Ground Robot Energy Prediction and Reachability in Off-Road Environments Through Spatial Terrain Mapping

by

Michael Andrew Quann

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2019

Doctoral Committee:

Associate Professor Kira Barton, Co-Chair
Associate Research Scientist Lauro Ojeda, Co-Chair
Assistant Professor Dimitra Panagou
Dr. Denise Rizzo, US Army CCDC Ground Vehicle Systems Center
Professor Anna Stefanopoulou

Michael A. Quann

maquann@umich.edu

ORCID iD: 0000-0002-3665-0369

To my parents, Richard and Paula

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Kira Barton, for taking me on in 2014 and providing patient and persistent guidance through these years. I'd also like to thank my co-advisor Lauro Ojeda, as well as our collaborators at the U.S. Army Ground Vehicle Systems Center: Dr. Denise Rizzo, Dr. William Smith, and Dr. Matthew Castanier, for their invaluable support on this project.

I was fortunate to have a lab, the Barton Research Group, that is filled with brilliant, capable, kind, and fun people who have provided support and feedback on my work on countless occasions. I thank them and wish them the best as many of them continue on to complete their PhDs.

I've had the privilege of many good friends in Ann Arbor, particularly Adam Carlson, Aritra Sasmal, Marie Rice, Lynn Garrett, Kevin Weld, Deema Totah, Ilya Kovalenko and Callan Luetkemeyer. They made the long road of the PhD so much more enjoyable through homebrewing, golfing, and happy hours. To Victoria Edwards: thank you for your unwavering support of me and your belief in me. I am grateful for the many adventures we have had together and I look forward to many more in the future!

Finally, I feel lucky to have such a wonderful family, which has grown quickly through my 5 years at Michigan. Thank you to my parents, sisters, brother, grandmother, brothers-in-law, and my three nieces and two nephews.

The work in this thesis has been supported by the Automotive Research Center, a US Army Center of Excellence for modeling and simulation of ground vehicles, headquartered at the University of Michigan.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figures

LIST OF TABLES

Tables

LIST OF APPENDICES

Appendices

ABSTRACT

For robotic applications, energy is a key resource that can both enable and limit the tasks that a robot can perform in an environment. In off-road environments, ground robots may traverse numerous different terrains with significantly and spatially varying energy costs. The cost of a particular robot moving through such an environment is likely to be uncertain, making mission planning and decision-making challenging. In this dissertation, we develop methods that use information on terrain traversal energy costs, collected during robot operation, so that future energy costs for the robot can be more accurately and confidently predicted. The foundation of these methods is to build a *spatial map* of the energy costs in an environment, while characterizing the uncertainty in those costs, using a technique known as Gaussian process regression (GPR). This map can be used to improve performance in important robotic applications, including path and mission planning.

First, we present a 2-dimensional energy mapping formulation, based on GPR, that properly considers the correlation in path energy costs for computing the uncertainty in the predicted energy cost of a path through the environment. With this formulation, we define a robot's chance constrained reachability as the set of locations that the robot can reach, under a user-defined confidence level, without depleting its energy budget. Simulation results show that as a robot collects more data on the environment, the reachable set becomes more accurately known, making it a useful tool for mission planning applications. Next, we extend the spatial mapping formulation to 3-dimensional environments by considering both data-driven and vehicle modeling strategies. Experimental testing is performed on ground robot platforms in an environment with varied terrains. The results show that the predictive accuracy of the spatial mapping methodology is significantly improved over baseline approaches. Finally, we explore information sharing between heterogeneous robot platforms. Two different robots are likely to have different spatial maps, however, useful information may still be shared between the robots. We present a framework, based multi-task Gaussian process regression (MTGP), for learning the scaling and correlation in costs between different robots, and provide simulation and experimental results demonstrating its effectiveness. Using the framework, robot heterogeneity can be leveraged to improve performance in planning applications.

CHAPTER I

Introduction

1.1 Background

Ground robots face numerous challenges in off-road environments due to rough terrain and topography. Importantly, it is hard to predict energy costs through an environment with varying terrains. Energy constrains the range and effectiveness of ground robots across autonomous, manned, and tele-operation applications. Typically, only limited knowledge of future energy costs is available in off-road environments, making robot range difficult to quantify. The uncertainty in predicted energy costs makes mission planning for ground robots difficult, resulting in overly conservative decisions or risking energy depletion. ***Informed methods of predicting costs throughout off-road environments are necessary for robust operation of ground robots.*** Such methods can be used to plan energy-efficient paths, extend robot range, and inform robust mission planning.

Energy limitations play a significant role in off-road ground robot applications, particularly in the field of autonomous systems. Such limitations exist in large vehicles, but are especially prevalent in small ground robots as well. Ground robots are being used for numerous applications, such as: taking soil measurements for precision agriculture [105] and environmental health monitoring [23, 86], gas distribution modeling [99], radiation detection [44, 20], and planetary exploration [41, 69], among many others. See [29] for an extensive review of robotic environmental monitoring applications. In addition, numerous military applications of ground robots exist for reconnaissance/surveillance tasks [90]. These applications often take place in uncertain off-road environments with a complex set of terrains and are limited in range and reliability by energy considerations.

Internal factors play a major role in energy for ground robots, and there are the subject of significant, on-going research efforts. Batteries are the primary means of energy storage for small robotic vehicles. Research to improve battery technology continues on many fronts, including for materials discovery [55], modeling, and health monitoring [48]. For the forseeable future, however, batteries remain a limiting factor in the duration of missions for robotic vehicles [87]. For larger vehicles, many other energy storage and power generation options can also be considered, including: traditional internal combustion engines, hybrid vehicles, hydrogen fuel cells [113], solar powered vehicles [74], etc. Further sources of energy consumption onboard the robot come from increasingly sophisticated sensor suites, along with the computational power needed to process sensor data. It is important for an energy management system to consider these factors.

Environmental factors also have a significant impact on energy usage. For example, paths or roads crowded with humans or other traffic are more costly to traverse [8]. In off-road environments, the robot may traverse terrain that causes higher energy costs than pavement [87, 66], as seen in Figure 1.1. Increased costs are due to the wheel-terrain interaction that resists motion, called rolling resistance [82]. 3-dimensional topography (hills/changes in elevation) also affect energy usage, with much research devoted to planning in such environments [26, 116]. Travelling uphill clearly increases costs, while downhill costs are reduced (energy may even be gained if regenerative breaking is considered).

Predicting the cost of future paths must take environmental factors into account.

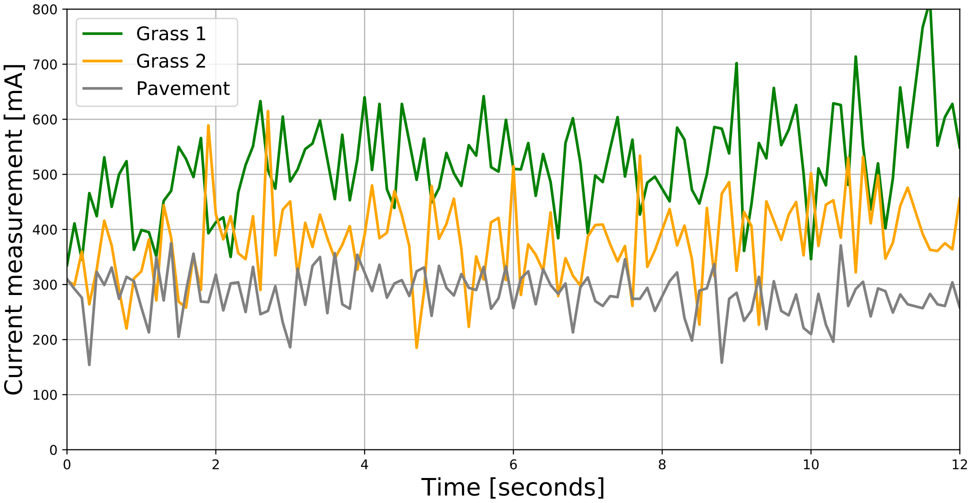


Figure 1.1: Current draw measurements from a small ground robot traversing three types of terrain (and two different types of grass) at a constant velocity.

However, there may be little prior information on the energy costs in an environment. Spatial soil maps and remote sensing may provide a good indication on some terrain properties, or to segment terrain into different regions [61]. The energy cost of traversal, however, is robot dependent and can only be accurately measured during robot operation in the environment. Camera and LiDAR data can also be used for segmentation and classification of terrains in off-road environments [59]. However, visually similar terrains may have varying energy costs (e.g. dry grass versus thick grass, as seen in Figure 1.1). Crude segmentation and classification based on exteroceptive data may give little indication of these differing costs.

On-board measurements of a robot's energy consumption is the most accurate way to estimate the energy costs of traversing a particular terrain. In some scenarios, a robot may repeatedly explore the same environment, collecting data on the terrain to improve performance in the future. There may even be multiple ground robots in the environment, with the ability to communicate information on the terrain between them to inform energy predictions. For many applications, however, robots may have traversed some parts of an off-road environment, but may not have explored the entire

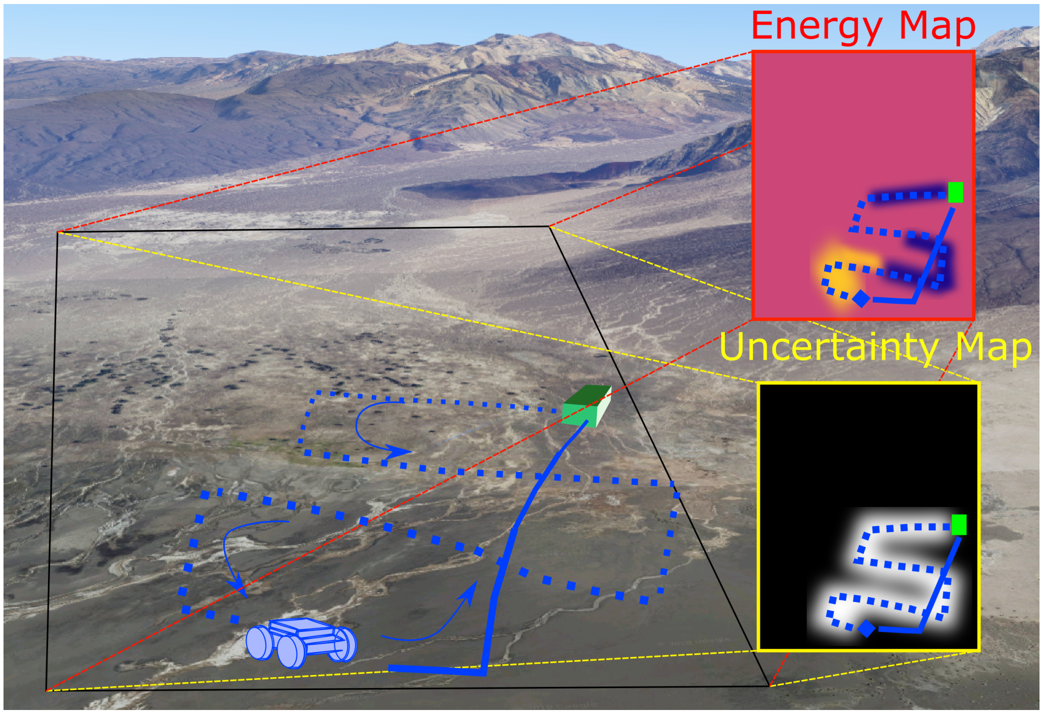


Figure 1.2: A robot traverses through an off-road environment, collecting data on the costs of the terrains. The data is used to produce spatial energy and uncertainty maps. Here, the robot has already traversed from a recharging base (green box) along the blue dotted line, and looks to predict the cost of the future path (solid red line). Lower and higher cost areas on the energy map are shaded blue and red, respectively. Lower and higher uncertainty areas are shaded white and black, respectively. Regions which have not been visited have more uncertainty in their energy costs.

area. Data on the energy costs of the environment are then sparse, implying that uncertainty should be taken into account when making predictions. The predicted cost of a future path should consider both data collected in previously traversed areas and uncertainty in areas not yet traversed by the robot.

This dissertation focuses on the development of methodologies for enabling informed energy predictions in off-road environments, based on data collected during robot operation, for use in robotic applications such as mission planning and path planning. The methodologies developed focus on using energy cost data to inform a spatial map of an environment. An illustration of the strategy for energy prediction is shown in Figure 1.2. To summarize, a spatial map of energy costs (or a metric related to energy), along with a corresponding uncertainty map, of the environment is built based on data collected during robot operation. The map informs the energy costs associated with a location in the environment, along with the uncertainty of those costs.

Energy predictions can be used to inform ground robotics applications through both mission planning and path planning in uncertain, off-road environments. Mission planning for single or multiple robots requires an understanding of the range of the robots in an environment, to ensure the robots have sufficient energy resources. One question we approach is: where can a robot safely traverse to, or reach, in an environment without depleting its energy? As more energy data is collected by the robot on the environment, a more accurate computation of the robot's *reachable set* will be available. Data on the energy costs of the environment also enables energy-efficient path planning. In both mission planning and path planning applications, uncertainty adds a complicating factor, so precautions must be taken in order to ensure robustness. Applied to mission planning, uncertainty may mean choosing the less risky path back to a recharging base over a more risky, yet informative path.

One challenge in using spatial mapping for energy prediction is that, due to the complex physics underlying robot-terrain interaction, energy costs due to the terrain will vary for different robots. Moreover, the same robot under a different operating condition (higher speed, heavier load, different weight distribution, etc.) can see a change in the rolling resistance coefficient in ways that are difficult to model physically. For robots of a similar type (for example, two ground robots with different wheel radii and weights), however, we expect the costs to be positively correlated. In other words, a terrain that is more costly for one robot is also likely to be more costly for another robot. Under this assumption, information may effectively be shared between robots, and the spatial terrain map from one robot can be informative to another robot.

1.2 Background on Off-Road Energy Prediction for Ground Robots

By definition, off-road environments lack predefined roads/routes that constrain vehicles, yet provide better known energy prediction and clearly defined path planning problems. Instead, a robot travelling in an off-road environment may traverse a highly unstructured area with spatial changes in terrain characteristics and topography. The energy consumption of a robot is then highly dependent on both its location and its heading. One way to represent spatially varying quantities (such as terrain) is through spatial modeling techniques [22]. A common tool for spatial modeling and machine learning is Gaussian process regression (GPR), also known as Kriging [84]. GPR is a non-parametric, kernel-based method for interpolation that computes both a mean prediction and an uncertainty of that prediction based on collected data. This is especially useful for making predictions in regions of the environment in which data is sparse. In particular, regions with little information have higher uncertainty. Properly accounting for this uncertainty can lead to more robust planning.

Recent work in ground robotics has looked at applying GPR for spatial mapping of energy costs in an environment. For example, a ground robot was used to spatially map the solar energy distribution over an environment with GPR [74]. The map was then used for energy-efficient path planning. Martin and Corke [56] use a robot to construct an energy map of the environment based on the terrain, and find an energy-minimal tour of the environment. These works do not, however, predict the uncertainty of the cost of a path, and rely on the mean prediction for planning. Oliveira et al. [67] predict the mean and uncertainty of the energy cost of a path through an environment represented by GPR. However, the predictions ignore spatial correlations inherent in such models, leading to overconfidence in those predictions.

Energy prediction and characterization for ground robots is a focus of research beyond spatial mapping as well. Such work considers vehicle models and takes into account a number of factors such as slope, speed, weight, terrain, driving style, etc. [87, 14, 27, 60, 104]. Longitudinal vehicle models consider the rolling resistance due to the wheel-terrain interaction that resists motion [82, 87]. To predict future costs, the rolling resistance is typically assumed to be known [87].

It is unlikely that rolling resistances are known throughout an off-road environment prior to a mission. However, vehicle models can still serve as a useful tool for estimating the rolling resistance and characterizing the terrain [27]. Other methods attempt to classify terrain using proprioceptive sensing [66, 109]. Classification tools can be useful, but do not directly provide mapping or prediction of energy costs.

1.3 Applications of Energy Prediction: Path Planning and Mission Planning

In this section, a brief overview of some of the applications of energy prediction is provided. The focus here is path planning and mission planning, with additional discussion on how uncertainty affects these applications.

Energy and the prediction of thereof is a staple of efficient planning to extend operation time in robotics. For ground robots, it has been applied to the topics of coverage path planning [15, 26, 116] and energy-minimal path planning [104, 62], as a more effective use of resources than an approach based purely on minimum distance travelled. This problem extend beyond ground robotics to, for example, unmanned aerial vehicles and autonomous underwater vehicles, in which wind fields [31] and ocean currents [40, 101, 102], respectively, have been considered for their effect on energy usage and path planning. For all of these applications, there is significant uncertainty in energy costs, particularly with respect to environmental factors and disturbances.

Beyond traditional path planning are higher level robotic mission planning algorithms. Generally speaking, there is no formal mission planning definition in robotics. We refer to robotic mission planning, however, as the problem space in which a robot or robots must complete a set of tasks in an environment, typically constrained by limited resources (energy, time, etc.). A number of inter-related research areas fall under this umbrella, including orienteering problems [119, 107], vehicle routing problems [16, 121], and informative path planning (IPP) [10, 119, 105]. While this dissertation does not focus on the mission planning problem space, we note that it is often necessary to consider a robot's range (or reachability) in an environment in order to safely plan. Furthermore, in real world applications, the usage of resources such as energy or time is uncertain due to the environment.

For both path planning and mission planning, it is common to discretize the environment space and represent it as a graph with deterministic edge costs. A minimum cost path is then found with an algorithm such as Dijkstra's algorithm or A*. For many environments, however, the energy costs are unknown rather than deterministic. One way of handling this uncertainty is to represent edge costs with random variables [24, 21]. The costs in many environments have an additional, less studied factor: spatial correlation. These correlations imply that if a location is high cost, then nearby locations are also more likely to be high cost.

When faced with uncertainty, common approaches to minimum cost path planning are to follow the path with the minimum expected value [74, 64], though ignoring uncertainty could result in overly aggressive planning, risking energy depletion. To address this, some strategies take the approach of generating many random samples of the costs in an environment and observing the result of path planning over these samples [63, 24]. Additional methods focus on path reliability, and attempt to find a path that maximize the probability of achieving the objective (such as minimum time or energy) [93, 73]. However, correlation of uncertain costs (including spatially correlated costs) result in shortest path problems without optimal substructure [93]. This means that the prior path matters when determining the optimal future path. As a consequence, dynamic programming algorithms are not guaranteed to provide optimal solutions.

There is significant research in robotics for handling many types of uncertainty, including state, sensor, process, and environment uncertainty. For example, methods related to stochastic reachability seek to determine a probabilistic safe set within which a robot can operate [1, 19, 38]. Partially observable Markov decision processes (POMDPs) are a common way of representing agent decision-making in an uncertain environment. Unfortunately, POMDPs are often computationally intractable to solve. Randomized sampling-based planners such as rapidly exploring random trees (RRTs) [42] and probabilistic roadmaps (PRMs) [42, 43] are also popular choices for robotic motion planning problems, including in off-road terrains [46, 103]. Variants of these algorithms, like RRT*, provide asymptotic optimality guarantees under certain conditions. Recent RRT variants have considered uncertainty in planning, such as obstacle location and plant uncertainty [50, 51, 5]. These algorithms seek to guarantee safety in terms of probabilistic chance-constraints. To our knowledge, little work has focused directly applying these methods to path planning problems with spatially correlated and uncertain costs. In general, further research is needed to develop computationally efficient and effective methods for both modeling uncertainty and planning under uncertainty.

1.4 Contributions and Dissertation Overview

In this dissertation, we primarily address the following research questions:

1. How can energy costs be accurately predicted in uncertain and complex off-road environments in which costs vary spatially depending on the terrain? (Chapters

II, III, and IV)

2. How can the reachability of a robot with a limited energy budget be determined when energy costs in an environment are uncertain? Furthermore, can reachability predictions be improved by incorporating new data on the environment? (Chapter II)

3. Can information on spatially varying energy costs be effectively shared between heterogeneous robots of a similar type? (Chapter IV)

These questions are addressed in this dissertation through original research presented in Chapters II-IV.

Chapter II provides a new method for computing a robot's energy constrained reachability in an environment with spatially varying energy costs. The method includes a formulation for path energy cost prediction in 2-d environments, with proper characterization of uncertainty through considering spatial correlation in costs. Chance constrained reachable sets (CCRS) are based on the probability that a robot can reach a given location on a graph, given the robot's energy budget, the predicted energy cost, and a user-defined confidence level. A method is provided for computing an under-approximation of the CCRS. Simulations demonstrate that as a robot collects more data on the environment, the True Positive Rate increases and the False Positive Rate decreases, providing a significant performance improvement over the commonly used distance-based energy cost assumption. The work in this Chapter is based on [79, 81].

Chapter III presents an extension of terrain mapping and energy prediction to a 3-d environment, accompanied by extensive experimental results on robotic platforms. The methodology considers spatial mapping, as well as additional factors, including vehicle modeling and satellite imagery, to predict energy costs along a path, and the uncertainty in those costs. To evaluate the methodology, experimental testing is performed in an outdoor environment with varying terrains. The results demonstrated significantly improved path energy prediction accuracy and uncertainty quantification for both data-driven and vehicle modeling approaches, as compared to a baseline approach. In addition, we observed the results of predictions in which the robot has little data on the environment, and there is high predictive uncertainty. In general, the uncertainty quantification in such a case is good, however, the existence of outliers in the data can still result in overconfident predictions. The results presented in Chapter III are based on work in [80, 77].

Chapter IV provides a framework for sharing spatial terrain energy cost information between multiple heterogeneous robots. The framework, based on multi-task Gaussian process regression (MTGP) [12], learns the scaling and correlation between heterogeneous robot power consumptions. Further, it applies strategies for computationally efficient predictions and for multi-task hyperparameter optimization in ways that are effective for the problem of multi-robot power prediction. Simulations with several robots show the scalability of this framework. Furthermore, experimental results with a small and a large robot platform demonstrate the effectiveness of MTGP predictions for improving the accuracy of power predictions, even when the robots are quite different in size. The results presented in Chapter IV are based on work in [78].

In Chapter V we conclude the dissertation and provide several directions for future work based on the research presented here. Additionally, further experimental results related to Chapter III are provided in Appendices A and B, and a derivation related to Chapter IV is shown in Appendix C.

To summarize, the main contributions of this dissertation are:

1. A method for computing a robot's energy constrained reachable set in an environment with uncertain and spatially varying costs.
2. A spatial mapping strategy for predicting path energy costs in 3-D off-road

environments, with computation of the uncertainty in those predictions through correlated costs.

3. An extension to the spatial mapping and prediction strategy for the case of multiple heterogeneous ground robots.

Furthermore, this research has resulted in 2 conference papers, [79, 80], a journal paper published in *Robotics and Autonomous Systems* [81], as well as a submitted journal paper [77], and a journal paper in preparation [78].

CHAPTER II

Chance Constrained Reachability in Environments with Spatially Varying Energy Costs

This chapter presents the initial 2-d spatial mapping and energy prediction formulation and applies it to the problem of determining a robot's chance constrained reachable set (CCRS). Spatial correlation in costs is emphasized as an important component in predicting energy costs with uncertainty. The work presented in this chapter was originally published in [81], building off preliminary work presented in [79].

As first discussed in the Introduction of this dissertation, the range of an autonomous robot in an environment is constrained by both internal factors (energy capacity, speed, etc.) and energy costs due to the environment. One important problem for robust mission planning is to determine where in the environment a robot can reach, given its energy budget. In an environment with varying energy costs, the spatial distribution of such costs is critical for computing reachability. The main challenge is that energy costs are often uncertain, with camera data, satellite imagery, and soil maps providing only some indication of costs. However, in situ energy measurements can be used to more accurately model spatially varying energy costs. This chapter presents a method for computing a CCRS by spatially mapping energy costs. The CCRS defines a set of locations that meet (within a desired probability) the design constraint of returning back to a base within the energy budget of the robot.

Reachability is important in a variety of contexts, including: energy-limited informative or coverage path planning [10, 96], computing the set of reachable locations for a hot-air balloon in wind fields [33, 45], and dynamic obstacle avoidance [53]. Figure 2.1 visually depicts a reachability scenario for an energy constained robot. Reachability can be used to define an allowable search space for informative path planning algorithms [10, 9], or to quantify trade-offs between energy storage capacity and other relevant factors (speed, sensing capabilities, etc.). Typically, costs are assumed to be known [10, 96, 33, 45], allowing reachability to be computed using standard path planning methods, such as Dijkstra's algorithm [25]. In realistic scenarios, however, the mapping between the environment and the energy costs for a particular robot is uncertain.

To predict energy costs and characterize uncertainty, we leverage a strategy for energy cost prediction presented in our recent work [79, 80]. A spatial mapping from position to energy, referred to as an *energy map*, is built with measured data using Gaussian process regression (GPR) [84]. The mapping is used to compute the mean and variance of the energy cost of a particular path through the environment.

To determine a CCRS, paths for a robot to traverse between locations, along with the feasibility of those paths as a function of the predicted energy cost versus the remaining energy of the robot, must be identified. Energy costs of paths through the environment are *spatially correlated* under the strategy in [79, 80]. Spatial correlation of costs are key to characterizing uncertainty and therefore reachability within an environment. While there are several examples of path planning methods in the literature that incorporate uncertainty and correlated costs [24, 47, 65, 93, 120], these current methods are computationally intensive, making them impractical for determining reachability. In our approach, we identify the minimum expected cost path, which is readily computed with Dijkstra's algorithm. Uncertainty and spatially correlated costs are considered through checking the feasibility of a chance constraint on the energy cost of the minimum expected cost path.

One significant challenge in predicting energy costs is that the accuracy of predictions and the quantification of uncertainty depend on the structure of the environment. There are key questions that must be addressed, such as: (1) What is the distribution of energy costs for a robot in a given environment? (2) How do energy costs vary spatially (e.g. smoothly or discontinuously) and subsequently impact reachability? Through the use of GPR, the impact of changes in the environment and uncertainty in predictions can be captured by selection of the kernel and the prior mean [63].

To demonstrate our method, an information gathering scenario is considered in simulation. A robot, constrained by its limited energy capacity to return to a recharging base, measures energy costs in the environment and uses that information to compute the CCRS. We show the flexibility of our CCRS method through results on two different types of example environments: one with smoothly varying energy costs and one with discontinuous changes in the environment that lead to abrupt changes in the energy costs.

2.0.1 Contributions

The main contributions in this chapter include:

1. A method for computing a CCRS in environments with uncertain and spatially correlated energy costs.
2. A demonstration of the improved prediction of a CCRS over time as a robot collects information to build the energy map. Simulation results show that the method achieves significantly higher true-positive rates while maintaining low false-positive rates for reachable locations.

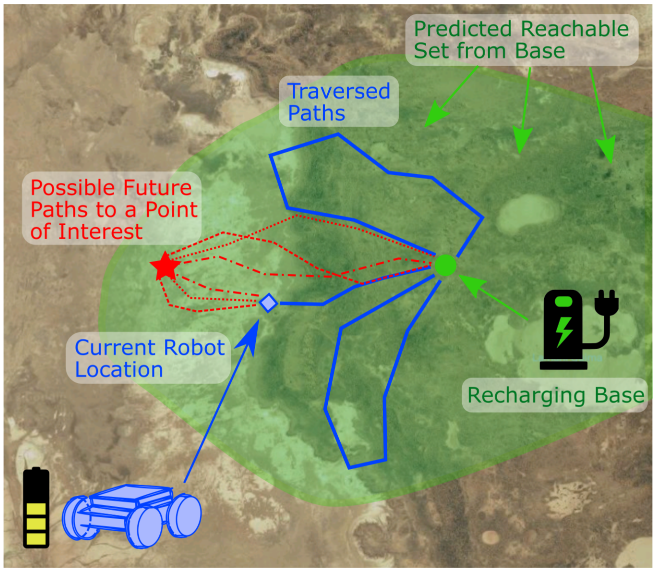


Figure 2.1: A scenario in a which a ground robot explores an off-road environment with terrains of varying energy cost. The robot (blue diamond) must determine whether the location of interest (red star) is reachable and return to the recharging base (green circle) without depleting its energy resources. Previously traversed paths (solid blue line) provide sparse information on the energy cost of traversing the terrain that can be used to predict the energy cost of future paths (red dotted/dashed lines).

3. An exploration of how varying kernel hyperparameters affects the predicted chance constrained reachability.

This research builds on the authors' previous work in energy mapping and probabilistic energy cost prediction presented in [79, 80] with the important advancement of developing of a new method for computing a chance constrained reachable set based on data collected during a mission, along with a demonstration of building the reachable sets through two simulation case studies.

The remainder of the chapter proceeds as follows: Section 2.1 overviews related work. Section 2.2 provides the strategy for predicting energy costs based on collected data using GPR. Section 2.3 defines the CCRS and describes our method for computing it. Section 2.4 presents simulation results for a robot information-gathering scenario in two types of environments. Section 2.5 gives concluding remarks and future directions.

2.1 Related Work

Reachability is considered in the context of mission planning when robotic systems have constraints on resources such as energy or time. For example, Kuwata et al. determine the set of reachable locations for a hot air balloon in a time-varying wind field, as well as how long it would take to reach such locations [33, 45]. One important application of reachability is Informative Path Planning (IPP) [10, 9, 11, 37, 95, 119]. The objective of IPP is for a robot to maximize information gain on an environment, subject to an energy/cost constraint (typically in terms of distance travelled by the robot). Unfortunately, many IPP algorithms are computationally costly, especially as the search space of potential locations increases. Binney et al. [10] and Best et al. [9] use reachability to provide cost-constrained upper bounds on the search space, reducing computation time. These works assume edge costs on a graph are known, though in many applications, the costs are uncertain.

The concept of reachability has appeared frequently in control systems and robotics research, often with the objective of providing safety guarantees against disturbances and uncertainties [106]. For example, Akametalu et al. [3] learn system dynamics with a Gaussian process and perform reachability analysis to determine safe sets. Senarathne and Wang [92] find reachable frontiers on occupancy grids for use in robot exploration. A related area that considers uncertainty is stochastic reachability. Malone et al. [53] use stochastic reachable sets for obstacle avoidance when the future location of the obstacle is uncertain. Stochastic reachability typically relies on dynamic programming methods which become infeasible with an increasing number of states [32]. Subramani et al. [101] performed energy-optimal path planning in known flow fields through time-optimal reachability fronts, and furthermore considered uncertain flow fields [102]. The work on time-optimal reachability and stochastic reachability is often defined with respect to the dynamics of the underlying system. While our use of reachability has conceptual similarities, we do not explicitly consider robot dynamics and instead define a graph-based reachability with respect to the robot's energy budget, similar to Binney et al. [10].

When computing reachability, it is important to predict future energy costs for the robot. For ground robots, Sadrpour et al. [87] use a vehicle model and information on terrain rolling resistance, road grade (slope), and tele-operative user aggressiveness to predict future energy usage with a Bayesian method. Dogru and Marques [27] use power consumption data from skid-steered ground robot data for low power coverage path planning. Pentzer et al. [70] considered energy minimal path planning for a ground robot in varying terrains. These works focus on energy usage factors particular to the robot and assume the energy costs due to the environment (i.e. the terrain) are known. We instead focus on determining the effects of environment energy factors through spatial mapping.

Ground robots have been used for spatial mapping of energy and traversability with respect to terrain information [56] and solar radiation maps [74]. Plonski et al. [74] use a solar powered robot to build a solar map with GPR, which is then used for energy-efficient path planning with dynamic programming. Martin and Corke [56] use a robot to construct an energy map of the environment with GPR, and find an energy-minimal tour of the environment. References [56] and [74] only use the predictive mean of GPR and not the variance, leading to overconfidence in predictions for areas of the environment for which there is little information. Murphy et al. [63] use a robot to build terrain traversibility maps and perform path planning with probabilistic costmaps. While spatial energy maps have been successfully used for energy minimal path planning, further research is necessary to fully make use of the uncertainty characterization provided by GPR.

