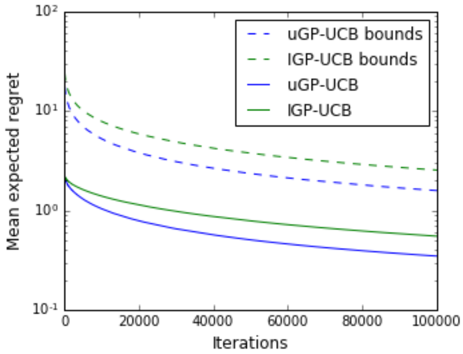
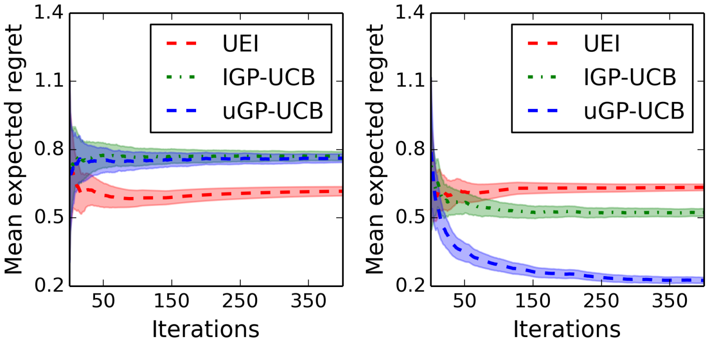


FIGURE 6.1: Theory check: The plot shows the mean expected regret and its theoretical upper bound for each UCB algorithm. Results were averaged over 10 trials, each with randomly generated input and output noise.

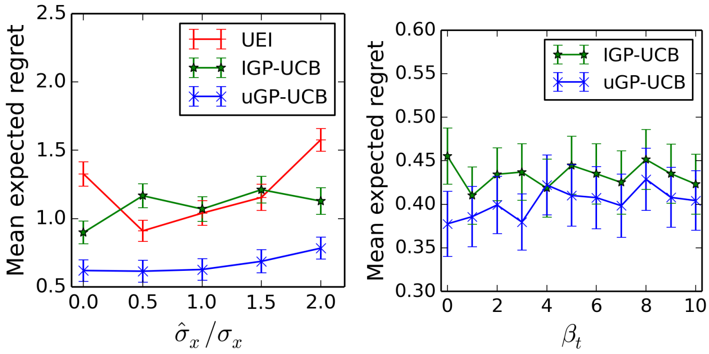


(A) 2D, auto

(B) 2D, fixed

FIGURE 6.2: Mean expected regret for IGP-UCB, UEI, and uGP-UCB in the optimisation of functions in the same RKHS: in (a) the UCB confidence-bound parameter β_t was set according to the theoretical results, while in (b), it was fixed at $\beta_t = 3$ for all t. The results were averaged over 20 trials, and the shaded areas correspond to one standard deviation.

space was set to the unit box $\mathcal{S} = [0, 1]^2 \subset \mathbb{R}^2$. To verify the theoretical results for the UCB algorithms, the parameters are set as $\delta = 0.4$, and b was directly computed as $\|f\|_k$. The querying execution noise on $P^E_{\mathbf{x}}$ was generated from i.i.d from $N(\mathbf{0}, \sigma^2_{\mathbf{x}}\mathbf{I})$ with $\sigma_{\mathbf{x}} = 0.1$. The output noise parameters for the GP model were computed according to Proposition 6.6, but each method was configured assuming an underestimated execution noise distribution $N(\mathbf{0}, \hat{\sigma}^2_{\mathbf{x}}\mathbf{I})$, with $\hat{\sigma}_{\mathbf{x}} = \frac{1}{2}\sigma_{\mathbf{x}}$. Noise on the localisation estimates P^L_t was also set at half the standard deviation of the execution noise. Kernel length-scales were set isotropic to 0.1. The maximum information gain was set according to the results in Srinivas et al. (2012).

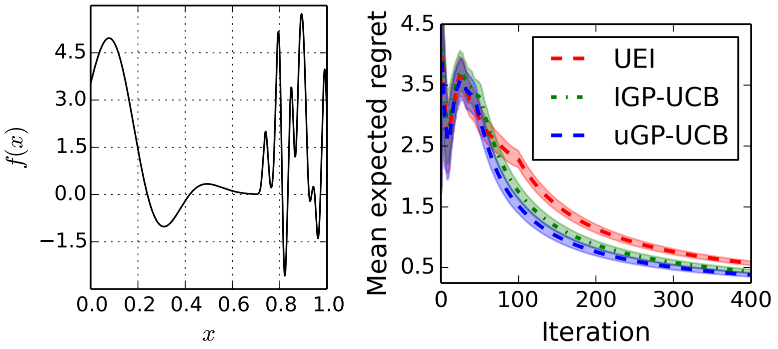


(A) Noise mis-specification (B) Parameter sentitivity

FIGURE 6.3: Robustness: (a) presents how each algorithm's performance is affected by mis-specification of the execution noise variance for the experiment with functions in the same RKHS, while (b) presents how the UCB methods performance varies under different settings for β_t in the case of the objective function in a different RKHS. In both cases, the mean expected regret shown is computed at the end of 200 iterations and averaged over 10 trials.

Figure 6.2a presents performance results, in terms of mean expected regret, obtained by uGP-UCB under the assumptions of Theorem 6.8, IGP-UCB with parameters adapted according to Theorem 6.5's settings, and UEI. As the results show, in this setup, uGP-UCB is able to outperform IGP-UCB, if following the setup in Theorem 6.8. UEI is able to significantly outperform both UCB methods, though with a growing mean expected regret. This fact indicates that UEI's cumulative expected regret $\hat{R}_n$ is not growing sub-linearly with the number of iterations n. One of the main reasons for the large regret of the UCB methods in this experiment is that cumulative regret bounds are valid up to a constant factor, due to the use of the $\mathcal{O}$ notation, and their focus is on guaranteeing asymptotic convergence, i.e. $\lim_{n\to\infty} \hat{R}_n/n = 0$, as most theoretical results in the UCB literature (Srinivas et al., 2012; Chowdhury and Gopalan, 2017). Thus, the method might need many more iterations to show a vanishing regret, at prohibitive computational costs due to the GP's $\mathcal{O}(n^3)$ update complexity (Rasmussen and Williams, 2006). In Chowdhury and Gopalan (2017) and Section 6.6.1, results were shown for n in the order of 10^4 iterations, which was possible due to low-cost updates for the GP predictive equations over discrete search spaces (see Chowdhury and Gopalan, 2017, App. F). In this experiment, the method is evaluated over a continuous search space, instead, making it unable to apply the same approximations.

As Figure 6.2b presents, for a fixed β_t, UCB-based methods can quickly outperform UEI. For the latter, the expected improvement over the best objective function value observed so far is computed over the GP model by applying the unscented transform to the standard EI estimate. This transformed estimate



(A) Objective

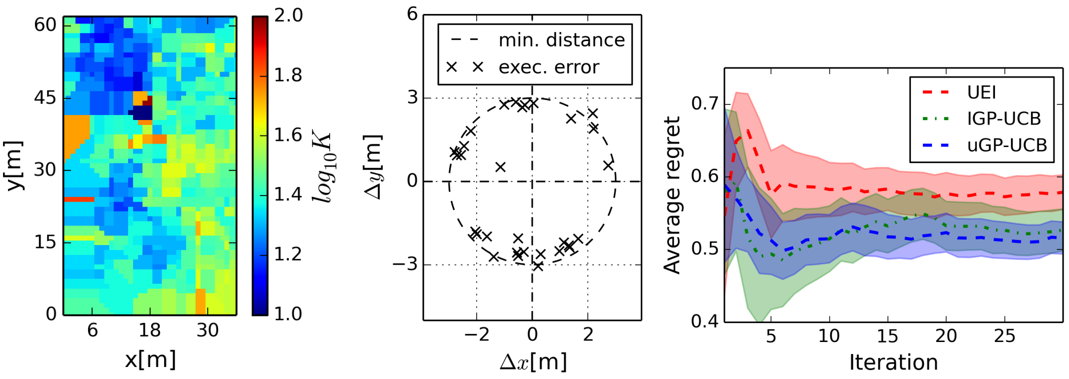
(B) Performance

FIGURE 6.4: Performance results for objective function in different RKHS: (a) presents the objective function and (b) the mean expected regret for each algorithm. Results were averaged over 10 trials.

corresponds to the expected value of the EI under the input-noise distribution, which in this experiment is under-estimated with a quarter of the true variance. As Figure 6.3a presents, UCB methods are more robust to mis-specification of the execution noise having smaller variation in performance, being actually favoured by assuming less execution noise than the true amount. This behaviour can be explained by considering that, assuming less execution noise, GP-UCB's estimated upper bound on the objective function value is higher than needed, but still a valid upper bound, maintaining the algorithm's assumptions.

6.6.3 Objective function in different RKHS

To verify uGP-UCB's performance under wrong kernel assumptions, the next experiment performed tests with an objective function in a different RKHS. This function, shown in Figure 6.4a also appeared in experiments in Nogueira et al. (2016). Figure 6.4b presents performance results for fixed $\beta_t = 3$. Figure 6.3b evaluates each algorithm's sensitivity to the choice of β_t as a way to asses the robustness of the methods when theoretical assumptions are not met. Input noise was set with $\sigma_{\mathbf{x}} = 0.01$. As seen, both the proposed uGP-UCB and IGP-UCB can outperform the unscented BO approach UEI. In addition, one can see that uGP-UCB shows consistently better performance than that of IGP-UCB across varying settings for β_t.



(A) Potassium concentration

(B) Execution error

(C) Performance

FIGURE 6.5: Robotic exploration experiment: (a) presents the Broom's barn data as distributed over the search space; (b) presents a typical execution error distribution as observed from one of the runs in the experiment; and (c) shows the performance of each BO approach, averaged over 4 runs.

6.6.4 Robotic exploration problem

This section presents results obtained in a simulated robotic exploration problem. In this experiment, a robot is set to explore an environmental process by measuring the concentration of potassium in the soil of a farm. The underlying process is based on the Broom's Barn dataset [6], consisting of the log-concentration of potassium in the soil of an experimental agricultural area. This data was spread over a 36 by 62 metres search space as shown in Figure 6.5a. The robot is allowed to perform up to 30 measurements on different locations. Each BO method sequentially selects the locations where the robot should make a measurement in the usual online decision making process, based on the observations it gets. The robot was simulated using the OpenRobots' Morse simulator [7]. In this scenario, execution noise is not following a stationary distribution due to the dynamic constraints of the robot, the imperfections in motion control etc. Figure 6.5b presents a typical execution noise distribution, when marginalised over different starting and goal locations from a given run of the experiment. As seen in the figure, the noise distribution is not Gaussian, but it's concentrated on average 3 metres away from the target location, since this was the threshold given to the robot's path-following control as the minimum distance to a target location. The noise around the ring formed by the distribution is a consequence of localisation uncertainty and imperfections on the robot's motion control. The hyper-parameters for each GP were learnt online via log-marginal likelihood maximisation, after each sample. β_t is set at a fixed value for

[6] Dataset available at `http://www.kriging.com/datasets/`

[7] Morse: `https://www.openrobots.org/morse`

this experiment, again with $\beta_t = 3$. Figure 6.5c presents the performance of each algorithm in terms of regret. The plots show that uGP-UCB is able to significantly outperform UEI, while performing still better than IGP-UCB in the long run. This result confirms that it is possible to obtain better performance in practical BO problems by taking advantage of distribution estimates and by directly considering execution uncertainty.

6.7 Summary

This chapter proposed a novel approach to optimise functions where both the samples of the function values as well as the location at which the function is sampled are stochastic. Theoretical guarantees were provided on the optimality of the proposed uGP-UCB algorithm and the standard IGP-UCB method, when adapted to the noisy-inputs settings. Experiments have shown competitive performance when compared to state-of-the-art methods (Nogueira et al., 2016), while at the same time providing the first theoretical guarantees for this kind of setup. For future work, it would be worth investigating learning techniques for the approximate querying distribution $\hat{P}_{\mathbf{x}}$ and better bounds for the uncertain-inputs GP model information gain, possibly by means of information geometry (Ay et al., 2017) techniques.

CHAPTER 7

Conclusions

This thesis addressed the problem of planning under uncertainty. Bayesian optimisation was applied as a general framework to select actions to optimise partially-observable objective functions. Different methods were developed tackling issues that arise in real-world problems which uncertainty affects in different ways. This chapter summarises the thesis' contributions and presents future research directions.

7.1 Contributions

The following reviews the contributions of this thesis organised per topic.

7.1.1 Trajectory optimisation

Chapter 3 presented a method to optimise trajectories over a terrain model that is learnt online. The optimisation goal was to minimise the required energy expenditure for a robot to travel from a given starting point to a final location. This method employed Gaussian process regression to learn a mapping from individual robot configurations to the power demanded by moving over the terrain at that configuration. An acquisition function taking integrals over the GP estimates was proposed as a method to evaluate candidate paths. Experiments showed that the BO approach outperforms other optimisation methods and is also applicable to practical situations where a physical robot needs to learn how to efficiently traverse unforeseen terrain.

7.1.2 Search in high-dimensional parameter spaces

Chapter 4 addressed problems involving a large number of parameters to optimise. Classic BO algorithms traditionally degrade in performance as the dimensionality of the search space increases. The method proposed in Chapter 4 addressed this issue by combining BO with coordinate-descent strategies, which allow simplifying the optimisation problem. In addition, the method also took advantage of the fact that solutions to heavily-constrained optimisation problems are usually close to each other. In this case, starting the optimisation from a known valid solution, instead of at a random point, helps the algorithm to reduce the number of iterations needed to find a promising solution. The proposed BO method was applied to optimise control policies for a simulated race car. Experiments demonstrated that the proposed approach is able to maintain performance across increasing search space dimensionality.