Path planning is important for determining energy constrained reachability on graphs. For deterministic edge costs, standard shortest path algorithms such as Dijkstra's algorithm [25] may be used. Planning under uncertainty is more challenging, with interesting cases such as finding the shortest expected path when there is a probability that edges are not traversable [108, 64]. We consider the case in which edges are known to be traversable, however, the edge costs are uncertain. For example, Chung et al. [21] provide a graph search algorithm for risk-aware planning in which edge costs are represented by normal distributions. The algorithm finds paths with a high probability of low cost. Interestingly, even a naive approach of considering only the mean path cost has demonstrated reasonable performance in the risk based environment, as well as low computation time [21]. However, these works did not consider spatial correlation in the cost of paths; an important consideration for characterizing uncertainty. Partially-observable Markov decision processes (POMDPs) have also been used as frameworks for planning because of their expressive accounting of uncertainty. POMDPs tend to suffer computationally from exponential growth, though recent work has been addressing this problem [7, 2].

2.2 Spatial Energy Prediction

This section provides a background on GPR as it is used in this work for spatial energy mapping of an environment based on collected data. The energy cost of a path through the environment is then computed probabilistically by integrating along the path.

2.2.1 Spatial Energy Mapping

The following assumptions are made for modeling the spatial energy cost distribution of the environment:

Assumption II.1. *The environment is flat and static.*

Assumption II.2. *Only energy costs associated with traversing the environment are considered.*

$\mathbf{x}$	Location in $\mathbb{R}^2$
y	Measurement of energy density
$\mathcal{D}$	Data set, $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$ of n points
e	Spatial energy distribution function modelled by data $\mathcal{D}$ using GPR
$k(\mathbf{x}_i, \mathbf{x}_j)$	Kernel used to define the covariance matrix for GPR
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Graph with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$
$v_s \in \mathcal{V}$	Starting vertex of the robot
$b \in \mathcal{V}$	Recharging base vertex
$\mathcal{P}(v_s, v, b)$	Path on $\mathcal{G}$ from vertices v_s to v to b
$E_{\mathcal{P}} \sim \mathcal{N}(\mu_{\mathcal{P}}, \sigma_{\mathcal{P}}^2)$	Gaussian random variable associated with the energy cost of path $\mathcal{P}$ with mean $\mu_{\mathcal{P}}$ and variance $\sigma_{\mathcal{P}}^2$
a	Robot's remaining energy budget
β	User-defined confidence level
$\mathcal{R}_\beta(v_s, b)$	CCRS with starting vertex v_s and recharging base b at confidence level β
$w(\epsilon_i)$	Edge weight for the energy cost of traversing edge $\epsilon_i \in \mathcal{E}$
$\mathcal{R}_w(v_s, b)$	Deterministic reachable set under edge weight function w

Table 2.1: Frequently Used Notation

These assumptions are made so that the initial focus of this research is on the mathematical development of energy predictions and reachability analysis. The work presented in Chapter III relaxes the assumption of the flatness of the environment.

The spatial energy distribution of the environment is predicted based on energy usage measurements from robots in terms of energy per unit distance. To model and predict energy usage, the strategy used here is adapted from [79], which spatially modelled the energy usage of the environment with Gaussian process regression using point data collected by robots.

Consider an environment $Q \subset \mathbb{R}^2$ where a robot takes measurements of energy per distance travelled $y \in \mathbb{R}$ at location $\mathbf{x} \in Q$. For example, y could be based on the robot's speed and current draw from the batteries at a location $\mathbf{x}$. The data is collected as $\mathcal{D} = \{(y_i, \mathbf{x}_i)\}_{i=1}^n$ and denoted $\mathbf{y} \in \mathbb{R}^n$ and $X \in \mathbb{R}^{2\times n}$ for the collected y_i and $\mathbf{x}_i$, respectively.

The spatial energy distribution of the environment is given by an unknown ground truth function $e_{true} : Q \to \mathbb{R}$. The function e_{true} is an isotropic energy density and describes the energy cost per distance travelled at a given location. The true energy function is modelled based on data $\mathcal{D}$ using a Gaussian process (GP), denoted $e \sim \mathcal{GP}(m, k)$. The GP is characterized by a prior mean function $m : Q \to \mathbb{R}$ and a kernel $k : Q \times Q \to \mathbb{R}$, which is a symmetric function that computes the correlation between two points in Q. A positive definite covariance matrix is defined by the kernel: $[K]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The noisy model for regression is then given by:

$$\begin{aligned} y &= e(\mathbf{x}) + \eta \\ \eta &\sim \mathcal{N}(0, \sigma_\eta^2) \\ e &\sim \mathcal{GP}(m, k) \end{aligned} \tag{2.1}$$

where $\sigma_\eta^2 > 0$ is known.

GPR defines an n-variate normal distribution based on m and k (see [84] for further information on GPR). The posterior predictive distribution computes the distribution of $e(\mathbf{x}_*)$ at a target point, $\mathbf{x}_* \in Q$ based on the data. The probability density function for $e(\mathbf{x}_*)|\mathcal{D}$ is known to be [84]:

$$\begin{aligned} p(e(\mathbf{x}_*)|\mathcal{D}) &= \int p(e(\mathbf{x}_*)|e)p(e|\mathcal{D})de \\ &= \mathcal{N}\big(\bar{e}(\mathbf{x}_*), \operatorname{cov}(e(\mathbf{x}_*), e(\mathbf{x}_*))\big) \end{aligned} \tag{2.2}$$

where

$$\bar{e}(\mathbf{x}_*) = m(\mathbf{x}_*) + K(\mathbf{x}_*, X)[K(X, X) + \sigma_n^2 I]^{-1}(\mathbf{y} - m(X)) \tag{2.3}$$

is the posterior (or predictive) mean at $\mathbf{x}_*$ and

$$\begin{aligned} \operatorname{cov}(e(\mathbf{x}_*), e(\mathbf{x}'_*)) = K(\mathbf{x}_*, \mathbf{x}'_*) \\ - K(\mathbf{x}_*, X)\big[K(X, X) + \sigma_n^2 I\big]^{-1} K(X, \mathbf{x}'_*) \end{aligned} \tag{2.4}$$

is the covariance where $\operatorname{cov}(e(\mathbf{x}_*), e(\mathbf{x}_*))$ is the predictive variance at $\mathbf{x}_*$.

2.2.2 Energy Prediction Along a Path

The goal of this section is to predict the energy cost for a robot moving on a straight line at a constant velocity between two points. The strategy used is adapted from [79], though it is extended here through a more thorough mathematical formulation presented in Proposition II.3. To predict the energy cost, we integrate along a path parameterized by the piecewise linear curve $\Gamma \subset Q$ over the Gaussian process (this is sometimes called Bayesian Quadrature [35]). The integration produces a Gaussian random variable that describes energy cost of the path. Integrating along Γ, with commonly used notation for line integrals:

$$E = \int\limits_{\Gamma} e(\mathbf{x}_p) ds \tag{2.5}$$

where $\mathbf{x}_p : [a, b] \to \Gamma$, $a < b$, is the parameterization of a piecewise linear curve with endpoints $\mathbf{x}_p(a)$ and $\mathbf{x}_p(b)$. While other curves are allowable in this formulation, linear paths are considered in this dissertation to define edges on a graph.

Computing the mean, denoted μ, and variance, σ^2, of the path energy cost E is performed by averaging over the GP, as shown in Eqns. (2.8) and (2.9).

Proposition II.3. *The energy cost of a path has the probability density* $p(E|\mathcal{D}) = \mathcal{N}(\mu, \sigma^2)$ *where:*

$$\mu = M + K_{sx}(K_{xx} + \sigma_\eta^2 I)^{-1}(\boldsymbol{y} - m(X)) \tag{2.6}$$

$$\sigma^2 = K_{ss} - K_{sx}(K_{xx} + \sigma_\eta^2 I)K_{xs} \tag{2.7}$$

where $M = \int_\Gamma m(\boldsymbol{x}_p)ds$, $K_{xx} = k(X, X)$, $K_{xs} = \int_\Gamma k(X, \boldsymbol{x}'_p)ds'$, $K_{sx} = \int_\Gamma k(\boldsymbol{x}_p, X)ds$, *and* $K_{ss} = \int_\Gamma \int_{\Gamma'} k(\boldsymbol{x}_p, \boldsymbol{x}'_p)dsds'$.

Proof. The generalized proof of this result is shown in [13]. We show the case of integrating along a path over a Gaussian process prior. The notation $\mu := \mathbb{E}_{e|\mathcal{D}}[E]$ indicates that the expected value is computed by averaging over the distribution of e given the data $\mathcal{D}$. Then by substitution and Fubini's Theorem:

$$\begin{aligned}
\mathbb{E}_{e|\mathcal{D}}[E] &= \mathbb{E}_{e|\mathcal{D}}[\int\limits_{\Gamma} e(\mathbf{x}_p)ds] \\
&= \int \int\limits_{\Gamma} e(\mathbf{x}_p)p(e|\mathcal{D})dsde \\
&= \int\limits_{\Gamma} \left(\int e(\mathbf{x}_p)p(e|\mathcal{D})de \right) ds \\
&= \int\limits_{\Gamma} \bar{e}(\mathbf{x}_p)ds
\end{aligned} \tag{2.8}$$

The result follows from substituting for $\bar{e}(\mathbf{x}_p)$ in Eqn. (2.3). The variance, $\sigma^2 := \text{cov}_{e|\mathcal{D}}[E, E]$, is calculated similarly by considering the covariance between two paths, E and E':

$$\begin{aligned}
\text{cov}_{e|\mathcal{D}}[E, E'] &= \mathbb{E}_{e|\mathcal{D}}[(E - \mathbb{E}_{e|\mathcal{D}}[E])(E' - \mathbb{E}_{e|\mathcal{D}}[E'])] \\
&= \int (E - \mathbb{E}_{e|\mathcal{D}}[E]) \\
&\qquad (E' - \mathbb{E}_{e|\mathcal{D}}[E'])p(e|\mathcal{D})de \\
&= \int\limits_{\Gamma} \int\limits_{\Gamma'} \text{cov}(e(\mathbf{x}_p), e(\mathbf{x}'_p))dsds'
\end{aligned} \tag{2.9}$$

The result follows from substitution with Eqn. (2.4). □

The strategy presented here takes point data over the spatial energy distribution and predicts the energy cost of a path. The formulation is easily extensible to predicting the cost of a path based on previously measured costs of alternative paths. Path-to-path prediction would be useful in scenarios in which direct measurements of power consumption are unavailable, but the change in fuel is an accessible quantity.

2.3 Energy-Constrained Reachability

In this section, we consider how to compute the locations in the environment that can be reached, based predicted energy costs, the location of a recharging base in the environment, and the energy capacity of the robot. Two types of reachable sets are defined: deterministic and chance constrained.

2.3.1 Graph Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected, undirected graph with vertex set, $\mathcal{V}$, and edge set, $\mathcal{E}$. Each $v \in \mathcal{V}$ corresponds to a location $\mathbf{x} \in Q$. Let $\mathcal{P} \subset \mathcal{E}$ be a path through $\mathcal{G}$.

For computing the reachable sets, we associate both an edge weight function, $w(\epsilon_i)$, and a random variable, E_{ϵ_i}, with every edge $\epsilon_i \in \mathcal{E}$. Let $w : \mathcal{E} \to \mathbb{R}_{\geq 0}$ be the edge weight function representing a deterministic energy cost of traversing an edge. The cost of a path under edge weight function w is defined as $C_w(\mathcal{P}) := \sum_{\epsilon_i \in \mathcal{P}} w(\epsilon_i)$.

Let $E_{\epsilon_i} \sim \mathcal{N}(\mu_{\epsilon_i}, \sigma^2_{\epsilon_i})$ be the random variable associated with the *predicted* energy cost of traversing edge ϵ_i. The mean, μ_{ϵ_i}, and variance, $\sigma^2_{\epsilon_i}$, are computed with Eqns. (2.6) and (2.7), respectively. GPR and the subsequent path predictions are normal distributions with infinite support, making negative energy predictions possible. While such values may lack physical meaning, normal distributions are often used in stochastic planning problems [120, 21] because of their analytical advantages and utility as a reasonable approximation of real-world data [47]. We do not assume that the predicted edge costs are independent and instead treat them as correlated. The predicted energy cost of the robot's path, denoted $E_{\mathcal{P}}$, is computed as the summation of Gaussian random variables associated with the l edges of a path:

$$E_{\mathcal{P}} := \sum_{i=1}^{l} E_{\epsilon_i} \tag{2.10}$$

where, by the summation of Gaussians, $E_{\mathcal{P}} \sim \mathcal{N}(\mu_{\mathcal{P}}, \sigma^2_{\mathcal{P}})$, such that

$$\mu_{\mathcal{P}} = \sum_{i=1}^{l} \mu_{\epsilon_i}, \tag{2.11}$$

$$\sigma^2_{\mathcal{P}} = \sum_{i=1}^{l} \sigma^2_{\epsilon_i} + 2 \sum_{1 \le i < j \le l} \mathrm{cov}_{e|\mathcal{D}}[E_{\epsilon_i}, E_{\epsilon_j}]. \tag{2.12}$$

The covariance is computed as derived in Eqn. (2.9). The computation of the path variance involves finding the covariance between every edge in the path, reflecting the fact that edge energy costs are correlated. Ignoring the correlation in costs between paths would result in significant overconfidence in the prediction.

2.3.2 Reachable Set Definitions

We define both deterministic and chance constrained reachable sets based on whether a robot starting at $v_s \in \mathcal{V}$ can reach a node, $v \in \mathcal{V}$, and return back to a recharging base, $b \in \mathcal{V}$, within the robot's remaining energy budget, $a \ge 0$. Paths from v_s to v and v to b are denoted $\mathcal{P}(v_s, v)$ and $\mathcal{P}(v, b)$ respectively. For conciseness, let $\mathcal{P}(v_s, v, b)$ denote the concatenation of the paths $\mathcal{P}(v_s, v)$ and $\mathcal{P}(v, b)$.

First, let the *deterministic reachable set*, $\mathcal{R}_w(v_s, b) \subset \mathcal{V}$, be the set of vertices $v \in \mathcal{V}$ for which there exists the path $\mathcal{P}(v_s, v, b)$ on $\mathcal{G}$ satisfying

$$C_w(\mathcal{P}(v_s, v, b)) \le a. \tag{2.13}$$

Given the edge weights, computing this set is the simple matter of using Dijkstra's shortest path algorithm to determine the optimal paths $\mathcal{P}(v_s, v)$ and $\mathcal{P}(v, b)$.

Clearly, $\mathcal{R}_w(v_s, b)$ depends on the edge weights, which we define in three ways:

1. If the true energy distribution, e_{true}, is known, then $w_{true}(\epsilon_i) := \int_\Gamma e_{true}(\mathbf{x}_p) ds$, where Γ and $\mathbf{x}_p$ are defined in Eqn. (2.5).

2. Assuming a constant energy cost per unit distance, $c > 0$, then $w_d := cd_{\epsilon_i}$, where d_{ϵ_i} is the length of a given edge ϵ_i.
3. The mean predicted energy cost of a path due to the energy mapping is given as $w_{exp}(\epsilon_i) := \mu_{\epsilon_i}$.

Second, the *chance constrained reachable set* (CCRS), $\mathcal{R}_\beta(v_s, b)$, is defined based on the probability of running out of energy. Let $\mathcal{R}_\beta(v_s, b) \subset \mathcal{V}$ be the set of vertices $v \in \mathcal{V}$ for which there exists a path $\mathcal{P}(v_s, v, b)$ on $\mathcal{G}$ satisfying

$$\text{Prob}(E_{\mathcal{P}(v_s,v,b)} \leq a|\mathcal{D}) \geq \beta \tag{2.14}$$

where $0 \leq \beta \leq 1$ is a user-defined confidence level. $E_{\mathcal{P}(v_s,v,b)} \sim \mathcal{N}(\mu_{\mathcal{P}(v_s,v,b)}, \sigma^2_{\mathcal{P}(v_s,v,b)})$ is a normally distributed random variable with mean path cost and variance that are computed with Eqns. (2.11) and (2.12). Then

$$\text{Prob}(E_{\mathcal{P}(v_s,v,b)} \leq a|\mathcal{D}) = \Phi\Big(\frac{a - \mu_{\mathcal{P}(v_s,v,b)}}{\sigma_{\mathcal{P}(v_s,v,b)}}\Big) \tag{2.15}$$

where Φ is the cumulative distribution function for a normal distribution. The chance constraint is equivalent to the following deterministic constraint:

$$\Phi\Big(\frac{a - \mu_{\mathcal{P}(v_s,v,b)}}{\sigma_{\mathcal{P}(v_s,v,b)}}\Big) = \frac{1}{2}\Big[1 + \text{erf}\Big(\frac{a - \mu_{\mathcal{P}(v_s,v,b)}}{\sqrt{2}\sigma_{\mathcal{P}(v_s,v,b)}}\Big)\Big] \geq \beta \tag{2.16}$$

where erf is the error function. Equation (2.16) can be used to easily check the feasibility of a path at the desired confidence level. While chance constraints are often defined in the context of an optimization problem, chance constraints have also been employed in robotics for other purposes, such as collision checking under uncertainty [28]. We use the chance constraint in Eqn. (2.14) to check whether a path is feasible, given the predicted energy cost, the remaining energy, and the desired confidence level.

2.3.3 Reachable Set Computation

The method for computing the CCRS is illustrated in Algorithm 1. To summarize:

1. Edge weights, $w_{exp}(\epsilon_i) = \mu_{\epsilon_i}$, are updated based on collected data.
2. The minimum expected cost path,

   $$\overline{\mathcal{P}}(v_s, v, b) := \underset{\mathcal{P}(v_s,v,b)}{\text{argmin}}\ \mu_{\mathcal{P}(v_s,v,b)}, \tag{2.17}$$

   is found for every v by building two shortest path trees using Dijkstra's algorithm [25]. The trees originate from v_s and b and connect to every v. $\overline{\mathcal{P}}(v_s, v, b)$ is the concatenation of the paths, $\overline{\mathcal{P}}(v_s, v)$ and $\overline{\mathcal{P}}(v, b)$, from each tree.
3. The feasibility of $\overline{\mathcal{P}}(v_s, v, b)$ is checked with Eqn. (2.16) for each v.

To use Dijkstra's algorithm, it must be true that $w_{exp}(\epsilon_i) \geq 0$ and therefore $\mu_{\epsilon_i} \geq 0$. In practice, non-negativity can be enforced with $\mu_{\epsilon_i} \leftarrow \max(\mu_{\epsilon_i}, 0)$. Minimum expected cost paths are computed efficiently by Dijkstra's algorithm, and are good paths to check for feasibility. However, these paths are planned without information on uncertainty. The computed uncertainty, including spatial correlations, is taken into account by checking the feasibility of the constraint in Eqn. (2.16).

Path planning that considers uncertainty could, for example, maximize the probability that the predicted energy cost is less than the robot's remaining energy. Maximum probability and related problems of path planning under uncertainty are the subject of ongoing research [47, 65]. Our case, in which edge costs are correlated, is particularly challenging because edge costs are non-linear and non-additive. As a consequence, the problem lacks the sub-path optimality property, meaning that both dynamic programming and Dijkstra's algorithm do not provide a solution. Various methods have been developed for this problem, see [24, 93, 120], and could be used instead of minimum expected cost paths, though at significantly greater computational

Algorithm 1: CCRS Computation

```
Input  : $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: graph of locations
         $\mathcal{D}$: collected data
         $\beta$: confidence level
         $a$: remaining energy budget
Output: $\mathcal{R}_\beta(v_s, b)$: CCRS for a given $\beta$
for $\epsilon_i \in \mathcal{E}$ do
    $w_{exp}(\epsilon_i) = \mu_{\epsilon_i}$ % update edge weights with the mean predicted energy cost with data $\mathcal{D}$
% Compute trees of shortest paths from $b$ and $v_s$ with Dijkstra's algorithm
$\mathcal{T}_{v_s} \leftarrow$ ShortestPathTree $(\mathcal{G}, w_{exp}, v_s)$
$\mathcal{T}_b \leftarrow$ ShortestPathTree $(\mathcal{G}, w_{exp}, b)$
for $v \in \mathcal{V}$ do
    % Get branches from $v_s$ to $v$ and $v$ to $b$
    $\overline{\mathcal{P}}(v_s, v) \leftarrow$ Branch $(\mathcal{T}_{v_s}, v)$
    $\overline{\mathcal{P}}(v, b) \leftarrow$ Branch $(\mathcal{T}_b, v)$
    $\overline{\mathcal{P}}(v_s, v, b) =$ Concatenate$(\overline{\mathcal{P}}(v_s, v), \overline{\mathcal{P}}(v, b))$
    Compute $\mu_{\overline{\mathcal{P}}(v_s,v,b)}$ and $\sigma^2_{\overline{\mathcal{P}}(v_s,v,b)}$ with Eqns. (2.11) and (2.12).
    if CheckFeasibility $(\mu_{\overline{\mathcal{P}}(v_s,v,b)}, \sigma^2_{\overline{\mathcal{P}}(v_s,v,b)}, a, \beta)$ then
        $\mathcal{R}_\beta(v_s, b) \leftarrow \mathcal{R}_\beta(v_s, b) \cup v$ % Include vertex in $\mathcal{R}_\beta(v_s, b)$ if Eqn. (2.16) is satisfied.
return $\mathcal{R}_\beta(v_s, b)$
```

cost.

We now note relevant facts on the use of minimum expected cost paths over maximum probability paths. The minimum expected cost path is defined by Eqn. (2.17). Let the maximum probability path be defined as

$$\mathcal{P}^*(v_s, v, b) := \underset{\mathcal{P}(v_s,v,b)}{\operatorname{argmax}} \operatorname{Prob}(E_{\mathcal{P}(v_s,v,b)} \leq a | \mathcal{D}) \tag{2.18}$$

Note that the costs of both paths are normal distributions with $E_{\mathcal{P}^*(v_s,v,b)} \sim \mathcal{N}(\mu_{\mathcal{P}^*}, \sigma^2_{\mathcal{P}^*})$ and $E_{\overline{\mathcal{P}}(v_s,v,b)} \sim \mathcal{N}(\mu_{\overline{\mathcal{P}}}, \sigma^2_{\overline{\mathcal{P}}})$, where the means and variances can be computed with Eqns. (2.11) and (2.12).

Lemma II.4. *Let $\overline{\mathcal{R}}_\beta(v_s, b)$ and $\mathcal{R}^*_\beta(v_s, b)$ be computed with minimum expected cost and maximum probability paths, respectively. Then $\overline{\mathcal{R}}_\beta(v_s, b) \subseteq \mathcal{R}^*_\beta(v_s, b)$.*

Proof. By contradiction: Let $\overline{\mathcal{R}}_\beta(v_s, b) \not\subseteq \mathcal{R}^*_\beta(v_s, b)$. Then there exists a v such that $v \in \overline{\mathcal{R}}_\beta(v_s, b)$ but $v \notin \mathcal{R}^*_\beta(v_s, b)$, implying that $\text{Prob}(E_{\mathcal{P}^*(v_s,v,b)} \leq a|\mathcal{D}) < \beta \leq \text{Prob}(E_{\overline{\mathcal{P}}(v_s,v,b)} \leq a|\mathcal{D})$. This is a contradiction since, by definition, $\text{Prob}(E_{\mathcal{P}^*(v_s,v,b)} \leq a|\mathcal{D}) \geq \text{Prob}(E_{\overline{\mathcal{P}}(v_s,v,b)} \leq a|\mathcal{D})$. □

Lemma II.4 states, in other words, that the CCRS computed with minimum expected cost paths, as in Algorithm 1 is an under-approximation.

When comparing the minimum expected cost and maximum probability paths, interesting properties arise:

Lemma II.5. *If* $\mu_{\overline{\mathcal{P}}} < a$ *then* $0 \leq \sigma^2_{\mathcal{P}^*} \leq \sigma^2_{\overline{\mathcal{P}}}$.

Lemma II.6. *If* $\mu_{\overline{\mathcal{P}}} > a$ *then* $\sigma^2_{\overline{\mathcal{P}}} \leq \sigma^2_{\mathcal{P}^*}$.

These follow directly from the fact that since $\overline{\mathcal{P}}$ is the minimum expected cost path, then $\mu_{\overline{\mathcal{P}}} \leq \mu_{\mathcal{P}^*}$. Similar results are shown in [18, 120]. Lemma II.6 indicates that if the mean of the minimum expected cost path is higher than the energy budget, then only a path with higher uncertainty could increase the probability of being within the energy budget. These results are useful in that if bounds can be found on the variance of the optimal path, then the under-approximation by the minimum expected reachable set is also bounded.

2.3.4 Complexity

For Algorithm 1, the sources of increasing computational complexity are from shortest path planning, updating the edge weights of the graph, and checking the feasibility of the energy constraint. For computing the shortest path tree, an efficient variant of Dijkstra's algorithm has a worst case time complexity of $O(|\mathcal{E}|+|\mathcal{V}|\log|\mathcal{V}|)$. Recalculating the edge weights for energy predictions scales with the number of data points, n, and edges by $O(n^3|\mathcal{E}|)$. Checking the feasibility of the chance constraint has a time complexity of $O(n^3|\mathcal{V}|)$. A major source of the computational burden comes from the energy prediction using GPR, which involves integration and matrix inversion that grows with the number of data points (resulting in the n^3 from the previous statements). Significant research into managing the complexity of GPR has been performed and numerous strategies exist for dealing with increasing data, including: truncating or placing data into bins [56, 118], sparse GPs [97], or using an informative subset of the data [36]. For simulations, we use a simple strategy of only updating edges which are well correlated with new data.

2.4 Simulation Demonstration

In this section, the CCRS method is evaluated through simulation results of a robot information gathering scenario. The robot, which is constrained by its energy budget to return to a recharging base, collects data on the spatially varying energy costs of simulated environments. The collected data is used to update the reachable set of locations from the recharging base. Two types of environments are considered: one with smoothly varying energy costs, as seen in Section 2.4.2, and an environment with obstacles and discontinuous cost changes, provided in Section 2.4.3.

The reachability plots (e.g. Fig. 2.2) show the results of computing the set of locations that can be traversed to (using the base as the initial location of the robot), while returning to the base within the energy budget with a confidence level of at least β. Consider the case where we want to identify the set of locations that are reachable within a 99% confidence level ($\beta = 0.99$). The grey dotted or dashed lines labelled 0.99 indicate a boundary that captures this set of locations (in our nomenclature, this set is the CCRS $R_\beta(b, b)$ for $\beta = 0.99$). As the confidence level is relaxed ($\beta = 0.5, 0.01$), the boundary lines that denote these regions are pushed further away from the initial location.

The boundary for the true reachable set $\mathcal{R}_{w_{true}}(b, b)$ is expected to lie within the grey lines representing the CCRS boundaries for $\beta = 0.01$ and $\beta = 0.99$. Outside of

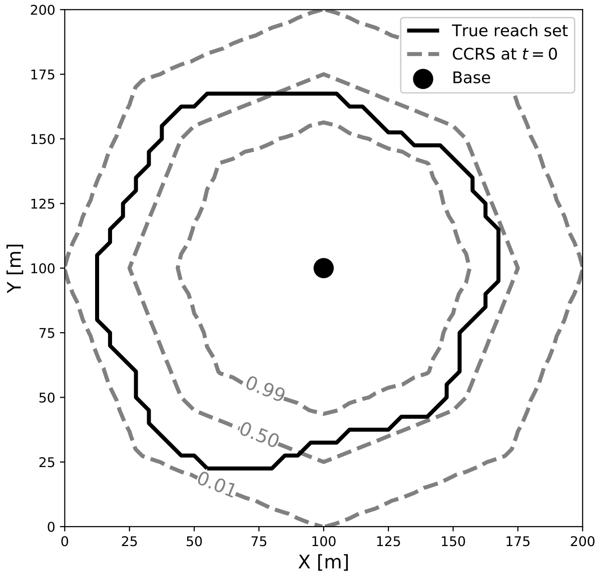


Figure 2.2: CCRS boundaries, $\mathcal{R}_\beta(b, b)$ for $\beta = 0.01, 0.50, 0.99$. The true reachable set, $\mathcal{R}_{w_{true}}(b, b)$, from the recharging base (black dot) lies within the black line. Note: the octonogonal shape of the sets is due to the discretization of the environment into an 8-connected grid.

the $\beta = 0.01$ boundary, there is a probability < 0.01 that the locations are reachable, whereas inside the $\beta = 0.99$ boundary, there is a probability ≥ 0.99 that locations are reachable. As more data is collected, the grey lines representing the boundaries of $R_\beta(b, b)$ for $\beta = 0.01$ and $\beta = 0.99$ converge to a closer proximity around the boundary for the true reachable set $\mathcal{R}_{w_{true}}(b, b)$. The closer proximity indicates that there is more certainty about where the boundary for $\mathcal{R}_{w_{true}}(b, b)$ lies.

The simulation results are evaluated using the True Reachable Rate (TRR) and Failure Rate (FR), as defined below. If $\mathcal{R}_{pred}$ is the predicted reachable set (such as the CCRS $\mathcal{R}_\beta$) and $\mathcal{R}_{w_{true}}$ is the true reachable set then

$$TRR = \frac{|\mathcal{R}_{pred} \cap \mathcal{R}_{w_{true}}|}{|\mathcal{R}_{w_{true}}|}$$

and

$$FR = \frac{|\mathcal{R}_{pred} \cap \overline{\mathcal{R}}_{w_{true}}|}{|\overline{\mathcal{R}}_{w_{true}}|}$$

where $|\cdot|$ and $\overline{\cdot}$ are set cardinality and complement, respectively. TRR is the rate of reachable locations in the true set that are predicted to be reachable, and is equivalent to a True Positive Rate. A TRR closer to 1 is better because it indicates that more locations have been correctly identified as reachable. FR is the rate of the *un*reachable locations that have falsely been identified as reachable, and is equivalent to a False Positive Rate. An FR close to 0 is preferred as it would indicate there are few failures. A failure would indicate that the robot would run out of energy prior to returning to the base.

2.4.1 Simulation Setup

For each simulation, the environment is discretized into a 41 × 41 8-connected grid with dimensions 200m × 200m, from which 1681 nodes and 6480 edges of graph $\mathcal{G}$ are defined. Each simulation consists of a robot with maximum energy capacity $a = 1.8\text{kJ}$ and a recharging base at $b = [100\text{m}, 100\text{m}]$. The robot is initialized at the recharging base at its maximum energy capacity with no data on the environment. When the robot moves to a new vertex, it collects a point measurement on the energy cost at the new location, $(y_i, \mathbf{x}_i)$, and appends it to the collected data set $\mathcal{D}$.

The energy map is built from $\mathcal{D}$ with GPR, defined previously as $e \sim \mathcal{GP}(m, k)$. The prior mean m and kernel k have a significant effect on predictions. A constant prior mean, $m(\mathbf{x}) = 12.0\frac{\text{J}}{\text{m}}$, was used for all simulations. In addition, the noise variance is assumed to be known as $\sigma_\eta^2 = 0.04$. A higher measurement noise tends to result in a slower convergence around the true reachable set, as the robot must collect more data points to reduce uncertainty. The kernels k used in the simulations depend on the environment type and are described in the following sections.