7.1.3 Optimisation under uncertain inputs

The last two chapters (5 and 6) addressed problems where uncertainty affects the estimates of where a sample was collected. This uncertainty affects conventional models learnt with BO in a way that is not usually addressed. Chapter 5 then proposed using GP models that take input noise into account as priors for BO. These models can take probability distributions as representations of an uncertain input, instead of simply using point estimates or assuming that the query was performed exactly where it was specified to be. Chapter 5 was concerned with an application where a robot needs to learn a terrain roughness model as it drives while keeping itself safe by avoiding areas that cause excessive vibration to its chassis. The next chapter generalised this idea to a general optimisation setup and provided theoretical guarantees for such a BO method. Experiments in both chapters also demonstrated the performance of the proposed BO methods when compared to alternatives in simulation and with a physical robot.

7.2 Future work

This section presents a few areas for future work where the methods in this thesis can contribute to.

7.2.1 Functional optimisation

Both Chapter 3 and Chapter 4 can fit into the general problem of functional optimisation (Devolder et al., 2010; Vien and Toussaint, 2018). A functional is basically a function of a function. An integral, for example, is a functional, since it takes as input a function and returns a scalar corresponding to the area under the function's graph. Both the energy cost of a path and the reward proportioned by a control policy are functional objectives. Recently proposed methods for Bayesian optimisation in infinite-dimensional spaces (Kasim and Norreys, 2016; Vien and Toussaint, 2018) can be combined with the techniques in chapters 3 and 4 to yield new methods for functional optimisation.

7.2.2 Partially observable Markov decision processes

Chapters 5 and 6 addressed problems where both the execution of a query and the estimation of the true query location are uncertain. These problems are prevalent in partially-observable Markov decision process (POMDP) frameworks (Mcallister and Rasmussen, 2017). Although BO methods have been previously applied to solve POMDP problems (Marchant and Ramos, 2014; Morere et al., 2016), such methods still use point estimates in observations, while marginalising over state distributions. Therefore, the methodology presented in Chapter 6, which can operate directly on the space of probability distributions, may lead to new possibilities for BO methods to solve POMDP problems.

7.2.3 Policy search

The CDBO method in Chapter 4 can be applied to other tasks where policies are parameterised by a large number of parameters. An issue that happens in such problems is that the solution space might be heavily constrained presenting sharp transitions and a non-smooth reward surface. CDBO was able to perform well in such problems by using initial guesses in the valid region of the parameter space. However, combining CDBO with methods that can learn the constrained region of the search space (Achiam et al., 2017) and adapt to the irregularities of the reward function (Martinez-Cantin, 2017) should allow for even more efficient optimisation schemes for control policies.

7.2.4 Non-stationary priors

Non-stationary covariance functions allow Gaussian process models to learn functions whose level of smoothness varies across the domain (Rasmussen and Williams, 2006). In the BO case, these have been applied to learn objective functions presenting sharp transitions, as in the policy-search cases mentioned above, or modelling complex physical simulations (Martinez-Cantin, 2017). The type of covariance function applied in Chapter 6 presents non-stationary properties if considering maps $\mathbf{x} \mapsto \hat{P}_{\mathbf{x}}$, where the distribution $\hat{P}_{\mathbf{x}} \in \mathcal{P}$ allows locally varying levels of smoothness. Input-warping methods, which similarly allow this kind of control, have been previously combined with BO (Snoek et al., 2014) to optimise over functions with non-stationary characteristics, though not providing strong theoretical guarantees. Reinterpreting maps $\mathbf{x} \mapsto \hat{P}_{\mathbf{x}}$ might then allow deriving BO methods with strong theoretical guarantees in problems involving non-stationarity.

Bibliography

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. PMLR, Sydney, Australia.

Mauricio A Alvarez, Lorenzo Rosasco, and Neil D Lawrence. 2011. Kernels for Vector-Valued Functions: a Review. Technical report, MIT - Computer Science and Artificial Intelligence Laboratory.

Anuntapat Anuntachai, Olarn Wongwirat, and Arit Thammano. 2014. An application of ant algorithm for searching energy-efficient route a mobile robot takes using energy as a weighting factor. *Artificial Life and Robotics*.

Dirk V. Arnold and Nikolaus Hansen. 2010. Active covariance matrix adaptation for the (1+1)-CMA-ES. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO '10*. Portland, OR.

N. Aronszajn. 1950. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68(3):337–404.

Raman Arora, Ofer Dekel, and Ambuj Tewari. 2012. Online Bandit Learning against an Adaptive Adversary: from Regret to Policy Regret. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*. Edinburgh, Scotland.

Haim Avron, Vikas Sindhwani, Jiyan Yang, and Michael W. Mahoney. 2016. Quasi-monte carlo feature maps for shift-invariant kernels. *Journal of Machine Learning Research*, 17(120).

Nihat Ay, Jürgen Jost, Hông Vân Lê, and Lorenz Schwachhöfer. 2017. *Information Geometry*, volume 64 of *A Series of Modern Surveys in Mathematics*. Springer.

J Andrew Bagnell and Jeff Schneider. 2003. Policy Search in Kernel Hilbert Space. Technical report, Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA.

H. Bauer. 1981. *Probability theory and elements of measure theory*. Probability and mathematical statistics. Academic Press.

Justin J Beland and Prasanth B Nair. 2017. Bayesian Optimization Under Uncertainty. In *31st Conference on Neural Information Processing Systems (NIPS 2017) Workshop on Bayesian optimization (BayesOpt 2017)*. Long Beach, CA.

Faïz Ben Amar and Philippe Bidaud. 1995. Dynamic Analysis of Off-Road Vehicles. In *4th International Symposium on Experimental Robotics (ISER)*, pages 363–371. Stanford.

Felix Berkenkamp, Angela P. Schoellig, and Andreas Krause. 2019. No-regret Bayesian Optimization with Unknown Hyperparameters. *Journal of Machine Learning Research (JMLR)*, 20:1–24.

Alain Berlinet and Christine Thomas-Agnan. 2004. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Publishers.

Dimitri P. Bertsekas. 2015. *Parallel and Distributed Computation: Numerical Methods*. Athenas Scientific.

Alessandro Bessi and Emilio Ferrara. 2016. Social bots distort the 2016 u.s. presidential election online discussion. *First Monday*, 21(11).

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. 2013. *Concentration inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press.

Eric Brochu, Vlad M Cora, and Nando de Freitas. 2010. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning. Technical report, University of British Columbia.

Chris Brown. 2012. Autonomous vehicle technology in mining. *Engineering and Mining Journal*, 213(1):30–32.

Adam D. Bull. 2011. Convergence Rates of Efficient Global Optimization Algorithms. *Journal of Machine Learning Research (JMLR)*, 12:2879–2904.

Martin V. Butz and Thies D. Lonneker. 2009. Optimized sensory-motor couplings plus strategy extensions for the torcs car racing challenge. In *2009 IEEE Symposium on Computational Intelligence and Games*.

Alberto Candela, David Thompson, Eldar Noe Dobrea, and David Wettergreen. 2017. Planetary Robotic Exploration Driven by Science Hypotheses for Geologic Mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3811–3818. IEEE, Vancouver, BC, Canada.

Joseph Carsten, Arturo Rankin, Dave Ferguson, and Anthony Stentz. 2007. Global path planning on board the mars exploration rovers. In *2007 IEEE Aerospace Conference*, pages 1–11.

K. Chatzilygeroudis, R. Rama, R. Kaushik, D. Goepp, V. Vassiliades, and J. Mouret. 2017. Black-box data-efficient policy search for robotics. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 51–58.

Bo Chen, Rui M. Castro, and Andreas Krause. 2012. Joint Optimization and Variable Selection of High-dimensional Gaussian Processes. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*.

Sayak Ray Chowdhury and Aditya Gopalan. 2017. On Kernelized Multi-armed Bandits. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney, Australia.

Patrick Dallaire, Camille Besse, and Brahim Chaib-Draa. 2011. An approximate inference with Gaussian process to latent functions from uncertain data. *Neurocomputing*, 74:1945–1955.

Andreas C. Damianou, Michalis K. Titsias, and Neil D. Lawrence. 2016. Variational Inference for Latent Variables and Uncertain Inputs in Gaussian Processes. *Journal of Machine Learning Research*, 17(1):1–62.

Marc Peter Deisenroth. 2013. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*.

Olivier Devolder, François Glineur, and Yurii Nesterov. 2010. Solving infinite-dimensional optimization problems by polynomial approximation. *Recent Advances in Optimization and its Applications in Engineering*, pages 31–40.

Nicolae Dinculeanu. 2000. *Vector Integration and Stochastic Integration in Banach Spaces*. John Wiley & Sons.

Josip Djolonga, Andreas Krause, and Volkan Cevher. 2013. High-Dimensional Gaussian Process Bandits. *Advances in Neural Information Processing Systems 26*.

Rick Durrett. 2010. *Probability: theory and examples*. Cambridge University Press, New York, NY, fourth edition.

Nuwan Ganganath, Chi-Tsun Cheng, and Chi K. Tse. 2014. Finding energy-efficient paths on uneven terrains. In *2014 10th France-Japan/ 8th Europe-Asia Congress on Mecatronics (MECATRONICS2014-Tokyo)*, pages 383–388. IEEE.

Arjan Gijsberts and Giorgio Metta. 2013. Real-time model learning using Incremental Sparse Spectrum Gaussian Process Regression. *Neural Networks*, 41:59–69.

Agathe Girard. 2004. *Approximate methods for propagation of uncertainty with Gaussian process models*. Ph. d, University of Glasgow.

Philipp Hennig and Christian J. Schuler. 2012. Entropy Search for Information-Efficient Global Optimization. *Journal of Machine Learning Research (JMLR)*, 13:1809–1837.

James Hensman, Nicolo Fusi, and Neil D. Lawrence. 2013. Gaussian Processes for Big Data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 282–290.

Ken Ho, Thierry Peynot, and Salah Sukkarieh. 2013. Traversability estimation for a planetary rover via experimental kernel learning in a Gaussian process framework. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3475–3482. IEEE, Karlsruhe, Germany.

Roger A. Horn and Charles R. Johnson. 1985. *Matrix Analysis*. Cambridge University Press.

Steven G. Johnson. 2014. The nlopt nonlinear-optimization package. `http://ab-initio.mit.edu/nlopt`. Accessed: 2016-08-16.

Olav Kallenberg. 2017. *Random Measures, Theory and Applications*. Springer.

Kirthevasan Kandasamy, Jeff Schneider, and Barnabas Poczos. 2015. High Dimensional Bayesian Optimisation and Bandits via Additive Models. In *International Conference on Machine Learning (ICML)*.

Muhammad F. Kasim and Peter A. Norreys. 2016. Infinite dimensional optimistic optimisation with applications on physical systems. In *NIPS Workshop on Bayesian Optimization*. Barcelona, Spain.

Chong Hui Kim and Byung Kook Kim. 2007. Minimum-energy translational trajectory generation for differential-driven wheeled mobile Robots. *Journal of Intelligent and Robotic Systems: Theory and Applications*.

Jens Kober, Erhan Oztop, and Jan Peters. 2010. Reinforcement learning to adjust robot movements to new situations. In *Proceedings of Robotics: Science and Systems*. Zaragoza, Spain.

Philippe Komma, Christian Weiss, and Andreas Zell. 2009. Adaptive bayesian filtering for vibration-based terrain classification. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3307–3313. IEEE, Kobe, Japan.

Erwin Kreyszig. 1978. *Introductory functional analysis with applications*. John Wiley & Sons.

Andras Gabor Kupcsik, Marc Peter Deisenroth, Jan Peters, and Gerhard Neumann. 2013. Data-Efficient Generalization of Robot Skills with Contextual Policy Search. In *AAAI Conference on Artificial Intelligence*.

Henry Lam. 2016. Advanced tutorial: input uncertainty and robust analysis in stochastic simulation. In T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, editors, *Proceedings of the 2016 Winter Simulation Conference*, pages 178–192.

Tobias Lang, Christian Plagemann, and Wolfram Burgard. 2007. Adaptive non-stationary kernel regression for terrain modelling. In Wolfram Burgard, Oliver Brock, and Cyrill Stachniss, editors, *Proc. of the Robotics: Science and Systems Conference (RSS)*. Atlanta, GA.

Steven M. LaValle. 2006. *Planning Algorithms*. Cambridge University Press.

Nicholas R J Lawrance and Salah Sukkarieh. 2009. A guidance and control strategy for dynamic soaring with a gliding UAV. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3632–3637.

Peter D. Lax. 2002. *Functional Analysis*. John Wiley & Sons.

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, and Aníbal R. Figueiras-Vidal. 2010. Sparse Spectrum Gaussian Process Regression. *Journal of Machine Learning Research*, 11:1865–1881.

Quoc V Le, Tamas Sarlos, and Alex Smola. 2013. Fastfood âĂŤ Approximating Kernel Expansions in Loglinear Time. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28. JMLR: W&CP, Atlanta, Georgia, USA.

Michel Ledoux and Michel Talagrand. 1991. *Probability in Banach Spaces*. Springer, New York, NY.

Chun-Liang Li, Kirthevasan Kandasamy, Barnabas Poczos, and Jeff Schneider. 2016. High Dimensional Bayesian Optimization via Restricted Projection Pursuit Models. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*.

Daniel Li and Hervé Queffélec. 2017. *Introduction to Banach Spaces: Analysis and Probability*, volume 1 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press.

Chun Kai Ling, Kian Hsiang Low, and Patrick Jaillet. 2016. Gaussian Process Planning with Lipschitz Continuous Reward Functions: Towards Unifying Bayesian Optimization, Active Learning, and Beyond. In *AAAI*.

Alexander Liniger, Alexander Domahidi, and Manfred Morari. 2015. Optimization-based autonomous racing of 1:43 scale RC cars. *Optimal Control Applications and Methods*, 36:628–647.

Lifang Liu, Trever G. Crowe, and Joseph N. Bakambu. 2008. Efficient Exploration Algorithms for Rough Terrain Modeling Using Triangular Mesh Maps. In *IEEE International Conference on Robotics, Automation and Mechatronics*.

Daniel James Lizotte. 2008. *Practical Bayesian Optimization*. Ph. d, University of Alberta.

David J. C. MacKay. 2003. *Information theory, inference, and learning algorithms*. Cambridge University Press, Cambridge, UK.

Takashi Maekawa, Tetsuya Noda, Shigefumi Tamura, Tomonori Ozaki, and Ken-ichiro Machida. 2010. Curvature continuous path generation for autonomous vehicle using B-spline curves. *Computer-Aided Design*, 42(4):350–359.