The robot explores the environment with the following strategy: starting from the base, the robot traverses a minimum expected cost path to a location near the boundary of the CCRS. A high GPR predictive variance location is chosen from among the locations near the boundary. Once the location is reached, a new minimum expected cost path is planned back to base, given the data collected (this is sometimes the same path taken to reach the location). For the following simulations, the strategy uses the CCRS with $\beta = 0.99$ so that there is high confidence that the robot will not run out of energy.

The strategy has the goal of exploring the edges of the CCRS, though further methods could be developed with a focus on other factors such as convergence rate. We note that there are numerous approaches in the literature for optimizing decision-making and control in robotic information gathering missions (see [11, 37, 95], etc.), but that such strategies are not the focus of this research.

2.4.2 Example 1 Results and Discussion: Smoothly Varying Environment

First, an environment with energy costs that vary smoothly over space is considered. Such an environment would consist of, for example, gradual transitions between terrains such as grass and sand. For this environment type, a Squared Exponential (SE) kernel is used to build the energy map. The SE kernel is chosen because samples drawn from a GP with an SE kernel are infinitely differentiable, making it a good fit for modeling a smooth environment. The kernel is defined as:

$$k_{SE}(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp(-\frac{1}{2l^2}||\mathbf{x} - \mathbf{x}'||_2^2) \tag{2.19}$$

where $l = 20.0$ is the length-scale hyperparameter and $\sigma_s^2 = 4.0$ is the signal variance.

A true energy map with smoothly varying costs is randomly generated (sampled) from a Gaussian process defined by a constant prior mean $m(\mathbf{x}) = 12.0\frac{\mathrm{J}}{\mathrm{m}}$ and the same SE kernel hyperparameters. The resulting ground truth energy map is seen in

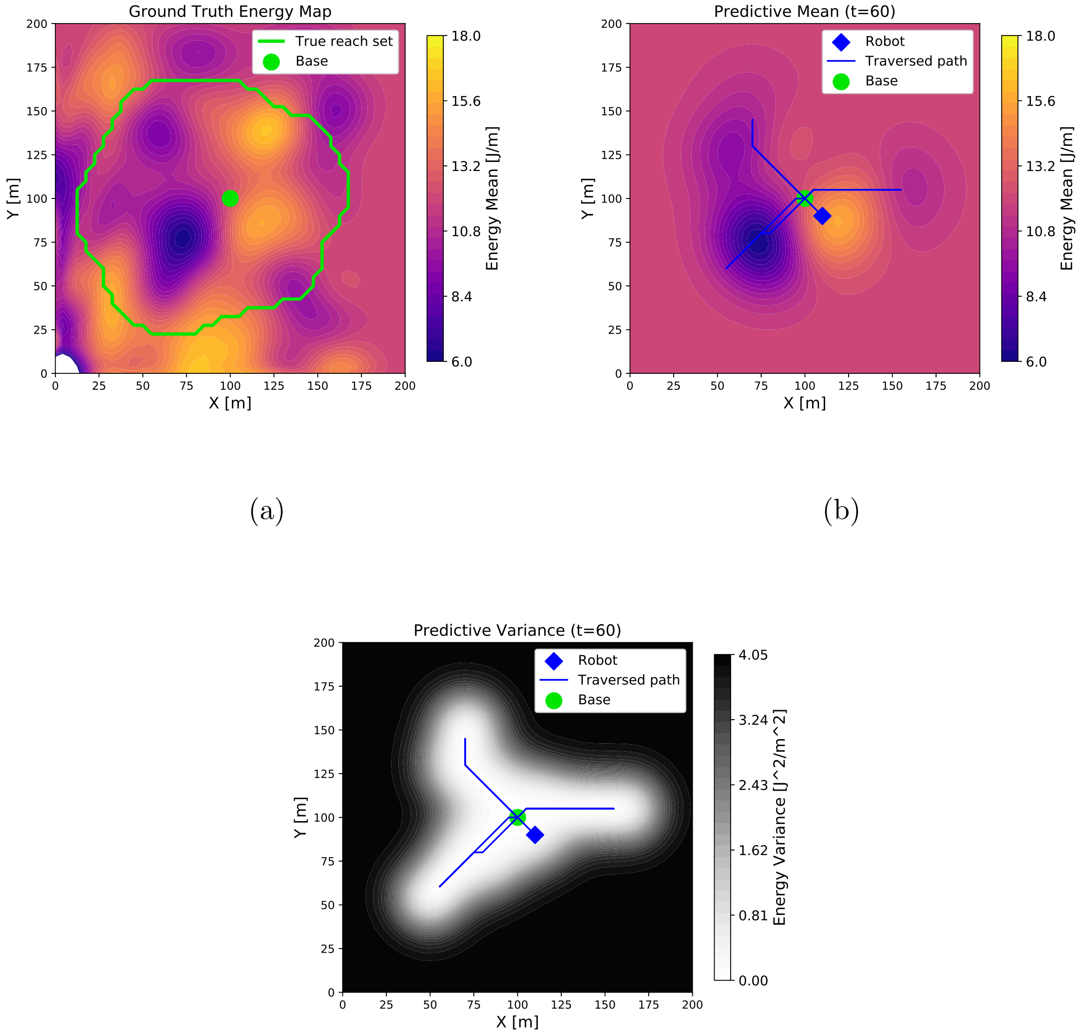
Fig. 2.3a, as well as the true reachable set, $\mathcal{R}_{w_{true}}(b, b)$. For the plots in Figs. 2.3 and 2.4, the hyperparameters of the SE kernel are assumed to be known. In practice, the hyperparameters can be chosen based on expert knowledge or learned from collected data by optimizing the marginal likelihood [84]. Further strategies consider on-line learning of hyperparameters, along with computation of the mean and variance, as data is collected [39].

The results after 60 time steps of data collection using the previously described strategy are shown in Fig. 2.3. The robot has traversed some parts of the environment and has begun to construct the energy map. There is still significant uncertainty in the map. The predicted energy cost of unexplored areas tend toward the prior mean.

The CCRS boundaries for specified βs are provided in Fig. 2.4 at $t = 0$ and $t = 60$. Initially the boundaries are spread apart from each other, implying significant uncertainty in where the true reachable set boundary lies. At $t = 60$, however, data on the environment has been collected and the boundaries are converging around the true reachable set, especially near areas where the robot has visited. In those areas, there is high confidence as to where the true reachable set boundaries lie. The robot has not visited the bottom right portion of the environment, resulting in more uncertainty about the reachable set in that area.

Simulations were performed on 50 smooth environments, generated randomly in the same way as Fig. 2.3a, and the reachable sets were observed over 150 time steps in simulation. The CCRS method was compared against the commonly used naive approach of assuming constant energy costs over the environment. The reachable set for the naive case was defined as $\mathcal{R}_{w_d}(b, b)$ with $w_d = cd_{\epsilon_i}$, where c is the energy cost per unit distance and d_{ϵ_i} is the length of a given edge ϵ_i.

The results for the comparison between the CCRS and naive cases are provided in Fig. 2.5. In both cases, there is a clear trade-off between performance, given by a high TRR, and robustness against failure, with a low FR. For example, the naive case

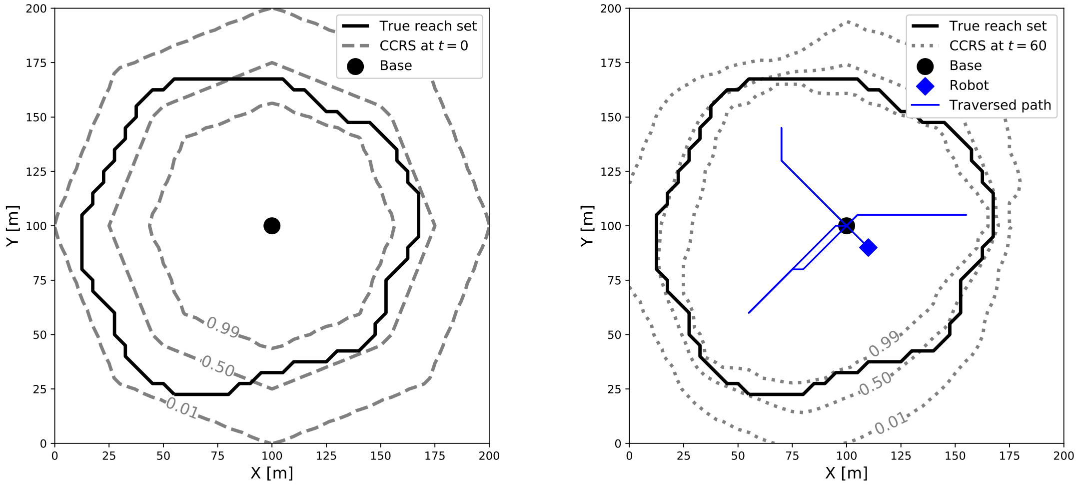


(a) (b)

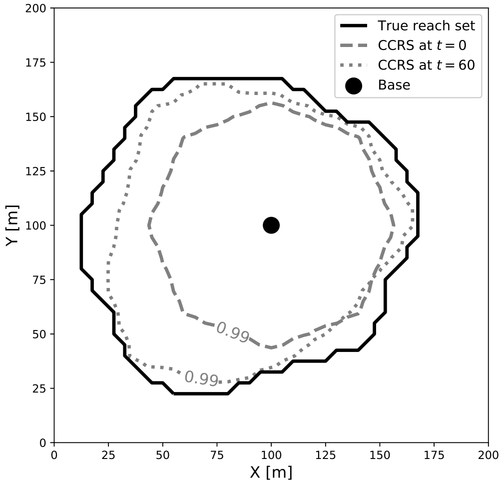
(c)

Figure 2.3: A robot (blue diamond), starting at at the recharging base (green circle), explores the environment, collecting measurements on the energy map and returning to the base for recharging. Plotted results are after 60 time steps, in which the robot moves to a new node on graph $\mathcal{G}$ at each time step. (a) True energy map, $e_{true}(\mathbf{x})$. The true reachable set, $\mathcal{R}_{w_{true}}(b, b)$, lies within the green outline. (b) GPR predictive mean, $\bar{e}(\mathbf{x})$, at $t = 60$. (c) GPR predictive variance, $\text{cov}(e(\mathbf{x}), e(\mathbf{x}))$, at $t = 60$.

of $c = 10.0$ assumes that the energy costs in the environment are low, so the reachable set is large. Unfortunately, this results in many locations being falsely identified as reachable. The naive case of $c = 12.0$ matches the average energy of the randomly



(a) $t = 0$ (b) $t = 60$

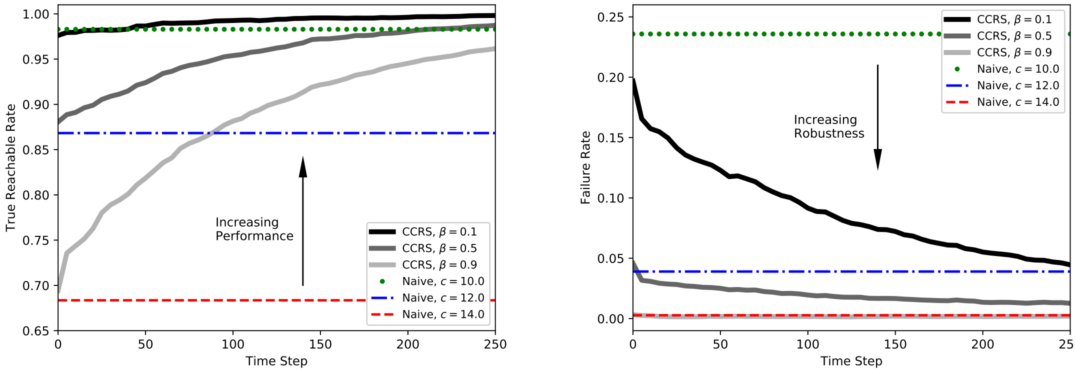


(c) $t = 0$ and $t = 60$ for $\beta = 0.90$

Figure 2.4: CCRS boundaries (gray dashed lines at $t = 0$, dotted lines at $t = 60$), $\mathcal{R}_\beta(b, b)$ for $\beta = 0.01, 0.50, 0.99$. The true reachable set, $\mathcal{R}_{w_{true}}(b, b)$, from the recharging base (black dot) lies within the black line. The boundaries are shown at (a) $t = 0$, when no data has been collected, and (b) $t = 60$, when data has been collected corresponding to the predicted maps in Fig. 2.3. (c) The expansion of the $\beta = 0.99$ set from $t = 0$ to $t = 60$ is shown.

generated energy maps, giving a moderately high TRR and a moderately low FR.

The CCRS cases are shown for varying levels of confidence: $\beta = 0.1, 0.5, 0.9$. By building a spatial map of energy costs, the results show the best of both worlds. For



(a) True Reachable Rate

(b) Failure Rate

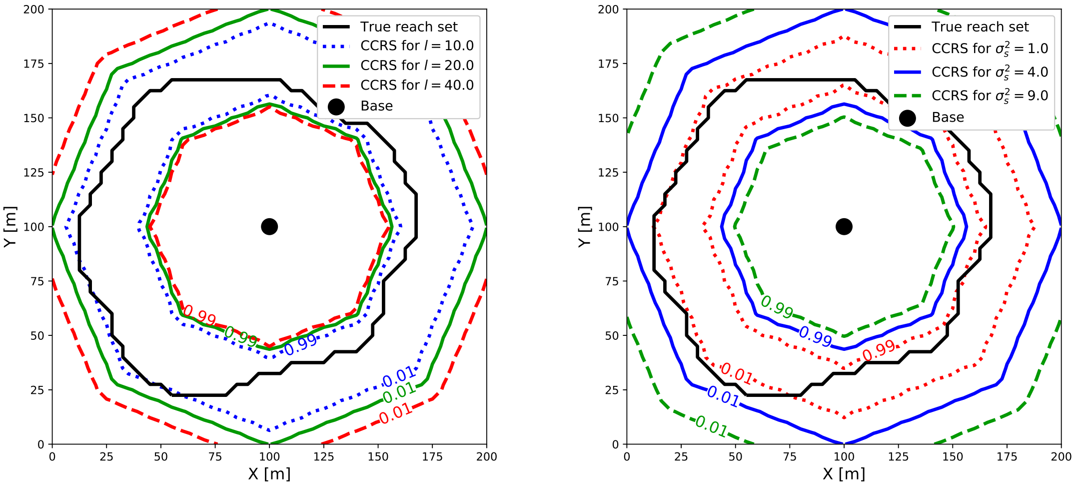
Figure 2.5: Comparison of simulation results for our CCRS method and a naive method that assumes constant energy cost per unit distance, c. Results averaged over 50 runs in randomly generated smooth environments. A higher user-defined confidence level, $\beta = 0.9$, results in a more conservative CCRS with low TRR and very low FR, though the TRR improves over time as data is collected. The robot moves to a new node on graph $\mathcal{G}$ at each time step. For the naive case, a higher assumed constant energy cost, $c = 14.0$, also results in a conservative reachable set with low TRR and FR, however, it does not use data to improve performance.

all betas, the TRR improves and the FR decreases (except for $\beta = 0.9$, which already has an FR near 0). For $\beta = 0.5$, the CCRS method achieves a significantly higher TRR while at the same time cutting the FR in half. Additionally, when there is no data at $t = 0$, the performance of both the Naive and CCRS methods are quite similar.

2.4.2.1 Hyperparameter Exploration

While the hyperparameters for making predictions were assumed to be known in these case studies, in general hyperparameters are unknown at the beginning of exploration. They instead must be chosen using a reasonable guess, until enough data has been collected on which the hyperparameters can be optimized. A reasonable guess should be conservative against running out of energy. This motivates a brief exploration of the effects of kernel hyperparameters on the CCRS.

To provide justification for the selection of hyperparameters, the effect of varying the length scale, l, and signal variance, σ_s^2, of the SE kernel is observed in Fig. 2.6 with no data at $t = 0$. Increasing σ_s^2 or l results in a smaller CCRS for $\beta = 0.99$, implying more certainty in where the true reachable set boundary lies. There is no change in the $\beta = 0.50$ CCRS boundary when σ_s^2 or l are varied (this is not shown in the figure). For σ_s^2, the explanation is that the energy costs vary over a larger range. For l, the explanation is more nuanced: a larger l implies less frequent spatial variations in energy cost. The result is higher spatial correlations between spatial input points that are some distance away from each other. With no data, this results in more uncertainty in the CCRS; when data is added there is less uncertainty with increasing l due to the higher correlation. The prior mean and energy budget parameters also have the effect of scaling the size of the reachable sets. The results of doing so are not shown here, but clearly a higher prior mean or a smaller energy budget both imply a smaller reachable set.



(a) Varying length scale, l.

(b) Varying signal variance, σ_s^2.

Figure 2.6: The effect on CCRS boundaries, $R_\beta(b, b)$, when varying hyperparameters (green, blue, and red lines) is shown for $\beta = 0.01, 0.99$. The true reachable set, $\mathcal{R}_{w_{true}}(b, b)$, (black line) was computed using a environment randomly drawn from a GP with SE kernel hyperparameters $l = 20.0$, $\sigma_s^2 = 4.0$. For (a), σ_s^2 was held constant at 4.0 and for (b), l was held constant at 20.0.

The results suggest design considerations for hyperparameters when data is not yet available. If the hyperparameters are unknown initially, the conservative recommendation is to set a high signal variance and high prior mean, as well as a smaller length-scale, then to optimize once sufficient data is available.

2.4.3 Example 2 Results and Discussion: Discontinuously Varying Environment

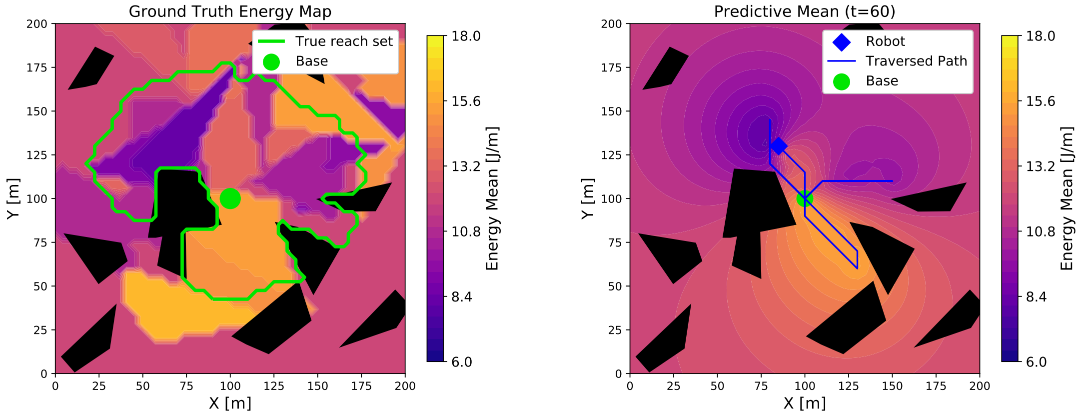
In general, operating environments may have discontinuities between terrains with different energy costs (for example, a transition from grass onto pavement). In this simulation example, the energy map is randomly generated as a set of regions with constant energy costs, as seen in Fig. 2.7a. The energy costs are drawn randomly from a uniform distribution between $8.0\frac{\mathrm{J}}{\mathrm{m}}$ and $16.0\frac{\mathrm{J}}{\mathrm{m}}$. Obstacles, which are also randomly generated and assumed to be known, are included in the simulation by removing nodes and edges that produce collisions. The obstacles do not change the computation of the reachable set except to reduce the number of edges that must be considered.

A Matern kernel with $\nu = \frac{1}{2}$ is used for GPR. A key feature of the kernel is that its smoothness can be scaled by varying the ν parameter (see [84]). A GP defined by a Matern kernel with $\nu = \frac{1}{2}$ is non-differentiable and provides more flexibility in handling rough transitions between energy costs. It is, however, continuous, resulting in an imperfect fit for a truly discontinuous function. The Matern kernel is defined as:

$$k_{Matern}(\mathbf{x}, \mathbf{x}') = \sigma_s^2 \exp(-\frac{1}{l}||\mathbf{x} - \mathbf{x}'||_2) \tag{2.20}$$

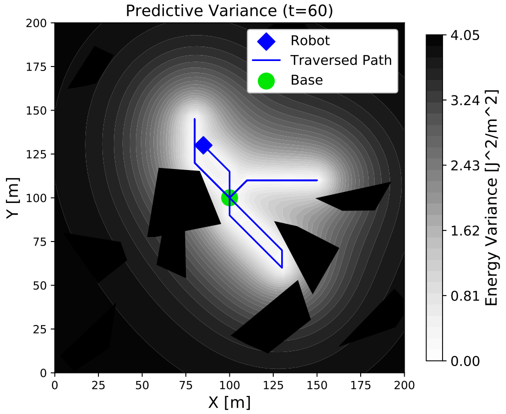
where $l = 40.0$ is the length-scale hyperparameter and $\sigma_s^2 = 4.0$ is the signal variance. We omit an exploration of the hyperparameters of the Matern kernel as the results would not differ significantly from the SE kernel.

The results of robot exploration of the environment after 60 time steps can be seen



(a) True energy map

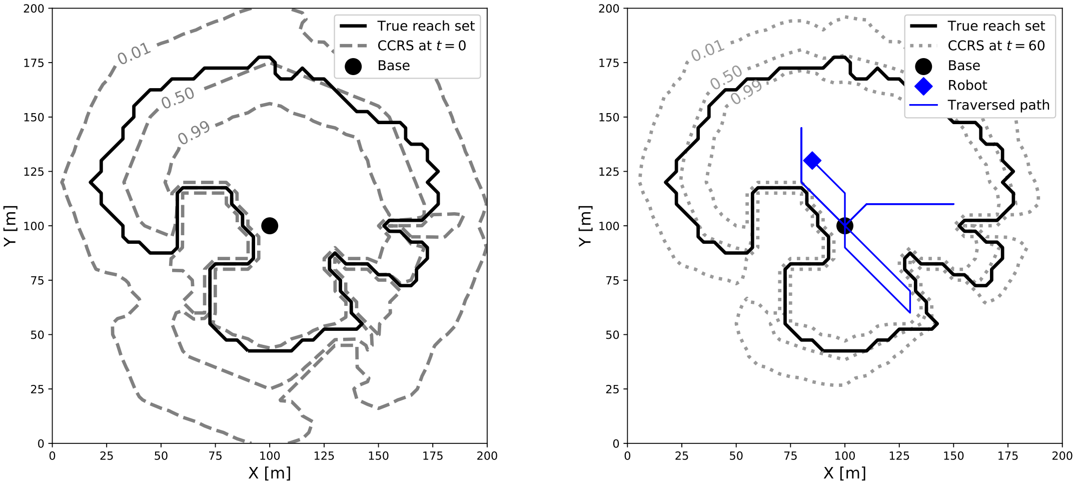
(b) GPR predictive mean at $t = 60$



(c) GPR predictive variance at $t = 60$

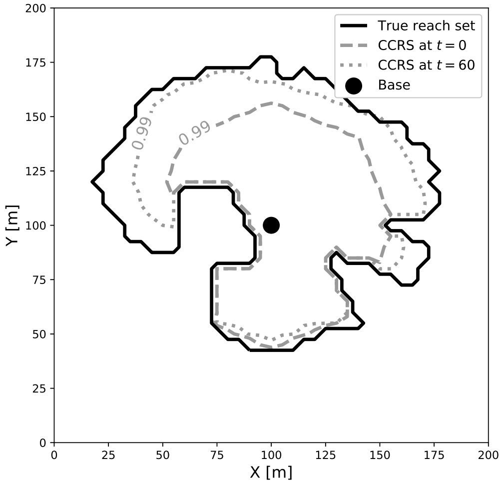
Figure 2.7: A robot (blue diamond), starting at at the recharging base (green circle), explores an environment with discontinuously varying energy costs. Plotted results are after 60 time steps. (a) True energy map, $e_{true}(\mathbf{x})$. The true reachable set, $\mathcal{R}_{w_{true}}(b, b)$, lies within the green outline. (b) GPR predictive mean, $\bar{e}(\mathbf{x})$, at $t = 60$. (c) GPR predictive variance, $\text{cov}(e(\mathbf{x}), e(\mathbf{x}))$, at $t = 60$.

in Fig. 2.7. Though broad regions are successfully discovered by $t = 60$, additional data would need to be collected to characterize the outlines of the different regions. Despite this, Fig. 2.8 reveals that even with limited characterization of the energy



(a) $t = 0$

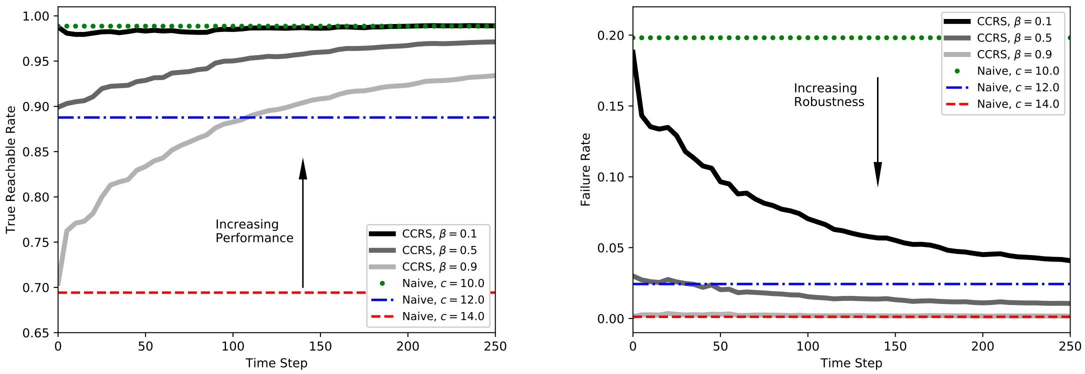
(b) $t = 60$



(c) $t = 0$ and $t = 60$ for $\beta = 0.99$

Figure 2.8: CCRS boundaries (gray dashed lines at $t = 0$, dotted lines at $t = 60$), $\mathcal{R}_\beta(b, b)$ for $\beta = 0.01, 0.50, 0.99$. The true reachable set (black line) from the recharging base (black dot). The boundaries are shown at (a) $t = 0$, when no data has been collected, and (b) $t = 60$, when data has been collected corresponding to the predicted maps in Fig. 2.7. The expansion of the $\beta = 0.99$ set from $t = 0$ to $t = 60$ is shown in (c).

map at $t = 60$, the CCRS (evaluated at different βs) has converged significantly over $t = 0$. This implies that modest data collection can still provide significant reduction in uncertainty for the reachable sets.



(a) True Reachable Rate

(b) Failure Rate

Figure 2.9: Comparison of simulation results for our CCRS method and a naive method that assumes constant energy cost per unit distance, c. Results averaged over 50 runs in randomly generated discontinuous environments. Behavior is similar to the smooth environment case.

Simulations were performed for 50 randomly generated discontinuous environments. The TRR and FR results, shown in Fig. 2.9 are quite similar to those of the smooth environment. This shows that by selecting an appropriate kernel, the CCRS method has the flexibility to be used for different types of complex and interesting environments. Importantly, model mis-specification (e.g. using the SE kernel in a discontinuously varying environment) or bad hyperparameter selection, can lead to a poor fit. If the generalization error shows poor prediction accuracy, one could resort to conservative, deterministic energy cost predictions.

In summary, the CCRS method allows performance (here in terms of TRR) to be improved as data is collected, while reducing the likelihood of failure. This complements the fact that a user can set β based on mission specifications as a trade-off between performance and robustness.

2.4.4 Computation Time

GP prediction was performed using the scikit-learn Python package and integration to compute expected path energy costs and the corresponding variance was performed numerically. Graph-based analysis and shortest path computations were performed using the NetworkX Python package. The computation of the CCRS for the smooth environment took 10 seconds at $t = 0$ with no collected data and 15 seconds at $t = 150$ with 150 data points. On average, it took 0.05 seconds to verify the reachability of a node. For the discontinuous environment, the computation took 9 seconds at $t = 0$ and 13 seconds at $t = 150$. The shorter computation time is due to fewer nodes as a result of obstacles. Simulations were performed on a laptop with a Intel Core i7-2600 processor.

2.5 Conclusions

A method was presented for computing chance constrained reachable sets in environments with spatially varying energy costs. The proposed approach builds a spatial map of the energy cost data, finds minimum cost paths to locations in the environment based on the predicted costs of paths, and checks the chance constrained feasibility of paths to those locations. The simulation results show that by spatially mapping energy cost data, the method can significantly improve the true reachable rate and reduce the failure rate of the predicted reachable sets as data is collected. The results also demonstrate that the method can be used in environments with smoothly or discontinuously varying energy costs.

The methods presented here could be extended beyond scalar fields to vector fields and non-flat environments with 3-D topography, with energy prediction such as described in the next chapter. While undirected graphs were considered here, straightforward extensions to directed graphs are possible. Understanding a robot's reachability in an environment can be used to inform future mission planning by constraining or expanding the known area that a robot can plan to traverse through. The CCRS method could be applied to risk-aware and informative path planning problems, as well as methods for optimal data collection for building the reachable sets. Furthermore, in Example 2, obstacle locations were assumed to be known. Future work could consider uncertainty in obstacle location and geometry in the formulation, for example, as in Axelrod et al. [6].

CHAPTER III

Off-Road Ground Robot Path Energy Cost Prediction Through Probabilistic Spatial Mapping

In the previous chapter, a 2-d formulation for terrain mapping and energy prediction was presented and demonstrated in simulation through a reachability application. In this chapter, we extend our previous work and provide a methodology for 3-d environments, both through a simple vehicle model and with a fully data-driven approach. Importantly, the methodology is experimentally tested in an outdoor environment with varying terrain. Performance, both in terms of error and uncertainty quantification, is evaluated on two ground robot platforms. The work in this chapter is primarily based on [77], with preliminary work presented in [80].

As stated in previous chapters, prior work has explored the use of robotic spatial mapping for predicting power usage and path energy costs on varying terrains [63, 56, 67]. However, there has been little experimental validation that path energy cost predictions made based on spatial mapping are actually accurate, particularly in 3D environments. 3D environments are challenging because power consumption is likely to be different at the same location, depending on the robot's heading. In addition, longitudinal vehicle models, which are commonly used for robot energy prediction on 3D terrain [104, 88], often assume that the friction coefficient (or rolling resistance coefficient) is the same on identical terrain, regardless of slope or direction. Such an assumption can result in biases that throw off predictions.

Path energy cost prediction (as opposed to predicting the power at a single location) involves the summation (or integration) over power predictions along a path. One challenge is that the errors in predicted power consumption tend to be correlated for points along a path. When summed together, existing biases in predictions compound and produce significant errors. Another result of correlated costs is that the predicted energy cost of a path is at risk of being very overconfident if the correlations are not taken into account when computing uncertainty [67].

In this chapter, a spatial mapping method for predicting energy costs is presented and experimentally tested in a 3D environment. The approach uses Gaussian process regression (GPR) and vehicle modeling to build a map from inputs (including position, heading, slope and satellite imagery) to power using collected data. The predicted energy cost of a path is then computed through a summation of correlated power predictions. The approach presented in this chapter, while building on previous strategies for energy prediction through spatial mapping, makes improvements in a number of ways. First, the formulation for path energy cost prediction considers correlated costs when characterizing uncertainty, addressing the previously stated problem of overconfident predictions. Second, the energy prediction formulation considers the impact of prior vehicle modeling on predictions. Third, a decimation scheme based on a desired sampling length is used to reduce the data and computation time needed for prediction.