A. Rupam Mahmood, Dmytro Korenkevych, Gautham Vasan, William Ma, and James Bergstra. 2018. Benchmarking Reinforcement Learning Algorithms on Real-World Robots. In *2nd Conference on Robot Learning (CoRL 2018)*, pages 1–31. Zurich, Switzerland.

Cédric Malherbe and Nicolas Vayatis. 2017. Global optimization of Lipschitz functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 2314–2323. PMLR, International Convention Centre, Sydney, Australia.

Vidyadhar Mandrekar and Barbara Rüdiger. 2015. *Stochastic integration in Banach spaces: Theory and Applications*, volume 73. Springer.

Roman Marchant and Fabio Ramos. 2012. Bayesian Optimisation for Intelligent Environmental Monitoring. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Roman Marchant and Fabio Ramos. 2014. Bayesian Optimisation for Informative Continuous Path Planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6136–6143.

Román Marchant Matus. 2016. *Bayesian Optimisation for Planning in Dynamic Environments*. Ph. d, The University of Sydney.

Zita Marinho, Anca Dragan, Arun Byravan, Byron Boots, Siddhartha Srinivasa, and Geoffrey Gordon. 2016. Functional Gradient Motion Planning in Reproducing Kernel Hilbert Spaces. In *Robotics: Science and Systems (RSS)*.

Steven Martin and Peter Corke. 2014. Long-term exploration & tours for energy constrained robots with online proprioceptive traversability estimation. *IEEE International Conference on Robotics and Automation (ICRA)*.

Steven Martin, Liz Murphy, and Peter Corke. 2013. Building Large Scale Traversability Maps Using Vehicle Experience. In *The 13th International Symposium on Experimental Robotics (ISER)*, volume 88, pages 891–905. Springer.

R. Martinez-Cantin. 2017. Bayesian optimization with adaptive kernels for robot control. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3350–3356.

Ruben Martinez-Cantin, Nando de Freitas, Arnaud Doucet, and José A. Castellanos. 2007. Active Policy Learning for Robot Planning and Exploration under Uncertainty. In *Robotics: Science and Systems (RSS)*. Atlanta, GA.

Rowan Thomas Mcallister and Carl Edward Rasmussen. 2017. Data-Efficient Reinforcement Learning in Continuous State-Action Gaussian-POMDPs. In *31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA.

Colin McDiarmid. 1989. *On the method of bounded differences*, page 148âĂŞ188. London Mathematical Society Lecture Note Series. Cambridge University Press.

Andrew Mchutchon and Carl E. Rasmussen. 2011. Gaussian Process Training with Input Noise. In *Advances in Neural Information Processing Systems*, pages 1341–1349.

Mitchell Mcintire, Daniel Ratner, and Stefano Ermon. 2016. Sparse Gaussian Processes for Bayesian Optimization. In *Uncertainty in Artificial Intelligence (UAI)*.

M. D. McKay, R. J. Beckman, and W. J. Conover. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245.

Charles A. Micchelli and Massimiliano Pontil. 2003. On Learning Vector-Valued Functions. Technical report, Department of Computer Science, University College London.

Jonas Mockus. 1989. *Bayesian approach to global optimization: theory and applications*. Kluwer Academic, Boston;Dordrecht;.

V. Molino, R. Madhavan, E. Messina, A. Downs, S. Balakirsky, and A. Jacoff. 2007. Traversability metrics for rough terrain applied to repeatable test methods. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

T. Moore and D. Stouch. 2014. A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer.

Jesús Morales, Jorge Martínez, Anthony Mandow, Alfonso García-Cerezo, Jesús Gómez-Gabriel, and Salvador Pedraza. 2006. Power Analysis for a Skid-Steered Tracked Mobile Robot. In *IEEE International Conference on Mechatronics*.

Fabio Morbidi, Roel Cano, and David Lara. 2016. Minimum-energy path generation for a quadrotor uav. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1492–1498.

Philippe Morere, Roman Marchant, and Fabio Ramos. 2016. Bayesian Optimisation for solving Continuous State-Action-Observation POMDPs. In *NIPS Workshop on Bayesian Optimization*. Barcelona, Spain.

Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. 2012. Learning from Distributions via Support Measure Machines. In *Proceeding of the 26th Annual Conference on Neural Information Processing Systems (NIPS 2012)*.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. 2016. Kernel Mean Embedding of Distributions: A Review and Beyond. *arXiv*.

Mustafa Mukadam, Jing Dong, Frank Dellaert, and Byron Boots. 2017. Simultaneous Trajectory Estimation and Planning via Probabilistic Inference. In *Proceedings of Robotics: Science and Systems*. Cambridge, Massachusetts.

James Raymond Munkres. 1975. *Topology: a first course*. Prentice Hall, Edgewood Cliffs, NJ.

Kevin P. Murphy. 2012. *Machine Learning A Probabilistic Perspective*. The MIT Press, Cambridge, MA.

Iain Murray and Ryan Prescott Adams. 2010. Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing (NIPS)*.

Mojmír Mutný and Andreas Krause. 2018. Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features. In *Conference on Neural Information Processing Systems (NeurIPS)*. Montreal, Canada.

Hongseok Namkoong, Aman Sinha, Steve Yadlowsky, and John C Duchi. 2017. Adaptive Sampling Probabilities for Non-Smooth Optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2574–2583. Sydney, Australia.

Yurii Nesterov. 1983. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27.

Yurii Nesterov. 2012. Efficiency of Coordinate Descent Methods on Huge-Scale Optimization Problems. *SIAM Journal on Optimization*, 22:341–362.

José Nogueira, Ruben Martinez-Cantin, Alexandre Bernardino, and Lorenzo Jamone. 2016. Unscented Bayesian Optimization for Safe Robot Grasping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1967–1972. Daejeon, Korea.

Peter Nordin. 2012. *Mobile Robot Traversability Mapping*. Licentiate thesis, Linköping University.

Simon T. O'Callaghan and Fabio T. Ramos. 2012. Gaussian process occupancy maps. *The International Journal of Robotics Research (IJRR)*, 31(1):42–62.

Rafael Oliveira, Lionel Ott, Vitor Guizilini, and Fabio Ramos. 2017a. Bayesian Optimisation for Safe Navigation under Localisation Uncertainty. *arXiv:1709.02169*.

Rafael Oliveira, Lionel Ott, Vitor Guizilini, and Fabio Ramos. 2018a. Leveraging Localisation Information into Bayesian Optimisation for Planning in Robotics. In *Workshop on Informative Path Planning and Adaptive Sampling (WIPPAS) at the 2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia.

Rafael Oliveira, Lionel Ott, Vitor Guizlini, and Fabio Ramos. 2017b. Bayesian Optimisation for Safe Navigation under Localisation Uncertainty. In *International Symposium on Robotics Research (ISRR) (to appear)*. Puerto Varas, Chile.

Rafael Oliveira, Lionel Ott, and Fabio Ramos. 2016. Active Perception for Modelling Energy Consumption in Off-Road Navigation. In *Australasian Conference on Robotics and Automation (ACRA)*. Brisbane, Australia.

Rafael Oliveira, Lionel Ott, and Fabio Ramos. 2019. Bayesian optimisation under uncertain inputs. In *International Conference on Artificial Intelligence and Statistics (AISTATS) (to appear)*. Naha, Okinawa, Japan.

Rafael Oliveira, Fernando H M Rocha, Lionel Ott, Vitor Guizilini, Fabio Ramos, and Valdir Grassi. 2018b. Learning to Race through Coordinate Descent Bayesian Optimisation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6431–6438. Brisbane, Australia.

Michael Pearce and Juergen Branke. 2017. Bayesian simulation optimization with input uncertainty. In W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, editors, *Proceedings of the 2017 Winter Simulation Conference*, pages 2268–2278. IEEE.

Arvind A. Pereira, Jonathan Binney, Geoffrey A. Hollinger, and Gaurav S. Sukhatme. 2013. Risk-aware Path Planning for Autonomous Underwater Vehicles using Predictive Ocean Models. *Journal of Field Robotics*, 30(5):741–762.

Ken Perlin. 1985. An image synthesizer. In *Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*.

Christian Plagemann, Sebastian Mischke, Sam Prentice, Kristian Kersting, Nicholas Roy, and Wolfram Burgard. 2008. Learning predictive terrain models for legged robot locomotion. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 3545–3552.

Patrick A. Plonski, Joshua Vander Hook, and Volkan Isler. 2013. Environment and Solar Map Construction for Solar-Powered Mobile Systems. *IEEE Transactions on Robotics*, 32(1):70–82.

Michael J. D. Powell. 1973. On search directions for minimizaiton algorithms. *Mathematical Programming*, 4.

Michael J. D. Powell. 1998. Direct search algorithms for optimization calculations. *Acta Numerica*, 7:287–336.

Michael J. D. Powell. 2007. A view of algorithms for optimization without derivatives. Technical report, Cambridge University DAMTP, Cambridge, United Kingdom.

Ali Rahimi and Ben Recht. 2007. Random features for large-scale kernel machines. In *Advances in Neural Information Processing (NIPS)*.

Carl E. Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, MA.

Carl Edward Rasmussen and Zoubin Ghahramani. 2003. Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems 15*, pages 489–496.

Martin Riedmiller and Heinrich Braun. 1993. A direct adaptive method for faster backpropagation learning: The Rprop algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591. San Francisco, CA, USA.

Tizar Rizano, Daniele Fontanelli, Luigi Palopoli, Lucia Pallottino, and Paolo Salaris. 2013. Global path planning for competitive robotic cars. In *IEEE Conference on Decision and Control*. Florence, Italy.

Paul Rolland, Jonathan Scarlett, Ilija Bogunovic, and Volkan Cevher. 2018. High-dimensional bayesian optimization via additive models with overlapping groups. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 298–307. PMLR, Playa Blanca, Lanzarote, Canary Islands.

H. Rosenbrock. 1960. An automatic method for finding the greatest or least value of a function. *Comput. J.*, 3(x).

Thomas Rückstieß, Frank Sehnke, Tom Schaul, Daan Wierstra, Yi Sun, and Jürgen Schmidhuber. 2010. Exploring parameter space in reinforcement learning. *Paladyn*, 1(1).

Thomas Philip Runarsson and Xin Yao. 2005. Search biases in constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(2):233–243.

S. Salan, E. Drumwright, and K. I. Lin. 2015. Minimum-energy robotic exploration: A formulation and an approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):175–182.

Ahmad El Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. 2017. Deep Reinforcement Learning framework for Autonomous Driving. In *IS&T International Symposium on Electronic Imaging*, pages 70–76. IS&T.

Stéphane Sanchez and Sylvain Cussat-Blanc. 2014. Gene regulated car driving: Using a gene regulatory network to drive a virtual car. *Genetic Programming and Evolvable Machines*, 15(4).

Bernhard Schölkopf and Alexander J. Smola. 2002. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, Cambridge, Mass.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. 2016. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc.

Jasper Snoek, Kevin Swersky, Richard Zemel, and Ryan P. Adams. 2014. Input Warping for Bayesian Optimization of Non-Stationary Functions. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1674–1682. JMLR: W&CP, Beijing, China.

Jefferson R. Souza, Roman Marchant, Lionel Ott, Denis F. Wolf, and Fabio Ramos. 2014. Bayesian Optimisation for Active Perception and Smooth Navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. 2010. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the 27th International Conference on Machine Learning (ICML 2010)*, pages 1015–1022.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. 2012. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5):1–16.

Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R. G. Lanckriet. 2011. Universality, Characteristic Kernels and RKHS Embedding of Measures. *Journal of Machine Learning Research (JMLR)*, 12:2389–2410.

Ingo Steinwart and Andreas Christmann. 2008. *Support Vector Machines*, chapter 4, pages 110–163. Springer New York, New York, NY.

Anthony Stentz. 1994. The D* Algorithm for Real-Time Planning of Optimal Traverses. Technical report, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. 2017. Deeply AggreVaTeD: Differentiable Imitation Learning for Sequential Prediction. In *Proceedings of the 2017 International Conference on Machine Learning (ICML)*.

Zheng Sun and John H. Reif. 2005. On finding energy-minimizing paths on terrains. *IEEE Transactions on Robotics*, 21(1):102–114.

Charles Suquet. 2009. Reproducing kernel Hilbert spaces and random measures. In *More Progresses in Analysis*, pages 143–152. World Scientific.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2006. *Probabilistic Robotics*. The MIT Press, Cambridge, MA.

Michalis Titsias and Neil Lawrence. 2010. Bayesian Gaussian Process Latent Variable Model. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 844–851.

Pratap Tokekar, Nikhil Karnad, and Volkan Isler. 2011. Energy-optimal velocity profiles for car-like robots. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Anthony Tompkins, Ransalu Senanayake, Philippe Morere, and Fabio Ramos. 2019. Black Box Quantiles for Kernel Learning. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Doniyor Ulmasov, Caroline Baroukh, Benoit Chachuat, Marc Peter Deisenroth, and Ruth Misener. 2016. Bayesian Optimization with Dimension Scheduling: Application to Biological Systems. In *Computer Aided Chemical Engineering*, volume 38. Elsevier Masson SAS.

James Underwood, Alexander Wendel, Brooke Schofield, Larn McMurray, and Rohan Kimber. 2017. Efficient in-field plant phenomics for row-crops with an autonomous ground vehicle. *Journal of Field Robotics*.

S. Vasudevan, F. Ramos, E. Nettleton, H. Durrant-Whyte, and A. Blair. 2009. Gaussian Process modeling of large scale terrain. In *2009 IEEE International Conference on Robotics and Automation*, pages 1047–1053. IEEE.

Ngo Anh Vien, Peter Englert, and Marc Toussaint. 2016. Policy Search in Reproducing Kernel Hilbert Space. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*.

Ngo Anh Vien and Marc Toussaint. 2018. Bayesian Functional Optimization. In *AAAI Conference on Artificial Intelligence*, pages 4171–4178. New Orleans, LA, USA.

Grace Wahba. 1990. *Spline Models for Observational Data*. SIAM, Philadelphia, PA.

Eric A. Wan and Rudolph van der Merwe. 2000. The Unscented Kalman Filter for Nonlinear Estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC)*, pages 153–158.

Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. 2018a. Batched Large-scale Bayesian Optimization in High-dimensional Spaces. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018*. JMLR: W&CP, Lanzarote, Spain.

Zi Wang and Stefanie Jegelka. 2017. Max-value Entropy Search for Efficient Bayesian Optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*. Sydney, Australia.

Zi Wang, Beomjoon Kim, and Leslie Kaelbling. 2018b. Regret bounds for meta Bayesian optimization with an unknown Gaussian process prior. In *Thirty-second Conference on Neural Information Processing Systems (NeurIPS)*. Montreal, Canada.

Ziyu Wang and Nando de Freitas. 2014. Theoretical Analysis of Bayesian Optimisation with Unknown Gaussian Process Hyper-Parameters. *arXiv*, pages 1–16.

Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, and Nando De Freitas. 2013. Bayesian Optimization in a Billion Dimensions via Random Embeddings. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI)*.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James Rehg, Byron Boots, and Evangelos Theodorou. 2017. Information theoretic MPC for model-based reinforcement learning. In *Proceedings of the 2017 IEEE Conference on Robotics and Automation (ICRA)*.

Aaron Wilson, Alan Fern, and Prasad Tadepalli. 2014. Using Trajectory Data to Improve Bayesian Optimization for Reinforcement Learning. *Journal of Machine Learning Research*, 15.

Bernhard Wymann, Eric Espié, Christophe Guionneau, Christos Dimitrakakis, Rémi Coulom, and Andrew Sumner. 2014. TORCS, The Open Racing Car Simulator. http://www.torcs.org.

Ying Xiong. 2010. *Racing Line Optimization*. M.sc., Massachusetts Institute of Technology.

Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. 2012. Nystrom Method vs Random Fourier Features: A Theoretical and Empirical Comparison. In *Advances in Neural Information Processing Systems*, pages 485–493.

Y. Yoshikawa, T. Iwata, and H. Sawada. 2015. Non-linear regression for bag-of-words data via Gaussian process latent variable set model. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 4:3129–3135.

M. Zucker, N. Ratliff, a. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa. 2013. CHOMP: Covariant Hamiltonian optimization for motion planning. *The International Journal of Robotics Research (IJRR)*, 32(9-10):1164–1193.

Appendix A

Proofs

This chapter presents proofs for auxiliary theoretical results in Chapter 6. The first section provides some background with general definitions and basic lemmas applied in the proofs. The following sections then present the proofs for each result, starting with Proposition 6.6. For reference, a notation summary is presented in Table A.1.

Table A.1: Notation

$\mathbb{R}$	the field of real numbers, or the real line
$\mathbb{R}^d$	the Euclidean space of d-dimensional vectors
$\mathcal{S}$	BO's search space, a subset of $\mathbb{R}^d$
$\mathbf{x}$	a location vector, considered as an element of $\mathbb{R}^d$
$\tilde{\mathbf{x}}$	random location, an $\mathbb{R}^d$-valued random variable
f	deterministic-inputs objective function
$\hat{f}$	uncertain-inputs objective function
$\mathcal{X}$	input domain of the objective function, often $\mathbb{R}^d$
$\mathcal{P}$	set of all probability measures on $\mathcal{X}$
P	a probability measure or distribution
P_t^L	location distribution estimated by localisation
$P_{\mathbf{x}}^E$	query location distribution given a target $\mathbf{x}$
$\hat{P}_{\mathbf{x}}$	model for $P_{\mathbf{x}}^E$ used by uGP-UCB
ψ_P	mean map of P
k	a positive-definite kernel
$\mathcal{H}_k$	reproducing kernel Hilbert space of k

A.1 Background

Definition A.1 (Bounded differences property). *Let* $\mathbf{x} = [x_1, \dots, x_d]^\mathsf{T}$ *and:*

$$\mathbf{x}'_i = [x_1, \dots, x_{i-1}, x'_i, x_{i+1}, \dots, x_d]^\mathsf{T} , \tag{A.1}$$

where $x_i, x_i' \in \mathcal{X}_i \subset \mathbb{R}$ and $\mathcal{X} = (\mathcal{X}_1 \times \cdots \times \mathcal{X}_d)$. A function $f : \mathcal{X} \to \mathbb{R}$ has the bounded differences property if:

$$|f(\mathbf{x}) - f(\mathbf{x}_i')| \leq c_i,\ \forall i \in \{1, \dots, d\}\ , \tag{A.2}$$

where c_i are non-negative constants.

LEMMA A.2 (Corollary 4.36 in Steinwart and Christmann (2008)). *Let $f \in \mathcal{H}_k$, where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a twice-differentiable kernel on $\mathcal{X} \subseteq \mathbb{R}^d$. Then f has bounded first-order partial derivatives, such that for any $\mathbf{x} \in \mathcal{X}$:*

$$\left|\frac{\partial f(\mathbf{x})}{\partial x_i}\right| \leq \|f\|_k \sqrt{\frac{\partial^2 k(\mathbf{x}, \mathbf{x}')}{\partial x_i \partial x_i'}\Big|_{\mathbf{x}'=\mathbf{x}}}\ . \tag{A.3}$$

LEMMA A.3 (Theorem 5.5 in Boucheron et al. (2013)). *Let $\tilde{\mathbf{x}} \sim N(\mathbf{0}, \mathbf{I})$ be an $\mathbb{R}^d$-valued standard Gaussian random vector. Let $f : \mathbb{R}^d \to \mathbb{R}$ denote a ℓ-Lipschitz function, i.e.:*

$$|f(\mathbf{x}) - f(\mathbf{x}')| \leq \ell \|\mathbf{x} - \mathbf{x}'\|_2,\ \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d\ . \tag{A.4}$$

Then, for all $\lambda \in \mathbb{R}$:

$$\mathbb{E}[e^{\lambda(f(\tilde{\mathbf{x}}) - \mathbb{E}[f(\tilde{\mathbf{x}})])}] \leq e^{\frac{1}{2}\lambda^2\ell^2}\ . \tag{A.5}$$

DEFINITION A.4 (Bounded linear operator). *A linear operator $\mathbf{L} : \mathcal{X} \to \mathcal{Y}$ mapping a vector space $\mathcal{X}$ to another vector space $\mathcal{Y}$ is any operator such that, for all $x, y \in \mathcal{X}$ and a scalar α:*

A1. $\mathbf{L}(x + y) = \mathbf{L}x + \mathbf{L}y$

A2. $\mathbf{L}(\alpha x) = \alpha \mathbf{L}x$

If $\mathcal{X}$ and $\mathcal{Y}$ are normed vector spaces, the operator $\mathbf{L}$ is bounded if:

$$\|\mathbf{L}\| := \sup_{x \in \mathcal{X} : \|x\| = 1} \|\mathbf{L}x\| < \infty\ , \tag{A.6}$$

where $\|\mathbf{L}\|$ is called the operator norm of $\mathbf{L}$.

LEMMA A.5 (Bounded linear extension (Kreyszig, 1978, Thr. 2.7-11)). *Let $\mathbf{M} : \mathcal{W} \to \mathcal{Y}$ be a bounded linear operator, where $\mathcal{W}$ lies in a normed vector space $\mathcal{X}$, and $\mathcal{Y}$ is a Banach space. Then $\mathbf{M}$ has an extension $\overline{\mathbf{M}} : \overline{\mathcal{W}} \to \mathcal{Y}$, where $\overline{\mathbf{M}}$ is a bounded linear operator with norm $\|\overline{\mathbf{M}}\| = \|\mathbf{M}\|$.*

A.2 Proof of Lemma 6.1

LEMMA 6.1 (Expected function). *Any $f \in \mathcal{H}_k$ is continuously mapped to a corresponding $\hat{f} \in \mathcal{H}_{\hat{k}}$, which is such that:*

$$\begin{aligned} \forall P \in \mathcal{P}, \quad \hat{f}(P) &= \mathbb{E}_P[f] \\ \|\hat{f}\|_{\hat{k}} &= \|f\|_k \ . \end{aligned} \tag{6.8}$$

Under the additional assumption that k is characteristic, we have that the mapping $f \mapsto \hat{f}$ constitutes an isometry between $\mathcal{H}_k$ and $\mathcal{H}_{\hat{k}}$ and is also invertible.

PROOF. By the definition in Equation 6.7, we have the following equivalence:

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \quad k(\mathbf{x}, \mathbf{x}') = \hat{k}(D_{\mathbf{x}}, D_{\mathbf{x}'}) \ , \tag{A.7}$$

where $D_{\mathbf{x}} \in \mathcal{P}$ is the Dirac measure associated with $\mathbf{x}$ (see Equation 2.74). Therefore, the linear map $k(\cdot, \mathbf{x}) \mapsto \hat{k}(\cdot, D_{\mathbf{x}})$ is bounded. As any $f^0 \in \mathcal{H}_k^0$ is composed of finite linear combinations of features $k(\cdot, \mathbf{x}_i)$, we then have that f^0 is continuously mapped to a $\hat{f}^0 \in \mathcal{H}_{\hat{k}}$, given by the equivalence:

$$f^0 = \sum_{i=1}^{q} \alpha_i k(\cdot, \mathbf{x}_i) = \sum_{i=1}^{q} \alpha_i \hat{k}\left(D_{(\cdot)}, D_{\mathbf{x}_i}\right) = \hat{f}^0\left(D_{(\cdot)}\right) \ , \tag{A.8}$$

where $D_{(\cdot)}$ represents the map $\mathbf{x} \mapsto D_{\mathbf{x}}$. Given $P \in \mathcal{P}$, we can see that:

$$\hat{f}^0(P) = \sum_{i=1}^{q} \alpha_i \hat{k}\left(P, D_{\mathbf{x}_i}\right) = \sum_{i=1}^{q} \alpha_i \langle k(\cdot, \mathbf{x}_i), \psi_P \rangle_k = \mathbb{E}_P[f] \ . \tag{A.9}$$

Furthermore, the RKHS norm of $\hat{f}^0$ is such that:

$$\|\hat{f}^0\|_{\hat{k}}^2 = \sum_{i=1}^{q} \sum_{j=1}^{q} \alpha_i \alpha_j \hat{k}\left(D_{\mathbf{x}_i}, D_{\mathbf{x}_j}\right) = \sum_{i=1}^{q} \sum_{j=1}^{q} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) = \|f^0\|_k^2 \ . \tag{A.10}$$

Therefore, the mapping from $\mathcal{H}_k^0$ to $\mathcal{H}_{\hat{k}}$ defined by $f^0 \mapsto \hat{f}^0$ is bounded and isometric. Since $\mathcal{H}_k^0$ is dense in $\mathcal{H}_k$, $f^0 \mapsto \hat{f}^0$ can be continuously extended to the closure $\overline{\mathcal{H}_k^0}$, which is the entire $\mathcal{H}_k$, by the bounded linear extension theorem (Kreyszig, 1978, Thr. 2.7-11).

Conversely, consider any $\hat{f}^0 = \sum_{i=1}^{q} \alpha_i \hat{k}(\cdot, P_i) \in \mathcal{H}_{\hat{k}}^0$, which is the pre-Hilbert space spanned by $\hat{k}$. If k is a characteristic kernel (Sriperumbudur et al., 2011), the mean map ψ is injective, i.e. one-to-one. Therefore, every $\psi_P \in \mathcal{H}_k$ is associated with a unique $P \in \mathcal{P}$. By construction, we have that

$\hat{k}(\cdot, P) = \langle\psi_{(\cdot)}, \psi_P\rangle_k$. With an injective ψ, we can then define a bounded linear map from $\mathcal{H}^0_{\hat{k}}$ to $\mathcal{H}_k$ using $\hat{k}(\cdot, P) \mapsto \psi_P$. The rest of the proof follows by repeating the steps applied in the first part. □

A.3 Proof of Proposition 6.6

This section presents the proof for Proposition 6.6. The result is restated below, followed by the proof.

PROPOSITION 6.6. *Let $f \in \mathcal{H}_k$ and $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be an at least twice-differentiable positive-definite kernel with finite $\ell_k^2 \geq \sup_{\mathbf{x}\in\mathbb{R}^d} \sup_{i\in\{1,\ldots,d\}} \frac{\partial^2 k(\mathbf{x},\mathbf{x}')}{\partial x_i \partial x'_i}|_{\mathbf{x}=\mathbf{x}'}$. Then, for a given $P \in \mathcal{P}$ and $\tilde{\mathbf{x}} \sim P$, we have that $\Delta f_P := f(\tilde{\mathbf{x}}) - \mathbb{E}_P[f(\tilde{\mathbf{x}})]$ is a σ_F-sub-Gaussian random variable in the following cases:*

(1) $\sigma_F = \|f\|_k \ell_k \mathrm{tr}(\mathbf{\Sigma})^{1/2}$, if P is Gaussian with covariance matrix $\mathbf{\Sigma}$;

(2) $\sigma_F = \frac{1}{2}\|f\|_k \ell_k \sqrt{\sum_{i=1}^d \sigma_i^2}$, if P has compact support, with $|\tilde{x}_i - \hat{x}_i| \leq \frac{1}{2}\sigma_i$, $i \in \{1, \ldots, d\}$, where $\tilde{\mathbf{x}} \sim P$ and $\hat{\mathbf{x}} = \mathbb{E}[\tilde{\mathbf{x}}]$.

PROOF. The following proof is split in two parts. The derivation firstly covers the case where the inputs follow a Gaussian distribution and then the case for arbitrary probability distributions with compact support.

Gaussian inputs (1): In the case of Gaussian inputs, Proposition 6.6 is a direct consequence of Lemma A.3 when applied to functions $f \in \mathcal{H}_k$. Notice that, by the definition of $\mathcal{H}_k$, any f in it is continuously differentiable and Lipschitz continuous according to Lemma A.2. All we have to do is to generalise the inequality in Equation A.5 for the case of general Gaussian random vectors $\tilde{\mathbf{x}} \sim N(\hat{\mathbf{x}}, \mathbf{\Sigma})$.