The approach is evaluated experimentally in terms of predictive accuracy through cross-validation for the case of a small, well-mapped environment with varying terrains. The effects of different Gaussian process inputs and kernels are observed. We also consider predictions when data on the environment is limited (as in, only some of the environment has been traversed). When data is limited, predictions are highly uncertain and are especially dependent on the values of a set of hyperparameters used in GPR. Since optimization of hyperparameters with sparse data is not reliable, the effect of varying different hyperparameters is observed, providing guidance on hyperparameter selection for future use.

3.0.1 Contributions

The main contributions of this chapter are:

1. An methodology for probabilistic energy cost prediction in 3D off-road environments through spatial mapping of collected data with GPR.
2. An experimental validation of probabilistic energy predictions, consisting of:
    - Evaluating the predictive performance in both well-mapped and sparse data cases.
    - Assessment of other methodological factors, including: kernel choice, hyperparameters, computation time, and prior satellite imagery.

The remainder of the chapter proceeds as follows: Section 3.1 describes related literature in robotic energy consumption, Section 3.2 provides the methodology, Section 3.3 reviews the experimental results, and Section 3.4 concludes and describes future work. Further experimental results are provided in Appendices A and B.

3.1 Related Work

GPR has been used as a tool for spatial mapping of robot energy costs to enable energy prediction, path energy minimization, and high-level mission planning. The authors of this paper demonstrated the application of energy prediction to a multi-robot information gathering problem [79]. Martin and Corke [56, 57] used GPR to build a spatial map of a *flat* environment based on robot power consumption and subsequently find energy minimal tours. Murphy et al. [63, 62] built spatial maps of terrain traversability for metrics such as power and slip. These maps were used for planning on a probabilistic costmap with a sampling-based method.

Experimental validation of the reliability of path energy cost predictions is necessary for planning applications. Oliveira et al. [67] used GPR to model and predict power consumption and path energy cost based on a robot's position and heading. The uncertainty in the predictions was used for energy minimal path optimization. However, the computed uncertainty was highly overconfident, likely because correlations in costs along a path were not considered. Further work is needed to validate path energy cost predictions, particularly in 3D environments.

Previous strategies have focused mostly on purely data-driven GPR mappings. Vehicle or environment modeling in combination with GPR could aid predictions, especially when data on the environment is sparse. Plonski et al. [74, 75] used a ground robot equipped with a solar panel to build a spatial solar energy map of an environment based on shadows. Including further modeling through environment reconstruction was shown to improve predictive performance over standard GPR, showing the potential benefits of combining GPR with additional modeling.

Ground robot power prediction has often relied on longitudinal vehicle models, perhaps due to their simplicity [82, 88]. Sadrpour et al. [88, 87] used a longitudinal power model to predict mission energy costs for a ground robot, as well as update those predictions in real-time with collected data. Prior knowledge on factors, including terrain friction, slope, and teleoperator aggressiveness, inform the predictions. Sun and Reif [104] focused on energy minimal path planning using a longitudinal model as well. Previous work with such models typically assumes that the terrain friction (or rolling resistance coefficient) is known throughout an environment. A constant friction assumption is unrealistic in an off-road environment. We build a spatial map of the terrain with GPR using data collected during robot operation, in order to account for changes in terrain friction.

While longitudinal models are quite useful, turning has a significant and non-trivial impact on power consumption. Dogru and Marques [27] develop a physics-based model for a skid-steer robot that considers a number of factors, including: speed, radius of curvature, and temperature. Canfield et al. [17] develop a power model for a skid steered robot based on the equations of motion. These works provide useful modeling for a robot's power usage, but are highly particular to the vehicle type. We instead focus on the longitudinal dynamics and spatially dependent terrain energy costs.

3.2 Methodology

In this section, the methodology for predicting the energy cost of a path, based on spatially mapped data collected by a robot, is described. First, GPR is used to build a map from input points $\mathbf{x}$ (position, heading, etc.) to power usage P with the aid of vehicle model information. Next, the energy cost of a path is predicted, in terms of its mean and variance, as the summation of a set of correlated power predictions produced by GPR. In addition, the inputs to the Gaussian process used in the chapter are detailed, along with the optimization of hyperparameters. Finally, a decimation strategy is provided to reduce the dimensionality of the training data.

3.2.1 Vehicle Power Modeling with Gaussian Process Regression

For modeling and prediction, we consider cases in which the robot is traversing along straight-line paths. The focus is on the longitudinal dynamics of the vehicle (small turns to keep the robot along the path are assumed to be negligible in terms of energy cost). Robot energy consumption due to turning is a topic of ongoing research [17, 27], and is highly specific to the vehicle type (e.g. Ackermann steering vs skid-steering). Longitudinal dynamics, on the other hand, are similar across many types of ground vehicles.

To predict power consumption of a robot operating in an environment, a mapping is built from d inputs $\mathbf{x} \in \mathbb{R}^d$ to power $P \in \mathbb{R}$ using GPR.We consider the following inputs: the position x, y of the robot, the robot's heading ψ, the slope θ in the direction of the robot's heading, and an imagery input of the terrain $s(x, y)$ from the grayscale pixel intensity at a given location. The power model is assumed to be of the following form:

$$P = a(\beta)[f(\mathbf{x}) + \eta] + b(\beta) \tag{3.1}$$

where $a(\beta), b(\beta)$ are provided based on known model parameters β. $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$ is independent and identically distributed Gaussian noise with variance $\sigma_\eta^2 > 0$, and $f : \mathbb{R}^d \to \mathbb{R}$ is a terrain-dependent function learned through GPR.

The inputs to the GP are assumed to be known, both for training the model and for future predictions. For example, information regarding the slope throughout an environment could be determined from digital elevation maps (DEMs) in large-scale environments, or directly from information collected in real-time such as from LIDAR.

The functions $a(\beta)$ and $b(\beta)$ can incorporate prior information that is known about the power model. Two models are considered in this chapter:

1. The longitudinal vehicle power model (LVM) is a commonly used physics-based model [87, 104], with

   $$a(\beta) = uW \cos(\phi) \tag{3.2}$$

   and

   $$b(\beta) = u(W \sin(\theta) + m a_{accel} + C_I) + b_{int}. \tag{3.3}$$

   The parameters u, W, m, a_{accel} are the robot's speed, weight, mass, and acceleration, respectively. ϕ is the gradient of the terrain face (we assume that ϕ is small and let $\cos(\phi) \approx 1$). C_I and b_{int} are losses due to internal resistances and other robot electronics power consumption, respectively.

2. A direct power model (DPM) assumes no prior model information, with $a(\beta) = 1$ and $b(\beta) = 0$.

For the LVM, the output of the function $f(\mathbf{x})$ can be interpreted as the rolling resistance or friction coefficients of the terrain. The coefficient depends on the wheel-terrain interaction and contributes to energy losses through terrain and wheel deformation. In contrast, the DPM ignores vehicle model information, and $f(\mathbf{x})$ is a mapping learned directly from the inputs to power. The benefits and drawbacks of these models are explored in the experimental portion of this chapter.

To perform GPR and learn the function $f(\mathbf{x})$, a set of training data is collected. The output data points, z_i, are defined based on Eqn. (3.1) as:

$$z_i := \frac{P_i - b(\beta_i)}{a(\beta_i)} \tag{3.4}$$

where P_i and β_i correspond to a power measurement and model input, respectively. The output data, along with the input measurements, $\mathbf{x}_i$, are collected as $\mathcal{D} = \{(z_i, \mathbf{x}_i)\}_{i=1}^{n}$ and denoted $\mathbf{z} \in \mathbb{R}^n$ and $X \in \mathbb{R}^{n \times d}$ for the stacked z_i and $\mathbf{x}_i$, respectively.

The training data $\mathcal{D}$ is modelled using a Gaussian process (GP), denoted $f \sim \mathcal{GP}(c, k)$. The GP is characterized by a prior mean function $c : \mathbb{R}^d \to \mathbb{R}$ and a kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, which is a symmetric function that computes the correlation between two points in $\mathbb{R}^d$.

Gaussian process regression (GPR) defines an n-variate normal distribution based on c and k (see [84] for further information on GPR). The posterior predictive distribution computes the distribution of $f(\mathbf{x}_*)$ at a set of m target points, $X_* \in \mathbb{R}^{m \times d}$ based on the data. The probability density function for $f(X_*)|\mathcal{D}$ is known to be [84]:

$$p(f(X_*)|\mathcal{D}) = \mathcal{N}\big(\bar{f}(X_*), \mathrm{cov}(f(X_*), f(X_*))\big) \tag{3.5}$$

where

$$\begin{aligned}\bar{f}(X_*) := \mathbb{E}[f(X_*)|\mathcal{D}, X_*] = c(X_*) \\ + K(X_*, X)[K(X,X) + \sigma_\eta^2 I]^{-1}(\mathbf{z} - c(X))\end{aligned} \tag{3.6}$$

are the posterior (or predictive) means for the points in $\mathbf{x}_*$ and

$$\begin{aligned}\mathrm{cov}(f(X_*), f(X_*)) := K(X_*, X_*) \\ - K(X_*, X)[K(X,X) + \sigma_\eta^2 I]^{-1} K(X, X_*)\end{aligned} \tag{3.7}$$

is the covariance matrix for the target set of points, X_*. The notation $K(X, X_*)$ defines the $n \times m$ covariance matrix of the point-wise comparison of the n input training points with the m input target points, based on the kernel k. The point-wise evaluation of the prior mean results in an $n \times 1$ vector for $c(X)$.

To compute the predicted power, the GPR prediction is passed through the linear power model given in Eqn. (3.1). The following terms are defined to simplify computation: $A := \mathrm{diag}(a(\beta_{*,1}), \dots, a(\beta_{*,m}))$ and $B := \begin{bmatrix} b(\beta_{*,1}) & \dots & b(\beta_{*,m}) \end{bmatrix}^T$, $\mathcal{P}_* := \begin{bmatrix} P_{*,1} & \dots & P_{*,m} \end{bmatrix}^T$, $H := \mathbf{1}\eta$. Then,

$$\mathcal{P}_* = A[f(X_*) + H] + B \tag{3.8}$$

where P is a multivariate normal distribution that defines the joint power distribution over the set of target points. The probability density function is then:

$$p(\mathcal{P}_*|\mathcal{D}) = \mathcal{N}\big(\bar{\mathcal{P}}_*, \mathrm{cov}(\mathcal{P}_*)\big) \tag{3.9}$$

where the mean can be computed based on the linearity of expectation and by assuming zero mean noise ($\mathbb{E}[H] = 0$):

$$\bar{\mathcal{P}}_* := \mathbb{E}[\mathcal{P}_*|\mathcal{D}] = A\bar{f}(X_*) + B \tag{3.10}$$

The covariance matrix is derived by standard manipulations as:

$$\text{cov}(\mathcal{P}_*) = A\Big(\text{cov}(f(X_*), f(X_*)) + \sigma_\eta^2 I\Big)A^T. \tag{3.11}$$

The covariance matrix describes the uncertainty in the predicted power for a given set of target points along a path, as well as the correlations between those points.

3.2.2 Probabilistic Path Energy Cost Prediction

The energy cost of a path, denoted E, is computed by integrating power usage over time:

$$E = \int_0^T P(t)dt \approx \sum_{j=1}^{m} P_{*,j}\Delta t \tag{3.12}$$

where Δt is time interval of the Riemann sum approximation of the integral and $P_{*,j}$ is the jth element of $\mathcal{P}_*$. For path energy prediction, the summation is over a set of power predictions, $P_{*,j}$, that are spaced along the predefined path with inputs $\mathbf{x}_{*,j}$. The velocity profile of the vehicle for the predicted path is assumed to be known, allowing inputs $\mathbf{x}_{*,j}$ to be generated based on known time intervals Δt.

Since the energy cost of the path is computed as a summation over normally distributed random variables, then E is normally distributed as well, with $E|\mathcal{D} \sim \mathcal{N}(\bar{E}, \text{var}(E))$, where

$$\bar{E} := \mathbb{E}[E|\mathcal{D}] = \sum_{j=1}^{m} \big(A\bar{f}(\mathbf{x}_{*,j}) + B\big)\Delta t \tag{3.13}$$

is computed based on the linearity of expectation and

$$\begin{aligned}\operatorname{var}(E) &= \operatorname{var}(\sum_{j=1}^{m} P_{*,j}\Delta t) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} \operatorname{cov}(P_{*,i}, P_{*,j})(\Delta t)^2,\end{aligned} \tag{3.14}$$

where $\operatorname{cov}(P_{*,i}, P_{*,j})$ are the entries of $\operatorname{cov}(\mathcal{P}_*)$ from Eqn. (3.11). These computations follow from the standard summation of correlated random variables.

The variance of E includes the correlations between power costs along a path, as computed by the GP kernel. Such correlations are necessary to properly characterize uncertainties in predicted energy costs. For example, the power usage for two inputs, $\mathbf{x}_{*,i}$ and $\mathbf{x}_{*,j}$, that are close together is likely to be similar. The result is an increase in uncertainty when the outputs are summed together. Ignoring correlations between costs has been shown to result in severe overconfidence in predictions [67].

3.2.3 Inputs to the Gaussian Process

The inputs $\mathbf{x}$ to the Gaussian process $f(\mathbf{x})$ play an important role in predictions. The inputs considered in the experimental results are: position x and y, the slope in the direction of robot motion θ, the robot's heading ψ, and satellite imagery s. The purpose of the x and y inputs is to identify the spatial changes in energy costs due to the terrain that the robot is traversing. The ψ and θ inputs serve to capture power model behaviors that are otherwise unknown. These inputs are related, however, the results show that they affect predictive performance in different ways. In implementation, ψ was projected onto the unit circle as $\begin{bmatrix}\cos(\psi) & \sin(\psi)\end{bmatrix}^T$, making it amenable to kernel computations.

In many scenarios, satellite imagery, soil maps, or other forms of exteroceptive sensing may be available to the robot. Such information on the environment is useful, for example, in detecting abrupt terrain changes (such as a sudden transition between grass and pavement). In this chapter, a function $s(x, y)$ is constructed based on pixel intensities of a grayscale satellite image of the experimental environment. Two input points may be spatially close together, however, if the inputs have very different pixel intensities, then the inputs will have a small correlation as defined by the kernel. A similar strategy was employed by Murphy et al. [63], considering pixel chromaticity values in mapping.

3.2.4 Kernels and Optimization of Hyperparameters

The structure of the kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ plays an important role in GPR by defining the correlation between two inputs. For the experimental results, we consider the same kernels used in Chapter II: the Matern and the Squared Exponential (SE) kernels. The kernels are based on the distance between two inputs, $\mathbf{x}_i$ and $\mathbf{x}_j$, defined as $d = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T M^{-1} (\mathbf{x}_i - \mathbf{x}_j)}$, where $M = \text{diag}(l_1^2, \ldots, l_d^2)$. The distance shown here is anisotropic in that the distance depends on length-scale hyperparameters, $l_1^2, \ldots, l_d^2$, that vary between the inputs. For this application, we consider the spatial inputs x and y to have the same length-scale l_{xy}. However, l_θ, l_ψ, and l_s are likely to have different length-scales. For both kernels, as input distance decreases, correlation increases.

The Matern kernel has a parameter ν that scales the smoothness of the GP. For example, the Matern kernel with $\nu = \frac{1}{2}$ (Ma1/2) is defined as:

$$k_{Ma12}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \text{exp}(-d) \tag{3.15}$$

where $\sigma^2 > 0$ is the signal variance hyperparameter. The resulting GP is not differentiable. In the experimental results, we also consider the Matern kernel with $\nu = \frac{3}{2}$ (Ma3/2) and $\nu = \frac{5}{2}$ (Ma5/2); the GPs of these kernels are once and twice differentiable, respectively. As $\nu \to \infty$, the Matern kernel equals the Squared Exponential

(SE) kernel [84], defined as:

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp(-0.5 d^2). \tag{3.16}$$

A GP with an SE kernel is infinitely differentiable.

The hyperparameters of the GP, $\Theta := \{\sigma^2, l_1^2, \ldots, l_d^2, \sigma_\eta^2\}$, are optimized by maximizing the log marginal likelihood [84]:

$$\begin{aligned} \log p(\mathbf{z}|X, \Theta) = &-\frac{1}{2}\mathbf{z}^T(K(X, X) + \sigma_\eta^2 I)\mathbf{z} \\ &- \frac{1}{2}\log|K(X, X) + \sigma_\eta^2 I| - \frac{n}{2}\log 2\pi. \end{aligned} \tag{3.17}$$

In other words, the probability of the output data given the model assumptions is maximized. The gradient of the log marginal likelihood can be computed analytically, leading to reasonably fast convergence. However, the log marginal likelihood can have local minima and increasing the number of inputs can exacerbate this problem due to the higher dimensional optimization. If an input is irrevelant or redundant, the optimization tends to result in the length-scale for that input to become very large, effectively making the input insignificant in predictions. This process is often referred to as Automatic Relevance Determination (ARD) [84].

3.2.5 Decimation of Collected Data

A well-known challenge with GPR is that its time complexity scales as $O(n^3)$ with the number of training data points. For the experimental platform used in this chapter, the sampling time was $T_s = 0.1$. If the dataset $\mathcal{D}$ grows by one training data point for every T_s, predictions will quickly become infeasible.

We use a filtering and downsampling (also known as decimation) strategy as a preprocessing step. This step significantly reduces the dimensionality of the problem, with little impact on performance in practice. An integer factor, M, is determined for downsampling the rolling resistance coefficients. M is designed based on the following heuristic:

$$M = \left\lfloor \frac{L}{uT_s} \right\rfloor \tag{3.18}$$

where L is a specified sampling length and u is the speed of the robot. In other words, M is the downsampling factor needed in order to produce one data point per specified sampling length. The actual sampling length based on M (due to the floor function) is $L_M = MuT_s$. A zero-phase low-pass filter is used prior to downsampling. Measured signals on the robot, including the power, position, and slope are decimated. Satellite imagery along the path is treated as a signal and is also decimated.

The decimated data is used for GPR and hyperparameter optimization. The noise variance, σ_η^2, also determined during hyperparameter optimization, can express noise in the measured signal, as well as variability in the terrain. Depending on the sampling length, σ_η^2 can capture different processes with varying levels of significance. In practice, a longer sampling length can provide a better fit of the model described in this section. Additionally, path energy cost prediction includes σ_η^2, so the inputs $\mathbf{x}_{*,j}$ for the predicted path should be determined based on L_M, as should Δt.

3.3 Experimental Results

We validate the methodology described in the previous section on a dataset collected from an experiment performed with a ground robot in an outdoor environment. The results are evaluated in several ways. First, spatial maps of the terrain are produced using the collected data and are qualitatively assessed. The accuracy of path energy cost predictions is then evaluated through a cross-validation in a well-mapped environment. Further cross-validation results from a second dataset, collected by a larger robot in the same environment, are presented in the Appendix.

Next, the case of sparse data on the environment is considered. LVM and DPM predictions are compared for varying amounts of data. The method is also evaluated with respect to: the sensitivity of the predictions to kernel choice and hyperparameters, the effect of including prior satellite imagery, and accuracy versus computation time for varying sampling length.

The performance of path energy prediction is evaluated in terms of the percent error between the mean predicted and the measured energy cost:

$$\text{Percent Error} = \frac{\bar{E} - E_{true}}{E_{true}} \times 100\% \tag{3.19}$$

where E_{true} is the experimentally measured energy cost and $\bar{E}$ is computed as in Eqn. (3.13). E_{true} is computed using the raw experimentally measured power data (without decimation).

The predictive distribution of the path energy cost E is also evaluated. The probabilistic log likelihood (PLL) is one common metric [63]:

$$\text{PLL} = -\frac{1}{2}\log\,\text{var}(E) - \frac{(\bar{E} - E_{true})^2}{2\text{var}(E)} - \frac{1}{2}\log 2\pi \tag{3.20}$$

where var(E) is computed as in Eqn. (3.14). A smaller variance is better, though a prediction that falls outside a reasonable confidence interval given by var(E) is heavily penalized. A higher PLL is better.

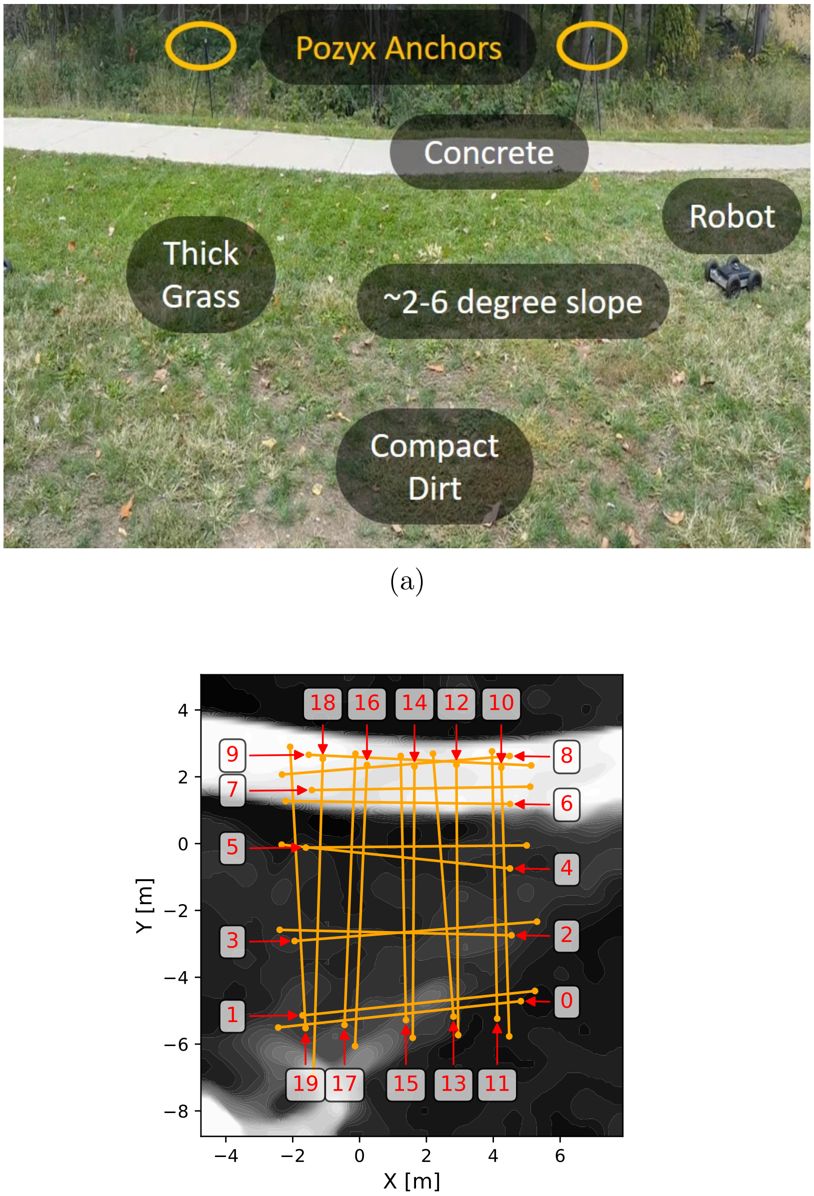
3.3.1 Experimental Setup

The experiment was performed on a small robot platform (SuperDroid Robots, IG32-DM4, 4WD), pictured in Fig. 3.2. Further results on a large platform are also provided in Appendix B. Current draw from the robot's batteries (two 11.1V batteries wired in series) was measured with a current sensor (INA169 Analog DC) and used to compute power, $P = IV$.

The environment, shown in Fig. 3.1a, was chosen for the significantly varying resistances associated with the terrains, which include: concrete, thick grass, and compact dirt. The environment itself is on a slope that varies between 2 and 6 degrees. The position of the robot was obtained using an absolute localization system (Pozyx NV, Ghent, Belgium). The system uses ultra-wideband technology to determine the position of a tag (placed on the robot) based on a set of anchors in the environment. The tag includes an inertial measurement unit for measuring attitude angle (roll, pitch and yaw). For energy prediction computations we only used pitch θ.

The robot was operated remotely by a user to traverse a set of 20 straight-line paths ($\sim$ 6 to 9 meters in length) through the environment at a constant velocity. Slight turns were made to keep the robot along the desired path. Velocity was maintained through onboard encoders and PID control. Position measurements showed that the set velocity was maintained to within a reasonable error on all terrains in this environment. To minimize operator error as a factor in results, the initial and final positions of each path, shown in Fig. 3.1b, are defined based on positions measured by the Pozyx system. The data set used for mapping and prediction consists of the portion of the paths in which the robot has reached a steady state velocity. The data for each path ends exactly when the stop command was given, marking the final position in the path.

While the robot itself collects data on current and receives motor commands through a laptop at 10Hz, data from the Pozyx system is collected off-board, with occasional packet loss. Position and slope data is interpolated in post-processing and merged with the current sensor data. GPR computations were performed using the scikit-learn Python package. The grayscale image (satellite imagery that has been passed through a median filter), shown in Fig. 3.1b, is overlaid with the spatial map and the function $s(x, y)$ is constructed using the Python-SciPy function *interp2d* with pixel intensities normalized between 0 and 1. All results presented are based on post-processing of the data set collected during the experiment.



(a)

(b)

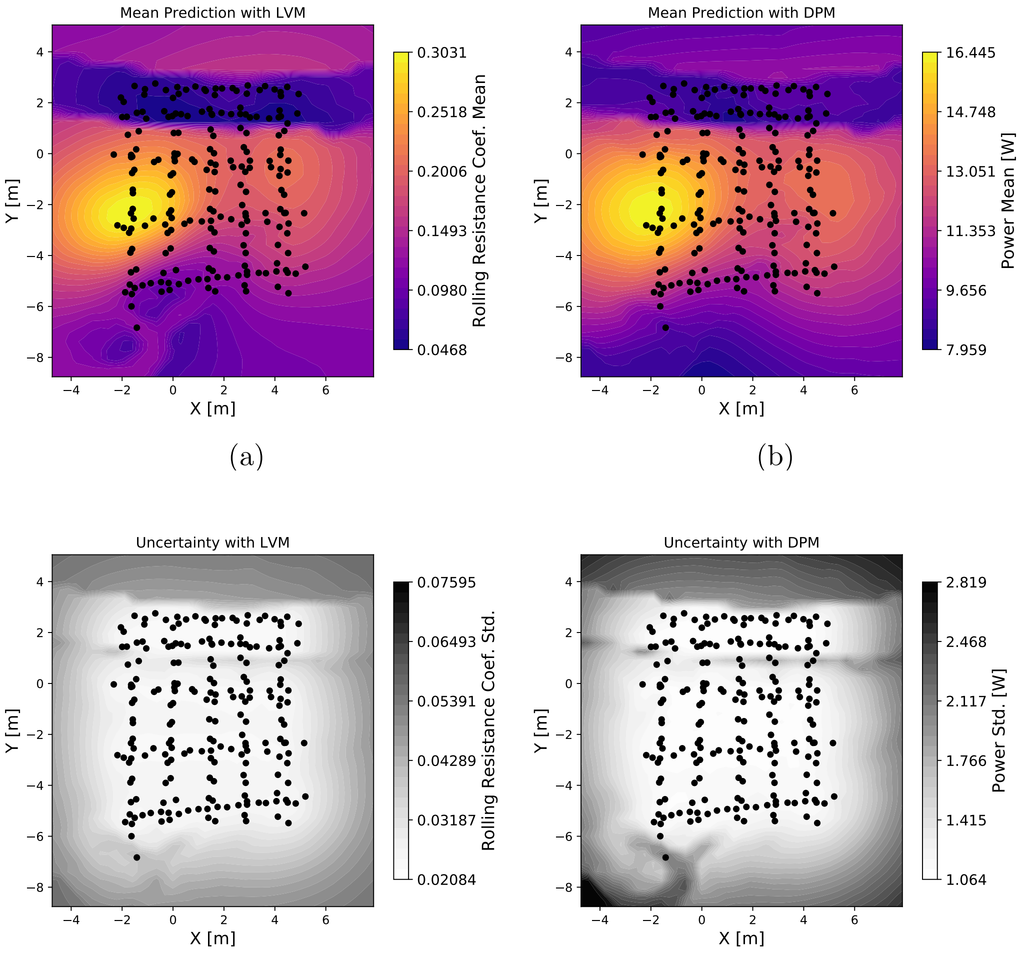
Figure 3.1: (a) Image of experiment environment (2 of 4 Pozyx anchors used for robot positioning are shown). (b) Grayscale top-down satellite image of the experiment environment used to define the GP input s. Twenty paths traversed by the robot are overlaid as orange lines. The initial location of each path and the path number is indicated in red. The white portion at the top of (b) corresponds with the concrete walk in (a).

Figure 3.2: Large (10 inch wheels and 15.15kg mass) and small robots (6 inch wheels and 7.25kg mass) used for experiments.

3.3.2 Spatial Terrain Mapping

The spatial mapping results based on the 20 traversed paths in the experimental environment are qualitatively observed in Fig. 3.3. The LVM spatial map is interpreted as a map of the rolling resistance coefficient (where $\psi = \pi$), whereas the DPM map is of power usage, with $\psi = \pi$. The sampling length is set as $L = 0.8$[m]. The hyperparameters of the Ma3/2 kernel were optimized using the log marginal likelihood and the resulting values are provided in Table 3.1. For the optimization, the prior mean c was fixed to the mean of the training data, and the satellite length-scale hyperparameter was set at $l_s = 0.3$. The physical parameters for LVM predictions, determined through off-line calibration, are provided in the Appendix in Table B.1.

Qualitatively, both spatial maps capture the higher "cost" (i.e. rolling resistance coefficient or power) in the grass area over the concrete path, as well as lower cost in the compact dirt area. The predictive means in Figs. 3.3a and 3.3b differ more strongly for predictions further away from the data, due to the different quantities that the LVM and DPM represent, and that the predictions are shown for the robot moving uphill (for $\psi = \pi$). Away from the data, predictions tend to revert to the prior mean. In addition, the uncertainty in the maps increases further away from the data set, as is expected given a Matern or SE kernel.



(a) (b)

(c) (d)

Figure 3.3: The mean prediction $\bar{f}(X_*)$ for GPR is shown in (a) and (b), and the uncertainty $cov(f(X_*), f(X_*)) + \sigma_\eta^2 I$ is in (c) and (d). The predictions are made using the data (black dots) collected from 20 paths and with the following models: (a),(c) LVM with GP inputs x, y, ψ, s. (b),(d) DPM with GP inputs x, y, ψ, s.

Hyperparameter	LVM	DPM
σ	0.0551	2.96
σ_η	0.0209	0.972
l_{xy}	3.88	6.09
l_ψ	2.84	2.6
l_s	0.3	0.3
c	0.119	8.812

Table 3.1: Optimized Hyperparameters for Kernel Ma3/2

The effect of including the satellite imagery as an input to the GP can be seen by the clearly defined discontinuity between the concrete and the rest of the area. This is because points on either side of the discontinuity have very different grayscale intensity values. Thus, two of such points have low correlation as defined by the kernel.

3.3.3 Path Energy Prediction Error

The accuracy of the spatial mapping methodology is evaluated through a cross-validation strategy. In this strategy, 1 path is removed from the data set and the measured cost of that path is compared against the predicted cost using the data from the other 19 paths. This process is repeated for all paths and the hyperparameters are optimized each time. Note: The measured energy cost of the path is computed from the raw measured current sensor data.

For the path to be predicted using the methodology described in Section 3.2, some knowledge about the path is assumed:

1. The initial and final positions of the path are known.
2. The robot follows a straight-line path between those positions and moves at the constant commanded velocity.
3. The average slope along the path is known and is considered as the slope for the entire path.