If $\tilde{\mathbf{x}}^s$ is a standard Gaussian random vector, $\tilde{\mathbf{x}} = \hat{\mathbf{x}} + \mathbf{A}\tilde{\mathbf{x}}^s$, where $\mathbf{\Sigma} = \mathbf{A}\mathbf{A}^\mathsf{T}$, due to the translational and rotational invariance of Gaussian random vectors. We can define a function g, such that:

$$g(\tilde{\mathbf{x}}^s) = f(\hat{\mathbf{x}} + \mathbf{A}\tilde{\mathbf{x}}^s) = f(\tilde{\mathbf{x}}) \,. \tag{A.11}$$

Since f is Lipschitz continuous, g also is, for some Lipschitz constant ℓ_g. Then we can apply Lemma A.3 to g, which yields:

$$\mathbb{E}[e^{\lambda(g(\tilde{\mathbf{x}}^s) - \mathbb{E}[g(\tilde{\mathbf{x}}^s)])}] \leq e^{\frac{1}{2}\lambda^2\ell_g^2} \,. \tag{A.12}$$

In addition, by definition (Equation A.11), $g(\tilde{\mathbf{x}}^s)$ and $f(\tilde{\mathbf{x}})$ follow the same distribution, so that $\mathbb{E}[f(\tilde{\mathbf{x}})] = \mathbb{E}[g(\tilde{\mathbf{x}}^s)]$. As a result, $\Delta f_P = f(\tilde{\mathbf{x}}) - \mathbb{E}[f(\tilde{\mathbf{x}})]$ is ℓ_g-sub-Gaussian, according to Definition 6.2.

Now ℓ_g is any constant uniformly upper-bounding the Euclidean norm of g's gradient, and $\|\nabla g\|_2 = \|\mathbf{A}^\mathsf{T}\nabla f\|_2$. Without loss of generality, let's assume that $\boldsymbol{\Sigma}$ is a matrix of diagonal entries $\sigma_i^2, 1 \leq i \leq d$. Then we have that:

$$\begin{aligned} \|\nabla g\|_2^2 &= \|\mathbf{A}^\mathsf{T}\nabla f\|_2^2 \\ &= \nabla f^\mathsf{T}\mathbf{A}\mathbf{A}^\mathsf{T}\nabla f \\ &= \nabla f^\mathsf{T}\boldsymbol{\Sigma}\nabla f \\ &= \sum_{i=1}^{d}\sigma_i^2\left|\frac{\partial f}{\partial x_i}\right|^2 \leq \ell_f^2 \mathrm{tr}(\boldsymbol{\Sigma}) \, , \end{aligned} \tag{A.13}$$

where $\ell_f = \|f\|_k \ell_k$. Therefore, the inequality in Equation A.12 holds with $\ell_g = \ell_f\sqrt{(\mathrm{tr}(\boldsymbol{\Sigma}))}$.

For a non-diagonal $\boldsymbol{\Sigma}$, by spectral decomposition, we have that $\boldsymbol{\Sigma} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\mathsf{T}$, where $\boldsymbol{\Lambda}$ is a matrix whose diagonal is composed of $\boldsymbol{\Sigma}$'s eigenvalues and $\mathbf{V}\mathbf{V}^\mathsf{T} = \mathbf{I}$. Observe that the result in Equation A.13 would also hold for a zero-mean Gaussian random vector $\tilde{\mathbf{x}}^v$ with covariance matrix $\boldsymbol{\Lambda}$. Then we could define $h(\tilde{\mathbf{x}}^v) = f(\hat{\mathbf{x}} + \mathbf{V}\tilde{\mathbf{x}}^v)$ and follow similar steps to the ones we took for g. However, f and h, as defined, have the same Lipschitz constant, since:

$$\|\mathbf{V}\mathbf{x} - \mathbf{V}\mathbf{X}'\|_2^2 = (\mathbf{x} - \mathbf{x}')^\mathsf{T}\mathbf{V}^\mathsf{T}\mathbf{V}(\mathbf{x} - \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2^2 \, , \tag{A.14}$$

where we applied $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{V}\mathbf{V}^\mathsf{T} = \mathbf{I}$. In addition, as $\boldsymbol{\Sigma}$ is positive definite, $\mathrm{tr}(\boldsymbol{\Sigma}) = \mathrm{tr}(\boldsymbol{\Lambda})$. Therefore, the same result in Equation A.13 holds for general $\boldsymbol{\Sigma}$ and $\hat{\mathbf{x}}$, which can also be seen as a consequence of the translational and rotational invariance of Gaussian random vectors. Making $\sigma_F = \ell_g = \|f\|_k \ell_k \mathrm{tr}(\boldsymbol{\Sigma})^{1/2}$ concludes the first part of the proof.

Distributions with compact support (2). By Lemma A.2, we can observe that $f \in \mathcal{H}_k(\mathcal{X})$ is Lipschitz continuous with respect to the 1-norm on $\mathbb{R}^d$, in particular:

$$|f(\mathbf{x}) - f(\mathbf{x}')| \leq \|f\|_k \ell_k \|\mathbf{x} - \mathbf{x}'\|_1, \; \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d \, , \tag{A.15}$$

where $\ell_k \geq 0$ is any constant such that $\ell_k^2 \geq \sup_{\mathbf{x}\in\mathcal{X}} \sup_{i\in[d]} \frac{\partial^2 k(\mathbf{x},\mathbf{x}')}{\partial x_i \partial x_i'}|_{\mathbf{x}=\mathbf{x}'}$. Therefore, according to Definition A.1, f satisfies the bounded differences property for any $\mathbf{x}$ in the support of P with $c_i = \|f\|_k \ell_k \sigma_i$. Applying McDiarmid's inequality (McDiarmid, 1989), we have that:

$$\mathbb{P}\left\{|f(\tilde{\mathbf{x}}) - \mathbb{E}_P(f)| \geq t\right\} \leq 2\exp\left(-\frac{2t^2}{\|f\|_k^2 \ell_k^2 \sum_{i=1}^d \sigma_i^2}\right) \, . \tag{A.16}$$

As a result, Δf_P is σ_F-sub-Gaussian with $\sigma_F = \frac{1}{2}\|f\|_k \ell_k \sqrt{\sum_{i=1}^d \sigma_i^2}$, according to Definition 6.2 and Lemma 2.2 in Boucheron et al. (2013). This concludes the proof. □

A.4 Proof of Proposition 6.7

PROPOSITION 6.7. *Let k be a bounded, measurable, translation-invariant kernel on $(\mathcal{X}, \mathfrak{X})$, where $\mathcal{X}$ is a topological vector space, and $\mathfrak{X}$ is its Borel σ-algebra. Assume that the mapping $\mathbf{x} \mapsto P_{\mathbf{x}} \in \mathcal{P}(\mathcal{X})$ is such that, for any $\mathbf{x} \in \mathcal{S} \subset \mathcal{X}$, $\tilde{\mathbf{x}} \sim P_{\mathbf{x}}$ is decomposable as $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is independent and identically distributed, i.e. $\boldsymbol{\epsilon} \sim P_E \in \mathcal{P}(\mathcal{X})$. Then we have that, for any $f \in \mathcal{H}_k$, the mapping $\mathbf{x} \mapsto \mathbb{E}_{P_{\mathbf{x}}}[f]$ defines a function $g \in \mathcal{H}_k(\mathcal{S})$, and $\|g\|_k \leq \|f\|_k$.*

PROOF. To prove this result, we will consider properties of the inner product in $\mathcal{H}_k$ when k is translation invariant. These properties essentially allow us to transfer the noise in the evaluation of f to f itself and then represent g as the expectation of this noisy version of f. Similar to the proof of Lemma 6.1, we start by defining an operator on $\mathcal{H}_k^0$ and then extend it to $\mathcal{H}_k$.

To develop the proof, we need to represent f in terms of the kernel k. Let $f = \sum_{i=1}^q \alpha_i k(\cdot, \mathbf{x}_i) \in \mathcal{H}_k^0$, where $\mathcal{H}_k^0$ is the pre-Hilbert space spanned by the kernel (see Section 2.4.2). Considering the evaluation of the expected value of f, we have that:

$$\forall \mathbf{x} \in \mathcal{S},\ \mathbb{E}_{P_{\mathbf{x}}}[f] = \mathbb{E}_{\boldsymbol{\epsilon} \sim P_E}[f(\mathbf{x} + \boldsymbol{\epsilon})] = \mathbb{E}_{\boldsymbol{\epsilon} \sim P_E}\left[\sum_{i=1}^q \alpha_i k(\mathbf{x} + \boldsymbol{\epsilon}, \mathbf{x}_i)\right] . \tag{A.17}$$

For a fixed $\boldsymbol{\epsilon}$, we have that $k(\mathbf{x}', \mathbf{x} + \boldsymbol{\epsilon}) = k(\mathbf{x}' - \boldsymbol{\epsilon}, \mathbf{x})$, $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}$, by translation invariance. Applying this property, we obtain:

$$\begin{aligned} f(\mathbf{x} + \boldsymbol{\epsilon}) = \sum_{i=1}^q \alpha_i k(\mathbf{x} + \boldsymbol{\epsilon}, \mathbf{x}_i) &= \sum_{i=1}^q \alpha_i k(\mathbf{x}, \mathbf{x}_i - \boldsymbol{\epsilon}) \\ &= \langle \sum_{i=1}^q \alpha_i k(\cdot, \mathbf{x}_i - \boldsymbol{\epsilon})\ ,\ k(\cdot, \mathbf{x}) \rangle_k \\ &= f^{\boldsymbol{\epsilon}}(\mathbf{x}) \end{aligned} \tag{A.18}$$

where $f^{\boldsymbol{\epsilon}} := \sum_{i=1}^{q} \alpha_i k(\cdot, \mathbf{x}_i - \boldsymbol{\epsilon})$ is equivalent to a version of f with inputs shifted by $\boldsymbol{\epsilon}$. As the shift $\boldsymbol{\epsilon}$ is the same for all $\mathbf{x}_i,\ i \in \{1, \dots, q\}$, the norm is unaffected, as:

$$\begin{aligned} \|f^{\boldsymbol{\epsilon}}\|_k^2 = \langle f^{\boldsymbol{\epsilon}}, f^{\boldsymbol{\epsilon}} \rangle_k &= \sum_{i=1}^{q} \sum_{j=1}^{q} \alpha_i \alpha_j k(\mathbf{x}_i - \boldsymbol{\epsilon}, \mathbf{x}_j - \boldsymbol{\epsilon}) \\ &= \sum_{i=1}^{q} \sum_{j=1}^{q} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ &= \langle f, f \rangle_k = \|f\|_k^2 \ , \end{aligned} \tag{A.19}$$

where the second equality follows by translational invariance. Defining the mapping $f \mapsto f^{\boldsymbol{\epsilon}}$ as an operator from $\mathcal{H}_k^0$ to $\mathcal{H}_k$, one can easily show that this operator is also linear. Applying the bounded linear extension theorem (Kreyszig, 1978, Thr. 2.7-11), then we have that $f \mapsto f^{\boldsymbol{\epsilon}}$ is actually well defined over the entire $\overline{\mathcal{H}_k^0} = \mathcal{H}_k$.

Now we can return to the derivation in Equation A.17. Since k is measurable, we have that $\boldsymbol{\epsilon} \mapsto f^{\boldsymbol{\epsilon}}$ defines a $\mathcal{H}_k$-valued random variable (see Section 2.6 and Berlinet and Thomas-Agnan (2004, Ch. 4, Sec. 2)). In addition, since $\|f\|_k$ is finite, $\boldsymbol{\epsilon} \mapsto \|f^{\boldsymbol{\epsilon}}\|_k$ is bounded, so that expectations are well defined (Berlinet and Thomas-Agnan, 2004, Ch. 4, Sec. 5). Applying these results to Equation A.17 yields:

$$\forall f \in \mathcal{H}_k, \forall \mathbf{x} \in \mathcal{S},\ \mathbb{E}_{P_{\mathbf{x}}}[f] = \mathbb{E}_{\boldsymbol{\epsilon} \sim P_E}[f^{\boldsymbol{\epsilon}}(\mathbf{x})] = \langle \mathbb{E}_{\boldsymbol{\epsilon} \sim P_E}[f^{\boldsymbol{\epsilon}}], k(\cdot, \mathbf{x}) \rangle_k \ . \tag{A.20}$$

Defining $g' := \mathbb{E}_{\boldsymbol{\epsilon} \sim P_E}[f^{\boldsymbol{\epsilon}}]$ and restricting the domain to $\mathcal{S}$, set $g := g'|_{\mathcal{S}} \in \mathcal{H}_k(\mathcal{S})$. By properties of the Bochner integral (see Equation 2.93), which defines $\mathbb{E}[f^{\boldsymbol{\epsilon}}]$, we know that:

$$\|g'\|_k = \|\mathbb{E}[f^{\boldsymbol{\epsilon}}]\|_k \leq \mathbb{E}[\|f^{\boldsymbol{\epsilon}}\|_k] = \|f\|_k \ . \tag{A.21}$$

Regarding the norm of the domain-restricted function, we then have that:

$$\|g\|_{\mathcal{H}_k(\mathcal{S})} = \inf_{h \in \mathcal{H}_k : h|_{\mathcal{S}} = g} \|h\|_k \leq \|g'\|_k \leq \|f\|_k \ . \tag{A.22}$$

The result in Proposition 6.7 immediately follows, which concludes the proof. □

A.5 Proof of Theorem 6.12

This section presents the continuation of the proof of Theorem 6.12. The first part of the proof proved the following lemma.

LEMMA A.6. *For a fixed $u : \mathcal{W} \to \mathbb{R}$ and $s := \{s_i\}_{i=1}^{\infty}$, $s_i \in \mathbb{R}$, we have that $\{m_t^{u,s}\}_{t=0}^{\infty}$ is a non-negative super-martingale with respect to the filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$, where:*

$$m_t^{u,s} := \exp\left(\boldsymbol{\nu}_t^\mathsf{T}\mathbf{u}_{t,s} - \frac{\sigma_\nu^2}{2}\|\mathbf{u}_{t,s}\|_2^2\right) , \quad t \geq 0 . \tag{A.23}$$

Based on Lemma A.6, the derivations presented here will follow the proof of the IGP-UCB version of Theorem 6.12 Chowdhury and Gopalan (2017, App. A), which the uGP-UCB result is based on. Besides Theorem 6.12, the proof makes use of the following theoretical results. For the definitions, consider an arbitrary probability space $(\mathcal{E}, \mathfrak{E}, P_\mathcal{E})$.

DEFINITION A.7 (Stopping time (Bauer, 1981, Def. 11.2.1)). *A stopping time relative to a filtration $\{\mathfrak{F}_t\}_{t\in\mathcal{I}}$ in $\mathcal{E}$ is a mapping $\tau : \mathcal{E} \to \mathcal{I}$, where $\mathcal{I}$ is a partially ordered index set, such that:*

$$\{\omega \in \mathcal{E} \mid \tau(\omega) \leq t\} \in \mathfrak{F}_t, \quad \forall t \in \mathcal{I} . \tag{A.24}$$

Given a stopping time $\tau : \mathcal{E} \to \mathcal{I}$, denote $\{\tau \leq t\} := \{\omega \in \mathcal{E} \mid \tau(\omega) \leq t\}$ for any $t \in \mathcal{I}$. Considering a filtration $\{\mathfrak{F}_t\}_{t\in\mathcal{I}}$ in $\mathcal{E}$, define:

$$\mathfrak{F}_\tau := \{\mathcal{A} \in 2^\mathcal{E} \mid \mathcal{A} \cap \{\tau \leq t\} \in \mathfrak{F}_t, \forall t \in \mathcal{I}\} . \tag{A.25}$$

From this definition, $\mathfrak{F}_\tau$ constitutes a σ-algebra in $\mathcal{E}$ (Bauer, 1981, p. 346). Namely, $\mathfrak{F}_\tau$ is called the σ-algebra of the τ-past events, i.e. the events up to time τ. Then, given a family of random variables $\{m_t\}_{t\in\mathcal{I}}$, one can define the map $m_\tau : \omega \mapsto m_{\tau(\omega)}(\omega)$, $\omega \in \mathcal{E}$. We have that m_τ defines a random variable if $\mathcal{I}$ is a countable set, such as $\mathbb{N}$, by the following lemma.

LEMMA A.8 (Bauer (1981, Lem. 11.2.3)). *Given a stopping time $\tau : \mathcal{E} \to \mathcal{I}$ on $(\mathcal{E}, \mathfrak{E}, P_\mathcal{E})$, the function $m_\tau : \omega \mapsto m_{\tau(\omega)}(\omega)$ is $\mathfrak{F}_\tau$-measurable if the set of all values t' of τ with $t' \leq t$ is countable for every $t \in \mathcal{I}$.*

Now we can define a stopped version of a super-martingale, which by itself is also a super-martingale, according to the next result.

LEMMA A.9. *Let $\tau : \mathcal{E} \to \mathbb{N}$ be a stopping time relative to the filtration $\{\mathfrak{F}_t\}_{t\in\mathbb{N}}$. If $\{m_t\}_{t\in\mathbb{N}}$ is a super-martingale relative to the filtration $\{\mathfrak{F}_t\}_{t\in\mathbb{N}}$ in $\mathcal{E}$, we have that $\{m_{\min\{\tau,t\}}\}_{t\in\mathbb{N}}$ is a super-martingale.*

PROOF. From the definition a super-martingale (see Section 2.6.4), all we need to do to prove that $\{m_{\min\{\tau,t\}}\}_{t\in\mathbb{N}}$ is a super-martingale is to verify that $\mathbb{E}[|m_{\min\{\tau,t\}}|] < \infty$ and that $\mathbb{E}[m_{\min\{\tau,t\}}|\mathfrak{F}_{t-1}] \leq$

$m_{\min\{\tau,t-1\}}$ (a.s.), for all $t \in \mathbb{N}$. Firstly, we have that:

$$\forall t \in \mathbb{N}, \quad \mathbb{E}[|m_{\min\{\tau,t\}}|] \leq \sum_{i=1}^{t} \mathbb{E}[|m_i|] < \infty \; . \tag{A.26}$$

Secondly, using indicator functions:

$$\mathbb{1}_{\mathcal{A}}(\omega) = \begin{cases} 1, & \omega \in \mathcal{A} \\ 0, & \omega \notin \mathcal{A} \; , \end{cases} \tag{A.27}$$

where $\mathcal{A} \in \mathfrak{E}$, $\omega \in \mathcal{E}$, we have that:

$$m_{\min\{\tau,t\}} = m_t \mathbb{1}_{\{\tau \geq t\}} + \sum_{i=1}^{t-1} m_i \mathbb{1}_{\{\tau = i\}} \tag{A.28}$$

for any $t \in \mathbb{N}$. Notice that, $\{\tau \geq t\} = \{\tau > t-1\} = \mathcal{E} \backslash \{\tau \leq t-1\} \in \mathfrak{F}_{t-1}$ by Definition A.7, so that $\mathbb{1}_{\{\tau \geq t\}}$ is $\mathfrak{F}_{t-1}$-measurable. Now taking the conditional expectation, we obtain:

$$\mathbb{E}[m_{\min\{\tau,t\}} | \mathfrak{F}_{t-1}] = \mathbb{1}_{\{\tau \geq t\}} \mathbb{E}[m_t | \mathfrak{F}_{t-1}] + \sum_{i=1}^{t-1} m_i \mathbb{1}_{\{\tau = i\}} \; . \tag{A.29}$$

Using Equation A.28 for $m_{\min\{\tau,t-1\}}$ and taking the difference with the last result yields:

$$\forall t > 1, \quad m_{\min\{\tau,t-1\}} - \mathbb{E}[m_{\min\{\tau,t\}} | \mathfrak{F}_{t-1}] = \mathbb{1}_{\{\tau \geq t\}} (m_{t-1} - \mathbb{E}[m_t | \mathfrak{F}_{t-1}]) \geq 0 \qquad \text{(a.s.)} \; , \tag{A.30}$$

since $\mathbb{E}[m_t | \mathfrak{F}_{t-1}] \leq m_{t-1}$ (a.s.). The latter implies that $\mathbb{E}[m_{\min\{\tau,t\}} | \mathfrak{F}_{t-1}] \leq m_{\min\{\tau,t-1\}}$ (a.s.), for all $t \in \mathbb{N}$. The result in Lemma A.9 then follows. □

Lastly, the proof of Theorem 6.12 will also make use of the following two lemmas:

LEMMA A.10 (Fatou's lemma (Bauer, 1981, Lem. 2.7.1)). *For every sequence $\{m_t\}_{t \in \mathbb{N}}$ of non-negative $\mathfrak{E}$-measurable functions:*

$$\int \liminf_{t \to \infty} m_t \, \mathrm{d}P_{\mathcal{E}} \leq \liminf_{t \to \infty} \int m_t \, \mathrm{d}P_{\mathcal{E}} \; . \tag{A.31}$$

LEMMA A.11 (Super-martingale convergence (Bauer, 1981, Thr. 11.4.1)). *Every super-martingale $\{m_t\}_{t \in \mathbb{N}}$ relative to $\{\mathfrak{F}_t\}_{t \in \mathbb{N}}$ which satisfies the condition:*

$$\sup_{t \in \mathbb{N}} \mathbb{E}[|m_t|] < \infty \tag{A.32}$$

converges almost surely to an integrable random variable m_∞.

LEMMA A.12 (Markov's inequality). *Given a real-valued random variable ξ and a real number $\alpha > 0$, we have:*

$$\mathbb{P}\left\{|\xi| \geq \alpha\right\} \leq \frac{\mathbb{E}[|\xi|]}{\alpha}\,, \tag{A.33}$$

which is known as Markov's inequality (Boucheron et al., 2013).

Now we are ready to continue with the proof of Theorem 6.12. The result is restated below.

THEOREM 6.12. *Consider an arbitrary domain $\mathcal{W}$ and a σ-algebra $\mathfrak{W}$ on it. Let $\{\xi_t\}_{t=1}^{\infty}$ be a $\mathcal{W}$-valued discrete-time stochastic process adapted to the filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$. Let $\{\nu_t\}_{t=1}^{\infty}$ be a real-valued stochastic process such that, for some $\sigma_\nu \geq 0$ and for all $t \geq 1$, ν_t is (a) $\mathfrak{F}_t$-measurable, and (b) σ_ν-sub-Gaussian conditionally on $\mathfrak{F}_{t-1}$. Let $\hat{k} : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$ be a $\mathfrak{W}$-measurable, bounded positive-definite kernel, and let $0 < \delta \leq 1$. For a given $\eta > 0$, with probability at least $1 - \delta$, the following holds simultaneously for all $t \geq 0$:*

$$\|\boldsymbol{\nu}_t\|^2_{((\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1}+\mathbf{I})^{-1}} \leq 2\sigma_\nu^2 \log \frac{\sqrt{|(1+\eta)\mathbf{I} + \hat{\mathbf{K}}_t|}}{\delta}\,, \tag{6.39}$$

where $\|\mathbf{x}\|^2_{\mathbf{M}} := \mathbf{x}^{\mathsf{T}}\mathbf{M}\mathbf{x}$, for a positive-definite matrix $\mathbf{M}$. We also have that, if $\hat{\mathbf{K}}_t$ is almost-surely positive-definite for all $t \geq 1$, the above result holds for $\eta = 0$.

PROOF OF THEOREM 6.12 (CONT.) Except for a few adaptations, the following steps of the proof follow almost verbatim with the proof in the supplementary material of Chowdhury and Gopalan (2017, App. A). Considering Lemma A.6, observe that:

$$\forall t \geq 0, \quad \mathbb{E}[m_t^{u,s}] \leq \mathbb{E}[m_{t-1}^{u,s}] \leq \cdots \leq \mathbb{E}[m_0^{u,s}] = 1\,. \tag{A.34}$$

Let τ be a stopping time relative to the filtration $\{\mathfrak{F}_t\}_{t=0}^{\infty}$. By Lemma A.11, we have that $m_\infty^{u,s} = \lim_{t\to\infty} m_t^{u,s}$ almost surely exists. And by Lemma A.8, $m_\tau^{u,s}$ is a well-defined random variable. Consider now a stopped version of $\{m_t^{u,s}\}_{t=0}^{\infty}$, denoted as $q_t^{u,s} := m_{\min\{\tau,t\}}^{u,s}$, $t \geq 0$, which is also a non-negative super-martingale by Lemma A.9. By Fatou's lemma (Lemma A.10), we then have that:

$$\begin{aligned}\mathbb{E}[m_\tau^{u,s}] &= \mathbb{E}\left[\lim_{t\to\infty} q_t^{u,s}\right] = \mathbb{E}\left[\liminf_{t\to\infty} q_t^{u,s}\right] \\ &\leq \liminf_{t\to\infty} \mathbb{E}[q_t^{u,s}] \\ &= \liminf_{t\to\infty} \mathbb{E}\left[m_{\min\{\tau,t\}}^{u,s}\right] \leq 1\,.\end{aligned} \tag{A.35}$$

The next step in the proof is to consider a super-martingale construction using a random noise sequence and a random function drawn from the GP prior, instead of s and u, respectively. Let $\mathfrak{F}_\infty := \mathfrak{A}\left(\bigcup_{t=0}^{\infty} \mathfrak{F}_t\right)$, and let $\tilde{s} := \{\tilde{s}_t\}_{t=1}^{\infty}$ be a sequence of i.i.d. zero-mean Gaussian random variables with variance η independent of $\mathfrak{F}_\infty$. Recall that a Gaussian process $\mathrm{GP}(0, \hat{k})$, with $\hat{k} : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$, defines a distribution over the function space $\mathbb{R}^{\mathcal{W}}$, which is the vector space of all functions mapping $\mathcal{W}$ to $\mathbb{R}$. Let $\tilde{u}$ denote a random function distributed according to $\mathrm{GP}(0, \hat{k})$ and independent of both $\mathfrak{F}_\infty$ and $\tilde{s}$.

Define $m_t := \mathbb{E}[m_t^{\tilde{u},\tilde{s}} \mid \mathfrak{F}_\infty]$, $t \geq 0$. Notice that, for all $t \geq 0$, ξ_t and ν_t are $\mathfrak{F}_\infty$-measurable. For any realisation (u, s) of $(\tilde{u}, \tilde{s})$, we also have that $\{m_t^{u,s}\}_{t=0}^{\infty}$ is a super-martingale adapted to $\{\mathfrak{F}_t\}_{t=0}^{\infty}$ by Lemma A.6. It is then easy to see that $\{m_t\}_{t=0}^{\infty}$ is also a non-negative super-martingale relative to $\{\mathfrak{F}_t\}_{t=0}^{\infty}$. Again by Lemma A.11, we have that $m_\infty = \lim_{t\to\infty} m_t$ almost surely exists. Then we can write:

$$\mathbb{E}[m_t] = \mathbb{E}\left[m_t^{\tilde{u},\tilde{s}}\right] = \mathbb{E}\left[\mathbb{E}\left[m_t^{\tilde{u},\tilde{s}} \,\middle|\, \tilde{u}, \tilde{s}\right]\right] \leq \mathbb{E}[1]\,, \quad \forall t \geq 0 \tag{A.36}$$