Predictions from the cross-validation for Path 0 through Path 9 are shown in Fig. 3.4 for the LVM and DPM models. The two plots show the measured energy cost of each path, the predicted cost based on GPR, and a baseline prediction. The baseline prediction is made assuming a constant function $f(\mathbf{x}) = c_b$ for each model, where $c_b \sim \mathcal{N}(\mu_b, \sigma_b^2)$ and μ_b, σ_b^2 are chosen to be the mean and variance of the training data $\mathbf{z}$, respectively. The baseline prediction is made by averaging over 100 randomly sampled c_b values.

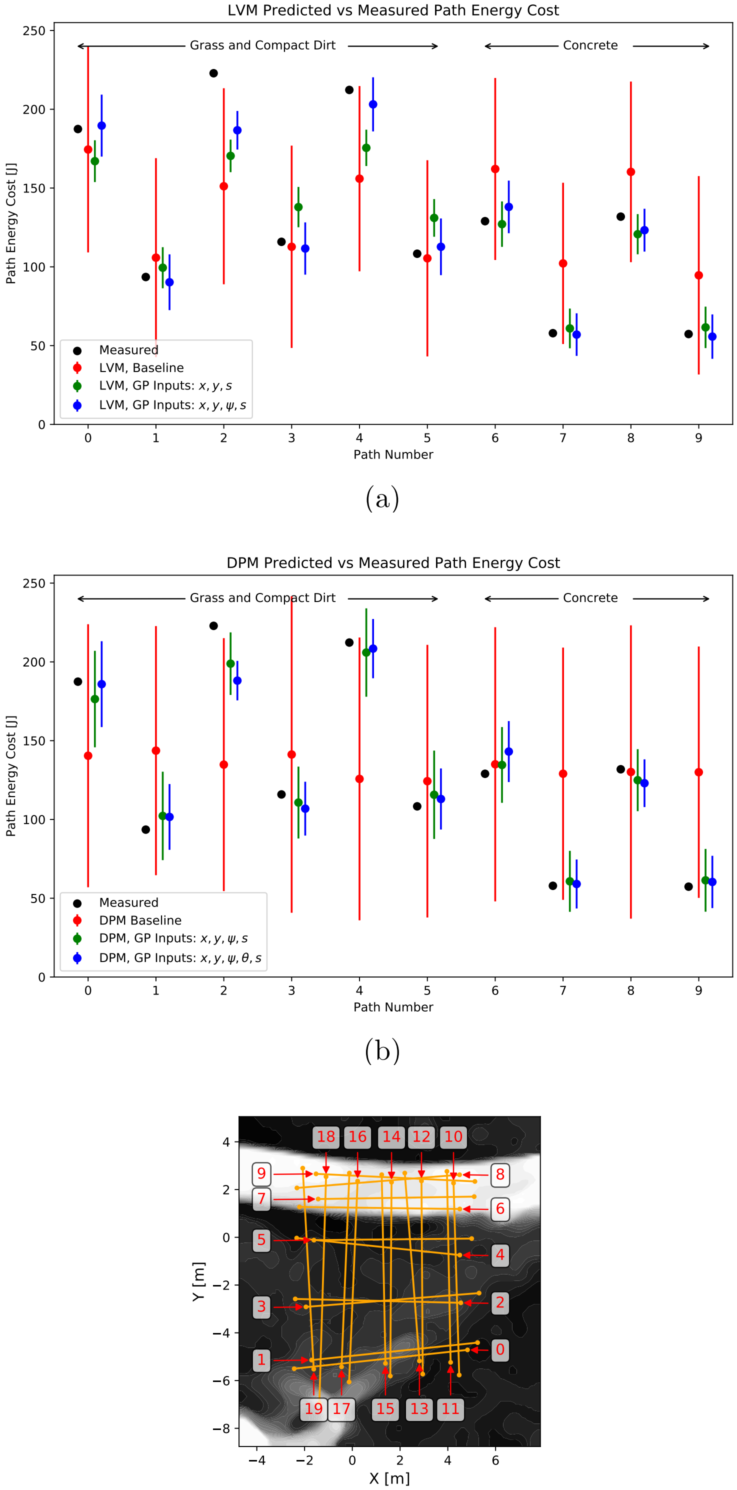
For the LVM with GP inputs x, y, s, shown in Fig. 3.4a energy cost predictions for Paths 6, 7, 8, and 9 are quite accurate and within the uncertainty bars. On the other hand, Paths 2, 3, 4, and 5 show more significant errors and overconfidence in the predictions. Paths 2 and 4 are uphill on the grass terrain, whereas paths 3 and 5 are downhill along the same lines as 2 and 4, respectively. These inaccuracies are caused by model errors (due to the LVM model) which result in rolling resistance coefficient estimates that differ depending on whether the robot is going uphill or downhill on the same path.

The performance when the GP input includes ψ is also shown in Fig 3.4a. Including ψ allows that the effective rolling resistance coefficient experienced by the robot may depend on the direction that the robot is travelling on the terrain. By fitting a more expressive GP model from the data, the errors of paths 2, 3, 4, and 5 have been reduced in comparison to the case with inputs x, y, s. Performance is improved, however, there is still overconfidence in Path 2.

The DPM model avoids an explicit model entirely by building the power model simultaneously with the spatial mapping from the training data set. The prediction results, shown in Fig. 3.4b, are provided for the case of GP inputs x, y, ψ, s and x, y, ψ, θ, s. The x, y, ψ, s case has the smallest error and least overconfidence with respect to Path 2.

The results for the cross-validation are collated in terms of percent error and PLL in Fig. 3.5. In particular, the figure shows the importance that including specific inputs to the GP has on predictive performance. Further results on the mean absolute percent error and the mean PLL over all 20 paths are shown for different kernels in Table 3.2 and Table 3.3.

We draw the following conclusions from these results (and from the results on the



(a)

(b)

(c)

Figure 3.4: Path energy costs: (blue/green) predicted cost $\bar{E}$ with $\pm 2\sqrt{\text{var}(E)}$) bars, (black) measured cost, and (red) baseline prediction (± 2 std.). The path numbers corresponded to the numbers in (c). Note that the environment is on a slope, resulting in different costs between, for example, paths 0 and 1. Models used: (a) LVM, (b) DPM.

GP		Kernel			
Model	Inputs	Ma1/2	Ma3/2	Ma5/2	SE
DPM	Baseline	27.38	27.60	28.52	26.87
	x, y, s	28.24	28.30	28.36	28.44
	x, y, ψ, s	5.86	4.49	4.44	5.04
	x, y, θ, s	7.00	6.98	6.95	8.09
	x, y, ψ, θ, s	5.18	4.56	4.62	5.09
LVM	Baseline	14.51	15.13	14.69	15.31
	x, y, s	7.88	7.68	7.63	7.58
	x, y, ψ, s	3.88	3.82	3.80	3.81
	x, y, θ, s	6.94	6.25	5.98	6.44
	x, y, ψ, θ, s	4.03	3.94	3.90	3.87

Table 3.2: Mean absolute percent error in path energy cost prediction (best performance highlighted green).

GP		Kernel			
Model	Inputs	Ma1/2	Ma3/2	Ma5/2	SE
DPM	*Const.*	-5.29	-5.34	-5.32	-5.25
	x, y, s	-8.45	-8.58	-8.62	-8.67
	x, y, ψ, s	-3.76	-3.57	-3.54	-3.64
	x, y, θ, s	-5.06	-5.71	-5.91	-6.39
	x, y, ψ, θ, s	-3.98	-4.12	-4.21	-4.31
LVM	*Const.*	-4.83	-4.88	-4.82	-4.84
	x, y, s	-7.65	-7.62	-7.62	-7.59
	x, y, ψ, s	-4.15	-4.11	-4.11	-4.13
	x, y, θ, s	-6.57	-6.59	-6.52	-6.59
	x, y, ψ, θ, s	-4.28	-4.23	-4.22	-4.24

Table 3.3: Mean PLL in path energy cost prediction (best performance highlighted green).

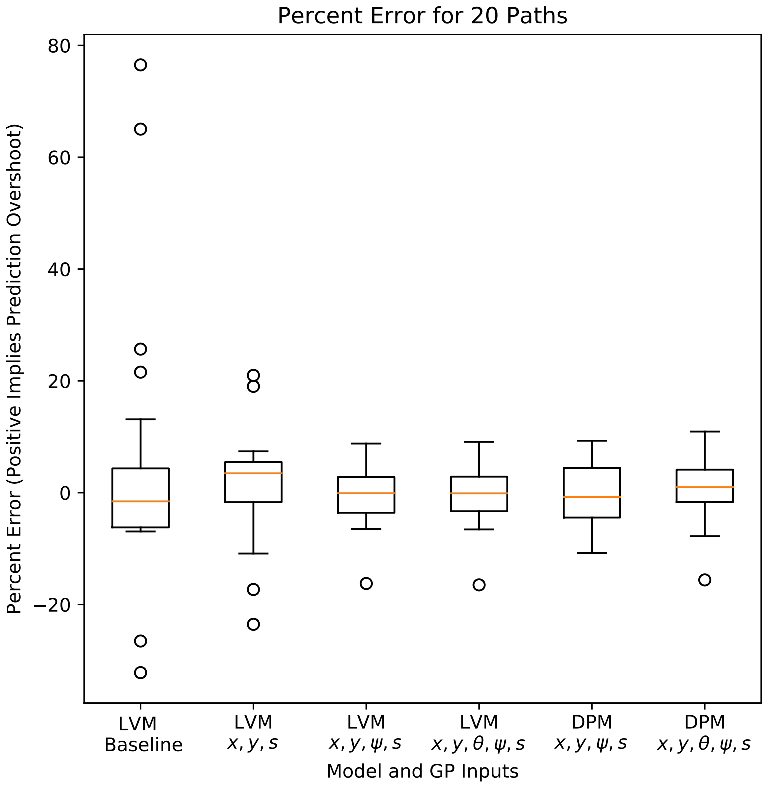


Figure 3.5: Percent error for the 20 path cross-validation with different models and GP inputs. The boxplot show the baseline LVM prediction result (baseline DPM has much higher error), as well as results when using different inputs to the GP (e.g. x, y, s, or x, y, θ, ψ, s, etc.).

additional dataset in the Appendix):

1. The spatial mapping methodology for both LVM and DPM significantly improves path energy prediction accuracy over the baseline approach of using the LVM and a constant rolling resistance coefficient over the whole environment.
2. The robot's heading ψ has a greater impact on improving performance than the slope θ. The slope is clearly a factor in power consumption, and of course, the slope depends on the heading of the robot. However, the heading may account for additional factors, including: the robot's roll, weight distribution, unequal tire pressures or directional terrain effects (e.g. grass being pointed in a particular direction). Such factors are particularly evident in the small ground robot platform in these experiments, but also appear in the results for the larger robot, seen in the Appendix.

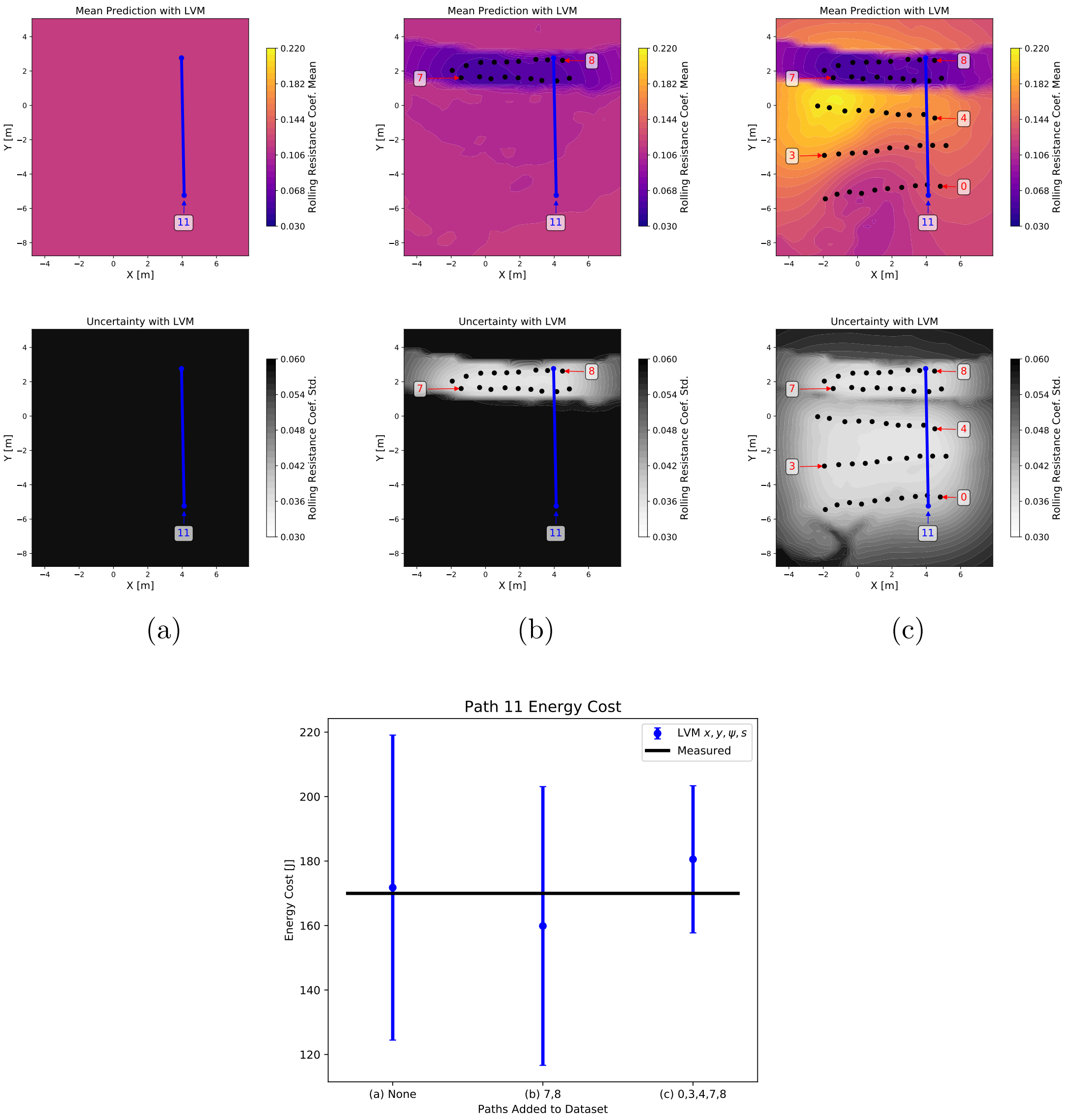
Furthermore, the results show that the spatial mapping methodology can improve predictive performance in terms of PLL, but important caveats exist. Uncertainty predictions by GPR are especially dependent on hyperparameter selection. Hyperparameters here were optimized with respect to the log marginal likelihood. The optimization did not account for correlated errors that occur when the robot is traversing along a path, leading to overconfidence in some cases. Hyperparameter selection is addressed further in Section 3.3.5. Related to hyperparameter selection is kernel choice, in which minor variations were found in accuracy based on kernel smoothness. In particular, the Ma1/2 had slightly worse prediction accuracy.

3.3.4 Path Energy Prediction with Sparse Data

We next observe predictive performance when data on the environment is sparse. Path energy cost prediction with sparse data can have significant uncertainty (i.e. a large variance $\mathrm{var}(E)$). As more data is collected on the environment, that uncertainty will be reduced. This fact is demonstrated with an example, shown in Fig. 3.6, in which the hyperparameters from Table 3.1 were used. The cost of a path is predicted repeatedly as more data is added to the data set. Initially, there is a large uncertainty, as given by the dark spatial uncertainty map in Fig. 3.6a and the ±2 standard deviation bars in Fig. 3.6d. As more data is added from previously traversed paths, the spatial uncertainty map is reduced (Figs. 3.6b and 3.6c), and the uncertainty bars in the predicted cost of the path narrow around the measured cost of the path. The example also highlights the effect of including satellite imagery as a GP input. In Fig. 3.6b, only data from the concrete path has been included in the data set, resulting in low uncertainty along the path defined by the satellite image.

The LVM and DPM models were compared in the sparse data case for a series of individual path predictions. The results of predicting path energy costs for Paths 2, 7, and 11 are shown in Fig. 3.7, in which data from previously traversed paths is added, demonstrating a reduction in uncertainty. The variance is significantly greater for the DPM because the LVM provides basic model structure that is helpful in making predictions, even without prior data. On the other hand, the DPM is reliant on the constant prior mean of the GP and must wait for data to inform further model structure. The predictions for Path 2 were overconfident for both models, implying that more conservative hyperparameters or robust modeling are necessary to account for the highly irregular nature of off-road terrain. Further examples are provided in Fig. A.1 in the Appendix.

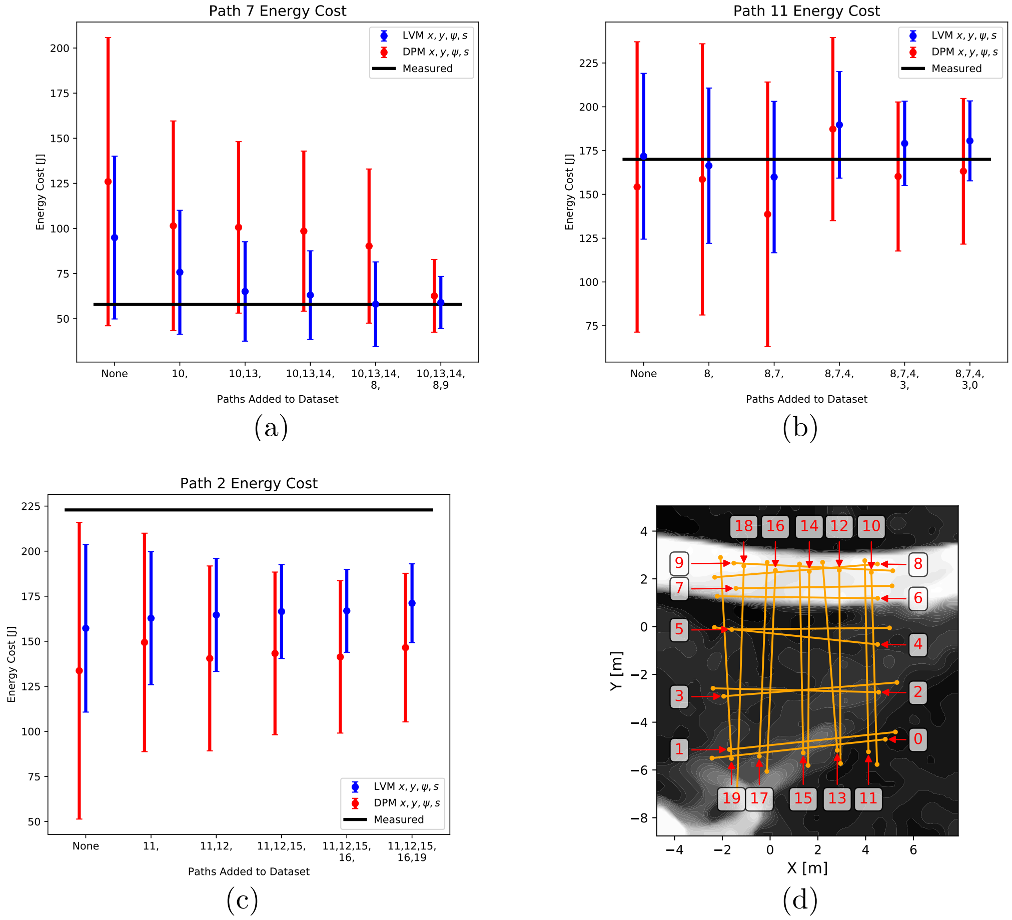
A model based on physical parameters, such as the LVM, is especially useful for informing predictions if a physical parameter (e.g. speed, weight, etc.) changes. More expressive models could also be extended to include other considerations, such as turning. However, models like the LVM ignore complex physics. For example, rolling resistance coefficient estimates tend to change with vehicle speed [27]. One potential solution is to specify model structure with explicit basis functions [84, 85].



(a) (b) (c)

(d)

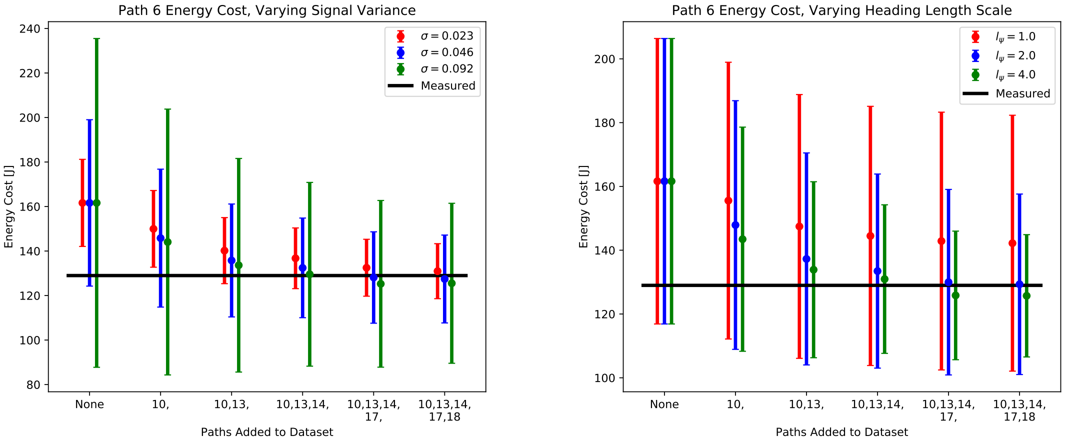
Figure 3.6: The objective is to predict the energy cost the blue line (Path 11). As data from more paths is added, from column (a) to column (c), the mean GP prediction is updated and the uncertainty is reduced. Red arrows point to the start of each path of added data, and data points themselves are black dots. (d) The predicted energy cost corresponding to each column (with ±2 std. uncertainty bars) is shown in blue and the measured cost of Path 11 is provided by the black line.



(a) (b)

(c) (d)

Figure 3.7: Predicted path energy costs for LVM (blue) and DPM (red) models as data from traversed paths is added. The average PLL over the 6 predictions in each model are (better performance is in bold): (a) Path 7; **LVM: -3.86**, DPM: -5.22 (b) Path 11; **LVM: -4.03**, DPM: -4.45 (c) Path 2; LVM: -11.73, **DPM: -9.34**. (d) Numbered paths. More examples are shown in the Appendix.



(a) (b)

Figure 3.8: Effect of hyperparameters on the prediction of Path 6 using the LVM with GP inputs x, y, ψ, s: (a) signal variance σ^2, (b) heading length-scale l_ψ.

Parameters of the model can then found in conjunction with hyperparameters through log marginal likelihood optimization.

3.3.5 Impact of Hyperparameter Selection

Performing log marginal likelihood optimization with sparse data can lead to overfitting and poor predictions. If a diverse dataset is unavailable to support reliable hyperparameter optimization, a user must specify hyperparameters based on previous experience or knowledge of the system. Fig. 3.8 shows an example of how predictions from the LVM are affected by varying the signal variance σ^2 and heading length-scale l_ψ hyperparameters. A larger σ^2 directly results in a larger variance because it implies that there is a larger range over which energy costs in the environment are expected to vary. In contrast, if l_ψ is small, then only data that have a similar heading as the predicted path will significantly reduce variance. If, for example, the robot traverses the same path multiple times, then the prediction will have high confidence.

Additional hyperparameters, including the prior mean c, noise variance σ_η^2, and spatial length-scale l_{xy}, play a role in predictions as well and are detailed in Fig. A.2 in the Appendix. Based on these results, we provide the following guidelines for tuning hyperparameters:

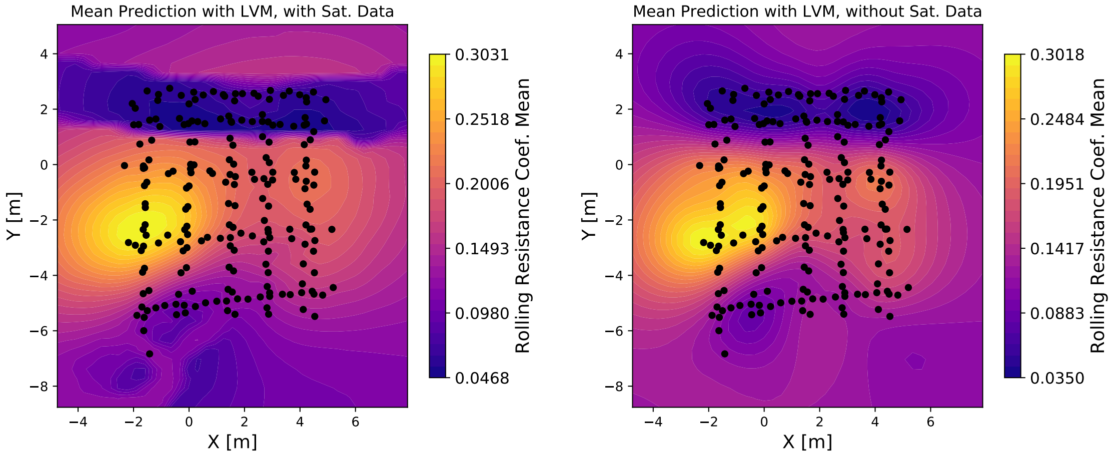
1. Set a larger signal variance σ^2 and smaller heading length-scale l_ψ for more conservative (higher variance) predictions.
2. Set a higher than expected prior mean c if robot energy depletion is a concern. Data will override the prior mean once it has been collected.

When determining hyperparameters under sparse data, it can also be effective to set informative prior distributions on the hyperparameters and then select values for the hyperparameters, given the collected data, using maximum a posteriori (MAP) [117] or Markov chain Monte Carlo (MCMC) methods. [34]. MCMC methods are computationally expensive, but can be useful for determining hyperparameters off-line.

3.3.6 Impact of Satellite Imagery

The effect of including a satellite image as an input to the GP was observed in Fig. 3.6. To expand on this, Fig. 3.9 provides a direct comparison of the spatial mapping, both with and without the imagery input to the GP. Without the satellite imagery, the discontinuity is not well captured, potentially leading to inaccurate predictions near the boundary.

Another important consideration is that, to fit the transition between grass and concrete, the optimized l_{xy} hyperparameter is smaller for the case without satellite imagery. Including the satellite image enables a larger l_{xy} because the transition is accounted for through the additional input dimension s. A larger l_{xy} can imply that less data on the environment is needed to inform predictions. While this experiment was performed in a small environment, a much larger length-scale would be necessary for the energy prediction methodology to be effective in a large-scale environment. Thus, satellite imagery or other spatial maps of the area would be particularly useful



(a) (b)

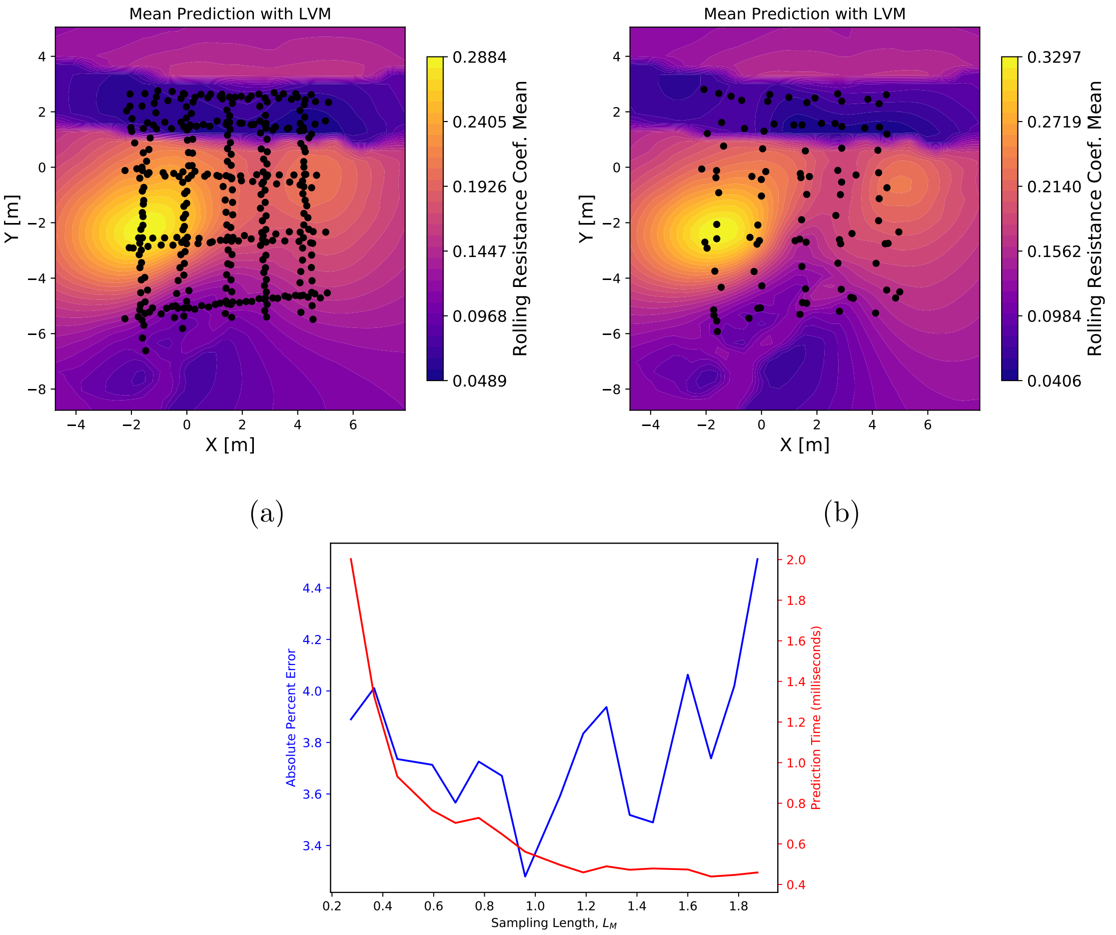
Figure 3.9: Spatial maps built (a) with the satellite image input (optimized $l_{xy} = 3.88$[m]) and (b) without the image input (optimized $l_{xy} = 2.53$[m]).

in helping to increase the length-scale through detection of known transitions between terrains.

3.3.7 Computation Time

The effectiveness of the decimation strategy described in Section 3.2.5 was evaluated by observing the accuracy of predictions against the computation time involved in path energy cost prediction as the sampling length L_M is reduced. The computation time shown in Fig. 3.10 is specifically for path energy prediction (but does not include hyperparameter optimization time). The computation times and errors are averaged over the cross-validation of the 20 predicted paths.

The results in Fig. 3.10 show that errors remain consistent, with small fluctuations, as L_M increases, and computation time decreases significantly. A very small L_M does not necessarily improve performance, but certainly increases computation time. However, for $L_M > 1.75$ there is a notable increase in the error. The results demonstrate that decimation can be used to reduce the computational burden in-



(a)

(b)

(c)

Figure 3.10: Spatial maps generated from data sets using sampling lengths (a) $L_M = 0.46$[m] and (b) $L_M = 1.69$[m]. Data points are given by black dots. (c) GP prediction time and mean absolute percent error vs. sampling length L_M.

volved in GPR with limited effect on predictions. Furthermore, numerous effective strategies exist for managing the computational cost of GPR, including using local approximations [110], achieving sparsity through subsampling the dataset [52], or Hilbert space approximations [98].

3.4 Conclusions

In this chapter, a GPR-based methodology was presented for using spatial mapping to predict energy costs in environments with varying terrain. The method was demonstrated experimentally to improve the accuracy of predictions over a baseline approach. Furthermore, the advantage of using vehicle modeling was demonstrated through reduced uncertainty in predictions in sparse data scenarios. Path energy cost predictions based on standard log marginal likelihood optimization were over-confident in some cases, however, we provided guidance for selection of more conservative hyperparameters. Additional factors, that could be accounted for through further modeling, also led to over-confidence in predictions, such as uncertainty in the path itself. Off-road terrain is often highly unstructured and irregular, leading to outliers in power consumption data. A discussion of future work for addressing the ongoing challenges of energy prediction in off-road environments is provided in Chapter V.