Following an argument similar to Equation A.35 also leads to $\mathbb{E}[m_\tau] \leq 1$ for any stopping time τ. For the next steps, without loss of generality, let us assume $\sigma_\nu = 1$, which is always achievable through appropriate scaling[1]. Computing m_t, we have:

$$\begin{aligned} m_t &= \mathbb{E}\left[\exp\left(\boldsymbol{\nu}_t^{\mathsf{T}}(\tilde{\mathbf{u}}_t + \tilde{\mathbf{s}}_t) - \frac{1}{2}\|\tilde{\mathbf{u}}_t + \tilde{\mathbf{s}}_t\|_2^2\right) \,\middle|\, \mathfrak{F}_\infty\right] \\ &= \int_{\mathbb{R}^t}\int_{\mathbb{R}^t} \exp\left(\boldsymbol{\nu}_t^{\mathsf{T}}(\mathbf{u}_t + \mathbf{s}_t) - \frac{1}{2}\|\mathbf{u}_t + \mathbf{s}_t\|_2^2\right) \mathrm{d}N(\mathbf{u}_t; \mathbf{0}, \hat{\mathbf{K}}_t)\,\mathrm{d}N(\mathbf{s}_t; \mathbf{0}, \eta\mathbf{I}) \\ &= \int_{\mathbb{R}^t} \exp\left(\boldsymbol{\nu}_t^{\mathsf{T}}\boldsymbol{\lambda} - \frac{1}{2}\|\boldsymbol{\lambda}\|_2^2\right) p(\boldsymbol{\lambda}; \mathbf{0}, \hat{\mathbf{K}}_t + \eta\mathbf{I})\,\mathrm{d}\boldsymbol{\lambda}\,, \end{aligned} \tag{A.37}$$

where the second equality comes from the GP, for $\tilde{\mathbf{u}}_t = [\tilde{u}(\xi_1), \dots, \tilde{u}(\xi_t)]^{\mathsf{T}} \sim N(\mathbf{0}, \hat{\mathbf{K}}_t)$. For a given positive-definite matrix $\boldsymbol{\Sigma}$, the third equality uses $\mathrm{d}N(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma}) = p(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma})\,\mathrm{d}\mathbf{x}$, where $p(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma})$ is the probability density function (with respect to the Lebesgue measure) of the corresponding Gaussian distribution. Thus, we have:

$$\begin{aligned} m_t &= \frac{1}{\sqrt{(2\pi)^t|\hat{\mathbf{K}}_t + \eta\mathbf{I}|}} \int_{\mathbb{R}^t} \exp\left(\boldsymbol{\nu}_t^{\mathsf{T}}\boldsymbol{\lambda} - \frac{1}{2}\|\boldsymbol{\lambda}\|_2^2 - \frac{1}{2}\|\boldsymbol{\lambda}\|^2_{(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1}}\right) \mathrm{d}\boldsymbol{\lambda} \\ &= \frac{\exp\left(\frac{\|\boldsymbol{\nu}_t\|_2^2}{2}\right)}{\sqrt{(2\pi)^t|\hat{\mathbf{K}}_t + \eta\mathbf{I}|}} \int_{\mathbb{R}^t} \exp\left(-\frac{1}{2}\|\boldsymbol{\lambda} - \boldsymbol{\nu}_t\|_2^2 - \frac{1}{2}\|\boldsymbol{\lambda}\|^2_{(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1}}\right) \mathrm{d}\boldsymbol{\lambda}\,. \end{aligned} \tag{A.38}$$

[1] For example, if $y_t = \hat{f}(\xi_t) + \nu_t$, let $y'_t := cy_t$, $c > 0$. Then y'_t is such that $y'_t = \hat{f}'(\xi_t) + \nu'_t$, where ν'_t is $c\sigma_\nu$-sub-Gaussian and $\|\hat{f}'\|_{\hat{k}} = c\|\hat{f}\|_{\hat{k}}$

Given two positive-definite matrices $\mathbf{P}$ and $\mathbf{N}$ with the same dimensionality, we have:

$$\|\mathbf{x}-\mathbf{a}\|_{\mathbf{P}}^2 + \|\mathbf{x}\|_{\mathbf{Q}}^2 = \|\mathbf{x}-(\mathbf{P}+\mathbf{Q})^{-1}\mathbf{P}\mathbf{a}\|_{\mathbf{P}+\mathbf{Q}}^2 + \|\mathbf{a}\|_{\mathbf{P}}^2 - \|\mathbf{P}\mathbf{a}\|_{(\mathbf{P}+\mathbf{Q})^{-1}}^2 \,. \tag{A.39}$$

Applying this last equation to Equation A.38, we obtain:

$$\begin{aligned}\|\boldsymbol{\lambda}-\boldsymbol{\nu}_t\|_{\mathbf{I}}^2 + \|\boldsymbol{\lambda}\|_{(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1}}^2 =& \|\boldsymbol{\lambda}-(\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1})^{-1}\mathbf{I}\boldsymbol{\nu}_t\|_{\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1}} \\ &+ \|\boldsymbol{\nu}_t\|_{\mathbf{I}}^2 - \|\mathbf{I}\boldsymbol{\nu}_t\|_{(\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1})^{-1}}^2 \,,\end{aligned} \tag{A.40}$$

which yields:

$$\begin{aligned} m_t &= \frac{1}{\sqrt{(2\pi)^t|\hat{\mathbf{K}}_t+\eta\mathbf{I}|}} \exp\left(\frac{1}{2}\|\boldsymbol{\nu}_t\|_{(\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1})^{-1}}^2\right) \\ &\quad\times \int_{\mathbb{R}^t} \exp\left(-\frac{1}{2}\left\|\boldsymbol{\lambda}-(\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1})^{-1}\boldsymbol{\nu}_t\right\|_{\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1}}^2\right) \mathrm{d}\boldsymbol{\lambda} \\ &= \frac{1}{\sqrt{|\hat{\mathbf{K}}_t+\eta\mathbf{I}||(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1}+\mathbf{I}|}} \exp\left(\frac{1}{2}\|\boldsymbol{\nu}_t\|_{(\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1})^{-1}}^2\right) \\ &= \frac{1}{\sqrt{|\mathbf{I}+\hat{\mathbf{K}}_t+\eta\mathbf{I}|}} \exp\left(\frac{1}{2}\|\boldsymbol{\nu}_t\|_{(\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1})^{-1}}^2\right) \,, \end{aligned} \tag{A.41}$$

where the second equality follows from:

$$\int_{\mathbb{R}^t} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{a})^\mathsf{T}\mathbf{A}(\mathbf{x}-\mathbf{a})\right) \mathrm{d}\mathbf{x} = \int_{\mathbb{R}^t} \exp\left(-\frac{1}{2}\|\mathbf{x}-\mathbf{a}\|_{\mathbf{A}}^2\right) \mathrm{d}\mathbf{x} = \sqrt{\frac{(2\pi)^t}{|\mathbf{A}|}} \,, \tag{A.42}$$

for a positive-definite matrix $\mathbf{A} \in \mathbb{R}^{t\times t}$.

Done with the algebraic manipulations, let's turn our attention back to the super-martingale $\{m_t\}_{t=0}^{\infty}$ at a given stopping time τ. Considering the result in Equation A.41, we can use Markov's inequality (Lemma A.12) to derive concentration bounds on the norm of the noise vector at the stopping time τ. As $\mathbb{E}[m_\tau] \le 1$, for any $\delta \in (0,1)$, we have:

$$\mathbb{P}\left\{\|\boldsymbol{\nu}_\tau\|_{(\mathbf{I}+(\hat{\mathbf{K}}_\tau+\eta\mathbf{I})^{-1})^{-1}}^2 > 2\log\left(\sqrt{|(1+\eta)\mathbf{I}+\hat{\mathbf{K}}_\tau|}/\delta\right)\right\} = \mathbb{P}\left\{m_\tau > 1/\delta\right\} < \delta\mathbb{E}[m_\tau] \le \delta \,, \tag{A.43}$$

which follows by applying Lemma A.12 and noticing that $\mathbb{P}\{m_\tau > 1/\delta\} = \mathbb{P}\{\log(m_\tau) > \log(1/\delta)\}$, as $\log$ is a monotonic function.

Now we can derive a specific stopping time $\tau : \mathcal{E} \to \mathbb{N} \cup \{0, \infty\}$ to indicate the time when $\boldsymbol{\nu}_\tau$ violates the bound in Equation A.43. Consider the abstract probability space $(\mathcal{E}, \mathfrak{E}, P_{\mathcal{E}})$ where all the random variables in Theorem 6.12 are defined. Following the same stopping time construction as in Chowdhury and Gopalan (2017), for each $t \geq 0$, define the bad event:

$$\mathcal{B}_t(\delta) = \left\{ \omega \in \mathcal{E} \;\middle|\; \|\boldsymbol{\nu}_t\|^2_{(\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1})^{-1}} > 2\log\left(\sqrt{|(1+\eta)\mathbf{I}+\hat{\mathbf{K}}_t|}/\delta\right) \right\} , \tag{A.44}$$

so that:

$$P_{\mathcal{E}}\left[\bigcup_{t\geq 0} \mathcal{B}_t(\delta)\right] = \mathbb{P}\left\{ \exists t \geq 0 : \|\boldsymbol{\nu}_t\|^2_{(\mathbf{I}+(\hat{\mathbf{K}}_t+\eta\mathbf{I})^{-1})^{-1}} > 2\log\left(\sqrt{|(1+\eta)\mathbf{I}+\hat{\mathbf{K}}_t|}/\delta\right) \right\} . \tag{A.45}$$

Bounding this last probability by δ, the opposite event, which is that the norm of $\boldsymbol{\nu}_t$ stays within the proposed bound for all $t \geq 0$, will have probability at least $1 - \delta$, as stated by Theorem 6.12.

Let τ_δ be the time when the first $\mathcal{B}_t(\delta)$ happens, i.e. $\tau_\delta(\omega) := \min\{t \geq 0 \mid \omega \in \mathcal{B}_t(\delta)\}$, with the convention that $\min\{\varnothing\} := \infty$. By the measurability requirements in Theorem 6.12's statement, every random variable defining a set $\mathcal{B}_t(\delta)$ in Equation A.44 is $\mathfrak{F}_t$-measurable[2], so that $\mathcal{B}_t \in \mathfrak{F}_t$, $\forall t \geq 0$. Then τ_δ is a stopping time according to Definition A.7, as:

$$\forall t \geq 0, \quad \{\tau_\delta \leq t\} = \bigcup_{i=0}^{t} \mathcal{B}_i(\delta) \in \mathfrak{F}_t , \tag{A.46}$$

and $\{\tau_\delta \leq \infty\} = \mathcal{E}$, which is in any σ-algebra, including $\mathfrak{F}_\infty$. Finally, we obtain:

$$\begin{aligned} P_{\mathcal{E}}\left[\bigcup_{t\geq 0} \mathcal{B}_t(\delta)\right] &= \mathbb{P}\left\{\tau_\delta < \infty\right\} \\ &= \mathbb{P}\left\{ \|\boldsymbol{\nu}_{\tau_\delta}\|^2_{(\mathbf{I}+(\hat{\mathbf{K}}_{\tau_\delta}+\eta\mathbf{I})^{-1})^{-1}} > 2\log\left(\sqrt{|(1+\eta)\mathbf{I}+\hat{\mathbf{K}}_{\tau_\delta}|}/\delta\right), \tau_\delta < \infty \right\} \\ &\leq \mathbb{P}\left\{ \|\boldsymbol{\nu}_{\tau_\delta}\|^2_{(\mathbf{I}+(\hat{\mathbf{K}}_{\tau_\delta}+\eta\mathbf{I})^{-1})^{-1}} > 2\log\left(\sqrt{|(1+\eta)\mathbf{I}+\hat{\mathbf{K}}_{\tau_\delta}|}/\delta\right) \right\} \\ &\leq \delta \end{aligned} \tag{A.47}$$

by Equation A.43. When $\hat{\mathbf{K}}_t$ is guaranteed to be a positive-definite matrix for all $t \geq 1$, repeating the first part of the proof with $\eta = 0$ leads to the conclusion in Theorem 6.12 for $\eta = 0$. Joining Equation A.47 with Equation A.45 yields the result in Theorem 6.12, which concludes the proof. □

[2] Recall that, for being a filtration, $\mathfrak{F}_t \supset \mathfrak{F}_{t-1} \supset \cdots \supset \mathfrak{F}_0$. Then $\{\nu_i\}_{i=1}^t$ are $\mathfrak{F}_t$-($\mathfrak{B}^1$-)measurable, and $\{\xi_i\}_{i=1}^\infty$ are $\mathfrak{F}_t$-$\mathfrak{W}$-measurable. The kernel $\hat{k}$ is $\mathfrak{W}$-($\mathfrak{B}^1$-)measurable. The remaining variables are continuous mappings between Euclidean spaces, which are Borel-measurable (see Section 2.6). Therefore, the entire composition is $\mathfrak{F}_t$-measurable.

A.6 Proof of Proposition 6.9

This section presents the derivation of a proof for Proposition 6.9. The proof will make use of the following background. For further details, see Bauer (1981) and Boucheron et al. (2013).

DEFINITION A.13 (Absolute continuity). *A measure V on a σ-algebra $\mathfrak{X}$ is said to be absolutely continuous relative to a measure U on $\mathfrak{X}$ if every U-null set is also a V-null set.*

Given a measure M on $\mathfrak{X}$, a M-null set is simply any set $\mathcal{U} \in \mathfrak{X}$, such that $M[\mathcal{U}] = 0$.

DEFINITION A.14 (Kullback-Leibler divergence). *Let P and P' be two probability measures on $(\mathcal{X}, \mathfrak{X})$. The Kullback-Leibler divergence between the two measures is defined as:*

$$D_{\mathrm{KL}}(P'||P) := \int \log \frac{\mathrm{d}P'}{\mathrm{d}P} \, \mathrm{d}P' \,, \tag{A.48}$$

case P' is absolutely continuous relative to P, or ∞ otherwise.

LEMMA A.15 (Pinsker's inequality). *Let P and P' be two probability measures on $(\mathcal{X}, \mathfrak{X})$, and let U be a common dominating measure of P and P'. Assume that P' is absolutely continuous relative to P. Then it holds that:*

$$\int_{\mathcal{X}} \left|p(\mathbf{x}) - p'(\mathbf{x})\right| \, \mathrm{d}U(\mathbf{x}) \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(P'||P)} \,, \tag{A.49}$$

where $p = \frac{\mathrm{d}P}{\mathrm{d}U}$ and $p' = \frac{\mathrm{d}P'}{\mathrm{d}U}$ are the respective densities of each probability measure.

Now we can proceed to the proof of Proposition 6.9, which is restated below.

PROPOSITION 6.9. *Let $\mathcal{X} = \mathbb{R}^d$, $f \in \mathcal{H}_k$ and $\|f\|_k \leq b$. Assume that, for any $\mathbf{x} \in \mathcal{S}$, the query distribution $P^E_{\mathbf{x}}$ is Gaussian with mean $\mathbf{x}$ and positive-definite covariance $\mathbf{\Sigma}^E$. Then, using a Gaussian model $\hat{P}_{\mathbf{x}}$ with same mean and a given constant positive-definite covariance matrix $\hat{\mathbf{\Sigma}}$, we have that:*

$$\forall \mathbf{x} \in \mathcal{S}, \quad \left|\mathbb{E}_{P^E_{\mathbf{x}}}[f] - \mathbb{E}_{\hat{P}_{\mathbf{x}}}[f]\right| \leq \frac{b}{2} \left(\mathrm{tr}(\hat{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}^E) - d + \log \frac{|\hat{\mathbf{\Sigma}}|}{|\mathbf{\Sigma}^E|}\right)^{1/2} .$$

PROOF. Proposition 6.9 refers to the approximation error between the model $\hat{P}^t_{\mathbf{x}}$ and the actual distribution $P^E_{\mathbf{x}}$ in terms of difference in the expected value of a function $f \in \mathcal{H}_k$. The result simply follows by applying Pinsker's inequality (Lemma A.15).