CHAPTER IV

Power Prediction for Heterogeneous Ground Robots through Spatial Mapping and Sharing of Terrain Data

In the previous chapter, experimental results were presented for spatial mapping and energy prediction in 3-d environments. The results demonstrated that, for a single robot, spatial mapping can be used to improve predictive accuracy for energy costs. In this chapter, we extend the applicability of spatial mapping for energy prediction to multiple heterogeneous ground robots. The research demonstrates how multi-task Gaussian process (MTGPs) can be used to effectively transfer information on the terrain between different robots, reducing error in predictions. The work in this chapter is primarily based on [78].

As discussed in the Introduction to this dissertation, one drawback of these methods is that the spatial maps are particular to a single robot, and often do not apply directly to a different robot or to the same robot under different operating conditions (e.g. tire pressure, weight distribution). However, information collected by one robot may still be useful for another robot. It is expected that, for robots of a similar type, a terrain that is higher cost for one robot will also be higher cost for the other. In other words, the costs are *positively correlated.* Such correlations can be used to inform power predictions for one robot, based on data collected by another robot, as well as to speed up spatial mapping.

To address the problems of spatial mapping and correlation between multiple robots' power consumption, we propose using multi-task Gaussian process regression (MTGP) [12, 114, 30]. Previous spatial terrain mapping approaches with a single robot, such as the authors' previous work [80], are equivalent to single-task Gaussian process regression (STGP). A MTGP extends the STGP to consider similarities between tasks (or outputs). MTGPs are a multi-output regression method originating in the field of geostatistics [111]. A wide variety of applications have used MTGPs, including learning robot manipulator inverse dynamics with different loads [114] and modeling correlated physiological signals [30]. For a review of multi-output methods that use kernels, see [4].

While MTGPs are highly effective at improving performance when there are correlations between multiple tasks, they suffer from computational limitations. The time complexity of prediction and hyperparameter optimization for MTGP is $O(M^3N^3)$, where M is the number of tasks and N is the number of data points for each task. Furthermore, increasing the number of tasks causes a proliferation in the number of hyperparameters. Without effective measures to handle these problems, MTGPs will not be applicable to real-time robotics applications.

To counter computational problems, we develop a framework of efficient hyperparameter optimization and prediction for the application of multi-robot power prediction in environments with spatially varying costs. The literature on managing Gaussian process complexity is rich (see Liu et al. for a review of related methods [49]). Thus, our framework builds on effective strategies from previous work in ways that are particular to the application of predicting power consumption along a path. First, we assume that all robots collect an *isotopic* training dataset, in which the robots share the same input data. In practice, this implies that all robots initially traverse the same path through the environment as a calibration step. The assumption of isotopic data allows the use of efficient hyperparameter optimization through eigenvalue decompositions [100, 83].

Isotopic data cannot be assumed beyond calibration, as the robots must be able to explore different parts of the environment for MTGP to be advantageous. Therefore, in the *heterotopic* case, in which different robots have data from different locations in the environment, we use a Subset of Data approach (SoD) [49] as an approximation to reduce computation time. The most likely scenario for which power predictions will be used is in predicting the energy cost of paths (perhaps for path planning or mission planning). To support planning, we provide a local approximation strategy that uses a k-dimensional tree (KD tree) to efficiently find a set of nearest neighbor points in order to predict path costs with little reduction in performance.

4.0.1 Contributions

This chapter builds on Chapter III, which presented a methodology for spatial mapping and path energy cost prediction using GPR in the single robot case. We make the following contributions to the case of multiple heterogeneous ground robots:

- A framework, based on multi-task Gaussian process regression, for predicting ground robot power consumption on spatially varying terrain through shared terrain information between robots, including:
  - Efficient training of hyperparameters through isotopic data collection.
  - Efficient prediction of power consumption along paths through nearest neighbor data selection.
- A simulation study demonstrating the effectiveness of the framework.
- An experimental demonstration of increased power prediction accuracy through shared information between a large and small ground robot.

The chapter is organized as follows: In Section 4.1, the methods for power prediction using MTGPs are described. Next, these methods are unified into an efficient framework in Section 4.2, with a corresponding derivation for the log marginal likelihood in Appendix C. Simulation results for the case of many robots are provided in Section 4.3. Experimental results demonstrating the feasibility of the approach two robots are detailed in Section 4.4. Finally, conclusions and future work are discussed in Section 4.5.

4.1 Methods

In this section, the longitudinal parametric power model is described and the MTGP formulation is provided. Further, we provide the log likelihood function used for selecting hyperparameters in the isotopic data case.

4.1.1 Robot Power Model

As in Chapter III, we focus on longitudinal motion and do not consider turning, though it is important to note that turning can be a large factor in energy costs and could be accounted for through additional modelling [27]. In a deviation from Chapter Chapter III, we propose using an explicit basis function data-fit model [84], the form of which is inspired by physical modeling. The parameters of the model of learned through marginal likelihood optimization, rather than using physical parameters, as done in Chapter III.

Let $P, \theta, \eta \in \mathbb{R}$ denote the power, slope in the direction of the robot's heading, and model error, respectively. Let $\mathbf{x} \in \mathbb{R}^d$ be set a set of d inputs. In this chapter, x, y, $s(x, y)$, and ψ are used as inputs, where x, y is the spatial position, $s(x, y)$ is a grayscale imagery pixel intensity at a given position, and ψ is the robot's heading. A latent terrain function $f : \mathbb{R}^d \to \mathbb{R}$ is a learned map from the inputs to a terrain cost.

The power model is:

$$P = f(\mathbf{x}) + \eta + \mathbf{g}(\theta)^T \mathbf{c} \tag{4.1}$$

where $\mathbf{g}(\theta)$ is a vector of basis functions (e.g. polynomials) and $\mathbf{c}$ is a vector of model coefficients to be estimated. While the coefficients may be selected based on the physical parameters of a vehicle model, we choose to treat $\mathbf{c}$ as hyperparameters to be learned from data. Note: the desired quantity for planning is typically energy, whereas power is energy per unit time. The energy cost of a path must take into account the speed of the robot. Therefore, implicit in this work is that power is measured at a known speed.

To learn the latent terrain function f from data, a map is built from input points $\mathbf{x}_i$ and slope θ_i to power P_i. The data for N points is then collected for the inputs $\mathbf{x}_i$ as $X \in \mathbb{R}^{N \times d}$, the slope $\Theta \in \mathbb{R}^N$ and for the power $\mathbf{P}$ as $\mathbf{P} \in \mathbb{R}^N$. The data set is denoted $\mathcal{D} := \{X, \Theta, \mathbf{P}\}$.

For standard STGPs, let $f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$, where $c(\mathbf{x})$ is the prior mean and $k(\mathbf{x}, \mathbf{x}')$ is a symmetric, positive semidefinite kernel. The kernel is used to define the covariance matrix between inputs in a dataset, $K(X, X) \in \mathbb{R}^{N \times N}$, where $[K(X, X)]_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $K(X, X)$ is positive semidefinite. For this chapter, we consider the Matern class of kernels with an automatic relevance determination (ARD) parameterization [84]. The ARD parameterization allows a length-scale hyperparameter for each input dimension, which we define as $l_1, \ldots, l_d > 0$. For extensive details on Gaussian process regression, see Rasmussen [84].

4.1.2 Multitask GPR for Multirobot Spatial Power Prediction

The formulation for the MTGP is provided for the case of M tasks/robots. We begin with the *heterotopic* case, in which the input data for each task may be different. Similar MTGP formulations can be found in [12, 30], however, ours differs slightly in that we do not assume that two tasks share the same exact inputs (as this is unlikely in the spatial mapping applications).

MTGPs center around the definition of a positive semidefinite matrix $\mathbf{B} \in \mathbb{R}^{M \times M}$, that we call the *task matrix*. The task matrix computes the scaling and correlation between two tasks. There are many ways of defining the task matrix, including the *free-form* parameterization [12] and the *spherical* parameterization [72, 68, 85]. The free-form parameterization uses the Cholesky decomposition, $\mathbf{B} = LL^T$ where $L \in \mathbb{R}^{M \times M}$ is a lower triangular matrix. Unfortunately, the number of parameters increases as $M(M+1)/2$, making log likelihood optimization difficult. As suggested in [12], we use a rank-T approximation, where $\hat{\mathbf{B}} = \hat{L}\hat{L}^T$ where $\hat{L}$ is an $M \times T$ matrix.

The subscripts of (1) and (M) are used to identify first and Mth tasks, respectively. The data for tasks 1 to M are collected as $\mathcal{D}_{(1)}, \ldots, \mathcal{D}_{(M)}$, where $N_{(1)}$ and $N_{(M)}$ are the number of data points for tasks 1 and M respectively.. The full covariance matrix, considering the data sets from both tasks, is defined as:

$$\mathbf{K}_{MT} := \begin{bmatrix} [\mathbf{B}]_{1,1} K(X_{(1)}, X_{(1)}) & \ldots & [\mathbf{B}]_{1,M} K(X_{(1)}, X_{(M)}) \\ \vdots & \ddots & \vdots \\ [\mathbf{B}]_{M,1} K(X_{(M)}, X_{(1)}) & \ldots & [\mathbf{B}]_{M,M} K(X_{(M)}, X_{(M)}) \end{bmatrix} \tag{4.2}$$

where $K(X_{(M)}, X_{(1)})$, for example, defines the covariances for the input data of tasks 1 and M. $\mathbf{K}_{MT}$ defines the covariances for every input point within *and* between each task, as scaled by $\mathbf{B}$. Importantly, if the off-diagonals of $\mathbf{B}$ are 0, then the covariances between each dataset is 0, resulting in no dependence (and no information transfer) between the datasets for use in predictions.

Based on the $\mathbf{K}_{MT}$, a power prediction for task $j \in \{1, \ldots, M\}$, denoted $\mathbf{P}_{(j),*}$, given a set of inputs X_* and slopes Θ_*, can be computed similarly to standard GPR [12]. For the following, $\mathbf{I}_{N \times N}$ denotes an N by N identity matrix. Let N_* denote the number of target points. The noise variances associated with each task are collated as $\Sigma := \text{blockdiag}(\sigma^2_{(1),\eta}\mathbf{I}_{N_1\times N_1}, \dots, \sigma^2_{(M),\eta}\mathbf{I}_{N_M\times N_M})$ and

$$\bar{\mathbf{Z}} := \Big[\mathbf{P}_{(1)} - \mathbf{G}_{(1)}(\Theta_{(1)})\mathbf{c}_{(1)} \quad \dots \quad \mathbf{P}_{(M)} - \mathbf{G}_{(M)}(\Theta_{(M)})\mathbf{c}_{(M)}\Big] \tag{4.3}$$

where $\mathbf{G}_{(M)}(\Theta_{(M)})$ is the stacked basis functions $\mathbf{g}$ for each $\theta_i \in \Theta$.

The probability distribution for the power prediction is then:

$$p(\mathbf{P}_{(j),*}|X_*, \Theta_*, \mathcal{D}_{(1)}, \dots, \mathcal{D}_{(M)}) = \mathcal{N}\big(\bar{\mathbf{P}}_{(j),*}, \text{cov}(\mathbf{P}_{(j),*})\big) \tag{4.4}$$

where the predictive mean is

$$\begin{aligned} \bar{\mathbf{P}}_{(j),*} &:= \mathbb{E}[\mathbf{P}_{(j),*}|X_*, \Theta_*, \mathcal{D}_{(1)}, \dots, \mathcal{D}_{(M)}] \\ &= \mathbf{G}_{(j)}(\Theta_*)\mathbf{c}_{(j)} + \mathbf{K}_{MT*}\big(\mathbf{K}_{MT} + \Sigma\big)^{-1}\text{vec}(\bar{\mathbf{Z}}) \end{aligned} \tag{4.5}$$

where $\text{vec}(\bar{\mathbf{Z}})$ is the vertical stacking of the columns of $\bar{\mathbf{Z}}$ and

$$\mathbf{K}_{MT*} := \begin{bmatrix} [\mathbf{B}]_{1,1}K(X_*, X_{(1)}) \\ \vdots \\ [\mathbf{B}]_{1,M}K(X_*, X_{(M)}) \end{bmatrix}^T$$

computes the covariance between the target point $\mathbf{x}_*$ and both data sets. The predictive variance is

$$\text{cov}(\mathbf{P}_{(j),*}) = \text{cov}(f_{(j)}(X_*)) + \sigma^2_{(j),\eta}\mathbf{I}_{N_*\times N_*} \tag{4.6}$$

where

$$\text{cov}(f_{(j)}(X_*)) = [\mathbf{B}]_{j,j}K(X_*, X_*) - \mathbf{K}_{MT*}\big(\mathbf{K}_{MT} + \Sigma\big)^{-1}\mathbf{K}^T_{MT*}. \tag{4.7}$$

4.1.3 Hyperparameter Selection

For hyperparameter optimization, we consider the *isotopic* data case in which $\bar{X} := X_{(1)} = \cdots = X_{(M)}$ and N is the number of data points. For the isotopic case, the full covariance matrix is expressed concisely as

$$\bar{\mathbf{K}}_{MT} := \mathbf{B} \otimes K(\bar{X}, \bar{X}) \tag{4.8}$$

where $\otimes$ is the Kronecker product. Let $\bar{\Sigma} := \mathrm{diag}(\sigma_{(1)}, \ldots, \sigma_{(M)})$. A common way to select hyperparameters is through optimizing the log marginal likelihood [84], which in the isotopic multi-task case is [83]:

$$\begin{aligned} L = &-\frac{NM}{2}\ln(2\pi) - \frac{1}{2}\ln|\bar{\mathbf{K}}_{MT} + \bar{\Sigma} \otimes \mathbf{I}_{N\times N}| \\ &- \frac{1}{2}\mathrm{vec}(\bar{\mathbf{Z}})^T(\bar{\mathbf{K}}_{MT} + \bar{\Sigma} \otimes \mathbf{I}_{N\times N})^{-1}\mathrm{vec}(\bar{\mathbf{Z}}). \end{aligned} \tag{4.9}$$

The inversion of $\bar{\mathbf{K}}_{MT}$ has time complexity of $O(M^3N^3)$, quickly leading to intractable optimizations. For the case of isotopic data, methods have been developed to reduced this complexity through eigenvalue decomposition and properties of the Kronecker product [83]. For completeness, we include a modified version of the result in [83] that has been simplified for this case. A full derivation is provided in Appendix C. Let $\tilde{\mathbf{B}} = \bar{\Sigma}^{-\frac{1}{2}}\mathbf{B}\bar{\Sigma}^{-\frac{1}{2}}$ and let the eigenvalue decompositions be given by $\tilde{\mathbf{B}} = U_{\tilde{B}}S_{\tilde{B}}U_{\tilde{B}}^T$ and $K(\bar{X}, \bar{X}) = U_K S_K U_K^T$. Then

$$\begin{aligned} L = &-\frac{NM}{2}\ln(2\pi) - \frac{1}{2}\ln|S_{\tilde{B}} \otimes S_K + \mathbf{I}_{MN\times MN}| - \frac{N}{2}\ln|\bar{\Sigma}| \\ &- \frac{1}{2}\mathrm{vec}(U_K^T\tilde{\mathbf{Z}}U_{\tilde{B}})^T(S_{\tilde{B}} \otimes S_K + \mathbf{I}_{MN\times MN})^{-1}\mathrm{vec}(U_K^T\tilde{\mathbf{Z}}U_{\tilde{B}}) \end{aligned} \tag{4.10}$$

where $\mathrm{vec}\tilde{\mathbf{Z}} = \mathrm{vec}(\bar{\mathbf{Z}}\Sigma^{-\frac{1}{2}})$. The resulting time complexity is significantly reduced, at $O(M^3 + N^3)$. Furthermore, gradients of Eqn. (4.10) can be found for efficient hyperparameter optimization [83].

4.2 Multi-Robot Power Prediction Framework

The overall framework for predicting power consumption along paths with data from multiple heterogeneous robots is now described. The following assumptions are made:

Assumption IV.1. *Perfect communication between all robots.*

Assumption IV.2. *Input data, including* $\boldsymbol{x}$ *and slope* θ*, are known for training and testing data.*

While there is significant research on dealing with poor communication between robots, such work is not our focus, leading to Assumption IV.1. Assumption IV.2 is valid with accurate robot position and if the topography of the environment is known, for example, through LiDAR or Digital Elevation Maps.

There are two major steps in the framework:

1. *Optimization of hyperparameters*: An isotopic data set is collected, through all robots traversing the same path in an environment. The power model, kernel, and task matrix hyperparameters are found through maximizing the log marginal likelihood, in Eqn. (4.10). The rank-T approximation of the task matrix is used, $\hat{\mathbf{B}} = \hat{L}\hat{L}^T$, where T is selected on a case-by-case basis to trade off computation time and performance.

2. *MTGP-NN power prediction*: A Subset of Data (SoD) approach is used, in which the nearest neighboring points to a given candidate path are selected, with heterotopic data from all robots. The points are then used to compute the MTGP predictive distribution for power consumption along the path of test points for a given robot, using Eqns. (4.4)-(4.7). The details of this approach, which we call MTGP-NN, are provided in Algorithm 2.

Algorithm 2: MTGP-NN (Step 2)

```
Input  : $\mathcal{D}_{(1)} \cup \cdots \cup \mathcal{D}_{(M)}$: data from all robots
         $X_{\mathcal{P}}$: Set of candidate paths
         $k$: Number of nearest neighbors per test point
         $l_1, \ldots, l_d$: Input length-scales
         $j \in \{1, \ldots, M\}$: Robot for power prediction
Output: $\mathcal{P}$: Set of power predictions for candidate paths
$\mathcal{D} \leftarrow \mathcal{D}_{(1)} \cup \cdots \cup \mathcal{D}_{(M)}$
$\mathcal{P} \leftarrow \{\}$
$\mathcal{T} \leftarrow$ KDTree($\mathcal{D}, l_1, \ldots, l_d$) % Construct KD Tree using standardized euclidean distance with all data
for $X_* \in X_{\mathcal{P}}$ do
    $\mathcal{D}_p \leftarrow \{\}$
    for $\boldsymbol{x}_i \in X_*$ do
        $\mathcal{D}' \leftarrow$ FindNN($\mathcal{D}, \mathbf{x}_i, k$) % Find $k$ nearest neighbors for each test point
        $\mathcal{D}_* \leftarrow \mathcal{D}_* \cup \mathcal{D}'$ % Collect unique data points
    % Predict power with Eqns. (4.5),(4.7)
    $\bar{\mathbf{P}}_{(j),*}, \mathrm{cov}(\mathbf{P}_{(j),*}) \leftarrow$ MTGPPred($\mathcal{D}_*, X_*$)
$\mathcal{P} \leftarrow \mathcal{P} \cup \{\bar{\mathbf{P}}_{(j),*}, \mathrm{cov}(\mathbf{P}_{(j),*})\}$
return $\mathcal{P}$
```

Hyperparameter optimization and prediction are typical for Gaussian process applications; however, the details of how we perform these steps are particular to the application of multi-robot power prediction along paths.

Step 1 serves as a calibration step for which the hyperparameters needed for MTGP are determined. To enable efficient optimization, as described in Section 4.1.3, the data collected is assumed to be isotopic. In general, the robots are likely to traverse different locations in the environment, leading to heterotopic data sets. However, an initial calibration data set, consisting of isotopic data in which all robots traverse the same path, is a reasonable condition for effectively and efficiently learning hyperparameters.

Step 2 addresses the scalability of power prediction along a path when considering large data sets. In this case, data between robots is heterotopic, so the efficient computations used in [83] are no longer applicable. However, there has been significant research into handling large data sets when working with GPs [49]. One simple but effective approach is to perform a local approximation, aided by a KD tree [110]. A KD tree is a data structure that partitions points in a space, allowing for efficient nearest neighbor search [76]. The approach provided in [110] finds the training points nearest to a testing point and uses that subset of the training data for prediction.

For robotics applications such as path planning, it is often desirable to predict costs along a path. Algorithm 2 provides a simple extension of the nearest neighbor local approximation approach for paths. In summary, using a KD tree, a set of k nearest neighbor training points to each of the m testing points along a candidate path (including data from all robots) are used to predict the cost of that path. To determine nearest neighbors, a *standardized* euclidean distance is used to account for the possibility of the d inputs in $\mathbf{x}$ having different length scales. To perform this, the input data is normalized based on the previously optimized length scales of the kernel, $l_1, \ldots, l_d$.

The resulting time complexity for MTGP prediction in Algorithm 2 is, at worst, $O(k^3m^3)$. However, the actual computation time is typically much lower since many of the training data points found through nearest neighbor search are redundant. Additionally, the average time complexity for KD tree search is $O(d \log NM)$, where d is the input dimension and NM is the size of the data set from all robots. While constructing the tree is slower, at $O(dNM \log NM)$, it need only be done once before a path planning step, and then many paths can be tested based on that tree.

Finding nearest neighbors along the entire path, rather than just for individual test points, is an important step, as it enables the computation of the joint distribution of power predictions, provided in Eqn. (4.4). The joint distribution describes the strong correlations between power predictions along a path. Such correlations are necessary for computing the uncertainty in the total energy cost of the path [81].

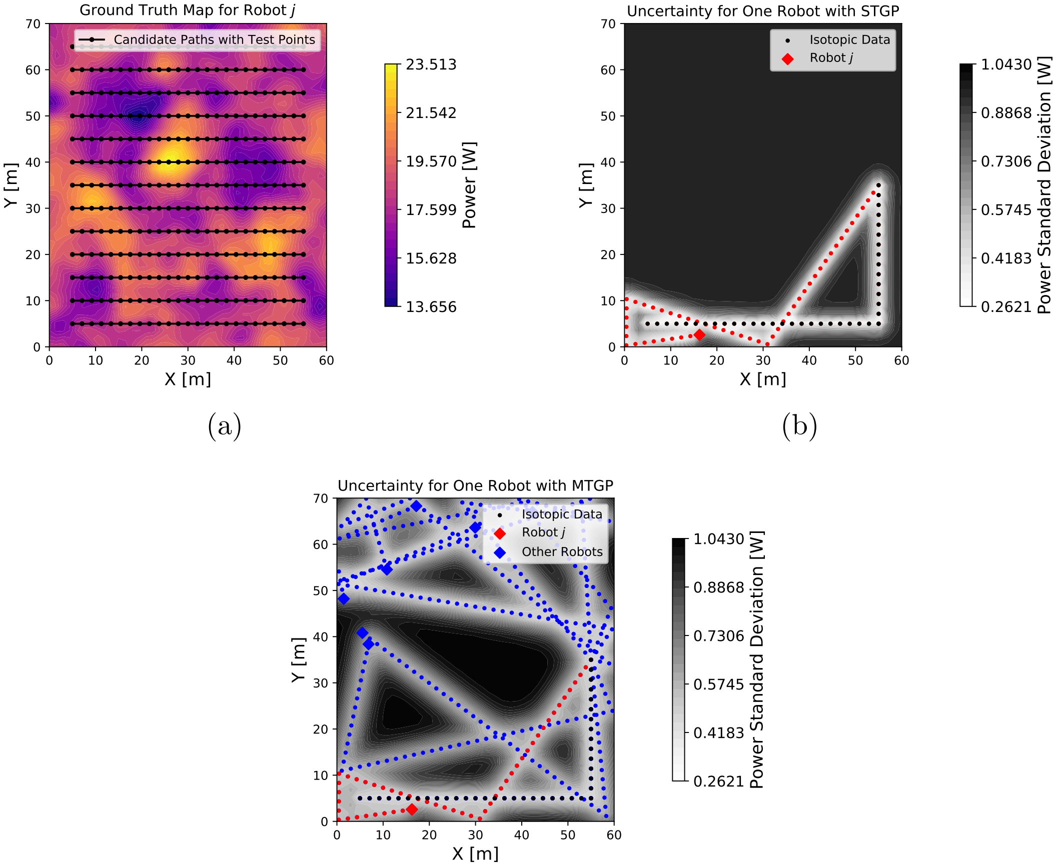
4.3 Simulation Results

The multi-robot power prediction framework is first evaluated in simulation for the purpose of comparing computation time and performance of predictions between STGP, MTGP, and the efficient approximation: MTGP-NN.

For this simulation, there are $M = 8$ robots that traverse a flat environment ($\Theta = 0$) for which we let $\mathbf{G}(\Theta)\mathbf{c}_{(j)} = c_{(j)}$, where j is the robot number and $c_{(j)}$ is a randomly selected constant power. The noise level, $\sigma^2_{(j),\eta}$, for each robot is also selected randomly. The environment is generated from a random sampling of an MTGP parameterized by a randomly generated task matrix, $\mathbf{B}$, and a Matern 5/2 kernel [84]. The ground truth map for one robot is shown in Fig. 4.1a (the ground truth maps are different for each robot). The noise hyperparameters are selected randomly and are different for individual robots. The inputs are $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$ with the length scale hyperparameter, $l_{xy} = 5.0$, which is the same for all robots. These parameters, including $\mathbf{B}$ are all unknown to the robots, and must be estimated from data.

As described in the framework, all robots initially collect an isotopic data set by traversing the same path through the environment, shown by the black dots in Fig. 4.1c. Once this data has been collected, $\hat{\mathbf{B}}$ (using a rank-$T = 2$ approximation) and the remaining hyperparameters are found using the isotopic data set and optimization with respect to Eqn. (4.10). The isotopic optimization took 7.64 seconds. In comparison, without the efficient approach used in Eqn. (4.10), the optimization took 26.24 seconds and achieved the same result. Following the isotopic data set collection and hyperparameter optimization, the robots separate and follow random paths through the environment, collecting a heterotopic power consumption data set that is used for predictions.

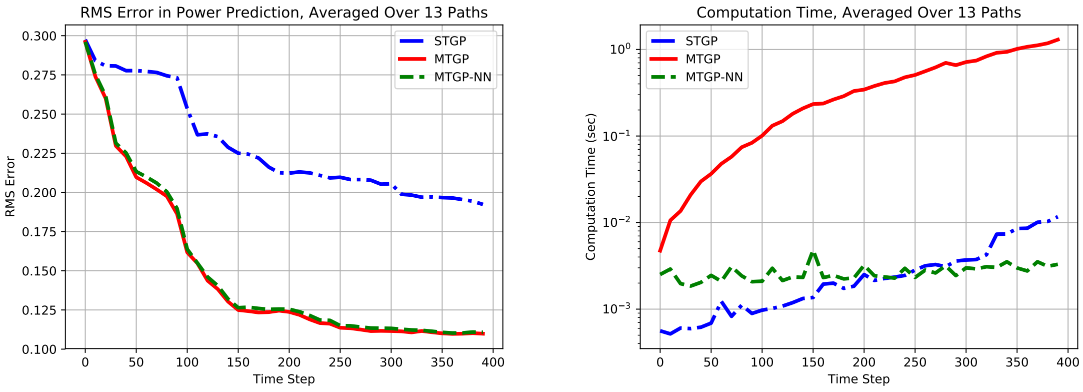
To evaluate performance in a uniform manner, a set of 13 candidate paths through the environment is defined, shown in Fig. 4.1a. The power predictions are evaluated



(a) (b)

(c)

Figure 4.1: (a) (a) The ground truth map of power consumption for one robot, robot j, and the set of 13 candidate test paths (black dots and lines) used for evaluating performance. (b), (c) The predictive uncertainty for the power consumption of robot j after 50 time steps using (b) STGP with data from just robot j and (c) MTGP with data from all robots. All robots initially traverse the same path (black dots) and the collected isotopic data (black dots) is used for hyperparameter optimization. The robots then move in random directions for further heterotopic data collection.



(a) (b)

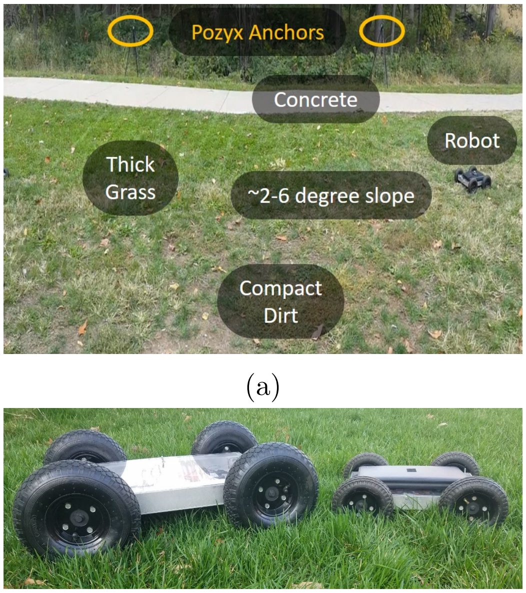
Figure 4.2: Simulation results for (a) root mean squared (RMS) power prediction error and (b) computation time. Results are averaged over 13 predicted paths through the simulated environment.

for one robot over the same set of candidate paths throughout the heterotopic data collection phase. The results, shown in Fig. 4.2, are provided for 3 cases:

1. STGP: the robot uses only its own data for power predictions.
2. MTGP: all data from all of the robots is used.
3. MTGP-NN: a subset of data from all the robots, selected with Algorithm 2 and a setting of $k = 10$ nearest neighbors, is used.

Both MTGP and MTGP-NN have similarly good performance, with low error seen in Fig. 4.2a. As expected, the performance improves quickly for both cases, as it takes little time for 8 robots to explore the environment. For the STGP, the robot only uses data that it has collected and ignores data from the other robots. Thus, significantly more time is needed to reduce error. However, once the robot has explored the environment, similar performance would be achieved.

While the error for MTGP and MTGP-NN are similar, the computation time is vastly improved for MTGP-NN, as seen in Fig. 4.2b. The computation of MTGP increases rapidly, due to matrix inversion of a data set that increases in size by MN



(a)

(b)

Figure 4.3: (a) Experiment environment with small robot. (b) Large (10 inch wheels and 15.15kg mass) and small robots (6 inch wheels and 7.25kg mass) used for experiments.

at every time step. In contrast, the computation time of MTGP-NN increases slowly, due to the KD tree build and search times. MTGP-NN uses the data relevant to candidate path predictions much more efficiently.

4.4 Experimental Results

4.4.1 Experimental Setup and Data Preprocessing

The MTGP-NN strategy was tested on the experimental dataset used in Chapter III. The make use of the dataset from the small robot shown in Fig. 4.3b, as well as the dataset from the large robot. To review, the two robots traversed 20 straight-line paths through the environment seen in Fig. 4.3a. The robots were remotely controlled by a user to move along the predefined path. The position of the robots was measured by an absolute localization system (Pozyx NV, Ghent, Belgium) using 4 anchors placed around the environment. A Pozyx tag was placed on the robot, allowing for both tracking and measurement of the attitude angle (roll, pitch and yaw) through an inertial measurement unit.

The sampling frequency of the system was 10Hz, which leads to a rapidly increasing number of data points. To address this problem, the data was decimated (filtered and downsampled), as described in Chapter III, to a desired "sampling length," which was set to be 0.8[m]. This preprocessing step greatly reduces the dimensionality of the data, with little loss of useful information.

4.4.2 MTGP Setup and Hyperparameter Optimization

For the MTGP, a Matern 5/2 kernel was used again, and the inputs were $\mathbf{x} = \begin{bmatrix} x & y & \cos(\psi) & \sin(\psi) & s(x,y) \end{bmatrix}^T$ where $s(x,y)$ is the pixel intensity of a satellite input at position x, y and ψ is the heading of the robot, which is projected onto the unit circle. Let l_{xy}, l_ψ, $l_s > 0$ be the length scales associated with inputs. During optimization, l_s was held at 0.3. For the power model, let $\mathbf{g}(\theta) = \begin{bmatrix} 1 & \theta \end{bmatrix}^T$. Since there are 2 robots, we let $\mathbf{B} = LL^T$ with no approximation.

For the results presented here, the hyperparameters are optimized using a separate data set that was collected similarly to the data set shown in Fig. 4.4a, but on a different day. The optimization problem was small enough to allow for heterotopic marginal likelihood optimization.

The optimized hyperparameters are shown in Table 4.1. Interestingly, for the task matrix $\mathbf{B}$, the optimal correlation between the tasks was found to be very high (≈ 1.0). However, this is countered by the high noise level for the large robot, which limits the effect of information transfer between robots.

4.4.3 Spatial Mapping

A qualitative comparison is provided for the spatial maps produced using standard STGPs with the data collected by both robots. Figure 4.4 depicts the predictive mean

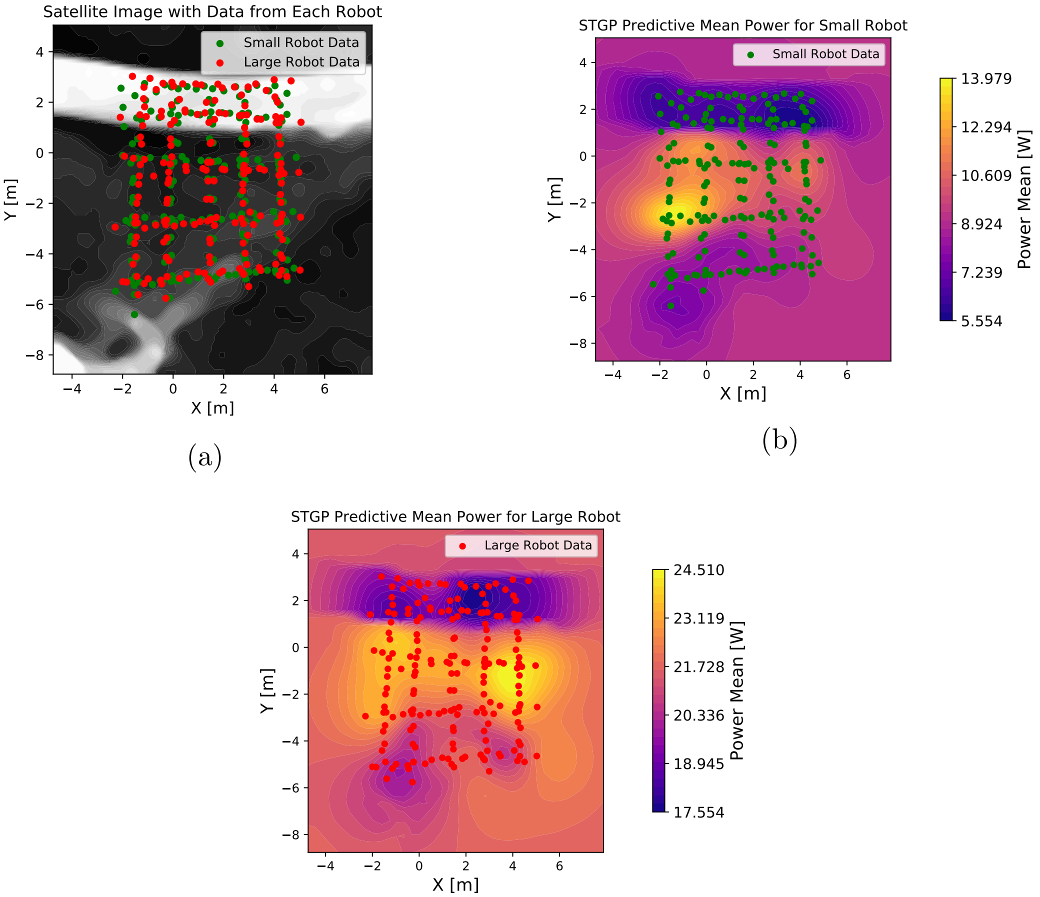
Hyperparameter	Large	Small
l_{xy}	1.47	1.47
l_s	0.3	0.3
l_ψ	4.45	4.45
$\mathbf{c}_{(j)}$	$\begin{bmatrix}22.15\\83.58\end{bmatrix}$	$\begin{bmatrix}9.10\\39.31\end{bmatrix}$
$\sigma^2_{(j),\eta}$	3.85	0.45
$\mathbf{B}$	$\begin{bmatrix}4.15 & 3.53\\3.53 & 3.01\end{bmatrix}$	

Table 4.1: Optimized hyperparameters for MTGP

of power over the environment for both robots. Clearly, the concrete section of the environment is less costly to traverse than the grass and compact dirt areas. Both Figs. 4.4b and 4.4c have similar trends (i.e. a higher cost terrain for one robot is likely to be higher cost for the other robot). This fact indicates that the robots are good candidates for MTGP through their strong correlations. Furthermore, it is interesting to note that while the large robot consumes much more energy than the smaller robot, the ratio between the highest and lowest predicted power is ≈ 1.2 for the large robot and ≈ 1.9 for the small robot. In other words, the small robot is much more sensitive to the changes in the terrain than the large robot. However, the differences in the terrain are still noticeable, in terms of power consumption, for the large robot.

4.4.4 MTGP-NN Evaluation

The performance of MTGP-NN is now evaluated in terms of the accuracy of power predictions. For the following results, $k = 20$ nearest neighbor points were used for Algorithm 2. First, Fig. 4.5 shows the credible intervals for power predictions in both the MTGP-NN and STGP cases. In Fig. 4.5a, the power is predicted for the small robot, using only data from the large robot's mapping of the environment. Similarly, Fig. 4.5b shows the predicted power for the large robot, only using data from the small robot. In both cases, MTGP-NN appears to reduce uncertainty over

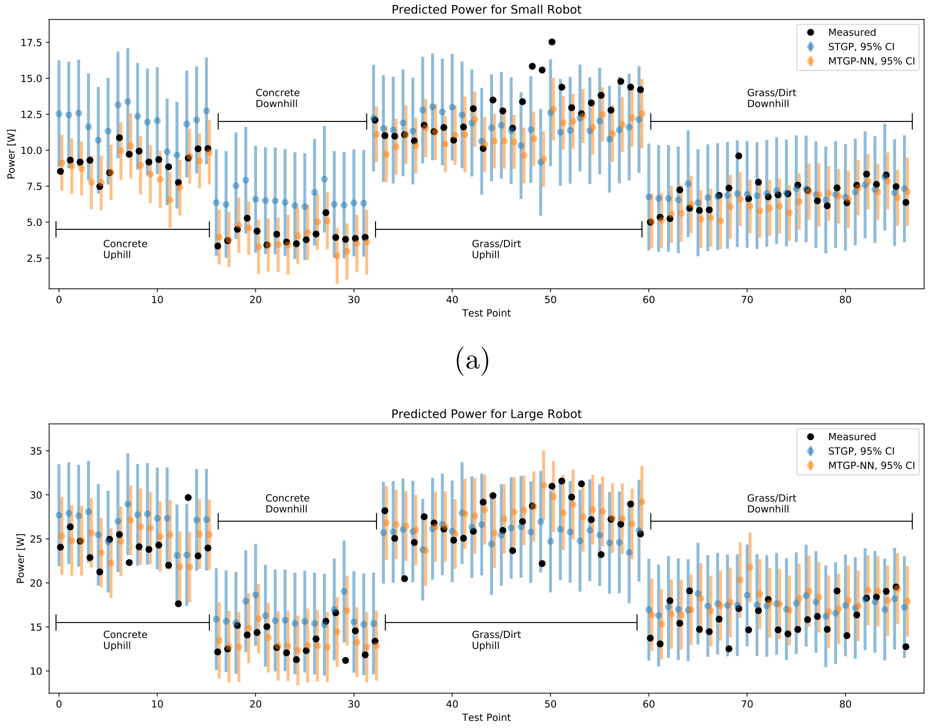


(a)

(b)

(c)

Figure 4.4: (a) Top-down grayscale satellite image used to define the s(x,y) input. Data collected on both the small (green) and large (red) robots are overlaid. (b),(c) STGP predictive mean over the environment with $\theta = 0$ using data from the (b) small robot and the (c) large robot.



(a)

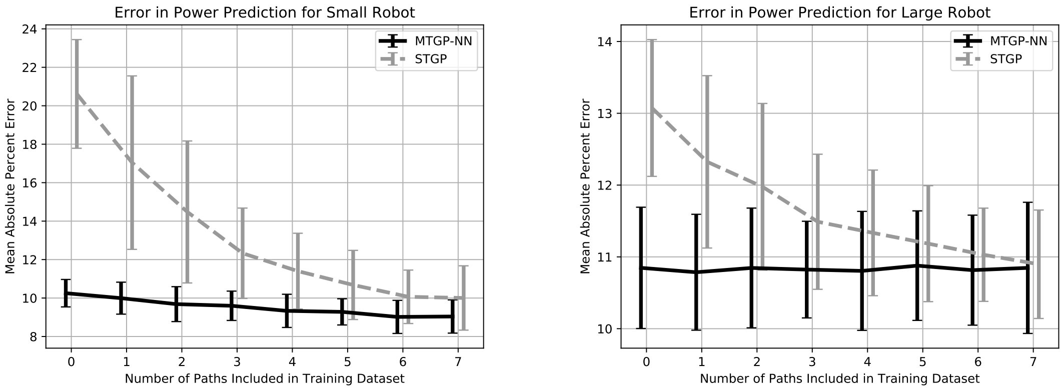
(b)

Figure 4.5: Power prediction 95% credible intervals (CI) for (a) small robot using only data from the large robot and (b) vice versa. Test points are given by black dots. Note that data are from 10 separate paths for each robot and have been concatenated together.

STGP. There are outlier test points that are not well captured by either STGP or MTGP-NN.

A more quantitative comparison is provided in Fig. 4.6. The following case is evaluated: predictions are made for robot A, given all the data on the environment from robot B, and varying levels of data from robot A. The following testing procedure was used to evaluate the aggregate performance of STGP and MTGP-NN over varying levels of data:

1. A set of training paths (ranging from 0 to 7 paths) were randomly selected from the small robot's data set to be included in predictions.
2. A set of 10 testing paths from the remaining data set of robot A were ran-



(a) (b)

Figure 4.6: Power prediction error as more training data is added, with ±1 standard deviation over 100 randomized tests. (a) Error for the small robot, in which MTGP-NN uses the entire data set from the large robot, and the number of paths included from the small robot's data set increases. The STGP case uses only data from the small robot. (b) Error for the large robot.

domly selected and the STGP and MTGP-NN power predictions were compared against this testing set.

3. Steps 1 and 2 were repeated 100 times for each "number of paths" in the x-axis of Figs. 4.6a and 4.6b.

The testing procedure allows evaluation of predictive performance specifically with respect to paths. Each randomized test took approximately 0.01 seconds of computation time for MTGP-NN.

The results in Fig. 4.6 clearly demonstrate that, through MTGP-NN, using data from the large robot improves performance, particularly when the small robot has less data from the specific environment. However, as the robot collects more data (the number of paths traversed in the environment increase), the difference in performance between STGP and MTGP-NN is smaller. This results shows the utility of information sharing when data on the environment is limited.

The same testing procedure was repeated for making predictions with the large robot, using the small robot's data set. Corresponding results are provided in Fig.

4.6b. While the improvement in performance for MTGP over STGP is clear, the difference is less significant than for the small robot. This follows from the fact that the large robot is less impacted by variations in the terrain, and the power difference between concrete and grass is less significant than for the small robot.

4.5 Conclusions and Future Work

In this chapter, a multi-robot power prediction framework for heterogeneous ground robots, based on MTGP, was proposed. Simulation results demonstrate the improved performance and efficiency of the approach in scaling the number of robots and data points. Experimental results provided a demonstration that the sharing of data between robots can significantly improve prediction accuracy, particularly when one robot has little data on the environment.

One major challenge with modeling off-road terrain is its irregularity, sometimes leading to brief spikes in power consumption (e.g. hitting a rock that was previously missed by the robot). This challenge motivates the development of further robust methods that better handle outliers, such as using a Student-t process [94], rather than a Gaussian process. Additionally, our work assumes that all robots can traverse over the same parts of the environment, there are scenarios in which some robots may be able to traverse terrain that other robots can not. Handling such a scenario would provide an interesting and useful extension of the presented framework.

Heterogeneous robot predictions also have fascinating implications in planning. For example, the large robot used in the experiments is less impacted by variations in the terrain. It can however, detect such variations and inform the small robot of terrain that is likely to be very costly for it. Under the MTGP framework, the resulting predictions could be naturally incorporated into planning. Planning strategies that take advantage of this can improve performance over the multi-robot system.

CHAPTER V

Conclusions and Future Work

5.1 Conclusions

Energy is a central aspect of robot platforms and must be considered in both robot design and operation. During operation, such as in missions of reconnaissance or exploration, energy requirements are highly uncertain due to the complex nature of environments. Energy cost prediction and, subsequently, planning strategies must account for the significant uncertainty in off-road environments. To do this, methods that take advantage of both prior knowledge and data collected during operation must be developed.

This dissertation has contributed methods that begin to address these problems through spatial mapping, reachability, and multi-robot information sharing. In summary, we have shown that:

1. Spatial mapping of collected power consumption data can be used to significantly improve the accuracy of path energy cost predictions, while accounting for uncertainty in predictions.

2. Path energy cost predictions can be used to improve the computation of a robot's reachability in an environment with significant uncertainty.

3. Terrain cost information can be shared between heterogeneous robots in order to improve predictive accuracy and reduce the time needed to map an environment.

One important conclusion of this work is that, to make accurate predictions with uncertainty, correlations in costs must be considered. Predicting the total cost of a path involves the summation of individual costs along that path. Many prediction and stochastic planning approaches consider those individual costs to be independent. However, that assumption can lead to very overconfident predictions. Furthermore, overconfidence often results in poor decision-making (in robotics applications and beyond). By considering that costs along a path are correlated, a more reasonable quantification of uncertainty can be obtained.

Many other factors can lead to poor predictions. A common dilemma is deciding what modeling approach to use on the spectrum of purely physical to purely data-driven. There is, of course, no approach that is best for every system. Data-driven modeling can be advantageous when the physical system is complex and poorly understood. However, a physical model can effectively provide a general form for the data-driven model to fit. Otherwise, the data-driven model may require significant amounts of data to be effective. In the case of off-road energy prediction, a data-driven approach (Gaussian process regression) was used for spatial mapping, while both physical and data-driven models were considered for vehicle specific factors. While the physical model required much less data to make more accurate predictions, it also tended to be more over-confident than the data-driven model. Computation time is also a major consideration in model selection, particularly in robotics, where learning is expected to be fast. For example, a data-driven approach may be more flexible, however, many such methods require significant time for training, with few guarantees on convergence.

5.2 Real-Time System Implementation

The energy prediction strategy presented in this dissertation was evaluated off-line with experimentally collected data. Implementation on an online, real-time system, requires that the dataset be effectively managed to avoid excessive computation time.

As discussed previously, the computation time of GPR prediction grows as $O(n^3)$, where n is the number of data points. Without proper data management, prediction of energy costs would quickly become infeasible as a robot collects more data. There is a significant body of research on GPR with large data sets [49] which could aid in this problem. In Chapter IV, we described a strategy specific to making efficient path energy cost prediction. In short, the strategy uses a subset of the data comprised of the nearest neighbors to the candidate path. The number of data points included in the prediction can be tuned to trade-off computation time and accuracy. Simulation results in Chapter IV, Fig. 4.2, showed that computation time for a candidate path using this strategy is quite low (under 0.003 seconds), with similar error as using the whole dataset. Theoretical bounds on error and computational complexity for a related strategy of truncating the dataset have been shown by Xu et al. [118]. In practice, our path energy cost prediction strategy could be applied to robotic path planning via commonly used graph-based algorithms such as RRT* [42] or A*.

Local subsets of the full data set are particularly useful for predicting the energy cost of a given path. Some path planning strategies, however, are based on a pre-computed set of cells on a costmap [62] in which a full map of the environment, given all the data, must be maintained. The computation time for the full spatial map of the experimental environment, provided in Chapter III, Fig. 3.3a, was about 0.26 seconds. To update the map in real-time, cells in the costmap can be recomputed as more data on energy costs is collected. To avoid unnecessary computation, only cells that are close to the new data should be updated, and those cells should be updated with a local subset of data.

A further challenge is that data can accumulate rapidly if the sampling rate of the system is high. Chapter III provides a decimation strategy in which the data is filtered and downsampled to a desired sampling length, before use in training and prediction. The sampling length can be set by a user, for example, based on the size of the robot. In practice, this has allowed for a significant reduction in computation time with little impact on performance.

5.3 Future Directions

There are several interesting directions that the research presented in this dissertation could take, both with respect to the theory and applications of spatial mapping for energy prediction, reachability, and multi-robot power prediction. Several areas of active research currently exist that could benefit from the contributions of this dissertation. A few of such directions are provided here:

- *Metrics other than energy:* While this dissertation focused on energy, the methods developed have implications for other metrics, such as time or risk of detection by an adversary. In some scenarios, completing a mission or task in minimum time is more important than minimum energy. However, a robot's maximum speed is dependent on the terrain. A methodology similar to our energy prediction methodology may be used to predict the time needed to traverse a path, or the time-limited reachability of the robot.

- *Non-Gaussian distributions:* A rigorous quantification of uncertainty in robotics will inevitably crash into computational problems. While methods such as Markov chain Monte Carlo are very effective for understanding probability in off-line applications, applications in robotics typically require real-time predictions and decision-making. The work in this dissertation has largely focused on using Gaussian process regression as a tool for tractable analytical prediction. Due to their analytical advantages and reasonable approximation of the data, Gaussians are commonly used in stochastic planning problems [47, 21]. However, the unstructured nature of off-road environments makes outliers in the data likely. Further research could explore using robust methods of handling outliers. For example, the Student-t distribution and the Student-t process [94] have heavier tails than the Gaussian distribution, allowing for better modeling of outliers.

- *Robotic mission planning:* Informative path planning (IPP) is a type of robotic planning in which a robot must efficiently gather information, possibly subject to a budget (such as energy or time) [10, 119, 105]. IPP is an active area of research, with numerous formulations and solutions for single and multi-robot problems. However, the budget is typically used in a known, deterministic manner. For example, many formulations constrain the robot by distance travelled. By expanding the reachable space using data collected during the mission, as demonstrated in Chapter II, better performance than naïve distance-based approach could be achieved in an IPP task, while maintaining robustness. The direct application of these methods to IPP problems remains future work, and the results and conditions of performance improvements have not yet been explored.

- *Multi-robot planning:* The results in Chapter IV provide a framework for power prediction with multiple heterogeneous robots. Many robotic mission planning algorithms assume that robots are homogeneous. Robot heterogeneity, however, is a growing area of interest due to practical necessity and the utility of robots with different capabilities [58, 112, 86, 91, 54]. Under the framework of Chapter IV, robot heterogeneity can be an advantage in robotic mission planning algorithms. The framework could be used to enable automatic replanning if, for example, a large robot discovers terrain that is likely to be very costly for a small robot. Resources (or robots) could subsequently be allocated more efficiently based on shared information.

- *Aerial and marine robots:* In addition to ground robots in off-road environments, there are similar energy dependent applications for autonomous marine robots [101, 102] and aerial robots [24]. For marine and aerial applications, there are ocean currents and wind fields, respectively, that are spatially and temporally varying, as well as uncertain. Optimal planning under uncertainty is the subject of ongoing research in these applications [102], and the methods developed in this dissertation could be extended for flow fields. For example, consider a heterogeneous team of quadcopters performing a task. By spatially mapping a wind field through multi-task Gaussian processes, information about the field could be transferred between the robots. Furthermore, smaller quadcopters could avoid particularly windy areas that have been observed by a larger quadcopter, saving energy.

- *Vehicle modeling:* The work in this dissertation used a simple longitudinal vehicle model for understanding power consumption, whereas the research contributions focused on spatial mapping. However, the literature on the modeling of vehicle dynamics and power consumption is vast [115, 82]. Energy predictions based on spatial mapping are likely to benefit from thorough modeling research, especially by accounting for factors specific to the type of vehicle. For example, Salama et al. [89] provide longitudinal modeling for UGVs that accounts for tire slippage for the purpose of energy efficiency. Dogru and Marques [27] model turning on a skid-steered robotic platform. Additionally, larger slopes have impacts on energy consumption that are vehicle specific, e.g. regenerative breaking. The wealth of knowledge in vehicle modeling should be leveraged for future systems that perform spatial mapping and prediction.

- *Additional prior information:* Grayscale satellite imagery was used in this research to help inform discontinuities between different terrains. In general, however, there are many different sources of information that could be useful for more accurate energy prediction. Such sources could include full RGB data [62], or segmentation of the environment based on camera imagery [59]. Furthermore, soil moisture content is known to have an impact on wheel-terrain interaction, often leading to increased energy consumption. Prior information (such as on weather conditions) would likely improve predictions.

These future directions address both research questions and the necessary practical considerations for applying the methods presented in this dissertation to robotic systems being developed for real-world applications. It is both exciting and daunting to consider the difficult problems, in both theory and applied settings, that remain to enable robust operation of autonomous robots in realistic, complex environments.

APPENDICES

APPENDIX A

Path Predictions for Sparse Data and Varying Hyperparameters

This Appendix provides further results related to Chapter III through additional path predictions in the case of sparse data. The results are shown in Fig. A.1. The LVM maintains an advantage over the DPM in providing improved PLL, showing that the vehicle modeling can aid predictions when data is sparse.

Further results for varying hyperparameters are provided in Fig. A.2. Varying the prior mean c is most significant in its effect on the prediction with no data, as seen in Fig. A.2a. A higher prior mean (which is interpreted as the rolling resistance coefficient in the LVM case) results in higher predicted energy costs. As paths are added to the data set, however, the disparity between predictions with different prior means is reduced. Thus, the prior mean is extremely important is cases where data is very sparse. But the prior is quickly overwhelmed by the data.

The spatial length-scale l_{xy} has a nuanced impact on predictions, shown in Fig. A.2b. A higher l_{xy} increases the distance over which two GP inputs are highly correlated. The correlations effect the variance of the path energy cost prediction in Eqn. (3.14), specifically through summation over the off-diagonal elements of $\text{cov}(\mathcal{P}_*)$.

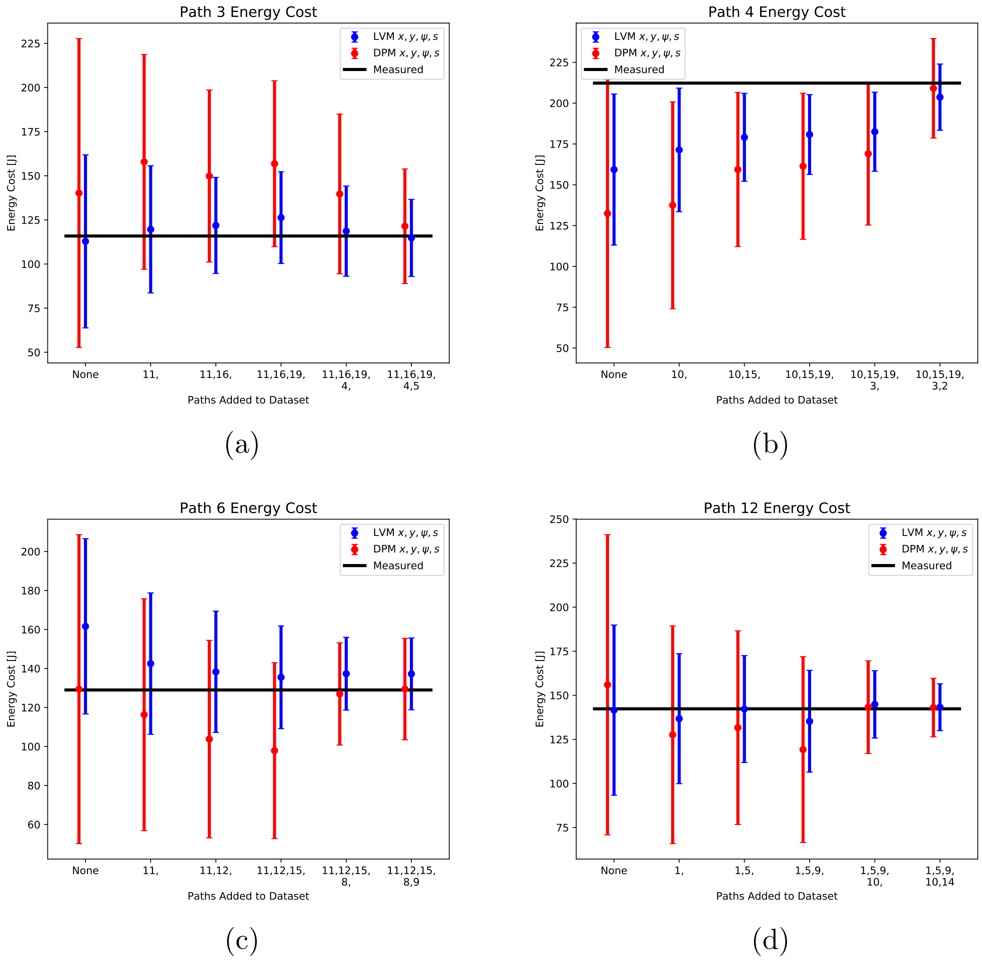
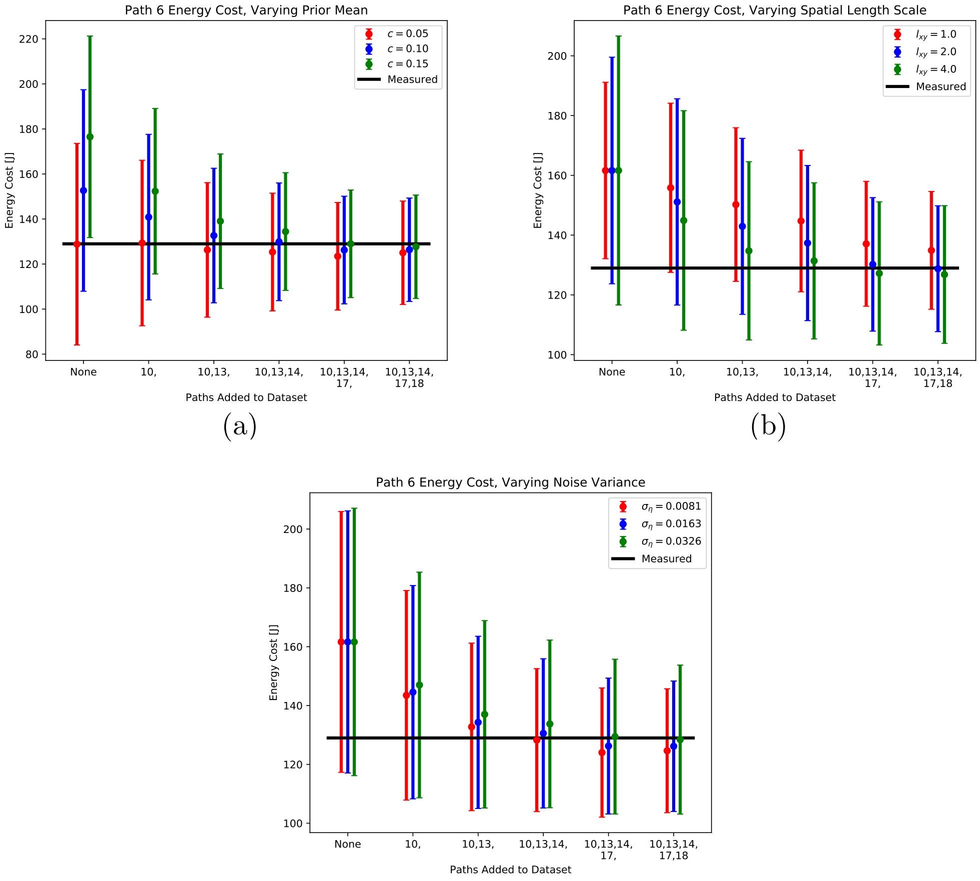


Figure A.1: Predicted path energy costs for LVM (blue) and DPM (red) models as data from traversed paths is added. The average PLL over the 6 predictions in each model are (better performance is in bold): (a) Path 3; **LVM: -3.70**, DPM: -4.86 (b) Path 4; **LVM: -6.04**, DPM: -6.09 (c) Path 6; **LVM: -3.96**, DPM: -4.27 (d) Path 12; **LVM: -3.56**, DPM: -4.10

When no paths have been added to the data set, the higher correlation results in increased uncertainty. However, the uncertainty is reduced when path 10 is added to the data set. The larger l_{xy} increases the distance over which path 10 provides information, causing a greater change in the mean prediction over smaller l_{xy} values. Varying the other length-scale hyperparameters (l_s, l_θ is mathematically similar to varying l_{xy} and l_ψ). The practical effects of doing so may be different and are not explored here.

A higher noise variance σ_η^2 also increases uncertainty in predictions, as shown in Fig. A.2c. Increased uncertainty happens in two different ways: First, by reducing the impact of new data in the GP variance prediction of Eqn. (3.7). Second, by increasing the uncertainty of the predicted power, as in Eqn. (3.11).



(a) (b)

(c)

Figure A.2: Effect of hyperparameters on the prediction of Path 6 using LVM. (a) The prior mean c, (b) spatial length-scale l_{xy} (c) noise variance σ_η^2.

APPENDIX B

Path Energy Prediction Results on a Large Robot

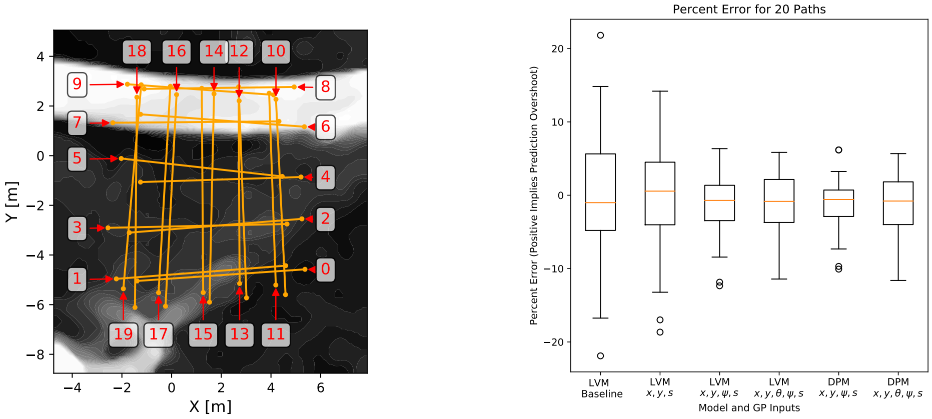
This Appendix provides a further set of experimental results related to Chapter III. A similar experiment to the experiment described in Chapter III was performed in the same environment on a larger robot (SuperDroid Robots, IG42-SB4-T, 4WD). The results of the leave-one-path-out cross-validation are shown in Fig. B.1b and Table B.2. The spatial length-scale hyperparameter was bounded from above during optimization as $l_{xy} \leq 10$. This is due to relatively small variations in terrain costs for the large robot, leading to a large l_{xy} under log marginal likelihood optimization. As with the small robot, the spatial mapping methodology shows an improvement over the baseline energy prediction. The improvement is somewhat less drastic than for the small robot because the large robot was less impacted by variations in the terrain, leading to more accurate baseline LVM predictions.