For any $t \geq 1$ and $\mathbf{x} \in \mathcal{S}$, let $\hat{p}^t_{\mathbf{x}}$ and $p_{\mathbf{x}}$ denote the probability density functions of $\hat{P}^t_{\mathbf{x}}$ and $P^E_{\mathbf{x}}$, respectively. Then we have that:

$$|\mathbb{E}_{P^E_{\mathbf{x}}}[f]-\mathbb{E}_{\hat{P}^t_{\mathbf{x}}}[f]| = \left|\int_{\mathcal{X}} f(\mathbf{x})(p_{\mathbf{x}}(\mathbf{x}') - \hat{p}^t_{\mathbf{x}}(\mathbf{x}')\,\mathrm{d}\mathbf{x}'\right| \leq \sup_{\mathbf{x}''\in\mathcal{X}} |f(\mathbf{x}'')| \int_{\mathcal{X}} |p_{\mathbf{x}}(\mathbf{x}')-\hat{p}^t_{\mathbf{x}}(\mathbf{x}')|\,\mathrm{d}\mathbf{x}' \quad (A.50)$$

Now note that, by the Cauchy-Schwartz inequality and k's reproducing property, for any $\mathbf{x} \in \mathcal{X}$,

$$\sup_{\mathbf{x}\in\mathcal{X}} |f(\mathbf{x})| = \sup_{\mathbf{x}\in\mathcal{X}} |\langle f, k(\cdot,\mathbf{x})\rangle_k| \leq \sup_{\mathbf{x}\in\mathcal{X}} \|f\|_k \sqrt{k(\mathbf{x},\mathbf{x})} = \|f\|_k \,, \quad (A.51)$$

since $k(\mathbf{x},\mathbf{x}) = 1$ under our general regularity assumptions.

As both $\hat{P}^t_{\mathbf{x}}$ and $P^E_{\mathbf{x}}$ are Gaussian measures, their support is the whole $\mathcal{X}$, so that they are absolutely continuous with respect to each other. Then we can apply Pinsker's inequality to upper bound the remaining term in Equation A.50, which yields:

$$\int_{\mathcal{X}} |p_{\mathbf{x}}(\mathbf{x}') - \hat{p}^t_{\mathbf{x}}(\mathbf{x}')|\,\mathrm{d}\mathbf{x}' = \int_{\mathcal{X}} |\hat{p}^t_{\mathbf{x}}(\mathbf{x}') - p_{\mathbf{x}}(\mathbf{x}')|\,\mathrm{d}\mathbf{x}' \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(P^E_{\mathbf{x}}||\hat{P}^t_{\mathbf{x}})}\,. \quad (A.52)$$

Plugging this result and the one in Equation A.51 back into Equation A.50 yields:

$$\forall t \geq 1,\, \forall \mathbf{x} \in \mathcal{S}, \quad |\mathbb{E}_{P^E_{\mathbf{x}}}[f] - \mathbb{E}_{\hat{P}^t_{\mathbf{x}}}[f]| \leq \|f\|_k \sqrt{\frac{1}{2} D_{\mathrm{KL}}(P^E_{\mathbf{x}}||\hat{P}^t_{\mathbf{x}})}\,. \quad (A.53)$$

The Kullback-Leibler divergence between two Gaussian distributions on $\mathbb{R}^d$, $P^E_{\mathbf{x}}$ and $\hat{P}^t_{\mathbf{x}}$, with covariance matrices as stated and same mean vectors is given by:

$$D_{\mathrm{KL}}(P^E_{\mathbf{x}}||\hat{P}^t_{\mathbf{x}}) = \frac{1}{2}\left(\mathrm{tr}(\hat{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}^E) - d + \log\frac{|\hat{\mathbf{\Sigma}}|}{|\mathbf{\Sigma}^E|}\right)\,, \quad (A.54)$$

which comes from a known result (Rasmussen and Williams, 2006, p. 203) and the fact that:

$$\mathrm{tr}(\hat{\mathbf{\Sigma}}^{-1}(\mathbf{\Sigma}^E - \hat{\mathbf{\Sigma}})) = \mathrm{tr}(\hat{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}^E - \mathbf{I}) = \mathrm{tr}(\hat{\mathbf{\Sigma}}^{-1}\mathbf{\Sigma}^E) - d\,. \quad (A.55)$$

Replacing Equation A.54 into Equation A.53 yields the result in Proposition 6.9. □

A.7 Proof of Proposition 6.10

PROPOSITION 6.10. *Consider a compact set $\mathcal{S} \subset \mathcal{X}$, a distribution $P_{\boldsymbol{\epsilon}} \in \mathcal{P}$, with $\mathbb{E}_{P_{\boldsymbol{\epsilon}}}[\boldsymbol{\epsilon}] = 0$, and a set:*

$$\mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S}) := \{P \in \mathcal{P} \mid \tilde{\mathbf{x}} \sim P \Longrightarrow \tilde{\mathbf{x}} = \hat{\mathbf{x}} + \boldsymbol{\epsilon},\, \boldsymbol{\epsilon} \sim P_{\boldsymbol{\epsilon}},\, \hat{\mathbf{x}} \in \mathcal{S}\}\,, \quad (6.33)$$

which is the set of location distributions with mean on $\mathcal{S}$ and affected by i.i.d. $P_{\boldsymbol{\epsilon}}$-noise. Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a continuous translation-invariant kernel, and $\hat{k} : \mathcal{P} \times \mathcal{P} \to \mathbb{R}$ be according to Equation 6.7. Then we have that:

$$\forall n \geq 1, \quad \hat{\gamma}_n(\mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S})) \leq \gamma_n(\mathcal{S}) \ . \tag{6.34}$$

PROOF. Let's consider the definitions of the information gain bounds. Given a compact set $\mathcal{S} \subset \mathcal{X}$, the maximum information gain after n iterations for a model $\mathrm{GP}(0, k)$ is given by:

$$\gamma_n(\mathcal{S}) = \max_{\mathcal{Q} \subset \mathcal{S} : |\mathcal{Q}| = n} \frac{1}{2} \log |\mathbf{I} + \lambda^{-1} \mathbf{K}_{\mathcal{Q}}| \ , \tag{A.56}$$

where $\mathbf{K}_{\mathcal{Q}} = [k(\mathbf{x}, \mathbf{x}')]_{\mathbf{x}, \mathbf{x}' \in \mathcal{Q}}$. In the case of $\mathrm{GP}(0, \hat{k})$ taking inputs from $\mathcal{P}_{\boldsymbol{\epsilon}}$, we have:

$$\hat{\gamma}_n(\mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S})) = \sup_{\mathcal{R} \subset \mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S}) : |\mathcal{R}| = n} \frac{1}{2} \log |\mathbf{I} + \lambda^{-1} \hat{\mathbf{K}}_{\mathcal{R}}| \ , \tag{A.57}$$

where $\hat{\mathbf{K}}_{\mathcal{R}} = [k(P, P')]_{P, P' \in \mathcal{Q}}$. Both cases have the same parameter $\lambda > 0$.

Considering the definitions above, observe that, if one can always find a $\mathcal{Q} \subset \mathcal{S}$ that provides larger information gain than $\mathcal{R}$, for each choice of $\mathcal{R} \subset \mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S})$, $\gamma_n(\mathcal{S})$ will then be larger than $\hat{\gamma}_n(\mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S}))$. The information gain is dependent on the determinants of the matrices $\mathbf{I} + \lambda^{-1}\mathbf{K}_{\mathcal{Q}}$ and $\mathbf{I} + \lambda^{-1}\hat{\mathbf{K}}_{\mathcal{R}}$ and is related to the positive-definiteness of both matrices.

A classic result in matrix analysis is that, if two n-by-n-matrices $\mathbf{A}$ and $\mathbf{B}$ are positive definite, and $\mathbf{A} - \mathbf{B}$ is positive semi-definite, we have that $|\mathbf{A}| \geq |\mathbf{B}|$ (see Horn and Johnson, 1985, Cor. 7.7.4). Recall that a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive semi-definite if and only if $\forall \boldsymbol{\alpha} \in \mathbb{R}^n$, $\boldsymbol{\alpha}^\mathsf{T} \mathbf{A} \boldsymbol{\alpha} \geq 0$, and positive definite if equality only holds for $\boldsymbol{\alpha} = \mathbf{0}$. Hence, we shall prove that:

$$\forall \{P_i\}_{i=1}^n \subset \mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S}), \quad \exists \{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{S} : \quad \boldsymbol{\alpha}^\mathsf{T}(\mathbf{K}_n - \hat{\mathbf{K}}_n)\boldsymbol{\alpha} \geq 0, \quad \forall \boldsymbol{\alpha} \in \mathbb{R}^n \ , \tag{A.58}$$

where $[\mathbf{K}_n]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $[\hat{\mathbf{K}}_n]_{ij} = \hat{k}(P_i, P_j)$, $i, j \in \{1, \dots, n\}$. For two positive semi-definite matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, let $\mathbf{A} \succcurlyeq \mathbf{B}$ denote that $\mathbf{A} - \mathbf{B}$ is positive semi-definite, i.e. $\mathbf{A} - \mathbf{B} \succcurlyeq 0$. Since:

$$\mathbf{I} + \lambda^{-1}\mathbf{K}_n \succcurlyeq \mathbf{I} + \lambda^{-1}\hat{\mathbf{K}}_n \quad \iff \quad \mathbf{K}_n - \hat{\mathbf{K}}_n \succcurlyeq 0 \ , \tag{A.59}$$

the condition in Equation A.58, if satisfied, implies that $|\mathbf{I} + \lambda^{-1}\mathbf{K}_n| \geq |\mathbf{I} + \lambda^{-1}\hat{\mathbf{K}}_n|$.

For a given $\{P_i\}_{i=1}^n \subset \mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S})$, define $\tilde{\mathbf{x}}_i \sim P_i \in \mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S})$, for each $i \in \{1, \dots, n\}$. By the definition of $\mathcal{P}_{\boldsymbol{\epsilon}}(\mathcal{S})$, we also have that each $\tilde{\mathbf{x}}_i = \hat{\mathbf{x}}_i + \boldsymbol{\epsilon}_i$, with $\hat{\mathbf{x}}_i \in \mathcal{S}$ and $\boldsymbol{\epsilon}_i \sim P_{\boldsymbol{\epsilon}}$. Recall that, for any $P, P' \in \mathcal{P}$,

$\hat{k}(P, P') = \langle \psi_P, \psi_{P'} \rangle_k$ and $\psi_P = \mathbb{E}[k(\cdot, \tilde{\mathbf{x}})]$, $\tilde{\mathbf{x}} \sim P$. Then we can write:

$$\begin{aligned}
\forall \boldsymbol{\alpha} \in \mathbb{R}^n, \quad \boldsymbol{\alpha}^\mathsf{T} \hat{\mathbf{K}}_n \boldsymbol{\alpha} &= \sum_{i=1}^n \sum_{i=1}^n \alpha_i \alpha_j \hat{k}(P_i, P_j) \\
&= \sum_{i=1}^n \sum_{i=1}^n \alpha_i \alpha_j \langle \psi_{P_i}, \psi_{P_j} \rangle_k \\
&= \left\| \sum_{i=1}^n \alpha_i \psi_{P_i} \right\|_k^2 \\
&= \left\| \sum_{i=1}^n \alpha_i \mathbb{E}\left[k(\cdot, \hat{\mathbf{x}}_i + \boldsymbol{\epsilon}_i) \right] \right\|_k^2
\end{aligned} \tag{A.60}$$

Now, as $\boldsymbol{\epsilon}_i$ are i.i.d. random variables following $P_{\boldsymbol{\epsilon}}$, using $\boldsymbol{\epsilon} \sim P_{\boldsymbol{\epsilon}}$, it holds that:

$$\forall i \in \{1, \dots, n\}, \quad \mathbb{E}[k(\cdot, \hat{\mathbf{x}}_i + \boldsymbol{\epsilon}_i)] = \mathbb{E}[k(\cdot, \hat{\mathbf{x}}_i + \boldsymbol{\epsilon})] \ . \tag{A.61}$$

Applying this fact, we have that:

$$\begin{aligned}
\forall \boldsymbol{\alpha} \in \mathbb{R}^n, \quad \boldsymbol{\alpha}^\mathsf{T} \hat{\mathbf{K}}_n \boldsymbol{\alpha} &= \left\| \mathbb{E}\left[\sum_{i=1}^n \alpha_i k(\cdot, \hat{\mathbf{x}}_i + \boldsymbol{\epsilon}_i) \right] \right\|_k^2 \\
&= \left\| \mathbb{E}\left[\sum_{i=1}^n \alpha_i k(\cdot, \hat{\mathbf{x}}_i + \boldsymbol{\epsilon}) \right] \right\|_k^2 \\
&\leq \mathbb{E}\left[\left\| \sum_{i=1}^n \alpha_i k(\cdot, \hat{\mathbf{x}}_i + \boldsymbol{\epsilon}) \right\|_k^2 \right] \\
&= \mathbb{E}\left[\sum_{i=1}^n \sum_{i=1}^n \alpha_i \alpha_j k(\hat{\mathbf{x}}_i + \boldsymbol{\epsilon}, \hat{\mathbf{x}}_j + \boldsymbol{\epsilon}) \right] \\
&= \sum_{i=1}^n \sum_{i=1}^n \alpha_i \alpha_j k(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j) \\
&= \boldsymbol{\alpha}^\mathsf{T} \mathbf{K}_n \boldsymbol{\alpha} \ ,
\end{aligned} \tag{A.62}$$

where the first inequality follows by properties of the Bochner integral (see Equation 2.93), the fourth equality follows from k's translation invariance, and $\mathbf{K}_n$ is defined by $[\mathbf{K}_n]_{ij} = k(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)$. Therefore, the set of mean locations $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ satisfies the condition in Equation A.58, which concludes the proof. □