Parameter	Small Robot	Large Robot
u	$0.457\ \frac{\mathrm{m}}{\mathrm{s}}$	$0.435\ \frac{\mathrm{m}}{\mathrm{s}}$
m	7.25 kg	15.15 kg
a_{accel}	$0\ \frac{\mathrm{m}}{\mathrm{s}^2}$	$0\ \frac{\mathrm{m}}{\mathrm{s}^2}$
C_I	9.0 N	30.0 N
b_{int}	0.76 W	1.11 W

Table B.1: Physical parameters used for LVM

GP		Kernel	
Model	Inputs	Mean abs. % error	Mean PLL
DPM	Baseline	19.08	-5.86
	x, y, s	20.14	-9.05
	x, y, ψ, s	3.14	-4.32
	x, y, θ, s	5.83	-5.21
	x, y, ψ, θ, s	3.65	-4.39
LVM	Baseline	7.14	-5.13
	x, y, s	6.70	-6.28
	x, y, ψ, s	3.86	-4.58
	x, y, θ, s	5.53	-5.34
	x, y, ψ, θ, s	3.87	-4.50

Table B.2: Mean absolute percent error and mean PLL in path energy cost prediction (best performance highlighted green). Results are shown for Ma1/2.



(a) (b)

Figure B.1: (a) Grayscale top-down satellite image of the experiment environment used to define the GP input s, along with the paths traversed by the large robot. (b) Boxplot percent error for the 20 path cross-validation results with the large robot. The boxplot show the baseline LVM prediction result (baseline DPM has much higher error), as well as results when using different inputs to the GP (e.g. x, y, s, or x, y, θ, ψ, s, etc.).

APPENDIX C

Derivation of Multi-Task Log Marginal Likelihood Through Eigendecomposition

The derivation of Eqn. (4.10) is provided in this appendix. The derivation is analogous to that of [83], though we provide more thorough details for clarity. We show also the case of noise that is independent between tasks (rather than the structured noise shown in [83]).

Recall that $\bar{\mathbf{K}}_{MT} := \mathbf{B} \otimes K(\bar{X}, \bar{X})$ and the log marginal likelihood [84] in the isotopic multi-task case is [83]:

$$\begin{aligned} L = &-\frac{NM}{2}\ln(2\pi) - \frac{1}{2}\ln|\bar{\mathbf{K}}_{MT} + \bar{\Sigma} \otimes \mathbf{I}_{N\times N}| \\ &- \frac{1}{2}\mathrm{vec}(\bar{\mathbf{Z}})^T(\bar{\mathbf{K}}_{MT} + \bar{\Sigma} \otimes \mathbf{I}_{N\times N})^{-1}\mathrm{vec}(\bar{\mathbf{Z}}). \end{aligned} \tag{C.1}$$

Let $\tilde{\mathbf{B}} = \bar{\Sigma}^{-\frac{1}{2}}\mathbf{B}\bar{\Sigma}^{-\frac{1}{2}}$. In the derivation, the following properties are used [71]:

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$$

$$\mathrm{vec}(\mathbf{AXB}) = \mathbf{B}^T \otimes \mathbf{A}\mathrm{vec}(\mathbf{X})$$

$$|\mathbf{A} \otimes \mathbf{B}| = |\mathbf{A}|^{\mathrm{rank}(\mathbf{B})}|\mathbf{B}|^{\mathrm{rank}(\mathbf{A})}$$

$$(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$$

Then,

$$\begin{aligned}
L =& -\frac{NM}{2}\ln(2\pi) \\
& -\frac{1}{2}\ln|(\bar{\Sigma}^{\frac{1}{2}} \otimes \mathbf{I}_{N\times N})(\tilde{\mathbf{B}} \otimes K(\bar{X},\bar{X}) + \mathbf{I}_{M\times M} \otimes \mathbf{I}_{N\times N})(\bar{\Sigma}^{\frac{1}{2}} \otimes \mathbf{I}_{N\times N})| \\
& -\frac{1}{2}\mathrm{vec}(\bar{\mathbf{Z}})^T(\bar{\Sigma}^{-\frac{1}{2}} \otimes \mathbf{I}_{N\times N})(\tilde{\mathbf{B}} \otimes K(\bar{X},\bar{X}) + \mathbf{I}_{M\times M} \otimes \mathbf{I}_{N\times N})^{-1}(\bar{\Sigma}^{-\frac{1}{2}} \otimes \mathbf{I}_{N\times N})\mathrm{vec}(\bar{\mathbf{Z}}). \\
=& -\frac{NM}{2}\ln(2\pi) - \frac{N}{2}\ln|\bar{\Sigma}| - \frac{1}{2}\ln|\tilde{\mathbf{B}} \otimes K(\bar{X},\bar{X}) + \mathbf{I}_{M\times M} \otimes \mathbf{I}_{N\times N}| \\
& -\frac{1}{2}\mathrm{vec}(\tilde{\mathbf{Z}})^T(\tilde{\mathbf{B}} \otimes K(\bar{X},\bar{X}) + \mathbf{I}_{M\times M} \otimes \mathbf{I}_{N\times N})^{-1}\mathrm{vec}(\tilde{\mathbf{Z}}).
\end{aligned} \tag{C.2}$$

where $\mathrm{vec}\tilde{\mathbf{Z}} = \mathrm{vec}(\bar{\mathbf{Z}}\Sigma^{-\frac{1}{2}})$. The eigenvalue decompositions are then given by $\tilde{\mathbf{B}} = U_{\tilde{B}}S_{\tilde{B}}U_{\tilde{B}}^T$ and $K(\bar{X},\bar{X}) = U_K S_K U_K^T$. Using that fact that $U_{\tilde{B}}$ and U_K are orthogonal matrices,

$$\begin{aligned}
& \tilde{\mathbf{B}} \otimes K(\bar{X},\bar{X}) + \mathbf{I}_{M\times M} \otimes \mathbf{I}_{N\times N} \\
=& U_{\tilde{B}}S_{\tilde{B}}U_{\tilde{B}}^T \otimes U_K S_K U_K^T + \mathbf{I}_{M\times M} \otimes \mathbf{I}_{N\times N} \\
=& (U_{\tilde{B}} \otimes U_K)[S_{\tilde{B}} \otimes S_K + (U_{\tilde{B}}^T \otimes U_K^T)(U_{\tilde{B}} \otimes U_K)](U_{\tilde{B}}^T \otimes U_K^T) \\
=& (U_{\tilde{B}} \otimes U_K)[S_{\tilde{B}} \otimes S_K + \mathbf{I}_{M\times M} \otimes \mathbf{I}_{N\times N}](U_{\tilde{B}}^T \otimes U_K^T)
\end{aligned} \tag{C.3}$$

Finally, substituting Eqn. (C.3) into Eqn. (C.2):

$$\begin{aligned}
L =& -\frac{NM}{2}\ln(2\pi) - \frac{1}{2}\ln|S_{\tilde{B}} \otimes S_K + \mathbf{I}_{MN\times MN}| - \frac{N}{2}\ln|\bar{\Sigma}| \\
& -\frac{1}{2}\mathrm{vec}(\tilde{\mathbf{Z}})^T(U_{\tilde{B}} \otimes U_K)(S_{\tilde{B}} \otimes S_K + \mathbf{I}_{MN\times MN})^{-1}(U_{\tilde{B}}^T \otimes U_K^T)\mathrm{vec}(\tilde{\mathbf{Z}}) \\
=& -\frac{NM}{2}\ln(2\pi) - \frac{1}{2}\ln|S_{\tilde{B}} \otimes S_K + \mathbf{I}_{MN\times MN}| - \frac{N}{2}\ln|\bar{\Sigma}| \\
& -\frac{1}{2}\mathrm{vec}(U_K^T\tilde{\mathbf{Z}}U_{\tilde{B}})^T(S_{\tilde{B}} \otimes S_K + \mathbf{I}_{MN\times MN})^{-1}\mathrm{vec}(U_K^T\tilde{\mathbf{Z}}U_{\tilde{B}})
\end{aligned} \tag{C.4}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Abate, A., M. Prandini, J. Lygeros, and S. Sastry (2008), Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems, *Automatica*, *44*(11), 2724–2734.

[2] Agha-mohammadi, A.-a., S. Agarwal, S.-K. Kim, S. Chakravorty, and N. M. Amato (2018), Slap: Simultaneous localization and planning under uncertainty via dynamic replanning in belief space, *IEEE Transactions on Robotics*, *34*(5), 1195–1214.

[3] Akametalu, A. K., J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin (2014), Reachability-based safe learning with gaussian processes, in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 1424–1431, IEEE.

[4] Alvarez, M. A., L. Rosasco, N. D. Lawrence, et al. (2012), Kernels for vector-valued functions: A review, *Foundations and Trends® in Machine Learning*, *4*(3), 195–266.

[5] Aoude, G. S., B. D. Luders, J. M. Joseph, N. Roy, and J. P. How (2013), Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns, *Autonomous Robots*, *35*(1), 51–76.

[6] Axelrod, B., L. P. Kaelbling, and T. Lozano-Pérez (2018), Provably safe robot navigation with obstacle uncertainty, *The International Journal of Robotics Research*, *37*(13-14), 1760–1774.

[7] Bai, H., S. Cai, N. Ye, D. Hsu, and W. S. Lee (2015), Intention-aware online pomdp planning for autonomous driving in a crowd, in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 454–460, IEEE.

[8] Bartlett, O., C. Gurau, L. Marchegiani, and I. Posner (2016), Enabling intelligent energy management for robots using publicly available maps, in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 2224–2229, IEEE.

[9] Best, G., and R. Fitch (2016), Probabilistic maximum set cover with path constraints for informative path planning, in *Australasian Conference on Robotics and Automation*, ARAA.

[10] Binney, J., and G. S. Sukhatme (2012), Branch and bound for informative path planning, in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2147–2154, IEEE.

[11] Binney, J., A. Krause, and G. S. Sukhatme (2013), Optimizing waypoints for monitoring spatiotemporal phenomena, *The International Journal of Robotics Research*, *32*(8), 873–888.

[12] Bonilla, E. V., K. M. Chai, and C. Williams (2008), Multi-task gaussian process prediction, in *Advances in neural information processing systems*, pp. 153–160.

[13] Briol, F.-X., C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic (2016), Probabilistic integration: A role for statisticians in numerical analysis?, *stat*, *1050*, 4.

[14] Broderick, J. A., D. M. Tilbury, and E. M. Atkins (2014), Characterizing energy usage of a commercially available ground robot: Method and results, *Journal of Field Robotics*, *31*(3), 441–454.

[15] Broderick, J. A., D. M. Tilbury, and E. M. Atkins (2014), Optimal coverage trajectories for a ugv with tradeoffs for energy and time, *Autonomous Robots*, *36*(3), 257–271.

[16] Bullo, F., E. Frazzoli, M. Pavone, K. Savla, and S. L. Smith (2011), Dynamic vehicle routing for robotic systems, *Proceedings of the IEEE*, *99*(9), 1482–1504.

[17] Canfield, S. L., T. W. Hill, and S. G. Zuccaro (2018), Prediction and experimental validation of power consumption of skid-steer mobile robots in manufacturing environments, *Journal of Intelligent & Robotic Systems*, pp. 1–15.

[18] Chen, B. Y., W. H. Lam, A. Sumalee, Q. Li, H. Shao, and Z. Fang (2013), Finding reliable shortest paths in road networks under uncertainty, *Networks and spatial economics*, *13*(2), 123–148.

[19] Chiang, H.-T. L., B. HomChaudhuri, A. P. Vinod, M. Oishi, and L. Tapia (2017), Dynamic risk tolerance: Motion planning by balancing short-term and long-term stochastic dynamic predictions, in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 3762–3769, IEEE.

[20] Christie, G., A. Shoemaker, K. Kochersberger, P. Tokekar, L. McLean, and A. Leonessa (2017), Radiation search operations using scene understanding with autonomous uav and ugv, *Journal of Field Robotics*, *34*(8), 1450–1468.

[21] Chung, J. J., A. J. Smith, R. Skeele, and G. A. Hollinger (2018), Risk-aware graph search with dynamic edge cost discovery, *The International Journal of Robotics Research*.

[22] Cressie, N., and C. K. Wikle (2015), *Statistics for spatio-temporal data*, John Wiley & Sons.

[23] Deusdado, P., et al. (2016), An aerial-ground robotic team for systematic soil and biota sampling in estuarine mudflats, in *Robot 2015: Second Iberian Robotics Conference*, pp. 15–26, Springer.

[24] Dey, D., A. Kolobov, R. Caruana, E. Kamar, E. Horvitz, and A. Kapoor (2014), Gauss meets canadian traveler: shortest-path problems with correlated natural dynamics, in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pp. 1101–1108, International Foundation for Autonomous Agents and Multiagent Systems.

[25] Dijkstra, E. W. (1959), A note on two problems in connexion with graphs, *Numerische mathematik*, *1*(1), 269–271.

[26] Dogru, S., and L. Marques (2015), Towards fully autonomous energy efficient coverage path planning for autonomous mobile robots on 3d terrain, in *Mobile Robots (ECMR), 2015 European Conference on*, pp. 1–6, IEEE.

[27] Dogru, S., and L. Marques (2018), Power characterization of a skid-steered mobile field robot with an application to headland turn optimization, *Journal of Intelligent & Robotic Systems*, pp. 1–15.

[28] Du Toit, N. E., and J. W. Burdick (2011), Probabilistic collision checking with chance constraints, *IEEE Transactions on Robotics*, *27*(4), 809–815.

[29] Dunbabin, M., and L. Marques (2012), Robots for environmental monitoring: Significant advancements and applications, *Robotics & Automation Magazine, IEEE*, *19*(1), 24–39.

[30] Dürichen, R., M. A. Pimentel, L. Clifton, A. Schweikard, and D. A. Clifton (2015), Multitask gaussian processes for multivariate physiological time-series analysis, *IEEE Transactions on Biomedical Engineering*, *62*(1), 314–322.

[31] Edwards, D. J., A. D. Kahn, M. Kelly, S. Heinzen, D. A. Scheiman, P. P. Jenkins, R. Walters, and R. Hoheisel (2016), Maximizing net power in circular turns for solar and autonomous soaring aircraft, *Journal of Aircraft*, pp. 1–11.

[32] Esmaeil Zadeh Soudjani, S., and A. Abate (2013), Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes, *SIAM Journal on Applied Dynamical Systems*, *12*(2), 921–956.

[33] Fathpour, N., L. Blackmore, Y. Kuwata, C. Assad, M. T. Wolf, C. Newman, A. Elfes, and K. Reh (2014), Feasibility studies on guidance and global path planning for wind-assisted montgolfiere in titan, *IEEE Systems Journal*, *8*(4), 1112–1125.

[34] Gelman, A., H. S. Stern, J. B. Carlin, D. B. Dunson, A. Vehtari, and D. B. Rubin (2013), *Bayesian data analysis*, Chapman and Hall/CRC.

[35] Gunter, T., M. A. Osborne, R. Garnett, P. Hennig, and S. J. Roberts (2014), Sampling for inference in probabilistic models with fast bayesian quadrature, in *Advances in neural information processing systems*, pp. 2789–2797.

[36] Herbrich, R., N. D. Lawrence, and M. Seeger (2003), Fast sparse gaussian process methods: The informative vector machine, in *Advances in neural information processing systems*, pp. 625–632.

[37] Hollinger, G. A., and G. S. Sukhatme (2014), Sampling-based robotic information gathering algorithms, *The International Journal of Robotics Research*, *33*(9), 1271–1287.

[38] HomChaudhuri, B., A. P. Vinod, and M. M. Oishi (2017), Computation of forward stochastic reach sets: Application to stochastic, dynamic obstacle avoidance, in *American Control Conference (ACC), 2017*, pp. 4404–4411, IEEE.

[39] Huber, M. F. (2014), Recursive gaussian process: On-line regression and learning, *Pattern Recognition Letters*, *45*, 85–91.

[40] Inanc, T., S. C. Shadden, and J. E. Marsden (2005), Optimal trajectory generation in ocean flows, in *American Control Conference, 2005. Proceedings of the 2005*, pp. 674–679, IEEE.

[41] Ishigami, G., K. Nagatani, and K. Yoshida (2007), Path planning for planetary exploration rovers and its evaluation based on wheel slip dynamics, in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 2361–2366, IEEE.

[42] Karaman, S., and E. Frazzoli (2011), Sampling-based algorithms for optimal motion planning, *The international journal of robotics research*, *30*(7), 846–894.

[43] Kavraki, L. E., P. Svestka, J.-C. Latombe, and M. H. Overmars (1996), Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE transactions on Robotics and Automation*, *12*(4), 566–580.

[44] Kochersberger, K., K. Kroeger, B. Krawiec, E. Brewer, and T. Weber (2014), Post-disaster remote sensing and sampling via an autonomous helicopter, *Journal of Field Robotics*, *31*(4), 510–521.

[45] Kuwata, Y., L. Blackmore, M. Wolf, N. Fathpour, C. Newman, and A. Elfes (2009), Decomposition algorithm for global reachability analysis on a time-varying graph with an application to planetary exploration, in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 3955–3960, IEEE.

[46] Lee, S. U., R. Gonzalez, and K. Iagnemma (2016), Robust sampling-based motion planning for autonomous tracked vehicles in deformable high slip terrain,

in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 2569–2574, IEEE.

[47] Lim, S., H. Balakrishnan, D. Gifford, S. Madden, and D. Rus (2011), Stochastic motion planning and applications to traffic, *The International Journal of Robotics Research*, *30*(6), 699–712.

[48] Lin, X., H. E. Perez, J. B. Siegel, A. G. Stefanopoulou, Y. Li, R. D. Anderson, Y. Ding, and M. P. Castanier (2013), Online parameterization of lumped thermal dynamics in cylindrical lithium ion batteries for core temperature estimation and health monitoring, *IEEE Transactions on Control Systems Technology*, *21*(5), 1745–1755.

[49] Liu, H., Y.-S. Ong, X. Shen, and J. Cai (2018), When gaussian process meets big data: A review of scalable gps, *arXiv preprint arXiv:1807.01065*.

[50] Luders, B. D., S. Karaman, and J. P. How (2013), Robust sampling-based motion planning with asymptotic optimality guarantees, in *AIAA Guidance, Navigation, and Control (GNC) Conference*, p. 5097.

[51] Luders, B. D., I. Sugel, and J. P. How (2013), Robust trajectory planning for autonomous parafoils under wind uncertainty, in *AIAA Infotech@ Aerospace (I@ A) Conference*, p. 4584.

[52] Ma, K.-C., L. Liu, H. K. Heidarsson, and G. S. Sukhatme (2018), Data-driven learning and planning for environmental sampling, *Journal of Field Robotics*, *35*(5), 643–661.

[53] Malone, N., H.-T. Chiang, K. Lesser, M. Oishi, and L. Tapia (2017), Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field, *IEEE Transactions on Robotics*, *33*(5), 1124–1138.

[54] Manjanna, S., A. Q. Li, R. N. Smith, I. Rekleitis, and G. Dudek (2018), Heterogeneous multi-robot system for exploration and strategic water sampling, in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, IEEE.

[55] Marom, R., S. F. Amalraj, N. Leifer, D. Jacob, and D. Aurbach (2011), A review of advanced and practical lithium battery materials, *Journal of Materials Chemistry*, *21*(27), 9938–9954.

[56] Martin, S., and P. Corke (2014), Long-term exploration & tours for energy constrained robots with online proprioceptive traversability estimation, in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 5778–5785, IEEE.

[57] Martin, S., and P. Corke (2015), Long term optimisation of a mobile robot with proprioceptive perception, in *Proc. Australasian Conf. Robotics and Automation (ACRA15), Canberra, Australia.*

[58] Mathew, N., S. L. Smith, and S. L. Waslander (2015), Planning paths for package delivery in heterogeneous multirobot teams, *IEEE Transactions on Automation Science and Engineering*, *12*(4), 1298–1308.

[59] Maturana, D., P.-W. Chou, M. Uenoyama, and S. Scherer (2018), Real-time semantic mapping for autonomous off-road navigation, in *Field and Service Robotics*, pp. 335–350, Springer.

[60] Morales, J., J. L. Martinez, A. Mandow, A. J. García-Cerezo, and S. Pedraza (2009), Power consumption modeling of skid-steer tracked mobile robots on rigid terrain, *IEEE Transactions on Robotics*, *25*(5), 1098–1108.

[61] Mulder, V., S. De Bruin, M. Schaepman, and T. Mayr (2011), The use of remote sensing in soil and terrain mappinga review, *Geoderma*, *162*(1-2), 1–19.

[62] Murphy, L., and P. Newman (2013), Risky planning on probabilistic costmaps for path planning in outdoor environments, *IEEE Transactions on Robotics*, *29*(2), 445–457.

[63] Murphy, L., S. Martin, and P. Corke (2012), Creating and using probabilistic costmaps from vehicle experience, in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 4689–4694, IEEE.

[64] Nikolova, E., and D. R. Karger (2008), Route planning under uncertainty: The canadian traveller problem., in *AAAI*, pp. 969–974.

[65] Nikolova, E., M. Brand, and D. R. Karger (2006), Optimal route planning under uncertainty., in *ICAPS*, vol. 6, pp. 131–141.

[66] Ojeda, L., J. Borenstein, G. Witus, and R. Karlsen (2006), Terrain characterization and classification with a mobile robot, *Journal of Field Robotics*, *23*(2), 103–122.

[67] Oliveira, R., L. Ott, and F. Ramos (2016), Active perception for modelling energy consumption in off-road navigation, in *Australasian Conference on Robotics and Automation (ACRA). Brisbane, QLD, Australia.*

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

[69] Pan, S., and G. Ishigami (2017), Strategy optimization for energy efficient extraterrestrial drilling using combined power map, *IEEE Robotics and Automation Letters*, *2*(4), 1980–1987.

[70] Pentzer, J., K. Reichard, and S. Brennan (2016), Energy-based path planning for skid-steer vehicles operating in areas with mixed surface types, in *American Control Conference (ACC), 2016*, pp. 2110–2115, IEEE.

[71] Petersen, K. B., M. S. Pedersen, et al. (2008), The matrix cookbook, *Technical University of Denmark*, *7*(15), 510.

[72] Pinheiro, J. C., and D. M. Bates (1996), Unconstrained parametrizations for variance-covariance matrices, *Statistics and computing*, *6*(3), 289–296.

[73] Platt, R., L. Kaelbling, T. Lozano-Perez, and R. Tedrake (2017), Efficient planning in non-gaussian belief spaces and its application to robot grasping, in *Robotics Research*, pp. 253–269, Springer.

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

[75] Plonski, P. A., J. Vander Hook, and V. Isler (2016), Environment and solar map construction for solar-powered mobile systems, *IEEE Transactions on Robotics*, *32*(1), 70–82.

[76] Preparata, F. P., and M. I. Shamos (2012), *Computational geometry: an introduction*, Springer Science & Business Media.

[77] Quann, M., L. Ojeda, W. Smith, D. Rizzo, M. Castanier, and K. Barton (), Off-road ground robot path energy cost predictionthrough probabilistic spatial mapping, *(Under Review)*.

[78] Quann, M., L. Ojeda, W. Smith, D. Rizzo, M. Castanier, and K. Barton (), Power prediction for heterogeneous ground robots through spatialmapping and sharing of terrain data, *(In Preparation)*.

[79] Quann, M., L. Ojeda, W. Smith, D. Rizzo, M. Castanier, and K. Barton (2017), An energy-efficient method for multi-robot reconnaissance in an unknown environment, in *American Control Conference (ACC), 2017*, pp. 2279–2284, IEEE.

[80] Quann, M., L. Ojeda, W. Smith, D. Rizzo, M. Castanier, and K. Barton (2018), Ground robot terrain mapping and energy prediction in environments with 3-d topography, in *2018 Annual American Control Conference (ACC)*, pp. 3532–3537, IEEE.

[81] Quann, M., L. Ojeda, W. Smith, D. Rizzo, M. Castanier, and K. Barton (2019), Chance constrained reachability in environments with spatially varying energy costs, *Robotics and Autonomous Systems*.

[82] Rajamani, R. (2011), *Vehicle dynamics and control*, Springer Science & Business Media.

[83] Rakitsch, B., C. Lippert, K. Borgwardt, and O. Stegle (2013), It is all in the noise: Efficient multi-task gaussian process inference with structured residuals, in *Advances in neural information processing systems*, pp. 1466–1474.

[84] Rasmussen, C. E., and C. K. I. Williams (2006), *Gaussian Processes for Machine Learning*, the MIT Press.

[85] Richardson, R. R., M. A. Osborne, and D. A. Howey (2017), Gaussian process regression for forecasting battery state of health, *Journal of Power Sources*, *357*, 209–219.

[86] Roldán, J., P. Garcia-Aunon, M. Garzón, J. de León, J. del Cerro, and A. Barrientos (2016), Heterogeneous multi-robot system for mapping environmental variables of greenhouses, *Sensors*, *16*(7), 1018.

[87] Sadrpour, A., J. J. Jin, and A. G. Ulsoy (2013), Mission energy prediction for unmanned ground vehicles using real-time measurements and prior knowledge, *Journal of Field Robotics*, *30*(3), 399–414.

[88] Sadrpour, A., J. Jin, and A. G. Ulsoy (2014), Real-time energy-efficient path planning for unmanned ground vehicles using mission prior knowledge, *International Journal of Vehicle Autonomous Systems*, *12*(3), 221–246.

[89] Salama, M. A., V. V. Vantsevich, T. R. Way, and D. J. Gorsich (2018), Ugv with a distributed electric driveline: Controlling for maximum slip energy efficiency on stochastic terrain, *Journal of Terramechanics*, *79*, 41–57.

[90] Sapaty, P. (2015), Military robotics: latest trends and spatial grasp solutions, *International Journal of Advanced Research in Artificial Intelligence*, *4*(4), 9–18.

[91] Schillinger, P., M. Bürger, and D. V. Dimarogonas (2018), Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems, *The international journal of robotics research*, *37*(7), 818–838.

[92] Senarathne, P., and D. Wang (2015), Incremental algorithms for safe and reachable frontier detection for robot exploration, *Robotics and Autonomous Systems*, *72*, 189–206.

[93] Seshadri, R., and K. K. Srinivasan (2010), Algorithm for determining most reliable travel time path on network with normally distributed and correlated link travel times, *Transportation Research Record*, *2196*(1), 83–92.

[94] Shah, A., A. Wilson, and Z. Ghahramani (2014), Student-t processes as alternatives to gaussian processes, in *Artificial Intelligence and Statistics*, pp. 877–885.

[95] Singh, A., A. Krause, C. Guestrin, and W. J. Kaiser (2009), Efficient informative sensing using multiple robots, *Journal of Artificial Intelligence Research*, pp. 707–755.

[96] Sipahioglu, A., G. Kirlik, O. Parlaktuna, and A. Yazici (2010), Energy constrained multi-robot sensor-based coverage path planning using capacitated arc routing approach, *Robotics and Autonomous Systems*, *58*(5), 529–538.

[97] Snelson, E., and Z. Ghahramani (2006), Sparse gaussian processes using pseudo-inputs, in *Advances in neural information processing systems*, pp. 1257–1264.

[98] Solin, A., M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä (2018), Modeling and interpolation of the ambient magnetic field by gaussian processes, *IEEE Transactions on robotics*, *34*(4), 1112–1127.

[99] Stachniss, C., C. Plagemann, A. J. Lilienthal, and W. Burgard (2008), Gas distribution modeling using sparse gaussian process mixture models., in *Robotics: Science and Systems*.

[100] Stegle, O., C. Lippert, J. M. Mooij, N. D. Lawrence, and K. Borgwardt (2011), Efficient inference in matrix-variate gaussian models with\iid observation noise, in *Advances in neural information processing systems*, pp. 630–638.

[101] Subramani, D. N., and P. F. Lermusiaux (2016), Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization, *Ocean Modelling*, *100*, 57–77.

[102] Subramani, D. N., Q. J. Wei, and P. F. Lermusiaux (2018), Stochastic time-optimal path-planning in uncertain, strong, and dynamic flows, *Computer Methods in Applied Mechanics and Engineering*, *333*, 218–237.

[103] Suh, J., J. Gong, and S. Oh (2017), Fast sampling-based cost-aware path planning with nonmyopic extensions using cross entropy, *IEEE Transactions on Robotics*, *33*(6), 1313–1326.

[104] Sun, Z., and J. H. Reif (2005), On finding energy-minimizing paths on terrains, *IEEE Transactions on Robotics*, *21*(1), 102–114.

[105] Tokekar, P., J. Vander Hook, D. Mulla, and V. Isler (2016), Sensor planning for a symbiotic uav and ugv system for precision agriculture, *IEEE Transactions on Robotics*, *32*(6), 1498–1511.

[106] Tomlin, C. J., I. Mitchell, A. M. Bayen, and M. Oishi (2003), Computational techniques for the verification of hybrid systems, *Proceedings of the IEEE*, *91*(7), 986–1001.

[107] Tsiogkas, N., and D. M. Lane (2018), An evolutionary algorithm for online, resource-constrained, multivehicle sensing mission planning, *IEEE Robotics and Automation Letters*, *3*(2), 1199–1206.

[108] Urcola, P., M. T. Lázaro, J. A. Castellanos, and L. Montano (2017), Cooperative minimum expected length planning for robot formations in stochastic maps, *Robotics and Autonomous Systems*, *87*, 38–50.

[109] Valada, A., and W. Burgard (2017), Deep spatiotemporal models for robust proprioceptive terrain classification, *The International Journal of Robotics Research*, p. 0278364917727062.

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

[111] Wackernagel, H. (2013), *Multivariate geostatistics: an introduction with applications*, Springer Science & Business Media.

[112] Wang, L., A. Ames, and M. Egerstedt (2016), Safety barrier certificates for heterogeneous multi-robot systems, in *2016 American Control Conference (ACC)*, pp. 5213–5218, IEEE.

[113] Wang, Y., J. B. Siegel, and A. G. Stefanopoulou (2016), Control strategies for power quantized solid oxide fuel cell hybrid powertrains: In mobile robot applications, *SAE International Journal of Alternative Powertrains*, *5*(1), 58–67.

[114] Williams, C., S. Klanke, S. Vijayakumar, and K. M. Chai (2009), Multi-task gaussian process learning of robot inverse dynamics, in *Advances in Neural Information Processing Systems*, pp. 265–272.

[115] Wong, J. Y. (2009), *Terramechanics and off-road vehicle engineering: terrain behaviour, off-road vehicle performance and design*, Butterworth-heinemann.

[116] Wu, C., C. Dai, X. Gong, Y.-J. Liu, J. Wang, X. Gu, and C. C. Wang (2019), Energy-efficient coverage path planning for general terrain surfaces, *IEEE Robotics and Automation Letters*.

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

[118] Xu, Y., J. Choi, and S. Oh (2011), Mobile sensor network navigation using gaussian processes with truncated observations, *Robotics, IEEE Transactions on*, *27*(6), 1118–1131.

[119] Yu, J., M. Schwager, and D. Rus (2016), Correlated orienteering problem and its application to persistent monitoring tasks, *IEEE Transactions on Robotics*, *32*(5), 1106–1118.

[120] Zeng, W., T. Miwa, Y. Wakita, and T. Morikawa (2015), Application of lagrangian relaxation approach to α-reliable path finding in stochastic networks with correlated link travel times, *Transportation Research Part C: Emerging Technologies*, *56*, 309–334.

[121] Zhang, R., and M. Pavone (2016), Control of robotic mobility-on-demand systems: a queueing-theoretical perspective, *The International Journal of Robotics Research*, *35*(1-3), 186–203.